Atas da 12ª Conferência sobre Redes de Computadores (CRC2012)

15-16 de Novembro de 2012, Aveiro, Portugal

apoios:

instituto de telecomunicações

universidade de aveiro

Editores

R. Aguiar
A. Martins
D. Gomes
P. Salvador
P. Pedreiras
A. Nogueira

Dep. Electrónica, Telecomunicações e Informática, Universidade de Aveiro

Atas da 12ª Conferência sobre Redes de Computadores (CRC 2012)

ISBN : 978-972-789-369-0

Nota de Boas-vindas

Foi com particular entusiasmo que o Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, com o apoio do Instituto de Telecomunicações, organizou esta 12ª Conferência sobre Redes de Computadores – CRC'2012!

Com início em 1998, a Conferência sobre Redes de Computadores – CRC – tem sido uma constante na vida da comunidade de Redes nacional, tendo já passado por imensos locais no país: Coimbra, Évora, Viseu, Covilhã, Faro, Bragança, Leiria, Portalegre, Oeiras e Braga. A CRC tem assumido um papel único na divulgação de trabalhos de Investigação, Desenvolvimento e Inovação (I&D+I) na área das comunicações por computador e redes.
Em 2012, esta 12ª Conferência sobre Redes de Computadores tem sido organizada debaixo de um ambiente particularmente complicado para a comunidade científica. As restrições orçamentais que enfrentamos são bem presentes no dia a dia de cada um de nós. No entanto, verificamos que continuamos a ser solicitados para trabalho de investigação e desenvolvimento nas nossas áreas! Embora o enquadramento financeiro seja crescentemente restritivo, temos um conjunto cada vez maior de desafios de índole técnica. A CRC'2012, abre assim as portas a uma temática crescente, cobrindo desde o desenvolvimento de redes de nova geração, as tecnologias de infra-estrutura emergentes, os serviços e infra-estruturas de Cloud, até aos sistemas de tempo-real. A CRC'2012 constituiu assim uma oportunidade especial para estudantes de mestrado e doutoramento compreenderem o dinamismo da nossa área. Nesse sentido, começamos uma sessão, que esperamos que se mantenha, de "Revista do ano", onde se poderá apresentar os pontos altos de investigação na área em Portugal durante este ano, e que demonstrará a vitalidade da nossa área.
Em termos de programa técnica, após cuidada revisão, foram seleccionados 16 artigos de elevada qualidade que foram apresentados nas sessões temáticas em que a Conferência foi organizada – Cloud; Redes Móveis; Segurança; Redes Veiculares; e Redes de Sensores – e que se encontram reunidos e editados neste livro de Actas. Estas sessões proporcionam uma excelente oportunidade para debate e aproximação entre a academia e os profissionais da indústria das redes e serviços de comunicações em Portugal.

Expressam-se os agradecimentos a todos os que contribuíram para este processo, especialmente a todos os colegas da Comissão Científica de Programa da CRC'2012, aos elementos da Comissão Coordenadora das CRCs, aos colegas do ATNOG (Advanced Telecommunications and Networking Group) e a toda a Comissão Organizadora desta 12ª Conferência sobre Redes de Computadores. Expressa-se também um agradecimento especial ao Instituto de Telecomunicações, por todo o apoio logístico.

Com os votos de uma Excelente e Vibrante CRC'2012

R. Aguiar
A. Martins
D. Gomes
P. Salvador
P. Pedreiras
A. Nogueira

Comissão Científica

Rui Aguiar	Universidade de Aveiro/IT
José Alberto Fonseca	Universidade de Aveiro/IT
Luís Almeida	Universidade do Porto
Luís Bernardo	Universidade Nova de Lisboa
Fernando Boavida	Universidade de Coimbra
Paulo Carvalho	Universidade do Minho
Augusto Casaca	Instituto Superior Técnico
António Casimiro	Universidade de Lisboa
Eduardo Cerqueira	Universidade Federal do Pará (Brasil)
Marília Curado	Universidade de Coimbra
Amaro de Sousa	Universidade de Aveiro/IT
Bruno Dias	Universidade do Minho
Joaquim Ferreira	Universidade de Aveiro/IT
Luís Lino Ferreira	Instituto Politécnico do Porto
Mário Freire	Universidade da Beira Interior
Diogo Gomes	Universidade de Aveiro/IT
Pedro Gonçalves	Universidade de Aveiro/IT
José Legauteaux Martins	Universidade Nova de Lisboa
Joaquim Macedo	Universidade do Minho
Arnaldo Martins	Universidade de Aveiro
Paulo Mendes	Universidade Lusófona
Edmundo Monteiro	Universidade de Coimbra
Adriano Moreira	Universidade do Minho
Maria João Nicolau	Universidade do Minho
Antonio Nogueira	Universidade de Aveiro/IT
José Luis Oliveira	Universidade de Aveiro
Paulo Pedreiras	Universidade de Aveiro/IT
Carlos Eduardo Pereira	Universidade Federal do Rio Grande do Sul (Brasil)
Paulo Pereira	Instituto Superior Técnico
Paulo Pinto	Universidade Nova de Lisboa
Paulo Portugal	Universidade do Porto
Manuel Ricardo	Universidade do Porto/INESC
Solange Rito	Universidade do Minho
Joel Rodrigues	Universidade da Beira Interior/IT
Luís Rodrigues	Instituto Superior Técnico
José Ruela	Universidade do Porto
Paulo Salvador	Universidade de Aveiro/IT
Alexandre Santos	Universidade do Minho
Susana Sargento	Universidade de Aveiro/IT
Paulo Simões	Universidade de Coimbra
Carlos Sá Da Costa	ISCTE - Instituto Universitário de Lisboa
Jorge Sá Silva	Universidade de Coimbra
Eduardo Tovar	Instituto Politécnico do Porto
António Varela	Instituto Superior Técnico
Teresa Vazão	Instituto Superior Técnico
André Zuquete	Universidade de Aveiro

Outros Revisores

André Cardote	Universidade de Aveiro
Tiago Cruz	Universidade de Coimbra
Carlos Gonçalves	Universidade do Aveiro
Tauseef Jamal	Universidade Lusfona
Waldir Moreira	Universidade Lusfona
Andre Riker	Universidade de Coimbra

Índice

Sessão 1 - Cloud

Towards a Cloud Service Broker for the Meta-Cloud 7
Carlos Gonçalves, David Cunha, Pedro Neves, Pedro Sousa, João Paulo Barraca and Diogo Gomes

Uma Abordagem Estratificada à Monitorização de Serviços Cloud 14
Nuno Palhares and Solange Lima

Network Virtualization - A Virtual Router Performance Evaluation 23
Bruno Parreira, Márcio Melo, João Soares, Jorge Carapinha, Susana Sargento and Romeu Monteiro

Sessão 2 - Redes Moveis

Protocolo de encaminhamento para redes móveis usando estruturas binárias eficientes 30
João Trindade and Teresa Vazão

EMICOM: Enhanced Media Independent COnnection Manager 38
André Prata, Daniel Corujo, Pedro Gonçalves and Diogo Gomes

Analysis of the logical proximity between 802.11 access points 47
Ricardo Sousa and Ricardo Morla

Encaminhamento Anycast em Redes IPv6: uma proposta 56
Hugo Ferreira, Maria João Nicolau and António Costa

Sessão 3 - Segurança

Detection of WPS Attacks Through Multiscale Analysis 65
Ivo Petiz, Eduardo Rocha, Paulo Salvador and António Nogueira

Multipass: Autenticação Mútua em Cenários Heterogéneos 70
Rui Ferreira, André Tomás, Pedro Estima, Rui Aguiar and Ricardo Azevedo

Sessão 4 - Redes Veiculares

Protocolos de Encaminhamento para Redes Veiculares com Ligações Intermitentes 77
Vasco Soares, João Dias, João Dias and Joel Rodrigues

Service Platform for Vehicular Networks 85
Pedro Cruz Sousa and Teresa Vazao

Simulação do uso de redes veiculares em situações de emergência numa auto-estrada Portuguesa 94
Teresa Vazao and Jacqueline Jardim

Sessão 5 - Redes de Sensores

On Multipath Aggregation of Dupicate Sensitive Summaries 103
Milton Cunguara, Tomás Oliveira E Silva and Paulo Pedreiras

Architecture for orchestration of M2M services 112
Gustavo Pires, Mário Antunes, Daniel Corujo, Diogo Gomes, João Paulo Barraca and Rui Aguiar

Adaptive Multimedia Transmission in Wireless Sensor Networks 120
Daniel G. Costa, Luiz Affonso Guedes, Francisco Vasques and Paulo Portugal

Estratégia centralizada energeticamente eficiente para RSSF heterogêneas utilizando Lógica Fuzzy 126
Christiano Maciel, Tassio Carvalho, José Junior, Alexandre Melo and Carlos Tavares

Towards a Cloud Service Broker for the Meta-Cloud

Carlos Gonçalves*, David Cunha†‡, Pedro Neves†, Pedro Sousa‡, João Paulo Barraca*, Diogo Gomes*
*Instituto de Telecomunicações, University of Aveiro, 3810-193 Aveiro, Portugal
{cgoncalves,jpbarraca,dgomes}@av.it.pt
†Portugal Telecom Inovação, SA, 3810-106 Aveiro, Portugal
{david-g-cunha,pedro-m-neves}@ptinovacao.pt
‡Centro ALGORITMI / Department of Informatics, University of Minho, 4710-057 Braga, Portugal
pns@di.uminho.pt

***Abstract*—Cloud Computing provides computing resources, middleware and (web-based) software on an on-demand basis. This model helps customers saving costs and allows access to the latest technology. With the exponential growth of IT companies offering cloud services, deploying applications to the cloud has become a complex task to engage. Almost each and every provider has its own terminology, providers do not share the same (or even similar) API, and costs of operation greatly diverge according to provider, region or availability.**

This paper propounds a Cloud Service Broker (CSB), and describes an early prototype, where users are, intelligently and autonomously, aid to deploy, manage, monitor and migrate their applications in a cloud of clouds. A single API is required to orchestrate the whole process in tandem with two truly decoupled managers: a Platform as a Service Manager (PaaS Manager) and an Infrastructure as Service Manager (Iaas Manager). Users also interact with the CSB through a Web portal and a command-line interface.

***Index Terms*—Cloud Service Broker, IaaS, PaaS, Cloud Interoperability, SOA**

I. Introduction

Cloud Computing is an emerging technology used to deliver on-demand services over the Internet. It is undoubtedly affecting the way business is conducted and is empowering a new generation of products and services. Cloud Computing can be summarized into three keywords: elasticity, on-demand, and (autonomously) fully-managed [1]. These three characteristics massively benefit organizations by reducing both CAPEX and OPEX while enabling them to channel their efforts to the strategic business sector [2].

Business models for cloud computing present the resulting system according to three major layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [3]. Several companies such as Amazon, RackSpace, Google, and Microsoft, expose their cloud solutions publicly as business-to-client and business-to-business models. These companies provide on-demand cloud services through a diverse set of business models and prepositions. Availability, continuity, API support, location, charging scheme and price, as well as technical characteristics such as computational capabilities, storage and processing memory, are some of the parameters of the resulting business offers.

Consider the following short list of presently available proprietary and open source PaaS software platformas available for public consumption. Heroku[1] is a well known cloud application provider supporting a variety of runtime languages including the Java, Ruby, Python and Node.js frameworks as add-ons. Microsoft's Windows Azure[2] is a 99% Service-Level Agreement (SLA) cloud platform proportioning users with extended services like Content Delivery Network (CDN), application load balancing, and service bus. On the open source side of PaaS platforms, efflorescent projects has come to light recently with a successful prospective enabling cloud provider enterprises to embrace such projects and contribute back to a better product, as well as enabling users to run their own PaaS instance on a private cloud. Cloud Foundry[3] is a open source platform project initiated by VMWare Inc. Cloud Foundry strives to be multi-language, multi-framework, multi-service, and multi-cloud. It is also in its goals to be horizontally scalable and self healing with no single point of failures, and fail fast, following basic patterns–event-driven, asynchronous, and non-blocking [4]. All these solutions are limited because cross-cloud management functionalities are lacking [5], and they are focused in interoperation, forgetting about many self-healing and business continuity aspects. Cloud consumers are therefore faced with the problem of choosing the best provider, considering the factors identified as critical for each owned system and individual applications.

The framework we propose addresses the aforementioned issue by aiming to develop a Cloud Service Broker (CSB) which eases crucial lifecycle application stages (deployment, management, and monitoring) from developers orchestrating the whole process in tandem with a PaaS Manager and an IaaS Manager. To achieve such goal, the CSB should be aware of user application requirements, be aware of available cloud offers, and transparently deploy to the best rated cloud provider based on given premises, being either and ideally a PaaS or if premises are not fulfilled, falling back to an IaaS solution in which a PaaS framework is setup on demand. Through the proposed CSB, end-user developers will have access to a feature-complete Web portal where they can auto-deploy, manage and monitor their appliances. Ultimately, the CSB allows the integration of intelligent mechanisms towards

[1]http://www.heroku.com/
[2]http://www.windowsazure.com
[3]http://www.cloudfoundry.com/

the live migration of SaaS between PaaS and live migration of PaaS between IaaS, in such a way that it will self heal from a misbehaving provider or in response to localized service demand, or cost cutting by switching-over to a less expensive provider.

The remainder of this paper is structured as follows. Section II briefly overviews the state of the art in the cloud computing area focusing on cloud mediators, decision making helpers and cloud API standards. Section III proposes a novel framework solution for the needs identified, which by taking a bottom-up approach describes two new decoupled products, the PaaS Manager and IaaS Manager. Finally, Section IV concludes the work presented.

II. Related Work

Industry and academic related work have been published throughout the last two years actively. Work includes, but not restricted to, decision making engines for cloud services, unified multi cloud management systems and cloud brokering. The majority are focused on IT infrastructure.

The work in [6] discusses a methodology for a multi-criteria cloud service selection. Cloudle [7], another multi-criteria engine, uses cloud ontologies for IaaS and PaaS, amongst others "as a Service" cloud offerings. On multi cloud manageability [8] proposes a platform that mediates between users and IaaS resources and hence reducing workload for IT operators by only requiring a single sign-on in one unified portal. A proven implementation of this kind of cloud management concept is Red Hat's sponsored Aeolus project[4].

An interesting reading is [9]. It identifies and compares common use cases on different IaaS and PaaS cloud platforms. Interfaces, documentation, ease of use and time taken to learn, and number of steps required to accomplish a task are measured.

Because hardly any PaaS offering will ever satisfy every application or system requirement, a new approach needs to be considered to fill in the gap. An approach which really integrates different cloud architectures, providing enhanced management capabilities. A first approach is to interoperate services across different providers creating a whole new and tailor-oriented PaaS per tenant. However, in order to consider a broader approach, we can not safeguard an absolute coverage of all functional prerequisites. As such an extended approach has to be contemplated to tackle the downside of the former strategy. The proposed approach presented in this paper requires the use of less abstract interfaces to the cloud, namely through the use of IaaS solutions, and perform some automatic and transparent work from the point of view of the cloud consumers. The objective is to meet the requirements specified by users from the set of available cloud providers, which could not be accomplished by existing single PaaS offerings.

Also considerable efforts on the cloud API standardization have been conducted by industry organizations. They are committed to unify incompatible APIs from various cloud providers as a single one. Examples of well-defined IaaS APIs are Distributed Management Task Force (DMTF)'s Cloud Infrastructure Management Interface (CIMI)[5] and Open Cloud Computing Interface (OCCI)[6] from Open Grid Forum (OGF). These two standards are already being used in a considerable number of projects such as on OpenStack, CloudStack, and OpenNebula. The freshly unveiled Cloud Application Management for Platforms (CAMP)[7], a joint effort by big companies including Oracle, Red Hat, Rackspace and others, strives to bridge the same gap as CIMI and OCCI do though on the PaaS layer of the cloud stack. PaaS management API's CAMP goal is to provide a common ground across multi-cloud solutions. Consumers interact in a REST-based approach, exchanging JSON encoded data for the model resources. Interoperability between platform clouds is made easier.

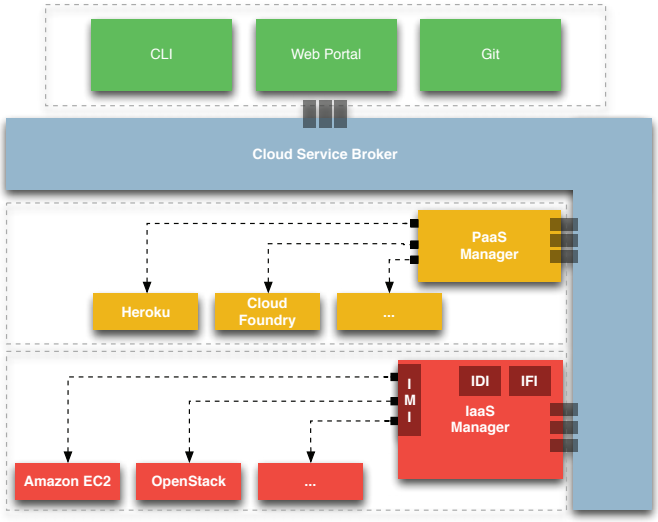


Fig. 1: Architecture overview

III. Proposed Architecture

The hereby presented CSB framework, and depicted in Figure 1, consists on four layers. The PaaS and IaaS managers support multi-provider and multi-cloud environments using each a single API, orchestrated by the intelligent CSB. Users have access to a Web portal and command-line interfaces where they can perform arbitrated operations on resources.

Combining all service oriented components of multiple cloud systems into one single piece of software, provides an efficient and practical platform. This approach is suitable not only for helping users decide which PaaS provider better fits current or future requirements in terms of deployment, execution and monitoring, but also for the case that no single provider fulfills these requirements. Intelligent handling of application specification enables fallback to different PaaS providers or smooth deployment to be built directly in to a IaaS cloud solution.

[4]http://aeolusproject.org

[5]http://dmtf.org/standards/cloud
[6]http://occi-wg.org
[7]http://cloudspecs.org/CAMP/

A. Platform as a Service Manager

The Platform as a Service Manager (PaaSManager), detailed in this section, is a framework which aggregates several PaaS public offerings based on shared similarities. The PaaSManager is intended to provide fundamental features for developers, such as, create, manage, monitor and acquiring information regarding applications and databases. Furthermore, the portability of applications between vendors is a foremost purpose of this architecture. For the development process of such approach, some PaaS offerings were selected to belong to an interoperable ecosystem.

Besides well-known vendors like Amazon, Google and Windows, there are several platforms which offer attractive solutions for developers. The selected platforms, CloudBees, CloudFoundry, IronFoundry and Heroku, were broadly studied as well as their native APIs and management processes. As result, 20 operations were specified based on similarities shared by the platforms from the ecosystem. Some additional developments had to be performed in order to provide a complete transparency for developers, like supporting Git as deployment tool for every PaaS. Figure 2 resumes some of the fundamental operations supported by the PaaSManager.

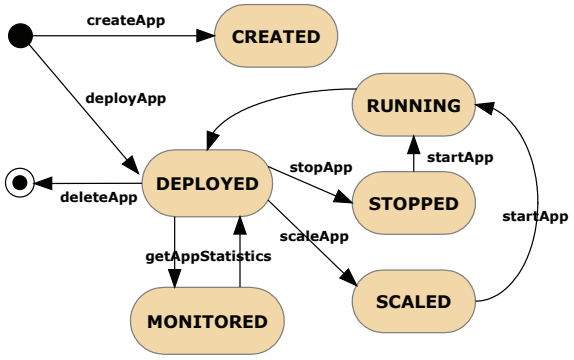


Fig. 2: Application Lifecycle

Besides these operations, the PaaSManager allows to restart, update, migrate applications and create and delete databases. As aforementioned, the portability of applications is also a main goal of this solution. The decision of migrating software to another provider may be based on the demand for better performance or more suitable business models. However, the PaaSManager approach does not intend to define any standard model or support interfaces to migrate, for instance, Java to .NET. Consequently, it requires a prior analysis to check if the new platform will be able to run the application properly. In terms of monitoring, each vendor provides distinct metrics and paradigms. CloudBees and Heroku have partnered with NewRelic for supplying real-time statistics. On the other hand, CloudFoundry and IronFoundry have a specific operation in their native APIs.

In the next section, the PaaSManager architecture will be presented as also all the modules that were designed for supporting the discussed features.

1) PaaSManager Architecture: The PaaSManager architecture is entirely modular so each vendor API has been implemented by different entities that abstract the background processes through a RESTFul interface. Figure 3 presents the framework architecture and respective operational modules.

2) PaaSManager API: The PaaSManager API is REST compliant supporting all the operations aforementioned in a lightweight and web oriented approach. This interface can be easily implemented by any HTTP client, such as, web, desktop or cURL. To be authenticated, the client application only needs to send an *api-key* through all the requests to the API. On the other side, the authentication with vendors is done through a unique account enabling the PaaSManager to act as a mediator between users and PaaS providers. Therefore the developer does not need to register in each vendor for having full access to their services.

3) Management Resources: The Management Resources presented in Figure 3 is a decision module responsible to interact with each PaaS adapter for management tasks (create, deploy, start, stop, etc.). Four adapters were defined, each one for each supported platform: CloudBees Mgmt, CloudFoundry Mgmt, IronFoundry Mgmt and Heroku Mgmt. These adapters implements the operations related to management features and exposed by the vendors APIs. Then, they process the acquired information returning unified responses in JSON or XML representations. Other key elements from the architecture are integrated with the Management Resources and the PaaS adapters. For instance, a central database keeps state of created applications storing the application framework identifier and the vendor name where the application is hosted. Also a Git server maintains all the applications source code repositories that are crucial for some PaaSManager operations, such as, create, deploy, update and migrate applications. Maintaining Git repositories may question the scalability of this architecture, however, to keep the system clean and efficient, the deletion of applications will enforce the removal of unnecessary repositories.

The migration of applications is becoming crucial in developers' point-of-view. In the scope of this work, the PaaSManager performs a prior analysis whether the platform to where the application will be migrated, supports the required technologies for the application to run properly. Figure 4 resumes the migration process and all the performed steps. Firstly, the request done to the PaaSManager API is routed to the Management Resources module, responsible for the migration operation. The database is then queried in order to return the PaaS identification where the application is hosted, as also the supported framework, e.g, Django, Sinatra. The Management Resources calls the specific PaaS adapter which in turn analyzes if the new provider supports the fundamental dependencies. In the case of correlation, the application code is deployed in the new selected PaaS through the required operations processes (create and deploy). Finally, the application deployed in the previous platform is removed and the state information in the database is updated.

Currently, the PaaSManager does not support the migration

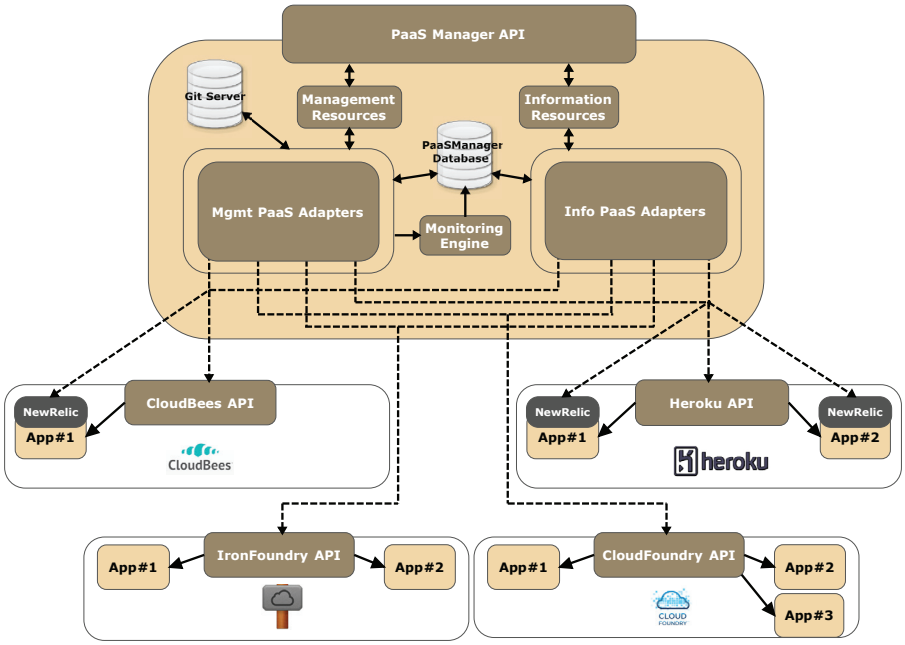


Fig. 3: PaaSManager Architecture

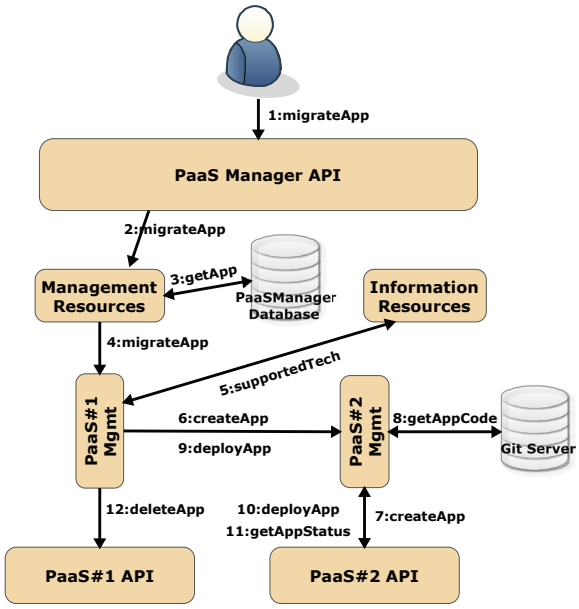


Fig. 4: Migration Process

of applications that require persistence, such as, databases. The limitations of such case would include the synchronizing of data between the two PaaS providers as well as the auto-reconfiguration of the application source code files in order to access the newly created database. However, the PaaSManager has a method that returns the database's access credentials for developers import or export data from the cloud.

4) Information Resources: The Information Resources presented in Figure 3 is a decision module that implements methods related to acquisition of information concerning applications and databases. It also offers four PaaS adapters: CloudBees Info, CloudFoundry Info, IronFoundry Info and Heroku Info, which are responsible for acquiring and processing information related to status, logs or monitoring tasks exposed by each vendor API. This information is fundamental for developers to manage their software and activate the scaling or migration processes.

5) Monitoring Engine: In recent years, some studies were conducted in cloud monitoring area aiming to define monitoring frameworks or metrics models for an efficient cloud management [10]. The metrics list is quite extensive, including availability, response time, RAM, CPU usage, database requests, threads or user sessions [11]. However at the moment, each PaaS provider offers different metrics and different tools for monitoring applications. As discussed previously, CloudBees and Heroku have partnered with NewRelic, which is a popular Application Management Performance used in cloud environments. On the other hand, CloudFoundry and IronFoundry supply a monitoring operation through their native APIs. The Monitoring Engine, presented in Figure 3, was developed in order to collect real-time metrics exhibited by each platform. After the application has been deployed in one of the PaaS, a background job is launched and kept alive until the application is stopped or until it is removed. This process is defined by a synchronous sampling performed every minute towards the NewRelic API or the native API according to the platform where the application is hosted. The achieved information is then stored in the central database and can be queried through the PaaSManager API.

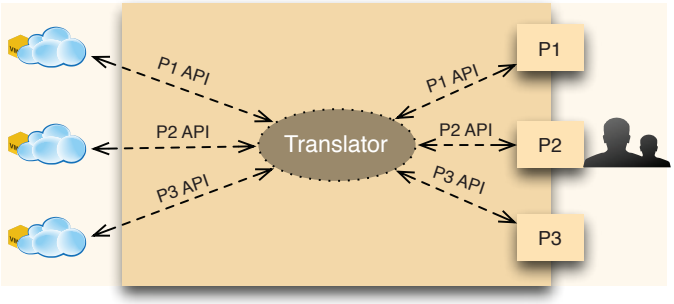


Fig. 5: IaaS Management: the 2-way translator

B. IaaS Manager

As mentioned in Section II, relying entirely on the success of the PaaSM doesn't cover all cloud customers' needs. A failed attempt to migrate an application to a cloud, fallbacks to the lowest cloud layer and results in the need for carrying out an endless low-level set of system administration tasks: provision the virtual machine, install the operating system, install and tweak user's application software dependencies, install security hardening configurations, add monitoring support, etc [12]. These are not the optimal sysadmin actions to be done. In our proposed solution an external entity could help distressing the workload by brokering the relationship between the two worlds. The IaaS Manager (IaaSM) acts as such an external entity.

The IaaSM is composed by three public APIs: IaaS Discovery Interface, IaaS Management Interface, and IaaS Functional Interface. The use of these APIs concedes users consolidated interfaces they can use, despite each cloud providers API variations.

1) IaaS Discovery Interface: The role of the IaaS Discovery Interface (IDI) is to bring forth all the available IaaS cloud providers, their functionalities, geographical location, and service fees. This implies a good entry-point to sort out which provider best suits the customer interests.

2) IaaS Management Interface: The IaaS Management Interface (IMI) unifies all IaaS providers APIs into one single API, abstracting end-users from the differences between clouds and so easily achieving management interoperability. A high-level management interface will provide a full range of operations such as create, start, stop, reboot, and terminate instances, upload and delete images, administer public and private IP addresses, etc. It is imperative to endorse users whom were forced by cloud providers to use proprietary and non-standard compliant interfaces when they moved in to the cloud. An 2-way API translator (Figure 5) equipping users to keep using unchanged tools against a second provider is advisable. Because CIMI helps attain interoperability between cloud service provider APIs and users, it is used as the foundation for the 2-way API translator.

3) IaaS Functional Interface: Once a cloud infrastructure has been created and deployed, users can benefit from the IaaS Functional Interface (IFI) to automate IT operations, from software provisioning and configuration to patching. Utilizing IFI to control how the system gets configured throughout its lifecycle, ensures an automated rebuild of the system in case of disasters or quickly deploys a mirror of the current system state.

C. Cloud Service Broker

The Cloud Service Broker (CSB) is the most important component on this architecture as it connects the PaaSM, IaaSM and user interfaces through a Service Bus. It provides extra high-level services that support all operation and business logic associated to the proposed solution. While the PaaS Manager and the IaaS Manager are self-contained, CSB orchestrates, further augments and eases the process of deploying and running an application in the cloud from the ground up, by advising users which platform better matches their application dependencies. At the same time it enables additional savings in operational expenses by deploying the code to the chosen one, and provisioning the proper monitoring hooks. Also through the use of the CSB, users are qualified to effortlessly migrate their applications from on-premises to the cloud or between cloud providers, with as few modifications as possible. Upon user request, the CSB will auto-scale possible computational resources on the platforms where the application is hosted, in order to accommodate high demand periods in a resilient manner. After peak hour or when demand simply drops, applications can be migrated back to its normal state, in a lower cost cloud provider.

It is known cloud offerings can significantly differ on supported functionality. For instance, there are PaaS services that do not yield monitoring or logging information. Others, while committed to assure no cloud confinement exists, it is only possible to interoperate with another provider if they share the same API which at present implies sharing the exact same platform. The CSB has a database containing data of provider capabilities, including functional, technical and location data. This knowledge is highly import for discerning which characteristics some providers support but others do not and provide means to better advise users with a filtered set of providers matching their application requirements. Section III-C1 further details and demonstrates how this data can be used to users' advantage.

The implementation of the CSB is not narrowed to the greatest common denominator of functionality from supported providers. In case of unsupported attributes from one or more providers, the CSB does not discard those features from all other supported providers. On the web portal, depending on where the application was deployed to, the user will be presented with a richer or fewer set of actions available to their disposal.

1) Manifest and Decision-making assessment: A key component of the system proposed is the concept of a manifest, and its relation to users and cloud providers. A manifest is a structured document delineating application minimal and optimal requirements, in the form of computation, storage, communication, and business metrics. The manifest is supplied

by the developer through the existing interfaces (web portal or CLI). As an example, the content of a manifest can specify that storage latency should not exceed 1ms, that bandwidth required is of 100Mbits, or simply that the application should not leave a given jurisdiction (or even the local premises). Therefore, it effectively consists in the description of the desired Service Level Agreement (SLA) requested to the PaaS provider. Users create a manifest document for each application, and submit it to the CSB. The document is evaluated, and is used during the entire application lifetime. At the first, it will enable the CSB to decide where to deploy the application. Later it enables the CSB to know which metrics must be monitored, the thresholds set by the application, and how to react if values exceed the given thresholds. The manifest response should therefore allow rating cloud offerings, returning individual scores for each application in each PaaS provider.

2) Deployment: Along with the source code, users are asked to inform where the application should be deployed to. The CSB collects the code, creates a revision control and source code management repository and copies the data into it. Based on the manifest, the CSB will request the creation of databases on the desired PaaS and will replace environment variables with the access credentials returned in a temporarily repository, leaving untouched the code previously uploaded to the repository. The rest of the deployment stages will occur transparently to the end-user, taking place in the PaaS Manager and/or the IaaS Manager. Auto redeploy of a new version to the cloud is just a matter of the user submitting changes to the revision control repository and the CSB will take care of everything else from there on.

3) Monitoring and Billing: The monitoring system is of the foremost importance to the CSB. The monitor audits computational resources usage per application and assembles daily, weekly and monthly reports allowing users to control costs of operation over time and covering all supported cloud platforms. Notifications will be sent out in case of events–thresholds exceeded, service outages, etc. Since the broker is a multi-tenancy framework, it will issue invoices per customer, not per application. Monitoring is also important for the CSB to analyze if the requirements specified in the manifest are being fulfilled, and to rank existing providers according to the needs of each application.

4) Migration: By leaving the source code untouched and with the manifest stored on the CSB, future deployments to concurrent providers will turn out to be a simplified process to both user and broker. Users won't need to change credentials nor host names. They will continue to access the service through the same domain assigned when first deployed to the CSB. The CSB will ensure the synchronization of persistent data without succor to a noticeable downtime period.

IV. Conclusion and Future Work

With the growing number of cloud providers, the search for the best platform to deploy and manage applications is a critical factor and may not always lead to the best decision. This paper addresses the aforementioned inconvenients proposing a high-level comprehensive architecture intended to ease application lifecycle stages, such as, deployment, management, and monitoring of cloud applications over the two cloud models PaaS and IaaS.

The proposed CSB is a key part from the framework by delivering and evaluating, the most appropriate platform from a catalog of miscellaneous PaaS offerings, based on the application profile or defined cost thresholds. However, if the supported platforms don't cover all the users' needs, the developer shall be forwarded and assist in preparing and setting up a PaaS on demand settled on lowest cloud layer solutions. This approach opens a bond between PaaS and IaaS which is orchestrated via the CSB and implemented through the outlined PaaS Manager and IaaS Manager.

An example of market players who can take advantage of such multipurpose cloud framework are the communications service providers (CSP). With the new added-value service providers, operators are becoming just a data-pipe guarantying connectivity between both ends. CSPs are undoubtedly interested in taking a share of the growing cloud market, setting a major position. The exploitation of two-sided business revenues with third-parties developers toward the management and migration of applications to private or hybrid cloud products is a likely model.

As proof-of-concept, a first prototype of the CSB and PaaS Manager is under development as well as the user's web portal and command line with Git integration. The monitoring system for covering resources usages and costs notifications will be tested soon for running intensive tests and benchmarks through real applications.

Acknowledgment

The work hereby presented is a Portugal Telecom Inovação funded project conducted under the ATNoG research group, Instituto de Telecomunicações (Aveiro pole), Portugal.

References

[1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 2010, 10.1007/s13174-010-0007-6. [Online]. Available: http://dx.doi.org/10.1007/s13174-010-0007-6

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep., 2009.

[3] "The nist definition of cloud computing," http://www.csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2011, [Online; accessed 9 May 2012].

[4] D. Collison, "Distributed Design and Architecture of Cloud Foundry," http://www.slideshare.net/derekcollison/design-of-cloud-foundry, 2011, [Online; accessed 9 May 2012].

[5] F. Paraiso, N. Haderer, P. Merle, R. Rouvoy, and L. Seinturier, "A Federated Multi-Cloud PaaS Infrastructure," in *5th IEEE International Conference on Cloud Computing*. hawaii, États-Unis: IEEE Xplore Digital Library, Jun. 2012. [Online]. Available: http://hal.inria.fr/hal-00694700

[6] Z. ur Rehman, F. Hussain, and O. Hussain, "Towards multi-criteria cloud service selection," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, 30 2011-july 2 2011, pp. 44 –48.

[7] J. K. J. Kang and K. M. S. K. M. Sim, "Cloudle: A multi-criteria cloud service search engine," pp. 339–346, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5708589

[8] T. Liu, Y. Katsuno, K. Sun, Y. Li, T. Kushida, Y. Chen, and M. Itakura, "Multi cloud management for unified cloud services across cloud sites," in *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, sept. 2011, pp. 164 –169.

[9] M. Maiya, S. Dasari, R. Yadav, S. Shivaprasad, and D. Milojicic, "Quantifying manageability of cloud platforms," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, june 2012, pp. 993 –995.

[10] J. Shao and Q. Wang, "A performance guarantee approach for cloud applications based on monitoring," in *Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops*, ser. COMPSACW '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 25–30. [Online]. Available: http://dx.doi.org/10.1109/COMPSACW.2011.15

[11] X. Cheng, S. Yuliang, and L. Qingzhong, "A multi-tenant oriented performance monitoring, detecting and scheduling architecture based on SLA," in *Joint Conferences on Pervasive Computing*, ser. JCPC 2009. Tamsui, Taipei: IEEE Internet Computing, 2009, pp. 599–604.

[12] T. Wood, A. Gerber, K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, "The case for enterprise-ready virtual private clouds," in *Proceedings of the 2009 conference on Hot topics in cloud computing*, ser. HotCloud'09. Berkeley, CA, USA: USENIX Association, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855533.1855537

Uma Abordagem Estratificada à Monitorização de Serviços Cloud

Nuno Palhares, Solange Rito Lima
Departamento de Informática, Centro Algoritmi
Universidade do Minho
Campus de Gualtar
4710-057 Braga, Portugal
Email: nunopalhares89@gmail.com, solange@di.uminho.pt

***Resumo*—A monitorização de redes é uma tarefa essencial na gestão e engenharia das redes de comunicações atuais. Face a paradigmas como *Cloud Computing* e *Cloud Services*, os desafios colocados à monitorização de redes e serviços são ainda mais variados e exigentes. *Cloud Computing* inclui modelos de serviços distintos (IaaS, PaaS, SaaS), compartilhando algumas necessidades comuns na medição de infraestruturas, mas com especificidades de acordo com o tipo de serviço prestado e recursos envolvidos. Posto isto, é essencial ter uma visão geral dos distintos aspetos relacionados com a monitorização de serviços *Cloud*, para uma melhor compreensão dos pontos-chave e promover a qualidade dos serviços prestados. Neste contexto, este artigo apresenta uma abordagem estratificada à monitorização de Serviços *Cloud*. O objetivo principal prende-se com a identificação das várias dimensões da monitorização de serviços *Cloud*, combinando as perspetivas do fornecedor de infraestruturas e de serviços, e dos clientes. Consequentemente, a monitorização do estado dos recursos, da qualidade de serviço, qualidade de experiência e contratos de serviço são aspetos a cobrir. Este processo envolve a identificação de parâmetros e métricas relevantes para cada dimensão monitorizada.**

I. Introdução

Hoje em dia temos presenciado a um crescimento da oferta de serviços baseados em *Cloud Computing*. Este facto deve-se sobretudo à descida dos custos capitais e operacionais (CapEx e OpEx) associados à tecnologia, fruto do aumento da concorrência. Este fator, aliado às vantagens de *Cloud Computing* têm-se traduzido num aumento exponencial da tecnologia. Relativamente ao conceito, as *Clouds* são um grande conjunto de recursos virtualizados facilmente utilizáveis e acessíveis (como *hardware*, plataformas de desenvolvimento e/ou serviços). Esses recursos podem ser dinamicamente reconfigurados para se ajustarem a uma carga variável, permitindo também uma melhor utilização dos recursos. Este conjunto de recursos é tipicamente explorado por um modelo *pay-per-use* (também conhecido por *pay-as-you-go*), em que as garantias são oferecidas pelo fornecedor da infraestrutura, por meio de SLAs (*Service Level Agreements*) personalizados [1].

Uma das preocupações constantes das empresas está relacionada com a monitorização e gestão de redes e serviços. A gestão dos serviços *Cloud* é sustentada por ações como visualização, controlabilidade e automação em ambientes virtuais. Este tipo de ações assume um papel importante no apoio à gestão da complexidade associada a *Cloud Computing*. A visualização é uma vantagem para um gestor de serviços, na medida em que ajuda a responder rapidamente a eventos e a tomar melhores decisões, enquanto o controlo ajuda a gerir riscos e a automação a reduzir custos. Em *Cloud Computing*, na perspetiva do serviço *Cloud* para o fornecedor e cliente, a gestão destes serviços e a garantia de QoS (*Quality of Service*), tornou-se uma das dificuldades do desenvolvimento da tecnologia. Perante este cenário, torna-se essencial ter uma visão global de todos os aspetos que possam interferir na qualidade do serviço prestado e que por sua vez devem ser tidos em conta no processo de monitorização e gestão.

Uma das questões mais importantes passa pela monitorização dos diferentes modelos de implementação de serviços *Cloud*. De facto, cada modelo de implementação possuí as suas próprias características e necessidades, logo necessita de diferentes abordagens de monitorização. As diferenças existem principalmente entre as *Private Clouds* e as *Public Clouds* (ver conceitos na secção II). Nas *Private Clouds* uma empresa apenas tem de lidar com os seus próprios recursos. Devido às suas políticas de segurança, os dados relevantes estão sob o controle da organização. Por outro lado, as *Public Clouds* exigem um maior investimento na monitorização do tráfego, devido sobretudo à difusão geográfica e aos grandes conjuntos de recursos envolvidos, assim como à necessidade de mais flexibilidade, escalabilidade e segurança. As questões de segurança, tais como configurações de *firewalls*, podem afetar e limitar aspetos relacionados com a monitorização entre os fornecedores de serviços *Cloud*. As *Public Clouds* necessitam ainda de fornecer informações sobre a monitorização aos seus clientes, o que requer mais flexibilidade, segurança e customização.

As características das *Public Clouds* levantam ainda preocupações relacionadas com a distribuição geográfica dos recursos que constituem a base da infraestrutura *Cloud*. À primeira vista, pode parecer que em *Cloud Computing* já não existe a preocupação sobre a localização geográfica. Essa ideia advém da sua natureza permitir um amplo acesso à rede, característica herdada da Internet e na qual se baseia. Contudo, a computação em nuvem não pode sobrepor-se às leis da física e os atrasos consequentes numa transmissão de dados podem tornar-se um verdadeiro problema ao fornecimento de

um serviço de qualidade. Utilizadores em localizações remotas podem estar sujeitos a latências inaceitáveis, limitando o grau de interatividade e interferindo nos parâmetros relacionados com QoS e QoE (*Quality of Experience*).

De outra perspetiva, um tópico bastante pertinente e que deve ser tido em consideração na monitorização *Cloud* são cada vez mais as questões energéticas. As mudanças do clima e o aquecimento global são dois dos problemas mais relevantes para o nosso planeta, onde o aumento das temperaturas está diretamente relacionado com a quantidade de dióxido de carbono produzida. Posto isto, nos vários ramos da ciência e da tecnologia, nos últimos anos tem existido um investimento na investigação e desenvolvimento de soluções "amigas" do ambiente. Neste contexto surgiram termos como *Green IT* e *Green Cloud Computing*. O desafio em *Green Cloud Computing* passa por minimizar a utilização de recursos e continuar a satisfazer a qualidade dos serviços requisitados e a sua robustez, contribuindo não só para uma redução dos custos operacionais como também do impacto ambiental [6], [7]. Assim, as questões energéticas são sobretudo um aspeto a ser associado à monitorização das infraestruturas ao nível dos recursos físicos.

Outra das grandes preocupações inerentes à monitorização prende-se com as questões de segurança. Este é um tópico em que tem havido grande investigação por ser crucial para a implantação do paradigma pois relaciona-se com a confiança e adesão dos clientes. A segurança está associada a aspetos de elevada importância, tais como a integridade, disponibilidade, privacidade e autenticidade dos dados e dos utilizadores.

Numa perspetiva fornecedor/cliente, na prestação de serviços *Cloud* levantam-se questões económicas e contratuais, devendo os serviços ser prestados de acordo com os SLAs e requisitos de QoS e QoE estabelecidos. Um fornecedor de um serviço *Cloud* deve estar em constante contacto com o cliente e deve levar em consideração o seu *feedback* para melhorar a qualidade do serviço prestado. Assegurar a qualidade de serviço e outros requisitos pertinentes, como a segurança, são metas que se revelam um processo complexo e difícil, fruto do uso de ambientes virtuais e por vezes de infraestruturas de outras empresas. No caso da ocorrência de conflitos de interesse que possam surgir entre fornecedores e clientes, a adoção de uma *third-party* neutra, responsável pela monitorização do desempenho do serviço, parece ser a melhor solução [3]. Respeitar um SLA é o primeiro passo para uma boa interação entre fornecedores e clientes, já que o objetivo de um SLA é garantir que a QoS e QoE são compreendidas da mesma forma por ambos. Os SLAs funcionam como um dos instrumentos primários de controlo por parte do utilizador. Face a estes fatores, a necessidade de um sistema de monitorização eficiente e transparente revela-se um requisito de particular importância.

Neste contexto, e conciliando as várias vertentes da monitorização focadas, o presente artigo apresenta uma abordagem estratificada à monitorização dos serviços *Cloud*. Para cada camada do modelo proposto apresentam-se os principais parâmetros e métricas a considerar, reunindo assim num modelo integrado as diferentes necessidades de monitorização e as perspetivas das diferentes entidades participantes. Pretende-se, desta forma, reunir e clarificar os principais aspectos envolvidos na monitorização em *Cloud* e fomentar o desenvolvimento de plataformas de monitorização abrangentes e flexíveis.

Este artigo está organizado da seguinte maneira. Na secção II é fornecida uma contextualização das definições usadas em *Cloud Computing*. Na secção III é apresentado o trabalho relacionado na área da monitorização, com a identificação de ferramentas e plataformas/*frameworks* neste contexto. Seguidamente, na secção IV é apresentada a proposta estratificada à monitorização dos Serviços *Cloud*, identificando num modelo por camadas as várias dimensões envolvidas na monitorização destes serviços. Por fim, na secção V são apresentadas as principais conclusões do trabalho efetuado, assim como as propostas de trabalho futuro.

II. Conceitos

De seguida são apresentados alguns conceitos inerentes à tecnologia *Cloud Computing*, tais como os Modelos de Serviço e os Modelos de Implementação existentes.

A. Modelos de Serviço

Os serviços *Cloud Computing* estão divididos em modelos de serviço, de acordo com a sua natureza. Os três principais modelos que estruturam a arquitetura *Cloud* são os IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*) e SaaS (*Software as a Service*), descritos de seguida. Os serviços relativos às infraestruturas são considerados a camada inferior, seguidos pelos ambientes/plataformas de desenvolvimento. As aplicações são o *front-end* do utilizador e residem no topo da pilha *Cloud* [2], [3]. Na Tabela I estão ilustrados alguns exemplos de serviços classificados segundo os modelos discutidos, assim como a relação Vendedor/Comprador de cada modelo de serviço.

- **Infrastructure as a Service (IaaS)**: este modelo de serviço fornece recursos computacionais virtuais, nomeadamente poder de processamento, de armazenamento e de comunicação. O cliente não tem privilégios para controlar a infraestrutura *Cloud* subjacente, porem possui controlo sobre as aplicações desenvolvidas, sistemas operativos, armazenamento e alguns componentes de rede.
- **Platform as a Service (PaaS)**: este é um modelo que disponibiliza um conjunto de ferramentas necessárias ao desenvolvimento de aplicações *online*, sem que haja preocupações com a sua hospedagem. As ferramentas disponibilizadas fornecem um conjunto bem integrado e especializado de serviços que incorporam tudo o que um programador necessita nas áreas de desenvolvimento, teste, publicação, hospedagem e manutenção de aplicações.
- **Software as a Service (SaaS)**: o objetivo deste tipo de modelo passa por disponibilizar aos clientes, aplicações que correm sobre uma infraestrutura *Cloud*. O ambiente de execução é a Internet e as aplicações estão acessíveis através de vários dispositivos clientes, com recurso a

interfaces como um *web browser*. O utilizador não está necessariamente a pagar pela compra de um sistema, ou seja, está apenas a adquirir o direito de utilizar um serviço, o que na sua essência é um *software* como muitos outros existentes.

Tabela I
EXEMPLOS DE SERVIÇOS.

Modelos de Serviço	Exemplos	Vendedores	Compradores
IaaS	Amazon Web Services, Microsoft Hyper-VGoGrid, Proofpoint, Rackspace, RightScale, IBM (Blue Cloud), VMWare VCloud, Sun (Project Carloine), HP Adaptative IaaS EMC, Windows Azure...	Fornecedores de Datacenters	Empresas
PaaS	Google App. Engine, Windows Azure, dotCloud, Salesforce, Redhat, Oracle, Cloudera, Cloud Foundry...	Fornecedores de Plataformas de Serviço	Companhias de Desenvolvimento de Software
SaaS	Office 365, Salesforce, Google Apps., Yahoo (Zimbra), Concur, Taleo, Netsuite, Proofpoint, Dropbox, Workday, Hotmail...	Companhias de Software	Utilizadores Finais

B. Modelos de Implementação

Numa perspetiva organizacional, existem formas básicas nas quais os serviços *Cloud* podem ser implementados. Os três modelos mais populares são os seguintes [2], [3]:

- **Public Cloud**: disponibilizam recursos de computação ao público em geral ou a um grande grupo de indústrias, através da Internet. Os utilizadores deste modelo utilizam serviços que são disponibilizados por organizações especializadas na venda de serviços *Cloud*. Neste caso o termo "*public*" nem sempre significa que o serviço não tem custos, apenas caracteriza o modo de acesso à sua interface. Os clientes alugam o acesso dos recursos conforme necessitam, baseando-se num modelo de pagamento *pay-as-you-go*.
- **Private Cloud**: neste tipo de implementação os serviços e os recursos computacionais estão exclusivamente dedicados a uma organização particular e não são partilhados com outras organizações. A diferença para as *Public Clouds* reside no facto dos dados e processos serem geridos dentro de uma organização. Não existem restrições ao nível da largura de banda da rede, exposições a falhas de segurança ou requisitos legais inerentes à sua utilização. As questões de segurança não são uma questão fundamental como nas *Public Clouds*, uma vez que os recursos estão protegidos pelas políticas de segurança (*firewall*) da própria empresa.
- **Community Cloud**: a infraestrutura *Cloud* é partilhada por várias organizações e suporta uma comunidade que partilha alguns interesses (e.g. uma missão comum, requisitos específicos de segurança, políticas entre outras considerações). Os membros da comunidade partilham o acesso aos dados e aplicações da *Cloud* em questão. Estão localizadas tanto no local como fora do estabelecimento e a sua gestão pode ser levada a cabo por uma organização ou por uma *third party*.
- **Hybrid Cloud**: este é um modelo que resulta da combinação das características dos tipos de modelos anteriormente descritos, tal como o nome sugere. Resumidamente, uma *Private Cloud* possui as suas infraestruturas locais complementadas com o poder de computação de uma *Public Cloud*.

III. TRABALHO RELACIONADO

Uma das preocupações constantes dos fornecedores de serviços está relacionada com a monitorização e gestão dos serviços *Cloud*. Existe a necessidade de identificar, antecipar e reportar falhas, com vista a uma otimização dos serviços e respetiva satisfação dos clientes. Os administradores de sistemas e os próprios utilizadores finais podem monitorizar e gerir os seus recursos de várias formas, dependendo do tipo de serviço em questão. Na Tabela II estão indicadas algumas das ferramentas de monitorização disponíveis, estando divididas de acordo com a técnica e paradigma a que obedecem. A monitorização local, tal como o nome indica, é feita localmente nos respetivos ambientes. Na monitorização remota, as ferramentas de monitorização são distribuídas e escaláveis, suportando sistemas de computação de alto desempenho, como *clusters* ou *grids*. Uma outra forma de monitorização é através de plataformas de gestão web, cuja oferta no mercado é maior, devido sobretudo à competitividade entre empresas de desenvolvimento deste tipo de produtos.

Tabela II
FERRAMENTAS DE MONITORIZAÇÃO.

Tipo	Exemplos
Local	Sysstat (Isag, Ksars), Dstat.
Remota	Nagios, Ganglia, GroundWork, Cacti, MonALISA, GridICE.
Plataformas de Gestão Web	RightScale, Landscape, Amazon CloudWatch, Gomez, Hyperic/Cloud Status, 3Tera, Zenos, Logic Monitor, Nimsoft, Monitis, Kaavo, Tap in systems, CloudKick, Enstratus, YLastic, TechOut, ScienceLogic, Keynote, NewRelic.

Para além das ferramentas acima citadas, existem ainda algumas propostas de *frameworks* e sistemas de monitorização que procuram dar os primeiros passos. Dois exemplos de projetos neste âmbito são o *Lattice* [4] e o *PCMONS* [5].

Em relação ao *framework Lattice*, este foi desenhada sobretudo para monitorizar recursos e serviços em ambientes virtuais. Resumidamente, este *framework* foi desenvolvido e implementado em conjunto com o projeto RESERVOIR. O RESERVOIR é um serviço *Cloud* que distingue fornecedores de serviços dos fornecedores de infraestruturas e tem como objetivo aumentar a eficácia da computação, permitindo o desenvolvimento de serviços complexos. São abrangidas questões geográficas e de QoS, tentando também assegurar garantias de segurança. Por sua vez, o *Lattice* recorre a

um sistema de monitorização de *probes* para coletar dados para o sistema de gestão. Os autores tiveram o cuidado de implementarem um sistema que não fosse intrusivo, de modo a não afetar adversamente o desempenho do sistema ou de qualquer aplicação em execução. Para aumentar o poder e a flexibilidade da monitorização é introduzido o conceito de "fonte de dados" (*data source*). As fontes de dados podem conter de uma maneira dinâmica múltiplos *probes*. As fontes de dados delineadas são os recursos físicos, os recursos virtuais e as aplicações de serviço. Contudo, o *framework Lattice* é flexível e não se limita somente a este tipo de fonte de dados. O seu design permite que tanto as fontes de dados como os próprios tipos de *probes* sejam desenhados e planeados conforme as necessidades e objetivos.

Quando ao PCMONS, é um sistema que tem presente a ideia de que a monitorização pode beneficiar de ferramentas e conceitos já estabelecidos na gestão de computação distribuída. O seu objetivo principal passa por implementar um sistema de monitorização em *Private Clouds*, com recurso a *software open source*, nomeadamente o Nagios. Os autores argumentam que devido às características únicas de cada modelo de serviço, não é possível chegar a uma solução de gestão genérica. Face a este facto, para a solução proposta, optaram por um modelo IaaS, devido sobretudo à sua flexibilidade, e por *Private Clouds*, uma vez que estão sob o controlo das políticas de segurança da respetiva empresa. A arquitetura do sistema de monitorização é composta por três camadas e equipara-se a um modelo centralizado onde é utilizada a ligação cliente/servidor. A camada base corresponde às infraestruturas e basicamente contém as instalações (*hardware* e rede), assim como *software*. A camada do meio (*Integration Layer*) é responsável por abstrair os detalhes das infraestruturas e é composta por vários módulos, permitindo ao sistema ser adaptável e extensível (*plug-ins*) a outros cenários/ferramentas. A camada superior corresponde à visualização e fornece uma interface (no caso a do Nagios) onde através da análise das várias informações disponíveis, pode ser comprovado o cumprimento das políticas e dos SLAs estabelecidos.

IV. Monitorização Estratificada de Serviços Cloud

Esta secção apresenta a abordagem estratificada proposta para a monitorização de serviços *Cloud*, indicando as várias dimensões da monitorização destes serviços. Para cada dimensão são identificadas e propostas métricas relevantes para a monitorização a efetuar. Após uma análise da bibliografia e de referências relevantes na área, é possível constatar que ainda não existe um consenso na classificação de métricas que satisfaçam todos os requisitos impostos pelos ambientes *Cloud*. Portanto, uma apropriada classificação das métricas e a sua normalização são um grande desafio a cumprir, tendo em vista uma gestão eficiente e a otimização dos serviços *Cloud*.

Na secção IV-C são abordadas questões relacionadas com QoE. Por fim é estabelecida uma relação entre a abordagem proposta e os modelos de serviço existentes.

A. Abordagem Estratificada Proposta

Conforme mencionado, o modelo definido tem como objetivo abranger as várias dimensões envolvidas na monitorização de serviços *Cloud*. O modelo está estratificado em 4 camadas principais, que por sua vez se subdividem em algumas categorias. As 4 camadas principais correspondem à Infraestrutura, à Rede, ao Serviço/Aplicação e à relação Cliente/Fornecedor, conforme ilustrado na Figura 1. A camada referente às Infraestruturas abrange tanto os recursos físicos como os recursos virtuais envolvidos no complexo ambiente de *Cloud Computing*. Para além da necessidade de monitorizar os diversos componentes que constituem toda uma infraestrutura, existem ainda outras questões que devem ser monitorizadas a este nível, como as questões energéticas e de segurança. Na camada de Rede são abrangidos aspetos relacionados sobretudo com o serviço IP, como o débito e as questões de desempenho e disponibilidade/fiabilidade. Ao nível da camada de Serviço/Aplicação, a monitorização incide em questões que permitem avaliar a disponibilidade/fiabilidade, desempenho e segurança de um serviço, entre outros aspetos. Por fim a relação Cliente/Fornecedor de serviço deve ser alvo de uma monitorização ao nível da auditoria dos SLA, da contabilização do uso/custo e dos aspetos de segurança. De seguida são abordadas as 4 camadas em maior detalhe.

Figura 1. Modelo estratificado proposto para a monitorização de serviços Cloud.

1) Infraestrutura: Como base e suporte de toda uma arquitetura complexa que envolve o ambiente *Cloud Computing*, as infraestruturas físicas são um dos focos principais a ter em conta no processo de monitorização. Todos os componentes físicos, desde dispositivos de processamento, armazenamento, até aos de rede (*switches*, *routers*) devem ser monitorizados. A maioria das referências na área é unânime em considerar como métricas mais relevantes a percentagem de utilização de CPU, RAM, memória de armazenamento, assim como as estatísticas das interfaces de rede das máquinas físicas [4], [6], [7], [8], [9]. Como já referido, os dispositivos de rede devem ser igualmente monitorizados, pois problemas ao nível dos *switches*, *routers* ou mesmo dos *links* podem afetar a conetividade da topologia. Uma topologia instável pode acarretar problemas que influenciem todo um conjunto de aspetos, como, por exemplo, a engenharia de tráfego, o débito, a disponibilidade de um serviço, violações dos SLA, questões económicas, entre outros.

No que toca às questões energéticas, uma parte significativa da energia elétrica consumida pelos recursos de computação

é transformada em calor, o que por sua vez acarreta alguns problemas. As altas temperaturas reduzem o tempo de vida dos dispositivos/componentes e acabam também por influenciar a fiabilidade e disponibilidade do sistema. Por sua vez, os procedimentos de gestão de energia podem afetar o desempenho do sistema de uma maneira complexa, dado que a taxa de computação global é resultado da velocidade e da coordenação de múltiplos elementos dentro de um sistema [10]. Assim, tendo em atenção as questões energéticas na monitorização *Cloud*, sobretudo ao nível das infraestruturas, na Tabela III estão indicadas algumas das métricas a considerar, retiradas de referências na área que podem ser associadas a este tópico. Segundo [10], onde são levados em consideração aspetos ecológicos e de desempenho no sistema de gestão de recursos (métricas, técnicas, modelos, políticas, algoritmos), o consumo de energia é uma boa métrica para abordar as questões relacionadas com a energia. Também em [7], o consumo de energia é uma métrica levada em consideração. Por sua vez, em [6] são propostas também métricas ao nível do controlo das temperaturas e sistemas de *backup* de energia (geradores, UPS). Estas são métricas que surgem no contexto da solução proposta pelos autores para a gestão de recursos, baseada em modelos organizados e compostos por agentes autónomos. O objetivo passa sobretudo por otimizar a utilização de energia e reduzir a emissão de dióxido de carbono.

Relativamente à segurança das infraestruturas ao nível dos recursos físicos é tido como base o trabalho proposto em [11], [12], onde são recomendadas algumas restrições e auditorias à segurança *Cloud*. Os autores baseiam-se no trabalho efetuado pela *Cloud Security Alliance* (CSA), onde a *Cloud* é modelada em sete camadas, nomeadamente: *Facility, Network, Hardware, OS, Middleware, Application e User*. Visto que nesta etapa são abordados os recursos físicos *Cloud*, podem ser associadas as três primeiras camadas propostas pela CSA. Portanto, tendo em conta a análise feita às camadas *Facility, Network* e *Hardware* podem ser extraídas métricas de alguns dos procedimentos propostos, conforme os exemplos apresentados na Tabela III. No que toca às instalações (*Facility*) a segurança é sobretudo ao nível físico, onde podem ser implementados controlos de acesso através de videovigilância, vários sistemas de autenticação, sistemas de alarmes e sensores, entre outros. Os objetivos principais passam por evitar infiltrações maliciosas, manipulações de dados e assegurar a própria integridade das instalações e componentes. Ao nível do *Hardware* propriamente dito, as medidas de segurança estão em conformidade com as adotadas nas instalações, onde devem ser seguidos os protocolos de segurança. No que diz respeito à camada *Network*, devido à sua natureza, que pode ser descrita como a fronteira entre os dados dos clientes e os próprios clientes (inseridos por vezes em redes sujeitas a ameaças), podem ser adotadas *Firewalls*, *Intrusion Detection Systems* (IDS), *Intrusion Prevention Systems* (IPS), entre outros mecanismos.

Ainda a nível das infraestruturas, os recursos virtuais assumem um papel muito interventivo num ambiente de *Cloud Computing*, pelo que a sua monitorização se torna um

Tabela III
EXEMPLOS DE MÉTRICAS PARA A CAMADA DOS RECURSOS FÍSICOS.

Layer	Categoria	Exemplo de Métricas
Infraestrutura	Componentes	CPU (percentagem de utilização, nº de cores), RAM (percentagem de utilização), memória de armazenamento (percentagem de utilização, velocidade de leitura e escrita), estatísticas das interfaces de rede, conectividade da topologia.
Recursos Físicos	Energia	Consumos de energia, temperaturas, estado dos geradores e UPS.
	Segurança	Alarmes/sensores de incêndio, vigilância, controlo de acessos, sistemas de autenticação, monitorização de *firewalls*, IDS, IPS.

aspeto essencial. Os processos envolvidos na virtualização são constituídos por algumas operações importantes, como a suspensão/reinício/migração e início/paragem de *Virtual Machines* (VMs). Estas são operações abordadas em muitos dos recentes trabalhos de investigação, tendo em vista o desenvolvimento de métricas, como por exemplo a "utilização" [10]. Várias referências apontam os diversos componentes dos recursos virtuais como aspetos a monitorizar. As métricas mais comuns a este nível estão relacionadas sobretudo com a percentagem de utilização do CPU, RAM e memória de armazenamento das VMs (ver Tabela IV). As estatísticas das interfaces de rede das VMs são igualmente relevantes. Operações relacionadas com os processos de criação e migração de VMs ou número de instâncias ativas são também informações úteis [4], [6], [7], [8], [9].

Quanto à segurança nos recursos virtuais das infraestruturas, podem ser associadas as camadas *OS* e *Middleware* abordadas em [11], [12]. Neste caso as métricas a serem levadas em consideração devem ser extraídas da monitorização dos Sistemas Operativos ao nível dos eventos assim como do sistema de chamadas entre as VMs e o *hardware*. O objetivo passa sobretudo por evitar a cópia e modificações de dados. A camada *Middleware*, segundo os autores em [12], é considerada um potencial ponto fraco, pois encontra-se entre as camadas do OS e das Aplicações, envolvendo assim bastantes componentes, conforme o serviço e respetiva arquitetura em questão. Nesta camada, as métricas devem estar assim associadas à monitorização da virtualização e dos sistemas de segurança em arquiteturas *Cloud* heterogéneas.

2) Rede: Passando para a camada de "Rede", a classificação das categorias que a caracterizam e as respetivas métricas são sobretudo ao nível do Serviço IP. A subdivisão nesta camada é efetuada nas seguintes categorias: Débito, Desempenho e Disponibilidade/Fiabilidade (ver Tabela V). As métricas propostas para estas categorias estão associadas às métricas tradicionais das redes de computadores e telecomunicações, provenientes sobretudo dos esforços dedicados a este tópico pela International Telecommunications Union - Telecommunication Standardization Sector (ITU-T) e pelo grupo de trabalho IP Performance Metrics (IPPM) do IETF.

No que diz respeito ao Débito são várias as referências que

Tabela IV
EXEMPLOS DE MÉTRICAS PARA A CAMADA DOS RECURSOS VIRTUAIS.

Layer	Categoria	Exemplo de Métricas
Infraestrutura Recursos Virtuais	Componentes	CPU (percentagem de utilização, nº de cores), RAM (percentagem de utilização), memória de armazenamento (percentagem de utilização, velocidade de leitura e escrita), estatísticas das interfaces da VM, migrações de VM, número de instâncias ativas.
	Segurança	Monitorização de eventos e do OS ao nível do sistema de chamadas entre VMs e o *hardware*.

Tabela V
EXEMPLOS DE MÉTRICAS PARA A CAMADA DE REDE.

Layer	Categoria	Exemplo de Métricas
Rede	Débito	Volume de tráfego por unidade de tempo, largura de banda utilizada e disponível, capacidade.
	Desempenho	Duplicação de pacotes, perda de pacotes (OWPL, OWLP, IPLR), atraso (OWD, RTT, IPTD, IPDV), IPER, SPR.
	Disponibilidade/ Fiabilidade	*UP time*, (in)disponibilidade da rede, conetividade (*one* ou *two-way*), tempo de resposta (médio e máximo), tempo médio de restauro em caso de falhas, tempo médio entre falhas.

o classificam como essencial no processo de monitorização *Cloud* [13], [14]. Para além da importância ao nível das decisões relacionadas com a engenharia de rede, as questões económicas também estão "atentas" a este aspeto. Devido à sua natureza, o débito pode variar constantemente, o que leva a que as métricas que lhe estão relacionadas sejam monitorizadas de perto e tendo em vista o cumprimento dos SLA [15]. Na análise do volume de tráfego por unidade de tempo, uma monitorização ao nível de classes de serviço pode trazer benefícios, nomeadamente para a otimização da utilização da rede, identificação de classes com problemas, etc. A largura de banda quantifica o volume de dados que um *link* ou caminho pode transferir por unidade de tempo. A largura de banda disponível, representa assim uma métrica variável no tempo, onde é identificada a capacidade disponível, levando em consideração a carga atual. A capacidade, representando o limite máximo à largura de banda disponível, é também uma métrica que se enquadra neste contexto.

Em [4], [8] as estatísticas referentes ao tráfego de rede são apontadas como fontes de dados importantes à monitorização. Esta informação pode ser útil também na camada de rede ao nível do serviço IP, para além da camada das Infraestruturas físicas e virtuais, tal como referido anteriormente.

Quanto às métricas referentes ao Desempenho ao nível da rede, estas englobam as tradicionais métricas de QoS como a duplicação de pacotes, perda de pacotes (OWPL - *One-way packet loss*, OWLP - *One-way loss pattern*, IPLR - *IP packet loss ratio*), atraso (OWD - *one-way delay*, RTT - *round-trip time*, IPTD - *IP packet transfer delay*, IPDV - *IP packet delay variation*), *IP packet error ratio* (IPER), *Spurious IP packet ratio* (SPR), entre outras [9], [16].

No que toca à Disponibilidade/Fiabilidade de uma rede, esta pode apresentar períodos de inatividade provocados por problemas que podem ter origem nos componentes de rede, configurações de routing, entre outros aspetos. Face a esta possibilidade, torna-se relevante monitorizar a (in)disponibilidade de uma rede, assim como o estado da conectividade. O tempo de resposta a uma configuração de rede também pode ser um indicador relevante para avaliar a disponibilidade da rede. Perante a ocorrência de falhas na rede, o tempo médio entre a ocorrência de falhas ou o tempo médio de restauro são bons fatores de avaliação da fiabilidade de uma rede.

3) Serviço/Aplicação: Na camada de Serviço/Aplicação, a natureza dos parâmetros monitorizados e a maneira como estes devem ser recolhidos depende essencialmente do *software* a ser monitorizado e não da infraestrutura *Cloud* em que está inserido. Uma das principais preocupações a ter em conta, passa pela disponibilidade de um Serviço/Aplicação. Um Serviço/Aplicação *Cloud* está sujeito a um conjunto de aspetos de diversas naturezas que podem afetar a sua disponibilidade, devido sobretudo à complexidade do ambiente *Cloud*. Posto isto, devem ser associadas métricas à disponibilidade de um Serviço/Aplicação, onde são registados os períodos de tempo em que um serviço está em funcionamento e quando se encontra indisponível. Este é um tópico que envolve questões económicas, pois em caso de indisponibilidade de um Serviço/Aplicação existem violações de SLA e posteriores penalizações do lado do fornecedor, uma vez que a qualidade do serviço foi afetada. Na Tabela VI estão ainda indicadas algumas métricas associadas à Disponibilidade e Fiabilidade de um Serviço/Aplicação. Quando estamos perante um cenário de falhas de serviço (indisponibilidade de um serviço ou quebra significativa da qualidade de serviço), a capacidade de recuperação e o tempo utilizado deve ser do conhecimento dos clientes ou de *third-parties* responsáveis pela monitorização. Para além dos aspetos relacionados com a recuperação de um dado Serviço/Aplicação, os intervalos de tempo entre a ocorrência de falhas também funcionam como indicadores da sua fiabilidade e eficiência.

Por sua vez, o tempo de resposta de um dado serviço pode funcionar como um fator de medição do seu desempenho. Neste contexto, em [15] são abordadas métricas referentes ao tempo de resposta médio e máximo num cenário de jogos *online* em *Cloud Computing*.

Devido à natureza insegura do ambiente onde alguns dos Serviços/Aplicações são disponibilizados, a segurança torna-se um aspeto relevante a controlar. Tal como indicado na Tabela VI, o número de vulnerabilidades de segurança deve ser uma métrica importante, uma vez que é necessário monitorizar comportamentos para detetar possíveis violações. Outros aspetos ao nível da camada da Aplicação que podem ser monitorizados e salvaguardados são sobretudo os certificados digitais, chaves privadas, *Domain Name System Security Extensions* (DNSSEC), etc. O comportamento do utilizador também pode ser associado a esta camada e as métricas relevantes prendem-

se sobretudo com os processos de *login*, padrões de acesso, IPs associados, entre outras. A monitorização deve incidir ainda na gestão de passwords, onde são fornecidos dados como o formato das *passwords* e frequência com que devem ser renovadas [13].

Para além das métricas e aspetos da monitorização referidos anteriormente, podem ainda ser acrescentadas métricas associadas especificamente ao tipo de Serviço/Aplicação em questão. Por outro lado, pode ser útil o registo de um histórico, onde podem constar os IPs de acesso e registos dos tempos de login referentes aos diversos clientes.

Tabela VI
EXEMPLOS DE MÉTRICAS PARA A CAMADA SERVIÇO/APLICAÇÃO.

Layer	Categoria	Exemplo de Métricas
Serviço / Aplicação	Disponibilidade Fiabilidade	*UP time*, (in)disponibilidade do serviço, tempo de restauro em caso de falhas, tempo médio entre falhas.
	Desempenho	Tempo de resposta (médio/máximo), processamento *batch*.
	Segurança	Número de vulnerabilidades de segurança, padrões de acesso, processos de login, gestão de passwords.
	Outras	Registos dos tempos de login e IPs de acesso (histórico), métricas específicas do tipo de aplicação.

4) Cliente/Fornecedor: A relação Cliente/Fornecedor envolve todo um conjunto de interesses comerciais, o que torna necessário o estabelecimento de um contrato onde sejam especificados todos os aspetos do serviço em questão. Neste contexto é importante esclarecer o conceito de SLA. Um SLA é um contrato estabelecido entre fornecedor e cliente e especifica quais as necessidades dos consumidores e o compromisso dos fornecedores para com eles. Num SLA estão contidos normalmente itens como: conjunto de serviços fornecidos, uma definição completa e específica de cada serviço, requisitos de QoS, tempo de atividade, segurança, privacidade, procedimentos de *backup*, responsabilidades de ambas as partes, entre outros [3]. Uma referência ainda para as questões relacionadas com a localização geográfica dos *datacenters* em relação às leis nacionais e internacionais. Este é tido como um critério importante pelas companhias que pretendem investir em soluções baseadas na *Cloud*. Para tal é necessário que os SLA incluam e abranjam este tipo de parâmetros. O estabelecimento de normas para *Cloud Computing*, que ainda não se encontram claramente definidas, podem ajudar a lidar com estes parâmetros geográficos e legais [17]. No que toca à gestão de serviços, o cliente deverá requerer sumários de todo um conjunto de auditorias, feitas pelo fornecedor do serviço, como parte da verificação do respetivo SLA. Os SLA funcionam assim como um dos instrumentos primários de controlo do utilizador. Portanto, uma das métricas vitais a este nível de monitorização passa pela auditoria dos SLAs, onde são registadas todas as violações e incumprimentos dos mesmos. Posto isto, a verificação do cumprimento dos SLAs está diretamente relacionada com as camadas anteriores, uma vez que pode ser necessário recorrer a métricas estabelecidas ao nível das infraestruturas, rede ou serviços. Por exemplo, em [7] é referida uma métrica relativa à média de violações de SLA, que representa a média de desempenho de CPU que não foi alocada a uma aplicação quando requerida. Em caso de ocorrência deste tipo de incumprimentos, existem consequências. Dependendo dos parâmetros dos SLAs estabelecidos entre clientes e fornecedores, podem ocorrer penalizações e compensações por parte dos fornecedores de serviço.

A monitorização da contabilização do uso do serviço também é um aspeto bastante importante, na medida em que existe a necessidade de assegurar os interesses económicos de ambas as partes. Devido à natureza elástica dos ambientes *Cloud*, aliado ao modelo comercial *"pay-as-you-go"*, a medição da utilização e o custo tornam-se aspetos vitais [9], [15]. A análise da contabilização dos serviços e respetiva receita, permite também aos fornecedores de serviço adaptarem os seus planos de preçários e estratégias comerciais conforme as necessidades do mercado. Este estudo pode fazer a diferença, numa altura onde a forte concorrência na área se notabiliza, fruto do aumento da oferta na web de ferramentas baseadas em *Cloud Computing*.

Relativamente à segurança, em [2] são abordados alguns parâmetros que devem estar incluídos num SLA, e que por sua vez se enquadram na relação entre cliente e fornecedor. Este tipo de parâmetros diz respeito sobretudo ao estado de aquisição e atualização dos padrões de segurança relevantes por parte do fornecedor, assim como dos certificados. Outros tipos de parâmetros indicados são o estado de certificação da parte responsável pela gestão; estado das restrições operacionais incluídas nas medidas de segurança impostas pelo sistema de gestão; estado da garantia de confidencialidade nas trocas de dados entre *Clouds*; localização dos dados; estado da aquisição de *logs* para a deteção de atos maliciosos e o período durante o qual estes são mantidos; estado do controlo da comunicação para bloquear comunicações maliciosas; estado das medidas que atuam contra o congestionamento da rede, evitando ataques *Denial of Service* (DoS)/*Distributed Denial of Service* (DDoS); estado das medidas contra *malware*.

Tabela VII
EXEMPLOS DE MÉTRICAS PARA A CAMADA CLIENTE/FORNECEDOR.

Layer	Categoria	Exemplo de Métricas
Cliente/ Fornecedor	Auditoria	Monitorização de violações e incumprimentos dos SLA, penalizações.
	Contabilização	Monitorização do uso e respetivo custo do serviço, receita.
	Segurança	Estado de aquisição e atualização dos padrões de segurança, certificados, localização dos dados.

B. Questões Relacionadas com QoE

Com a migração de aplicações pessoais e comerciais para a *Cloud*, a qualidade do serviço prestado torna-se um importante diferenciador entre os diversos fornecedores. Um fator que está diretamente relacionado com a QoS é a qualidade que é presenciada pelo utilizador final, ou seja, a qualidade de experiência (QoE) resultante da utilização de um dado serviço. Posto isto, torna-se também imprescindível monitorizar a QoE.

Neste tipo de monitorização são tidas em conta métricas como atraso, variações do atraso (*jitter*), perdas, latência, entre outras. Contudo, estes são aspetos que não fazem parte do conhecimento e do vocabulário comum dos utilizadores finais. Porém a sua opinião e *feedback* acerca da satisfação em relação aos serviços subscritos são um fator bastante relevante a ter em conta na avaliação de toda a infraestrutura. Devido aos diversos intervenientes de um ambiente *Cloud*, perceber e gerir a QoE dos serviços requer uma visão multidisciplinar, que integra a tecnologia, utilizador e aspetos comerciais da qualidade do acesso do utilizador final. O objetivo principal da gestão da QoE está assim relacionado com a intenção de fornecer uma aplicação *Cloud* de alta qualidade ao utilizador final e tentar minimizar os custos dos diversos intervenientes. Estes vão desde entidades relacionadas com os modelos de serviço da pilha *Cloud* (IaaS, PaaS e SaaS), até aos fornecedores das redes subjacentes (Telcos - *Telecommunications companies* e ISPs - *Internet Network Providers*).

No que toca aos atuais trabalhos de investigação da QoE na *Cloud*, estes focam-se sobretudo em aplicações multimédia, onde se encaixam serviços de *streaming* HTTP como o Youtube ou Netflix. O impacto dos tempos de espera na perceção do utilizador tem ganho especial atenção nas comunidades de investigação, dado o aumento de popularidade dos serviços multimédia *Cloud* [18]. Este paradigma dos tempos de espera pode ser também associado a aplicações interativas como *web browsing*. Quanto aos serviços *Cloud* mais complexos, como produtos *office*, edição colaborativa ou OS a correr na *Cloud*, os trabalhos de pesquisa relacionados com a QoE ainda estão a dar os primeiros passos. Existem ainda algumas questões em aberto como, por exemplo, o impacto da interatividade dos utilizadores e a sua influência na QoE ou o relacionamento da QoE com as expectativas dos utilizadores, resultantes do domínio do uso do serviço em questão, entre outras.

Relativamente à gestão da QoE em geral, em [18] são abordados os passos básicos a ter em consideração. Estes estão relacionados com o entendimento e mapeamento, monitorização e estimação, adaptação e controlo da QoE. Num primeiro passo, é necessário entender quais são os requisitos de uma aplicação e efetuar um mapeamento entre parâmetros mensuráveis e QoE. Um mecanismo típico de avaliação de QoE passa pelo cálculo de *Mean Opinion Scores* (MOS). O próximo passo consiste na monitorização (desde infraestruturas, condições da rede, SLAs e informações específicas das aplicações) e estimativa de QoE. A monitorização pode ser efetuada pelo fornecedor dentro da rede, onde são requeridas funções de mapeamento entre a QoS e QoE, ou ao nível de parâmetros específicos de uma aplicação, o que requer técnicas de *Deep Packet Inspection* (DPI). Como alternativa, existe ainda a opção da monitorização no utilizador final, dando a melhor perspetiva sobre a qualidade presenciada. Por fim, a adaptação e o controlo da QoE tem como objetivo possibilitar aos fornecedores que atuem antes que o utilizador possa notar algum problema e ficar insatisfeito ou abandonar o serviço.

Os mesmos autores identificam ainda alguns desafios que surgiram com a migração de serviços para a *Cloud* e que têm influência na qualidade presenciada pelos utilizadores finais. Os desafios identificados podem ir desde a distribuição geográfica do utilizador, artefactos introduzidos com o aumento das distâncias da rede entre o utilizador e o serviço, problemas de gestão de recursos derivados das localizações geográficas, ou até a questão do envolvimento de diversas entidades no fornecimento de um serviço. No caso da localização geográfica, esta pode limitar o grau de interatividade, uma vez que utilizadores em localizações remotas podem estar sujeitos a latências inaceitáveis, levando em consideração a distância entre o *data center* e o local onde o serviço é acedido. A grande quantidade de utilizadores em várias localizações geográficas também podem ter influência direta num serviço, pois podem ser afetados requisitos como a escalabilidade e a velocidade de acesso. Uma referência ainda para a dependência da QoE para com as condições da rede e os SLAs, na medida em que é definido o caminho entre o *datacenter* e o utilizador final, atravessando diferentes domínios administrativos.

C. Relação com os Modelos de Serviço

Devido às diferenças significativas entre os três modelos de serviço mais populares, é consensual que não exista uma solução genérica de monitorização *Cloud*. Cada modelo de serviço possui diferentes áreas e graus de controlo, assim como as suas próprias características de gestão. Posto isto é compreensível que seja difícil alcançar uma solução de gestão *Cloud* genérica. Face a este paradigma, um sistema de monitorização necessita de ser planeado e desenvolvido, com o intuito de ser adequado aos objetivos da gestão. Este processo para além de cobrir os vários constituintes de todo um ambiente *Cloud*, deve ainda levar em consideração aspetos como a QoS/QoE, os SLAs e características como a segurança, robustez, escalabilidade, elasticidade, entre outras [4].

Neste contexto torna-se útil relacionar o modelo estratificado proposto para a monitorização de serviços *Cloud* com os modelos de serviço. Essa relação está ilustrada na Tabela VIII, levando em consideração os três modelos de serviço mais populares (IaaS, PaaS e SaaS) e as camadas do modelo proposto (Infraestrutura, Rede, Serviço/Aplicação e Cliente/Fornecedor).

No que diz respeito à camada de monitorização das infraestruturas, esta pode ser essencialmente associada ao modelo de serviço IaaS. Na base desta associação estão as características dos componentes envolvidos, uma vez que são comuns. Na camada das infraestruturas estão incluídos componentes dos recursos físicos e virtuais relativos ao processamento, armazenamento e comunicação em rede, ou seja, aspetos que também caracterizam o modelo de serviço IaaS.

Quanto à camada de Rede, se for levada em consideração uma monitorização fim-a-fim, esta relaciona-se com os três modelos de serviço direta ou indiretamente. Tendo em consideração todo o ambiente *Cloud*, dependendo dos intervenientes e do serviço em questão, a monitorização da Rede pode ser efetuada de várias maneiras. No caso de um serviço fornecido a um cliente com base num modelo SaaS, as questões relacionadas com os aspectos de Rede (como Débito

e Desempenho), devem ser monitorizadas desde a origem (infraestruturas, *datacenters*) até ao local onde o utilizador acede ao serviço. Neste caso estão envolvidos os modelos de serviço IaaS e SaaS. No caso de existirem intermediários, como as companhias de desenvolvimento de *software*, ou seja, fornecedores de plataformas de serviço inseridos num modelo de serviço PaaS, a monitorização da camada de Rede também está relacionada com este modelo. Portanto esta camada pode estar relacionada com os diversos intervenientes, uma vez que pode atravessar os vários modelos de serviço, tendo em consideração uma monitorização fim-a-fim.

Por sua vez, a camada Serviço/Aplicação está associada ao modelo de serviço SaaS. Na base desta relação está a natureza da camada, uma vez que os parâmetros monitorizados e a maneira como devem ser recolhidos dependem sobretudo do *software* e não da infraestrutura.

Por fim, a camada Cliente/Fornecedor ao abordar aspetos como contabilização e auditorias, está diretamente relacionada com os três modelos de serviço. Devido à complexidade de um ambiente *Cloud* e à existência de diversos intervenientes, é normal existirem relações comerciais entre fornecedores e clientes a vários níveis. Os fornecedores tanto podem disponibilizar *datacenters*, como plataformas de serviço ou *software*, enquanto os clientes podem ser empresas de desenvolvimento de plataformas e *software* ou os utilizadores finais.

Tabela VIII
RELAÇÃO COM OS MODELOS DE SERVIÇO.

	IaaS	PaaS	SaaS
Infraestrutura	✓		
Rede	✓	✓	✓
Serviço/Aplicação			✓
Cliente/Fornecedor	✓	✓	✓

V. CONCLUSÃO E TRABALHO FUTURO

O rápido crescimento de *Cloud Computing* como um novo modelo de prestação de serviços é um facto que não pode ser negado. A noção da necessidade da existência de um sistema de monitorização para operar com eficiência um ambiente *Cloud* já está presente. Devido à falta de maturidade típica das novas tecnologias, podem ser apontadas algumas limitações, nomeadamente no controlo e gestão de uma *Cloud*. A monitorização *Cloud* pode beneficiar de metodologias, conceitos e ferramentas já consolidados na gestão da computação distribuída tradicional. Contudo, a natureza complexa de um ambiente *Cloud* torna difícil chegar a uma solução de gestão genérica, nomeadamente devido à natureza e às características próprias de cada modelo de serviço (IaaS, PaaS e SaaS) e de cada modelo de implementação (*Public* e *Private*).

Uma observação relevante que se constata no contexto da monitorização *Cloud* é a falta de normas. Este é um facto que assume particular importância quando se tenta realizar uma monitorização através de múltiplas *Clouds*, envolvendo questões geográficas e legais para além da QoS e QoE. Como parte dos esforços dedicados à normalização, o presente trabalho contribuí com algumas sugestões de parâmetros, métricas e boas práticas para uma monitorização eficiente dos serviços e ambientes *Cloud Computing*.

Como trabalho futuro, pretende-se aplicar os conhecimentos adquiridos num cenário prático, comprovando a utilidade do modelo proposto, nomeadamente as vantagens associadas a efetuar a monitorização de uma forma estratificada.

Agradecimentos: Este trabalho é financiado pelo FEDER através do Programa Operacional Factores de Competitividade-COMPETE e pela FCT - projeto FCOMP-01-FEDER-0124 022674.

REFERÊNCIAS

[1] Luis Vaquero, Luis Merino, Juan Caceres and Maik Lindner. *A Break in the Clouds: Towards a Cloud Definition*. SIGCOMM CCR, 39(1):50–55, January 2009.

[2] ITU-T FGCloud. *Part 1: Introduction to the Cloud Ecosystem: Definitions, Taxonomies, Use Cases and High-level Requirements*. Technical report, February 2012

[3] Cloud Computing Use Case Discussion Group. *Cloud Computing Use Cases white paper v4.0*. Technical report, July 2010

[4] Stuart Clayman, Alex Galis, Clovis Chapman, Giovanni Toffetti, Luis Merino, Luis Vaquero, Kenneth Nagin and Benny Rochwerger. *Monitoring Service Clouds in the Future Internet*. IOS Press, pages 115–126, April 2010.

[5] Shirlei Chaves, Rafael Uriarte, and Carlos Westphall. *Toward an Architecture for Monitoring Private Clouds*. IEEE Communications Magazine, pages 130–137, December 2011.

[6] Jorge Werner, Guilherme Geronimo, Carlos Westphall, Fernando Koch and Rafael Freitas. *Simulator Improvements to Validate the Green Cloud Computing Approach*. In Network Operations and Management Symposium (LANOMS) 7th Latin American, October 2011.

[7] Anton Beloglazov, Jemal Abawajy and Rajkumar Buyya. *Energy Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing*. ELSEVIER, Future Generation Computer Systems, 28, pages 755–768, May 2012.

[8] Ya-Shiang Peng and Yen-Cheng Chen. *SNMP-Based Monitoring of Heterogeneous Virtual Infrastructure in Clouds*. In Network Operations and Management Symposium (APNOMS) 13th Asia-Pacific, September 2011.

[9] Taesang Choi, Nodir Kodirov, Tae-Ho Lee, Doyeon Kim and Jaegi Lee. *Autonomic Management Framework for Cloud-based Virtual Networks*. In Network Operations and Management Symposium (APNOMS) 13th Asia-Pacific, September 2011.

[10] Mehdi Sheikhalishahi and Lucio Grandinetti. *Revising Resource Management and Scheduling Systems*. In CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science, page 121 – 126, 2012.

[11] Jonathan Spring. *Monitoring Cloud Computing by Layer, Part 1*. IEEE Security & Privacy Magazine, pages 66–68, March/April 2011.

[12] Jonathan Spring. *Monitoring Cloud Computing by Layer, Part 2*. IEEE Security & Privacy Magazine, pages 52–55, May/June 2011.

[13] Shirlei Chaves, Carlos Westphall and Flavio Lamin. *SLA Perspective in Security Management for Cloud Computing*. In Sixth International Conference on Networking and Services, IEEE Computer Society, pages 212–217, March 2010.

[14] Flávio Sousa, Leonardo Moreira, Gustavo Santos and Javam Machado. *Quality of Service for Database in the Cloud*. In CLOSER 2012 - 2nd International Conference on Cloud Computing and Services Science, page 595 – 601, 2012.

[15] Pankesh Patel, Ajith Ranabahu, and Amit Sheth. *Service Level Agreement in Cloud Computing*. Technical report, September 2009.

[16] Solange Lima. A Distributed Admission Control Model for Class-based Networks. PhD thesis, University of Minho, Braga, 2005.

[17] Katerina Stamou, Jean-Henry Morin, Benjamin Gateau and Jocelyn Aubert. *Service Level Agreements as a Service - Towards Security Risks Aware SLA Management*. In CLOSER 2012 -2nd International Conference on Cloud Computing and Services Science, page 663 – 669, 2012.

[18] Tobias Hoßfeld, Raimund Schatz, Martin Varela and Christian Timmerer. *Challenges of QoE Management for Cloud Applications*. IEEE Communications Magazine, pages 28 – 36, April 2012.

Network Virtualization - A Virtual Router Performance Evaluation

Bruno Parreira*†, Márcio Melo*†, João Soares*†, Jorge Carapinha*
*Portugal Telecom Inovação
Aveiro, Portugal
{bruno-m-parreira,joao-m-soares,marcio-d-melo,jorgec}@ptinovacao.pt

Romeu Monteiro†, Susana Sargento†
†Instituto de Telecomunicações
University of Aveiro
Aveiro, Portugal
{romeumonteiro7,susana}@ua.pt

***Abstract*—Network Virtualization is claimed to be a key component of the Future Internet by enabling the coexistence of heterogeneous (virtual) networks on the same physical infrastructure [1], providing the dynamic creation and support of different networks with different paradigms and mechanisms. In order for virtualization to be used in a network operator's infrastructure, its impact on the network traffic must be studied.**

In this paper, we perform an analysis of the impact of network virtualization on two types of traffic, TCP and UDP. To deploy the virtual networks, the Network Virtualization System Suite is used. This platform enables the creation of virtual networks on top of a substrate network, isolating the traffic in the different layers. The tests performed evaluate the effect that the increase of virtual routers and data flows has on throughput and packet delay. The effect of CPU load on throughput is also analyzed.

The results obtained using TCP demonstrate that the CPU load has a more adverse effect on throughput than increasing the number of virtual routers, with a loss of 25% in the first case and 15% in latter case. The UDP tests revealed that increasing virtual routers leads to an increase in packet delay variation.

***Index Terms*—Network Virtualization, Virtual Router, Virtual Network, Network Performance.**

I. Introduction

In the last few years the Internet has been walking steadily towards the Networks of the Future. These necessary changes still face a lot of resistance from legacy networks, which are based on technologies designed decades ago. Current networks lack the dynamism and the flexibility necessary for these changes to take place. Cloud Computing can be seen as an example of a paradigm being hindered by current network infrastructures. For example, a company moving its Information Technology (IT) resources to the cloud will probably use Virtual Private Networks (VPNs) based solutions to connect virtual infrastructures with their premises. VPNs were not designed to adapt to the users demand, a characteristic that is very popular with Cloud Computing [2].

Network virtualization can play an important role in the development of the Networks of the Future. It brings great improvements in terms of flexibility, isolation and dynamism, which will foster the development of new architectures and technologies while improving current network based services [3].

Currently there are various alternatives to deploy virtual networks, with one of them being the Network Virtualization System Suite (NVSS) [4] developed under the 4WARD project [5]. The NVSS is a platform for the creation, discovery, monitoring and management of virtual networks; it will be the one used in the tests performed in this paper. Although some tests have proven the functional capabilities of the platform, see [6] and [7], a data quantitative analysis is still missing. Several web based services like video streaming or voice calls have minimum requirements, in terms of throughput or jitter, that need to be met for these services to be deployed. This paper will cover performance parameters like throughput and packet delay that have a direct impact on the provisioning of services.

To perform these tests two types of traffic will be generated, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The results obtained using TCP demonstrate that the Central Processing Unit (CPU) load has a more adverse effect on throughput than increasing the number of virtual routers, with a loss of 25% in the first case and 15% in latter case. The UDP tests revealed that increasing virtual routers leads to an increase in packet delay variation.

The rest of the paper is organized as in the following. After summarizing the related work in section II, section III describes the architecture of the NVSS platform and presents the existing functionalities. Section IV describes the testbed used and evaluates the influence on the packet delay variance with the number of virtual routers and network flows. Section V analyses the virtual router throughput and investigates the influence of the CPU load on the overall throughput, and section VI concludes the paper and describes the future work.

II. Related Work

Future Internet research projects such as the PlanetLab [8] or the 4WARD [5] have investigated and promoted the use of network virtualization as a way to evaluate and deploy, in the latter, future Internet architectures.

With that in mind, Egi et. al [9] evaluated the use of Xen hypervisor [10] as a way to implement virtual routers. The performance of virtual routers on commodity hardware has been assessed in [11] and [12], where the virtual routers throughput is similar to the one of underlying hardware when using control plane from forwarding plane separation.

A platform for high performance and flexible virtual routers on commodity hardware based on multiple input queues has been proposed in [13]. The virtual router migration feature

has proposed by Wang et. al [14] as a primitive of network management operations. To reduce the Virtual Router (VR) downtime due to the migration process, Wang et. al [15] proposed the separation of the forwarding plane from the control plane.

Despite the existing performance research results provided in [11] and [12] on virtual routers running on commodity hardware, we argue that a deeper analysis on either the traffic types or the influence on the CPU load is still not tackled. We also argue that the impact on the packet delay variance with the number of virtual routers and without using control and forwarding plane separation is not assessed.

In the following section we present and describe the NVSS architecture and its built-in functionalities to handle virtual networks.

III. Network Virtualization Architecture

The goal of the developed virtualization platform is to provide the operators with a network virtualization solution that is easy to use, versatile, and efficient in virtual network discovery. The resulting platform provides the necessary functionalities to discover, monitor, deploy and manage virtual networks running on top of a substrate network. It is designed to run on Fedora Core 8 and Debian Lenny Linux distributions with the Xen kernel. Figure 1 presents the network virtualization considered approach, which takes into account a heterogenous physical network and builds on top of it virtual networks with different types of topologies.

A. Network Virtualization System Suite Architecture

The NVSS is composed of 3 software modules: the Agent module, the Manager module and the Control Centre module; their hierarchical decomposition is demonstrated in Fig. 2. The Agent module is designed to work within the domain of a Xen virtualization environment, running on every substrate node, in order to perform network enforcements and periodically

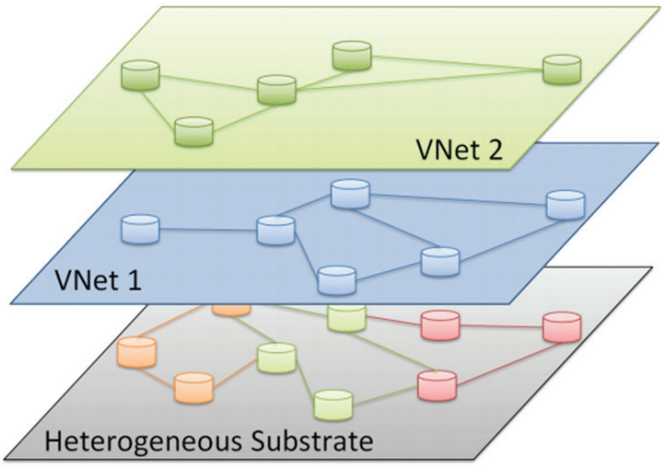


Figure 1. A Network Virtualization Approach

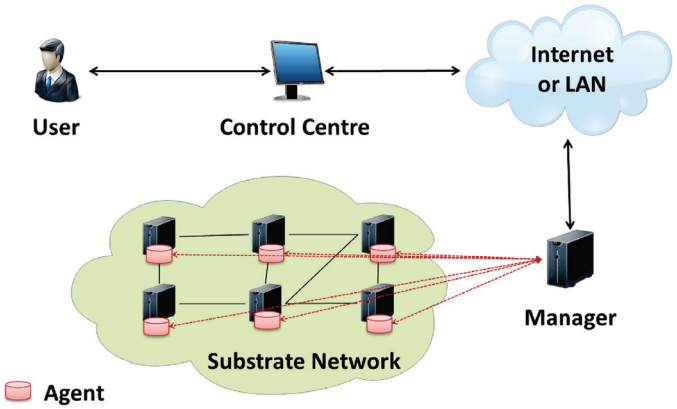


Figure 2. Network Virtualization System Suite - Architecture

gather data. The Agents, besides interacting with each other to share network topology information, also receive and send requests to the Manager, which is a centralized entity in charge of aggregating all Agents' knowledge and sending them commands. The Manager is also devoted to map new Virtual Network (VN) requests and to communicate with the Control Centre, which is the user's front-end, and provides him with graphical and simple to use virtual network creation, management, and monitoring functionalities.

B. Virtual Network Mapping and Creation

The Control Center module provides the user with means to create and embed a new VN. By selecting and placing resources on the platform Graphical User Interface (GUI) and by connecting them with links, a VN can be specified. The user may specify the resources' CPU capabilities, Random Access Memory (RAM) amount, location, number of interfaces and also perform network addressing configurations. The final step in creating a new virtual network is to commit it to the Manager, which will then map it in the physical infrastructure. The embedding problem, that includes both nodes and links mapping, is a complex one and requires a trade-off between computation time and embedding optimization. In order to lower the computational requirements, a heuristic mapping algorithm was developed, which aims to embed VNs taking into consideration both the substrate links' and nodes' loads.

C. Substrate and Virtual Network Monitoring

Dynamic resource monitoring is fundamental to provide an accurate view of the virtual and physical networks, and to quickly react to failures or configuration problems. The implemented monitoring functions periodically update the resources' information; therefore it is possible to quickly identify diverse situations, such as failures and high resource usage. Every Agent periodically checks its local resources' configuration and status, and reports back to the Manager if any change occurs. Several parameters are monitored: CPU load, RAM, Hard Disk Drive (HDD) usage, interface and link

status, interface bridge attachment and configuration, number of running virtual machines and their state.

D. Virtual Network Management

Just like the previously described monitoring ability, the management feature is also a crucial one; to that end, some functionalities are provided. It is possible to change the resource's state, i.e.: reboot, shutdown, suspend or power up; to change the assigned RAM memory in runtime; and to delete either a single resource or a complete VN, which greatly simplifies the administrator work.

IV. Evaluation - Packet Delay

In this section we start with the description of the testbed considered to evaluate several packet delay statistics as functions of the number of virtual routers and the number of network flows. The statistics considered are average packet delay, packet delay variance and packet delay variation.

A. Testbed Configuration

In order to analyze the impact of network virtualization on the packets' delay times, several aspects must be taken in consideration. For example, it must be guaranteed separation between the several flows of traffic, regardless of whether they traverse the same virtual router or not. Another important consideration is to guarantee that all traffic traversing the virtual routers is captured and limited to the one injected into the network. With this in mind, a testbed was designed, which is presented in this section.

The testbed used is composed of six computers using the configuration shown in figure 3. Three computers are used to generate the traffic flows, two computers are used to receive them, and one computer is used to deploy the virtual routers. The 5 computers used to generate and receive the traffic flows provided us with 14 ethernet interfaces, thus allowing for a maximum of 7 flows at a time, since we want to observe independent flows which don't start nor end at the same interfaces. Since there could be no more than 7 flows, the number of VRs was also limited to a maximum of 7. The computer, Eddie, where the virtual routers are instantiated, is an Intel Xeon E3220 with four cores (2.4GHz each) and 6GB of system memory. This computer runs Xen Hypervisor version 3.1. The machines responsible for traffic generation are connected to a switch which sends all incoming traffic to one port. This port is connected to a hub ensuring that the same packets which enter the virtual routers are captured by another machine using the *Wireshark* software [16]. On the right side of the virtual routers, a symmetric configuration is used.

To ensure independency between the different virtual links, Virtual Local Area Network (VLAN) tagging was used; VLAN tagging is only applied between the switches. In the physical machine Eddie, where the VRs reside, each virtual interface is associated with a specific bridge and VLAN tag.

Iperf [17] sessions are used to generate traffic flows with a packet size of 1300 bytes at a bit rate of 1Mbps. The data

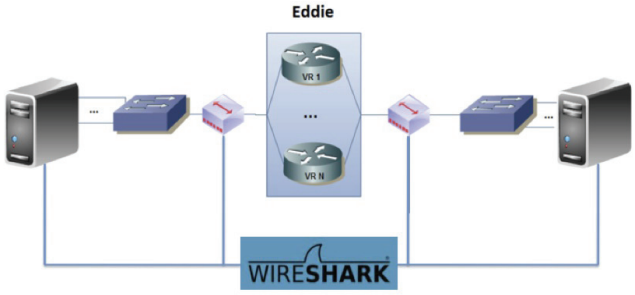


Figure 3. Experimental Apparatus

obtained in each test derive from 10 runs of 30s of traffic. To have an *end2end* view of the packet transmission, the *Wireshark* software was used in four different points along the path:

1) In the machine generating the flow;
2) In the hub before Eddie;
3) In the hub after Eddie;
4) And in the machine receiving the flow.

The *Wireshark* application uses the machine's system time to get the capture time of each packet.

The results obtained for packet delay are based on the time it takes for packets to go from hub1 to hub2, so as to evaluate the delay introduced by the virtual routers in Eddie. The timestamps of the packets were collected using the *Wireshark* software, and the data inside these packets was used to match the timestamps at the 2 hubs. This way, it was possible to obtain the individual delays for each packet in each flow.

To determine the packet delay variation, it was used the method defined in [18]. This method defines packet delay variation as the difference in delay times between two consecutive packets.

B. Results

In figures 4 and 5 we can observe the average packet delay for a set of 15 runs with confidence intervals of 90%. Figure 4 shows the effect of changing the number of active VRs while keeping constant the number of flows per VR; figure 5 shows the effect of increasing the flows in a single VR.

It is visible that the average packet delay is very stable regardless of the number of active VRs and flows per VR. The very small confidence intervals corroborate this stability among runs. This is likely due to the low amount of extra load imposed on the system by the extra traffic and extra VRs, which might lead to an average delay change that is small compared to the total average delay.

In figures 6 and 7 we can observe the variance of the packet delay for a set of 15 runs with confidence intervals of 90%. Figure 6 shows the effect of changing the number of active VRs while keeping constant the number of flows per VR, while figure 7 shows the effect of increasing the flows in a single VR.

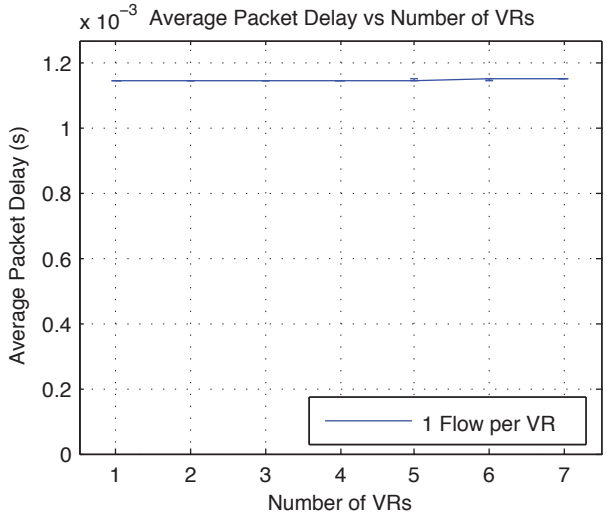


Figure 4. Average packet delay for a single flow per active VR and different numbers of VRs

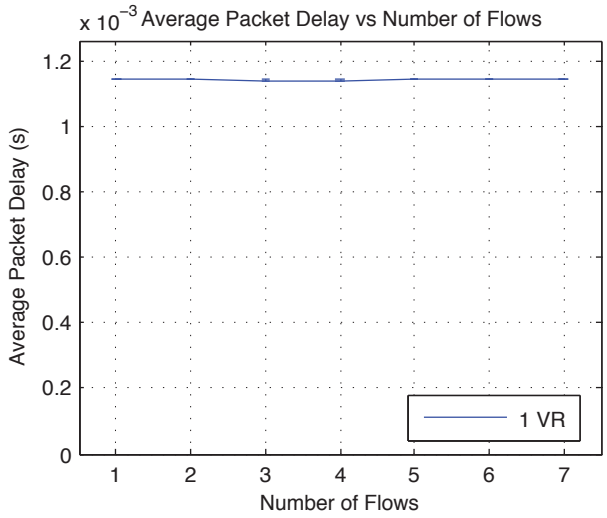


Figure 5. Average packet delay for a single active VR with different numbers of flows

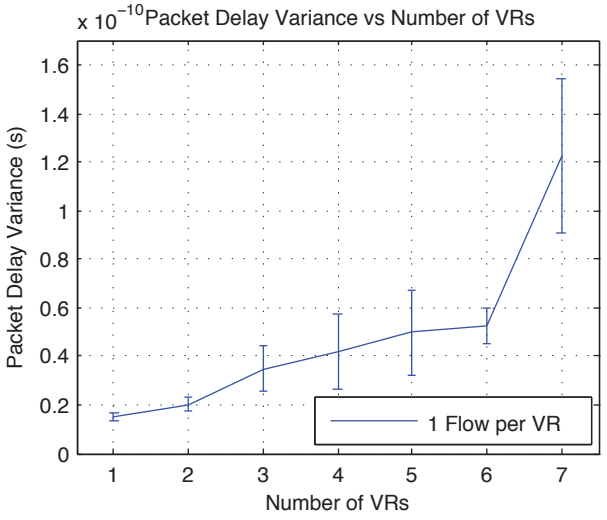


Figure 6. Packet delay variance for a single flow per active router for different numbers of VRs

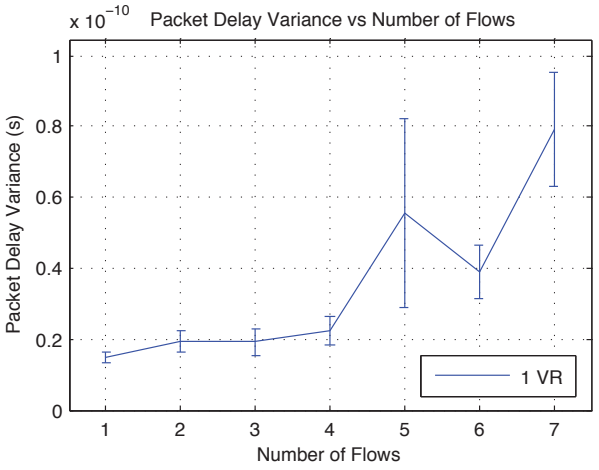


Figure 7. Packet delay variance for a single active VR with different numbers of flows

We can see that the general trend is that the variance of the packet delay, as well as the confidence intervals, increase with the number of active VRs and flows per VR, with a few exceptions. This means that, as the number of active VR and flows per VR grow, the delays become more statistically unstable on 2nd order time measures: while the average packet delay remains constant, the variance increases on average but becomes more unstable between runs.

In figures 8 and 9, it is possible to observe the behavior of the average packet delay, as well as the packet variance, when different combinations of VRs and flows per VR are activated up to a combined maximum of 7 packet flows.

For the average packet delay in Figure 8, we can see that it is similar regardless of the number of VRs and flows per VR, always close to 1.15 ms.

When we observe the variance of the delays of the packets in Figure 9, we can see very clearly that the larger the number of active VRs and the larger the number of flows per VR, the larger is the variance of the packet delay. For the same cases analyzed in Figures 8 and 9, we also studied the throughtput behavior. For all these cases we obtained values of throughput always averaging very close to 100% with very small confidence intervals. This is expectable due to the small bandwidth used by each flow (1Mbps) which, even when considering the maximum combination of 7 flows, is still too low to be affected.

In figure 10 it is represented the packet delay variation while varying the number of active routers.

The packet delay variation is similar for 1 and 2 VRs; for

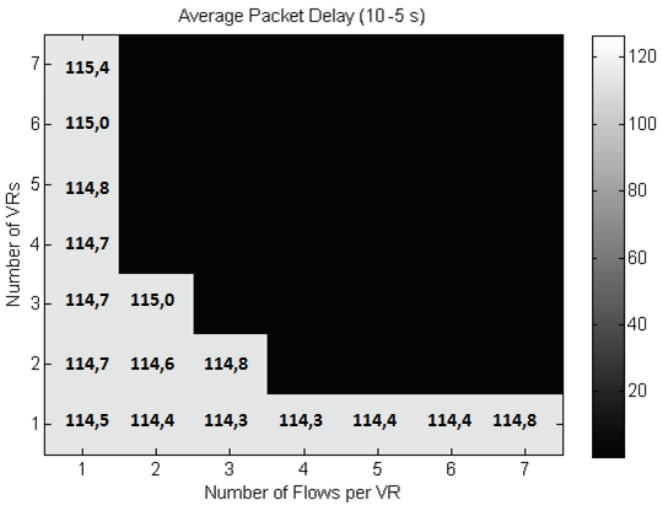


Figure 8. Average packet delay for a maximum total of 7 flows

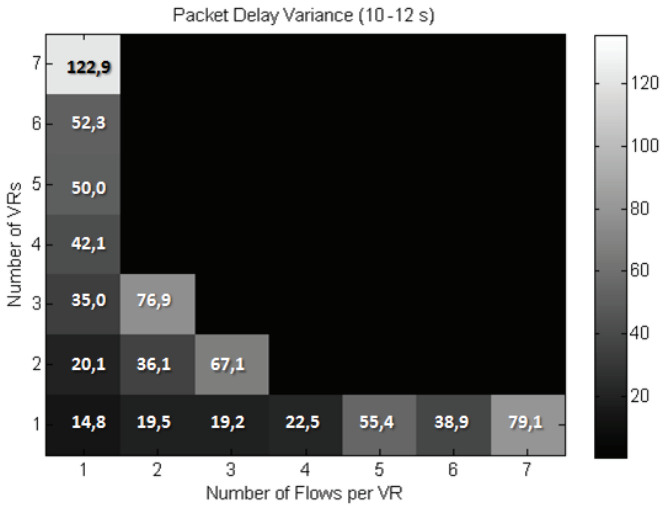


Figure 9. Packet delay variance for a maximum total of 7 flows

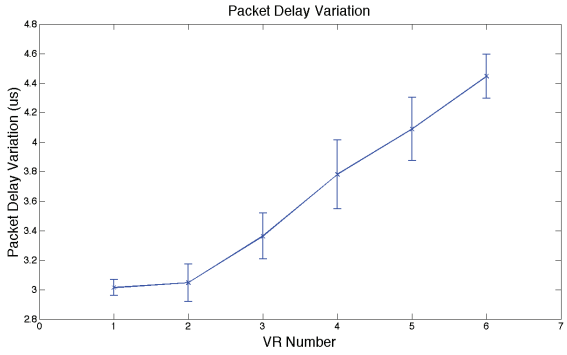


Figure 10. Packet Delay Variation while varying the number of VRs

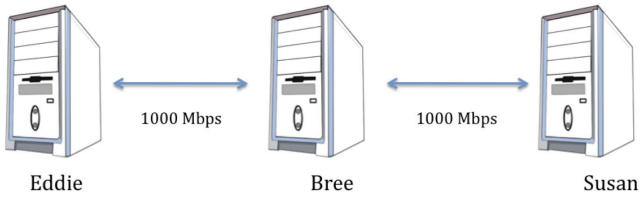


Figure 11. Experimental Apparatus - Throughput

2 or more VRs, it increases almost linearly with the number of VRs.

V. Evaluation - Throughput

In this section we start with the description of the testbed considered to evaluate the network throughput, and furthermore, we assess and analyze the network throughput as a function of the number of virtual routers, the number of network flows, and also as a function of the CPU load.

A. Testbed Configuration

The purpose of these tests is to analyze the behavior of the throughput with a different number of virtual routers and CPU load. The changes made to the configuration of the testbed are due to the fact that the ports on the hubs used in the previous setup are limited to 10 Mbps.

In this configuration, only three machines were used (see figure 11: Eddie as a transmitter; Susan as a receiver; and Bree where the VRs were mounted. Bree has an Intel Xeon E3110 with two CPU cores (3.0 GHz each) and 6GB of system memory.

The three nodes are directly connected through Ethernet cables with a bandwidth of 1000Mbps. To test the throughput the software *Iperf* was used, which will allow the measuring of the transmission rate between Eddie and Susan. In all the tests, 15 runs of 30 seconds of traffic were analyzed. The traffic is composed of TCP packets with a fixed size of 16 KB. To separate the different flows of traffic, VLAN tagging was used.

B. Results - Throughput

In order to establish a reference value, the throughput between Eddie and Susan was first measured without the use of VRs. This reference serves as a base comparison for the upcoming results. The throughput registered was constant and with a value of 940 Mbps.

In the first part of these tests, the throughput was measured while varying the number of active VRs; the number of VRs ranges from 2 to 7. The obtained results can be seen in figure 12 where the throughput shown is the combined value of all the flows. Assuming the reference value corresponds to 100% of the available throughput, the losses range from 13,5% to 4,9% (with 7 VRs and 4 VRs). Also, during these tests it was possible to observe that the hypervisor manages to make a fair distribution of resources. Looking at the flows' throughput individually, the values registered showed a small dispersion.

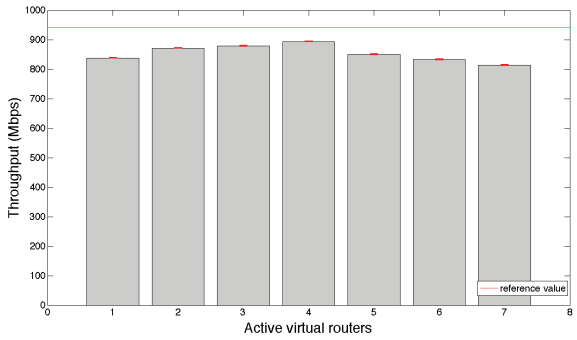


Figure 12. Throughput behavior while varying the number of VR

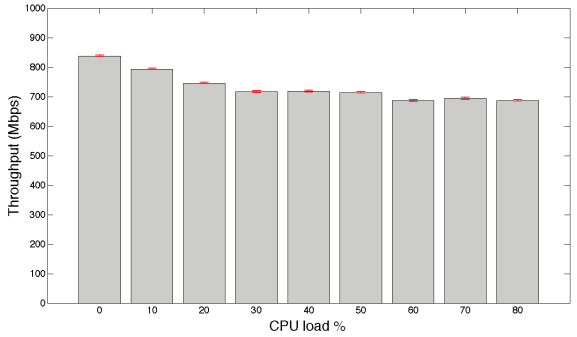


Figure 13. Throughput behavior while varying CPU load

To assess the influence of the CPU load in the VRs performance, a program called *Lookbusy* [19] was used. This software uses infinite loops in multiple threads to force the CPU load to a predetermined value. During this test only 1 VR was active.

The results can be seen in figure 13 with the confidence intervals of 90%. The results show that the amount of CPU load significantly affects the throughput. The throughput decreases steadily for CPU loads below 30%, while for higher loads the throughput seems to stabilize around 700 Mbps. Before making these tests, a trial run was made in which no virtual routers were used. During this trial run, the throughput seemed independent from the CPU load.

VI. Conclusion & Future Work

The main goal behind these tests was to establish a profile in terms of performance of network virtualization using the NVSS. The tests using TCP traffic allowed the retrieval of the throughput behavior while varying the number of virtual routers and CPU load. The tests with UDP traffic permitted an assessment of the behavior of packet delay in terms of average value and variance, as well as its variation, for different amounts of active routers and flows per router.

The results obtained for the throughput show that network virtualization has a negative impact on performance, as expected. The best results were obtained with 4 VRs, with the throughput presenting a loss of around 5%. Up to 4 VRs, the throughput increases with the number of active routers, which is probably due to a more efficient use of the bandwidth. Beyond this point, increasing the number of VRs decreases the global throughput. It should also be noted that having 7 VRs on the same machine never led the CPU load to exceed the 10% threshold, although it got close to this value. Looking at the results of the throughput behavior in terms of CPU load, at 10% of this load the throughput stays near 800Mbps. These two facts together lead us to believe that the throughput decrease after 4 VRs must be due to the increase of CPU load.

The measurements on the average packet delay and packet delay variance showed that, for different combinations on the number of VRs and flows per router, the average delay is mostly constant while the delay variance increases with both the number of VRs and the number of flows. We can also see that the confidence intervals for the variance also increase in the same way, which shows an increasing degree of instability for 2nd order parameters of the delay probabilistic distribution. Apart from the results for 1 VR, the packet delay variation increases almost linearly with the increase of VRs. In qualitative terms, the packet delay variation is low, even for the worst case where it reaches 0,4% of the packet delay. Also, looking at these results from an absolute value point of view, values in the order of microseconds indicate a good performance. Several types of data services which operate in real time, e.g. video or VoIP calls, can support a significant amount of average delay but need this delay to be stable. Therefore, one must make sure that the packet delay variance falls within certain limits. With this analysis, we analyzed how virtualization impacts this important aspect for real-time data services.

Looking at the combined results of the TCP and UDP traffic, the obvious conclusion is that up to 4 VRs network virtualization has minimum impact on throughput. Average packet delay is also minimally impacted, even though its variance and variation increase as the number of VRs and flows increases. Also, it seems that the CPU load should be the main factor behind the deterioration of the throughput performance. Even though using newer and better hardware should improve greatly the results obtained, given the age of the hardware used, the trend should remain that the more flows and VRs, the larger is the packet delay variance and jitter and the lower is the throughput.

There are several paths which can be trailed to proceed from this work. It would be important to study how the use of other virtualization technologies and hardware impacts the throughtput and packet delay statistics. On the other hand, it would also be relevant to have a direct comparison between the performances of virtualized and non-virtualized networks for the same kind of services. Experimenting with a larger number of VRs and network flows would allow us to have a more complete and detailed view of the impacts of virtualization; it would also allow us to relate the CPU load with the number of VRs, since for the number of VRs considered in this paper

the CPU load was always low.

REFERENCES

[1] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE*, vol. 47, no. 7, pp. 20 –26, july 2009.

[2] B. Ahlgren, P. Aranda *et al.*, "Content, connectivity, and cloud: ingredients for the network of the future," *Communications Magazine, IEEE*, vol. 49, no. 7, pp. 62 –70, july 2011.

[3] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE*, vol. 47, no. 7, pp. 20 –26, july 2009.

[4] J. Nogueira, M. Melo *et al.*, "Network virtualization system suite: Experimental network virtualization platform," in *TridentCom 2011, 7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2011.

[5] 4WARD Consortium, "Virtualisation approach: Concept," ICT-4WARD project, Deliverable D3.1.1, Tech. Rep., Sep. 2009.

[6] J. Nogueira, M. Melo *et al.*, "A distributed approach for virtual network discovery," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, dec. 2010, pp. 277 –282.

[7] ——, "Virtual network mapping into heterogeneous substrate networks," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, 28 2011-july 1 2011, pp. 438 –444.

[8] L. Peterson and T. Roscoe, "The design principles of PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, p. 11–16, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1113361.1113367

[9] N. Egi, A. Greenhalgh *et al.*, "Evaluating xen for router virtualization," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, 2007, p. 1256–1261. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4317993

[10] P. Barham, B. Dragovic *et al.*, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, p. 164–177, 2003.

[11] S. Bhatia, M. Motiwala *et al.*, "Trellis: A platform for building flexible, fast virtual networks on commodity hardware," in *Proceedings of the 2008 ACM CoNEXT Conference*, 2008, p. 72. [Online]. Available: http://dl.acm.org/citation.cfm?id=1544084

[12] N. Egi, A. Greenhalgh *et al.*, "Towards high performance virtual routers on commodity hardware," in *Proceedings of the 2008 ACM CoNEXT Conference*, 2008, p. 20. [Online]. Available: http://dl.acm.org/citation.cfm?id=1544032

[13] ——, "A platform for high performance and flexible virtual routers on commodity hardware," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 127–128, Jan. 2010. [Online]. Available: http://doi.acm.org/10.1145/1672308.1672332

[14] Y. Wang, J. van der Merwe, and J. Rexford, "VROOM: virtual routers on the move," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Networking*, 2007.

[15] Y. Wang, E. Keller *et al.*, "Virtual routers on the move: live router migration as a network-management primitive," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, p. 231–242, Aug. 2008.

[16] "Wireshark · go deep." http://www.wireshark.org/. [Online]. Available: http://www.wireshark.org/

[17] A. Tirumala, F. Qin *et al.*, "Iperf: The TCP/UDP bandwidth measurement tool," *htt p://dast. nlanr. net/Projects*, 2005.

[18] "RFC 3393 - IP packet delay variation metric for IP performance metrics (IPPM)," http://tools.ietf.org/html/rfc3393. [Online]. Available: http://tools.ietf.org/html/rfc3393

[19] "Lookbusy – a synthetic load generator." [Online]. Available: http://www.devin.com/lookbusy/

Protocolo de encaminhamento para redes móveis usando estruturas binárias eficientes

João Trindade
Instituto Superior Técnico / INESC-ID
Av. Rovisco Pais,
1049-001 Lisboa, Portugal
Email: jtrindade@tagus.inesc-id.pt

Teresa Vazão
Instituto Superior Técnico / INESC-ID
Av. Rovisco Pais,
1049-001 Lisboa, Portugal
Email: teresa.vazao@tagus.inesc-id.pt

***Resumo*—Protocolos de encaminhamento não geográficos são pouco eficientes quando aplicados a redes móveis de grande escala compostas por centenas de nós. Por outro lado, protocolos de encaminhamento geográficos possuem a desvantagem de necessitar de um sensor de localização. Este requisito aumenta o custo do equipamento presente em cada nó, bem como o consumo de energia dos dispositivos. Neste artigo propomos um protocolo de encaminhamento para redes móveis, que é escalável a redes compostas por centenas de nós. O protocolo não necessita de qualquer equipamento de localização e é adaptado para dispositivos com poucos recursos de memória e/ou processamento. Este objectivo é conseguido através do uso de filtros *bloom* para armazenar e espalhar informação topológica de uma forma eficiente. Na metodologia seguida, os nós não reencaminham para outros nós mensagens com informação topológica. Para tornar o processo eficiente, cada nó agrega a informação topológica que recebe dos seus vizinhos diretos com a sua própria e somente o resultado desta operação é transmitido para os restantes nós. Várias simulações foram efetuadas no simulador de redes Qualnet de modo a validar o algoritmo proposto pelo HRAN. Os resultados obtidos foram comparados com outros protocolos não-geográficos para redes móveis.**

I. Introdução

Atualmente o uso ubíquo de dispositivos *wireless* por parte de grande parte da população tem enaltecido o papel das redes móveis *Ad hoc*, também denominadas por MANETs. Este tipo de redes tem sido alvo de vários estudos devido às suas características, como o requisito de uma fonte de energia limitada e/ou as frequentes mudanças topológicas [1] [2]. A maior parte dos protocolos de encaminhamento propostos para uso em MANETs são tradicionalmente categorizados em protocolos pró-ativos, reativos ou um modelo hibrido destas duas propriedades. Protocolos de encaminhamento pró-ativos são caracterizados por manter rotas permanentes para todos os possíveis destinos, independente do tráfego existente na rede. Pelo contrário, protocolos reativos somente estabelecem novas rotas quando são estas são requeridas por parte de um nó de origem. Entre os protocolos pró-ativos podemos encontrar o Optimized Link State Routing protocol (OLSR), o Destination-Sequence Distance Vector (DSDV) e o Temporally Ordered Routing Algorithm (TORA). Como representantes dos protocolos de encaminhamento reativos, podemos encontrar os protocolos Dynamic source routing (DSR) ou o Ad hoc On-demand Distance Vector (AODV). Existem também protocolos, normalmente designados como híbridos, que apresentam características reativas e pro-ativas. Um exemplo de um destes protocolos é o Zone Routing Protocol (ZRP).

Vários estudos foram realizados comparando as várias alternativas. Em [3], os autores simularam protocolos pro-ativos e reativos em redes de média escala e concluíram que este últimos providenciam bons resultados em termos de atraso e percentagem de pacotes entregues. Infelizmente estes protocolos também são os que causam mais sobrecarga na rede quando o pedido de rotas aumenta. Nesse estudo podemos igualmente verificar que os protocolos pró-ativos originam rotas com menos qualidade, bem como apresentam um maior número de pacotes perdidos. Por outro lado, se medirmos o número de mensagens de controlo enviadas, este conjunto de protocolos é considerado mais eficiente.

Os autores em [4] [5] demonstraram que os resultados dos vários protocolos variam conforme os cenários em que são testados. Nomeadamente em [6] e [7] ficou demonstrado que protocolos de encaminhamento não geográficos não escalam para redes compostas por centenas de nós. Nesse tipo de redes o número de mensagens de controlo cresce exponencialmente, penalizando severamente a rede, em termos de recursos, somente com a tarefa de descobrir novas rotas. Protocolos de encaminhamento geográficos têm a desvantagem de requerer um sensor de localização, como um dispositivo GPS, para executar os seus algoritmos de encaminhamento [8]. Em muitos casos o uso deste equipamento é inadequado pois origina um aumento de consumo de energia significativo e encarece o preço do equipamento presente em cada um dos nós. Além disso, o encaminhamento de pacotes é penalizado pois requere um componente adicional: um serviço de localização do destino.

Neste artigo apresentamos e estudamos um novo protocolo de encaminhamento denominado de HRAN (Heat Routing for Ad hoc Networks). O HRAN é um protocolo de encaminhamento escalável para redes de grande escala, no qual os nós possuem recursos limitados (em termos de memória, processamento e largura de banda) e que não requere o uso de dispositivos GPS.

O HRAN usa filtros *bloom* para armazenar e transmitir informação topológica de uma forma eficiente. No HRAN os nós não reenviam mensagens provenientes de outros nós. De forma a poupar recursos, os nós ao receberem men-

sagens com informação topológica, juntam através de uma operação binária de "*OU*" esta informação com o seu atual conhecimento da rede. Somente o resultado desta operação é partilhado com os outros nós. Esta operação permite ao HRAN espalhar informação topológica de uma forma eficiente, que depois pode ser utilizada na procura de rotas quando estas são requeridas. Este mecanismo permite igualmente criar zonas com informação topológica em torno de uma rota, que podem ser usadas quando for necessário reparar essa rota ou melhorá-la em termos de número de nós que a constituem.

Atualmente já existem algumas soluções propondo o uso de filtros *bloom* em protocolos de encaminhamento para redes *ad hoc* [9]. No protocolo Gradient-ascending routing via footprints (GRASP) [10], os filtros *bloom* são usados para indicar se um nó pertencente a uma rede de sensores é constituinte de uma rota. Devido ao âmbito deste protocolo se restringir somente a redes de sensores, não existe qualquer forma de lidar com mobilidade nos nós. Outro protocolo de encaminhamento que utiliza filtros *bloom* é o Table Attenuation Routing Protocol (TARP) [11]. Neste protocolo cada nó possui um conjunto de filtros *bloom* por vizinho. A posição de um nó nesse conjunto de filtros *bloom* indica a distancia a que o nó se encontra. O problema existente no TARP é ser muito penalizado por redes em que a densidade de nós seja elevada ou em redes com bastante mobilidade.

O resto deste artigo encontra-se estruturado da seguinte forma. Na secção 2 o HRAN é descrito, bem como o conceito de filtro *bloom* com informação temporal. Na secção 3 são apresentados os resultados da comparação do HRAN com outros protocolos. Finalmente na secção 4 são explicitadas as conclusões.

II. Descrição do HRAN

Esta secção irá apresentar o funcionamento do protocolo de encaminhamento HRAN. Inicialmente irá ser apresentada uma breve descrição do algoritmo, seguida de uma explicação dos filtros *bloom* com informação temporal. Finalmente irá ser realizada uma descrição detalhada do algoritmo.

A. Conceito

O HRAN tenta imitar um paradigma de "rastos de calor" existente no mundo físico mapeando-o para uma representação de topologia de rede. Cada nó cria uma região de calor à sua volta através de emissão periódica de mensagens de calor enviadas aos seus vizinhos. Nós dentro desta região "sentem" o calor e têm uma pista em como chegar ao destino. Nós mais perto do destino "sentem" o nó mais perto enquanto que nós mais afastados do destino sentem a pista de calor mais fria. Nós que se encontrem fora da região de calor de um nó, sempre que desejarem comunicar com ele terão que procurar lançar uma procura aleatória que prossegue pela rede até atingir uma região de calor do destino. A partir desse momento a procura passa a ser dirigida, passando de zonas de calor mais frias para mais quentes. Finalmente depois de uma rota ser encontrada, existe a criação de um túnel de calor. Este mecanismo é útil para auxiliar futuras procuras pelo mesmo destino, reparar possíveis falhas da rota e melhorar a rota em termos de número de nós que a constituem.

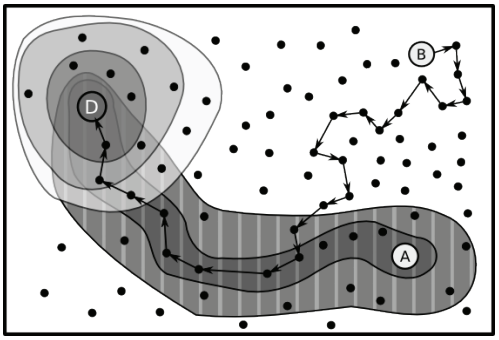


Figura 1. Funcionamento do HRAN: representação dos gradientes de calor do nó D; túnel de calor da rota existente entre o nó A e o nó D e a procura aleatória e procura direccionada entre o nó B e o nó D.

O funcionamento do HRAN pode ser visualizado na figura 1, na qual estão presentes os vários gradientes de calor do nó D (destino). Nós localizados em zonas com cores mais escuras indicam que a presença do nó D encontra-se num gradiente mais quente que em nós localizados em zonas com cores mais claras. A zona com riscas indica o túnel de calor. O processo de comunicação entre o nó B e D encontra-se também representado o conceito de procura aleatória e procura direcionada.

Toda a informação referente ao "calor" dos nós encontra-se representada através de uma adaptação de um filtro *bloom* [12] [13]. Um filtro *bloom* é uma estrutura de dados eficiente, em termos de espaço, que permite testar se um elemento é membro de um conjunto, mas que admite a existência de falsos positivos. Um filtro de *bloom* é instanciado através de um vector de bits de tamanho m e de k funções *hash* distintas. Cada uma destas funções de *hash* mapeia um elemento do conjunto para uma das posições da matriz m. Para adicionar um elemento a um conjunto, basta executar cada uma das funções de *hash* com o seu identificador para obter k posições. Cada uma destas posições do vector de bits é posta com o valor 1. Para consultar se um elemento pertence ao conjunto, volta-se a inserir o identificador em cada uma das k funções de *hash* para obter as posições do vector. Se algum dos bits nessas posições for 0, o elemento não pertence ao conjunto.

No protocolo HRAN uma variante destas estruturas binárias de reduzida dimensão é agrupada em gradientes cujas posições indicam a distância, em termos de número de nós intermédios, a que os nós se encontram uns dos outros. Estes filtros *bloom* adaptados são periodicamente trocados entre vizinhos, garantindo desta forma que informação sobre a topologia da rede é disseminada.

B. Filtro bloom com informação temporal

O protocolo de encaminhamento HRAN requer uma extensão de funcionalidade das estruturas tradicionais de filtros *bloom*, de modo a que se possa verificar se um objecto foi adicionado num certo período temporal. Esta extensão permite

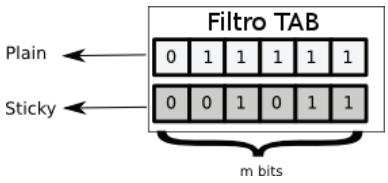


Figura 2. Filtro TAB

aos nós gradualmente perder informação do calor de vizinhos que se tenham afastado, ou que simplesmente tenham falhado. Para obter esta funcionalidade foi criado um novo tipo de filtro *bloom* denominado filtro TAB (*Time-Aware Bloom*).

O filtro TAB é um conjunto de dois filtros *bloom* com o mesmo tamanho M e o mesmo grupo de K funções de *hash*. Os dois filtros *bloom* são denominados como o filtro *sticky* e o filtro *plain* como é possível visualizar na Figura 2. O filtro *sticky* é usado para identificar se um bit foi inserido no último intervalo temporal (δ^n). O filtro *plain* contém os bits inseridos durante o último e o anterior intervalo temporal ($\delta^n || \delta^{n-1}$). Os filtros TAB são sempre utilizados em vectores, sendo de seguida explicado o funcionamento das operações de inserção, procura e atualização nestes filtros.

Para adicionar um elemento X na posição i de um vector de filtros *TAB*, o algoritmo realiza a operação de inserção normal nos filtros *bloom plain* e *sticky* presentes nessa posição do vector. Esta operação consiste em realizar as K funções de *hash* no elemento a inserir de modo a obter as k posições. Nessas posições é colocado o valor de 1 em ambos os filtros *bloom*.

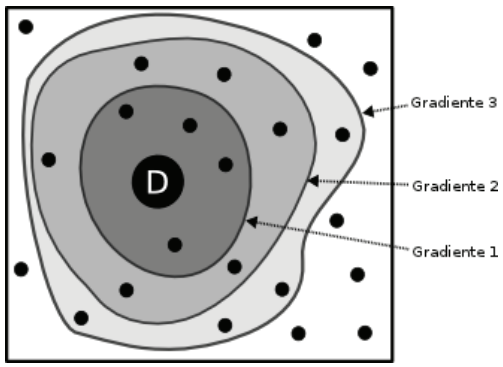
Para procurar um elemento Y num vector de *TAB*, a procura é somente realizada nos filtros *plain* presentes no vector. Esta operação consiste em fazer *hash* ao elemento a procurar com as K funções de hash e para cada uma das posições obtidas, verificar em cada filtro *plain* se todos os bits respectivos contêm o valor 1. Se isto for verdade, a procura encontrou o objecto. Se todos os filtros do vector forem percorridos e esta correspondência nunca for encontrada o objecto não foi encontrado.

Finalmente, a operação de atualização, realizada em períodos regulares δ, percorre todos os filtros *plain* existentes no vector e reescreve-os com o valor existente no filtro *sticky* presente na mesma posição do vector. Após esta operação, todos os filtros *sticky* são reescritos com o valor de 0 para iniciar um novo intervalo temporal δ.

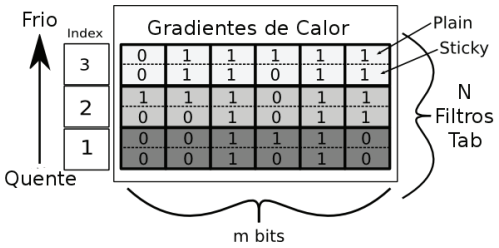
C. Descrição Funcional

O protocolo de encaminhamento HRAN encontra-se dividido em três fases de execução: construção da camada de calor, procura de calor e manutenção de rota. As próximas secções irão detalhar cada uma destas fases.

1) Construção da camada de calor: Na fase de construção da camada de calor cada nó envia a sua informação topológica da rede para os nós vizinhos seguindo um modelo pro-ativo. Cada nó armazena a informação que possui sobre a topologia da rede num vector de filtros *TAB* com N posições. Cada uma



(a) Camada de calor de um nó



(b) Gradiente de calor

Figura 3. Camada de calor de um nó

desta posições corresponde a um nível no gradiente de calor. Nós presentes em filtros TAB contidos em posições do vector menores são considerados como sendo mais quentes que nós contidos em posições mais próximas de N.

A figura 3 representa a camada de calor do nó D em que os filtros foram configurados para um tamanho de 6 bits. No exemplo apresentado o protocolo HRAN foi configurado com 3 gradientes ($N = 3$). Nós em camadas cinzentas escuras significam que possuem o nó D em posições do gradiente mais quentes do que nós localizados nas zonas cinzentas claras. Nós que não tem o nó D nos seus vectores de filtros TAB, não possuem qualquer indicação sobre a localização do nó D.

O processo de construção da camada de calor é executado por cada nó ao enviar mensagens de *HELLO* em intervalos temporais regulares. Cada mensagem de *HELLO* contem um vector com $N-1$ filtros *bloom*, pois o gradiente mais *frio* (N) não é transmitido. Os filtros de *bloom* contidos nas mensagens de *HELLO* são uma copia dos filtros *plain* do nó emissor do gradiente de calor 1 ao $N-1$. Somente filtros *plain* são enviados de forma a reduzir a quantidade de informação enviada para a disseminação de informação topológica.

Quando um nó recebe uma mensagem de *HELLO* uma operação binária de "OU" é realizada entre os filtros *bloom* transmitidos pela mensagem de *HELLO* e o gradiente de calor do próprio nó. Como demonstrado na figura 4 a cópia dos filtros contidos na mensagem começam na posição 2 enquanto a posição na estrutura do nó começa na posição 1. Este mecanismo permite que a informação sobre um só se vá ficando mais fraca quanto mais longe estivermos dele.

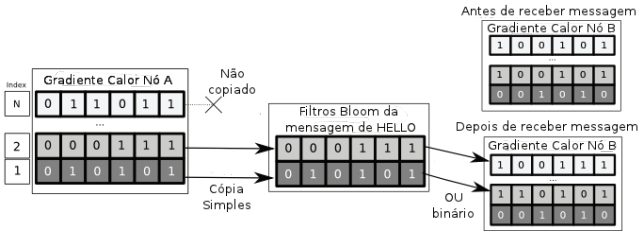
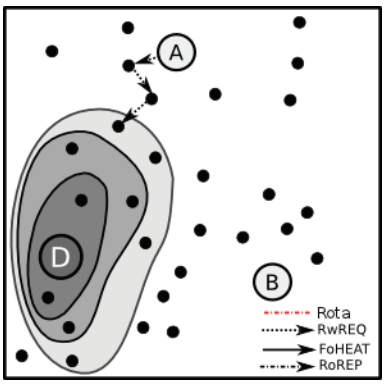


Figura 4. Troca de mensagens de *HELLO* entre nós vizinhos

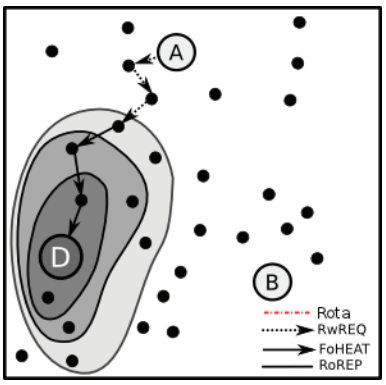
2) Descoberta de rota: Quando um nó requer uma nova rota inicia um processo descoberta de rota no qual uma mensagem de procura aleatória denominada *RwREQ* percorre a rede até encontrar o "calor" do nó destino. Quando um nó recebe uma destas mensagens analisa o seu gradiente de calor verificando se possui informação sobre o nó destino. Se a resposta for afirmativa, o nó entra no modo de procura direcionada, no qual uma mensagem de seguir calor (*FoHEAT*) é transmitida contendo o nível de calor encontrado. Se não encontrar informação reenvia a mensagem para um dos seus vizinhos. Nós que uma mensagem de *FoHEAT*, caso possuam um nível de calor superior ao indicado na mensagem, reenviam a mensagem atualizando-a com o seu próprio nível de calor. Somente nós nesta condição reenviam a mensagem. Nós que tenham o nó destino em gradientes mais frios ignoram a mensagem de *FoHEAT*. Esta opção permite reduzir o número de mensagens necessárias para a descoberta de uma nova rota. Quando a mensagem de *FoHEAT* chega ao nó destino, a rota é dada pela lista de nós presente na mesma. Nesta altura o protocolo inicia o modo de returno, no qual uma mensagem (*RoREP*) é enviado do destino à origem pelo caminho inverso ao descoberto anteriormente.

Devido à natureza aleatória do processo de descoberta de rotas, é possível que informação de calor do nó destino não seja encontrada em tempo útil ou que a existência de falsos positivos nos filtros *bloom* leve a procura do destino por direções erradas. Devido a estas razões tem que existir um mecanismo de redundância que seja executado quando nenhuma rota é descoberta. Caso o nó que pediu o estabelecimento de rota não receba uma mensagem de *RoREP* dentro de um certo período temporal, é executado um protocolo reativo (como o DSR) para o estabelecimento da rota. Esta opção permite a criação de rotas, e de túneis de calor, que irão ser descritos na próxima secção, facilitando assim futuras procuras.

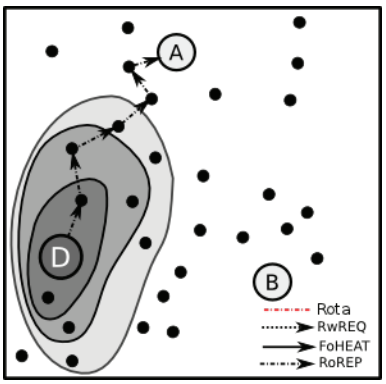
3) Manutenção de rotas: O processo de manutenção de rotas encontra-se dividido em três tarefa: (i) a *criação de túneis de calor* cria e mantém os túneis de calor quando uma rota está a ser usada para transferir pacotes de dados; (ii) a *reparação de rotas* recupera a quebra de uma rota devido à falha de um ou mais dos nós que a compõem; (iii) o *processo de melhoria de rotas* iterativamente melhora as rotas de modo a reduzir o impacto da mobilidade e da aleatoriedade existente no processo de descoberta de rota.



(a) Procura aleatória



(b) Procura direcionada

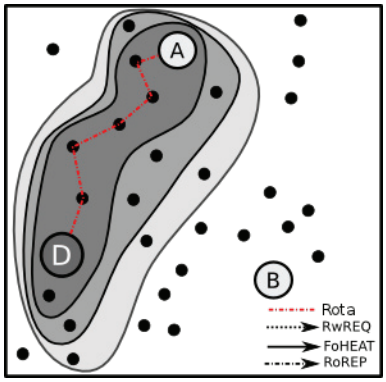


(c) Retorno

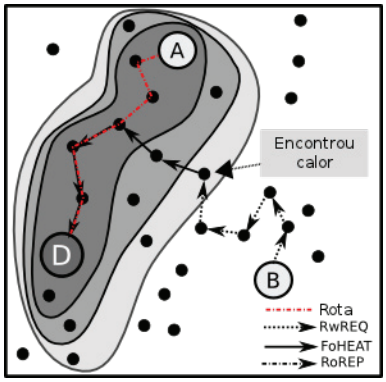
Figura 5. Processo de descoberta de rotas

a) Criação de túneis de calor: Quando um pacote dados é encaminhado por um nó intermédio, um túnel de calor é criado através da inserção do identificador no filtro *TAB* presente na posição 1 do gradiente de calor desse nó. Este processo mantém atualizada a informação sobre "calor" das rotas usadas atualizado, enquanto que, para rotas não usadas, essa informação é gradualmente perdida pelo processo normal de atualização dos filtros *TAB*.

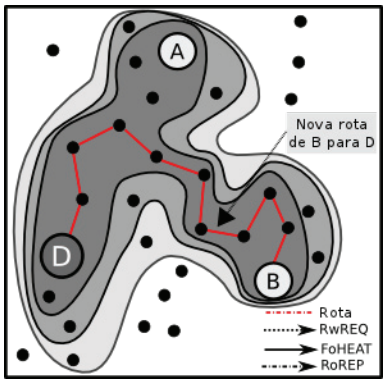
A figura 6a demonstra um túnel de calor do nó A ao nó D. Novamente nós em zonas de cinzento mais escuras possuem



(a) Túnel de calor para o nó D

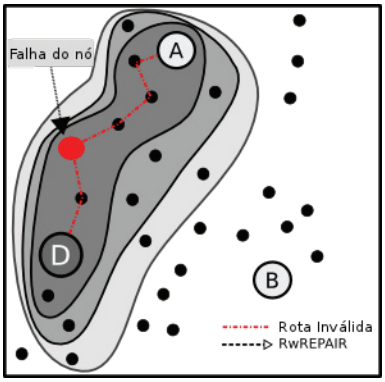


(b) Nova rota descoberta

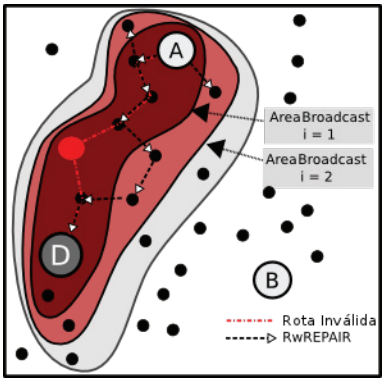


(c) Criação túnel de calor

Figura 6. Descoberta de rota e criação de túnel de calor respectivo



(a) Falha de nó intermédio



(b) Reparação de rota com i=2;

Figura 7. Processo de reparação de rota

informação sobre o nó D num gradiente mais baixo (mais quente) que nós que se encontram em zonas mais claras.

Existem dois objetivos para a criação de túneis de calor em rotas ativas. O primeiro objectivo consiste em auxiliar futuras procuras de rotas por parte de outros nós, como se encontra demonstrado na figura 6b. O outro objectivo é possibilitar a reparação de rotas localmente, através do uso de poucas mensagens. A figura 6c demonstra a criação de um novo túnel de calor após a descoberta da nova rota.

b) Reparação de rotas: Quando é detectada uma falha de rota é desencadeado um processo que utiliza o túnel de calor previamente criado para reparar a rota de uma forma eficiente e rápida. Durante este processo, é feito o *broadcast* iterativo de mensagens de reparação (*REPAIR*) em sucessivas zonas de calor. Este processo termina quando o destino é encontrado ou quando já não existam mais zonas de calor para procurar a nova rota.

Em mais detalhe, para executar o processo de reparação, uma mensagem *REPAIR* é enviada para todos os nós na área de reparação em modo de *broadcast*. Visto que a zona de calor em que os nós possuem o nó destino no gradiente mais quente (gradiente 1) ser a rota que falhou, a primeira área de reparação começa por ser o gradiente 2. A área de reparação é progressivamente incrementada até que atinja o gradiente N. Nesta fase, a rota é considerada como irreparável e um novo processo de descoberta de rota é iniciado. A figura 7 demonstra o processo de reparação, em que uma nova rota foi descoberta com o gradiente de 2.

c) Melhoria de rotas: Devido ao processo aleatório da descoberta de rotas, no protocolo HRAN não é garantido que as rotas descobertas sejam as mais curtas. Deste modo, para melhorar a qualidade das rotas, o protocolo HRAN possui um mecanismo de melhoria de rotas aproximando-as iterativamente da melhor rota possível. Este processo também é útil

para diminuir o efeito da mobilidade dos nós na qualidade nas rotas.

O processo de melhoria de rotas do HRAN usa os túneis de calor previamente criados para descobrir rotas mais curtas. Quando uma rota é usada, o HRAN conta o número de pacotes que são enviados por ela. Quando este valor atinge um limite pre-determinado, um processo de melhoria de rota é iniciado. Este processo consiste no envio em *broadcast* de mensagens de melhoria de rota (*RoIMP*) contendo a identificação do nó de destino. Esta mensagem é reenviada por todos os nós que contenham a identificação do nó destino em qualquer um dos seus níveis no gradiente de calor. Este processo acontece recursivamente até que uma destas mensagens chegue ao nó de destino. Quando isto acontece, é desencadeado um processo de retorno com mensagens *RoREP* similar à descoberta de novas rotas.

Ao usar este mecanismo, as mensagens de melhoria de rota só são enviadas para nós que pertençam ao túnel de calor do nó destino. Isto reduz o número de mensagens transmitidas e permite que, caso exista, seja descoberta uma rota com menos nós.

III. Avaliação de desempenho

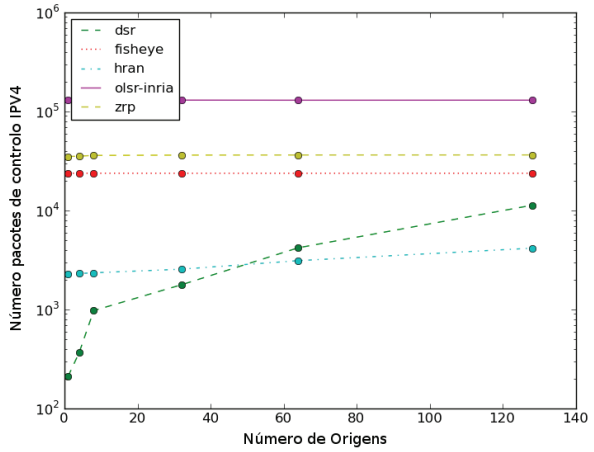
As simulações foram executadas no Qualnet versão 5.0.1. Foi escolhido este simulador pois suporta redes de larga escala com centenas de nós e possui uma biblioteca com vários protocolos para MANETs já implementados, possibilitando assim uma comparação fácil e fidedigna. Para todos os testes as condições de teste foram as seguintes:

- tamanho do mapa - *1200m* por *1200m*
- distância de comunicação máxima -*150m*
- modelo de propagação - *Two-ray*
- tempo simulado - *110s*
- velocidade dos nós - *15m/s*
- direção dos nós - *aleatória*

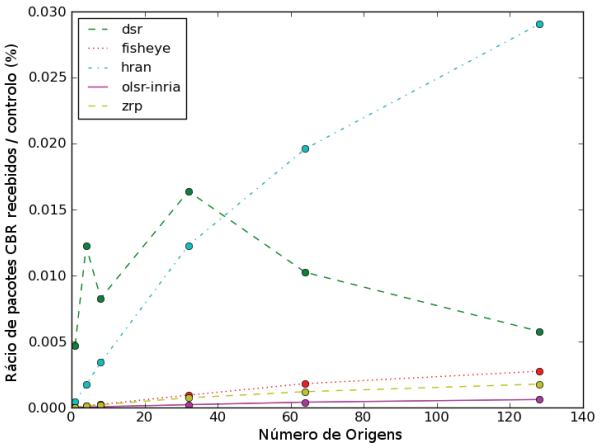
Os parâmetros configurados para o HRAN foram os seguintes. Intervalo entre mensagens de *HELLO* de 7s, uso de 4 níveis nos gradientes de calor, tamanho de filtros *bloom* de 256 bits, um tempo de atualização dos filtros TAB de 21 segundos e um intervalo de melhoria de rotas de 20 segundos.

A. Aumento do número de pedidos de rota

O primeiro teste de desempenho consiste em medir o número de pacotes enviados pelo o protocolo de encaminhamento para descobrir novas rotas em uma rede composta por 200 nós. Neste cenário a variável independente escolhida foi o número de nós de origem. O nó de destino mantem-se igual para todos os testes. Como é possível verificar na figura 8a os protocolos ZRP e OLSR (implementação INRIA) mantêm um valor perto de constante independente do número de rotas pedidas. Para valores de nós de origem reduzidos, o protocolo DSR necessita de enviar menos mensagens que o HRAN, mas à medida que o número de pedidos de rotas aumenta, o mecanismo de camadas e túneis de calor presentes no HRAN, melhora a eficiência do processo de procura de rotas. Na figura 8b é possível observar a percentagem de pacotes de



(a) Número de mensagens de controlo



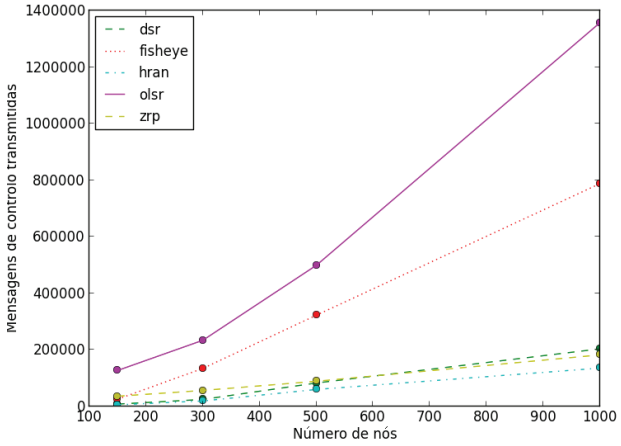
(b) Rácio de mensagens entregues para mensagens de controlo

Figura 8. Desempenho protocolos com o aumento de nós de origem

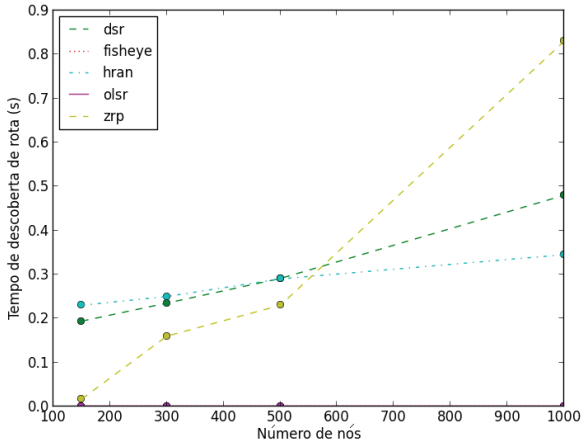
controlo enviados por cada pacote de dados recebido pelo destino, novamente o protocolo de encaminhamento HRAN apresenta os melhores valores para números de pedidos de rota superiores.

B. Aumento do número de nós

A principal funcionalidade do protocolo de encaminhamento HRAN é a capacidade de suportar redes compostas por centenas de nós. O segundo teste consiste em analisar o número de mensagens transmitidas pelo protocolo de encaminhamento à medida que o número de nós na rede aumenta. Na figura 9a é possível observar o que para redes compostas por 150 nós tanto o DSR como o protocolo HRAN requerem menos pacotes de controlo que os restantes protocolos. Apesar de o DSR possuir um valor ligeiramente mais reduzido para a rede menor composta por 150 nós, para redes maiores o HRAN apresenta valores mais reduzidos seguido do protocolo ZRP. Em termos de do atraso referente à descoberta da nova rota podemos verificar na figura 9b que os dois protocolos



(a) Mensagens de controlo



(b) Tempo de descoberta de rota

Figura 9. Desempenho com aumento de número de nós

pró-ativos (OLSR e fisheye) têm sempre rotas estabelecidas, logo não possuem atrasos quando novas rotas são pedidas. Em relação aos restantes protocolos de encaminhamento é possível verificar que para redes de grande dimensão, o HRAN possui valores mais reduzidos devido ao uso das suas estruturas de calor para guiar procuras de novas rotas.

C. Tamanho das rotas

Finalmente, o último teste centra-se na análise do tamanho das rotas usadas pelo protocolos HRAN e DSR à medida que a simulação decorre. A figura 10 apresenta os resultados onde é possível verificar que apesar de o protocolo HRAN apresentar rotas mais longas, quando comparado com o DSR, o mecanismo de melhoria de rota ativado a cada 20 segundos gradualmente diminui o tamanho de rota para valores semelhantes ao DSR. Estes resultados validam a utilidade deste mecanismo para mitigar um dos problemas associados ao uso do protocolo HRAN devido à procura de rota parcialmente aleatória.

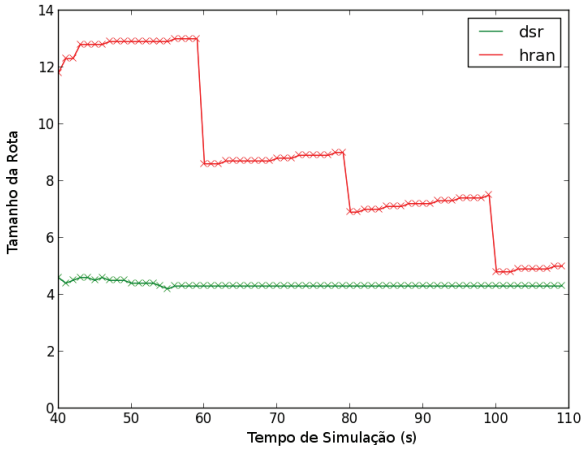


Figura 10. Tamanho médio das rota utilizadas

IV. Conclusões

Neste artigo foi apresentado o protocolo HRAN desenhado para redes MANET compostas por centenas de nós. O algoritmo desenvolvido assenta no uso de filtros *bloom* para armazenar informação topológica sendo esta usada para descobrir e manter rotas. Este tipo de estrutura de dados permite armazenar uma grande quantidade de informação binária de uma forma eficiente, reduzindo os recursos requeridos pelo protocolo de encaminhamento.

Os resultados das simulações demonstram que o HRAN requer menos mensagens de controlo que outros protocolo de encaminhamento e que o seu processo de reparação de rotas permite uma superior percentagem de pacotes entregues em cenários móveis. Como trabalho futuro será interessante estudar um mecanismo de adaptação dinâmica dos parâmetros de configuração do protocolo HRAN para as condições da rede.

Agradecimentos

Este trabalho foi financiado por fundos nacionais através da FCT - Fundação para a Ciência e a Tecnologia, no âmbito do projecto PEst-OE/EEI/LA0021/2011.

Referências

[1] Sarkar, N.I., Lol, W.G.: A study of manet routing protocols: Joint node density, packet length and mobility. In: Computers and Communications (ISCC), 2010 IEEE Symposium on. (2010) 515 –520

[2] Kumar, J., Rajesh, R.: Performance analysis of manet routing protocols in different mobility models. IJCSNS International Journal of Computer Science and Network Security **9** (2009) 22–29

[3] Das, S., Castaneda, R., Yan, J., Sengupta, R.: Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In: Computer Communications and Networks, 1998. Proceedings. 7th International Conference on. (1998) 153 –161

[4] wu Chin, K., Judge, J., Williams, A., Kermode, R.: Implementation experience with manet routing protocols. ACM SIGCOMM Computer Communications Review **32** (2002) 49–59

[5] Mohseni, S., Hassan, R., Patel, A., Razali, R.: Comparative review study of reactive and proactive routing protocols in manets. In: Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on. (2010) 304 –309

[6] Wang, Y., Dong, L., Liang, T., Yang, X., Zhang, D.: Cluster based location-aided routing protocol for large scale mobile ad hoc networks. IEICE Transactions **92-D** (2009) 1103–1124
[7] Boukerche, A.: Performance evaluation of routing protocols for ad hoc wireless networks. Mobile Network Applications **9** (2004) 333–342
[8] Karp, B., Kung, H.: Gpsr: Greedy perimeter stateless routing for wireless networks. In: Annual International Conference on Mobile Computing and Networking (Mobicom), ACM (2000) 243–254
[9] Broder, A., Mitzenmacher, M.: Network applications of bloom filters: A survey. Internet Mathematics **1** (2005) 485–509
[10] Lim, J.J., Shin, K.G.: Gradient-ascending routing via footprints in wireless sensor networks. Real-Time Systems Symposium, IEEE International **0** (2005) 298–307
[11] Gilbert, R., Johnson, K., Wu, S., Zhao, B.Y., Zheng, H.: Location independent compact routing for wireless networks. In: Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking. MobiShare '06, New York, NY, USA, ACM (2006) 57–59
[12] Tarkoma, S., Rothenberg, C., Lagerspetz, E.: Theory and practice of bloom filters for distributed systems. Communications Surveys Tutorials, IEEE (2011) 1–25
[13] Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM **13** (1970) 422–426

EMICOM: Enhanced Media Independent COnnection Manager

André Prata*, Daniel Corujo*, Pedro Gonçalves†, Diogo Gomes*
ESTGA† / DETI*, Universidade de Aveiro
Instituto de Telecomunicações
3810-193 Aveiro, Portugal
{andreprata, dcorujo, pasg, dgomes}@av.it.pt

***Abstract*—With the increasing amount of mobile interfaces combining different kinds of access technologies, ranging from Wi-Fi to 3G and LTE, the integration of flexible and media-independent link control mechanisms becomes of paramount importance. By employing an abstract way of obtaining access link status information and exercising control over the network interface operations, these control mechanisms become able to optimize device connectivity and network attachment. This paper presents EMICOM, an Enhanced Media Independent COnnection Manager framework where a GNU/Linux Network Manager and Link Service Access Points for the IEEE 802.3 and 802.11 technologies were implemented and integrated through cross-layer Media Independent Handover (MIH) mechanisms from the IEEE 802.21 standard. Through an open-source implementation of the framework, the (MIH) command set capabilities are extended, allowing the support of network association and authentication, as well as Layer 3 services such as IP configuration, providing a generic solution for optimal network connectivity management.**

***Index Terms*—802.21, Network Manager, Link layer**

I. Introduction

Current mobile devices offer various interfaces for network connectivity, either for Local, Metropolitan and Cellular networks. This motivates network providers to look at heterogeneous network deployments as a mean of expanding network capacity through the combination of high bandwidth solutions, such as Wi-Fi, with the geographically broader 3G and LTE solutions. User mobility in these scenarios presents an outstanding problem of optimal network technology selection. Current network managing solutions often base selection algorithms on signal quality and user preference policies, while requiring the various radios at the terminal to be enabled, in order to be aware of available alternatives for network connectivity.

This paper presents a framework for controlling and gathering information about different mobile node access interfaces via an abstract interface provided by the IEEE 802.21 standard [1]. This framework enables network selection algorithms to take as input variables such as energy consumption, cost per technology, cost per bit, application and service constraints, as well as network provided information regarding the surrounding environment, allowing for an optimized access interface selection. It also extends the IEEE 802.21 services to provide abstract Layer 2 association and security mechanisms, and Layer 3 static or dynamic IP configuration, enhancing the capabilities envisioned in the standard.

The remainder of this paper is structured as follows. Section II introduces the state of the art on network manager solutions residing in terminals. This is followed by Section IV were our framework is described, alongside extensions done to IEEE 802.21, as well as open-source implementations of 802.3 and 802.11 interfacing modules. This framework was subjected to an evaluation scenario presented in Section V. Finally, we conclude in Section VI.

II. Related work

With the different requirements placed by different existing access technologies being used to access the most diverse kind of services available through the Internet, different attempts for the provision of inter-technology network selection mechanisms have been proposed, at different levels. Most application deployments target one of two scenarios: management based on user preferences, and management based on network enforced policies. In this article, we focus on the solutions that current Operating Systems (OSs) and terminals provide for managing their different connectivity options.

Many network providers offer their own network manager solutions, for seamless integration with their services. These solutions are not available across all existing Operating Systems, and often focus on the largest user base. Network managing functions composing these solutions usually aim at giving users the best possible access to the provider's network solutions, sometimes using proprietary services for hotspot location or traffic management. Nowadays it is typical to find mobile terminal network applications pre-installed by laptop manufacturers, provided by mobile operators as integrated software in USB dongles or even as native applications from Operating Systems. In the following sections, we illustrate some examples.

A. O2 Connection Manager

The O2 Connection Manager[1] is a Windows application for managing Internet connections. It attempts to provide connection to the fastest available networks, including the user's home broadband. It suggests connection to Wi-Fi hotspots from the

[1]O2 Connection Manager, http://www.o2.co.uk/support/broadbandinternet/o2connectionmanager

operator, and integrates with the operator's cellular dongle hardware supporting SMS services in the desktop application.

B. AT&T Communication Manager

The AT&T Communication Manager[2] is a desktop application for taking advantage of the operator's 4G network in the United States. It works in Windows and Mac OS, and also manages Wi-Fi connections, handling the authentication to AT&T Wi-Fi hotspots. The application provides real time data usage management, and allows the creation of mobile hotspots for sharing internet access with multiple Wi-Fi enabled mobile devices. In addition, the application uses the operator's cellular network, or integrated GPS chips on proprietary devices, to provide location services, namely for locating Wi-Fi hotspots.

C. GNU/Linux NetworkManager

The GNU/Linux NetworkManager[3] (NM) application is present in most, if not all, desktop GNU/Linux distributions. It aims to provide constant network connectivity, featuring IPv4 and IPv6 support, WEP, WPA/WPA2 and 802.1X security mechanisms, and the ability to control many network devices including Ethernet, Wi-Fi, WiMAX and 3G modems.

NetworkManager provides a D-Bus interface for Desktop Environment and GUI configuration interfaces. Users provide configurations for each network they want to include, and NetworkManager tries its best to keep the user connected, preferring secure connections first, then selecting Access Points with stronger signals. Moreover, it always attempts to acquire a connection with every available network interface, unless the user explicitly requests disconnection of an interface.

D. InterDigital Smart Access Manager

InterDigital's Smart Access Manager[4] (SAM) is a client for mobile devices that makes network selection and traffic management decisions between Wi-Fi, 3G and LTE networks. This is a solution for network providers that aims to provide terminal functions for Wi-Fi offloading and maintaining user Quality of Experience (QoE) on the provider networks via the Access Network Discovery and Selection Function (ANDSF). The client supports dynamic selection between hotspots, giving preference to user-configured Wi-Fi networks. It performs IP traffic routing between networks when both Wi-Fi and cellular access are available, thus enforcing provider policies on terminal devices.

E. Solution Comparison

The previous sections do not attempt at an extensive overview of the available solutions. Solutions from service providers often fail to address multiple network technologies, since they focus mostly on their provided services, and usually Wi-Fi for offloading [2]. Generic solutions, on the other hand, do not provide much information or special features regarding interface and network selection. Of the mentioned solutions, only SAM uses a mechanism for requesting help from the network for attachment decisions. Table I provides a comparison of the various features for each solution.

SAM is considered a Multi-OS solution since it supports various smartphone devices, but it is not intended for desktop usage. Although also listed as non-Multi-Operator, it probably depends on the configuration that each operator requires for the deployment of the software.

	O2	AT&T	NM	SAM
Multi-OS	no	yes	yes	yes
Multi-Technology	yes	yes	yes	yes
Multi-Operator	no	no	yes	no
Dynamic Optimizations	no	no	no	yes
Network decisions	no	no	no	yes
Open Source	no	no	yes	no

Table I
SOLUTION FEATURES COMPARISON.

III. SUPPORTIVE TECHNOLOGIES

Considering the previous shortcomings presented by the network manager solutions illustrated in Section II, we introduce in this section a set of supporting mechanisms which, associated to those network managers, would complement them with the lacking behaviour.

A. IEEE 802.21

The IEEE 802.21 [1] standard enables seamless handover between heterogeneous technologies. This framework is based on a protocol stack implemented in all the devices involved in the handover. It exposes a common L2 interface for IEEE 802 and 3GPP technologies in order to facilitate network handovers, thus a Media Independent Handover (MIH) framework. It considers several network components:

- **Mobile Node (MN):** This is the network attachment candidate itself.
- **Point of Attachment (PoA):** The network endpoint for L2 connection to the MN.
- **Point of Service (PoS):** A network entity that provides information to the MN and takes network configuration actions in order to optimize MN handovers.
- **Non-PoS:** An IEEE 802.21 entity that contains static information regarding network policies and topologies.

These entities are interconnected by an MIH Function (MIHF) through the MIH Network Service Access Point (SAP) interface. It also handles the communication exchanges between the lower layer entities, through the MIH Link SAP and higher layer entities (MIH Users), through the MIH SAP. The MIH SAP provides MIH Users with a large set of technology-independent primitives for interface control and information gathering. The MIH Link SAP provides a mapping between the 802.21 primitives and the technology-specific lower layer interfaces. The MIH primitives are grouped in the Event, Command and Information Services.

[2]AT&T Communication Manager, http://www.wireless.att.com/businesscenter/solutions/wireless-laptop/communication-manager/index.jsp

[3]GNU/Linux Network Manager, http://projects.gnome.org/NetworkManager/

[4]InterDigital Smart Access Manager, http://www.interdigital.com/smart-access-manager

1) Event Service: Events can be initiated either by the MN or by the network. They originate from lower layers or MIHF, at the MN, at the network PoA or the PoS, and are propagated to interested MIH Users, local or remote. MIH Users declare their interest in certain events by subscribing them. MIH events pertain to the following factors:

- **Link State Change events:** This includes events to notify of link layer occurrences such as the loss or establishment of L2 connectivity.
- **Link Parameter events:** These events are generated in response to configured thresholds pertaining to link-layer parameters such as packet loss, signal strength, etc.
- **Predictive events:** Such events convey the likelihood of a change in the link conditions in the near future based on past and present conditions, such as the decay in signal strength in relation to a PoA.
- **Link Transmission events:** These events can be subscribed to receive indication of the link layer transmission status of individual upper layer PDUs.

Events are mostly advisory in nature, which means Users subscribing to a set of events are not required to act on them.

2) Command Service: MIH Users utilize command services to determine the status of links and/or control the device for optimal performance. Commands can be delivered locally or remotely, and are classified in two main categories: MIH Commands and Link Commands.

Link commands are delivered from the MIHF to the Link SAPs, on behalf of the MIH Users, for various control operations, through the following primitives:

- **Capability discovery and event subscription:** A link may be queried in order to determine its supported primitives both for the command and event service. Link events must be subscribed for receiving notifications on a per-link basis.
- **Parameter retrieval:** Various active link parameters can be obtained through Link commands, such as the current bit rate, Quality of Service (QoS) statistics, signal strength, etc.
- **Threshold configuration:** Link parameter events can be configured for future reception over the event service. Thresholds may be configured for obtaining periodic reports or only indications that a parameter value has been crossed.
- **Link control:** A single primitive offers various actions to request a link to perform, including radio scans, changing its operational state or going into power save mode.

MIH Commands are sent by the higher layers to the MIHF. They may originate locally, through the MIH SAP, or remotely, through the MIH Network SAP. MIH Commands may translate into specific Link Commands operations, or aid in handover decision procedures by the following requests:

- **Handover candidate query:** Both the MN and the Network may issue this command in order to exchange suggested networks and associated points of attachment information for possible handovers.
- **Query resources:** This command is used to assess or prepare network resources in a target network for MN handover.
- **Handover commit:** For MN controlled handovers, this is used to inform the serving network of the target decision. For Network controlled handovers, this command informs the MN of which network to attach to.
- **Handover completion:** This command allows both serving and target networks to indicate the completion status of a handover operation.

3) Information Service: The Information service is a collection of mostly static information elements about networks and operators. These elements provide information essential to the network selection algorithm to make a successful handover across heterogeneous networks and technologies. A Mobile Node benefits mostly from this service by being able to query geographically surrounding networks, as well as their service capabilities before making a handover decision or even powering other radio interfaces. For example, information about a nearby Wi-Fi hotspot could be obtained using a 3G interface without the need to power the Wi-Fi radio. It may also be used to query target network information regarding security or QoS mechanisms, thus influencing the target selection.

4) Media Specific Mappings for SAPs: The MIHF aggregates disparate interfaces with respective media dependent lower layer instances into a single abstract interface for MIH Users, reducing the inter-media differences to the extent possible. For the most part, existing primitives and functionality provided by different access technology standards are used. Amendments to existing standards are proposed when deemed necessary to fulfil the MIHF capabilities.

The Link Service Access Point (LSAP), defined in the IEEE 802.2 [3] standard, provides the interface between the MIHF and the Logical Link Control (LLC) sublayer across both IEEE 802.3 and 802.11 networks. However, the complete MIH_LINK_SAP set of primitives mappings for 802.11 requires an enhancement proposal defined in IEEE 802.11u [4], in the form of the MAC State Generic Convergence Function (MSGCF).

The interface for 802.16 networks depends on the C_SAP and M_SAP, both defined in IEEE 802.16g [5].

A special SAP, MIH_3GLINK_SAP, is defined for interfacing with the MIHF and the different protocol elements of the 3G system.

B. Converging decision processes

With different kinds of access technologies available to multi-interface mobile terminals, achieving an optimal handover decision depends on multiple parameters [6], ranging from:

1) The dynamics of the wireless strata (e.g., signal-noise ratio, available bandwidth, cell load);
2) Requirements placed by the service content being accessed (e.g., minimum latency);
3) Requirements placed by the user (e.g., perceived video and/or audio quality, cost);

4) Network conditions (e.g., cell load, requested service, policies).

The different criteria involved must not only take into consideration the capabilities of the service being provided, but also the resources available in the network and, ultimately, the user satisfaction. As such, optimal handover decisions have the need to assess different objectives, from different layers of the network stack, in order to achieve an Always Best Connected [7] solution.

In this sense, different schema are possible, varying between mobile terminal centric decisions [8], and network controlled decisions [9], or even combinations of both where the perspective of the terminal and network come together to optimize the handover decision to a broader set of requirements [10].

However, a common requirement for optimized handover processes relies on the flexibility and simplicity of network manager application design, contributing for the facilitated deployment of such mechanisms in different kinds of mobile terminals using a broad spectrum of access technologies. As such, in this work, we consider the integration of Media Independent mechanisms within the fabric of network manager applications, aiming at mainly two things: abstracting information and control capabilities from different kinds of access interfaces; and empowering such applications with the means to disseminate that information and control with local decision entities, as well as network-controlled remote entities when available. This constitutes the setting for our Enhanced Media Independent COnnection Manager framework, EMICOM, presented next.

IV. Framework

The developed framework is based on an open source MIHF implementation, ODTONE[5], supporting the whole range of MIH services. It provides a set of Application Programmable Interfaces (APIs), enabling the development of custom MIH Users and Links to interface with the provided MIHF. These APIs are based on socket message transport, which enables the reuse of the solutions between multiple Operating Systems.

The mechanisms presented by the IEEE 802.21 standard can be considered as an abstraction for Media Independent interface management. Not only this common interface provides easier interface control, it also exposes common information enabling network managers to employ a plethora of new network selection algorithms based on application and service requirements, power constraints, and other requirements. However, the 802.21 services do not provide all the necessary primitives for interface management. For example, the procedures for network association are outside of the scope of the standard. Also, since it aims to provide an abstraction only at the link layer, there is no support for network layer configuration.

This section introduces our network manager framework by detailing how such concepts can be integrated with 802.21, as well as identifying key enhancements done over the base standard, enhancing its functionality.

[5]ODTONE, Open Dot Twenty ONE, http://atnog.av.it.pt/odtone

A. MIHF Cross-layer

The open-source nature of the ODTONE software collection allowed an extension to the 802.21 protocol in order to keep the advantages of interface agnostic control, and thus not requiring specific hardware or OS support for operation. The proposed extension refers to operations within the Mobile Node only.

Association and security procedures require link layer support from supplicant software. IP configuration requires assigning IP addresses to interfaces, as well as default gateways, custom routes, and DNS servers. As such, for these tasks, two additional Command Service primitives were integrated into the 802.21 mechanics, for both the MIH SAP and MIH Link SAP, providing the following functionality:

- **Link configuration:** Attach to a given network, providing the necessary authentication, association and security information.
- **L3 configuration:** Configure a set of networking properties on an interface, such as IP address, static routes and list of DNS servers.

Network authentication protocols such as 802.1X [11] provide many authentication mechanisms, some requiring an indefinite number of exchanges between the supplicant and the authentication server (i.e., depending on deployment scenarios), and possibly demanding user input. This can be interpreted as a network request for the user, and can be implemented through the 802.21 Event Service. The following primitive was added to the 802.21 mechanics:

- **Link configuration required:** Indicates a network request for authentication material from the supplicant.

Using these commands, MIH Users are abstracted from specific protocols such as DHCP or stateless auto-configuration. An MIH User requesting Layer 3 configuration may request attribution of specific IPv4 or IPv6 addresses, but it may as well ask the framework to attempt DHCP configuration, for example. The same is true for the Link configuration primitive; the required parameters for the specific network are provided immediately in the Link configuration request. The framework then implements the architecture of Figure 1.

In this sense, our Network Manager assumes the role of MIH User, interfacing with an enhanced MIHF implemented over the ODTONE open-source software, thus empowering it to not only access different kinds of link layer technologies in a media-independent way, but also to abstract security association and address requesting processes.

B. Link Interfaces Service Access Points

Supporting this framework requires implementing Link SAPs for various network device technologies. These components require direct communication with the OS software or interface drivers, whose implementation is platform dependent. For the specific case of GNU/Linux, two Link SAP implementations were developed, and added as open-source software to the ODTONE project, for the IEEE 802.3 and 802.11 technologies.

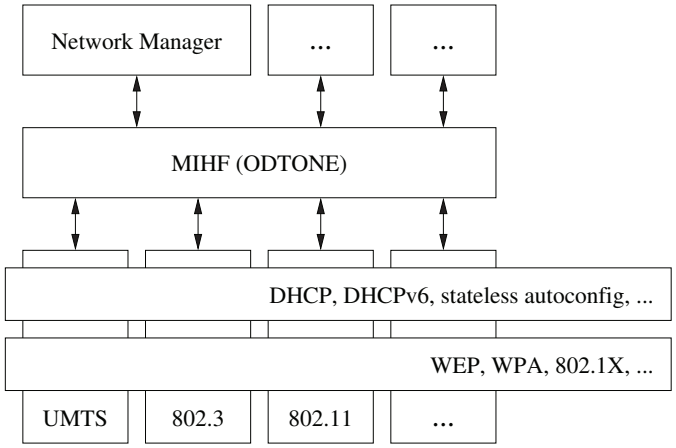


Figure 1. EMICOM framework architecture.

The Linux kernel offers two main interfaces for network device control, both over the Netlink [12] protocol: Route Netlink and the *nl80211* family over the Generic Netlink protocol. These interfaces provide some simple link control functionalities exposed by the Linux kernel, which were exploited in our work to produce the LINK SAPs. The Route Netlink interface enables access to the networking subsystem of the Linux kernel by providing a set of primitives for adding, getting and removing information from its internal table-oriented data structures. For security mechanisms support, and dynamic IP configuration, external tools are also used.

1) Route Netlink: This interface provides the full mapping for 802.3, and a partial mapping for the 802.11 Link SAP, according to 802.21 primitives and parameters. It offers three groups of messages of particular interest, for manipulation of different aspects common to all Linux network interfaces:

- **LINK messages:** Allow a user of the Route Netlink protocol to manipulate information about network interfaces on the system. These messages contain attributes for indicating link state and device power changes.
- **ADDR messages:** Used for manipulation of IP addresses of the network interfaces. These messages support IPv4 and IPv6 addresses, and an interface can be assigned multiple IP addresses.
- **ROUTE messages:** These messages baptise the whole protocol name and, as it points out, they refer to IP routing table management.

These messages are used to get or modify attributes of a link, but can also be subscribed in order to receive notifications when the kernel signals changes to these objects.

2) nl80211: This protocol offers an interface for the Linux *mac80211* framework, for Wi-Fi specific operations. Among others, the following mechanisms are exposed:

- **Power control:** In addition to the various interface states offered by the Route Netlink protocol, the nl80211 interface allows controlling a Wi-Fi interface's power save mode for sleeping between Access Point (AP) beacons, as defined in the 802.11 standard.
- **Scanning:** The interface supports triggering scans at any moment as well as the configuration of scheduled scans at regular intervals. The scan result provides a listing of the detected APs containing basic BSS attributes, as well as the full listing of the beacon Information Elements.
- **Association and Authentication:** Linux supports various 802.11 authentication methods. For security mechanisms, there are facilities for userspace applications to transmit raw packets as well as subscribing received packets by means of matching the first few bytes of the desired frames.
- **QoS:** The *mac80211* framework supports the 802.11e [13] extension, and will assign frames to specific hardware queues based on the Type of Service (TOS) field of the packet IPv4 header. IPv6 does not support the TOS field and uses Traffic Class codes instead. Unfortunately, current versions of the Linux kernel do not provide statistics such as minimum, average and maximum packet delay for individual Classes of Service, which is one of the features of the 802.21 framework.

Similarly to the Route Netlink interface, *nl80211* allows the subscription of certain occurrences. This includes events for L2 connection and disconnection, but also the signalling of new scan results and the crossing of a given signal strength threshold.

For easier Netlink message parsing and program memory management, both the Route Netlink and *nl80211* access are mediated by a custom C++ wrapper on top of the *libnl*[6] library.

3) Security Mechanisms: Support for network authentication mechanisms such as WPA2 and 802.1X is achieved with the *wpa_supplicant*[7] software. *wpa_supplicant* provides a text-based socket interface as well as a D-Bus API. The Link SAPs interface with *wpa_supplicant* via the D-Bus API, which supports adding network configurations and keying material at runtime. Requesting authentication to a network can be performed immediately or later. If the authentication mechanism requires further parameters, the daemon signals the occurrence of network requests indicating the required fields.

4) IP Configuration: Dynamic IP configuration is supported via the Internet Systems Consortium's DHCP[8] software bundle. It provides a DHCP client, *dhclient*, supporting DHCPv4 and DHCPv6, as well as IPv6 Stateless Address Autoconfiguration. Unfortunately, this program does not offer any sort of API, and its integration is done via scripted calls to the command line daemon, providing the appropriate parameters as command line options each time. DNS servers are configured by direct manipulation of system configuration files.

C. Network Manager

The aim of the Network Manager development was to replace the existing NetworkManager software which is inte-

[6]libnl libraries, http://www.infradead.org/~tgr/libnl/
[7]WPA Supplicant daemon, http://hostap.epitest.fi/wpa_supplicant/
[8]DHCP Client, https://www.isc.org/software/dhcp

grated with most desktop environments via network configuration GUIs and panel applets via a freedesktop.org[9] endorsed D-Bus API. This API includes the following interfaces for Desktop applications:

- **NetworkManager:** A main central object that offers an interface for overall management tasks. This object holds the network device structure, the overall machine state and individual connection states.
- **Device:** This is an abstract interface to represent common attributes of all network devices. An object implementing this interface is associated to a unique network device in the system. Each technology extends this interface for providing technology-specific features. The **Device.Wireless** interface, for example, provides a D-Bus API for requesting scans, list of known APs, etc.
- **AccessPoint:** An AccessPoint object exposes properties to identify networks and determine how to properly prepare the association to the BSS, such as the SSID, Frequency, Maximum bitrate, etc.
- **Settings:** The Settings interface may be regarded as the system-global repository for the various configured networks. It exposes methods to add and retrieve configuration objects, containing the necessary information for each network, including authentication material, IP configurations (whether dynamic or static), etc.

The services exposed by the NetworkManager D-Bus interface, otherwise implemented directly for each specific hardware implementation, must be converted to the MIH Services primitives. In essence, regardless of the Device type exposed by the D-Bus API, the underlying object for interfacing through the MIHF is the same for every type. This base object uses a reduced set of primitives for media independent device control:

- **Link actions:** Used to initiate scanning, disconnecting a link, and setting the device power state between on, off and power save mode.
- **Link configure thresholds:** Used to configure threshold parameters for future event notifications.
- **Link get parameters:** Retrieve certain parameters of the currently established link.
- **Link configuration:** Used to request association and setup of security mechanisms to a network.
- **L3 configuration:** Used to request IP configuration on a link.

Similarly, reaction to device events is supported via the set of primitives of the MIH Event service:

- **Link up:** L2 connection is established on a link.
- **Link down:** L2 connection is lost on a link.
- **Link detected:** A PoA of a new network was detected following a scan.
- **Link parameters report:** Events generated regarding configured thresholds.
- **Link configuration required:** Additional procedures are required for security mechanisms.

[9]freedesktop.org, http://www.freedesktop.org/

The MIH User, and the D-Bus interface, should be extended with further intelligence and algorithms for network selection decisions, depending on the target environment.

V. Evaluation

The first aspect that needs to be evaluated from this abstract access to the interfaces is the framework footprint in a system. Since it is a multi process solution that relies on Inter Process Communication (IPC) mechanisms, there is an obvious overhead in communication. It is also relevant to compare other aspects such as the amount of code that composes the framework, in comparison with media dependent solutions, as well as the memory consumption and other aspects such as battery drainage, which are important in mobile devices. The following sections provide a comparison with the existing GNU/Linux NetworkManager application, as well as view of various different scenarios that distinguish the EMICOM framework from other solutions.

A. Test Setup

The tests were run in a laptop computer with the specifications defined in Table II. The testbed for network experiments contains two wireless Linksys WRT54G Access Points with the *DD-WRT*[10] firmware, connected by Ethernet to a video server machine, as depicted in Figure 2.

Component	Value
Operating System	Archlinux
Kernel version	3.5.4
Processor	Intel Core i7 M620 ($2 \times 2.67GHz$)
Memory	$4GB$ at $1333MHz$
Ethernet card	Intel PRO/1000 CT
Wi-Fi card #1	Intel Centrino Advanced-N 6200
Wi-Fi card #2	ASUSTeK WL-167g

Table II
Computer attributes.

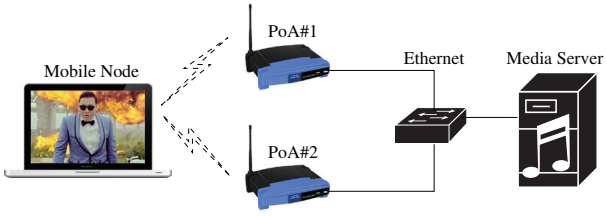


Figure 2. Testbed architecture.

B. Inter process overhead

In practice, the proposed solutions adds a layer to the existing kernel interfaces for network management, which abstracts the media dependent control. Communicating through this layer introduces an overhead that causes delays between operations, and implies data transmission between processes, at a cost.

[10]DD-WRT firmware, http://www.dd-wrt.com/

The 802.21 standard defines data transmission between remote entities via transport protocols such as UDP and TCP, encoding the necessary information in the Type-Length-Value (TLV) format. ODTONE uses this format for local transmission over transport sockets as well, allowing it to achieve the ability of being OS-independent, but at an obvious cost. Communication between the higher and lower layers require transmitting from the MIH User to the MIHF, then from the MIHF to the Links; answers traverse the inverse path. Messages from the event service travel only from the Link SAP to the MIH User, via the MIHF as well. Communication with remote entities is a necessary overhead and does not account for IPC analysis. Table III shows the number of MIH messages exchanged for various situations, including the total payload of transmitted data.

Operation	# of Messages	Total payload (bytes)
Power DOWN	4	162
Power UP	4	164
L2 Connect (simple)	4	197
L2 Connect (WPA + EAP)	4	508
L3 Configure (DHCP)	4	193
Disconnect	4	162
Link Detected event	2	144
Link Down event	2	80
Link Up event	2	102

Table III
MIH MESSAGE SIZES.

From these values it is visible that Link events take the fewest amount of bytes required (between 80 and 144 bytes). Link commands demand more information (between 162 and 197 bytes), but it is the new more complex commands, where the DHCP and security association and capabilities have been added to the standard 802.21 behaviour, that require the most amount of information (between 193 and 508 bytes, respectively).

The delay for the transmission of these messages is not analyzed, since the performance is internal and highly dependent on the load and capacity of each system. However, [14] shows that, for message sizes of up to 512 bytes, a low end (by today's standards) Linux machine delivers a rate of over 70 000 messages per second, translating to just $14\mu s$ per message. Furthermore, [15] shows a great performance improvement in transfer rates for Linux IPC by using UNIX domain sockets instead of UDP. ODTONE does not support UNIX sockets yet but, given its open source nature and the message-oriented IPC mechanism, the support should be implemented with little effort.

C. Code base

When compared to a native solution, the extra layer for media abstraction requires a greater amount of code for translating operations. However, the higher layers require less code for controlling individual interfaces, since the procedures are reused across device technologies.

Table IV offers a direct comparison between the provided framework and the Linux NetworkManager (version 0.9.6.0) solution, using the *SLOCCount*[11] tool. Despite being developed in different programming languages (C vs. C++), the number of code lines is nonetheless an acceptable measure of development effort. It should be noted that not the entire codebase of NetworkManager is considered; the counting excludes NetworkManager components not yet included in the EMICOM framework such as automatic VPN setup, Bluetooth and WiMAX support, etc.

NetworkManager	EMICOM		
Total: 70 815	MIHF (ODTONE)	11 606	Total: 21 960
	MIH User	5 277	
	802.11 Link SAP	1 610	
	802.3 Link SAP	897	
	libnl wrapper	1 347	
	dhclient wrapper	77	
	wpa_supplicant wrapper	1 146	

Table IV
CODE BASE COMPARISON, IN NUMBER OF SOURCE CODE LINES.

It is clear that the whole EMICOM framework, providing the same features as the considered for the NetworkManager software, requires less than a third of the source code. Several reasons contribute to this fact, the most prominent being the different programming languages. Other factors include the used libraries. ODTONE, and EMICOM, are highly dependent on the *boost* libraries[12] for data manipulation, which could also be a relevant contribution to the decrease in code size.

D. Memory usage

Process memory usage is a common limiting factor in some deployment scenarios. Embedded systems usually have limited memory. Even in desktop computers, it is desirable that resident applications account for a small impact on the overall system capacity.

Measuring process memory usage in modern Operating Systems is a complex task. Processes commonly make use of system libraries that, once loaded into memory, can be reused several times by several processes, and thus the system does not allocate multiple instances of the library. These libraries can be considered components of a program, but the program may not be the sole responsible for loading the library into the memory.

The *Valgrind*[13] utilities allow developers to track memory allocations of individual processes. This utility can accurately report the memory that each process allocates both in the Heap and Stack memory segments. Table V shows the size of a snapshot of the combined Heap and Stack memory allocated by each solution, captured after an initial launch, after the attachment to both an 802.3 and 802.11 network (for this specific test, the computer is also attached by Ethernet, not represented in Figure 2).

Again, comparing similar situations for both solutions, the EMICOM framework shows a great benefit, compared to

[11]SLOCCount tool, http://www.dwheeler.com/sloccount/
[12]Boost Libraries, http://www.boost.org
[13]Valgrind utilities, http://valgrind.org/

NetworkManager	EMICOM		
Total: 967 064	MIHF (ODTONE)	35 688	Total: 553 496
	MIH User	401 192	
	802.11 Link SAP	52 200	
	802.3 Link SAP	64 416	

Table V
MEMORY USAGE BY EACH SOLUTION, IN BYTES.

the NetworkManager software. NetworkManager is openly developed, and has existed for a long time. Apart from the base ODTONE library, the EMICOM software has not been reviewed by external developers, and has not been submitted to optimization procedures, which means there could still be improvements in this area. It should be noted that the *boost* libraries do not directly contribute to this factor, since they are mostly header-only, thus not loaded as shared system libraries.

E. Benefits

The real world benefit for this framework is the plethora of scenarios where it may be considered for network selection and handover optimization. One of the main aspects that benefit a system with an 802.21-based networking solution is the Information Service, that will allow obtaining information about neighboring networks without powering additional radios. This service also allows the exchange of many network configurations and policies that will allow decision algorithms to take into account variables such as the cost for each network, the throughput or delay requirements for each application, and much more.

1) Battery life: This is increasingly relevant, as more and more mobile devices provide multiple radios for multiple network technologies. Figure 3 shows two different test runs, where a single laptop computer is retrieving a video stream via a Wi-Fi interface at a fixed rate of $500KB/s$. In one test run, the laptop is running the GNU/Linux NetworkManager, and the second is using the EMICOM framework. Both have one Ethernet interface and two Wi-Fi interfaces. This need not be the case, as there should be little benefit in having two similar radio interfaces, but it serves to compare the impact of having more than one wireless device in the same system. Moreover, due to EMICOM's usage of 802.21 abstraction mechanisms, the same events and commands used in this scenario would still be valid in scenarios featuring other technologies such as 802.16 and 3GPP links.

A regular NetworkManager typically employs very basic and static connectivity strategies, which implies having all the devices active at all times. In this specific scenario, we enhanced the base behaviour by allowing the second device to not be active at all times, but instead waking up at regular intervals of 30 seconds (halving in frequency every half hour) and performing a scan. A 802.21 solution, however, does not need to power the secondary device for learning about neighbouring networks, because it can rely on the Information service for the task of finding neighbouring networks, so in this scenario the MIH User always keeps the second device off, only activating it when an optimized handover opportunity occurs.

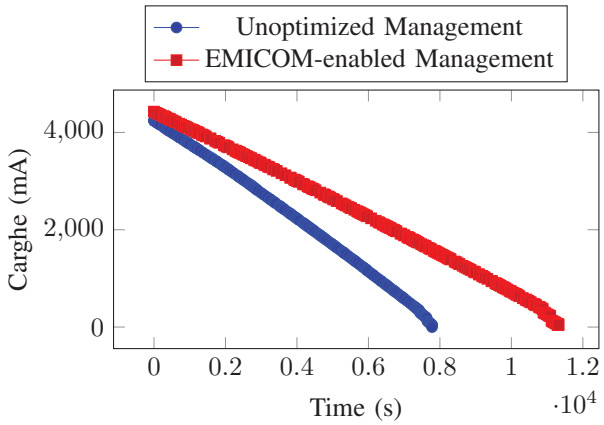


Figure 3. Battery drain comparison.

In the collected results, it is clear that the EMICOM run starts with higher initial capacity. This is common, as batteries often report different maximum capacity values at each charge cycle. The impact of powering an additional interface and performing scans is nonetheless noticeable and very significant, reducing the total autonomy of the device by a factor of 30%. Interestingly, though, the chart does not portray the increase in scan intervals after each 30 minute period. The initial interval of 30 seconds is perhaps a low initial value, resulting in maximum scan interval of 8 minutes in the experiment, although this would be an acceptable value for connectivity in a high mobility scenario.

2) Optimal selection: Figure 4 provides a 60 second test scenario where the EMICOM tool benefits from information from an MIH User in the Network, assisting with the handover decision. In this specific test, two Wi-Fi APs offer access to the same network. The computer is trying to maximize its TCP throughput by retrieving an on-demand video from the server. One AP has a stronger signal than the other ($-23dBm$ versus $-39dBm$), but it offers a lower throughput. This could be because the stronger AP is serving a greater amount of users, or has a low downlink, or many other reasons. In the specific test case, the rate was throttled at the AP on purpose. The GNU/Linux NetworkManager always stays connected to the strongest AP, since it only bases its connectivity decision on signal level. EMICOM however, using 802.21, receives information pushed by the network (e.g., via an *MIH Net Handover Commit request* command) to the user after 30 seconds, suggesting a handover to the other AP. This enables better load balancing on the network, while directly benefiting the user service.

A period with total loss of connectivity is noticeable. This is the occurrence of the hard handover, since the integration with IP mobility management engines achieving seamless mobility are out of scope from this paper. However, EMICOM, is able to leverage from mobility management primitives provided by 802.21, when such IP mobility schemes are employed.

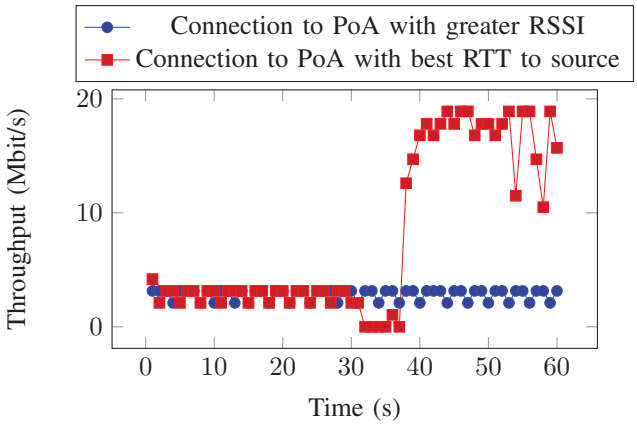


Figure 4. Optimal AP selection for throughput.

Nonetheless, via the stimuli provided by our 802.21-enabled Network Manager, the EMICOM framework was able to enhance the throughput of the video reception by performing a link switch to an AP with better downlink connectivity.

VI. Conclusion

The increase in connectivity opportunities, in terms of access technologies, to reach for different kinds of on-line content, places stringent requirements in terms of optimized connectivity and interface selection. Network Manager applications are becoming standard in different Operating Systems, but lacks the flexible capabilities for abstract access technology interfacing as well as disseminating and receiving handover optimization information from other sources, both local to the node or remotely available in network controlling entities.

This paper overthrew those shortcomings by integrating the Media Independent mechanisms of the IEEE 802.21 standard into Network Manager application concepts, defining EMICOM, an Enhanced Media Independent COnnection Manager framework.

This framework, more than just integrating 802.21 with Network Manager functionality, extended the first with the support for security association and address negotiation procedures, which are invaluable in today's seamless connectivity procedures in mobile networks. Moreover, it contributed to the open-source community by making available such extensions over ODTONE, an open-source implementation of the IEEE 802.21 standard. To that end, a set of Link Service Access Points, enabling ODTONE to interact with 802.3 and 802.11 interfaces in the GNU/Linux operating system, were also contributed to the project. Indeed, the 802.11 Link Service Access Point has been adopted by the FP7 ICT MEDIEVAL project[14], increasing the derived project with media independent control to Wi-Fi interfaces, as well as providing a complex testing environment for the work presented here.

The implementation of EMICOM also allowed the realization of an extensive evaluation effort, providing insight on the benefits of using Media Independent information and control capabilities to assist optimized handover and interface selection. Obtained results were compared to a popular NetworkManager from the GNU/Linux Operating System, showing a reduced code base, better battery consumption and allowing optimized handover procedures for opportunistic network attachment.

As future work, the evaluation of the proposed framework does not attempt to provide a performance analysis on handover operations, or even provide a full featured solution for the network management problem. Currently, these mechanisms are being evaluated under the scope of the MEDIEVAL project, where much other information can be used by a Network Manager MIH User in order to achieve the best connectivity, depending on user policies.

[14]MEDIEVAL, http://www.ict-medieval.eu/

Acknowledgment

This work has been partially funded by the European Community's Seventh Framework Programme (FP7-ICT-2009-5) under grant agreement n. 258053 (MEDIEVAL project).

References

[1] IEEE Standard for Local and Metropolitan Area Networks- Part 21: Media Independent Handover. *IEEE Std 802.21-2008*, 2009.

[2] de la Oliva, A. and Bernardos, C.J. and Calderon, M. and Melia, T. and Zuniga, J.C. IP flow mobility: smart traffic offload for future wireless networks. *Communications Magazine, IEEE*, 49(10):124 –132, oct. 2011.

[3] Logical Link Control. *ANSI/IEEE Std 802.2-1985*, 1984.

[4] IEEE Draft Std 802.11u WLAN MAC and PHY Amendment 7: Interworking with External Networks. *IEEE Unapproved Draft Std P802.11u/D5.0, Feb 2009*, 2009.

[5] IEEE 802 Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 3: Management PLANe Procedure and Services. *IEEE Std 802.16g 2007 (Amendment to IEEE Std 802.16-2004)*, 2007.

[6] Abid, M. and Yahiya, T.A. and Pujolle, G. A Utility-based Handover Decision Scheme for Heterogeneous Wireless Networks. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, volume , pages 650 –654, jan. 2012.

[7] Gustafsson, E. and Jonsson, A. Always best connected. *Wireless Communications, IEEE*, 10(1): 49 – 55, feb. 2003.

[8] Kassar, M. and Achour, A. and Kervella, B. A mobile-controlled handover management scheme in a loosely-coupled 3G-WLAN interworking architecture. In *Wireless Days, 2008. WD '08. 1st IFIP*, volume , pages 1 –5, nov. 2008.

[9] Bertin, P. and Guillouard, K. and Rault, J.-C. IP based network controlled handover management in WLAN access networks. In *Communications, 2004 IEEE International Conference on*, volume 7, pages 3906 – 3910 Vol.7, june 2004.

[10] Corujo, D. and Guimaraes, C. and Santos, B. and Aguiar, R.L. Using an open-source IEEE 802.21 implementation for network-based localized mobility management. *Communications Magazine, IEEE*, 49(9):114 – 123, september 2011.

[11] IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control. *IEEE Std 802.1X-2010 (Revision of IEEE Std 802.1X-2004)*, 2010.

[12] J. Salim, H. Khosravi, A. Kleen, and A. Kuznetsov. Linux Netlink as an IP Services Protocol. RFC 3549 (Informational), July 2003.

[13] IEEE 802 Part 11: WLAN MAC and PHY Amendment 8: MAC QoS Enhancements. *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, 2005.

[14] Brian F. G. Bidulock. STREAMS vs. Sockets Performance Comparison for UDP. *OpenSS7*, 2007. Accessed Oct. 2012, http://www.openss7.org/papers/strinet/testresults.pdf.

[15] Kwame Wright and Kartik Gopalan and Hui Kang. Performance Analysis of Various Mechanisms for Inter-process Communication, 2007. Accessed Oct. 2012, http://osnet.cs.binghamton.edu/publications/TR-20070820.pdf.

Analysis of the Logical Proximity between 802.11 Access Points

Ricardo Sousa
INESC TEC
Campus da FEUP
4200 - 465 Porto
Portugal
rmps@inescporto.pt

Ricardo Morla
INESC TEC
Campus da FEUP
4200 - 465 Porto
Portugal
ricardo.morla@inescporto.pt

Abel Maio
FEUP
Campus da FEUP
4200 - 465 Porto Porto
Portugal
ei07102@fe.up.pt

João Coelho
FEUP
Campus da FEUP
42 - 465 Porto
Portugal
ei07118@fe.up.pt

***Abstract*—802.11 campus networks have their access points deployed across the area the network has to cover. The physical proximity between these access points is often well understood. Network managers have maps of the campus and of the location of the access points. Due to reflections indoor and between buildings, a mobile station may have connectivity to only one of two physically nearby access points. This may mislead the network manager into thinking that an area is densely covered when in fact mobile stations cannot connect to all of the access points in that area. This problem can have a dynamic nature if we consider people and objects that move around and change the properties of the propagation medium. In this paper we explore an alternative measure to physical proximity based on mobile station connectivity to the access points, which we call logical proximity. We take the ping-pong effect as a proxy to logical proximity. We use this proximity to characterize 802.11 campus network with over 200 access points and 14k users over a 2 year period. We report on the magnitude of the ping pong effect, the clustering of access points, and the degree distribution of the resulting access point proximity network.**

***Index Terms*—Local area networks.**

I. Introduction

The deployment of wireless networks technology has attracted attention to the issues that network managers face. One of such issues is to adequately characterize the logical proximity of access points. This enables the network managers to view the network from the perspective of the user and of possible access points that mobile stations can connect to. In this work we aim to identify the logical proximity of the access points using the ping pong effect as a proxy for such proximity. The ping pong effect can be characterized by a series of consecutive connections to different access points and is the result of the aggressive nature of 802.11 interfaces that try to connect to an access point with a better signal once the signal from the current access point drops below a threshold.

In the past, research has been developed to understand network topology. The authors in [1] propose a mobility-aware clustering algorithm that uses roaming events as the metric to evaluate the proximity to access-points (APs) without using any geographical information. In this work, they make reference to three main goals for studying the mobility of the users on wireless networks: (1) to understand what are the implications and how mobility can have an impact on the network services, (2) to create realistic mobility models to evaluate the performance of protocols and algorithms, and (3) to propose new communication solutions that can adapt to the specificities of each user.

Another investigation that approach this subject is project SPOTS [2]. In this project, the aim is to create a better understanding of the daily working and living patterns of the MIT academic community, which changes due to the emergence of WiFi itself. Tang and Baker in the paper [3] analyze the network for overall user behavior (when and how intensively people use the network and how much they move around), overall network traffic and load characteristics (observed throughput and symmetry of incoming and outgoing traffic), and traffic characteristics from a user point of view (observing a mix of applications and number of hosts connected to by users).

[4] analyzes the Wi-Fi network as a proxy to space usage aiming to use it as a mean for the characterization of physical spaces and, consequently, as a source of information for a dynamic symbolic model representing those spaces. In [5] a general methodology is presented for extracting mobility information from wireless network traces, and for classifying mobile users and APs. Kim and Kotz [6] propose a model of user movements between APs. In their paper they define three goals in developing a mobility model. First, the model should reflect real user movements, second, the model should be general enough to describe the movements of every device and third, the model should consider the hourly variations over a day. Another area with many interesting works is network tomography, originating from a research by Vardi [7]. One of the main applications of network tomography is to detect heavily loaded links and subnets [8]–[10]. Another important work is presented in [11] describing the use of a novel and efficient data structure called neighbor graphs, which dynamically capture the mobility topology of a wireless network as a mean for pre-positioning the stations context ensuring that the stations context always remains one hop ahead. [12] proposes a new way of measuring and extracting proximity in networks called cycle free effective conductance (CFEC). Their proximity measure can handle more than two

Analysis of the Logical Proximity between 802.11 Access Points

Ricardo Sousa
INESC TEC
Campus da FEUP
4200 - 465 Porto
Portugal
rmps@inescporto.pt

Ricardo Morla
INESC TEC
Campus da FEUP
4200 - 465 Porto
Portugal
ricardo.morla@inescporto.pt

Abel Maio
FEUP
Campus da FEUP
4200 - 465 Porto Porto
Portugal
ei07102@fe.up.pt

João Coelho
FEUP
Campus da FEUP
42 - 465 Porto
Portugal
ei07118@fe.up.pt

***Abstract*—802.11 campus networks have their access points deployed across the area the network has to cover. The physical proximity between these access points is often well understood. Network managers have maps of the campus and of the location of the access points. Due to reflections indoor and between buildings, a mobile station may have connectivity to only one of two physically nearby access points. This may mislead the network manager into thinking that an area is densely covered when in fact mobile stations cannot connect to all of the access points in that area. This problem can have a dynamic nature if we consider people and objects that move around and change the properties of the propagation medium. In this paper we explore an alternative measure to physical proximity based on mobile station connectivity to the access points, which we call logical proximity. We take the ping-pong effect as a proxy to logical proximity. We use this proximity to characterize 802.11 campus network with over 200 access points and 14k users over a 2 year period. We report on the magnitude of the ping pong effect, the clustering of access points, and the degree distribution of the resulting access point proximity network.**

***Index Terms*—Local area networks.**

I. Introduction

The deployment of wireless networks technology has attracted attention to the issues that network managers face. One of such issues is to adequately characterize the logical proximity of access points. This enables the network managers to view the network from the perspective of the user and of possible access points that mobile stations can connect to. In this work we aim to identify the logical proximity of the access points using the ping pong effect as a proxy for such proximity. The ping pong effect can be characterized by a series of consecutive connections to different access points and is the result of the aggressive nature of 802.11 interfaces that try to connect to an access point with a better signal once the signal from the current access point drops below a threshold.

In the past, research has been developed to understand network topology. The authors in [1] propose a mobility-aware clustering algorithm that uses roaming events as the metric to evaluate the proximity to access-points (APs) without using any geographical information. In this work, they make reference to three main goals for studying the mobility of the users on wireless networks: (1) to understand what are the implications and how mobility can have an impact on the network services, (2) to create realistic mobility models to evaluate the performance of protocols and algorithms, and (3) to propose new communication solutions that can adapt to the specificities of each user.

Another investigation that approach this subject is project SPOTS [2]. In this project, the aim is to create a better understanding of the daily working and living patterns of the MIT academic community, which changes due to the emergence of WiFi itself. Tang and Baker in the paper [3] analyze the network for overall user behavior (when and how intensively people use the network and how much they move around), overall network traffic and load characteristics (observed throughput and symmetry of incoming and outgoing traffic), and traffic characteristics from a user point of view (observing a mix of applications and number of hosts connected to by users).

[4] analyzes the Wi-Fi network as a proxy to space usage aiming to use it as a mean for the characterization of physical spaces and, consequently, as a source of information for a dynamic symbolic model representing those spaces. In [5] a general methodology is presented for extracting mobility information from wireless network traces, and for classifying mobile users and APs. Kim and Kotz [6] propose a model of user movements between APs. In their paper they define three goals in developing a mobility model. First, the model should reflect real user movements, second, the model should be general enough to describe the movements of every device and third, the model should consider the hourly variations over a day. Another area with many interesting works is network tomography, originating from a research by Vardi [7]. One of the main applications of network tomography is to detect heavily loaded links and subnets [8]–[10]. Another important work is presented in [11] describing the use of a novel and efficient data structure called neighbor graphs, which dynamically capture the mobility topology of a wireless network as a mean for pre-positioning the stations context ensuring that the stations context always remains one hop ahead. [12] proposes a new way of measuring and extracting proximity in networks called cycle free effective conductance (CFEC). Their proximity measure can handle more than two

end points, directed edges, is statistically well-behaved, and produces an effectiveness score for the computed subgraphs.

The investigation presented on project NearMe [13] is a way to find people and things that are in your physical proximity using Wi-Fi. The project consists in a server, algorithms, and application programming interfaces (APIs) for clients equipped with 802.11 wireless networking (Wi-Fi) to compute lists of people and things that are physically nearby.

The research described in this paper is related to several other projects and technologies in ubiquitous computing, including location sensing, proximity measurement, and device discovery. The main difference is that it looks at the ping pong effect as an indicator of logical proximity. We intend to identify access points that are near each other, without any prior knowledge about the location or proximity between them. The main contribution of this paper is to develop an algorithm that will generate a graph to describe the closeness between logical access points using only historical data from the wifi network. In Section II we describe the dataset we use, followed by the algorithm to find the proximity of the access points in section III. Section IV presents results of applying the algorithm to the dataset. In Section V we present our conclusions.

II. Dataset

A. Attributes

Each record with index $r = 1..R$ in the dataset represents a session of a user at any given access point as recorded by the RADIUS authentication server [14], [15] and has the following attributes:

- User identifier, index $u = 1..U$
- Access point identifier, index $a = 1..A$
- Location description, index L^a
- Session start time (resolution in seconds), as $v\_start_i^{(u,a)}$.
- Session duration (resolution in seconds), as $s\_time_i^{(u,a)}$

Index i ranges from 1 to the number of sessions user u has in access point a.

B. Description

The data set that we use in this paper was collected from November 2006 to March 2009. $U = 14,167$ users were observed to connect to $A = 217$ access points in $R = 6,249,992$ sessions. Figure 1 shows the time series of the ratio of active users, active access points, and observed sessions per week. Active users and access points are those for which at least one session has been observed during the week.

The ratios are against the maximum of 2707 users in a week, 206 access points in a week, and over 115 thousand sessions in a week. We can observe a crest of the number of sessions and of active users during the summer terms, which drop from approximately 50% to less than 10%. The first semester (Fall) in both years shows a sustained increase of these numbers, whereas after January these are still high but much more irregular until the summer crest. The number of active access points also has a crest in the summer terms but with a much smaller variation (approximately 75% to 65% in the first crest and 95% to 80% in the second crest).

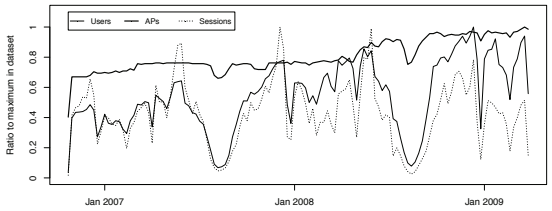


Fig. 1. Weekly time series of the dataset.

This means that the users that remain on campus during the summer use almost all of the access points that are used during the first and second semesters.

Figure 2 shows the cumulative distribution function (CDF) of the number of users and sessions per access point. A few access points seem to have much more sessions than the rest: more than 90% of the access points have less than 65k sessions while fewer than 10% of the access points have between 65k and 200k sessions. Users distribution per access point seems to be less skewed: more than 90% of the access points have sessions from less than 2800 users while fewer than 10% access points have between 2800 and 4500 users.

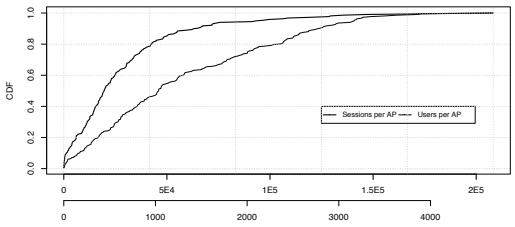


Fig. 2. CDF of the number of users and sessions per access point.

Figure 3 shows the CDF of session durations. More than 70% of the sessions have less than 5 minutes and more than 20% have sessions smaller than 20 seconds. This points to a large majority of small sessions. We also notice a log-linear distribution for sessions between 5 minutes and 4 hours. More than 99% of the sessions are smaller than 4 hours.

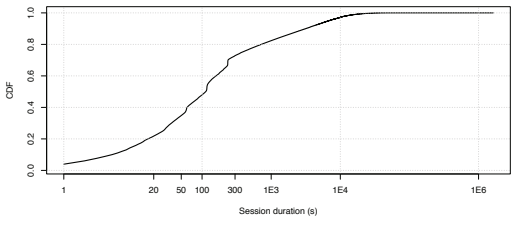


Fig. 3. CDF of session duration.

Figure 4 shows the CDF of the time between consecutive sessions by the same user. In more than 30% of the cases

the time between sessions is reported as 0s (notice that the granularity of the record is 1s) and in more than 24% of the cases it is reported as 1s (the log scale prevents plotting these values on the figure). In 71% of the cases, the time between sessions is smaller than 15s. This means that most of the session changes are due to handover and only a small portion due to users connecting and disconnecting their mobile devices.

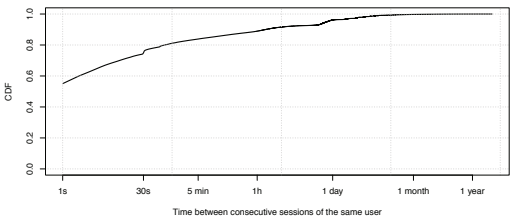


Fig. 4. Intersession CDF.

III. ALGORITHM

The goal of the algorithm presented in this paper is to identify the logical proximity of the access points of a wireless network. We take the ping-pong effect as a proxy to logical proximity. Ping-pong effect refers to the succession of associations-dissociations between two ore more Access points. So it is necessary to analyze the processes of commutation between access points that are made by each user on the network usage.

Before we start reading the data presented in Section II, we must introduce the threshold value that defines the segments of the session. The first step of the algorithm is to read the data, to organize and order the events by the user. This action also creates a list of users in the dataset. The central idea of this algorithm is to create different segments of sessions for each user. The segment of the session consists of grouping the set of access points, when the commutation time is lower than the threshold value. When creating segments of session of the user, the algorithm can identify the access points used in this segment. This way it is possible to identify the access points that are near each other.

In certain cases, when the connection is established, the user may use different access points. The purpose of the algorithm is to interpret the fast commutations made between access points to obtain a logical topology proximity between the access points using only the raw data network.

For consecutive access points, the algorithm checks if the commutation time between access points is lower than the threshold. If that's the case, the algorithm groups the access points in the same segment session. In case the commutation is higher than the threshold, the algorithm creates a new segment.

For each segment, the algorithm identifies which is the dominant access point. The dominant access point is the more frequent in the segment of the session. After the dominant access point is identified, the algorithm updates a commutation matrix. This matrix allows to draw a network graph and creates a view of the topology of the logical proximity of the access points. To update the commutation matrix (Matrix_Commutation), firstly the algorithm needs to identify the line position of the dominant access point. Then it needs to identify the position of the other access points in the columns of the matrix. When the position of the corresponding cell is identified for the two access points, the value 1 is added.

Algorithm 1: proximity of ap based on segmentation of user sessions

```
Data: Dataset → Collection of network events
Result: Matrix_Mean_Commutation
input  : time threshold
1 Dataset ⇒ Group and order the collection of events by user
  /* create a list of all the users                                  */
2 List_users→ all(username in Dataset)
3 for val_users ∈ List_users do
      /* foundRows is the structure collects the links of each user  */
      foundRows=subset(Dataset.username == val_user)
      for (i = 1; i <= foundRows.count; i++) do
          v_source=foundRows[i-1][Apcode]
          v_destine = foundRows[i][Apcode]
          v_start = foundRows[i-1][Time_Stamp]
          v_end = foundRows[i][Time_Stamp]
          commutation = Convert.ToDouble(End-Start);
          ResCom → function.save(val_user,v_start,commutation,v_end
          ,v_source,v_destine);
      end
  end
4 for val_users ∈ ResCom do
      result_Session_user → grouping the sessions when the
      commutation time is less threshold defined
      foreach Session ∈ result_Session_suser do
          Ap_Dominat → search the access point with the largest
          presence in the session
          List_near_ap → list of other access points in the session
          Matrix_Commutation → In Matrix update the line of the
          dominant access point
      end
  end
```

A. Example

To better understand the algorithm we analyze one simple example. After reading the data, the first step of the algorithm is to group and sort the events for each user, which means the events are ordered by Time_stamp. The second step is to identify the users present in the dataset and store them in a structure called List_users. For each user is created a structure called foundrows. This structure temporarily stores each user's events.

Username	Time_stamp	Sessiontime	Apcode
10	t_stamp_0	s_time_0	6
10	t_stamp_1	s_time_1	3
10	t_stamp_2	s_time_2	3
10	t_stamp_3	s_time_3	4
10	t_stamp_4	s_time_4	2
10	t_stamp_5	s_time_5	5
10	t_stamp_6	s_time_6	1
10	t_stamp_7	s_time_7	2
10	t_stamp_8	s_time_8	1

TABLE I
EXAMPLE THE STRUCTURE FOUNDROWS FOR ONE USER

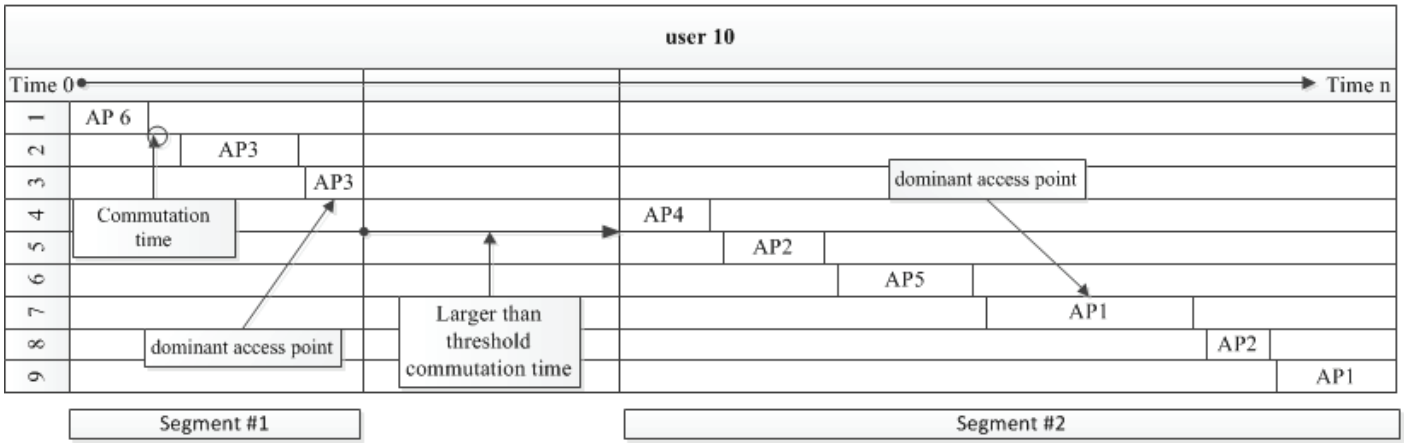


Fig. 5. Figure to represent the activity the user in network

The Third step of this algorithm is important since it will define the structure of the commutation times we call ResCom, which calculates the comutation time between access points. Therefore, we used the equation commut, which aims to determine the time required for commutation between access points. We calculate the time end (v_end) by removing the Sessiontime of the Time_stamp.

$$commut[i-1] = t\_stamp\_[i] - s\_time\_[i] \qquad (1)$$

From the subset of data relating to the example presented it is possible to obtain the following results in table II:

user	v_star	v_end	commut.	v_source	v_destine
10	t_stamp_0	t_stamp_1 - s_time_1	1	6	3
10	t_stamp_1	t_stamp_2 - s_time_2	6	3	3
10	t_stamp_2	t_stamp_3 - s_time_3	21785483	3	4
10	t_stamp_3	t_stamp_4 - s_time_4	1	4	2
10	t_stamp_4	t_stamp_5 - s_time_5	2	2	5
10	t_stamp_5	t_stamp_6 - s_time_6	8	5	1
10	t_stamp_6	t_stamp_7 - s_time_7	170	1	2
10	t_stamp_7	t_stamp_8 - s_time_8	1	2	1

TABLE II
STRUCTURE RESCOM FOR COMMUTATION BY USER

Observing graphically the results of the commutation time described in with the previous table, we get to the graph of 6:

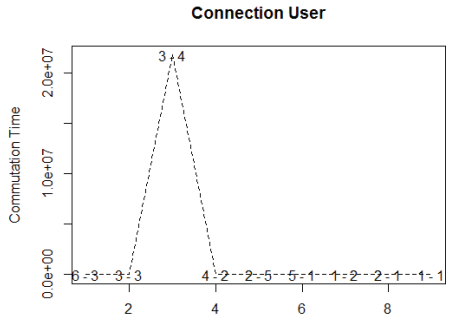


Fig. 6. Graph the commutation of user

For example, observing figures 5 and 6, it's can be seen that at a given moment the time of commutation between access points is much bigger, which means that for some reason the user has left the premises, e.g., he returned after a few hours or finished the day's work. The algorithm will identify for this case two different moments of use. This way it will create two distinct segments of user session.

To create the matrix commutation, the algorithm needs to examine the segments of session. First it identifies the dominant access point. When the dominant access point is identified the algorithm searches the remaining access points present in the same session.

Returning to the example, in the first block that defines the user session, the algorithm extracts the access point 3 as dominant. This happens because it is the access point with the largest presence in the session. Then, it finds the other access points in the same session. In this case we only have the access point 6.

To update the Matrix_Commutation, the first step consists of finding the position of the dominant access point in the first row. In the following step, it finds the position in the columns for the other access points present in the segment of session, adding the value 1 to the corresponding cell.

Updating the matrix, with the results of the first session of this user, the algorithm in cell $\{6,3\}$ adds the value 1.

For the second session of this user, the algorithm makes the same process, but now we obtain the access point 1 as the dominant. In this segment of session the following access points $\{4,5,2\}$ are present. To update the line of the dominant access point, we add 1 to the corresponding access points that are present in the session.

Next we assume that the user 11 presents the following events in the structure ResCom:

user	v_star	v_end	commut.	v_source	v_destine
11	t_stamp_0	t_stamp_1 - s_time_1	1	6	3

TABLE III
COMMUTATION BY USER

In this case, as there is no dominant access point, the algorithm assumes that the first access point is the dominant. In the previous example the dominant access point is 6. Then it increments the value 1 to the cell {6,3} in the Matrix_Commutation.

For this example, we obtain the Matrix_Commutation of the following table IV:

AP	1	2	3	4	5	6
1		1		1	1	
2						
3						
4						
5						
6			2			

TABLE IV
MATRIX WITH NUMBER COMMUTATION BETWEEN ACCESS POINT

The algorithm draws the network graph, as shown in the figure 7. The connection between the access points 6 and 3 is represented by one larger and darker line, Which means these two access points are probably closer. This happens because the number of commutations between these two access points is more evident.

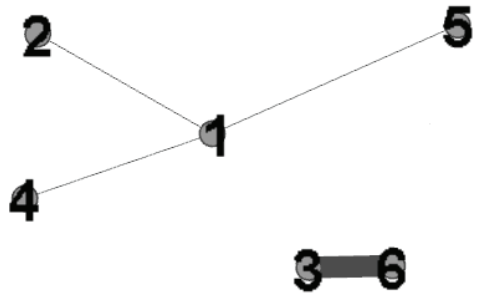


Fig. 7. Graph of frequency of AP

This algorithm has some advantages, the first of which is that it enables a idea of a ping-pong effect between access points. Other advantage is that identifies the dominant access point which has more impact in the network.

IV. RESULTS

To implement the algorithm we used the C# language and developed the tool ProxAP as can be seen in Figure 8. With this tool, it is possible to visualize the matrix commutation generated by the algorithm.

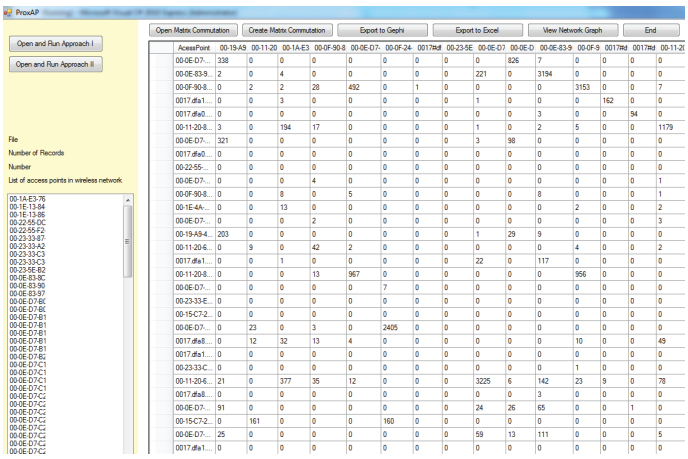


Fig. 8. Tool ProxAP

This tool allows to export the results to the Gephi [16]. It is an open-source software to visualize and analyze network graphs. This way, it is possible to observe the results generated by the algorithm presented in this article.

When we apply the data presented in section II to the algorithm, with a a threshold of 100 seconds, we obtain the commutation matrix. When these results are used in the Gephi tool, applying the algorithm Yifan Hu, with a filter higher than 200 connections, we obtain the figure 9.

We decided to use the algorithm Yifan Hu and filters it allows a better visualization of network graphs.

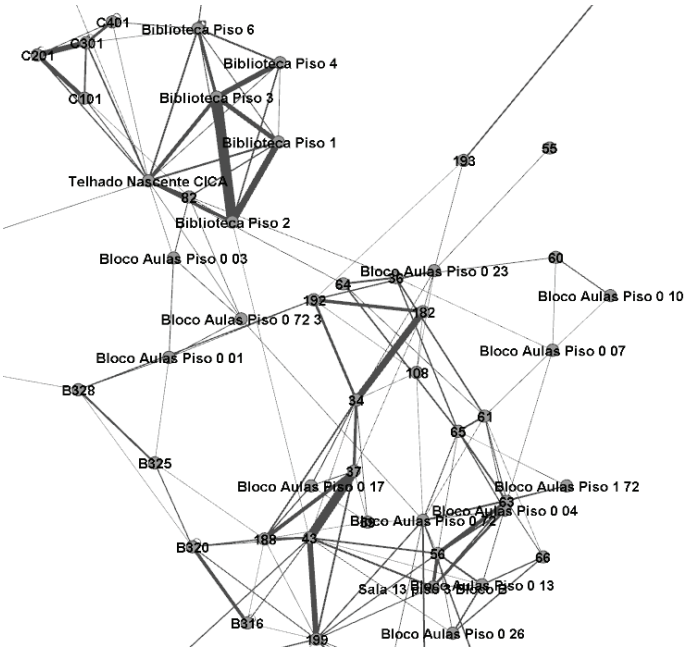


Fig. 9. Graph of frequency of AP

Using the application Gephi, the administrator can obtain many views of network topology. Making a closer view in the graph to the access point ”Biblioteca Piso 3” we obtain:

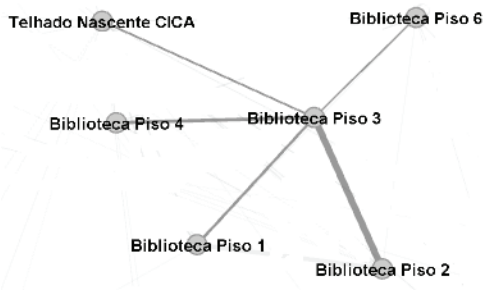


Fig. 10. Graph of frequency of AP

Using another location with a strong impact on the graph created, we obtain:

Fig. 11. Graph of frequency of AP

This way, the network administrator has the opportunity to see the access points with bigger ping-pong effect on the network.

A. Analysis of number of commutations

In this section we analyze the number of commutations of one user from one access point to the same access point and to other access points.

Figure 12 shows the CDF of the number of commutations to the same access point in absolute value and relative to the number of sessions in the access point. The number of commutations to itself is relatively small (up to tens of thousands) compared to the number of sessions which can go up to hundreds of thousands of sessions. This is because each commutation is a segment of a potentially large number of sessions. More than 90% of the access points have less than 10% ratio of the number of commutation to their number of sessions. There are 4 access points with this ratio above 20% have a small number of sessions (less than 300), which puts them in the very beginning of the session per AP distribution and says these are scarcely used access points.

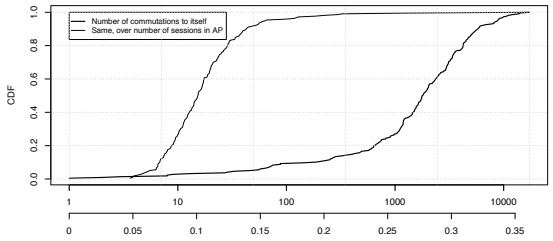


Fig. 12. CDF of commutation to self.

Figure 13 shows the CDF of the percentage of commutations to other access points relative to the total commutations (both to itself and to others). No access point has only commutations to itself (the minimum is 42% of the total commutations). Most access points have more commutations to other access points than to themselves (70% have between 50% and 80% commutation ratio). A significant percentage of access points (5%) has more than 85% of their commutations to other access points.

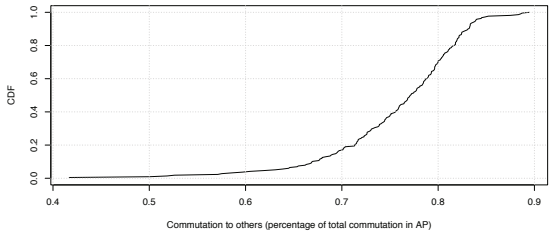


Fig. 13. CDF of commutation to others.

B. Chinese Whispers Clustering

The goal of applying this clustering technique is to easily identify access points considered near by the algorithm. Using the algorithm Chinese Whispers Clustering in Gephi tool we obtain 24 clusters. The Clustering is the process of grouping together objects based on their similarity to each other [17]. This means that for our example we have 24 clusters with access points considered near each other.

Selecting randomly one cluster we obtain the following figure 14.

Fig. 14. Selected the first cluster

For example, if we choose the cluster that has the largest number of access points, we get figure 15:

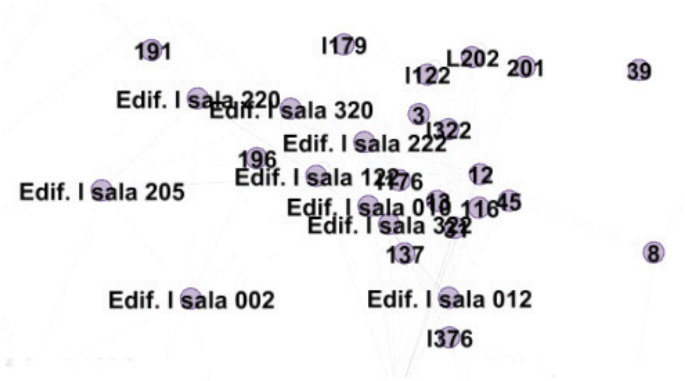


Fig. 15. Cluster with the largest number of elements

This way, the network administrator has information that many access points are concentrated in building "I". This important so that he a precise knowledge about the use and proximity between access points.

C. Average Degree Distribution

In this article we explore the Average Degree Distribution metric that is available in the Gephi [16] to evaluate the results generated by the algorithm.

In the study of networks, the degree of a node in a network is the number of connections it has to other nodes and the degree distribution is the probability distribution of these degrees over the whole network [18].

From the example discussed in this paper we obtain 6,958 as the Average Degree Distribution.

For Average Degree Distribution 6,958:

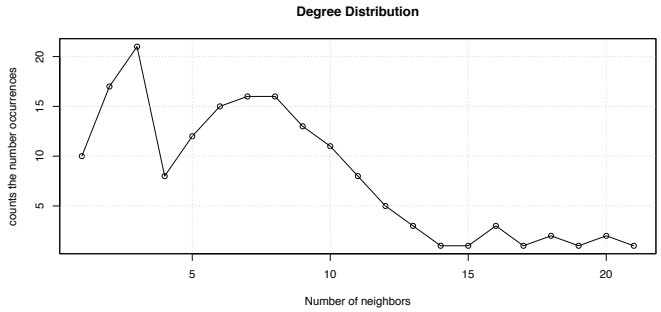


Fig. 16. Average Degree Distribution 6,958

One of the advantages in using this metric to enable the administrator is to understand how the access points are distributed. The figure 16 shows that there is a small set of access points where the number of occurrences is bigger. This information can be useful to identify the access points that have this behaviour and to understand the reason why this happens.

Figure 17 shows the CDF of the node degree. As we can see most access points have a value smaller than 10. This means most of the access points have less than 10 access points nearby.

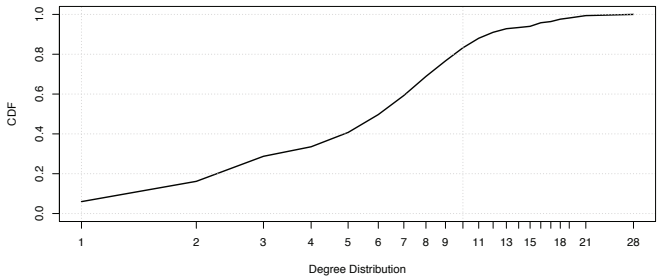


Fig. 17. CDF Degree Distribution

V. Conclusion

In this article we developed and explored an algorithm to identify the proximity of access points in 802.11 campus network. The principal objective of the algorithm developed is to provide network administrators with a logical view of the proximity between access points. For the logical topology of proximity between the access points, the algorithm makes the analysis of raw data collected for a period of about 2 years.

The main task of the algorithm is to characterize the ping-pong effect between access points. Thus, it builds the matrix which defines the logical proximity between access points. This way the network administrator can overlap a graph of nearby access points with metrics and indicators of network usage and performance and visually detect geographic correlation of lower quality indicators.

Acknowledgment

This work is financed by the ERDF- European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT- Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projectS PTDC/EIA-EIA/113933/2009 and (FCOMP-01-0124-FEDER-015064).

References

[1] M. Boc, A. Fladenmuller, and M. de Amorim, "Towards self-characterization of user mobility patterns," in *Mobile and Wireless Communications Summit, 2007. 16th IST*, july 2007, pp. 1 –5.

[2] A. Sevtsuk, S. Huang, F. Calabrese, and C. Ratti, *Mapping the MIT campus in real time using WiFi*. Hershey, PA: IGI Global, 2009.

[3] D. Tang and M. Baker, "Analysis of a local-area wireless network," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/345910.345912

[4] K. Baras and A. Moreira, "Symbolic space modeling based on wifi network data analysis," in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, june 2010, pp. 273 –276.

[5] W. jen Hsu and A. Helmy, "On modeling user associations in wireless lan traces on university campuses," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, april 2006, pp. 1 – 9.

[6] M. Kim and D. Kotz, "Modeling users' mobility among wifi access points," in *Papers presented at the 2005 workshop on Wireless traffic measurements and modeling*, ser. WiTMeMo '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 19–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1072430.1072434

[7] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, Mar. 1996. [Online]. Available: http://dx.doi.org/10.2307/2291416

[8] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statistical Science*, vol. 19, pp. 499–517, 2004.

[9] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, "Toward the practical use of network tomography for internet topology discovery," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[10] D. Ghita, K. Argyraki, and P. Thiran, "Network tomography on correlated links," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 225–238. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879170

[11] A. Mishra, M. Shin, and W. Arbaush, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, march 2004, pp. 4 vol.(xxxv+2866).

[12] Y. Koren, S. C. North, and C. Volinsky, "Measuring and extracting proximity graphs in networks," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 3, Dec. 2007. [Online]. Available: http://doi.acm.org/10.1145/1297332.1297336

[13] J. Krumm and K. Hinckley, "The nearme wireless proximity server," in *UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, September 7-10, 2004. Proceedings*, ser. Lecture Notes in Computer Science, N. Davies, E. D. Mynatt, and I. Siio, Eds., vol. 3205. Springer, 2004, pp. 283–300.

[14] C. Rigney, A. C. Rubens, W. A. Simpson, and S. Willens, "Remote authentication dial in user service (radius)," Internet RFC 2865, June 2000.

[15] C. Rigney, "RADIUS Accounting," RFC 2866 (Informational), Internet Engineering Task Force, June 2000, updated by RFCs 2867, 5080. [Online]. Available: http://www.ietf.org/rfc/rfc2866.txt

[16] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[17] C. Biemann, "Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, ser. TextGraphs-1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 73–80.

[18] Wikipedia, "Degree distribution," 2012, [Online; accessed 10-Set-2012]. [Online]. Available: http://en.wikipedia.org/wiki/Degree_distribution

Encaminhamento *Anycast* em Redes *IPv6*: uma proposta

Hugo Ferreira
Centro ALGORITMI
Universidade do Minho
E-mail: a50046@alunos.uminho.pt

Maria João Nicolau
Centro ALGORITMI
Universidade do Minho
E-mail: joao@dsi.uminho.pt

António Costa
Centro ALGORITMI
Universidade do Minho
E-mail: costa@di.uminho.pt

***Resumo*—O aparecimento do protocolo de comunicação *IPv6* introduziu um novo paradigma de comunicação, denominado *anycast* (um-para-um-de-muitos). Este novo paradigma, utiliza o conceito de grupo, à semelhança do que acontece com o *multicast*, mas em oposição a este, a informação é enviada apenas para um dos membros do grupo (tipicamente o mais próximo) e não para todos. Embora já se tenham passado alguns anos desde o seu aparecimento , o *anycast* tem sofrido uma lenta evolução, contribuindo para esta situação o facto de não existir ainda um protocolo normalizado, que permita às aplicações usar de forma generalizada este paradigma de comunicação.**

Tradicionalmente as soluções para o problema de encaminhamento *anycast* são simplesmente baseadas no encaminhamento *unicast* sem alterações. No entanto, e tratando-se de um paradigma que usa o conceito de grupo, é de esperar que os protocolos de encaminhamento *multicast*, ou alguma variante destes, possam vir a constituir uma boa solução para a implementação do *anycast* ao nível da rede. O presente artigo apresenta um levantamento de propostas relacionadas com o tema e propõe um novo protocolo de encaminhamento *anycast* baseado no protocolo *PIM-SM (Protocol Independent Multicast -Sparse Mode)*, denominado *Tree-based Anycast Protocol (TAP)*. As alterações propostas ao protocolo *PIM-SM* são apresentadas na especificação do sistema, tendo sido o seu correto funcionamento aferido recorrendo ao *Network Simulator 2 (ns-2.35)*.

Palavras-chave:*Anycast*, Encaminhamento, Redes, *IPv6*, *PIM-SM*, *TAP*.

I. Introdução

O protocolo de comunicação *IPv6* incluí três paradigmas de comunicação - o *unicast*, o *multicast* e o *anycast*. O *unicast* é o mais comum, estimando-se que noventa por cento do tráfego seja de um para um. O tráfego *multicast* (um-para-muitos) surgiu como uma evolução natural do antigo *broadcast* (um-para-todos), onde a comunicação deixa de ser ”para todos”e aparece o conceito de grupo. Passa a ser possível comunicar com um grupo definido de utilizadores. O *anycast* é originalmente proposto[1] a Novembro de 1993, sendo particularmente útil para implementar serviços que podem ser disponibilizados por múltiplos servidores. É atribuído um único endereço *anycast* a todos os sistemas terminais que disponibilize exatamente o mesmo serviço. Para os clientes desse serviço, basta-lhes dirigir o pedido ao endereço *anycast* e esperar que um dos sistemas (e apenas um) lhe responda. Do ponto de vista do serviço prestado é indiferente qual deles é, mas a escolha do sistema em melhores condições é um problema de encaminhamento específico da rede.

Os endereços *anycast* são sintaticamente indistinguíveis dos endereços *unicast*[2]. A lógica inerente a este tipo de endereçamento é de um endereço *unicast* em diferentes locais. A ideia visa permitir a um provedor de serviços aumentar a capacidade de balanceamento de carga, acrescentando um novo servidor numa outra rede. Quando um utilizador requer um determinado serviço, este é encaminhado para o servidor mais próximo da sua localização (com menor custo), utilizando o método de encaminhamento do *unicast*.

Na Figura 1 é possível observar um exemplo de uma comunicação *anycast*. Um endereço *anycast* único é atribuído aos três provedores de serviços - `Servidor Anycast 1`, `Servidor Anycast 2`, `Servidor Anycast 3`. Quando um `Cliente Anycast` deseja efetuar um pedido, cria um pacote *anycast* (pacote que contém um endereço *anycast* como destino) e envia esse pacote para a rede. Após o envio do pacote até à receção por parte do servidor mais próximo, é levado a cabo um processo de seleção (normalmente o caminho de menor custo). Aquando da receção do pacote *anycast* por parte do servidor mais próximo do cliente, é enviada a resposta ao pedido do cliente, idealmente utilizando o endereço *anycast* no campo de origem de modo a evitar problemas na altura da receção do pacote pelo cliente.

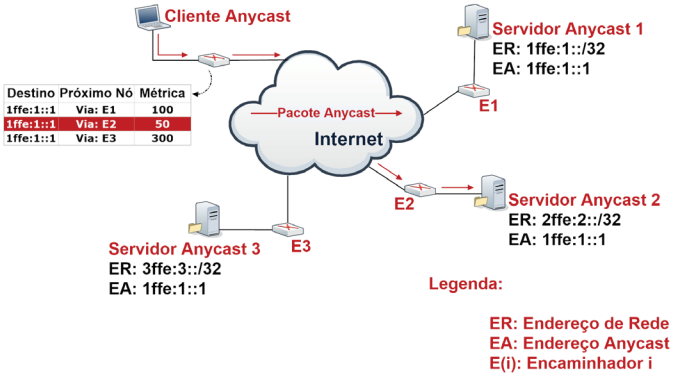


Figura 1: Esquema do encaminhamento *anycast*.

No encaminhamento global, o *anycast* prejudica a agregação de rotas pois permite que um mesmo endereço

apareça associado a várias redes distintas. Como podemos observar na Figura 1, é possível ao `Servidor Anycast 2` e ao `Servidor Anycast 3` estarem em redes diferentes da rede a que pertence o do endereço *anycast*. Um encaminhador que receba rotas relativas a estas redes, é facilmente iludido. Para conseguir distingui-las, é necessário adicionar entradas separadas a todos os encaminhadores ao longo da rede. Utilizando o mesmo exemplo, caso os servidores se encontrem em redes com endereços dispares pode ser muito complicado (ou mesmo impossível) efetuar a agregação de rotas. É necessário um novo sistema para encaminhamento, para retirar partido do potencial do *anycast*. Existem diversas propostas no sentido de atacar esta problemática, mas existem ainda muitos problemas por resolver[3][4][5][6][7][8].

Não obstante ao problema da agregação das rotas, o encaminhamento *anycast* possuí diversas características interessantes. Uma das características mais importantes do *anycast* quando realizado na camada de rede é não ser necessário ao cliente conhecer a topologia total da rede, o que lhe confere uma enorme escalabilidade. Quando o `Cliente Anycast` na Figura 1 faz um pedido, este é reencaminhado para o nó mais próximo dele (neste caso o `Servidor Anycast 1`). A grande vantagem acontece quando por qualquer motivo (por exemplo falha de *hardware*) o nó onde habita o `Servidor Anycast 1` deixa de estar acessível. Neste caso o seu pedido será na mesma respondido, mas por um servidor a uma maior distância.

O *anycast* tem enumeras aplicações, das quais se destacam os serviços dependentes da localização, a descoberta de serviços e o balanceamento de carga.

Atualmente, uma das mais populares aplicações são os serviços dependentes da localização. A escolha do servidor mais próximo do requerente do serviço é uma questão cada vez mais critica. Normalmente passa por apresentar uma lista de servidores distribuídos por múltiplos locais e oferecer o poder de seleção ao requerente do serviço. Com a utilização do *anycast* neste tipo de aplicações, garante-se que é feita uma seleção mais precisa para o encaminhamento do pedido. Em vez de confiar no juízo do requerente, realiza-se um encaminhamento mais transparente a partir de qualquer parte do mundo. Com isto o *anycast* assume um papel essencial pois permite escolher o nó mais próximo, fazendo assim com que exista uma resposta mais célere e eficaz.

Para ser possível a descoberta de serviços, primeiramente é atribuído um endereço *anycast* a um serviço, e de seguida é atribuído aos nós que suportam os servidores desse mesmo serviço. Isto permite uma grande tolerância a falhas na rede, pois o pacote *anycast* será reencaminhado para um outro local apropriado. Este tipo de serviço assume uma grande importância em redes com grande escalabilidade, como redes móveis *ad hoc* ou de sensores, onde a sua topologia muda constantemente. Nestas redes, mais importante que descobrir os serviços, é saber que existe a disponibilidade para o obter.

Com o volume de tráfego a aumentar diariamente na *Internet*, a necessidade de providenciar a resposta por partes dos servidores aos pedidos torna-se cada vez mais difícil. Existem diversas opções de como melhorar a performance do sistema (desde aumento da capacidade de processamento ao aumento dos sistemas). Hoje em dia, a melhor opção é a distribuição de vários servidores por diferentes zonas (onde se pretende providenciar o serviço). Os diferentes pedidos vão sendo distribuídos pelos diferentes servidores fazendo com que os pedidos não sobrecarreguem somente uma rede. Diminui-se ainda o tempo de resposta, pois o processamento de pedidos é dividido pelos vários elementos do grupo.

A principal motivação do presente artigo é criar um protocolo de encaminhamento *anycast* inter-domínio, denominado de *Tree-based Anycast Protocol (TAP)*.

O artigo encontra-se divido em 5 secções. A próxima secção apresenta um estudo dos trabalhos relacionados, analisando cada uma das propostas, realçando os problemas que estas conseguem resolver e os que ainda deixam em aberto. Na secção 3 é apresentado o *TAP*, sendo especificadas as suas principais vertentes. A secção 4 descreve a implementação do *TAP* no *NS-2*, ilustrando o seu funcionamento com recurso à ferramenta *NAM*. É ainda realizada uma pequena comparação com o principal protocolo da literatura estudada. Por fim, a secção 5 apresenta as conclusões e o trabalho futuro.

II. Trabalho Relacionado

A proposta *GIA (Global IP Anycast)*[3], tem como característica distintiva o facto de introduzir um novo sistema de endereçamento, obrigando a uma mudança no *IPv6*. Através do aparecimento do endereço *anycast*, é possível aos encaminhadores reconhecer e tratar o tráfego de acordo com as especificações *anycast*. Utilizando o argumento de que um sistema autónomo (SA) só deverá possuir rotas para grupos *anycast* que lhe interessam, os autores do *GIA*, propõem uma divisão em três diferentes grupos - Grupo Interno, Grupo Externo Popular e Grupo Externo Impopular.

Quando um domínio possuí um grupo *anycast* no seu interior, é considerado um Grupo Interno. Como se trata de encaminhamento intra-domínio, o número de rotas acrescentadas às tabelas não é crítico, possuindo no máximo uma entrada para cada servidor. Se um domínio deteta um número significativo de utilizadores a aceder a um endereço *anycast* externo, catalogará esse serviço como Grupo Externo Popular. Quando existe um grande interesse por parte dos utilizadores do domínio num grupo *anycast* especifico, desencadeia-se uma série de procedimentos para tentar encontrar o melhor caminho para esse grupo. O *GIA* define um novo pacote *BGP*[9], que tem como objetivo obter o melhor caminho. O encaminhador fronteira (*border router*) do domínio, envia o pacote, sendo este reenviado através dos encaminhadores até a sua validade expirar ($TTL < 0$) ou um encaminhador responder a esse pedido. Um encaminhador responde a um pedido de procura se conhecer um caminho para o grupo melhor que o atual. Quando o encaminhador que originou a mensagem recebe a resposta, atualiza a sua tabela de encaminhamento *anycast* e cria um túnel entre si e o encaminhador que respondeu. Os pacotes *anycast* seguintes, passam a ser encaminhados através do melhor caminho descoberto. Todos os outros grupos,

que não interessam ao domínio, são considerados de Grupo Externo Impopular. Assim, o domínio adiciona uma rota padrão (*default route*), não sobrecarregando as suas tabelas de encaminhamento.

A proposta *PIAS (Proxy IP Anycast Service)*[4] , introduz uma nova abordagem ao encaminhamento *anycast*, a inclusão de redes sobrepostas. A ideia é de distribuir um grande número de *proxys* pela *Internet*, com o intuito de resolver o problema de escalabilidade do encaminhamento *anycast* ao nível da camada de rede. Os *proxys* funcionam como gestores da rede *anycast*, anunciando rotas para grupos *anycast* através do *BGP*[9]. A responsabilidade da entrega dos pacotes não recai sobre os encaminhadores, sendo estes entregues pelo *proxy* mais próximo, através de encaminhamento *unicast* (Figura 2). As soluções baseadas em redes sobrepostas nem sempre encaminham pelo melhor caminho, mas consegue melhorar a escalabilidade. Com este sistema de gestão de tráfego *anycast*, excluem-se as rotas de todo o grupo das tabelas de encaminhamento dos encaminhadores normais, introduzindo a melhor. Esta proposta alivia a carga colocada nos encaminhadores, mas acarreta a implementação de um grande número de *proxies*.

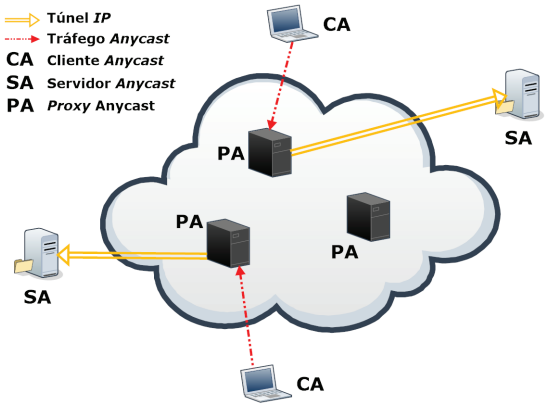


Figura 2: Arquitetura do *PIAS*.

O encaminhamento *multicast* e *anycast* possuem muitas propriedades semelhantes, o que levou a que vários investigadores estudassem a adaptação dos principais algoritmos *multicast* para a realidade *anycast*. Um do trabalhos mais notórios nesta vertente[6], escolheu três protocolos (*DVMRP*[10], *MOSPF*[11] e *PIM-SM*[12]) e procedeu ao estudo de uma possível implementação. Os dois primeiros protocolos consomem muitos recursos, sendo propostas para serem aplicadas a pequenas redes. O último protocolo (*PIM-SM*) é um dos protocolos de encaminhamento mais utilizados ao nível global, com muitas vantagens sobre os principais concorrentes. O facto de consumir poucos recursos e ser desenhado para redes alargadas, constitui-se como o maior candidato à adaptação.

Num segundo artigo[7], os investigadores focaram-se na adaptação do *PIM-SM*, para criar um protocolo de encaminhamento *anycast* denominado *PIA-SM*. O *PIA-SM*, mantém a lógica do seu protocolo base e constrói árvores unidirecionais, centradas num ponto de encontro (*RP*). O *RP* é a raiz da árvore que une os membros do grupo. Ao contrário do que acontece no *multicast*, o *RP* e todos os nós da árvore partilhada somente encaminham para uma das interfaces, a que possui melhor métrica (Figura 3). Quando o servidor recebe o pacote, estabelece a ligação com o cliente, por *unicast*.

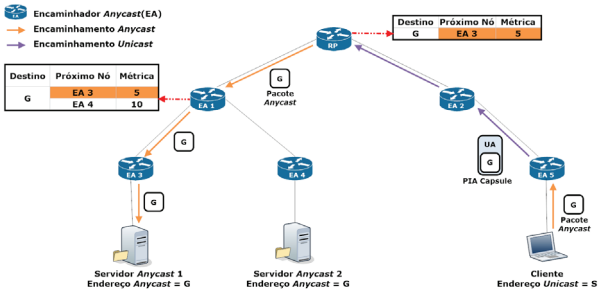


Figura 3: Encaminhamento dos pacotes *anycast* pelo *PIA-SM*

A implementação do *PIA-SM*, não contempla o caso de existir mais de que um cliente. Caso existam vários clientes simultaneamente, o servidor escolhido é sempre o mesmo, mesmo que esteja sobrecarregado. Uma outra variante do *PIA-SM*[8] aborda este problema, introduzindo um outro campo à tabela de encaminhamento para além da métrica, campo esse que tem em consideração o facto de o servidor estar ocupado.

O *PIA-SM* possui ainda uma última consideração a ser feita, relativamente à forma como é calculada a sua métrica. Como a métrica é calculada quando os servidores se juntam à árvore partilhada, a métrica obtida é relativa ao *RP*, o que pode originar que o servidor escolhido não é realmente o mais próximo do cliente, mas o mais próximo do *RP*, adulterando o princípio do encaminhamento *anycast*[2].

III. Especificação do *TAP*

O *PIM-SM* está otimizado para redes com recetores dispersamente distribuídos ao longo da rede. É designado por independente do protocolo, pois ao contrário de outros protocolos de encaminhamento *multicast* não depende de nenhum protocolo de encaminhamento *unicast* específico. Constrói árvores unidirecionais, centradas num ponto de encontro, normalmente designado por *Rendezvous Point (RP)*. O papel do *RP* é fundamental no protocolo, servindo de ponto de encontro entre as fontes e os recetores. O facto de utilizar um ponto de encontro para fazer esta ligação, permite aos encaminhadores reduzirem o número de entradas, pois simplesmente precisam de possuir a entrada (*,G). Devido a esta característica, a adaptação deste protocolo *multicast* à realidade *anycast*, parece ser muito vantajosa, tendo já sido realizadas algumas adaptações[7][8].

Um último aspeto a diferenciar entre a metodologia *multicast* e *anycast*, é a forma como decorre a comunicação. Enquanto no tráfego *multicast*, o grupo de clientes estabelece ligação a uma fonte (ou a um conjunto de fontes), no caso de tráfego *anycast* a ligação é entre o cliente e o servidor mais

próximo[1]. Comparando diretamente as duas metodologias, um recetor (ou recetores) no *multicast* equipara-se a um grupo de servidores no *anycast*, e a fonte no *multicast* equipara-se a um cliente.

A. *Atribuição do Endereço* Anycast *a um Grupo*

A primeira grande diferença entre o paradigma de comunicação *multicast* e o paradigma de comunicação *anycast* reside no facto de a primeira possuir um endereçamento próprio e distinguível dos endereços *unicast*. Os endereços *anycast* são absolutamente indistinguíveis dos endereços *unicast*[2], o que resulta num desafio extra para o seu encaminhamento na rede. O desafio acontece porque um encaminhador padrão[2] não consegue distinguir o tráfego e deste modo encaminhá-lo de forma distinta.

A abordagem seguida pelos autores de outra proposta que se focou igualmente no *PIM-SM*[7], considera que o endereço *anycast* atribuído a um grupo é o endereço *unicast* do nó inicial[3]. Quando um cliente efetuar um pedido, mesmo que nunca passe por um encaminhador que implemente o protocolo proposto (adiante designado por `Encaminhador Anycast (EA)`), viajará sempre até ao nó inicial.

Um encaminhador *anycast* (`EA`) é um encaminhador especial, que com um protocolo específico *anycast*, consegue identificar e encaminhar o tráfego, como acontece no *multicast*.

A criação de um esquema de endereçamento próprio é uma alternativa à abordagem anterior. O protocolo *GIA*[3] defende que, para ser possível um correto funcionamento do paradigma de comunicação *anycast*, é necessário conseguir distinguir o endereço *anycast* do *unicast*. A introdução deste novo tipo de endereço, dificulta a aceitação de um novo protocolo, pois o endereçamento utilizado é diferente do previsto na norma (indistinguível dos endereços *unicast*). Apesar desta desvantagem, as comunicações tornam-se mais seguras, pois o endereço *unicast* dos sistemas terminais nunca é conhecido. Os clientes comunicam sempre para o endereço do grupo, e os servidores respondem com o endereço de grupo no campo de origem. Outra vantagem é o facto da redução da carga sobre um `EA`, pois reduz o número de pesquisas (*lookup*) nas tabelas de encaminhamento. Um `EA` na primeira proposta (endereço indistinguível do *unicast*), é obrigado a verificar a tabela de encaminhamento *anycast* e, caso não encontre correspondência, também tem de verificar a tabela *unicast* para reencaminhar um pedido.

Na implementação do sistema, foi escolhida a introdução de um novo tipo de endereçamento *anycast*.

B. *Escolha do Melhor Servidor*

O paradigma de comunicação *anycast* necessita da introdução de um fator de seleção, que permita ao `EA` encaminhar o tráfego para o servidor mais capaz. Normalmente esse fator utilizado é o número de saltos, entre o cliente e o servidor.

Como a fixação de uma métrica pode ser algo limitador, é possível definir o cálculo da métrica de acordo com o desejo do criador do grupo *anycast*. Esta métrica pode passar por um sem número de parâmetros como o número de saltos, largura de banda, perda de pacotes, latência, carga atual do servidor, fiabilidade do trajeto, entre outros. Alguns destes parâmetros obrigam ao cálculo da métrica pelo percurso fim-a-fim, onde todos os nós a atualizam. A equação 1 é um exemplo de uma métrica combinada, sendo calculada fim-a-fim, onde todos os nós (ao longo da árvore) atualizariam a métrica.

$$\textbf{Métrica} = \textbf{f}(N^{o}deSaltos) \oplus \textbf{f}(\text{Carga do Servidor}) \oplus \oplus \textbf{f}(\text{Largura de Banda Disponível}) \quad (1)$$

Ao longo do capítulo, é adotado como métrica, um parâmetro fixo calculado previamente pelo servidor, de modo a simplificar a sua explicação e implementação. O parâmetro escolhido foi a carga total do servidor.

C. *Descoberta de Encaminhadores Vizinhos*

O funcionamento do protocolo baseia-se no facto que todos os `EA` têm conhecimento dos caminhos possíveis na sua topologia. No *PIM-SM* são trocadas mensagens *PIM Hello*, que funcionam como mensagens de reconhecimento de vizinhos. A Figura 4 ilustra como se procede a localização dos vizinhos. Todos os encaminhadores `EA` trocam mensagens de reconhecimento entre si (a laranja na Figura 4), sendo estas enviadas para o endereço *multicast* do segmento de rede. Com todas as ligações identificadas, designa-se um encaminhador responsável (*DR*) pelo envio de mensagens periódicas de aviso para o *RP*. Tal como no *PIM-SM*, o *DR* é eleito baseado no endereço lógico de cada interface, sendo escolhido o de maior valor. Cada segmento da rede tem de possuir um *DR*, como vemos na Figura 4.

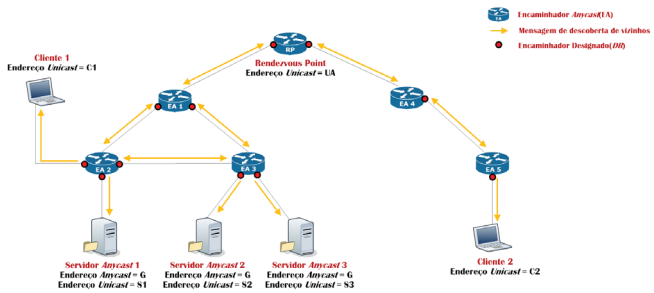


Figura 4: Descoberta de vizinhos.

Às mensagens de reconhecimento de vizinhos pode ainda ser adicionado uma opção que permita autenticar os *EA*, utilizando segurança ao nível da rede (`IPSec`).

[1] Servidor com melhor métrica.

[2] Encaminhador que não possui nenhum protocolo de encaminhamento *anycast*.

[3] Primeiro servidor a ativar-se.

D. Criação da Árvore Partilhada

A escolha do *RP* é uma das decisões mais importantes deste tipo de protocolo, existindo diversos estudos para otimizar o seu posicionamento[13][14]. Este é utilizado como ponto de registo dos servidores, para proporcionar o encaminhamento de pacotes. A construção da árvore partilhada pode ser dividida em duas parcelas, o registo dos servidores junto do `EA` mais próximo da sua origem e deste com o *RP*.

O registo de um servidor *anycast* é semelhante ao que acontece com os recetores no *multicast*. É gerada uma mensagem de ligação ao grupo desejado, enviando esta mensagem para o `EA` mais próximo. Como um administrador pode pretender implementar mais do que um servidor (para o mesmo grupo *anycast*) no mesmo segmento de rede, o `EA` deve comunicar sempre com o endereço de origem na mensagem de registo. O endereço de origem identifica o endereço local do servidor, sendo a componente inicial da mensagem de registo, como podemos verificar na Figura 5.

Figura 5: Pacote da mensagem de registo.

O registo de um servidor *anycast* junto do *RP* é efetuado pelo `EA` que é o *DR* no segmento de rede, aquando da receção da mensagem de ligação na rede local (a laranja na Figura 6). O `EA` reage ao pedido de admissão do novo servidor, criando na tabela de encaminhamento *anycast*, uma entrada (*,G). Através do envio sucessivo de mensagens de ligação (equivalente ao *join* no *PIM-SM*, a azul na Figura 6) em direção ao *RP*, constrói-se um ramo da árvore partilhada. O reenvio da mensagem de ligação serve-se sempre do nó do caminho mais curto até ao *RP*, ao longo de cada segmento.

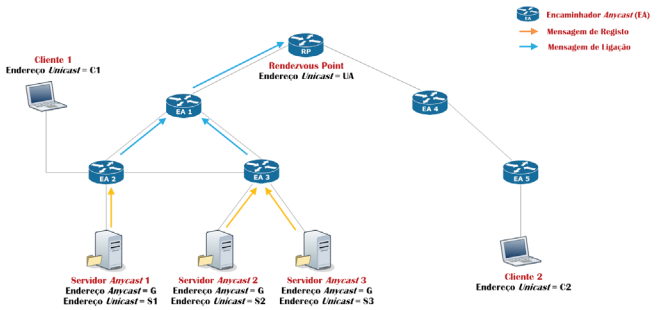


Figura 6: Criação da Árvore Partilhada.

O papel desempenhado pelo encaminhador designado, não se esgota apenas na receção de pedidos e estabelecimento das respetivas ligações, devendo acrescentar o endereço do *RP* para efetuar periodicamente notificações, de modo a conservar o(s) servidor(es) ativo(s). Um `EA` apaga a entrada (*,G) e retira o endereço do *RP* da lista de notificações, quando esse grupo não lhe interessar. O interesse deste pode ser por ter um membro diretamente ligado ou então servir de elo de ligação para outros membros.

E. Criação da Árvore Centrada no Cliente

Finalizado o processo da ativação dos servidores *anycast* junto do *RP*, é agora possível aos clientes efetuarem pedidos. A Figura 7 ilustra o processo de descobrimento dos servidores por parte dos clientes. Quando `Cliente 1 (C1)` deseja iniciar a comunicação com os servidores, "encapsula" o pacote *anycast* de descoberta numa mensagem *unicast*, enviando até ao *RP* do grupo (a azul na Figura 7). O facto de ser utilizado tráfego *unicast* permite que outros encaminhadores padrão sejam capazes de encaminhar o tráfego, mesmo não possuindo perceção do tráfego *anycast*. Ao receber o pacote, o *RP* "desencapsula" a mensagem e envia o pacote *anycast* com auxílio da árvore partilhada do grupo (a laranja na Figura 7)

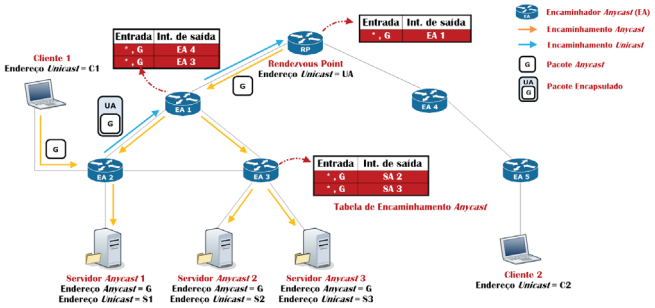


Figura 7: Pedido de localização por parte do Cliente 1.

Quando um cliente começa o processo de localização dos servidores, um temporizador é iniciado, aguardando que os servidores construam uma nova árvore centrada no cliente, de modo a poder realizar pedidos. No caso do tempo se esgotar sem nenhum servidor se conectar a si, é enviado um novo pedido e reiniciado o temporizador.

Logo que o pacote de localização alcança um servidor, este calcula a sua carga total. Como acontece na mensagem de registo na árvore partilhada, é enviado um pacote para o `EA` mais próximo. Sendo este processo replicado por todos os servidores do grupo, garante-se que um cliente poderá realmente escolher o servidor com menor carga. A Figura 8 ilustra a estrutura da mensagem para o registo na árvore centrada no cliente, onde é possível verificar uma alteração no pacote, a inclusão da métrica do servidor.

Figura 8: Pacote da mensagem de registo na árvore centrada no cliente.

Calculada a carga total para cada servidor no momento atual, é altura de comutar da árvore partilhada para a árvore

centrada na fonte. Como é possível verificar na Figura 9, todos os servidores efetuam a ligação ao C1, anunciando a sua métrica. Inicialmente todos os servidores apresentam a mesma carga, sendo escolhido o servidor que enviou a mensagem primeiro. Na Figura, o C1 receberia primeiro a mensagem do SA 1, pois este está mais próximo da sua localização.

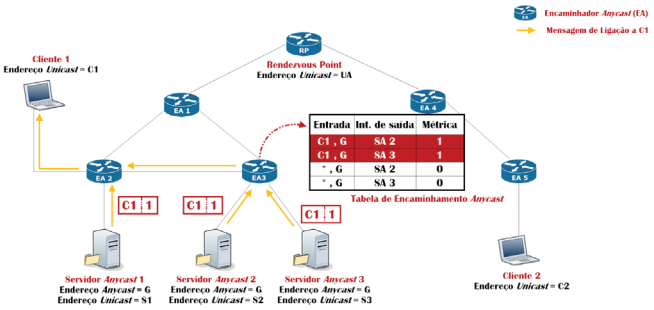


Figura 9: Criação da árvore centrada no cliente.

Quando um servidor decide centrar-se no cliente, o encaminhador designado (*DR*) do seu segmento de rede, começa por adicionar uma entrada (C,G) à sua tabela de encaminhamento *anycast*. A constituição da nova entrada possui uma grande diferença em relação ao comportamento do *PIM-SM*. Enquanto o *PIM-SM* inicialmente coloca a 0 a *flag* SPT-BIT, aqui é colocada a 1, pois a comunicação deverá ser automaticamente comutada para a árvore centrada no cliente.É enviado um pedido de exclusão para o grupo em direção ao *RP*, de modo a alerta-lo (e aos outros *EA*), que a partir daquele momento a comunicação decorrerá a através da árvore de centrada no cliente.

A exemplo do que acontece na construção de uma árvore partilhada, a árvore centrada no cliente deve adicionar o endereço do cliente à lista de notificações periódicas. Os *EA* entre o cliente e os servidores, devem atualizar/criar as suas entradas (C,G) em conformidade com o seu funcionamento.

F. Abandono do Grupo por parte dos Servidores

O abandono por parte de um servidor do grupo pode acontecer por dois motivos, por pedido do servidor ou por não efetuar a renovação da ligação.

Se o servidor decide abandonar o grupo, pode abandonar por completo (árvore partilhada e todas as centradas no cliente de que faça parte) ou então simplesmente uma árvore centrada no cliente. O primeiro caso está normalmente relacionado com o desejo de terminar todas as comunicações no sistema terminal, começando o EA por percorrer todas a entradas e remover a ligação de saída para o servidor. Caso o EA verifique que a lista de interfaces de saída ficou vazia, deve notificar o próximo nó do abandono, nó do caminho mais curto em relação ao destino final (*RP* para árvores partilhadas e cliente para árvores centradas). O segundo caso pode acontecer quando um servidor estiver com demasiadas comunicações e decide abandonar alguns clientes para prestar um melhor serviço. Esta consideração caberá sempre ao gestor dos servidores. A aplicação, utilizada pelo cliente, deverá enviar para o *RP*, periodicamente, pedidos de localização. O envio deste pedido tem como objetivo não deixar expirar o ramo de ligação ao servidor e permitir a novos servidores efetuarem as suas ligações.

A Figura 10 demonstra o pedido de abandono do SA 1 do grupo. Normalmente, quando um servidor abandona a rede, é necessário reiniciar o processo de descoberta de servidores. Supondo que se trata de uma comunicação importante, o tempo de resposta por parte dos servidores é minimizado.

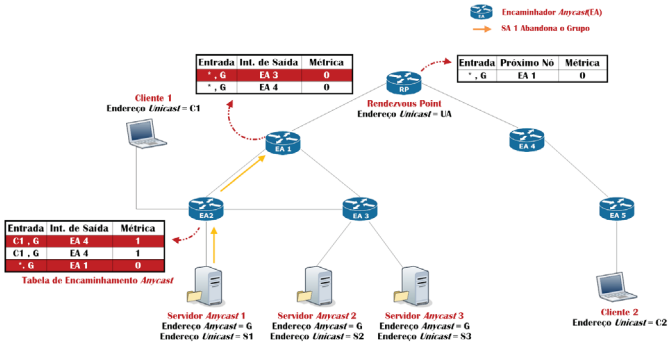


Figura 10: Abandono por parte do SA 1

A arquitetura proposta, proporciona uma funcionalidade que permite comutar automaticamente para outro servidor *anycast* sem voltar a refazer o serviço de localização. Tendo o servidor SA 1 abandonado o grupo, passa a ser o servidor com a segunda melhor métrica até então (SA 2), a responder aos pedidos do C1, como podemos verificar pela Figura 11.

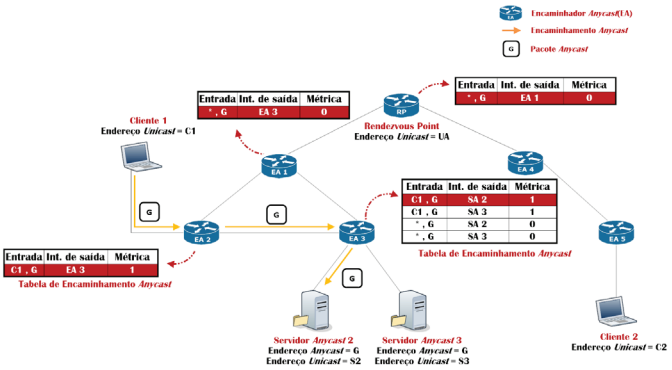


Figura 11: Resposta a pedido do C2 após o abandono do SA 1

G. Atualização da Métrica de um Cliente por parte dos Servidores

O balanceamento de carga é uma técnica consensual para quem quer implementar um serviço e manter uniforme a carga de trabalho dos seus sistemas terminais. Como o valor do atraso na resposta dos sistemas terminais é proporcional à carga que eles suportam no momento, é necessário repartir a carga or vários sistemas terminais, para evitar possíveis congestionamentos. A solução proposta engloba a possibilidade

de balanceamento de carga, através da atualização da métrica do servidor.

A Figura 12 demonstra o cenário de quando a métrica do SA 2 se degrada. Esta degradação pode passar por vários fatores, como por exemplo estar a receber demasiados pedidos. O SA 2 recalcula a sua métrica e comunica-o ao EA 3. Por sua vez, este verifica a sua tabela de encaminhamento e atualiza-a. Sendo que a menor métrica não sofreu nenhuma alteração (SA 3 tinha a mesma métrica), o EA 3 não necessita de informar o *EA 2*, o próximo encaminhador em direção ao cliente. Se houvesse uma alteração na menor métrica, o *EA 3* deveria reenviar a mensagem de atualização pelo caminho mais curto em direção do C1, sendo o processo repetido em todos os encaminhadores.

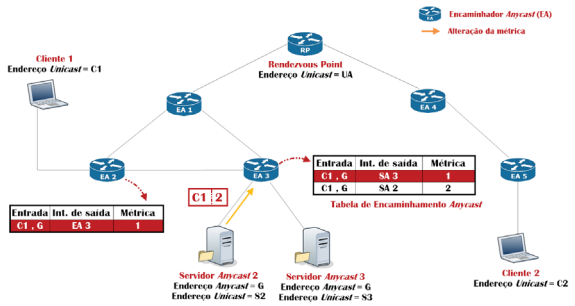


Figura 12: Alteração da métrica do SA 3

Perante a atualização da tabela de encaminhamento de EA 3, este agora passa a direcionar os pacotes provenientes do C1 para o SA 3. A Figura 11, demonstra o processo de encaminhamento após atualização da métrica, por parte do SA 2.

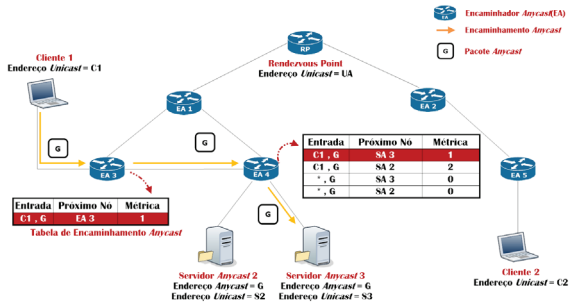


Figura 13: Comutação dos pedidos de C2 para SA 1

IV. Implementação e Teste do Sistema

A. Validação do Sistema

O sistema especificado na secção anterior, foi implementado com auxílio do simulador de rede *NS-2.35*[15]. O código implementado para este simulador é escrito em duas linguagens orientadas a objetos, o C++ para as tarefas amiudamente executadas (mais rápido) e o OTcl para o controlo de ações (mais flexível).

O código base utilizado na implementação do sistema é baseado numa implementação do *PIM-SM* existente[16], o que permitiu construir uma solução mais completa. Apesar de esta solução possuir um classificador e um agente para o protocolo *PIM-SM*, foi necessário adaptá-la ao sistema especificado. As classes classifier-pim.cc e classifier-pim.h são responsáveis pelo reenvio das mensagens entre nós, sendo escritas em C++. A implementação do agente do *PIM.tcl* ocorre na classe PIM.tcl, escrita em OTcl, sendo esta responsável pela troca de mensagens entre os nós, pela implementação do algoritmo e a construção das tabelas de encaminhamento.

O classificador implementado na abordagem base[16], reenvia os pacotes de acordo com os endereços de origem e o destino (grupo). Ao receber um pacote, o classificador verifica a sua tabela e encaminha os pacotes para todas as interfaces(nós) nele registados. Como no tráfego *anycast* é necessário escolher de acordo com os endereços de origem e destino mais a métrica, foi necessário acrescentar essa possibilidade. O método de encaminhamento *multicast* (para todos) foi conservado, sendo utilizado para as mensagens de descoberta de servidores. Foi necessário criar uma nova estrutura, com um par de campos interligados, métrica e respetivo nó. Assim, quando cada servidor se centrar no cliente, cria uma nova entrada, com a sua respetiva métrica. O encaminhamento passa a ser efetuado de acordo com o nó com melhor métrica, passando o cliente a comunicar com um único servidor. A Figura 14 mostra os campos constituintes do novo classificador *anycast*.

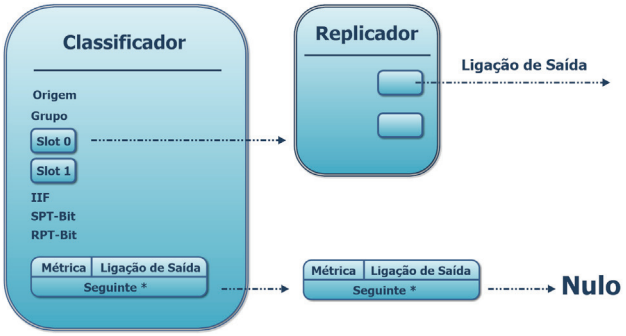


Figura 14: Composição do classificador *anycast*.

Os métodos tradicionais do *PIM-SM* para a manutenção das árvores (partilhada e centrada) tiveram que sofrer alterações, para que passassem a contemplar a métrica nas suas mensagens. A classe PIM.tcl sofreu várias alterações nas suas principais funções (*join-group*, *leave-group*, *recv-join*, entre outras), para passar a tratar o tráfego segundo o encaminhamento *anycast*. Foi ainda necessário incluir um novo tipo de mensagem de atualização, não disponível no protocolo base. Quando um nó recebe uma mensagem de atualização, atualiza a métrica do nó que originou a mensagem e verifica se o valor mínimo se alterou. Tendo este valor sido alterado, é enviado então uma mensagem para o nó seguinte, em direção ao destino, *RP* e/ou cliente(s). O processo descrito é repetido sucessivamente até que a métrica mínima não seja alterada ou chegue ao destino. De modo a proceder

a criação da nova mensagem de atualização, foi necessário modificar a classe `packet.h` e `mcast_ctrl.cc`. Para finalizar, foi necessário implementar duas aplicações, para os clientes e servidores, de modo a simular o comportamento especificado no capítulo anterior. A nova aplicação no cliente, efetua pedidos de descoberta de localização até receber a primeira mensagem de resposta de um servidor, passando de seguida a enviar pedidos para o servidor com a melhor métrica. A aplicação que corre nos servidores responde aos pedidos vindos dos clientes automaticamente, simulando o comportamento real de servidor, divulgando somente o seu endereço *anycast*.

O cenário utilizado para explicar o sistema na secção anterior, foi implementado para testar a solução, com o auxílio de uma aplicação de suporte do *NS*, o *NAM*. O *NAM* é uma ferramenta de animação, que permite criar topologias e visualizar a animação dos pacotes durante o seu encaminhamento. O primeiro passo foi criar a topologia e de seguida os nós, que seriam clientes, servidores ou simples encaminhadores. Finalizada a implementação, é altura de dar indicações aos nós para efetuarem a sua ligação, abandono ou atualização no instante de tempo pretendido. A Figura 15 demonstra o encaminhamento dos pacotes do `C1` para o `SA 1`, o servidor com melhor métrica.

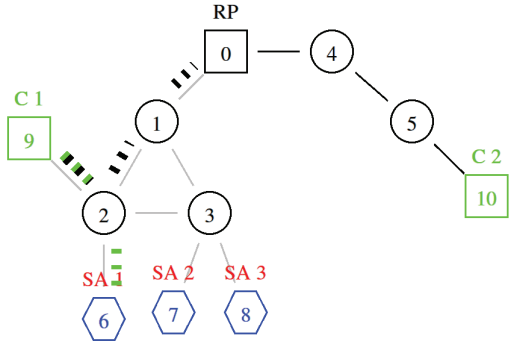


Figura 15: Resposta do pedido do `C1` pelo `SA 1`.

A falha (ou abandono) durante a comunicação de um servidor é sempre um problema para a rede, sendo muitas vezes necessário voltar a iniciar a comunicação. A implementação efetuada permite a fácil comutação de um servidor para outro, de modo a minimizar o tempo perdido. A Figura 16 capta o momento em que o `SA 1` deixa de poder atender o `C1`, podendo-se verificar a imediata comutação para o `SA 2`.

As ligações ao longo da *Internet*, estão constantemente a alterar a sua qualidade. O protocolo implementado prevê essa situação, podendo, a qualquer momento, um servidor proceder à atualização da sua métrica em relação a qualquer nó. Quando o `SA 2`, no cenário anterior, vê que a sua ligação deteriorou-se, procede a atualização da sua métrica. Na Figura 17 é possível verificar o encaminhamento após a atualização por parte do `SA 2`, passando o tráfego a ser encaminhado o `SA 3`.

O cenário elaborado permite testar o correto comportamento do *TAP* implementado e exequibilidade da proposta.

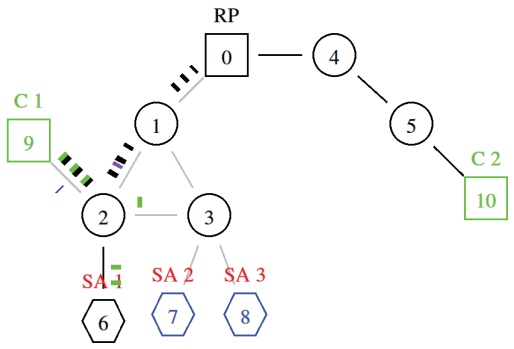


Figura 16: Resposta do pedido do `C1` comuta do `SA 1` para o `SA 2`.

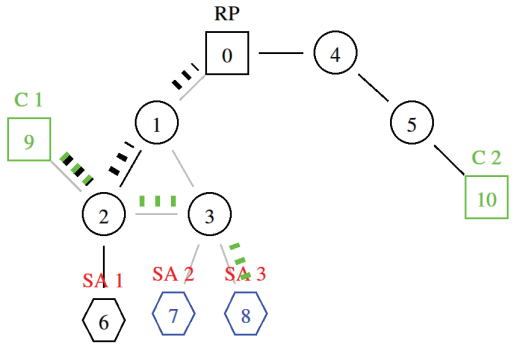


Figura 17: Resposta do pedido do `C1` comuta do `SA 2` para o `SA 3`.

B. Análise dos resultados em relação ao PIA

O *PIA*[7] foi o protocolo escolhido para comparar com o *TAP*, principalmente por este também utilizar uma abordagem baseada no *PIM-SM*. Embora o protocolo tenha sido implementado com sucesso, não houve tempo para desenvolver uma aplicação que tivesse um comportamento como especificado[7]. Esta limitação não permitiu efetuar grandes testes entre os dois protocolos, ficando a faltar a realização de um melhor conjunto de testes no futuro.

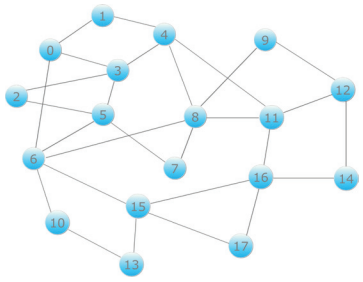


Figura 18: Topologia implementada com 18 nós.

No entanto, foi possível realizar uma comparação entre os dois protocolos, medindo a distância do servidor escolhido até ao cliente. A topologia presente na Figura 18 foi utilizada como cenário para os testes, sendo posicionados aleatoria-

mente nesses nós um *RP*, um cliente e quatro servidores. O cenário foi simulado 100 vezes para os dois protocolos, *TAP* e *PIA*, sendo o gráfico presente na Figura 19 o seu resultado.

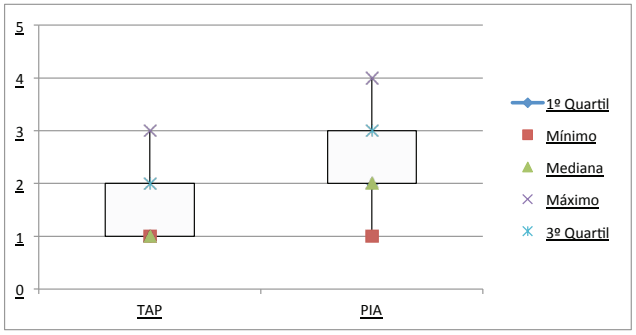


Figura 19: Distância entre o cliente e o servidor escolhido para os dois protocolos (*TAP* e o *PIA*).

Ao observar o gráfico na Figura 19, é possível afirmar que o protocolo *PIA* nem sempre escolhe o servidor mais próximo da localização. Uma importante conclusão a retirar das simulações é que o valor das localizações centrais das amostras é menor no *TAP*, o que atesta a eficiência do novo protocolo, em relação ao *PIA*. Apesar de o *TAP* ser melhor quando as métricas levam em conta a posição dos servidores, é necessário, para trabalho futuro, realizar um conjunto de testes exaustivos ao *TAP*, obtendo novos resultados em relação ao *PIA*.

V. Conclusão e trabalho futuro

O encaminhamento de tráfego *anycast* em redes *IPv6* ainda não é uma realidade, principalmente devido ao facto de não existir um protocolo normalizado que retire partido das suas potencialidades. Este artigo aborda uma proposta para um novo protocolo, tendo o *PIM-SM* como base, que pretende solucionar esse problema.

As principal vantagem deste novo protocolo é o facto de o cliente comunicar realmente com o servidor mais próximo (com melhor métrica) do seu local. A segurança é outra das suas vantagens, pois os clientes nunca conhecem o endereço do servidor, comunicando sempre para o grupo. A última grande vantagem é a grande disponibilidade da rede. No caso da falha de um servidor, os pacotes são automaticamente comutados para outro servidor. O grande entrave à aceitação desta proposta prende-se com o facto de ser necessário um endereçamento distinguível do *unicast*, obrigando a uma alteração da norma[2].

O trabalho ainda se encontra a decorrer, estando o protótipo a ser avaliado. O *TAP* está a ser testado em diferentes tipos de topologias de rede, para obter dados mais consistentes em relação ao seu desempenho. O trabalho futuro passa, claramente, por realizar mais testes ao *TAP* e comparar com a abordagem referida no trabalhado relacionado[7].

Agradecimentos

Este trabalho é financiado por Fundos FEDER através do Programa Operacional Fatores de Competitividade – COMPETE e por Fundos Nacionais através da FCT – Fundação para a Ciência e Tecnologia no âmbito do Projeto: FCOMP-01-0124-FEDER-022674

Referências

[1] C. Partridge, T. Mendez, and W. Milliken, “Host Anycasting Service.” RFC 1546 (Informational), Nov. 1993.
[2] R. Hinden and S. Deering, “IP Version 6 Addressing Architecture.” RFC 1884 (Historic), Dec. 1995. Obsoleted by RFC 2373.
[3] D. Katabi and J. Wroclawski, “A framework for scalable global ip-anycast (gia),” *SIGCOMM Comput. Commun. Rev.*, vol. 30, pp. 3–15, Aug. 2000.
[4] H. Ballani and P. Francis, “Towards a global ip anycast service,” *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 301–312, Aug. 2005.
[5] T. Stevens, M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, and P. Demeester, “Astas: Architecture for scalable and transparent anycast services,” *J. Commun. Netw.*, vol. 9, no. 4, pp. 1229–2370, 2007.
[6] S. Doi, S. Ata, H. Kitamura, and M. Murata, “Ipv6 anycast for simple and effective service-oriented communications,” *IEEE Communications Magazine*, vol. 42, pp. 163–171, 2004.
[7] S. Matsunaga, S. Ata, H. Kitamura, and M. Murata, “Design and Implementation of IPv6 Anycast Routing Protocol: PIA-SM,” in *Advanced Information Networking and Applications*, vol. 2, pp. 839–844, 2005.
[8] A. Sulaiman, B. Ali, S. Khatun, and G. Kurup, “An enhanced ipv6 anycast routing protocol using protocol independent multicast-sparse mode (pim-sm),” in *Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC 2007. IEEE International Conference on*, pp. 588 –593, may 2007.
[9] k. claffy, “Border Gateway Protocol (BGP) and Traceroute Data Workshop Report,” tech. rep., Cooperative Association for Internet Data Analysis (CAIDA), Oct 2011.
[10] D. Waitzman, C. Partridge, and S. Deering, “Distance Vector Multicast Routing Protocol.” RFC 1075 (Experimental), Nov. 1988.
[11] J. Moy, “MOSPF: Analysis and Experience.” RFC 1585 (Informational), Mar. 1994.
[12] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised).” RFC 4601 (Proposed Standard), Aug. 2006. Updated by RFCs 5059, 5796, 6226.
[13] F. Font and D. Mlynek, “Choosing the set of rendezvous points in shared trees minimizing traffic concentration,” in *Communications, 2003. ICC ’03. IEEE International Conference on*, vol. 3, pp. 1526 – 1530 vol.3, may 2003.
[14] D. Katabi, “The use of ip-anycast for building efficient multicast trees,” in *Global Telecommunications Conference, 1999. GLOBECOM ’99*, vol. 3, pp. 1679 –1688 vol.3, 1999.
[15] S. Mccanne, S. Floyd, and K. Fall, “ns2 (network simulator 2).” http://www-nrg.ee.lbl.gov/ns/.
[16] A. S. e. V. F. António Costa, Maria João Nicolau, “Implementação e Teste do PIM-SM no Network Simulator,” tech. rep., Conferência sobre Redes de Computadores (CRC2002), Faro, Portugal, Sep 26-27 2002.

Detection of WPS Attacks Through Multiscale Analysis

Ivo Petiz*, Eduardo Rocha*, Paulo Salvador*†, António Nogueira*†
*Instituto de Telecomunicacoes, Aveiro
†DETI, Universidade de Aveiro
{petiz,eduardorocha,salvador,nogueira}@ua.pt

***Abstract*—The wide spread adoption of 802.11 networks as the solution for providing an efficient network coverage with high data-rates raised several security concerns. In a first stage, WEP was used for protecting user's wireless networks from intrusions. Such intrusions' purposes could be simple free Internet accesses or more complex attacks to access confidential information. However, due to multiple technical flaws this approach was not sufficient which lead to the emergence of WPA and WPA2 technologies. WPA and WPA2 allow more secure networks but require more complicated configuration tasks.**

With the objective of creating a simple configuration interface, the Wi-Fi Alliance came up with a simple configuration approach: the Wi-Fi Protected Setup (WPS). WPS is present in major vendors products, providing a much easier configuration setup but a less efficient security environment. This less secure implementation is vulnerable to brute force attacks, that can be quick to execute, with little complexity and difficult to detect. After cracking the WPS, attackers can access to WPA/WPA2 wireless passphrase and consequently, illicitly connect to users' wireless networks.

Accessing and analyzing the content of the wireless frames is limited by technical requirements and legal constrains. Therefore, this paper presents a method to detect attacks on WPA routers with Wi-Fi Protected Setup based only on the amount of traffic generated. We propose a monitoring station which exclusively analyzes traffic flows from the router. By monitoring the traffic and using a multiscale analysis we are able to accurately identify this type of intrusion attempt over other traffic.

I. Introduction

With the increasing demand for Internet connectivity, several approaches were adopted for enabling a simple and efficient Internet access. Currently almost all (Internet Service Providers) ISPs provide wireless routers for homes and small office (SOHO) environments to their clients. Moreover, all network equipments manufacturers offer a wide-range of wireless routers. Modern wireless routers facilitate the setup of domestic wireless networks covering the users' home and office environment. However, the range of the wireless networks extend much further than the users' environment. Therefore, wireless routers physically allow the wireless access of unauthorized entities to the users data traffic, while it becomes difficult for users to know who is using (or watching) their connections. To address this issue, in the last years, wireless security became more complex in order to prevent an abusive use by undesirable users and attackers. One of the first proposed solutions was WEP security [1], which proved not so efficient [2] and was replaced by more efficient protocols, like WPA and WPA2 [3].

More secure protocols, such as WPA and WPA2, need more complex configurations which, for the common Internet user, could be a big problem, what sometimes led users to deactivate wireless security in order to avoid complicated setups. Wi-Fi CERTIFIED Wi-Fi Protected Setup (WPS) [4] was then created in order to simplify wireless setup, providing a PIN with 8 digits that could be introduced in user's computer and the connection would be established. Despite this simple method facilitates less expert users brings big security issues. WPS helps assure consumers that the Wi-Fi devices they purchase can be easily configured with security features enabled on their Wi-Fi networks, and that they can add new Wi-Fi Protected Setup devices to established networks with greater ease.

Wi-Fi Protected Setup is a certification program designed to support Wi-Fi CERTIFIED 802.11 products including consumer electronics and phones, as well as computers and routers. It applies to 802.11 devices for home and small office, including those that communicate through 802.11a/b/g/n, as well as multiple-band devices and those designed to operate Wi-Fi Direct™ features. The Wi-Fi Alliance [5],certified the first products with Wi-Fi Protected Setup in January of 2007. Since then, new features have been introduced to make the setup and configuration of security features even easier to use. In the past year alone, more than 1,000 products, including home access points, gateways and handsets have passed the testing necessary to be identified as Wi-Fi CERTIFIED Wi-Fi Protected Setup [4].

A very simple and easy to use software, which allows a user with not so much knowledge about computers and networks to get access to wireless networks, was recently released. This software, named Reaver [6], is a brute force attack program that exploits the WPS vulnerability by allowing users to subsequently send PINs trying to hit the right one. This PIN consists of a 7 digit number with a eighth digit corresponding to the parity number. With a minimum of 3 seconds per attempt it is possible to attacker to gain access to the WPA/WPA2 phrase pass in a few hours, much faster than traditional brute force and dictionary attacks, not being necessary capture any traffic from WLAN users connected. This type of attack is hard to detect and difficult to prevent. Event if the owner of an attacked wireless network changes the WPA/WPA2 phrase pass, the attacker can get the new phrase pass once again if the PIN from WPS continues the same. If the PIN has been changed, the attacker can always launch a

new brute-force attack in order to crack again the PIN.

In order to efficiently detect such attacks is necessary to access and analyze the content of the wireless frames. Capturing, decoding and analyzing all the wireless frames requires the usage of equipment with high processing capabilities. Moreover, the analysis of any layer of the frames data/headers is often limited by legal constrains. Therefore, in this paper we propose a method to detect the presented brute force WPS attack, based on the analysis of low level statistics (frame count) of the traffic sent by an attacked router. A multiscale decomposition and analysis of the collected traffic is performed and the obtained decomposition coefficients are then compared with the ones of regular legit traffic and other network attacks.

II. WPS Flaw and Attack

Wi-Fi Protected Setup presents two different methods to connect user's device to an access point, which is the Push Button Configuration method and the PIN method. The Push Button Configuration method consists in push a button on both devices, the button could be physical or virtual. A device has 2 minutes to authenticate in wireless router or will be presented a timeout and the connection will fail. In this 2 minutes any device could connect to wireless router, a desirable or undesirable one. The PIN method could be used by two different forms, by introducing a PIN from device in wireless router's interface or a PIN from the wireless router in the device's interface. PIN could be written in the device or wireless router or is possible to be a generated dynamically if requested.

In the case of using PIN method, by introducing the wireless router's PIN in the user's device, it is possible, to most of the vendors' wireless domestic routers, to make various attempts before the MAC address of the attacker's device becomes blocked. More concerning is the fact some wireless routers do not even block these devices, making possible a continuous brute force attack without any restrictions and of the wireless routers using WPS has the PIN feature enabled by default, without the possibility of shutting down. This flaw is even bigger looking at the PIN structure, shown in Figure 1, since the PIN consists only of a 8 digit number where the eighth digit is the check-sum. The number of attempts to find the PIN should be at most 10^8, but the attack can be optimized because the authentication proceeds by verifying the first 4 digits and then, if the first 4 digits are correct, there are only the last 4 digits left to discover. Therefore, in the worst case we need to try 10^4 numbers to find the first part of the PIN and then another 10^3 attempts to find the last part, once the last digit is the check-sum and can be calculated by attacker, what will give a total of 11000 attempts in the worst case to find the correct wireless router's PIN and get the WPA/WPA2 phrase pass.

III. Testbed Configuration

The lab hardware consists of two laptops named as machine 1 and machine 2 and a domestic wireless router as seen in

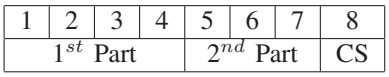


Figure 1. PIN Description

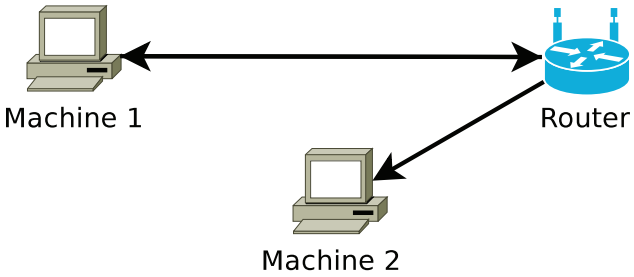


Figure 2. Lab configuration

Figure 2, working as a wireless router. Machine 1 was used as the attacker and machine 2 was the responsible for capturing the traffic sent by the router. The tested router was a Thomson, model TG784, is Wi-Fi Certified and 802.11b/g operating by default with WPS. Both machine 1 and machine 2 are laptops running Linux Ubuntu 11.10 and equipped with an Atheros wireless card, in order to make use of monitor mode, what is necessary to make the attack and capture the traffic.
To exploit this flaw is used Reaver 1.4 [6], after configuring the interface in monitor mode, using airmon-ng, from aircrack-ng suite software. To capture the traffic from router, machine 2 was configured in monitor mode to, running Tshark and configured in promiscuous mode.

Several attacks were launched using the several Reaver options, ranging interval between pin attempts and delay after a certain number of attempts, in order to simulate different types of WPS restrictions, as happens with the different kinds of routers, from the different vendors. It was used only traffic from the first part of attack, when an attacker tries to discover the first 4 digits of the PIN, since an continuously running system will always detect the beginning of the attack.

IV. Data Capture and Analysis

The collected traffic consisted of the traffic sent by the wireless router, in order to enable the detection of potential WPS attacks based on the analysis of the router responses to these attacks.

For both machines, WLAN cards were configured in monitor mode with the "airmon-ng start wlan0" command. For each Reaver attack configuration, the minimum capture duration was 6 hours, made using Wireshark in promiscuous mode, filtering the traffic to obtain just packets from router to machine 2. After the tests all captures were divided in 5 minutes files, size chosen by us trying to find the ideal duration, in order to predict attacks, with a maximum accuracy near real time.

Several captures, with different parameters, were performed in order to simulate the different responses from routers of

various vendors. For example, in order to prevent these kind of attacks, some routers restrict the number of attempts the attacker could execute before blocking the MAC address of the attacker. This can be avoided by using a delay after a certain number of PIN attempts.
- Regular attack, no alterations to the Reaver's default configuration.
- "–delay=2", set the delay between attempts in 2 seconds.
- "–delay=5", set the delay between attempts in 5 seconds.
- "–recurring-delay=$\langle 5 : 120 \rangle$", wait 120 seconds after 5 attempts.
- "–recurring-delay=$\langle 10 : 60 \rangle$", wait 60 seconds after 10 attempts.

To differentiate recurring-delay option from simple delay option it will be called lag to simple delay.
All captures were made in the first half of the attack period. A capture made in the second part of attack will necessarily have more packets exchanges and will present slightly different results.

We analyzed the number of captured bytes and packets per sampling interval. This interval was set 0.1 seconds and all collected flows were analyzed over 5 minutes intervals.

In order to compare the WPS attack with regular Internet traffic was also made traffic captures from some of most used Internet applications like Facebook[7], Gmail[8], Youtube[9] and online news. These captures follow the same method as the WPS. Captures were made from machine 2 and only low level statistics were collected.

V. Multi-Scale Analysis based on Wavelet Scalograms

In this section we present our traffic analysis approach which is based on a wavelet decomposition through the Continuous Wavelet Transform (CWT). In this manner, we can analyze any process in both time and frequency domains. Therefore, this tool is widely used in many different fields such as image analysis, data compression and, more recently, in traffic analysis. The CWT of a process $x(t)$ can be defined as [10]:

$$\Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{+\infty}^{-\infty} x(t)\psi^*(\frac{t-\tau}{s})dt \qquad (1)$$

where $*$ denotes the complex conjugation, $\frac{1}{\sqrt{|s|}}$ is used as an energy preservation factor, $\psi(t)$ is the *mother wavelet* while τ and s are the translation and scale parameters, respectively. The first parameter is used for shifting the mother wavelet in time, while the second parameter controls the width of the window analysis and, consequently, the frequency that is being analyzed. By varying these parameters, a multi-scale analysis of the entire captured process can be performed, providing a description of the different frequency components present in the decomposed process together with the time-intervals where each of those components is located. A Wavelet Scalogram can be defined as the normalized energy $\hat{E}_x(\tau, s)$ over all possible translations (set $\mathbf{T}$) in all analyzed scales (set $\mathbf{S}$), and is computed as:

$$\hat{E}_x(\tau, s) = 100 \frac{\left|\Psi_x^{\psi}(\tau, s)\right|^2}{\sum_{\tau' \in \mathbf{T}} \sum_{s' \in \mathbf{S}} \left|\Psi_x^{\psi}(\tau', s')\right|^2} \qquad (2)$$

The volume bounded by the surface of the scalogram is the mean square value of the process. The analysis of these scalograms enables the discovery of the different frequency components, for each scale (frequency) of analysis. For instance, the existence of a peak in the scalogram at a low frequency indicates the existence of a low-frequency component in the analyzed time-series while a peak in the scalogram at a high-frequency corresponds to an existing high-frequency component. In addition, assuming that the process $x(t)$ is stationary over time, several statistical information, such as the standard deviation, can be obtained:

$$\sigma_{x,s} = \sqrt{\frac{1}{|\mathbf{T}|} \sum_{\tau \in \mathbf{T}} (\hat{E}_x(\tau, s) - \mu_{x,s}), \forall s \in \mathbf{S}} \qquad (3)$$

where $\mu_{x,s} = \frac{1}{|\mathbf{T}|} \sum_{\tau \in \mathbf{T}} \hat{E}_x(\tau, s)$, and $|\mathbf{T}|$ denotes the cardinality of set $\mathbf{T}$.

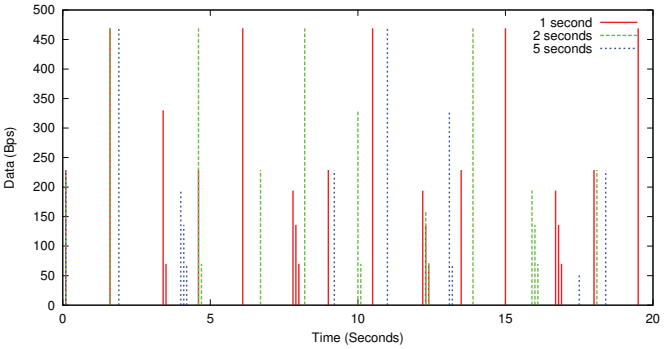


Figure 3. Data per second from 3 different delays configurations

VI. Results

In order to validate the proposed classification approach, several traffic measurements were performed as described in section III. The analyzed traffic was collected by using a promiscuous monitoring probe that captures all traffic sent from the wireless router of a 802.11 wireless network that was assembled at our networks laboratory. Since our monitoring probe does not connect to the wireless network, it does not access layer 3 traffic information. Consequently, the layer 2 metrics considered for analysis were the number of captured bytes per sampling interval (0.1 seconds). The same method was also used in Internet applications captures, in order to

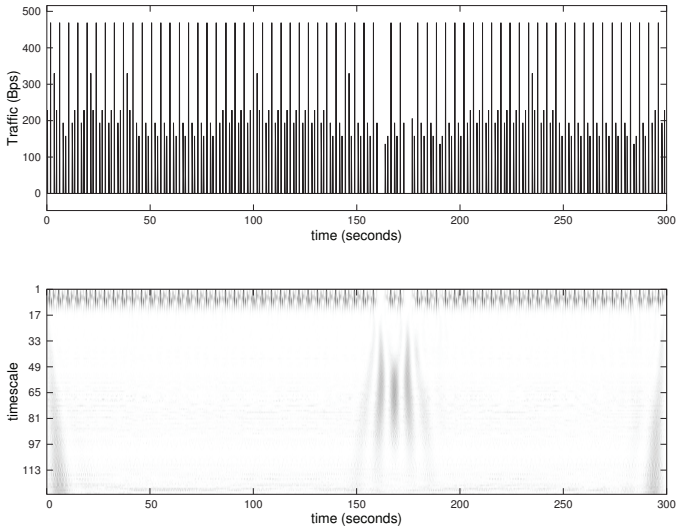


Figure 4. 5 minute capture bytes per 0.1 second scalogram

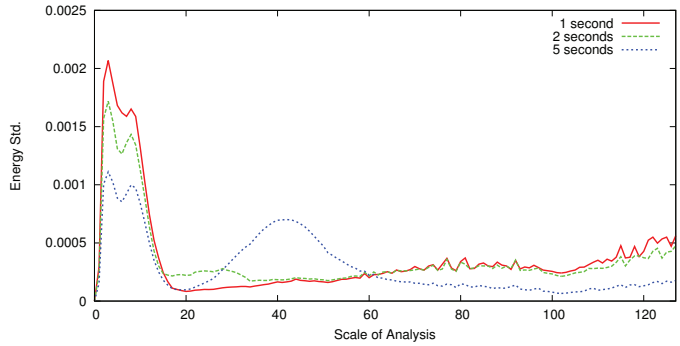


Figure 5. Comparison of 3 different delay intervals

compare with the WPS attack. The captures were made in Machine 2, only accessing layer 2 traffic.

In Figure 3 is possible to distinguish from the different PIN attempts' delays. All PIN attempts' data flows start at O seconds and it is possible to identify the different lags between PIN attempts. At 20 seconds, capture with a 5 seconds delay only had made 2 complete PIN attempts whereas the 1 second delay attempt has completed 5 cycles and 2 seconds attempts made almost 5 attempts.

The amount of traffic per 0.1 seconds for a 5 minutes capture and its scalogram is shown in Figure 4. It is possible to note a rhythm almost constant, except near 160 seconds, where the rhythm is broken for 15 seconds, and resumes to the initial rhythm. In Figures 5 and 6 are presented the comparisons between different attacks configuration's scenarios. In Figure 5 we compare the default configuration with an 1 second delay, 2 seconds delay and 5 seconds delay between attempts. Despite different configurations the 3 options show similar curves that can be identified using a multi-scale analysis based on wavelet scalograms. Figure 6 presents another configuration option where an additional delay is introduced after a ser of attempts (e.g. 10 attempts). When compared to the default configuration

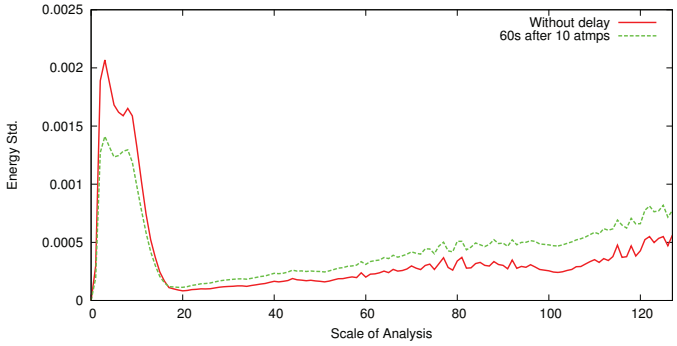


Figure 6. Comparison between different delay intervals

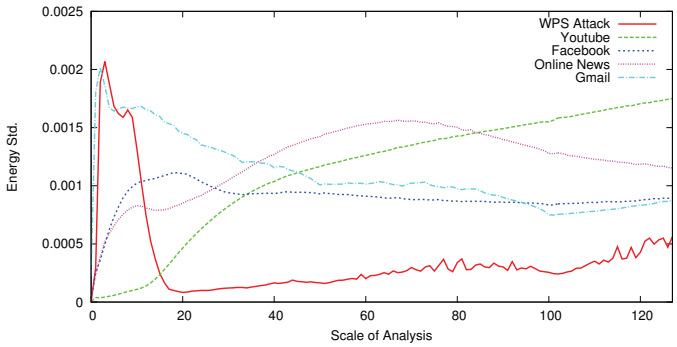


Figure 7. Comparison between WPS attack and some Internet services

(no additional delay) the analysis reveals a very similar curve. The WPS attack, when analyzed using a multi-scale analysis based on wavelet scalograms, shows a very characteristic curve, quite different from other Internet applications and services like Facebook, Youtube, Gmail and Online News, as shown in Figure 7, what makes it possible to identify using this methodology.

VII. Conclusion

Wireless networks are now widely used for providing an efficient and easy to use Internet access. Internet users use such networks to access important on-line services such as home banking, on-line shopping and to transfer important data. Users trust in these wireless connection by using secure protocols like WPA or WPA2 that provide a high level of security. Flaws such as the one detected in WPS feature compromise the security of wireless networks and compromise the security and confidentiality of the user's on-line communications and transactions.

In this paper, we proposed a method for the detection of malicious attacks to domestic routers based on Wi-Fi Protected Setup flaw. Making exclusively use of layer 2 traffic statistics and resorting to Continuous Wavelet Transform, it is possible to identify an attack. Indeed, the frequency components generated by the mentioned attacks can be easily differentiated from flows generated by other legit Internet applications like email, video streaming or social networks.

REFERENCES

[1] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-1997*, 1997.

[2] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *Security and Privacy, 2006 IEEE Symposium on*, may 2006, pp. 15 pp. –400.

[3] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements," *IEEE Std 802.11i-2004*, 2004.

[4] "Wi-fi protected setup white paper," Tech. Rep., January 2007. [Online]. Available: https://www.wi-fi.org/knowledge-center/white-papers/wi-fi-certified-wi-fi-protected-setup%E2%84%A2-easing-user-experience-home-a-0

[5] (2012, September) Wi-fi alliance. [Online]. Available: http://www.wi-fi.org/

[6] (2012, September) Reaver WPS - Brute force attack against wifi protected setup. [Online]. Available: http://code.google.com/p/reaver-wps/

[7] (2012, September) Facebook. [Online]. Available: https://www.facebook.com

[8] (2012, September) Gmail. [Online]. Available: https://mail.google.com

[9] (2012, September) Youtube. [Online]. Available: http://www.youtube.com

[10] J. Slavic, I. Simonovski, and M. Boltezar, "Damping identification using a continuous wavelet transform: application to real data," *Journal of Sound and Vibration*, vol. 262, no. 2, pp. 291 – 307, 2003.

Multipass: Autenticação Mútua em Cenários Heterogéneos

Rui Ferreira[1], André Tomás[1], Pedro Estima[1], Rui Aguiar[1], Ricardo Azevedo[2]

[1]Instituto de Telecomunicações
[2]PT Inovação

***Resumo*— A utilização de dispositivos móveis como dispositivos computacionais de uso geral é uma tendência crescente, com especial enfase para o mercado dos Smartphones. Esta tendência levou ao aparecimento de serviços que exploram o uso de dispositivos móveis como mecanismos de autenticação do utilizador, ou mesmo como mecanismo de pagamento de transações.**

Este artigo descreve a arquitetura e implementação que foram desenvolvidas para suportar os cenários de autenticação do projeto Multipass2 que visa explorar o uso de dispositivos móveis como mecanismos de autenticação do utilizador, num cenário multi-canal. Um dos objetivos primordiais é a garantia de mecanismos de autenticação mútua entre o utilizador e o serviço, num cenário heterogéneo em que as tecnologias de rede IP (Wifi ou 3G) se misturam com tecnologias de comunicação em proximidade (Bluetooth, NFC). O artigo apresenta ainda a implementação desenvolvida como prova de conceito e os resultados obtidos no decorrer do projeto.

***Palavras Chave*— Segurança, Gestão de Identidades, Android, Autenticação**

I. INTRODUÇÃO

OS dispositivos móveis têm vindo a desempenhar um papel cada vez mais preponderante como ferramentas do dia-a-dia cujo propósito vai muito para além de simples dispositivos de comunicação. Esta utilização tem sido explorada pelos Smartphones, que dispõem de recursos computacionais consideráveis, várias tecnologias de comunicação (3G, Wifi, Bluetooth) e que executam aplicações que se afastam dos tradicionais serviços providenciados sobre telefones móveis.

Esta tendência ganha novos contornos quando se considera que estes dispositivos podem substituir completamente as carteiras e chaves tradicionais no bolso do utilizador. Um Smartphone pode ser usado como sistema de autenticação do seu portador para abrir portas ou outros sistemas de controlo de acesso, ou ainda como sistema de pagamento de serviços, através de uma carteira digital, como é o caso do Google Wallet.

O projeto Multipass tem como principal objetivo explorar o uso de dispositivos móveis como mecanismo de autenticação do utilizador em ambientes quotidianos com diferentes tipos de serviços e tecnologias de acesso; sendo que se coloca especial enfase nos seguintes quatro aspetos:

1) Assegurar a autenticação mútua, confidencialidade e integridade entre o dispositivo móvel e os serviços com os quais comunica.
2) Operar tanto com base em mecanismos de autenticação com chave pública/privada, assim como com sistemas de gestão de identidades federadas como OpenID[1] e SAML[2].
3) Suportar mecanismos de simplificação de autenticação que suportem autenticação multi-canal.
4) Funcionar em ambientes heterogéneos, recorrendo tanto a tecnologias de comunicação sobre IP (Wifi ou 3G) como comunicação em proximidade (NFC e Bluetooth).

O projeto foca-se em particular em dois cenários distintos. O primeiro cenário, focado em ambientes de domótica ou Internet of Things (IoT) permite que um dispositivo móvel se autentique e interaja de forma segura com outros dispositivos nas suas imediações, e.g. para abrir a porta de uma casa, ou para comandar remotamente outros dispositivos. O segundo cenário envolve a utilização de um terminal público (quiosque), em que o dispositivo móvel é utilizado para autenticar um utilizador junto de um quiosque e autorizar o quiosque a agir em seu nome junto de outros serviços, recorrendo para isso a sistemas de gestão de identidade que garanta a autenticidade de todas as partes envolvidas.

No âmbito do projeto Multipass foi implementado um protótipo para instanciar os cenários do projeto. Este protótipo é composto por três componentes: i) uma aplicação Android que permite ao utilizador fazer a gestão dos tokens de autenticação que transporta consigo e interagir com serviços nas imediações; ii) serviços para gerar e consumir tokens baseados em chave pública/privada; iii) e um mecanismo de autenticação que permite a sistemas de Identity Management (IdM) recorrer a mecanismos multi-canal, mantendo as mensagens de autenticação num canal separado e recorrendo a mecanismos do operador para autenticar o dispositivo móvel.

Este artigo está estruturado da seguinte forma. Na secção II é apresentado o estado de arte relevante, na secção III são descritos os dois cenários que se pretende instanciar. A secção IV apresenta a arquitetura que foi criada para suportar ambos os cenários e na secção V é descrito o protótipo desenvolvido e são discutidos os desafios associados à sua implementação.

Por fim na secção VI são apresentadas as conclusões.

II. Estado da Arte

O uso de telemóveis como mecanismos de autenticação antecede o aparecimento dos Smartphones. São bastante comuns os sistemas de bilhetes eletrónicos enviados por SMS, como é o caso de [3], em que um identificador único é enviado num SMS, armazenado no telemóvel e apresentado no ato de consumo onde é verificado contra uma lista de identificadores válidos. Outros serviços recorrem a soluções semelhantes, diferindo na tecnologia de transporte e formato de armazenamento, por exemplo em [4] é descrito uma solução em que são usados códigos de barras para registar e apresentar os bilhetes e para permitir a leitura automática dos mesmos.

No entanto este tipo de sistemas não protege adequadamente o utilizador/consumidor ,porque o processo de autenticação não garante nem que o dispositivo adquiriu de facto aquele identificador, nem que o serviço é de facto fidedigno. Um ataque contra este tipo de sistemas consistiria em extrair o código único do telemóvel, ou em efetuar um ataque de man-in-middle. No sentido de melhorar a segurança deste tipo de sistemas alguns serviços implementaram mecanismos que autenticam o dispositivo móvel com base em mecanismos criptográficos. Por exemplo [5] fornece um serviço para entrada sem check-in em hotéis, que usa tokens únicos transmitidos na forma de sinal sonoro para abrir fechaduras. Mas mesmo neste caso a autenticação não é mútua, é apenas o terminal móvel, e não o serviço, que é autenticado.

O aparecimento dos Smartphones promove o uso de aplicações especializadas para cada tarefa, pelo que a disponibilidade de ferramentas no dispositivo não é uma limitação. Se considerarmos ainda que estes dispositivos possuem mecanismos de comunicação sem fios como Wifi e Bluetooth, ficam preenchidos os requisitos para implementar protocolos de autenticação nestes dispositivos.

Esforços no sentido de explorar o uso de Smartphones para este propósito têm sido particularmente vincados na área dos pagamentos eletrónicos, onde as iniciativas[6] da Google Wallet e Isis começam a promover este tipo de utilização. Estas iniciativas têm ainda resultado em alterações aos novos dispositivos móveis no mercado (pelo menos dos financiados pelo Google), promovendo a adoção de Near Field Communication (NFC) como mecanismo preferencial de comunicação de curto alcance para este tipo de transações, e adicionando chips para armazenamento seguro (semelhante ao Trusted Platform Module(TPM) ou Smartcard). O operador de telecomunicações desempenha um papel fundamental, que pode ir muito além do simples encaminhamento de bits. Nos tradicionais cenários, em que o dispositivo já tem um SIM Card, do qual o operador é dono, a autenticação pode ser baseada nesse elemento seguro e delegada para o Operador. Esta funcionalidade garante os níveis desejados de segurança, a autenticação é baseada em certificados e não é *phishable*. As garantias são bastante elevadas.

No âmbito do projecto Mutipass, pretende-se preencher algumas das lacunas identificadas anteriormente, providenciando um misto destes mecanismos, estabelecendo como requisito mínimo a necessidade de autenticação mútua – independentemente da tecnologia de transporte em utilização – e suportando tanto soluções análogas às de bilhetes eletrónicos como soluções integradas suportadas pelo operador.

III. Cenários

O projeto Multipass considera dois cenários distintos para efeitos de autenticação do utilizador. O primeiro cenário foca-se em ambientes em que o utilizador carrega no seu telemóvel tokens para se autenticar com dispositivos com os quais poderá interagir; simultaneamente esses tokens são capazes de autenticar também os dispositivos, garantindo assim que está a interagir com serviços que pretende. O segundo cenário implica uma terceira entidade (IdP – Identity Provider) na qual ambos os intervenientes confiam para autenticação das partes envolvidas com base em conhecimento prévio do utilizador e serviços.

A. Domótica e Ambientes Inteligentes

Em Ambientes Inteligentes, tipicamente demonstrados em cenários de domótica, o utilizador interage com múltiplos dispositivos em seu redor. Estas interações são normalmente de curta duração, com baixa complexidade para o utilizador, e com consequências diretas no contexto atual onde o utilizador se encontra.

Em casa são exemplos deste tipo de cenários o uso do telemóvel para a abertura de portas, controlar pequenos dispositivos ou iniciar tarefas comuns no ambiente envolvente, por exemplo, controlar pequenos eletrodomésticos como sistemas de ar-condicionado. Em ambientes empresariais, haverá cenários em que poderá ser usado para marcar presenças, fazer controlo de acesso a espaços, desbloquear o terminal de trabalho ou ativar a máquina do café – podendo depois interagir com sistemas de pagamentos.

Internamente este tipo de cenário pode estar circunscrito a apenas um dispositivo, ou integrado com Gateways especializadas em integração de serviços de domótica. A complexidade é invisível aos olhos do utilizador, facilitando a adoção. O que se pretende neste contexto é fortalecer estas interações, assegurando que o utilizador é sempre autenticado e está a interagir com dispositivos que já conhece.

B. Terminais Públicos

Os sistemas de gestão de identidades, têm como objetivo unificar a identidade do utilizador de forma reduzir o esforço de gestão de credenciais por parte do utilizador, ou orquestrar o uso de múltiplos serviços pelo mesmo utilizador. Estes sistemas permitem ao utilizador e serviços delegar o processo de autenticação noutra entidade. A forte ligação entre telemóvel e os sistemas IdM já fornecidos pelo operador potencia o uso do dispositivo como chave de acesso a sistemas fora da área de influência do utilizador – mas ligados de alguma forma ao operador.

Neste tipo de cenários o dispositivo móvel funciona como mecanismo de acesso a outros serviços, mas delegando o processo de autenticação no sistema de gestão de identidades. Mas de maior interesse no contexto deste projeto é o uso do dispositivo móvel para autenticar o utilizador em serviços que são acedidos a partir de dispositivos que não lhe pertencem. São exemplos disto quiosques públicos para acesso à internet ou outro tipo de terminais dedicados, para utilização pública.

No caso especifico do operador, há uma miríade de dispositivos pertencentes ao operador com os quais o utilizador interage diariamente. São exemplos disto, terminais nas lojas aderentes do operador, boxes televisivas e terminais públicos. Ou seja equipamentos que acedem a serviços do operador em nome do utilizador, para os quais a posse do dispositivo poderia dispensar o uso de credenciais, desde que usado o telemóvel como dispositivo para se autenticar.

IV. Arquitetura

A arquitetura do projeto Multipass suporta dois mecanismos de autenticação distintos. O primeiro faz uso de tokens de autenticação, baseados em criptografia de chave pública que são um conjunto de chaves assinadas por um gerador. O segundo consiste na integração com sistemas de IdM, através de um mecanismo de autenticação adicional, que recorre ao operador para autenticar o telemóvel e o seu portador por forma a iniciar sessões autenticadas noutros dispositivos. Estes dois mecanismos suportam respetivamente os cenários descritos na secções III-A e III-B.

A. Pressupostos

O pressuposto fundamental que suporta esta arquitetura é o uso de mecanismos de comunicação com garantias de confidencialidade (e mediante o cenário, autenticação). Na prática as troca de mensagens iniciadas pelo dispositivo móvel são feitas sobre túneis TLS (independentemente do protocolo de transporte), que se necessário, implementam autenticação com recurso a certificados, e nos permitem estender o sistema com recurso a verificação de certificados numa PKI ou mecanismos de revogação.

Apesar de se tentar garantir a confidencialidade dos dados armazenados (e.g. tokens) durante o processo de transferência, o problema de armazenamento seguro no dispositivo móvel não é abordado, já que consideramos que existem outras soluções de uso geral com este fim, por exemplo envolvendo cifragem do armazenamento ou dispositivos TPM, para proteger os dados do utilizador em caso de perda ou roubo do dispositivo.

B. Autenticação com recursos a tokens

Neste contexto o dispositivo móvel funciona como sistema de armazenamento de tokens do utilizador. Estes tokens são estruturas assinadas criptograficamente que associam as chaves públicas do dispositivo que armazena o token com os certificados do gerador do token, e também dos consumidores que podem autenticar o dispositivo com base no token.

Cada token é composto pelos seguintes campos (Fig. 1):

1) B_P: é a chave pública do dispositivo Multipass, à qual corresponde uma chave privada (B_K) disponível apenas no dispositivo. Esta chave (B_K) é usada pelo dispositivo para se autenticar.
2) *Metadata*: é um campo que contém dados adicionais específicos para a aplicação que gerou o token.
3) *E*: é um certificado que pode ser usado pelo dispositivo para autenticar o sistema que consome o token.
4) *V*: é um certificado associado ao sistema de controlo de acesso que gerou o token
5) *Sig(Vk)*: é uma assinatura feita pelo detentor da chave privada correspondente a *V*, ou seja o gerador do token.

Fig. 1. Estrutura de um token.

O processo de criação de tokens consiste na associação de uma chave pública, armazenada no dispositivo móvel, a um conjunto de metadata fornecido por um gerador de confiança. Não consideramos no entanto mecanismos para estabelecer essa relação de confiança, e assumimos que esta já existe, por intermédio de outros protocolos. Por exemplo no primeiro cenário (secção III-A) assumimos que existe uma associação segura com recurso a Bluetooth que foi estabelecida previamente pelo utilizador, ou que existe outro mecanismo de confiança pré-estabelecido entre o utilizador e o gerador de tokens.

A informação incluída em cada token é suficiente para que todas as partes se autentiquem mutuamente, após a criação do mesmo. O dispositivo móvel usa o certificado *E* incluído no token para verificar se está a entregar o token a um serviço em quem o gerador confia. Já o consumidor do token verifica a assinatura e certificado do gerador, para assegurar que o token é de confiança e com base na chave *Bp*, autentica o dispositivo que entrega o token.

Esta solução pode ser usada tanto em situações ad-hoc em que os certificados colocados no token são assinados pelo próprio detentor, assim como em cenários em que existe uma PKI para verificar a confiança nos consumidores e geradores de tokens.

C. Integração com sistemas de IdM

A segunda parte da arquitetura do Multipass dedica-se a lidar com integração com serviços de IdM. Esta integração tem fundamentalmente dois objectivos:

1) Suportar o uso de Security Assertion Markup Language (SAML) como protocolo de autenticação em serviços, juntamente com mecanismos do operador para autenticar o dispositivo móvel.
2) Permitir o uso destes protocolos em cenários onde o browser (que é autenticado para aceder ao serviço) não se encontra em execução no dispositivo móvel.

Usar os recursos do operador para autenticar o dispositivo móvel permite-nos separar as comunicações, fazendo que os dados relativos aos serviços passem pelo canal de dados IP (Wifi ou 3G), mas os dados relativos ao processo de autenticação com o IdP sejam transmitidos de forma segura sobre o canal de controlo do operador. Isto pode ser

conseguido recorrendo aos mecanismos de comunicação Over the Air (OTA) que podem iniciar um processo de autenticação no telemóvel, por intermédio de um pedido de PIN pelo cartão SIM – o arranque da aplicação SIMToolkit, instalada no cartão SIM é conseguida através de uma mensagem que é enviada entre do operador diretamente para o SIM card, através de uma plataforma OTA do operador. A integração de ambos os protocolos, faz-se com recurso a inserção de um cabeçalho adicional, que contém o identificador do utilizador, em todas mensagens HTTP enviadas ao IdP do utilizador (que é pré-configurado no browser). Este cabeçalho é então utilizado pelo IdP para identificar o utilizador e iniciar o processo de autenticação, comunicando diretamente com o dispositivo móvel através da OTA. O utilizador recebe um pedido de PIN iniciado pelo cartão, semelhante aquele que é usado para desbloquear o dispositivo.

Desta forma é possível integrar protocolos de IdM, como o SAML (Fig. 2), com um mecanismo de autenticação disponibilizado pelo OTA, e aplicação SIMToolKit. No entanto o processo é igualmente válido para OpenID[1].

Este mecanismo permite separar fisicamente o dispositivo em que o utilizador se autentica (terminal móvel) do dispositivo em que o browser que estabelece a sessão está em execução. É com recurso a esta solução que se pretende instanciar o segundo cenário descrito, em que o dispositivo móvel autentica um quiosque público para agir em nome do utilizador.

Para que se possa separar os dois componentes é necessário adicionar algumas condições para manter o pré-requisito de autenticação mútua:

1) Tem de existir um canal de comunicação entre o terminal móvel e o browser com garantias de confidencialidade
2) Se possível o IdP utilizado no processo de autenticação deve autenticar também o dispositivo onde o browser é executado

O canal seguro entre o terminal móvel e o browser, serve para transmitir o identificador do utilizador entre o telemóvel e o browser, e para manter estado sobre a associação do entre os dispositivos.

O ponto 2) visa minimizar o risco de ataques de Man-in-the-Middle que embora nunca pudessem capturar o PIN do utilizador (já que este passa num canal diferente), poderiam permitir que um atacante tomasse controlo da sessão. Caso não seja possível faze-lo, o IdP pode ainda tomar outro tipo de medidas para gerir o risco, como restringir os privilégios da sessão que foi autenticada para só permitir algumas operações.

Este tipo de soluções é funcionalmente muito semelhante aos cenários designados de "split-terminal" [7], existentes na Generic Bootstrap Architecture (GBA). A grande diferença prende-se com os protocolos utilizados tanto para iniciar a autenticação como para autenticar o dispositivo, que aqui pretendemos evitar, favorecendo a integração com um sistema de IdM e protocolos mais fáceis de implementar.

D. Descoberta de Serviços

Em qualquer um dos cenários descritos o processo de autenticação é sempre iniciado pelo dispositivo móvel. Embora seja possível que o dispositivo notifique o utilizador da presença de serviços nas imediações, o processo de autenticação é sempre iniciado de forma explícita pelo utilizador.

Para descobrir serviços é possível recorrer-se a três tecnologias distintas:

1) Pesquisa de serviços Bluetooth, recorrendo ao protocolo SDP
2) Interação com códigos de barras QR
3) Interação com tags NFC

Nem todos os mecanismos de pesquisa de serviços são práticos para todas as situações, em ambientes em que existem vários serviços do mesmo tipo é normalmente preferível tags ou códigos de barras, para que não exista ambiguidade sobre o dispositivo com que se está a interagir.

O resultado de qualquer uma destas opções é um endereço de destino, que pode ser um IP e porto TCP, ou um endereço e canal RFCOMM (Bluetooth). No caso particular dos serviços de tokens, é incluído no resultado da pesquisa de serviços, um

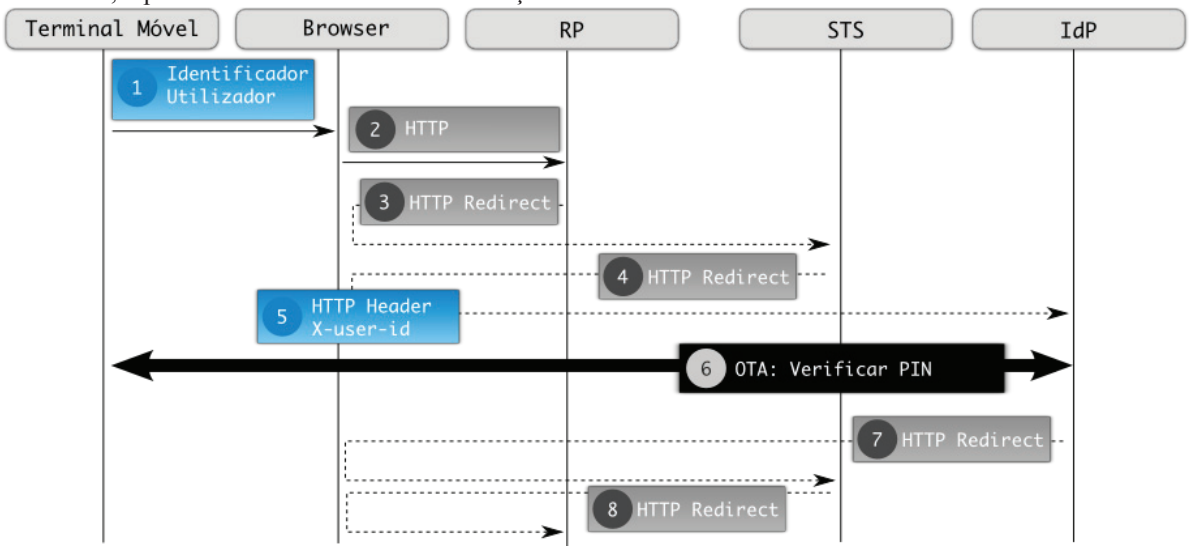


Fig. 2. Integração de Autenticação Over the Air com SAML

identificador que pode ser mapeado no certificado (*E*) que está incluído em cada token. Esta informação permite associar tokens a serviços descobertos, e ignorar serviços para os quais o dispositivo não possui tokens.

V. Implementação e Resultados

O principal objetivo deste projeto é desenvolver um protótipo capaz de demonstrar as interações de um dispositivo móvel com outros elementos do meio envolvente de forma segura e sem fios, instanciando os cenários descritos na secção III.

A implementação está dividida em três componentes (Fig. 3): uma aplicação Android que gere os tokens do utilizador no dispositivo, e controla o processo de autenticação; um serviço que instancia tokens e autentica o dispositivo com base nos mesmos; e um quiosque de acesso público onde o utilizador se pode associar com o seu dispositivo móvel para aceder a outros serviços através de um browser.

A. *Ambiente de desenvolvimento*

O ambiente de desenvolvimento considerado é baseado em ferramentas Java (versão 1.6 da Standard Edition) em execução em ambiente Linux. A implementação no dispositivo móvel Android é feita sobre a versão 2.3.3 (API 10) e compatível com superiores, não requerendo permissões especiais de acesso ao dispositivo.

O conteúdo trocado entre componentes é baseado em mensagens codificadas e com uma estrutura pré definida e conhecida pelos vários intervenientes da comunicação. A codificação de mensagens trocadas entre o dispositivo móvel e os restantes serviços é feita utilizando recorrendo à biblioteca de serialização de mensagens *Protocol Buffers*.

B. *Abstração de camada de transporte*

A interação entre componentes pode ocorrer sobre diversos mecanismos de transporte, sendo que uma rede IP pode não estar disponível como é o caso de Bluetooth.

Para permitir abstrair o protocolo de transporte da rede foi criado um proxy transparente para interligar clientes que utilizem diferentes protocolos de comunicação sem fios, com serviços implementados sobre TCP/IP.

Desta forma todos os serviços são implementados como servidores TCP/IP comuns, e apenas a implementação do terminal móvel e deste proxy é que tem lidar com as especificidades de cada protocolo de comunicação sem fios. De momento este componente lida apenas com dois protocolos orientados à conceção, TCP/IP e RFCOMM/Bluetooth e para além de fazer o mapeamento entre sockets nos dois protocolos, cria também as entradas para descoberta de serviços sobre Bluetooth. Apesar de ser feito uso de NFC para descoberta de serviços, o uso de NFC como protocolo de transporte ainda não foi considerado.

C. *Serviço de Tokens*

O serviço de tokens é responsável por negociar novos tokens com o dispositivo móvel, e consumir os tokens no processo de autenticação. Estas duas funcionalidades são funcionalmente independentes, ou seja uma instância do serviço de tokens pode criar um token que deve ser consumido por uma outra instância, desde que o consumidor confie no certificado do gerador do token.

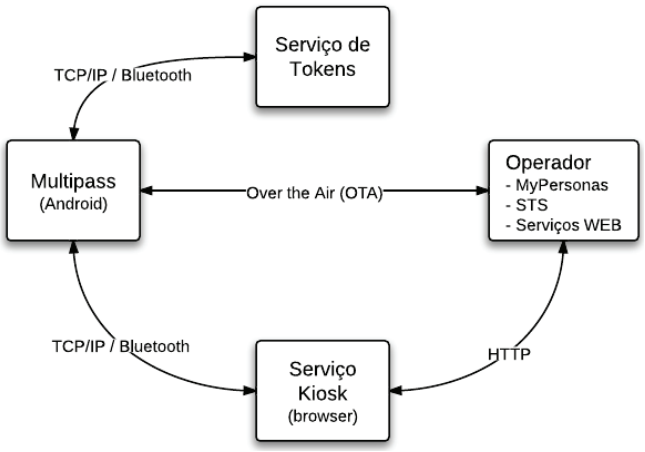


Fig. 3. Diagrama de componentes desenvolvidos.

O protótipo desenvolvido associa aos tokens gerado, e a metadata, com instruções arbitrárias para serem executados pelo consumidor do token. Neste caso o protótipo exige interação com o utilizador para escolher a ação que vai ser associada ao token. Isto permite controlar remotamente um dispositivo com recurso a estes tokens de autenticação mútua, despoletando ações como desbloquear o ecrã de um computador, ou desligando um terminal a partir do dispositivo móvel.

D. *Serviço Quiosque*

Este serviço pretende fornecer um terminal público de acesso à internet para consulta de serviços por parte das entidades que disponibilizem autenticação por intermédio do operador. O quiosque não é mais que um terminal público com um browser, que o utilizador pode usar.

Para suportar a autenticação do utilizador no quiosque foi implementada uma extensão para o browser Firefox que recebe o identificador do utilizador, enviado pelo dispositivo móvel que se liga ao quiosque, e o reenvia para um IdP pré-configurado no quiosque.

Quando o utilizador se liga ao quiosque com o seu dispositivo móvel, este inicializa um browser para ser utilizado. Apenas quando o utilizador tenta usar o browser para aceder a um serviço que requer autenticação através do IdP é que a extensão do browser reenvia o identificador. Uma vez enviado este identificador o processo é transparente para o serviço de quiosque, sendo o IdP a gerir o processo de autenticação com o dispositivo do utilizador.

Para garantir que a sessão do utilizador não permanece em funcionamento depois de terminado o uso no quiosque, foram implementados mecanismos que mantêm a sessão ativa enquanto existir uma ligação entre o dispositivo do utilizador e o quiosque. A ligação entre ambas as partes é mantida ativa com base na troca constante de mensagens em intervalos regulares e ao qual a falha na troca destas mensagens (e.g. se o utilizador se afastar do quiosque) resulta numa interrupção do sistema de comunicação de ambos os lados, voltando tanto a aplicação como o terminal ao seu estado inicial.

E. *Aplicação Multipass*

A aplicação Multipass é responsável por fazer uso dos dois serviços implementados, serviço de tokens e serviço quiosque. Com esta aplicação no dispositivo móvel, o utilizador pode fazer gestão dos seus tokens (aquisição, entrega e armazenamento) e interagir com serviços de quiosque.

A pesquisa de serviços nas imediações pode ser efetuada com recurso a Bluetooth, códigos QR ou tags NFC, embora em ambientes em que coexistam múltiplas instâncias destes serviços seja preferível recorrer às tags e códigos de barras.

Após a fase de pesquisa de serviços recorrendo a um dos protocolos disponíveis, a aplicação permite despoletar uma das seguintes ações:

1) Adquirir um token gerado por um dispositivo próximo
2) Usar um token gerado anteriormente, para se autenticar com um consumidor de tokens
3) Associar o dispositivo móvel a um quiosque para posteriormente autenticar uma nova sessão no quiosque

Mais ainda, a aplicação pode ser configurada para efetuar ela própria pesquisa autónoma de serviços de quiosque e lançar uma notificação ao utilizador sempre que seja encontrado algo nas redondezas. Esta pesquisa de serviços necessita obrigatoriamente de uma ligação Bluetooth constantemente ativa de forma a realizar as pesquisas.

Por fim para permitir instanciar túneis TLS sobre sockets Bluetooth foi desenvolvido um mecanismo que permite utilizar sockets não IP em APIs que usam sockets IP, permitindo desta forma utilizar a implementação de TLS existente nos dispositivos Android sobre outros tipos de sockets, como sendo Bluetooth. Internamente esta solução funciona como um proxy invertido, instanciado dentro da aplicação Android que reencaminha as mensagens através de outro tipo de sockets e mapeia os diferentes sockets. Esta solução sofre de algumas limitações resultantes da API de Android, uma vez que as sockets Bluetooth não suportam operações assíncronas, é necessário implementar esta solução numa thread separada.

F. *Resultados*

Um dos objetivos do projeto consistia na produção de um protótipo, que demonstrasse os cenários do projeto. No entanto interessa ainda medir o impacto da utilização de túneis TLS sobre o protocolo Bluetooth.

Os testes realizados medem o tempo de ida e volta de uma mensagem de 100KBytes. Embora este valor seja pelo menos uma ordem de grandeza acima de uma transação de um token Multipass (que ocupa menos de 10KB), consideramos este valor antevendo o uso de grandes quantidades de metadata nos tokens e futuras extensões à arquitetura.

O dispositivo móvel é o iniciador da ligação e os tempos incluem o estabelecimento e término das ligações. Não nos sendo possível medir com exatidão o custo temporal de cada um dos protocolos para a mesma ligação, optou-se por repetir esta experiência para três situações distintas:

1) Uma ligação Bluetooth através das APIs normais de Android
2) Uma ligação Bluetooth através do sistema de mapeamento de sockets, mas sem instanciar os mecanismos de TLS

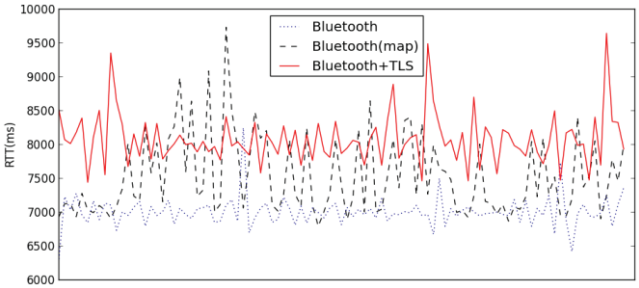


Fig. 4. RTT para um para envio e receção de 100KB sobre Bluetooth

3) Uma ligação TLS sobre um socket Bluetooth

O gráfico apresentado na Figura. 4, contém o tempo da ligação para um total de 98 experiências, para cada um dos casos. O tempo médio gasto por uma ligação Bluetooth normal, uma ligação Bluetooth mapeada e uma ligação TLS sobre Bluetooth foi respetivamente de 7009, 7527 e 8077 milissegundos.

Destes valores retira-se que a utilização de TLS sobre Bluetooth resultou na introdução (para esta operação) de um tempo adicional médio de 1069 milissegundos. Deste tempo, em média 518 milissegundos são resultantes da nossa implementação para mapear sockets, enquanto os restantes resultam do uso de TLS.

Face ao tempo total médio para esta operação (7009ms) houve uma degradação média de cerca de 15%, sendo que metade desse agravamento resulta da implementação que foi produzida.

VI. Conclusões

Neste artigo foram descritas a arquitetura e implementação desenvolvidas no decorrer do projeto Multipass2, cujo objetivo é recorrer a dispositivos móveis como mecanismo privilegiado para interação com serviços recorrendo a autenticação mútua.

A arquitetura apresentada visa suportar dois cenários distintos. O primeiro cenário incide sobre ambientes em que o dispositivo móvel é usado para interagir e controlar múltiplos dispositivos na proximidade do utilizador, como é o caso de ambientes de domótica e Internet of Things. O segundo cenário ambiciona usar o telemóvel como mecanismo para autenticar sessões noutros dispositivos, recorrendo aos sistemas de IdM, para autenticar o utilizador e aos serviços do operador para que as transações referentes ao processo de autenticação ocorram num canal separado da rede de dados.

Para instanciar os cenários pretendidos foi implementado um protótipo com o propósito de validar a arquitetura. Foi desenvolvida uma aplicação Android para gerir tokens de autenticação armazenados no dispositivo e interagir com outros serviços, através de redes IP ou Bluetooth e utilizando códigos QR ou tags NFC para facilitar o processo. Foram também criados componentes para integração com outros serviços, em particular com vista à integração com o IdP e mecanismos de comunicação diretamente com o SIM Card do operador.

AGRADECIMENTOS

O trabalho aqui apresentado decorre do projeto Multipass2 financiado pela PT Inovação, e desenvolvido em colaboração com grupo de investigação ATNoG no pólo de Aveiro do Instituto de Telecomunicações.

REFERÊNCIAS

[1] D. Recordon, J. Bufu, J. Hoyt, B. Fitzpatrick, and D. Hardt, OpenID Authentication 2.0 http://openid.net/specs/openid-authentication-2_0.html, December 2007.
[2] Cantor, S., Kemp, J., Philpott, R., and E. Maler. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard saml-core-2.0-os, March 2005.
[3] http://www.mobile.se
[4] http://web.twelvehorses.com/technology/ticketing/
[5] http://www.openways.com/
[6] Ross, P.E., "Phone-y money," Spectrum, IEEE , vol.49, no.6, pp.60-63, June 2012 doi: 10.1109/MSPEC.2012.6203971
[7] 3GPP TS 33.220; Generic Authentication Architecture (GAA); Generic bootstrapping architecture http://www.3gpp.org/ftp/Specs/html-info/33220.htm, June 2006.

Protocolos de Encaminhamento para Redes Veiculares com Ligações Intermitentes

Vasco N. G. J. Soares[1,2], João A. Dias[1], João N. Isento[1] e Joel J. P. C. Rodrigues[1]
[1]Instituto de Telecomunicações, Universidade da Beira Interior, Covilhã, Portugal
[2]Escola Superior de Tecnologia, Instituto Politécnico de Castelo Branco, Portugal
vasco.g.soares@ieee.org, joao.dias@it.ubi.pt, joao.isento@it.ubi.pt, joeljr@ieee.org

***Resumo*— As redes veiculares são constituídas por automóveis ou outros meios de transporte equipados com dispositivos de comunicação sem fios, que comunicam diretamente entre si ou com equipamentos de infraestrutura localizados junto às estradas. Nestas redes, o encaminhamento de dados é considerado um grande desafio tendo em conta as topologias de rede altamente dinâmicas, densidade variável, tempos de contacto reduzidos, ligações intermitentes e frequentes partições. Este artigo pretende apresentar uma análise comparativa do desempenho de protocolos de encaminhamento baseados no paradigma de "armazenamento, transporte e envio de agregados" (do inglês, *store-carry-and-forward*) em redes veiculares com ligações intermitentes (do inglês, *vehicular delay-tolerant networks* - VDTNs). Os protocolos considerados são o First Contact, Direct Delivery, Epidemic, Spray and Wait, PRoPHET, GeOpps e GeoSpray. O estudo é conduzido por simulação e analisa o comportamento destes protocolos com base nas métricas de desempenho: número de transmissões de agregados iniciadas, número de agregados descartados, probabilidade de entrega de agregados, tempo médio do atraso na entrega de agregados, número médio de saltos e sobrecarga de recursos.**

Palavras-chave — Redes Veiculares com Ligações Intermitentes; Protocolos de Encaminhamento; Paradigma "Armazenamento, Transporte e Envio de Agregados"; Avaliação do Desempenho.

I. INTRODUÇÃO

As redes veiculares [1-3] têm vindo a despertar um interesse crescente devido à sua potencial utilização numa ampla gama de aplicações com impacto na vida quotidiana [1, 4]. Entre inúmeras aplicações destacam-se a segurança rodoviária (p. ex. evitar acidentes), a eficiência dos sistemas de transporte (p. ex. optimização de fluxos de tráfego), a recolha e transmissão de dados de monitorização (p. ex. medições de poluição), as aplicações comerciais (p. ex. divulgação de informações turísticas e de lazer), e as aplicações de entretenimento (p. ex. partilha de conteúdos de multimédia). Estas redes poderão ter um papel essencial para assegurar a conectividade em regiões remotas ou comunidades rurais em países subdesenvolvidos que não dispõem de qualquer tipo de meio de comunicação convencional de acesso à Internet, ou em cenários de catástrofe natural quando as infraestruturas de rede tradicionais são destruídas ou muito afectadas.

No entanto, para que estas aplicações se tornem realidade, é necessário encontrar soluções para uma série de problemas e desafios técnicos que caracterizam estas redes. Alguns desses desafios são comuns a outras redes sem fios, enquanto outros são colocados pelas características particulares das redes veiculares. De acordo com [2, 3, 5-7], a maioria dos problemas é causada pela mobilidade e a velocidade dos veículos, a qual é responsável por uma topologia de rede altamente dinâmica e tempos de contacto entre nós de duração reduzida.

Acrescentam-se ainda as limitações no alcance da transmissão, os problemas de propagação causados por obstáculos (p. ex. edifícios, túneis, terreno e vegetação) e as interferências. Em conjunto, estas questões tornam estas redes susceptíveis a conectividade intermitente e divisão/partição da rede, resultando na impossibilidade frequente de estabelecer uma ligação extremo-a-extremo entre a origem e o destino da comunicação de dados. Além disto, nestas redes, a densidade de nós pode ser altamente variável. Por exemplo, uma rede veicular pode ser classificada como densa num engarrafamento, enquanto que no tráfego suburbano pode ser esparsa, ou até mesmo extremamente esparsa em zonas rurais/remotas.

Pelo acima exposto, torna-se claro que a disseminação de dados numa rede veicular é um problema complexo. A natureza volátil e imprevisível destas redes espontâneas e auto-organizadas aumenta a complexidade inerente ao desenvolvimento de protocolos de encaminhamentos que optimizem o desempenho destas redes.

Neste trabalho, procede-se a uma análise sobre a utilização de protocolos de encaminhamento baseados no paradigma de "armazenamento, transporte e envio de agregados" (do inglês, *store-carry-and-forward* - SCF), aplicados a cenários de redes veiculares com ligações intermitentes (do inglês, *vehicular delay-tolerant networks* - VDTNs).

O artigo está estruturado da forma descrita em seguida. Na Secção II apresentam-se os protocolos de encaminhamento (do tipo SCF) de referência propostos na literatura e em estudo. A comparação do desempenho destes protocolos é apresentada na Secção III. A Secção IV conclui o artigo e apresenta o trabalho futuro.

II. TRABALHO RELACIONADO

A maioria dos estudos publicados sobre protocolos de

encaminhamento para redes veiculares considera cenários de autoestradas e ruas citadinas, caracterizados por uma elevada densidade de nós ou até mesmo redes totalmente conectadas. Os protocolos de encaminhamento propostos na literatura para estas redes, designadas de redes veiculares ad-hoc (do inglês, *vehicular ad-hoc networks* - VANETs) [1-4, 8-10], assumem a existência de uma ligação extremo-a-extremo entre o emissor e o receptor de dados [11, 12]. Pelo que apresentam como grande limitação a incapacidade de lidar com ligações intermitentes, com a frequente desconexão da rede (partições da rede) e atrasos longos ou variáveis [2, 3, 13-15]. Tais situações são muito comuns em cenários urbanos ou rurais caracterizados por densidades de nós moderadas ou reduzidas, e pouca ou nenhuma infraestrutura de rede fixa disponível.

A arquitetura de rede proposta para redes com ligações intermitentes (do inglês, *delay/disruption-tolerant networks* - DTNs) [16], e inicialmente aplicada às redes interplanetárias, foi entretanto considerada uma solução promissora para resolver estes problemas. A arquitetura DTN propõe que o encaminhamento das unidades protocolares de dados, designadas de agregados (do inglês, *bundles*) [17], seja realizado de acordo com um paradigma de "armazenamento, transporte e envio de agregados", assente na utilização de armazenamento persistente, que tira partido de contactos oportunísticos e de comunicação assíncrona. A ideia subjacente a este paradigma é a de permitir que um nó armazene os agregados em memória persistente (p. ex. disco rígido) e os transporte até encontrar o nó de destino, ou um nó intermédio que poderá vir a encontrar o nó de destino num futuro próximo. Sempre que ocorre uma oportunidade de contacto, cada nó toma as suas decisões de encaminhamento de forma independente.

A utilização deste paradigma permite que o tráfego de dados de uma variedade de aplicações veiculares tolerantes ao atraso e à perda de alguns dados, seja encaminhado ao longo do tempo, explorando-se o movimento físico dos veículos e os contactos oportunísticos estabelecidos entre estes e com outros nós da rede. Um exemplo de uma arquitetura proposta para comunicações veiculares em redes esparsas/particionadas e oportunísticas, que segue este paradigma, foi apresentado em [18]. Este tipo de redes são designadas de redes veiculares com ligações intermitentes (do inglês, *vehicular delay-tolerant networks* - VDTN).

Algumas das características que diferenciam os protocolos de encaminhamento para redes do tipo DTN, tais como as VDTNs, são se assumem que não existe qualquer tipo de conhecimento, ou se consideram e eventualmente combinam informação sobre dados de histórico (p. ex. encontros recentes, tempo de contacto, frequência de contacto), localização (p. ex. informação de localização do passado, presente, ou futuro), ou ainda padrões de mobilidade.

Os protocolos de encaminhamento são também habitualmente classificados em duas classes: cópia-única e cópia-múltipla. Os protocolos com estratégia baseada em cópia-única permitem que apenas uma única cópia de um agregado esteja presente na rede num determinado instante de tempo. Pelo contrário, se a estratégia é baseada em cópia-múltipla, então os protocolos permitem a replicação de um mesmo agregado para vários nós. Com esta abordagem tenta-se aumentar a probabilidade de entrega e minimizar o atraso na entrega. Contudo, tal estratégia pode causar sobrecarga na utilização de recursos de largura de banda e de armazenamento nos nós de rede.

Apresentam-se de seguida alguns dos protocolos de encaminhamento mais populares para redes DTN, que podem ser utilizados em redes veiculares do tipo VDTN.

Direct Delivery [19] e First Contact [20], são exemplos de dois protocolos de encaminhamento com estratégia de cópia-única, que não utilizam qualquer tipo de informação sobre a rede. No protocolo Direct Delivery, o nó transporta um agregado até encontrar o nó de destino e entregar o mesmo. Este protocolo provoca sobrecarga (*overhead*) mínima, porém pode implicar atrasos de entrega muito elevados. No protocolo de encaminhamento First Contact, os nós comutam/despacham os agregados que transportam para o primeiro nó que encontram. Tal resulta numa procura aleatória pelo nó de destino. O principio de funcionamento destes protocolos encontra-se ilustrado nas Figuras 1 e 2 respectivamente.

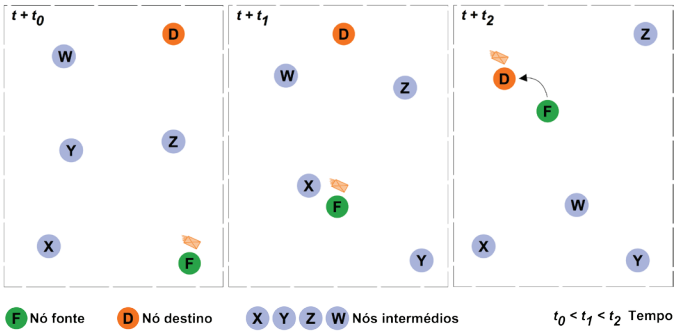


Fig. 1. Principio de funcionamento do protocolo Direct Delivery.

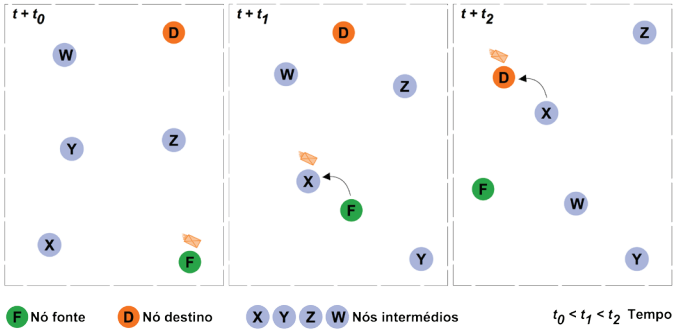


Fig. 2. Principio de funcionamento do protocolo First Contact.

O protocolo Epidemic [21] baseia-se num esquema de "inundação" (*flooding*) puro, representado na Figura 3. Ou seja, os agregados são replicados para todos os nós encontrados na rede. Intuitivamente, o *flooding* assegura a entrega dos agregados ao nó de destino no menor tempo possível, se não existirem limitações de recursos (p. ex.

largura de banda e armazenamento). Contudo, tal abordagem representa um consumo excessivo de recursos e pode degradar severamente o desempenho da rede.

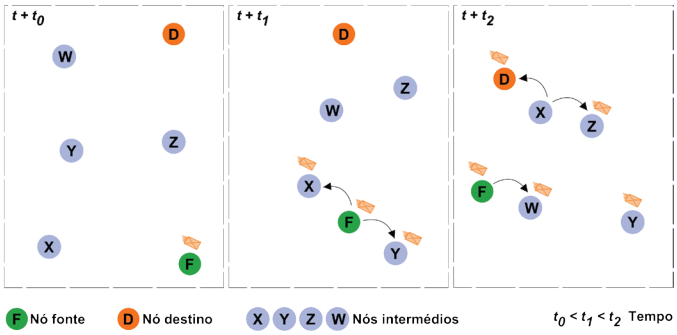


Fig. 3. Principio de funcionamento do protocolo Epidemic.

O protocolo Spray and Wait [22] tenta controlar o *flooding* limitando o número máximo de cópias criadas por agregado. Este protocolo inicialmente distribui um número de cópias (L) de um agregado para nós intermédios e depois espera até que um destes nós encontre o nó de destino. A distribuição inicial das cópias pode ser realizada de acordo com uma de duas estratégias: "normal" ou "binária". Na estratégia "normal", ilustrada na Figura 4, o nó fonte transmite uma das cópias do agregado para cada nó encontrado. Na estratégia "binária", representada na Figura 5, metade das cópias do agregado são enviadas para cada nó encontrado.

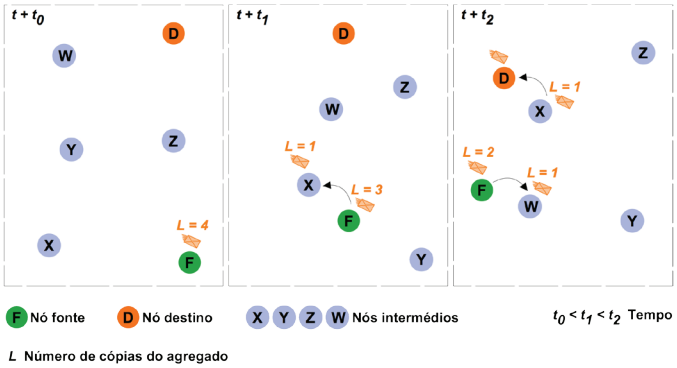


Fig. 4. Principio de funcionamento do protocolo Spray and Wait - Normal.

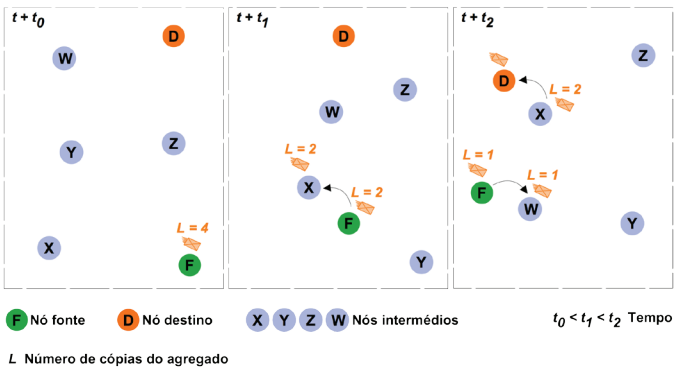


Fig. 5. Principio de funcionamento do protocolo Spray and Wait - Binário.

O protocolo PRoPHET [23] utiliza informação relativa ao histórico de contactos entre nós e da transitividade, representada na Figura 6. O histórico de contactos entre nós define $P_{(a,b)}$ como a probabilidade de dois nós, a e b, se encontrarem, e é calculada de acordo com a equação (1), onde P_{init} é uma constante de inicialização.

$$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{init} \quad , \quad P_{init} \in [0,1] \qquad (1)$$

Esta probabilidade ($P_{(a,b)}$), designada de previsibilidade de entrega, aumenta sempre que a e b se encontram. Se estes nós não se encontram frequentemente, então $P_{(a,b)}$ diminui à medida que o tempo decorre. Para esse efeito utiliza-se a equação (2), onde γ é a constante de envelhecimento e k o número de unidades de tempo que passaram desde a última vez que a métrica foi atualizada.

$$P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^{k} \quad , \quad \gamma \in [0,1) \qquad (2)$$

A previsibilidade de entrega tem uma propriedade transitiva $P_{(a,c)}$, que é baseada na observação de que se um nó a encontra frequentemente um nó b, e o nó b encontra frequentemente um nó c, então o nó c provavelmente é um bom nó para encaminhar agregados destinados ao nó a. A transitividade é calculada de acordo com a equação (3), onde β é uma constante que determina o impacto da transitividade no cálculo da previsibilidade de entrega.

$$P_{(a,c)} = P_{(a,c)_{old}} + (1 - P_{(a,c)_{old}}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \quad , \quad \beta \in [0,1] \qquad (3)$$

A métrica de previsibilidade de entrega é atualizada em cada contacto entre nós, e é utilizada para tomar decisões de encaminhamento. Um nó a apenas replica um agregado para um nó b encontrado se a previsibilidade de entrega for superior em b. Tal abordagem visa tentar limitar a replicação excessiva de agregados.

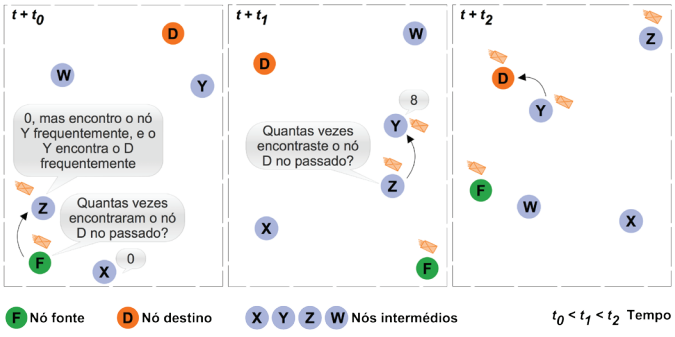


Fig. 6. Principio de funcionamento do histórico de contactos entre nós e da propriedade de transitividade do protocolo PRoPHET.

Os protocolos de encaminhamento geográficos são de particular interesse para as redes veiculares. Estes protocolos

utilizam informação de localização obtida através de dispositivos de posicionamento, tais como GPS (do inglês, *global positioning system*), e outros parâmetros de mobilidade, para auxiliar na tomada de decisões de encaminhamento.

O Geographical Opportunistic Routing (GeOpps) [24] é um exemplo de um protocolo de encaminhamento geográfico popular que apresenta uma estratégia de cópia-única. Este protocolo utiliza informação obtida dos sistemas de navegação disponíveis nos veículos, para determinar que veículos eventualmente se movimentarão até um ponto mais próximo, ou mais cedo, do nó de destino de um agregado. Esta métrica, designada de tempo mínimo estimado de entrega (METD), é utilizada aquando de um contacto para determinar qual veiculo deve ficar na posse de um determinado agregado, por aumentar a probabilidade de entrega do mesmo.

A Figura 7 ilustra o principio de funcionamento deste protocolo, descrito de seguida. O sistema de navegação do nó *F* é utilizado para calcular o valor do METD. Para tal é utilizada a informação sobre a posição geográfica, a velocidade e trajetória/rota do nó *F*, assumindo que este é um veículo. Assumindo que a posição geográfica do nó terminal de destino *D* (do agregado) é conhecida, então é possível determinar a menor distância de *D* ao ponto mais próximo no trajeto/rota de *F*, designado de NP_F. Com base nesta informação, o sistema de navegação permite estimar o tempo de chegada (ETA) de *F* ao ponto NP_F, e estimar também o tempo necessário para ir de NP_F a *D*. A soma destes valores de tempo, representada na equação (4), corresponde ao valor do METD.

$$METD = ETA\ a\ NP_F + ETA\ de\ NP_F\ a\ D \quad (4)$$

No exemplo ilustrado nesta figura, os veículos *F* e *X* encontram-se no ponto *P*. O cálculo do *NP* para *F* e *X*, com base trajetória/rota destes nós, permite concluir que o METD do *X* é inferior ao de *F*. Tal acontece porque o tempo requerido para ir do ponto *P* a NP_X e depois até *D* é inferior ao tempo necessário para ir do ponto *P* até NP_F e depois até *D*. Pelo que, *F* deve transmitir o agregado para *X*.

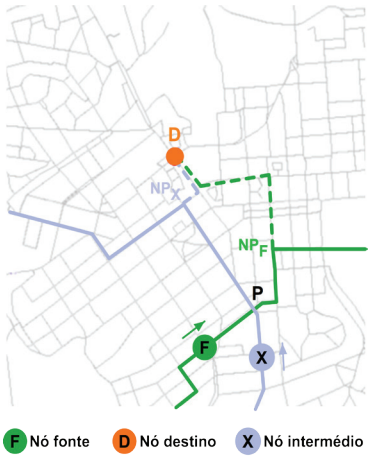


Fig. 7. Principio de funcionamento do protocolo GeOpps.

O GeoSpray [25] é também um protocolo de encaminhamento geográfico que foi proposto recentemente. Este protocolo combina uma abordagem híbrida entre os esquemas de cópia-múltipla e cópia-única. Inicialmente aplica um esquema de cópia-múltipla de estratégia binária, que tem por objectivo disseminar um conjunto limitado de cópias de um mesmo agregado, para diferentes nós intermédios que se aproximarão mais ou chegarão mais rapidamente ao nó de destino. Exploram-se assim caminhos alternativos, escolhendo os nós intermédios com base na métrica de encaminhamento METD. Uma vez distribuídas estas cópias, o protocolo passa a utilizar um esquema de cópia-única, que tira partido de oportunidades de contacto adicionais. Desta forma, é permitido aos nós intermédios na posse de uma cópia do agregado que a despachem/comutem para outro nó encontrado na rede que tenha uma melhor métrica de encaminhamento METD. Por forma a melhorar a utilização de recursos (p. ex. largura de banda, espaço de armazenamento, energia), este protocolo possui um mecanismo designado de recibos ativos que é responsável pela eliminação das cópias dos agregados entregues com sucesso mas que permanecem armazenadas nos nós intermédios da rede.

III. Avaliação do Desempenho

Nesta secção apresenta-se uma análise comparativa do desempenho dos protocolos de encaminhamento, descritos na secção anterior, em redes VDTN. O desempenho é comparado através de um estudo de simulação realizado utilizando a ferramenta VDTNsim [26] que é baseada no simulador Opportunistic Network Environment (ONE) [27].

O objectivo é perceber qual o impacto do número de nós móveis (veículos) no desempenho dos protocolos de encaminhamento. Para tal, foram consideradas seis métricas de desempenho neste estudo. O número de transmissões iniciadas de agregados é definido como o número de transmissões iniciadas entre nós. O número de agregados descartados é definido como o número de agregados que foram descartados dos *buffers* dos nós devido aos *buffers* estarem sobrecarregados ou por o tempo de vida dos agregados ter expirado. A probabilidade de entrega de agregados é definida como a razão entre o número de agregados únicos entregues e o número de agregados únicos criados. O tempo médio do atraso na entrega de agregados é definido como o tempo médio entre a criação dos agregados e a sua entrega. O número de saltos médio é definido como o número médio de saltos entre o nó fonte e o nó destino dos agregados. Finalmente, a sobrecarga de recursos é definida como o número de transferências que é necessário realizar (entre nós) para entregar os agregados.

Nas próximas duas subsecções serão descritos os cenários de simulação avaliados e a correspondente análise de resultados.

A. *Cenários de Simulação*

O cenário de rede é baseado num modelo do mapa de parte da cidade de Helsínquia - Finlândia, apresentado na Figura 8. Dez nós terminais e cinco nós fixos de *relay* [18, 28] foram colocados nas posições assinaladas no mapa. Cada um destes nós tem um *buffer* de tamanho 50 MB.

Durante um período de tempo de 6 horas simuladas (p. ex. das 8:00 às 14:00), 50, 100 ou 200 veículos movem-se ao longo das estradas, entre posições aleatórias, com uma velocidade media de 50 km/h, e com tempos de pausa aleatórios uniformemente distribuídos entre 5 e 15 minutos. Estes nós móveis têm um *buffer* de tamanho 12.5 MB.

Os agregados são originados pelos nós móveis e destinados a nós terminais aleatórios. São gerados aleatoriamente no intervalo de tempo uniformemente distribuído [55, 65] segundos. Têm um tempo de vida de 90 minutos e tamanhos aleatórios uniformemente distribuídos num de 3 intervalos [25 KB, 100 KB], [250 KB, 500 KB] e [750 KB, 1 MB] (bytes), que representam tráfego de diferentes aplicações.

De acordo com o indicado em [29], considera-se que os nós da rede comunicam entre si a uma taxa de transmissão de 4.5 Mbps e assume-se um alcance de 30 m. Nos três cenários simulados, utilizam-se os parâmetros por omissão definidos para o protocolo PRoPHET [23]. De acordo com [30], o parâmetro de número de cópias dos protocolos Spray and Wait e GeoSpray é configurado como 15% do número de nós móveis em cada um dos cenários.

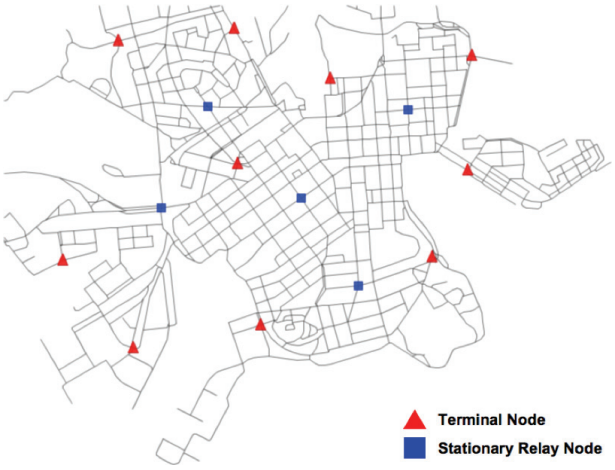


Fig. 8. Modelo do mapa de parte da cidade de Helsínquia - Finlândia, com as posições dos nós terminais e dos nós fixos de *relay* (área de 4500×3400 m).

Para cada um dos três cenários considerados, com diferentes densidades de nós móveis, foram geradas aleatoriamente 30 simulações, as quais foram aplicadas a todos os protocolos de encaminhamento. Os valores apresentados na próxima subsecção correspondem à média aritmética dos resultados das 30 simulações de cada protocolo em cada cenário. O valor do desvio padrão é desprezável e por esse motivo não é representado nos gráficos.

B. *Análise dos Resultados*

A comparação do desempenho dos protocolos de encaminhamento é iniciada com a análise do número de transmissões iniciadas de agregados. Na Figura 9 pode observar-se que o aumento da densidade dos nós móveis leva a um acréscimo do número de transmissões de agregados, dado que aumenta o número de oportunidades de contacto. Esta conclusão aplica-se a todos os protocolos de encaminhamento com exceção do Direct Delivery. No caso particular deste protocolo, os nós móveis (fontes de tráfego) apenas transmitirão os agregados originados em si para os nós terminais de destino, pelo que o incremento do número de nós móveis não implica um aumento do número de transmissões, pois estes nós não trocarão agregados entre si. O mau desempenho do First Contact deve-se à sua abordagem de procura aleatória assente numa estratégia de cópia-única, que resulta em valores elevados de números de transmissões iniciadas, nos três cenários considerados.

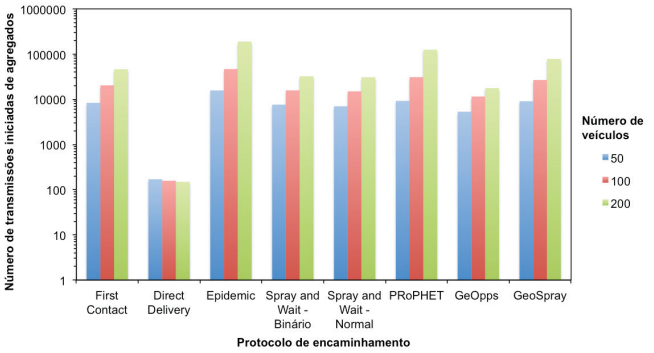


Fig. 9. Número de transmissões iniciadas de agregados em cenários com 50, 100 ou 200 veículos, para os protocolos de encaminhamento First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray.

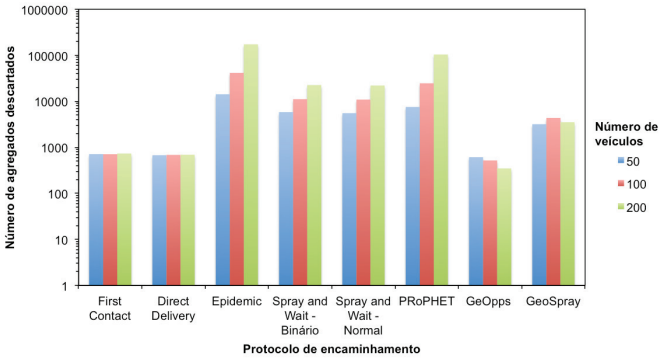


Fig. 10. Número de agregados descartados em cenários com 50, 100 ou 200 veículos, para os protocolos de encaminhamento First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray.

Uma conclusão interessante que se pode retirar da análise das Figuras 9 e 10 é que os protocolos que utilizam uma estratégia baseada em cópia-múltipla contribuem efetivamente para o aumento significativo do número de transmissões

registadas, quando comparados com os de cópia-única, com consequências óbvias no consumo de recursos tais como armazenamento e largura de banda. Na Figura 10, pode observar-se, de facto, que estratégias de cópia-múltipla resultam no descarte de números elevados de agregados devido à ocorrência de sobrecarga dos *buffers* disponíveis nos nós.

Como seria de esperar, o protocolo com pior desempenho em ambas as métricas avaliadas nestas figuras é o Epidemic devido à sua abordagem de *flooding* puro. Merece especial atenção o comportamento do protocolo GeoSpray, que embora por exemplo despolete um número de transmissões ligeiramente superior ao de ambas as variantes do Spray and Wait (Figura 9), apresenta um número de agregados descartados muito inferior (Figura 10). A razão para tal prende-se com o mecanismo de recibos ativos do GeoSpray, que elimina as cópias de agregados entregues na rede que permanecem armazenadas nos nós intermédios da rede, contribuindo para a optimização da utilização dos recursos. Tanto o First Contact como o Direct Delivery apresentam um número de agregados descartados muito reduzido, dado o baixo volume de tráfego de dados considerado, sendo os agregados descartados maioritariamente porque o seu tempo de vida expira e não devido à ocorrência de sobrecarga dos *buffers*.

Os valores registados para a métrica de desempenho de probabilidade de entrega de agregados são apresentados na Figura 11. Como expectável, o aumento do número de nós móveis contribui para uma melhoria da probabilidade de entrega de agregados, para todos os protocolos excepto o First Contact e o Direct Delivery. As abordagens simplistas destes protocolos não tiram partido das oportunidades de contacto adicionais, sendo responsáveis pelo seu mau desempenho.

A análise desta figura permite concluir a importância da informação de localização geográfica para auxiliar na tomada de decisões de encaminhamento. O GeOpps, mesmo sendo um protocolo geográfico com uma estratégia de cópia-única, apresenta um desempenho superior ao do Epidemic e do PRoPHET no cenário de 200 veículos. Por sua vez, o GeoSpray com a uma abordagem híbrida de cópia-múltipla e cópia-única auxiliada pela tomada de decisões com base em informação geográfica, evidencia-se de entre todos os protocolos avaliados, apresentando um acréscimo muito significativo da probabilidade de entrega nos três cenários estudados.

Também se observa que o protocolo PRoPHET embora controle o *flooding* puro (Figuras 9 e 10) utilizando informação relativa ao histórico de contactos entre nós e da transitividade, apresenta um desempenho inferior ao do protocolo Epidemic (Figura 11) em termos de probabilidade de entrega.

Relativamente ao tempo médio do atraso na entrega de agregados, mostrado na Figura 12, pode concluir-se que para os protocolos de encaminhamento Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray, o aumento da densidade de nós móveis resulta num decréscimo relevante desta métrica. Ou seja, nestes protocolos, para o mesmo volume de tráfego de dados, a existência de um maior número de veículos resulta na melhoria da probabilidade de entrega e simultaneamente na diminuição do tempo de entrega (Figuras 11 e 12).

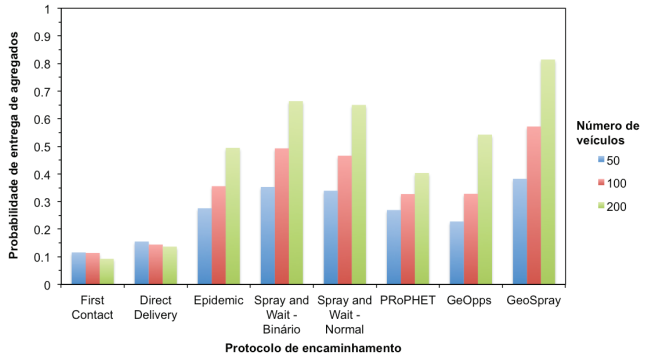


Fig. 11. Probabilidade de entrega de agregados em cenários com 50, 100 ou 200 veículos, para os protocolos de encaminhamento First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray.

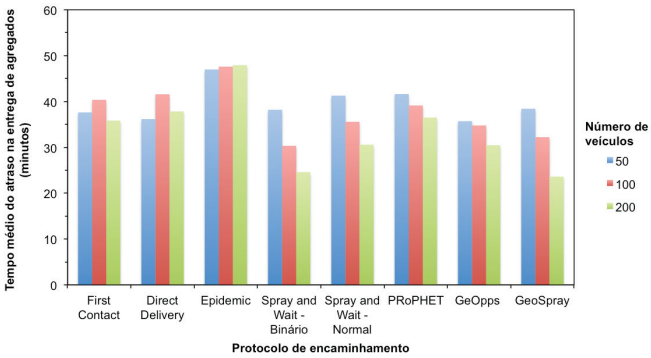


Fig. 12. Tempo médio do atraso na entrega de agregados em cenários com 50, 100 ou 200 veículos, para os protocolos de encaminhamento First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray.

A análise da Figura 13 revela que, como esperado, para os protocolos Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray, o incremento no número de nós móveis resulta no aumento do número médio de saltos entre o nó fonte e o nó destino. A comparação dos resultados obtidos para os dois protocolos geográficos, permite concluir que pelo facto de o GeoSpray permitir a existência de várias cópias de um mesmo agregado (explorando caminhos alternativos), em média requer um número significativamente menor de saltos para entregar os agregados do que o GeOpps.

Os resultados observados para o Direct Delivery e o First Contact seriam previsíveis. No Direct Delivery os nós móveis apenas entregam o agregado ao nó terminal de destino, portanto efetuando um único salto. No First Contact as transmissões aleatórias entre nós à procura do nó de destino levam a que este protocolo apresente os piores resultados nesta

métrica de desempenho.

A última métrica de desempenho considerada neste estudo é a sobrecarga de recursos. Esta métrica avalia a eficiência de utilização da largura de banda dos protocolos de encaminhamento, uma vez que mede o número de transferências que é necessário realizar para entregar os agregados. Os resultados apresentados na Figura 14 comprovam que o protocolo Direct Delivery não cria qualquer sobrecarga de recursos por entregar os agregados apenas ao nó de destino. Fica também demonstrada a completa ineficiência do First Contact.

De uma forma geral as estratégias de cópia-múltipla têm um desempenho inferior nesta métrica. Mas destaca-se que o protocolo GeoSpray consegue alcançar os melhores resultados de probabilidade e atraso de entrega, sendo eficiente em termos da utilização de recursos de largura de banda e de armazenamento.

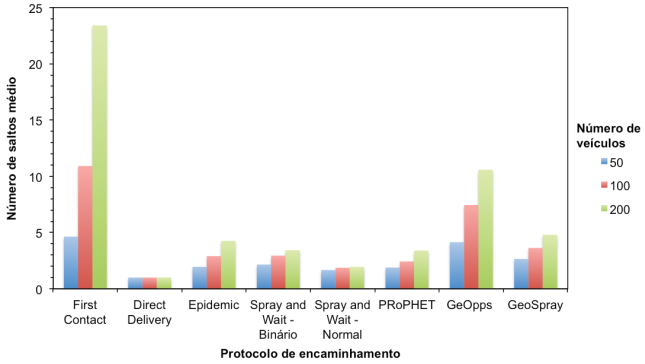


Fig. 13. Número de saltos médio em cenários com 50, 100 ou 200 veículos, para os protocolos de encaminhamento First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray.

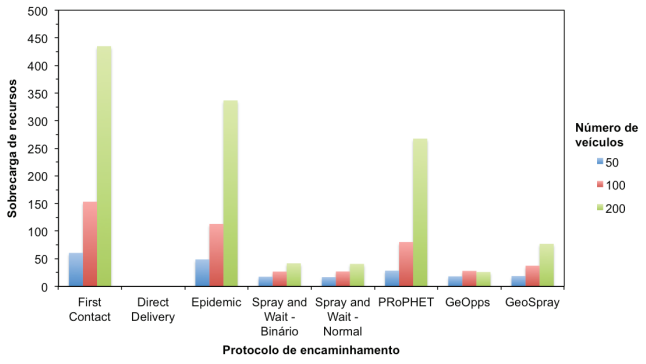


Fig. 14. Sobrecarga de recursos em cenários com 50, 100 ou 200 veículos, para os protocolos de encaminhamento First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray.

IV. Conclusões e Trabalho Futuro

Neste artigo discutiu-se a utilização de protocolos de encaminhamento baseados no paradigma de "armazenamento, transporte e envio de agregados" em redes veiculares com ligações intermitentes, designadas de redes VDTN. Apresentaram-se os protocolos de encaminhamento de referência propostos na literatura para redes do tipo DTN, os quais foram depois avaliados extensivamente através de um estudo de simulação.

Concretamente, os protocolos First Contact, Direct Delivery, Epidemic, Spray and Wait Binário e Normal, PRoPHET, GeOpps e GeoSpray foram avaliados em cenários de rede VDTN com diferentes densidades de nós móveis. Comparou-se o seu desempenho com base em várias métricas, designadamente: número de transmissões iniciadas de agregados, número de agregados descartados, probabilidade de entrega de agregados, tempo médio do atraso na entrega de agregados, número de saltos médio e sobrecarga de recursos.

Observou-se que o GeoSpray apresenta os melhores resultados. Melhora significativamente a probabilidade de entrega de agregados e reduz o tempo médio do atraso na entrega de agregados, quando comparado com os outros protocolos de encaminhamento. Além disso, apresenta uma baixa taxa de agregados descartados e baixa sobrecarga de recursos, sendo portanto eficiente em termos de utilização de largura de banda e armazenamento.

No trabalho futuro pretende-se implementar o protocolo GeoSpray na *testbed* real VDTN apresentada em [31], para avaliar e validar o desempenho deste protocolo num ambiente real.

Agradecimentos

Este trabalho foi parcialmente apoiado pelo Instituto de Telecomunicações, *Next Generation Networks and Applications Group (NetGNA)*, Portugal, e pela FCT – Fundação para a Ciência e a Tecnologia através do projecto PEst-OE/EEI/LA0008/2011.

Referências

[1] Y. Toor, P. Muhlethaler, A. Laouiti, and A. D. L. Fortelle, "Vehicle Ad Hoc Networks: Applications and Related Technical Issues," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 74-88, 2008.

[2] J. Jakubiak and Y. Koucheryavy, "State of the Art and Research Challenges for VANETs," in *Fifth IEEE Consumer Communications & Networking Conference (CCNC 2008) - 2nd IEEE Workshop on Broadband Wireless Access*, Las Vegas, Nevada, USA, January 10-12, 2008, pp. 912-916.

[3] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives," in *6th International Conference on ITS Telecommunications (ITST 2006)*, Chengdu, China, June 21-23, 2006, pp. 761-766.

[4] Y. Khaleda, M. Tsukadaa, J. Santab, J. Choia, and T. Ernst, "A Usage Oriented Analysis of Vehicular Networks: From Technologies to Applications," *Journal of Communications, Academy Publisher*, vol. 4, no. 5, pp. 357-368, June 2009.

[5] O. T. Cruces, "Applying Delay Tolerant Protocols to VANETs," Master Thesis, Universitat Politècnica de Catalunya, Barcelona, 2008.

[6] L. Franck and F. Gil-Castineira, "Using Delay Tolerant Networks for Car2Car Communications," in *IEEE International Symposium on*

Industrial Electronics 2007 (ISIE 2007), Vigo, Spain, 4-7 June, 2007, pp. 2573-2578.

[7] E. Schoch, F. Kargl, M. Weber, and T. Leinmüller, "Communication Patterns in VANETs," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 119-125, November 2008.

[8] H. Füßler, M. Torrent-Moreno, M. Transier, A. Festag, and H. Hartenstein, "Thoughts on a Protocol Architecture for Vehicular Ad-hoc Networks," in *2nd International Workshop on Intelligent Transportation (WIT 2005)*, Hamburg, Germany, March 15-16, 2005.

[9] C.-M. Huang, J.-L. Chen, and Y.-C. Chang, *Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications*. Information Science Publishing, 2009.

[10] M. Torrent-Moreno, A. Festag, and H. Hartenstein, "System Design for Information Dissemination in VANETs," in *3rd International Workshop on Intelligent Transportation (WIT 2006)*, Hamburg, Germany, March 14-15, 2006, pp. 27-33.

[11] M. Zhang and R. S. Wolff, "Routing Protocols for Vehicular Ad Hoc Networks in Rural Areas," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 126-131, 2008.

[12] M. Zhang and R. S. Wolff, "A Border Node Based Routing Protocol for Partially Connected Vehicular Ad Hoc Networks," *Journal of Communications, Academy Publisher*, vol. 5, no. 2, pp. 130-143, February 2010.

[13] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538-1556, October 2007.

[14] T. D. C. Little and A. Agarwal, "An Information Propagation Scheme for VANETs," in *8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, September 13-16, 2005, pp. 155-160.

[15] M. Abuelela and S. Olariu, "Traffic-Adaptive Packet Relaying in VANET," in *The Fourth ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2007), in conjunction with ACM MobiCom 2007*, Montréal, QC, Canada, September 10, 2007, pp. 77-78.

[16] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," RFC 4838, April 2007, [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt.

[17] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC 5050, November 2007, [Online]. Available: http://www.rfc-editor.org/rfc/rfc5050.txt.

[18] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "A Layered Architecture for Vehicular Delay-Tolerant Networks," in *Fourteenth IEEE Symposium on Computers and Communications (ISCC ' 09)*, Sousse, Tunisia, July 5 - 8, 2009, pp. 122-127.

[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy Routing in Intermittently Connected Mobile Networks," in *First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004).* Santa Clara, CA, USA, October 4-7, 2004, pp. 235- 244.

[20] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Portland, Oregon, USA, August 30 - September 3, 2004, pp. 145-158.

[21] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Duke University, Technical Report, CS-2000-06, April, 2000.

[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *ACM SIGCOMM 2005 - Workshop on Delay Tolerant Networking and Related Networks (WDTN-05)*, Philadelphia, PA, USA, August 22-26, 2005, pp. 252-259.

[23] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks," draft-irtf-dtnrg-prophet-09, April 3, 2011, [Online]. Available: http://tools.ietf.org/html/draft-irtf-dtnrg-prophet-09.

[24] I. Leontiadis and C. Mascolo, "GeOpps: Geographical Opportunistic Routing for Vehicular Networks," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2007 (WoWMoM 2007)*, Espoo, Finland, 18-21 June, 2007, pp. 1-6.

[25] V. N. G. J. Soares, J. J. P. C. Rodrigues, and F. Farahmand, "GeoSpray: A Geographic Routing Protocol for Vehicular Delay-Tolerant Networks," *Information Fusion Journal, Elsevier*, (accepted for publication).

[26] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "VDTNsim: A Simulation Tool for Vehicular Delay-Tolerant Networks," in *15th IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (IEEE CAMAD 2010)*, Miami, FL, USA, December 3-4, 2010, pp. 101-105.

[27] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Second International Conference on Simulation Tools and Techniques (SIMUTools 2009)*, Rome, March 2-6, 2009, pp. 1-10.

[28] J. J. P. C. Rodrigues, V. N. G. J. Soares, and F. Farahmand, "Stationary Relay Nodes Deployment on Vehicular Opportunistic Networks," in *Mobile Opportunistic Networks: Architectures, Protocols and Applications*, M. K. Denko, Ed. USA: CRC Press – Taylor & Francis Group (hardcover), 2011, pp. 227-243.

[29] A. Keränen and J. Ott, "Increasing Reality for DTN Protocol Simulations," Helsinki University of Technology, Networking Laboratory, Technical Report, July, 2007.

[30] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," in *Fifth IEEE International Conference on Pervasive Computing and Communications (PerCom 2007) Workshops - International Workshop on Intermittently Connected Mobile Ad hoc Networks (IEEE ICMAN 2007)*, White Plains, NY, USA, March 19-23, 2007, pp. 79-85.

[31] M. C. G. Paula, J. N. Isento, J. A. Dias, and J. P. C. Rodrigues, "A Real-World VDTN Testbed for Advanced Vehicular Services and Applications," in *16th IEEE International Workshop on Computer Aided Modeling Analysis and Design of Communication Links and Networks (IEEE CAMAD 2011)*, Kyoto, Japan, June 10-11, 2011.

Service Platform for Vehicular Networks

Pedro Cruz Sousa
pedro.cruz.sousa@ist.utl.pt
Instituto Superior Técnico, Taguspark
Lisboa, Portugal

Teresa Vasques Vazão
teresa.vazao@ist.utl.pt
Instituto Superior Técnico, Taguspark
Lisboa, Portugal

October 2012

Abstract

In this paper, we summarize a study on related Vehicular Ad-hoc Network (VANET) standards, some past research projects and propose a solution for a service platform in a vehicular environment. We present and analyse the measurements taken with our developed service supporting platform and evaluate the feasibility of store and forward mechanisms and also different types of communication: multi-hop, Vehicle-to-Infrastructure (V2I), Infrastructure-to-Vehicle (I2V) between two Road Side Unitss (RSUs) and an On-Board Unit (OBU) (installed in a vehicle). Our study shows the practicability and possibility of transmitting data in such an environment with little packet loss and latency.

Keywords: VANET, IEEE 802.11, Platform, Multi-hop communication, Store and forward, V2I communication, Experimental performance

1. Introduction

The population of the world, as well as the demand of vehicles for transportation, has been increasing in the past decades. Such demand results in higher congestion issues resulting in higher waiting times and higher pollution, thus degrading the social well-being. Such long waiting periods also wastes fuel and with its pricing soaring and a potential threat of its shortage, there is a clear need to improve road traffic.

Road safety is yet another problem. Every year, accidents cause 43.000 deaths in the USA, in Europe the number of people killed every day is comparable to a medium-haul plane crash (between 180 and 260 people) and in the Asia Pacific region 10 million people are severely injured or killed on the roads every year. Such death toll is major and grim.

In the past decade, numerous efforts that sought to mitigate the aforementioned problems, produced solutions like: information on traffic and hazardous situations being broadcast via FM radio band; variable message signs that warn drivers about changing conditions placed along freeways; electronic toll systems that collect fees with reduced or almost no disruption of traffic flow. These systems are currently being used in many countries, but are not enough to avoid or minimize the presented problems.

Also, with the rapid development of Internet there is need to integrate its applications which provide value-added services and extra comfort to trips

With the evolution of the automotive industry, sensors and Global Positioning System (GPS), vehicles are now equipped with various supporting technologies that can aid the driver. With the growth of mobile computing and communications, such information can be shared between vehicles and individual data can become a collective effort to support traffic in general. Also, with the cooperation of other systems one can enable the deployment of various types of applications and services, which can include: road and weather status information, congestion control, etc. Also, Internet applications, e-commerce or even vehicle operation services (such as vehicle inspection) can be included and further developed.

It is with this in mind that we proposed a solution for a services and applications platform. This platform provides means for their development in an easy and seamless way. It also supports communication between vehicles and existing systems, as well as facilities to easily obtain information of the surrounding environment and for the deployment of applications. One major contribution of the platform is the fact that - as we will see - it does not use a dual-stacked approach on the network and transport layer like most of the existing solutions, but rather makes use of a location protocol and runs it over the typical TCP/IP stack used on the Internet.

This document is divided as follows: Section 2 discusses the Related Work, Section 3 and Section 4 denotes the Architecture and Implementation Details, respectively, Section 5 shows the Results of

the performed tests and finally, we conclude and make the final remarks on Section 6.

2. Related Work

In this section we will discuss some of the most relevant standards, technologies and research projects that contribute to the design of a Vehicular Ad-hoc Network (VANET) architecture, as well as, the design of a platform to support communication within such architecture.

2.1. Mobile standards and technologies

IEEE 802.11 has been around for quite awhile and it is currently one of the most used wireless standard in the world. Depending on the needs - such as QoS guarantees or bandwidth - variations on the standard have been proposed, e.g., 802.11a/b/g/n. Unfortunately, and as shown in [6, 11], these standards are not fitting for a VANET environment, mainly because it was designed for low mobility network - such as people walking with their wireless devices. With this in mind, the ASTM and IEEE adopted the Dedicated Short-Range Communications (DSRC), which is based on the IEEE 802.11a PHY and MAC layer with a few modifications, in order to maximize reliability and provide a more robust and secure mean of transmission for an automotive environment [9]. DSRC is currently standardized as the IEEE802.11p protocol and it is used within the Wireless Access in Vehicular Environments (WAVE) standard.

WAVE defines an architecture, communications model, management structure, addresses security needs and physical access. Its architecture is composed of an OBU, RSU and a WAVE interface. The WAVE operation is regulated by the IEEE 802.11p and the IEEE 1609.x family of standards and it operates below a few management protocols. For more details, we refer the reader to [12].

Finally, the Communication Access for Land Mobile (CALM) architecture is a set of standards that support communication media and application diversity and allows all communication scenarios, such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, multi-hop, unicast and multicast. Its communication system is composed of four sub-systems: roadside, vehicle, central and personal, each one of them has their own instances, each including an Intelligent Transportation Systems (ITS) station performing CALM-specific functions. Services and applications run over a Communication Kernel - which is used to exchange information between stations - such kernel is composed of a CALM: Manager, Networking and Media. The CALM Manager is the core of each ITS station and it ensures that a given data flow is matched to a given communication medium according to a couple of parameters and configurations. The CALM Networking layer is composed of both IPv6 and ITS-specific protocol called CALM FAST for non-IP communications and includes mobility management protocols, like NEMO. Finally, the CALM Media implements reliability parameters indicating the quality of the link. Each CALM radio channel is virtualized and, as a result, this layer can access its properties and set them on a packet-by-packet basis. For further details, we refer the reader to [2, 8]

2.2. Research Projects and Discussion

In the following section, we will give an overview of the following research projects: Co-operative Vehicle-Infrastructure Systems (CVIS), Network on Wheels (NoW) and Safe Intelligent Mobility (simTD). These were chosen among many others (Safespot and COOPERS [1] or WiSafeCar [11], to name a few) because they propose a solution that intends to answer the same problem as ours - that is to provide a suitable VANET platform.

CVIS aims to enable a flexible, harmonised and open communication between its architecture components - vehicles, roadside system, central system - to improve existing services and develop new ones [7]. Its involved parties use a multi-homed approach for a continuous network connectivity, involving infrared light, radio communication based on WAVE, radio communication at frequencies above 40GHz, 2G/3G cellular radio technology and DSRC. All of these technologies are unified and integrated by the CALM standard [2, 1]. CVIS conducted several tests to analyse the performance and handover behaviour of the involving technologies. Such study [8] demonstrated that a vehicle travelling towards a RSU and away (at 100km/h) could achieve throughputs of 5Mb/s with packet sizes of 1KB and that handovers would not interfere with the applications' session.

NoW has two involving parties: vehicles equipped with OBUs and fixed stations alongside the road (RSU). Such design enables V2V and V2I communication. NoW divides applications between safety (e.g. avoid car crashes) and comfort (e.g. web browsing). Each of these have different requirements and needs and, so, it uses different approaches for both of them [3]. Its platform uses a dual stack approach in which: safety-related applications use a novel network and transport protocols that provide ad-hoc communication and multi-hop - such as GeoBroadcast, GeoAnycast, GeoUnicast and Topologically-scoped Broadcast - among OBUs and RSUs over a IEEE802.11p radio protocol; comfort-related applications use the traditional TCP/IP protocol stack over IEEE802.11a/b/g pro-

tocols. Although each type of application has its own stack, that does not mean that comfort-related applications cannot use the stack of its counterpart. This way, one can define basic systems and extend already existing ones [3]. The results achieved by NoW contributed to activities of standardization bodies and the presented platform was used within other working groups such as: Safespot, SEVECOM and Aktiv.

Finally, simTD aimed at the enhancement of components from previous projects (like the NoW project), as well as, the development of new ones. Its system architecture is composed of a vehicle domain and an infrastructure domain, and thus, it uses V2V and V2I communication. simTD vehicle station architecture is composed of two components: the Communication Control Unit (CCU) and the Application Unit (AU). The CCU integrates all communication modules and offers different abstraction layers for application access and is connected with the AU using Ethernet, standard protocols and proprietary adaptations. It uses IEEE802.11p, Universal Mobile Telecommunications System (UMTS) and GPS-based positioning. Based on NoW, it uses message queuing and forwarding schemes and includes congestion control. Above this layer, there is the transport layer and the facilities layer. The latter one enables an abstraction of V2V and V2I communication for the AU component [10].

As we have seen, the presented research projects share a common approach on the VANET architecture: the use of V2V and V2I communication. We believe that both are needed for various reasons. Particularly, the use of V2V is necessary in order for time-critical applications to experience low delays when transmitting important data.The use of V2I is necessary in order to expand the transmission range of a given vehicle and to connect it to other existing services and networks. As for the approach taken in the network layer, our viewpoint diverges. The use of a dual stack network layer has the following problems: uses too much resources as it is more complex and thus less adequate for hardware with low computational power and has more processing and message header overhead. We propose the use of a proper geo-routing protocol and location-based communication and a unique stack instead of two, thus eliminating the aforementioned issues. The same applies to the dual stack transport layer approach.

3. Architecture

In the previous sections, we have discussed the existing problems of a VANET environment, as well as, the different approaches and solutions taken by either standardization groups and/or research projects. In this section, we will explain our efforts in building a supporting platform for applications and services in a vehicular environment. But before hand, we will describe, yet, another platform - the Multi-Functional Platform - in which ours works over, and then describe our Service Platform.

3.1. Multi-Functional Platform

The multi-functional platform [4] is a modular platform that provides means for the development of applications with great flexibility and reconfigurability. Figure 1 denotes its architecture.

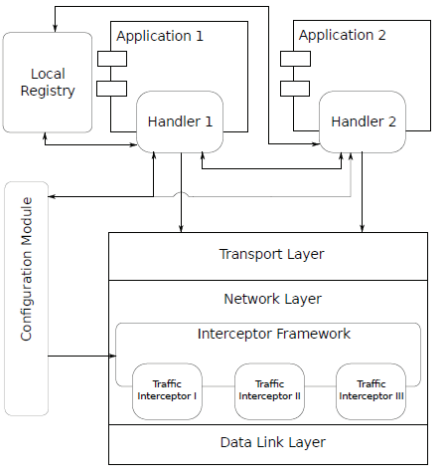


Figure 1: Multi-functional platform architecture overview

It is based on a group of key components that enable the development and configuration of applications. Such components are: *interceptor framework*, *handler factory* and a *configuration module*. The *interceptor framework* is based on a plug-in scheme that offers means for traffic manipulation by registering a set of rules (interceptors) in a rule manager. Such rules are composed of several filters that describe when and which packets should be intercepted. This scheme allows the implementation of features like security or packet manipulation. The *handler factory* creates handlers which are the main structure of the platform. The handlers goal is to abstract applications from transport details and traffic manipulations. Applications instantiate these handlers and configure them based on the type of transport they require. The platform provides two transport schemes, a unreliable and a reliable transport protocol based on a NACK scheme. A local registry is used to store the unique identifiers and the respective addresses of these handlers, in order for local and network communication to occur. Finally, the *configuration module* stores information about parameters used by each handler, such as how many messages the transport can

buffer.

3.2. Service Platform

When designing the platform, there were certain aspects that needed to be taken into account: (i) **scalability** - the system should work either in a sparse environment, as well as, in dense environments; (ii) **configuration** - the system should be configurable and allow as much configuration options, as possible; (iii) **flexibility** - applications should be provided with as much mean of communication and surrounding environment information as possible; (iv) **extensibility** - the system should provide ways to improve, extend and add features.

Taking into account the aforementioned requirements, we have designed a Distributed Architecture with asynchronous communication, in which, each component has a main focus. It is composed of four layers: Facilities Layer, Transport/Network Layer and Management. Figure 2 denotes our architecture and its integration with the multi-functional platform.

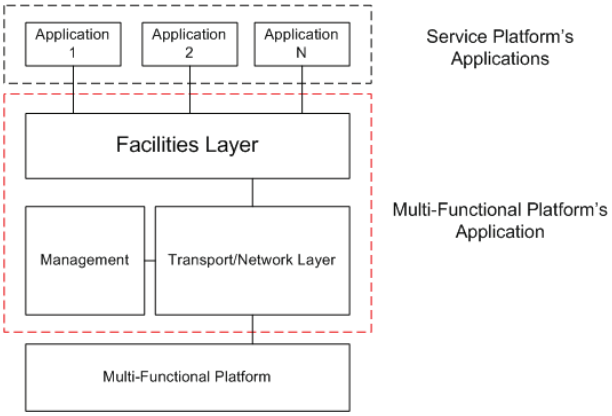


Figure 2: Service platform architecture overview

Further explaining, the *Facilities Layer* abstracts applications from lower layers by providing an API that facilitates messaging, neighbourhood information - such as surrounding vehicles and their correspondent speed, location, etc. - and real-time receival of nearby vehicles' information and data. The *Transport/Network Layer* is the core of the platform as it provides the typical TCP-based communication, a reliable scheme using NACKs and an unreliable scheme (UDP-based) for data transmission. It also provides means for broadcast, geocast and unicast transmission using location-based communication, as well as information of the surrounding environment. Finally, the *Management* is used to store the platform configuration, such as the time interval between the transmission of the vehicle's information (denoted as HELLO messages).

3.2.1 Management

The Management stores configuration parameters, so that the Transport/Network Layer can configure its components. Such parameters are as follows: an alphanumeric unique *ID*; *number of hops* which tells how many hops-distance should the platform store/send a given HELLO message; *timings* indicating the interval between each HELLO message and its removal; *data aggregation* which indicates the algorithms that should be used to aggregate the stored information; *cache cleanup* indicating the interval between each cache cleanup (applications' data); *buffer size* indicating the size of the local applications' data buffer.

3.2.2 Transport/Network Layer

The Transport/Network Layer is the core of the platform, and its architecture is shown in Figure 3. It is composed of: a *Network Manager*, *Table Manager*, *Sender/Receiver Module* and *Protocols*.

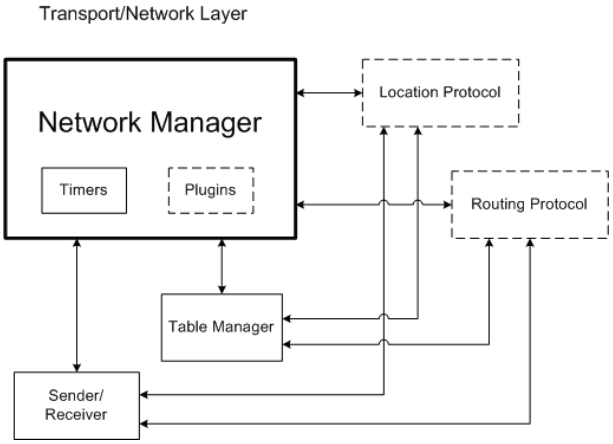


Figure 3: Transport/Network Layer components

As mentioned before, this architecture is mainly a Distributed System with asynchronous communication, in which parts of the layer can be replaced with similar components that can fulfil the task at hand. Such replaceable parts are denoted with dashed lines in Figure 3.

The main focus of the Network Manager is distributing all of the data between each of the components. Its timers are meant to either send the HELLO messages or to warn the Table Manager to remove any old instances of other vehicles and/or RSUs. The plugins are extensions to the platform and allow to subscribe to any received data and alter the times of each of the timers. Such subscription is made possible because of the design of the Sender/Receiver Module.

The Sender/Receiver Module is a simple component that communicates with the Multi-Functional

Platform and it is meant to send internal or receive external data. Because we have designed a Distributed System, we have developed a publish/-subscribe scheme within this module, allowing its components to receive data regarding HELLO, application data or protocol data - with a few restrictions, like protocols not allowed to receive application data and vice-versa.

The Table Manager is meant to store information regarding its own vehicle and the surrounding environment information at a hop-distance. Although the data is discretized, it is this component that makes use of the aggregation algorithms, so that it can aggregate data of the surrounding entities when sending data of its neighbours and thus allowing to send as much information, as possible within one message.

Finally, the Protocols are of two types: *Routing Protocols* and *Location Protocols*. They work closely with the Network Manager in order to find the location of a given vehicle/RSU or to route a given message. Such protocols can be replaced without altering the logic behind the Network Manager and thus allowing for the development of smarter and more efficient protocols.

3.2.3 Facilities Layer

The Facilities Layer abstracts applications from the lower layers. It includes a mechanisms to register with the platform, allow communication with other entities, as well as, subscription to received HELLO messages and requesting neighbours and their locations with the Table Manager.

4. Implementation Details

The implementation of the platform was made using the C programming language, alongside with Extensible Markup Language (XML) for configuration parameters and JavaScript Object Notation (JSON)-based formats for external and internal communication.

In order to understand how the Network Manager and the Protocols work together, we will show the implemented algorithms for the various types of message transmission.

Algorithm 1 shows the procedures used for broadcast. As observed, it is very simple and it is based on broadcasting a given message for a distance of N hops. The number of hops is defined by the application and this way we can avoid endless rebroadcasts. Another mechanism used to avoid network flooding and retransmitting the same message is storing them in a local cache. In order to avoid spending a great amount of memory in such storage, hashing and bloom filters are used.

Algorithm 2 shows the procedure for geocast transmission. The procedure is similar to broadcast in a way that uses the same mechanisms, but with the following differences:

1. if the vehicle is not within the range of the given location (in GPS coordinates) indicated by the received message, then the message is delivered to the routing protocol that will decide to which node should the message be forwarded;

2. if the vehicle is within range, then a broadcast procedure is started until the vehicles in the given radius are warned. In order to avoid flooding, the same cache procedure is used.

The fact that the decision of forwarding is transferred to the protocol, allows the detachment of the Network Manager from routing decisions, and, since the protocol can be changed, the platform can make better decisions with the improvement of such protocol without altering the Network Manager procedures.

Algorithm 1 Broadcast Algorithm

```
Require: code, message
1:  if code > 0 then                          ▷ Local generated message
2:      broadcast_message();
3:  else                                      ▷ Remote generated message
4:      if own_message() OR in_cache() then
5:          return;                           ▷ Discard message
6:      else
7:          message_hops ← message_hops − 1;
8:          broadcast_message();
9:          store_message();                  ▷ Store message in cache
10:     end if
11: end if
```

Algorithm 2 Geocast Algorithm

```
Require: code, message
1:  if code < 0 AND own_message() then
2:      return;
3:  end if
4:
5:  if in_cache() then
6:      return;
7:  else
8:      if vehicle_within_range() then
9:          broadcast_message();
10:     else
11:         send_to_routing_protocol();
12:     end if
13:
14:     store_message();
15: end if
```

Algorithm 3 shows the procedures used for unicast. While the previous algorithms always use an unreliable transmission channel, unicast can make use of the three types of transport provided by the Multi-Functional Platform. In all three cases, the algorithm is the same, only the transmission channel is changed.

The algorithm works as follows:

1. if the location of the destination vehicle is known, then the message is delivered to the routing protocol to forward the message;

2. if the location of the destination vehicle is not known, then the message is delivered to the location protocol, which will attach a location to the message, re-deliver it to the Network Manager and then, the previous item is started.

Once again, the fact that the Network Manager is detached from location decisions, allows for the platform to be loaded with better location protocols without altering the Network Manager procedures.

Algorithm 3 Unicast Algorithm

Require: *code*, *message*

```
1:  if code > 0 then
2:      found = check_local_table();        ▷ Check if destination is any of the neighbours
3:      if found == true then
4:          send_to_node(); ▷ Forwards the message to neighbour
5:      else
6:          if NOT message_has_location() then
7:              send_to_location_protocol();
8:          else
9:              send_to_routing_protocol();
10:         end if
11:     end if
12: else
13:     if NOT message_for_local_vehicle() then
14:         found = check_local_table();
15:         if found == true then
16:             send_to_node();
17:         else
18:             send_to_routing_protocol();
19:         end if
20:     end if
21: end if
```

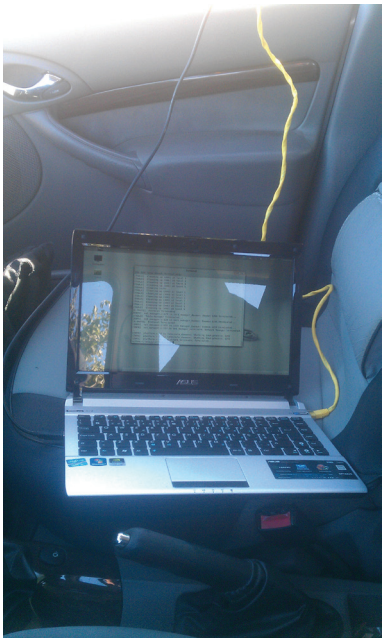
As for application development, we have defined an Application Programming Interface (API) for a key group of functionalities that allows the abstraction of communication, subscription with the Sender Module for HELLO messages and querying the Table Manager for any given neighbour.

5. Results

5.1. Equipment and software tools

To setup our test-bed, we used laptops running Linux operating system (Ubuntu 11.04) equipped with SMCWUSBS-N3 802.11b/g/n wireless pens (configured to run IEEE802.11g). We installed our platform within the laptops and developed a small application (using our facilities API) which, based on a configuration file, would either, send traffic to the network using our platform or receive the same transmitted traffic. Figure 4 shows our test-bed montage.

(a) Test-bed environment



(b) Laptop on car

(c) Wireless pen montage on car

Figure 4: Hardware and test-bed environment

5.2. Description of the tests

Our tests focused mainly on three important characteristics: I2V and V2I communication, store and forward and multi-hop. To do so, we performed the following test.

Observing Figure 5, the OBU (our vehicle) would start in point A to gain velocity (50 km/h would be our goal) and when reaching the RSU-B (at point B - 57m from point A), the transmission would begin from RSU-B to OBU. At this point, the OBU has not received any HELLO messages from RSU-D (at point D - 163m from point A) and so it does not know the existence of the messages' destination and thus, it would store them. Upon reaching RSU-D's transmission range (roughly at point C - 83m from point A) it would transmit the stored messages and then transmit all received data from RSU-B at the same rate that they are transmitted. The OBU would then stop at point E, turn back to point A and the test would be repeated. Point E was at 260m distance from point A.

We ran 4 different tests (shown in Table 1) with UDP, each with 5 runs, and thus making 20 travels.

Figure 5: Test route

Test	Packet Size (B)	Tx Rate (msgs/s)	Throughput (B/s)
1	64	16	1024
2	64	80	5120
3	512	4	2048
4	512	20	10240

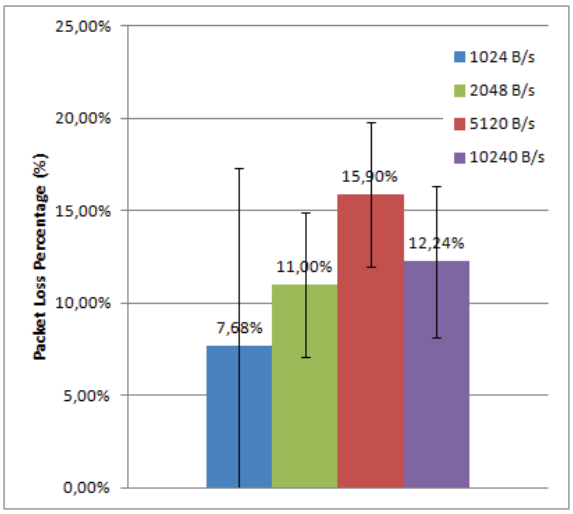
Table 1: Performed experiments

5.3. Results and analysis

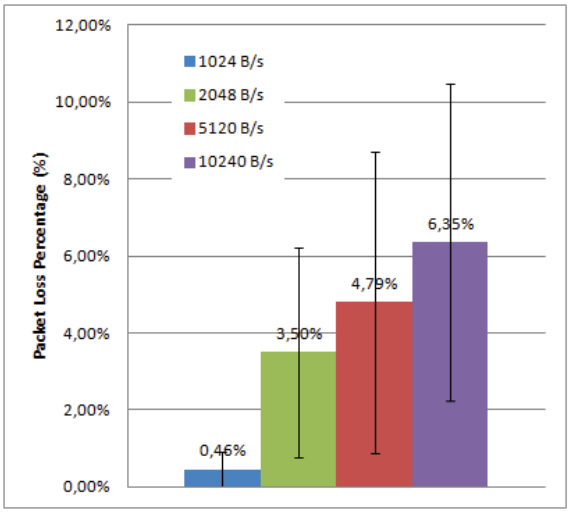
The performance metrics used to evaluate the connectivity between each involved party are: (i) **Packet Loss**: percentage of packet loss due to connectivity problems; (ii) **Latency**: the time that took a given message to travel to its destination; (iii) **Jitter**: the variation of the latency in the network.

Figure 6 shows an average packet loss (with error bars representing the calculated trust interval of 95%) for each test. We have discretized the total packet loss into two other plots, in order to understand where the most packets were lost. As we can see, The RSU-B transmitted, without great problems, most of the packets to the OBU, while the OBU and the RSU-D had the more losses. This could be because of the used protocol (802.11g instead of 802.11p) which is not meant for vehicular usage. Using the proper standard, the packet loss could decrease. Also, the surrounding environment could also have its influence; the street had a slight elevation which prevented the messages from being perfectly delivered, but also because the RSU-D was near a house with metal surroundings which could interfere with the transmission of the messages. Particularly, for 5120B/s case, in one of the runs other vehicles were passing by, during the execution of the test, thus increasing the packet loss percentage (to 15.58%), and therefore ruining the calculated average (which at the time was around 5%).

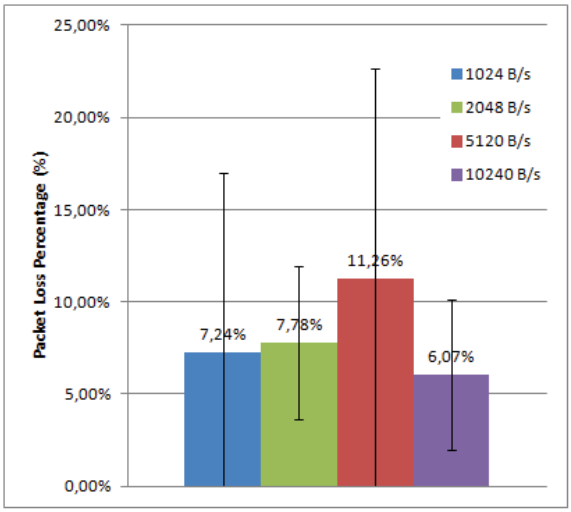
Figure 7 shows the behaviour of one of the performed tests. The points 1 and 2, show the store and forward behaviour of our platform; while in 1, the OBU had no knowledge of the RSU-D's existence, in 2, the OBU erased the RSU-D's reference from its local table and missed out a couple of HELLO messages. As such it stored the transmitted messages until the reception of a new HELLO message. The circled points show which packets



(a) Total Packet Loss



(b) From RSU-B to OBU Packet Loss



(c) From OBU to RSU-D Packet Loss

Figure 6: Packet Loss Total (a), from RSU-B to OBU (b) and from OBU to RSU-D (c)

were lost during the execution of the test and at the end of the plot, there were 3 undelivered messages that were stored within the OBU, but could not be transmitted because the OBU lost the reference of the RSU-D.

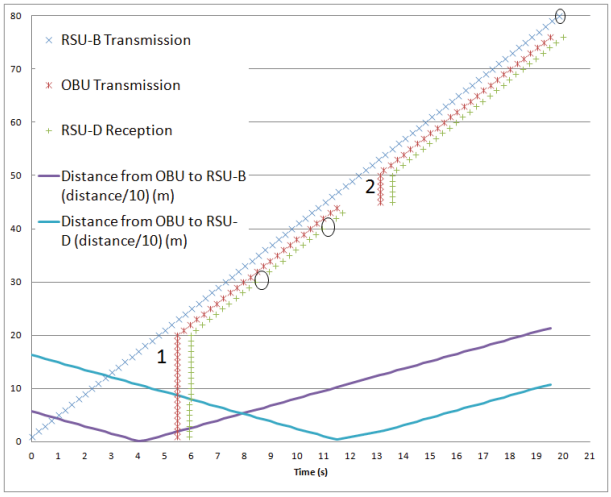


Figure 7: Packet Loss Distribution over Time

All of the presented graphics allow to calculate the desired performance metrics. Table 2 shows a summary of the packet loss with the respective trust intervals. From Figure 7 we can calculate the latency and jitter, which are 1,17s and 30,4ms respectively. It should be mentioned that the latency and jitter values are calculated only during a fraction of time that there was full connectivity between the RSU-B, the OBU and the RSU-D - that is, between 5 seconds and 12 seconds in. From the aforementioned calculations and values, we compared the packet losses with the study conducted by [5], analysed the other two metrics and verified that:

1. **Packet Losses**: Our packet losses are much lower than the ones presented in the study (ranging from 5% to 60%). This may be due to the fact that, although they used their RSU in a higher ground (which gives them a better transmission range), they also blocked the transmission with a bridge and thus increasing the packet loss percentage;

2. **Latency**: Our latency is around 1,17s and such values may be due to the fact that the laptops' clocks were not perfectly synchronized and thus increasing the value of this metric. So, we performed a ping test with a 64B data and the presented values fell into more realistic values; from the OBU to RSU-B we had an average of 3,437ms (with a deviation of 1,263ms) and from the OBU to RSU-D we measured an average of 6,792ms (with a deviation

	1024 B/s	2048 B/s	5120 B/s	10240 B/s
From RSU-B to OBU	0,46%	3,50%	4,79%	6,35%
Trust Intervals (95%)	0,44%	2,73%	3,92%	4,12%
From OBU to RSU-D	7,24%	7,78%	11,26%	6,07%
Trust Intervals (95%)	9,77%	4,18%	11,37%	4,09%
Total Packet Loss	7,68%	11,00%	15,90%	12,24%
Trust Intervals (95%)	9,66%	4,92%	10,01%	3,19%

Table 2: Packet Loss summary

of 4,431ms). These values are much more acceptable and plausible than those previously measured;

3. **Jitter**: From this metric, we can safely say that the presented values are acceptable and also, prove that the prior performance metric discrepancy is not due to lack of platform performance or bad measures, but to clock desynchronization.

6. Conclusions

In this study, we have presented the results and evaluation of the performance of I2V and V2I communication, alongside carry and forward mechanisms and multi-hop based on the IEEE802.11g technology. We have demonstrated that, it is possible and feasible to transmit data between two RSUs and an OBU without much loss and delay. Such experiments also proved the feasibility of extending the transmission range of an infrastructure by using multi-hop and carry and forward mechanisms. In the future and test-wise, we would like to test the platform with more vehicles and/or infrastructures, but also evaluate the TCP performance in a VANET environment. Implementation-wise, we would like to greatly improve our platform by either substituting the Multi-Functional Platform for a complete implementation of communication mechanisms or by improving its integration with our platform. Also, and given the easiness in developing protocols with our API, we would like to also refine the existing ones by taking into account other crucial factors (such as distance).

Acknowledgements

This work is supported by FCT (INESC-ID multiannual funding) through the funds of Programa PIDAC.

References

[1] J. Boussuge and C. Laurgeau. Comparative synthesis of the 3 main European projects dealing with Cooperative Systems (CVIS, SAFESPOT and COOPERS) and description of COOPERS Demonstration Site 4. *Traffic*, pages 809–814, 2008.

[2] T. Ernst, V. Nebehaj, and R. Sorasen. CVIS : CALM Proof of Concept Preliminary Results. *Media*, (December):80–85, 2009.

[3] A. Festag, G. Noecker, M. Strassberger, A. Lübke, and B. Bochow. NoW Network on Wheels : Project Objectives , Technology and Achievements. *Transportation*, (March):211–216, 2008.

[4] F. Gonçalves. Multi-Functional Platform for Indoor and Outdoor Monitoring. 2011.

[5] M. S. S. Jerbi, Moez. Characterizing Multi-Hop Communication in Vehicular Networks. 2008.

[6] A. Khan, S. Sadhu, and M. Yeleswarapu. A comparative analysis of DSRC and 802 . 11 over Vehicular Ad hoc Networks. *Ad Hoc Networks*.

[7] E. Koenders and J. Vreeswijk. Cooperative Infrastructure. pages 721–726, 2008.

[8] M. E. G. Moe, V. Nebehaj, and T. Ernst. CVIS Performance Test Results : Fast Handovers in an 802 . 11p Network. *Public Roads*, 2010.

[9] L. Stibor, Y. Zang, and H.-J. Reumerman. Neighborhood evaluation of vehicular ad-hoc network using IEEE 802.11 p. *Proceedings of the 8th European Wireless*, pages 1–5, 2007.

[10] H. Stübing, M. Bechler, D. Heussner, T. May, I. Radush, R. Horst, and P. Vogel. sim TD : A Car-to-X System Architecture for Field Operational Tests. *IEEE Communications Magazine*, (May):148–154, 2010.

[11] T. Sukuvaara, P. Nurmi, M. Hippi, R. Autio, D. Stepanova, P. Eloranta, L. Riihentupa, and K. Kaubo. Wireless Traffic Safety Network for Incident and Weather Information. *Network*, pages 9–14, 2011.

[12] Y. Toor, P. Muhlethaler, A. Laouiti, and A. De La Fortelle. *Vehicle Ad-hoc Networks: Applications and related technical issues*, volume 69. May 2008.

Simulação do uso de redes veiculares em situações de emergência numa auto-estrada Portuguesa

Jacqueline Jardim, Inesc-ID/IST
Teresa Vazão, Inesc-ID/IST
Jorge Lopes, Brisa Inovação

***Resumo* – As redes veículares (a.k.a.) VANETs (*Vehicular Ad-Hoc Networks*) oferecem novas formas de aumentar a segurança rodoviária através da disseminação de informação relativa a condições de circulação adversas ou a acidentes. No entanto, a disseminação de informação crítica para a segurança e bem-estar dos condutores tem elevados requisitos de tempo e fiabilidade, uma vez que as mensagens têm que ser recebidas, atempadamente, por todos os veículos envolvidos numa situação potencialmente perigosa, de forma a assegurar que medidas adequadas são tomadas para prevenir que a situação se concretize/escale. Garantir que tais requisitos são cumpridos corresponde a um desafio considerável devido à deterioração do desempenho característico dos sistemas de comunicação sem fios. Torna-se, assim, crucial recorrer a técnicas de simulação fiável e em larga escala para validar o conceito, numa fase que antecede a implementação de tais sistemas e integração dos mesmos na indústria automóvel.**

Neste artigo, é realizado um estudo de simulação realística e em larga escala de uma situação de emergência, baseado em dados reais de tráfego rodoviário recolhidos numa auto-estrada Portuguesa. Foram avaliados aspectos tais como a inclusão de unidades fixas de comunicação ao longo de um troço da auto-estrada e o desempenho da rede veicular ao notificar todos os veículos, direta- ou indiretamente, envolvidos nos diferentes cenários de acidente modelados neste mesmo troço. Após uma análise de resultados, verificou-se que, de uma forma geral, obtém-se um menor atraso na recepção da notificação de acidente quando as unidades fixas de comunicação são incluídas na infra-estrutura de rede. Esta melhoria no desempenho da rede não só permitirá com que os condutores dos veículos mais próximos do local de acidente reajam atempadamente e em segurança, como os que se encontrem mais distantes possam optar por sair da auto-estrada, de forma a evitar um eventual congestionamento de tráfego rodoviário.

***Palavras-Chave* — Acidente, Aplicações, Auto-estrada, *Geocast*, GPSR, I2V, Mobilidade, Modelação, Rede Veícular, Segurança Rodoviária, Simulação, RSU, VANET, Veículos, V2I, V2V.**

I. INTRODUÇÃO

O desenvolvimento de novas tecnologias de redes sem fios e a existência de sistemas embebidos de baixo custo e elevadas capacidades computacionais potenciou o aparecimento das redes veiculares, quer a nível de investigação, quer a nível de mercado. Este tipo de rede possibilita a comunicação entre veículos (*Vehicle-to-Vehicle* – V2V) e entre estes e a infra-estrutura rodoviária (*Vehicle-to-Infrastructure* – V2I).

O principal interesse nas VANETs surge pela possibilidade de utilização de novos paradigmas de segurança rodoviária, baseados na cooperação entre as diversas entidades envolvidas na comunicação, que permitam melhorar, de forma significativa, a segurança rodoviária e promover a mobilidade sustentável. Todavia, dada a criticidade deste tipo de aplicações, são necessários estudos de simulação em ambientes de grande escala e condições tão próximas quanto possível da realidade, para que se possa verificar as potenciais vantagens da tecnologia antes de se iniciar a sua introdução nos veículos e nas infra-estruturas.

Este trabalho pretende analisar a exequibilidade do uso de redes veiculares em cenários de auto-estrada para resposta a situações de acidente. Pretendem-se avaliar três aspectos diferentes:

- O impacto do uso de *Roadside Units* (RSUs) no processo de notificação de acidente, de forma a avaliar a necessidade de investir na sua instalação.
- A capacidade de avisar atempadamente os veículos que se encontram próximos da zona de acidente, de forma a evitar choques em cadeia.
- A capacidade de avisar os veículos que se encontram longe da região acidentada, de forma a garantir que estes possam escolher uma rota alternativa, minimizando o congestionamento de tráfego.

O estudo apresentado em simulação baseia-se no caso real da auto-estrada Portuguesa A5 (Costa do Estoril), a qual interliga a capital, Lisboa, à Costa do Estoril e Cascais. Esta auto-estrada constitui umas das várias auto-estradas sob a responsabilidade da Brisa Concessão.

II. ESTADO DA ARTE

A. *Aplicações de segurança*

O desenvolvimento das redes veiculares possibilita o desenvolvimento de novos tipos de aplicações de segurança rodoviária. Estas aplicações têm por base a cooperação e partilha de informação entre os veículos e o ambiente envolvente, e têm por objectivo alertar o condutor de situações que condicionam as condições de segurança e mobilidade ao longo da viagem.

Em [1], *Toor et al*, efectua-se uma caracterização dos diversos tipos de aplicações e concluiu-se que as aplicações de segurança devem ser utilizadas essencialmente para apoiar em situações de acidente, fornecer informação em cruzamentos e evitar congestionamentos de tráfego. No entanto, são deixadas muitas opções em aberto em relação à arquitectura de protocolos e mecanismos de comunicação mais adequados.

O estudo apresentado em [2] caracteriza mais detalhadamente os tipos de aplicações de segurança rodoviária, definindo aplicações para cinco propósitos diferentes:

- Alerta para características perigosas na infra-estrutura.
- Alerta para condições anormais de circulação.
- Alerta para o perigo de colisão.
- Aviso de choque iminente.
- Notificação de acidente.

Segundo o mesmo estudo, este tipo de aplicações requer a utilização de novos mecanismos de comunicação que permitam enviar informação para um conjunto de nós não especificado: a disseminação dentro duma área geográfica (*Geocast)* e a disseminação periódica para os nós adjacentes (*Beaconing*). A comunicação *multi-hop* e o *store-and-forward* são ainda utilizados para garantir a recepção de informação por nós que se encontram fora do alcance inicial [3][4] e a correlação para reduzir o tráfego de dados, especialmente em situações de elevada densidade de veículos.

Um aspecto determinante para o desempenho destas aplicações está relacionado com a definição da abrangência do *Geocas*t e com o tempo de validade da informação de segurança. Em [5], os autores estipulam como valores aceitáveis para um alcance máximo duma comunicação *Geocast* os 250 m e como limite temporal de validade da informação os 10 s. Todavia, um estudo experimental realizado no contexto do projecto Europeu *Cooperative Vehicle-Infrastructure Systems* (CVIS) com um conjunto de aplicações de segurança desenvolvidas pelo consórcio concluiram que o tempo de aviso não deveria ultrapassar os 5 s [6].

Para as aplicações destinadas a alertar os condutores sobre potenciais acidentes, existem outros factores determinantes para o seu sucesso, como sejam a precisão da localização do veículo e da previsão do seu movimento, os quais estão directamente relacionados com o período de tempo entre *beacons*. Estudos reportados em [7] demonstram que uma frequência de 5 Hz garante um desempenho adequado para este tipo de aplicações.

Estudos de desempenho sobre aplicações para evitar acidentes em cruzamentos têm sido também realizados por diversos autores, tais como [8][9]. Todavia, estes estudos não se aplicam ao cenário duma auto-estrada, dados os diferentes padrões de mobilidade e características do próprio cenário.

B. Aplicações de segurança para situações de emergência

Uma situação potencialmente perigosa pode desencadear a transmissão de mensagens geradas por diversas aplicações de segurança rodoviária, das quais as mais relevantes para um cenário de acidente são:

- Aviso de travagem brusca (*Emergency Electronic Brake Lights* - EEBL).
- Notificação de acidente (*Post-Crash Warning* - PCW).
- Alerta para o perigo de colisão iminente (*Cooperative Collision Warning* - CCW).

A aplicação EEBL permite que um veículo notifique os veículos à sua retaguarda quando trava subitamente. É especialmente útil em condições de fraca visibilidade, em que os veículos podem não aperceber-se, atempadamente, de que o veículo da frente travou/activou as luzes de travagem. A aplicação PCW notifica os veículos que se aproximam de um local de acidente da presença de um veículo imobilizado, devido a um acidente ou avaria mecânica. Por último, a aplicação CCW mitiga a ocorrência colisões ao enviar informação periódica acerca da posição, velocidade, aceleração, direcção de cada veículo.

Segundo [10] e [11], estas aplicações podem ser caracterizadas de acordo com diferentes parâmetros, conforme se ilustra na Tabela I.

TABELA I
CARACTERIZAÇÃO DE APLICAÇÕES DE SEGURANÇA RODOVIÁRIA

	EEBL	**PCW**	**CCW**
Modo de Comunicação	Geo-broadcast	Geo-broadcast	Geo-broadcast
Cardinalidade	Unidirec.	Unidirec.	Unidirec.
Tipo de Comunicação	V2V	V2I, V2V	V2V
Modo de transmissão	Por evento	Por evento	Periódico
Freq. mín. mensagens (Hz)	~10	~1	~10
Latência máxima (s)	0.1	0.5	0.1
Alcance (m)	~300	~300	~150

III. CARACTERIZAÇÃO DO CENÁRIO

A. Informação geral

A auto-estrada A5 é considerada uma das rodovias mais congestionadas do país, principalmente em hora de ponta, uma vez que é um dos principais eixos de acesso à cidade de Lisboa. Nos 25 quilómetros entre Lisboa e Cascais a A5 inclui 12 intersecções com 64 ramos de acesso.

Em termos tecnológicos, a A5 é uma auto-estrada equipada com uma infra-estrutura avançada, constituída por um *backbone*, que interliga um conjunto de sensores, câmaras de videovigilância (CCTV) e painéis de mensagens variáveis (PMVs). Os postes de CCTV estão colocados, lateralmente, em locais que oferecem uma boa visibilidade e os PMVs estão colocados em pórticos sobre-elevados, perpendiculares à A5. Em qualquer dos casos, estes sistemas estão espalhados ao longo de toda a auto-estrada, sendo colocados a cerca de 6 m de altura. Existem sensores nas entradas e saídas, e sensores localizados ao longo dos diversos troços da A5. Actualmente, esta infra-estrutura é utilizada para recolher informação de tráfego, identificar situações de perigo, detectar acidentes e fornecer informações aos condutores sobre as condições de circulação. Toda a informação é centralizada no Centro de

Coordenação Operacional - CCO, que efectua a gestão de todas as auto-estradas concessionadas pelo grupo Brisa.

A diversidade de equipamento de sensorização disponível permite recolher informação de natureza diversa, tal como a intensidade e a densidade do tráfego, a classe e peso dos veículos, e a velocidade média. Esta informação é enviada e processada nos sistemas centrais para processamento estatístico e enviada, em tempo quase-real, para os PMVs.

B. Caracterização do tráfego

Existe um registo histórico de informação de tráfego que permite modelar o tráfego de forma macroscópica [1]. Este registo contém a indicação da intensidade média de tráfego em diferentes locais, medida a cada 10 minutos. Complementarmente, existe também um registo da matriz origem-destino para cada um dos acessos da A5. Tendo por base a análise da informação registada foi possível determinar a intensidade de tráfego em diferentes troços ao longo do dia. A Tabela II resume os valores obtidos nos três troços mais representativos, em diferentes períodos do dia: o período de menor tráfego (2:00-3:00 h), o período de maior tráfego (8:00-9:00 h) e um período de tráfego médio (13:00-14:00h).

TABELA II
INTENSIDADE DO TRÁFEGO NA A5 (VEH/H)

	2:00-3:00h	**8:00-9:00h**	**13:00-14:00h**
Cascais – Alvide	18	1076	526
Carcavelos – Oeiras	25	2571	1501
Linda-a-Velha– Miraflores	222	6300	1850

C. Caracterização das condições de acidente

Para além do histórico de tráfego, a Brisa dispõe ainda de informação que permite identificar os locais mais propensos à ocorrência de acidentes e definir as causas mais frequentes. De acordo com dados disponibilizados pela Brisa, existem cinco zonas mais críticas.

A primeira zona ocorre no troço Cascais-Alcabideche, no sentido Cascais-Lisboa, entre as saídas 9 e 10. A causa principal de acidentes consiste na combinação de ventos fortes e excesso de velocidade, que resulta no despiste do veículo e, possivelmente, em acidente. A segunda zona situa-se junto à CREL, nas proximidades da saída 6, no sentido Lisboa-Cascais. Nesta zona, os acidentes são causados, principalmente, pela variação súbita da densidade de tráfego na curva que precede imediatamente a saída. Os condutores que se deslocam em excesso de velocidade não estão cientes da formação rápida de uma fila junto à saída e são obrigados a reduzir a velocidade bruscamente. Os dois locais seguintes situam-se próximos da saída para o Estádio Nacional: a curva apertada, aliada ao excesso de velocidade dos veículos são a principal causa de acidente. A última zona está localizada perto da saída 5, no sentido Lisboa-Cascais, em direcção a Carnaxide/Linda-A-Velha. Mais uma vez, a principal causa de acidente é o excesso de velocidade.

Conforme se pode constatar da descrição anterior dos cinco locais identificados, a causa mais comum de acidente é o excesso de velocidade. Todavia, em certas situações, os acidentes ocorrem em condições de tráfego elevado, enquanto noutras podem ocorrer mesmo quando o tráfego é esparso. Assim sendo, é fundamental modelar a ocorrência de acidentes em diferentes condições de tráfego.

IV. REDE VEICULAR DE SUPORTE ÀS APLICAÇÕES DE EMERGÊNCIA

A. Arquitectura da rede veícular

Um dos aspectos fundamentais a equacionar no desenho uma rede veícular prende-se com a necessidade de utilização de unidades de comunicação fixas (RSUs), devidamente posicionadas para aumentar a abrangência da comunicação, de forma a assegurar uma melhor conectividade.

No caso da A5, já existe uma infra-estrutura de rede, pelo que é relativamente fácil encontrar locais adequados à colocação das referidas RSUs, nomeadamente os locais de melhor visibilidade, onde estão, actualmente, colocadas as câmaras de CCTV ou os PMVs. Todavia, é necessário garantir que existe uma melhoria significativa de desempenho, que justifique o investimento a realizar nas RSUs. Assim sendo, serão considerados dois cenários distintos:

- Rede veícular sem RSUs que suporta apenas comunicação Veículo-a-Veículo (V2V).
- Rede veícular com RSUSs que também supporta comunicação Veículo-a-Infrastrutura (V2I/I2V).

Fig. 1 representa o caso mais completo, duma rede que contéms RSUs e que permite a comunicação V2V e V2I/I2V.

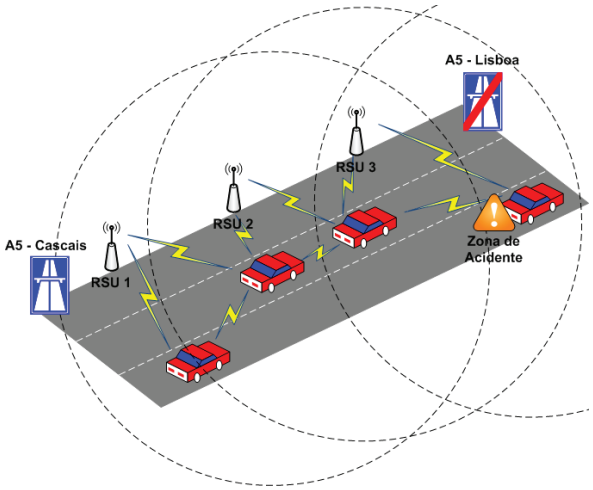


Fig. 1 – Arquitectura de rede de uma VANET

Numa situação real, os veículos e as RSUs poderão estar equipados com antenas com diferentes características, sendo aconselhável que as antenas das RSUs tenham um maior alcance e que estejam localizadas acima da altura dos

veículos de forma a evitar que estes funcionem como obstáculo à propagação do sinal [12].

B. Arquitectura dum nó

A Fig. 2, ilustra a arquitectura para os nós da rede, que deve ser semelhante para todos os nós, e deve ser adaptável a diversos tipos de aplicações. Assim sendo, veículos e RSUs suportam um conjunto de aplicações de segurança rodoviária, sendo a informação gerada por estas aplicações disseminada dentro duma dada área geográfica com recurso a um protocolo de transporte não fiável e a um protocolo de encaminhamento geográfico: o GPSR (*Greedy Perimeter Stateless Routing*) [13]. O protocolo de encaminhamento geográfico foi modificado para suportar a comunicação dentro duma área geográfica, limitada, ou não, a um conjunto restrito de nós. Em termos de acesso ao meio e transmissão física de dados, são utilizados os protocolos implementados na norma IEEE 802.11p, na qual os sistemas WAVE (*Wireless Access in Vehicular Environments*) se baseiam.

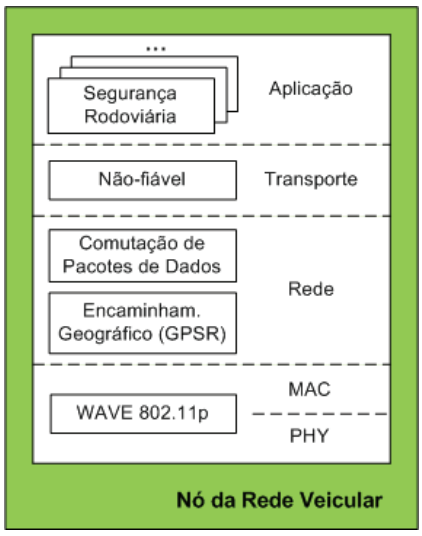


Fig. 2 – Arquitectura dum nó da rede

C. Aplicações de emergência

Em caso de acidente, é crucial garantir que, em primeiro lugar, o acidente não escale e, em segundo lugar, evitar que se formem longas e demoradas filas de trânsito, cujas caudas também podem implicar travagens súbitas, dando origem a segundos acidentes.

Considera-se que, no instante após acidente, o veículo notifica os que se aproximam pela sua retaguarda, caso os danos causados pelo acidente não o impossibilitem. Os veículos que se aproximam do local de acidente podem encontrar-se num ponto da auto-estrada que precede uma saída, pelo que, ao receber o aviso, podem antecipar a sua saída da auto-estrada. Os veículos que se encontrem muito próximos do local de acidente, a ponto de ter que reduzir drasticamente a sua velocidade quando percepcionam uma aproximação perigosa ao veículo dianteiro, devem transmitir avisos de travagem súbita/risco de colisão iminente na retaguarda aos restantes veículos.

Esta modelação permitirá analisar a coexistência de diversas aplicações de segurança rodoviária e o seu impacto na comunicação num cenário de emergência.

Do ponto de vista aplicacional, é necessário suportar as três aplicações definidas na secção II.B, nomeadamente:

- **Aplicação PCW** – usada para notificar a ocorrência dum acidente. O veículo acidentado gera notificações durante um curto espaço de tempo. Estas notificações são reenviadas pelos veículos que as recebem, de forma a garantir que a informação chega rapidamente para além da zona acidentada. No entanto, de forma a evitar situações de *broadcast storm*, como as que são descritas em [14], estas notificações são reenviadas apenas da primeira vez que são recebidas através dum mecanismos simples de correlação.
- **Aplicação EEBL** – usada para avisar os veículos duma travagem brusca do veículo da frente. O veículo que trava gera mensagens durante um curto espaço de tempo, a um ritmo relativamente elevado. Estas mensagens não são reenviadas pelos veículos que as recebem.
- **Aplicação CCW** – usada para detectar situações de potencial colisão com base na informação de localização que recebe periodicamente dos seus vizinhos. As mensagens geradas por cada nó tem um caracter exclusivamente local, não sendo reenviadas para os nós que não lhe são adjacentes.

Para uma resposta eficaz a uma situação de emergência é fundamental assegurar que, apesar da coexistência de tráfego de segurança rodoviária proveniente de diferentes aplicações, é possível cumprir os requisitos estipulados anteriormente para cada uma delas, nomeadamemte em termos da capacidade de notificar os veículos duma situação de acidente.

V. MODELAÇÃO DO CENÁRIO DE ACIDENTE

A. Caracterização do cenário de acidente

De forma a obter resultados fidedignos tentou-se reproduzir, em simulação um cenário de acidente. De entre os vários casos tipo identificados em III.C, escolheu-se o primeiro, por ser aquele onde foram reportados mais acidentes. Este local (+38° 43' 36.84", -9° 24' 20.52"), representado na Fig. 3, encontra-se localizado no troço Cascais – Alvide (sentido Cascais – Lisboa), próximo do km 22.1, numa zona em que a auto-estrada tem 3 vias. A distância entre a saída imediatamente anterior ao local de acidente (10ª saída em direcção a Cascais/Abuxarda) é de, aproximadamente, 1.2 km (marcador de local vermelho da figura). Conforme referido anteriormente, para aproveitar a infra-estrutura existente, as RSUs devem ser colocadas junto aos postes de CCTV ou junto aos PMVs (sinalizados na figura com os marcadores azuis). Contudo, uma vez que a distância entre as mesmas ronda poucas centenas de metros, não haverá necessidade de instalar RSUs em todos os locais. Optou-se assim por incluir na modelação do

cenário apenas RSUs com uma distância média entre as mesmas de cerca de 1 km (RSU 1, 2 e 3).

Conforme foi referido anteriormente, a causa mais frequente de acidentes é o excesso de velocidade. Todavia, os acidentes podem decorrer em situações de elevada intensidade de tráfego, ou mesmo em situações de baixa intensidade. Qualquer um destes cenários pode ter um forte impacto no desempenho da rede veicular, uma vez que o primeiro pode conduzir a uma sobrecarga de tráfego de dados e o segundo pode originar falta de conectividade na rede. Para melhor avaliar o impacto destes dois casos extremos, optou-se por selecionar os valores registados para o troço de Linda-a-Velha – Miraflores para o período de maior tráfego (8:00-9:00 h) e o valor do troço Cascais-Alvide para o período de menor tráfego (2:00-3:00 h).

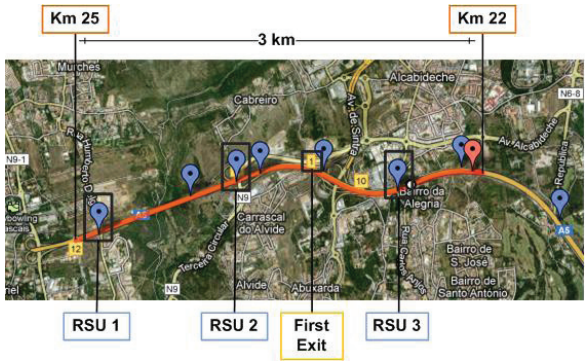


Fig. 3 – Troço Cascais-Alvide da auto-estrada A5

B. Modelo de mobilidade

Para a simulação da mobilidade usou-se o simulador de mobilidade MOVE[1] (*Mobility Model Generator for Vehicular Networks*), uma vez que este se adequa ao cenário em estudo. Este gerador de tráfego rodoviário permite a modelação de cenários com um elevado número de nós, tendo em conta ambos aspectos de macro- e micro-mobilidade [1], e possibilita a interligação ao simulador de rede usado - o *ns-3*[2] - apresentado na secção V.C.

Tendo por base o cenário anteriormente descrito, foi definido um troço no *Map Editor* do MOVE com as características indicadas nas tabelas III e IV. De salientar que o MOVE não permite a criação de perfis complexos com curvas e desníveis, pelo que não se consegue representar a geografia do troço em questão duma forma rigorosa.

TABELA III
CARACTERIZAÇÃO DO TROÇO CASCAIS-ALVIDE DA A5

Comprim. do troço (m)	3000 (km 22 – km 25)
Sentido do trânsito	Cascais - Lisboa
Número de vias	2/3
Limite de velocidade (km/h)	120

Para a modelação da mobilidade dos nós foi utilizado *Vehicle Movement Generator* tendo por base os valores de intensidade de tráfego anteriormente definidos. As RSUs foram representadas com recurso a nós fixos, localizados nas coordenadas exactas dos sistemas de CCTV previamente selecionados. O modelo de mobilidade usado pelo MOVE é do tipo *Car Following Model* [15][16], conseguindo modelar a aceleração, a travagem ou mudança de faixa em sequência da aproximação de outros veículos e a velocidade constante que permite manter a distância mínima de segurança ao veículo dianteiro. No entanto, não considera a resposta dos condutores a estímulos, o que poderia ser interessante para modelação de situações de acidente.

No modelo utilizado, no instante inicial os veículos encontram-se todos no ponto de partida, correspondente à entrada da auto-estrada. O instante de partida de cada veículo é definido aleatoriamente, seguindo uma distribuição uniforme, com valores compreendidos entre o instante inicial e o instante final da simulação (180 s). O veículo inicia a sua marcha com a velocidade máxima permitida (120 km/h). Esta é uma representação relativamente simples da realidade, uma vez que não tem em consideração a distribuição real dos instantes de partida dos veículos e do seu movimento, podendo desta forma condicionar a densidade de tráfego que se obtém e, consequentemente, o desempenho da rede. Todavia, os dados reais existentes não possibilitam uma caracterização mais detalhada.

TABELA IV
LOCALIZAÇÃO DOS PONTOS MAIS RELEVANTES

	Posição Real	**Posição MOVE (m)**
Local acidente	km 22.075	2925
Primeira saída	km 23.230	1770
RSU 1	km 24.840	160
RSU 2	km 23.790	1210
RSU 3	km 22.600	2400

A modelação do cenário de acidente foi realizada considerando duas situações distintas: na primeira, o veículo vítima do acidente encontra-se parado, impedindo a circulação numa das faixas de rodagem; na segunda, as três faixas estão bloqueadas, uma pelo veículo acidentado e as restantes, e.g., por veículos de assistência.

C. Modelação da rede veícular

A simulação do cenário de acidente foi realizada com recurso ao simulador de redes *ns-3*. A escolha deste simulador derivou do facto de ser uma ferramenta de uso livre, com grande utilização pela comunidade científica, que permite realizar simulações complexas com um nível de detalhe que permite reproduzir de forma bastante fiável os resultados reais.

A arquitectura da rede veícular é implementada importando o ficheiro de saída do MOVE, que contém a posição de cada nó em cada instante de tempo de simulação. Nas simulações que envolvam comunicação V2I/I2V, as RSUs são nós especiais cuja coordenada de posição se mantém ao longo da simulação. A entrada dos

[1] MOVE - http://lens.csie.ncku.edu.tw
[2] *ns-3* - http://www.nsnam.org/

veículos e o seu movimento é realizada de acordo com o modelo de mobilidade anteriormente descrito.

O facto do *ns-3* não dispor ainda de todos os módulos necessários à simulação de um nó com a arquitectura proposta (Fig. 2) conduziu à necessidade de efectuar adaptações ao modelo proposto anteriormente.

As aplicações anteriormente consideradas são modeladas através de geradores de tráfego CBR (*Constant Bit Rate*), com diferentes configurações de tempo entre geração de mensagens. Por uma questão de simplicidade, todas as mensagens têm o mesmo tamanho (512 bytes), o qual foi definido de forma a assegurar que a informação de segurança já se encontra incluída no pacote, seguindo as indicações definidas em [17], [18] e [19].

Conforme referido anteriormente, a informação da aplicação PCW tem de chegar rapidamente a todos os nós, o que se consegue colocando os nós receptores a reenviar a informação recebida pela primeira vez, enquanto que a informação das restantes aplicações tem um caracter local. Este comportamento foi modelado através da definição do campo *time to live* (TTL) na mensagem, que permite controlar o número de vezes que a mensagem é retransmitida. Assim sendo, para a aplicação PCW foi seleccionado um valor que permite que a mensagem seja retransmitida o número de vezes necessário que permita a sua recepção por nós que se encontrem antes da saída no instante do acidente. Este valor deve ser objecto de parametrização para os diferentes cenários que se possam considerar. De forma a respeitar os princípios funcionais definidos anteriormente, as aplicações PCW e EEBL só se iniciam quando ocorre o acidente. Os valores utilizados encontram-se representados na Tabela V.

TABELA V
PARAMETRIZAÇÃO DOS GERADORES DE TRÁFEGO

	PCW	**EEBL**	**CCW**
Tamanho das mensagens (B)	512	512	512
Frequência (Hz)	10	10	2
Destino	Todos	Todos	Todos
TTL	64	1	1
Instante de início (s)	$T_{acidente}$	$T_{acidente}$	0
Instante de fim (s)	$T_{simulação}$	$T_{simulação}$	$T_{simulação}$

Ao nível de transporte, o modelo de simulação desenvolvido utiliza como protocolo de transporte não fiável o protocolo UDP. Ao nível do encaminhamento foi utilizado um módulo que implementa o protocolo de encaminhamento geográfico GPSR [20], com um serviço de localização semelhante ao que é usado por Karp e Kung na especificação do próprio protocolo [13].

É ao nível da camada MAC que surgem as maiores limitações, uma vez que o *ns-3* ainda não suporta o protocolo 802.11p. Desta forma, foi necessário recorrer ao módulo que implementa a comunicação 802.11 em modo *ad-hoc*. Com esta solução não existe suporte de qualidade de serviço e comutação entre canais, que são mecanismos essenciais à coexistências das aplicações de segurança com outros tipos de aplicações. O facto de não se estarem a considerar outras aplicações para além das que se relacionam com cenários de emergência reduz fortemente o impacto desta limitação. Pode, no entanto, acontecer que, por incapacidade de se diferenciarem as várias aplicações em uso, o desempenho da aplicação mais crítica - PCW - seja afectado.

A camada física já tem suporte para a norma 802.11p, tendo sido este o modelo utilizado. Para obter resultados de simulação que se aproximem, tanto quanto possível, das condições reais, foi colocado um cuidado especial na escolha dos modelos de propagação. Optou-se pela utilização de um modelo que contabilize as perdas por atenuação do sinal com a distância (*path loss*) e outro que tem conta as perdas devido aos efeitos de dispersão do sinal (*multipath fading*). Assim sendo, usam-se, respectivamente, os modelos *Two Ray Ground Reflection* e o *Nakagami* [21], [22]. O alcance dos diversos tipos de antenas foi modelado com recurso à configuração da potência de transmissão e do ganho da antena. Os valores utilizados encontram-se representados na Tabela VI.

TABELA VI
PARAMETRIZAÇÃO DOS MODELOS DE PROPAGAÇÃO E DAS ANTENAS

			Veículo	**RSU**
Modelo *Nakagami*	**Nakagami *m*-factor**	**m0**	1.5	
		m1	0.75	
		m2	0.5	
Modelo *Two Ray Ground*	**Altura (m)**		1.7	6.3
Antenas	**Potência de transmissão (dBm)**		5	18
	Ganho (dBi)		2	9

VI. RESULTADOS OBTIDOS

A. *Cenário de teste*

Os testes realizados destinam-se a avaliar se os requisitos da aplicação mais crítica (PCW) conseguem ser garantidos, tendo em consideração que existe tráfego proveniente de outras aplicações de segurança em circulação. Esta avaliação foi realizada em vários cenários diferentes:

- Comunicação com e sem RSU.
- Bloqueio de uma ou das três vias.
- Tráfego com baixa e elevada intensidade.

B. *Métricas de avaliação*

Para aferir o desempenho da aplicação PCW foram estipuladas métricas de nível de aplicação e de nível de rede.

Do ponto de vista da aplicação, foram definidas as seguintes métricas:

- **Taxa de Aviso** – percentagem de veículos em circulação que receberam a notificação de acidente.
- **Taxa de Aviso Útil** – percentagem de veículos em circulação que receberam a notificação do acidente dentro dos limites de latência e alcance característicos da aplicação PCW.

- **Latência de Notificação** – tempo que decorre desde que o acidente ocorre até que o veículo é notificado.
- **Posição de Notificação** – posição do nó quando recebe a notificação de acidente, medida em relação à coordenada de entrada.

Ao nível da rede, as métricas consideradas foram:

- **Número de *hops*** – número de nós usados para retransmitir a mensagem.

C. Resultados obtidos – caso geral

A informação de cada veículo no instante de notificação do acidente é representada através dum conjunto de gráficos do tipo XY, nos quais coordenada X descreve a Latência da Notificação e a coordenada Y a Posição de Notificação.

Ambas, Fig. 4 e Fig. 5 ilustram os valores obtidos na situação de baixa intensidade veicular, e a Fig. 6 e Fig. 7 no caso em que a intensidade é elevada.

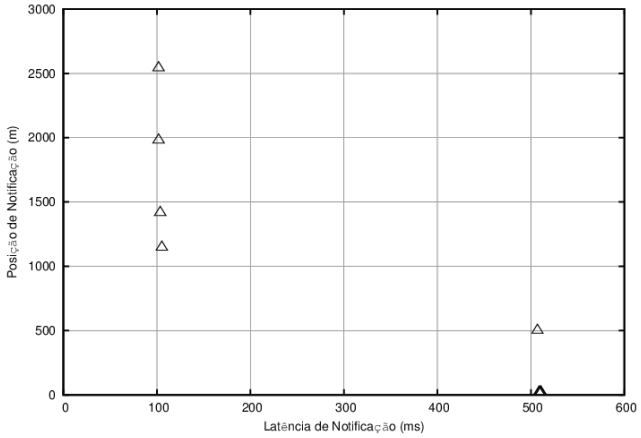


Fig. 4 – Baixa intensidade veícular; Comunicação V2V; 1 via bloqueada

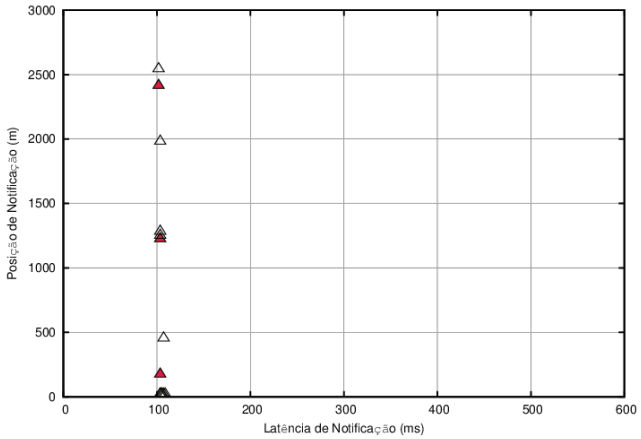


Fig. 5 – Baixa intensidade veícular; Comunicação V2I; 1 via bloqueada

À excepção dos veículos que se encontram mais distantes do acidente, em situação de baixa intensidade de tráfego com comunicação V2V, a maioria dos veículos recebe a notificação muito rapidamente, sendo a Latência de Notificação da ordem dos 100 ms. No entanto, os veículos mais distantes têm latências mais elevadas (cerca de 500 ms), embora consigam ser avisados quando ainda se encontram longe do local do sinistro. Como se pode observar na Fig. 5, a existência de RSUs (pontos ilustrados a vermelho) vem permitir reduzir o valor da latência para os veículos mais distantes.

Quando a intensidade do tráfego é elevada verifica-se uma maior variação das condições de notificação que se traduz, essencialmente, numa maior dispersão da Posição de Notificação. Relativamente à Latência de Notificação, embora se verifiquem variações de valor, a latência máxima observada é bastante inferior (cerca de 135 ms), uma vez que existem mais veículos capazes de retransmitir a notificação. A existência de RSUs permite que mais nós recebam a notificação mais rapidamente, o que é vísivel pela maior concentração de pontos junto ao eixo dos Y. Verifica-se ainda que o número de nós distantes que recebem a notificação mais cedo aumenta. Esta situação é particularmente evidente para o caso da RSU 2 (Y = ~1210 m), conforme se pode confirmar por observação da Fig. 7.

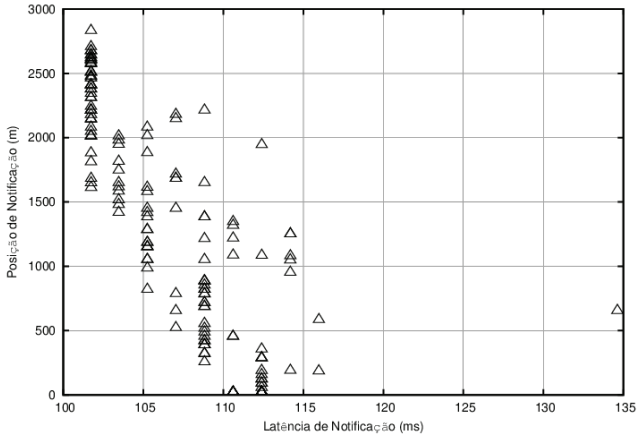


Fig. 6 – Elevada intensidade veícular; Comunicação V2V; 1 via bloqueada

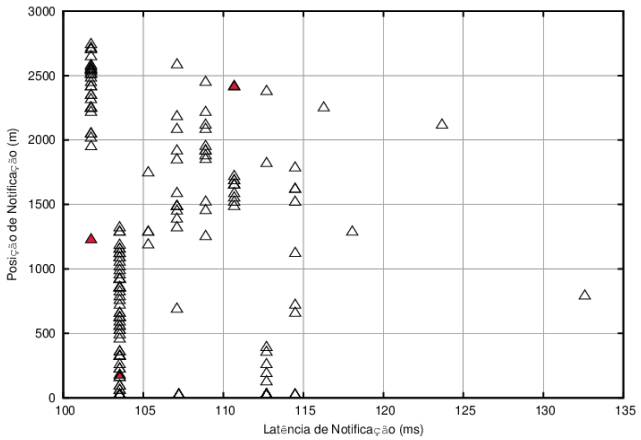


Fig. 7 – Elevada intensidade veícular; Comunicação V2I; 1 via bloqueada

Os resultados apresentados na Fig. 8 e Fig. 9 ilustram o histograma do número de *hops* utilizado e confirmam as conclusões anteriores. O uso de RSUs reduz o número de *hops* da comunicação, o que permite reduzir a Latência de Notificação, uma vez que a RSU têm um maior alcance e possibilita a transmissão de informação para nós distantes mais rapidamente.

D. Resultados obtidos – condições pós-acidente

Com base na informação recebida pode-se também avaliar em que medida a notificação permitiria aos condutores reagir atempadamente à situação de acidente. Existem duas situações diferentes que devem ser consideradas:

- Os condutores que se encontram na zona do acidente e que têm de ser avisados rapidamente, a fim de reagirem a tempo de evitar acidentes secundários.
- Os condutores que se encontram em rota para a zona do acidente mas ainda a tempo de receber o alerta de forma a poderem desviar-se, seguindo um percurso alternativo, evitando congestionamento.

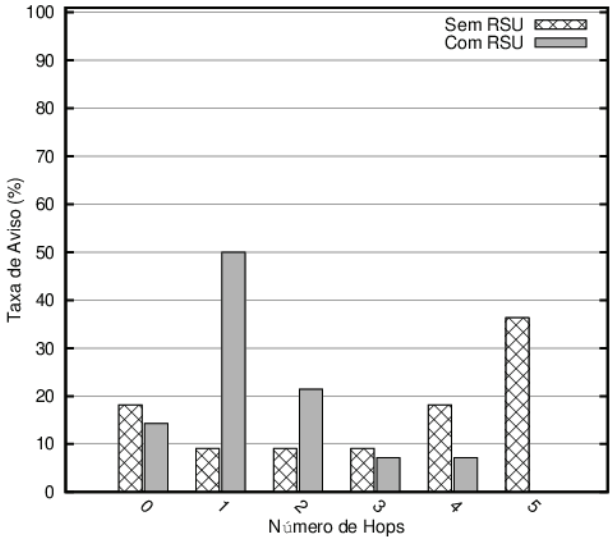


Fig. 8 – Baixa intensidade veícular; Comunicação V2V e V2I; 1 via bloqueada

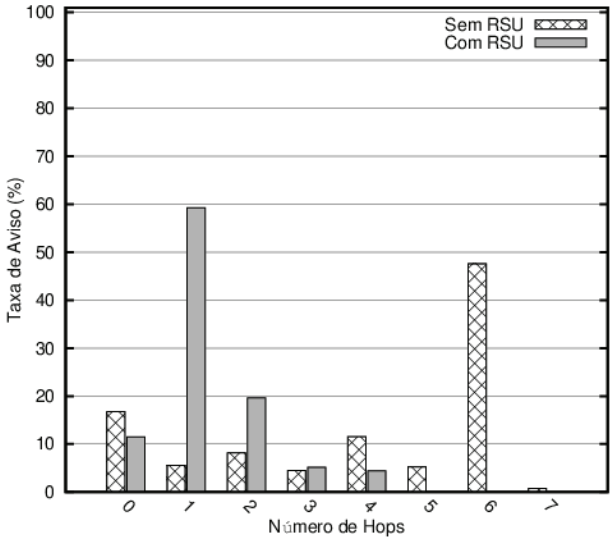


Fig. 9 – Elevada intensidade veícular; Comunicação V2V e V2I; 1 via bloqueada

Tendo por base estas premissas, foi medida a Taxa de Aviso, a Taxa de Aviso Útil e a Latência de Notificação quer para o caso geral, quer para cada uma das situações anteriormente identificadas. A Tabela VII apresenta os resultados obtidos para cada uma destas métricas.

Neste estudo considerou-se que, no caso geral e no caso dos nós antes da saída, o aviso só era útil se chegasse antes dos 500 ms, enquanto que, para os nós na região de acidente, o valor máximo aceitável era de 105 ms (o valor de latência definido para a aplicação EEBL com uma margem de 5%).

TABELA VII
RESULTADOS PARA AS DIFERENTES CONDIÇÕES PÓS-ACIDENTE

Estatísticas globais	Total	Antes da saída	Zona de acidente
Taxa de Aviso	100%	100%	100%
Taxa de Aviso Útil	100%	100%	86%
Latência de Notificação Mínima (ms)	101,73	101,73	101,73
Latência de Notificação Média (ms)	106,08	106,26	103,19
Latencia de Notificação Máxima (ms)	132,59	132,59	110,68

VII. CONCLUSÕES DE TRABALHO FUTURO

No presente trabalho, procedeu-se à análise do desempenho das redes veiculares em auto-estrada, tendo em conta aspectos tais como a viabilidade da colocação de RSUs ao longo da infra-estrutura, o seu impacto neste mesmo desempenho e a capacidade de possibilitar a recepção atempada de avisos referentes a situações de emergência, de forma a minimizar segundas colisões e mitigar o congestionamento de tráfego. Procedeu-se à análise dos diferentes aspectos recorrendo à modelação e simulação da mobilidade e comunicação entre veículos e, adicionalmente, entre veículos e infra-estrutura rodoviária – as RSUs.

Os resultados obtidos permitem concluir que o uso de RSUs permite melhorar o desempenho das aplicações de segurança rodoviária, na medida em que reduz a latência na recepção de informação. Da análise dos mesmos dados é ainda possível concluir que não é necessário instalar as referidas RSUs em todos os locais onde actualmente existem postes de CCTV ou PMVs, o que terá vantagens significativas em termos de custo.

Dos resultados numa fase pós-acidente, referentes às condições de elevada intensidade, com três vias bloqueadas e inclusão de RSUs, verifica-se que os veículos que se encontrem longe da zona de acidente, num ponto que antecede uma saída são todos avisados em tempo útil, i.e., antes dos 500 ms. Dos veículos que se encontram na zona de acidente, i.e., dentro do raio máximo de alcance de 300 m, todos receberam a notificação, porém, apenas cerca de 86% dos mesmos recebeu dentro do limite de latência, i.e., dos 105ms. No entanto, considerando os 500ms da aplicação PCW, verifica-se que todos os nós em estudo receberam a notificação com sucesso, com uma latência abaixo deste valor.

Como trabalho futuro pretende.se avaliar esta situação num protótipo experimental, de pequena escala, que permita avaliar em que medida os resultados obtidos por simulação são representativos da situação real. Este estudo permitirá determinar a importância de aspectos que não se conseguem modelar em simulação, tais como: presença de obstáculos na auto-estrada, a própria geometria real da auto-estrada (desníveis, pontes, curvas entre outros...).

AGRADECIMENTOS

Este trabalho foi financiado por fundos nacionais através da FCT – Fundação para a Ciência e a Tecnologia, no âmbito do projecto PEst-OE/EEI/LA0021/2011.

REFERÊNCIAS

[1] Yasser Toor, Paul Muhlethaler, Anis Laouiti and Arnaud de la Fortelle. "Vehicular Ad-hoc Networks: Applications and Related Technical Issues". IEEE Communication Surveys. Vol. 10. N. 3, 2008

[2] Elmar Schoch, Frank Kargl, and Michael Weber, Tim Leinmüller "Communication Patterns in VANETs". IEEE Communications Magazine, November 2008

[3] Ch. Maihöfer, T. Leinmüller, and E. Schoch, "Abiding Geocast: Time-Stable Geocast for Ad Hoc Networks,"VANET '05, 2005, pp. 20–29.

[4] T. Kosch et al., "The Scalability Problem of Vehicular Ad Hoc Networks and How to Solve It," IEEE Wireless Commun., vol. 13, no. 5, Oct. 2006, pp. 22–28

[5] Fan Bai and Bhaskar Krishnamachari. " Exploiting the Wisdom of Crowd: Localized, Distributed, Information-centric VANET". IEEE Communications Magazine, May 2010

[6] Steve PECHBERTI, Dominique GRUYER, Denis GINGRAS and Francis DUPIN, "Design of a Modular Demonstrator for Safety Application Systems: the CVIS Project". 2010 IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA, June 21-24, 2010

[7] Jihua Huang and Han-Shue Tan. Impact of communication reliability on a cooperative collision warning system. In Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, pages 355–360, Sep. 30-Oct. 3 2007.

[8] Steven E. Shladover. Effects of traffic density on communication requirements for cooperative intersection collision avoidance systems (CICAS). Technical Report UCB-ITS-PWP-2005-1, Institute of Transportation Studies, University of California, Berkeley, March 2005.

[9] Jason J. Haas and Yih-Chun Hu. "Communication Requirements for Crash Avoidance". VANET'10, September 24, Chicago, Illinois, USA, 2010

[10] U.S. Department of Transportation (U.S. DOT) NHTSA. (2005). "Vehicle Safety Communications Project Task 3 Final Report – Identify Intelligent Vehicle Safety Applications Enabled by DSRC".

[11] Bai, F., Krishnan, H., and Sadekar, V. (2006). "Towards Characterizing and Classifying Communication-based Automotive Applications from a Wireless Networking Perspective". *In Proceedings of IEEE Workshop on Automotive Networking and Applications (AutoNet)*.

[12] Experimental Study on the Impact of Vehicular Obstructions in VANETs. Rui Meireles1;3, Mate Boban2;3, Peter Steenkiste1, Ozan Tonguz2 and João Barros3 frui@cmu.edu, mboban@cmu.edu, prs@cs.cmu.edu, tonguz@ece.cmu.edu, jbarros@fe.up.ptg 1Department of Computer Science, Carnegie Mellon University, USA 2Department of Electrical and Computer Engineering, Carnegie Mellon University, USA 3Instituto de Telecomunicações, FEUP DEEC, University of Porto, Portugal, *IEEE vehicular networking conference*.

[13] Karp, B., and Kung, H. T. (2000). "GPSR : Greedy Perimeter Stateless Routing for Wireless Networks". *MobiCom 2000* (p. 243--254).

[14] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in Proceedings of the *5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, WA, USA, August 1999, pp. 151–162.

[15] Jérôme Harri, Fethi Filali, and Christian Bonnet. "Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy". IEEE Communications Surveys and Tutorials. Vol. 11, No. 4, 2

[16] Stefan Krauss, Peter Wagner, Christian Gawron, "Metastable States in a Microscopic Model of Traffic Flow" Physical Review E, volume 55, number 304, pages 55-97; May, 1997.

[17] Schmidt-Eisenlohr, F.; Torrent-Moreno, M.; Mittag, J.; Hartenstein, H.; , "Simulation platform for inter-vehicle communications and analysis of periodic information exchange". *Wireless on Demand Network Systems and Services, 2007. WONS '07. Fourth Annual Conference on* , vol., no., pp.50-58, 24-26 Jan. 2007

[18] Raya, M., and Hubaux, J.-pierre. (2005). "The Security of Vehicular Ad Hoc Networks". *SASN '05 Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks* (p. 11--21). New York, New York, USA: ACM Press.

[19] Petit, J.; Mammeri, Z.; , "Analysis of authentication overhead in vehicular networks," *Wireless and Mobile Networking Conference (WMNC), 2010 Third Joint IFIP* , vol., no., pp.1-6, 13-15 Oct. 2010

[20] Fonseca, A., Camões, A., and Vazão, T. (2012). Geographical routing implementation in NS3. *WNS3 2012* (p. 364--369).

[21] Killat, M., Rössel, C., Schmidt-eisenlohr, F., Vortisch, P., & Busch, F. (2007). Enabling Efficient and Accurate Large-Scale Simulations of VANETs for Vehicular Traffic Management. *VANET '07 Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks* (pp. 29-38). New York, New York, USA: ACM Press.

[22] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication," in Proc. *1st ACM Int. Workshop on Vehicular Ad Hoc Networks (VANET)*, Philadelphia, PA, USA, p. 88, Oct. 2004.

[23] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing," IEEE/ACM Trans. Networking, vol. 14, no. 3, 2006, pp. 479–91.

[24] Grau, G.P.; Pusceddu, D.; Rea, S.; Brickley, O.; Koubek, M.; Pesch, D.; , "Vehicle-2-vehicle communication channel evaluation using the CVIS platform", *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on* , vol., no., pp.449-453, 21-23 July 2010

On Multi-Path Aggregation of Duplicate-Sensitive Functions

Milton Armando Cunguara,Tomás António Mendes Oliveira e Silva
IEETA / University of Aveiro
Aveiro, Portugal
{milton.cunguara,tos}@ua.pt

Paulo Bacelar Reis Pedreiras
IT / University of Aveiro
Aveiro, Portugal
pbrp@ua.pt

***Abstract*—Previous decades brought about a revolution in radio and microprocessor technology that made possible a plethora of new applications. In particular, the possibility of using many inexpensive sensor nodes interconnected by wireless networks (WSN) for a number of ends, such as pollution monitoring and defense, draw the attention of the research community. WSN are usually heavily resource-constrained. Of particular relevance is energy, since in many applications nodes should operate during long periods from batteries. The literature on this topic reveals many techniques to improve energy efficiency, one of them being the use of network aggregation. However, the problem of aggregation of duplicate sensitive summaries (e.g. sum, average, histogram, etc.) in multi-path routing networks is not fully resolved. This paper addresses this problem by sending redundant aggregated information through different paths, so data can be reconstruct to obtain the exact summary, provided that there is at least one feasible path. Two algorithms are presented, one better suited for networks dominated by link errors and another suited to networks where the predominant error source is node failures. The algorithms are light during normal network operation, with the most intensive processing performed during the initialization phase. The approach presented herein outperform previous solutions found in the literature in two key aspects: complete topology independence and aggregation depth independence.**

***Index Terms*—Wireless Sensor Networks, Aggregation, Multi-path, Spatial Query**

I. Introduction

Wireless sensor networks (WSN) have recently emerged as a synergy of two related technologies, i.e. radio and microprocessors technology. Both of them had exponential improvements in size, cost and functionality in recent years, opening the door to applications so diverse as air monitoring, forest fire detection, structural monitoring, water monitoring, etc. Furthermore, such improvements are expected to keep the pace in the near future, thus WSN should become even more prevalent.

Nonetheless, there are a number of aspects that need to be addressed in order to make WSN reach their full potential. One of the key aspects is energy efficiency, since in many application domains node's energy source are batteries and WSN should operate during long periods, with low or no maintenance at all. This aspect turned energy conservation, i.e. devising mechanisms to maximize the lifetime of the network, one of the most studied aspect of WSN.

Techniques proposed to reduce nodes' energy consumption include the use of network aggregation, the use of cluster heads (with or without cluster head rotation), exploitation of the inherent spatial correlation of readings among neighbor nodes, variation of transmission energy, sleeping during long periods of inactivity, exploitation of temporal correlation of the signals (data caching, estimation, system identification and so on), among others.

One approach to data transmission from common nodes to the root is to have all nodes transmit their data integrally to the root and at the root perform the computations on the collected data. However, this approach is energy-wise inefficient due to the rather high number of messages that are sent. A more efficient approach is the gradual aggregation of values as they are transmitted upstream, which is called **in-network aggregation**.

On the other hand, WSN links are *fragile*. Particularly, they can be temporarily unavailable, are subject to relatively high error rates or be asymmetric, among other issues. Hence, the use of multi-path routing has been proposed [1]–[4] to lessen its effects.

However, the use of multi-path routing may cause errors in the aggregation of duplicate-sensitive functions. A function is said to be duplicate-insensitive if its result does not change upon the introduction of a duplicate argument. For example, $\min$ and $\max$ function are duplicate-insensitive. A function is said to be duplicate-sensitive if its output changes with the addition of an argument, even if it is a duplicate. For example the sum function, in which introducing a non-zero argument more than once causes different results, is duplicate-sensitive.

Many attempts have been made to solve this issue, for example, by giving approximate answers, by forcing single path routes, by searching for aggregate-insensitive versions or by decomposing these functions in a series of duplicate-insensitive functions and then query the WSN for each of the new functions, among others.

None of the already introduced solutions satisfies the initial goals, since all of them either have an error by nature (approximations based approaches) which contradicts the goals of multi-path, or use single routes, which does not offer any redundancy, or considerable increase the number of messages that are required to compute the aggregate (alternative func-

This work was partially supported by the Portuguese Government through FCT - "Fundao para a Ciłncia e a Tecnologia" in the scope of project Serv-CPS -PTDC/EEA-AUT/122362/2010 and Ph.D. grant - SFRH/BD/62178/2009

tions approaches), thus defeating the original purpose of the aggregation, which is to reduce the number of messages in order to save energy.

This paper addresses this problem by sending redundant aggregated information, so that data can be reconstructed to obtain the exact summary, provided that there is at least one possible path. Two algorithms are presented, one better suited for networks dominated by link errors and another suited to networks where the predominant error source is node failures. The algorithms are light during normal network operation, with the most intensive processing performed during the initialization phase. The approach presented herein outperforms previous solutions found in the literature in two key aspects: complete topology independence and aggregation depth independence.

The remaining of this paper is organized as follows. Section II presents an overview of the related work. Section III presents the methodology proposed in this paper to carry out the aggregate of duplicate-sensitive data in multi-path networks. Section IV presents simulation results carried out to assess the correctness of the algorithms and to evaluate its performance. Finally, section V concludes the paper.

II. Related Work

The field of WSN has a vast body of knowledge, too vast to be covered in the related work section of a single paper as can be seen in [5]. Therefore, this section is focused only in contributions closely related to problem addressed in this paper.

The use of aggregation in WSN excels in metrics such as energy expenditure and network lifetime, as shown in [6]. Furthermore, in the same reference it is also shown that the denser the networks the higher the benefits of aggregation. Similar results were reported in [7]. [8] presents a meta level view of aggregation and argues for a co-design of the integrating parts of the network.

Watfa *et al* [9] build an index tree similar to the SRT of TinyDB [10] (broadcast level and $\mathrm{level} = \min(\mathrm{level}) + 1$). The authors used an index table to decide the aggregation value, plus a common value agreement, which is a value that nodes with the same parent are supposed to verify $\mathrm{abs}(\mathrm{node}_{\mathrm{reading}} - \mathrm{cv}) < \delta$. cv is computed as the average of the children reading and is sent back to them. If a children verifies the last condition then it does not send its data. $\max / \min$ are also based on cv, thus may lead to erroneous values. In fact, this aspect is common to approaches that use a cluster head and spatial correlation. Nodes with more than one parent alternate between sending messages to each of its parents.

Liao *et al* [11] leverages the (ant) bio-inspired path finding algorithm to build a routing tree. In this algorithm, all source nodes explore all the paths to the sink, leaving a certain amount of pheromone at each link. The amount of pheromone left on each link of each path from a source to the sink is a function of several factors (e.g. path size). The amount of pheromone on each link is the sum of the amount left by on each paths that goes through the link. The higher a paths' pheromone levels, the more likely it will be active. However, this algorithm is not light. A similar approach is pursued in [12] and in references therein.

Ganesan *et al* [13] use a wavelet based approximate aggregation technique to store the results of several queries at different network hierarchies. Nodes at different hierarchies store results with different precisions. Whenever a query with a certain precision is done, it drills down the network until find a node with precision enough to answer it. Due to the high volume of data generated by this approach, an aging scheme was employed.

Considine *et al* [1] present the FM-SKETCH (introduced by Flajolet and Martin), which approximate the number of distinct elements of a superset by seeing only its initial part and such sketch are used to estimate the number of distinct nodes in a network — approximate count summary — which is well suited for multi-path networks. The paper generalizes FM-SKETCH to approximate sum summaries by having each node producing val (val is the value that the node sensed) distinct elements into the superset. Evidently, the number of distinct elements of this superset is equal to the sum summary. The sum summary is approximated by using the FM-SKETCH on the superset. A number of optimizations are provided. All the same, this approach that supposedly reduced the amount of data by computing approximates, actually for typical relative errors (0.85-0.95), it uses about the same amount of memory as traditional exact summaries. Therefore, the use of exact aggregation techniques with a data-caching or send-on-delta can be discarded. The authors compared their approach to TAG and to the LIST approach, i.e. each node sends the value of the aggregate and the set of nodes used to compute the aggregate, which is similar to the approach provided herein, however, their LIST approach is considerable less optimal. First, nodes always send a huge set, second the node that is receiving may not be capable of finding a way to aggregate the received data due to overlaps.

A Sparse aggregation scenario is studied in [14], by exploring dense WSNs with a small number of hotspots. A mechanism to find suitable routes was also presented. Other types of approximate aggregations have been proposed as is the case of quantile tracking [15], [16], top-k estimation [17], [18] though the later group tend to be more exact. [19] is an example in which an approximation of an histogram is used to compute queries in WSN, however, the algorithm used is exact if the histogram is exact, but the authors do not provide any algorithm to compute exact histograms.

In [20] and later at [21] the authors use a hybrid periodic\asynchronous model, in which data is sent periodically, however, rapid transition are responded to by immediately sending data asynchronously. Nevertheless, the authors did not provide any difference between this paradigm and event triggered transmission with a periodic *I'am alive* messaging. The asynchronous traffic is controlled using filters. Whenever a value is within the filter's range it does not get transmitted. [22] aims at similar goals, using data caching and

aggregation at each level of the tree.

In [23] it is introduced the direct diffusion of digests (aggregates), in which nodes compute their value as a function of their current value and the value received from neighbor nodes. And they start sending this new value, which can be piggy-backed in a periodic beacon. It takes at most the diameter of the network (in number of hops) to diffuse such digests. Obviously, in this form it only can compute digests of exemplary functions. Exemplary functions are the ones that can be computed as the result of an aggregate of previous values and one single new value. In fact, they have this name because their result depend on only one value, such as $\max$ and $\min$. For non-exemplary function, the authors propose that first, a direct digest that is always "won" by the sink is performed, second, each node would memorize the node from which it received the winning diffusion (parent) until a tree is formed, third, diffusion would be sent along the tree that emerged in the process. All the same, the authors do not show any difference between their proposal and the regular diffusion. Notwithstanding, the paper presents an interesting study of link asymmetry in WSN, which led the authors to propose a mechanism to switch parents whenever a certain link is asymmetric.

Nath *et al* [24] provide a formalization of the concept of diffusions. Three operations are considered synopsis generation, fusion and evaluation, that work as suggested by their names. The paper also provides a number of, so called, necessary and sufficient conditions for correctness, though, it also presents a situation that verifies all such conditions but does not produce the correct value, implying that the presented conditions are not sufficient. [25] uses a fuzzy logic approach to compute exemplary functions. Each node has a fuzzifier that decides whether to send the data. A comparison with the 'no aggregation scenario' was made, though no comparison with classical aggregations was presented. It was also used the sleep approach.

In [2] it is proposed to solve the problem of multi-path aggregation of duplicate sensitive nodes by keeping nodes from aggregating data if there is a possibility of duplication. To this end, upon the construction of the multi-path tree, nodes send theirs and their parents addresses along with their hop count to the root. Nodes that join the network, would know each of its parents and its parents' parents. Based on this information, if the node as more than one parent, then it chooses the parents parents with most paths from it as the aggregation point, otherwise it chooses its only parent as the aggregation point. Only two levels of the WSN are searched for, therefore it might happen that the link of the chosen parent's parents up the tree may fail while there is another parents' parents link which is operational. The fact that the lost of one (uptree) link/node can cause the loss of information of many of children nodes puts in question the very use of multi-path (redundancy). (The approach advocated in this paper has the property that for any set of link failures it can always compute the aggregate of the values that can still reach the sink.)

Manjhi *et al* [3] use an hybrid approach to routing. Using a tree approach closer to the leaves and a multi-path approach closer to the sink. This helps to leverage the advantages of trees (low latency, low messaging) with the advantages of multi-path (increased robustness).

Al-Karaki *et al* [26] present both an exact and an approximate algorithms to find optimal routes for aggregation in WNSs. The exact algorithm is stated as an integer linear programming problem, which the authors argue to be too complex to be solved in WSN. Hence, they propose an approximate genetic algorithm. However, the genetic algorithm is itself computationally heavy and involves many message exchanges, which is exacerbated by being performed in several rounds, thereby consuming the very same energy that the algorithm aims at saving.

Another type of aggregates that gain a certain moment in the community is the gossip based aggregation [27]–[32], in which each node 1) read its own value, 2)aggregates it with messages that it receives and 3) sends it to a random neighbor a fraction of the its current value while maintaining another fraction of it. After an algorithm-dependent number of rounds, the aggregates converge to the correct value. Usual objections include the apparent lack of benefits for the common WSN, higher latencies and increased number of communications.

Other types of optimizations to WSN that do not relate to the main contribution of this paper have been proposed, such as the exploitation of temporal correlation [33]–[38] and spatial correlation, such as [12], [14], [39]–[46].

A. Count Summary

The main contribution of this work is independent of which count summary is used. However, it requires that the count is performed prior to carrying out a duplicate-sensitive function, or at least, to have a cache with the nodes in the network. Therefore, it is paramount to use an efficient count summary that should be performed as infrequently as possible.

The following conditions reduce the need to perform count summaries: 1) parent nodes trigger a count summary action only if a given child does not communicate for a given period of time (T_{alive}), 2) all children send a message when they are initialized or whenever they notice that have a different parent node. Additionally, they also send at least one message (of whatever type) with a given time window (T_{alive}). The optimum value for T_{alive} is a compromise. On the one hand it must be as big as possible to not spoil too much energy. On the other hand, large T_{alive} impairs network reactivity.

To increase the efficiency of the count summary, nodes use a bitmap addressing, in which each node address corresponds to a bit in a address array. Addresses can be pre-programmed prior to the deployment of the network. Leaf nodes put themselves in the count summary by sending a message with their bit set. Aggregations of the count summary are performed by implementing a bitwise OR of partial results. The number of distinct nodes that are offspring of a given node is equal to the number of set bits on its bitmap array. Evidently, the number of nodes in the network is equal to the number of

offspring of the root node plus one (the root itself). Due to the simple nature of this aggregated, a formal proof of its correctness will not be provided.

It should be remarked that a similar counting mechanism was proposed in [24]. However, their approach started with the bitmap addressing but at the higher levels used the FM-SKETCH [1], which conditions their approach to provide approximate results, and uses an high amount of memory [1].

III. Multi-path Aggregation

This paper considers mesh-like networks, where each node can reach only a limited subset of other network nodes, normally the nearest neighbors. Nodes' data should be forwarded to a particular node, designated by sink. Links that connect nodes are subject to errors, either transient or permanent. It is assumed that the underlying communication protocol provides error detection capabilities, discarding erroneous frames. Nodes are also subject to errors and are fail-silent, i.e., they either operate correctly or do not send any information. Furthermore, the following definitions apply:

- The WSN consists of $N_i, i = 0...K$ nodes.
- Without loss of generality, in the remainder of this paper N_0 designates the sink node;
- Each node is connected to one or more neighbour nodes;
- Each link between two nodes is designated by $L_{i,j}, i, j \in 0...K$;
- T_i is a bitmap representing the addresses of the node itself and its offspring
- A_k=[a,b,c, ...] denotes the aggregation of values a, b, c, ...;
- M_i designates a message sent by node N_i to its parent;
- A message can contain several aggregates. Aggregates sent in the same message are connected by a + symbol.

The system undergoes three sequential phases: physical topology discovery, virtual topology set up and, finally, the data collection, which is the normal state. Permanent errors are considered as topology changes and thus this whole process may be repeated as often as necessary, although eventually only over specifics parts of the network affected by errors.

In the first phase all reachable nodes and its connections are identified. More concretely, each node discovers which nodes are its offspring. This is achieved, for example, by using a count summary as the one previously described in section II-A.

Once the topology is identified, phase 2, which consists in the virtual topology set-up, is started. At this stage each parent knows all the paths to each one of its offspring. Based on this information, each parent sends a message to each one of its children indicating if and how data should be aggregated, thereby creating several groups of aggregates. The decision about which data should be aggregated is taken primarily with a focus on minimizing the total number of messages.

Two different aggregation strategies are proposed in this paper, one more suitable to handle node errors and another more efficient in the presence of link errors. The difference between these strategies resides primarily in which messages are aggregated together. In the former case, messages are aggregated such that if all messages that come from a given child are lost the aggregates can still be *reconstructed*, provided that there is at least one redundant link, whereas in the latter case the focus is on messages lost on a given link. The best strategy to use in a particular WSN should be selected according with the most frequent error source.

Finally, after phase 2 is complete, the system enters the normal operation phase, in which the data is collected. During this phase each parent node receives messages from its offspring, eliminates or reconstructs data, depending on the existence of errors, and forwards the aggregates defined during phase 2 to its parent node. Data recovery requires only a few table look-ups and simple algebraic operations on the received aggregates, thus during normal operation the processing overhead is small.

A. Multipath Aggregation of Duplicate Sensitive Summaries — Node Error Case

The strategy proposed to deal with node failures consists in setting one of the children to aggregate its own data with the data of its offspring, while the other children send several aggregates. These aggregates consist in its own data aggregated with the data of its offspring, followed by aggregates that contain the data that do not intersect with previous siblings in the same level. To illustrate this strategy, consider the simple WSN depicted in figure 1.

As depicted in figure 1, node N_1 sends a message composed only by one field, which aggregates its own data and the data of its offspring (M_1 = [1,4,5]). Node N_2 sends one message with two fields, one aggregating its own data with its offspring (A_1=[2,5,6]) and another field that contains the data that does not intersect T_1 (i.e. $T_2 \setminus T_1$; A_2 = [2,6]). Thus, M_2={[2,5,6]+[2,6]}. A similar procedure is applied to node N_3. Table I presents the aggregates conveyed by messages M_1 to M_3.

In case there are no errors, the sink receives aggregates $A_i = [1,4,5],[2,5,6],[2,6],[3,6,7],[3,7], i \in 1..5$, respectively contained in messages M_1, M_2 and M_3. Note that in the third node there are some duplicate entries, that were removed. It can be trivially verified that the correct value can be recovered by taking the last field of each one of these messages. Therefore, this strategy meets the first goal addressed in this

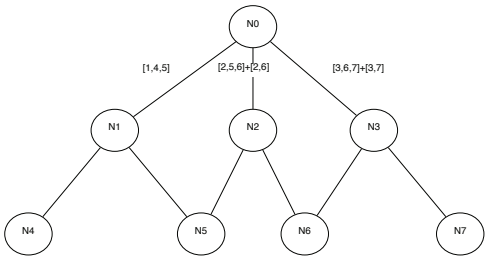


Fig. 1. Example 1: Node Error Strategy

[1] approximate queries are used primarily to reduce the amount of time and memory used to perform a given action, whereas exact queries are used when the correctness of the results have primacy

child index	message to send
1	$\{T_1\}$
2	$\{T_2\} + \{T_2 \backslash T_1\}$
3	$\{T_3\} + \{T_3 \backslash T_1 \cup T_3 \backslash T_2\} + \{T_3 \backslash \{T_3 \backslash T_1 \cup T_3 \backslash T_2\}\}$
$\vdots$	$\vdots$

TABLE I
EXAMPLE OF AGGREGATES FOR NODE ERRORS

paper, which is the ability to remove duplicates in the presence of redundant paths.

Lets now consider the case in which one of the nodes fails, e.g. node N_1. Observing figure 1, it can be seen that data from node N_1, which failed, and from node N_4, which has no redundant path to the sink, will be lost. However, data from all the other nodes should be recoverable. In fact, the sink node receives aggregates [2,5,6] + [2,6] and [3,6,7] + [3,7], respectively contained in messages M_2 and M_3. Data from all accessible nodes can be recovered by combining the first field of M_2 with the last field of M_3. If the failing node is node N_2, the sink node receives aggregates [1,4,5] and [3,6,7] + [3,7]. Data from all accessible nodes can be obtained by combining aggregates [1,4,5] and [3,6,7]. A similar reasoning could be carried out regarding the failure of any node in the network. Therefore, the proposed strategy meets the second goal of the paper, which is the ability to recover data that has redundant paths to the source in the presence of node errors.

This idea can be extended to a larger number of children, as shown in table I, i.e. each children first aggregate all its offspring, then aggregate all of its offspring minus the nodes that are reachable by nodes above it in the table, then do the same with minus sets from two nodes in above, then three and so on.

It can be seen that this approach may lead to an relatively high number of messages that must be sent by nodes further down the table, since the worst case number of aggregates grows as a power of two. Methods to dramatically reduce the number of entries of such table area addressed latter on.

B. Multipath Aggregation of Duplicate Sensitive Summaries — link Error Case

The link error case requires that whenever a link fails, the parent (or the root) node should receive all the data necessary to compute all aggregates, provided that there exists an alternative path.

The algorithm proposed to achieve this goal consists in having each node sending two aggregates, one containing the data related with its descendants that are reachable only by itself, and another aggregate composed by the data pertaining to nodes that communicate both with itself and its siblings. The algorithm resembles the one presented for the node error case, except that in the error node case the operation was a bitwise complement, whereas in this case (link error) the sets are disjoint in relation with the other children's children count summaries.

To illustrate this strategy, consider the simple WSN depicted in figure 2.

As depicted in figure 2, node N_1 sends a message composed by two fields, one aggregating its own data and the data of descendants that are reachable only by itself (node N_4, in the present case) and another aggregate with the data of descendants shared with each one of its siblings (only node N_5, in the present case), therefore M_1 = {[1,4]+[5]}. Node N_2 sends one message with three fields, one aggregating its own data only, since all its descendants are shared with its siblings, and two other aggregates with data of descendants shared with each one of its siblings, therefore M_2 = {[2]+[5]+[6]}. Following a similar reasoning, M_3 = {[3,7]+[6]}.

In this simple case it can be checked, by exhaustion, that this is a solution to the problem. In the absence of errors, the sink receives aggregates $A_i = [1,4],[5],[2],[5],[6],[3,7],[6], i \in 1..7$. The exact value can be recovered by adding A_1, A_2, A_3, A_5 and A_6. Thus, without errors the exact value can be recovered even in the presence of redundant paths, by sequentially adding the aggregates that have values not added before. By simple inspection of the aggregate set received by the sink, it can also be observed that each value appears in as many aggregates as the number of distinct paths to the sink. E.g. node N_5 that has two links appears in sets A_2 and A_4, while node N_4 appears only in one aggregate, since it has no redundant link. Thus, the redundancy is visible at the sink and it should be possible to recover the exact aggregate value even in the presence of errors. E.g. if link $L_{5,1}$ fails, the sink receives aggregates $A_i = [1,4],[x],[2],[5],[6],[3,7],[6]$. The exact value can be obtained e.g. by taking A_1, A_3, A_4, A_5 and A_6. The same reasoning can be applied to other sets to confirm that it is possible to recover the exact value of an aggregate function, even in the presence of link errors, by simple algebraic manipulation of the aggregates received by the sink, provided that there at least one alternative path.

C. Aggregate Generation and Data Reconstruction Algorithms

In previous sections, two algorithms to generate aggregates were introduced and illustrated in simple scenarios. Algorithms 1 and 2 describe how the aggregates can be generated for arbitrarily large WSN.

Even for the simple cases presented before, it was obvious that a rather high number of message exchanges could be necessary. In fact, in both cases the worst-case number of messages grew as a power of two with the number of siblings.

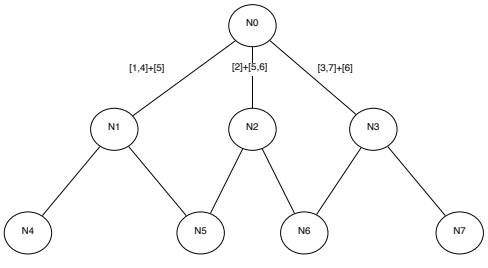


Fig. 2. Example 1: Link Error Strategy

Algorithm 1 Node Error Case: Parent Generated Transmission Lists

```
T_i ← Set of nodes reachable by offspring i
P← {} (Nodes that have been processed)
for {i = 1; i ≤ number of offsprings; i = i + 1} do
  M_i ← {}
  for {j = 0; j < i; j = j + 1} do
    Z ← Permutation of P taken j by j
    for all Z_l (elements of Z) do
      M_i ← M_i ⋃ {T_i \ ⋃_{x∈Z_l} T_x}
    end for
    j ← j + 1
  end for
  P← P ⋃{i}
  \\Remove duplicates and empty sets
  for j = 1; j < 2^{i−1} − 1; j = j + 1 do
    if M_i(j) = {} then
      remove M_i(j)
    else
      for k = j + 1; k < 2^{i−1}; k = k + 1 do
        if M_i(k) =M_i(j) then
          remove M_i(k)
        end if
      end for
    end if
  end for
end for
```

Algorithm 2 Link Error Case: Parent Generated Transmission Lists

```
T_i ← Set of nodes reachable by offspring i
N_i ← number of offsprings
for {i = 1; i ≤ N ; i = i + 1} do
  M_i ← {}
  Z ← Arrangements of {0, 1} with repetition taken N−1 by N−1
  for all Z_l (elements of Z) do
    M_i ← M_i ⋃ {T_i ⋂_{Z_l(x)=1} T_x ⋂_{Z_l(x)=0} T̄_x}
  end for
end for
\\Remove duplicates and empty sets
for j = 1; j < 2^{i−1} − 1; j = j + 1 do
  if M_i(j) = {} then
    remove M_i(j)
  else
    for k = j + 1; k < 2^{i−1}; k = k + 1 do
      if M_i(k) =M_i(j) then
        remove M_i(k)
      end if
    end for
  end if
end for
```

Algorithm 3 Node Error Case: Aggregate Computation From Children Messages

```
Q_k ← Set of aggregates that a node manages
R_i ← Set of aggregate values received from child i
RC_i ← Binary vector. 1 if received message form child i, 0 otherwise
U_k ← Set of reconstruction path to k^{th} aggregate
A← {} \\ A_k ← value of aggregate k
for {k = 1; k ≤ #[Q_k]; k = k + 1} do
  A_k ← self
  u← U_k
  for {i = 1; i ≤ #[R_i]; i = i + 1} do
    if not RC_i then
      for {j = i; j ≤ #[U_j]; j = j + 1} do
        remove instances of u related to node j
      end for
    end if
    A_k ← A_k+R_i(arg max u(i) = 1)
  end for
end for
```

However, it should be noted that some of the entries of the aggregation tables are empty sets, while others are repetitions. An empty set occurs, for example, if all offsprings reached by a given node can also be reached by at least one of its siblings. And a repetition may happen if, for example, the set of offspring that a node can reach minus the set of nodes from a given sibling is equal to a similar set excluding another siblings offspring. Therefore, the algorithms herein presented also include an optimization section, in which the generated aggregate set is pruned of duplicate and empty sets, thus reducing the number and size of exchanged messages.

Eliminating redundancies (duplicates and empty sets) allows to perform a significant reduction of messages and aggregates. In sparse networks this optimization would not make much difference, since it is more likely that each node would have a rather distinct set of offspring[2]. However, as the network density grows, there would be more nodes with rather similar offspring tables, thereby the use of this improvement tend to become significant. Nonetheless, it must be stressed that node density control is out of the scope of this paper.

Algorithm 3 describes how the aggregates can be recovered from the received messages in the case of node errors. For each aggregate that a node manages (Q_k), it must inspect the aggregates sent by each of its children (R_i). Then, all the entries that failed are removed from u. Finally, the correct aggregate value is obtained by taking the rightmost element of u that has been received from each descendant. As can be verified, the algorithm is not computationally intensive. The operations carried out are relatively simple and the number of iterations depends on the number of aggregates and on the number of children, which are frequently relatively low values.

Algorithm 4 describes how the aggregates can be recovered

[2]this is not a problem since under this circumstances each node would send only one message with the aggregate of its own value and all its children, i.e. sparse networks have less paths in its multi-path

Algorithm 4 Link Error Case: Aggregate Computation From Children Messages

$Q_k \leftarrow$ Set of aggregates that the node handles
$R_i \leftarrow$ Set of aggregate values received from child i
$RC_i \leftarrow$ Binary vector. 1 if received message form child i, 0 otherwise
$U_k \leftarrow$ Set of reconstruction path to k^{th} aggregate
A$\leftarrow \{\} \setminus\setminus$ $A_k \leftarrow$ value of aggregate k
for $\{k = 1; k \leq \#[Q_k]; k = k + 1\}$ **do**
  $A_k \leftarrow$ self
  E$\leftarrow \{\} \setminus\setminus$set of excluded aggregates
  for $\{i = 1; i \leq \#[R_i]; i = i + 1\}$ **do**
    u$\leftarrow U_k \setminus U_k(RC_i = 1)$
    u$\leftarrow u \setminus E$
    $A_k \leftarrow aggr\{A_k, R_i(u)\}$
    E$\leftarrow E \bigcup \{u\}$
  end for
end for

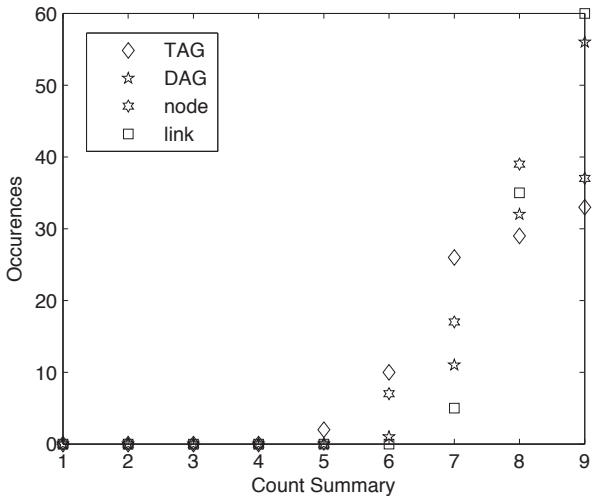


Fig. 3. Link Error Case.

from the received messages in the case of link errors. The process consists in taking the aggregates sent by each children sequentially, and merge them if they have not been already included in a previous children. To keep track of which values have already been processed, it is used an exclusion list (E). In each step this list is appended with the index of all values that have been correctly received ($RC_i = 1$) and not yet merged.

In terms of computational complexity, the algorithm is similar to the previous one.

IV. Evaluation

This section presents simulation results to assess the effectiveness of the approaches proposed in this paper. The simulations were carried out in the Matlab® software, with a standard error model, in which the error probability is as function of the distance $P_r = P_e d^{-\alpha}$, with $\alpha = 2$, transmission power was equal in all the nodes, and the probability of error as $(1 - erf(P_r/N_o))/2$.

In addition to the two algorithms presented in this paper, this section also presents simulations of TAG and the DAG approaches, described in section II. The TAG approach is the simplest of all. It does not have any type of redundancy, using only simple aggregation, thus it will be used as the base line. The DAG approach has two level deep link error correction capability, hence it will be used compare with link error approach presented herein. To the best of the authors knowledge, there is no approach that can be used to make a direct comparison with the node error case.

The simulation was made with a network with 9 nodes placed randomly with a uniform distribution in a square of side $2.7m$. The sink was chosen randomly, therefore, not being necessarily in one of the edges of the network. Communication parameters were tuned to ensure a communication range of about $2m$ with an error probability of 0.1. 100 simulations were done on the same network for each case, i.e. each protocol and each error mode. All nodes were programmed to act as if they had read a 1 from their respective sensor (including the sink) which allowed to have a global view into the number of nodes that were successfully aggregated. Recall that TAG is single path, hence in all case the nodes that were aggregated, were done so once. Results are as follows:

From figure 3, the link error case behaved as expected with approach tuned to link case having most of its occurrences with the correct reception of all nodes. The DAG approach also presented a reasonable/similar behavior. From figure 4, the node error case presents a small discrepancy with the expectation, i.e. the DAG approach behaved a little bit better than our approach tuned to the error case, there are a few reasons for this, 1)the DAG approach has a two level deep link error correction capabilities and since the network was small it could correct most of the errors (in fact, DAG in two

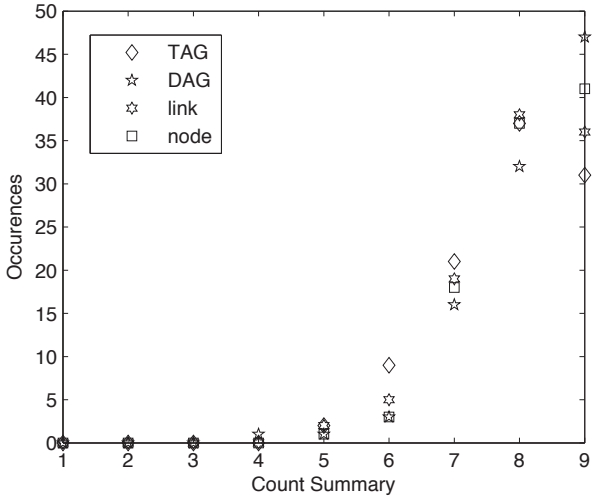


Fig. 4. Node Error Case.

level, i.e. sink plus two, is similar to our link scheme) and 2)the experiences were done in a single network which potentially means that a given protocol could have, out of serendipity, have ended up in a network with many nodes in level 1. These results call for simulations with more nodes, more levels and also simulations with several different networks.

V. Conclusions

WSN present data transmission/reception reliability issues, which can be dealt with by the use of multi-path routing. Nonetheless, this solution introduces another problem, namely aggregate reliability in the presence of duplicate-sensitive summaries. This paper presented a mechanism that ensures that the value of such aggregates will be as close as possible to the actual value, namely by taking advantage of redundant paths in the presence of errors and removing duplicates.

The mechanism is focused in the partition of the summaries into several messages that are recomposed to form the best possible message in their way to the sink. Two of such algorithms were devised, one best suited for networks in which the dominant failure mode is link failure and another in which the dominant failure mode is node failure.

Both scenarios were simulated and compared to the standard approach in the literature, having demonstrated a superior performance in their respective failure mode scenarios. However, there was also a reduction in the lifetime of the network.

Future work consists in providing formal proofs of the correctness of the proposed algorithms to generate the aggregates and recover the data, as well as consider less pessimistic failure modes, namely by controlling the number of nodes or links that may fail simultaneously.

References

[1] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," in *Data Engineering, 2004. Proceedings. 20th International Conference on*, march-2 april 2004, pp. 449 – 460.

[2] S. Motegi, K. Yoshihara, and H. Horiuchi, "Dag based in-network aggregation for sensor network monitoring," in *Applications and the Internet, 2006. SAINT 2006. International Symposium on*, jan. 2006, pp. 8 pp. –299.

[3] A. Manjhi, S. Nath, and P. B. Gibbons, "Tributaries and deltas: efficient and robust aggregation in sensor network streams," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '05. New York, NY, USA: ACM, 2005, pp. 287–298. [Online]. Available: http://doi.acm.org/10.1145/1066157.1066191

[4] D. Wu and M. H. Wong, "Fast and simultaneous data aggregation over multiple regions in wireless sensor networks," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 3, pp. 333 –343, may 2011.

[5] S. K. Chong, M. M. Gaber, S. Krishnaswamy, and S. W. Loke, "Energy-aware data processing techniques for wireless sensor networks: A review." *T. Large - Scale Data - and Knowledge-Centered Systems*, vol. 3, pp. 117–137, 2011.

[6] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*, ser. ICDCSW '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 575–578. [Online]. Available: http://dl.acm.org/citation.cfm?id=646854.708078

[7] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," 2002.

[8] A. Woo, S. Madden, and R. Govindan, "Networking support for query processing in sensor networks," *Commun. ACM*, vol. 47, pp. 47–52, June 2004. [Online]. Available: http://doi.acm.org/10.1145/990680.990706

[9] M. Watfa, W. Daher, and H. Al Azar, "A sensor network data aggregation technique," *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, pp. 19–26, April 2009.

[10] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 131–146, December 2002. [Online]. Available: http://doi.acm.org/10.1145/844128.844142

[11] W.-H. Liao, Y. Kao, and C.-M. Fan, "An ant colony algorithm for data aggregation in wireless sensor networks," in *Proceedings of the 2007 International Conference on Sensor Technologies and Applications*, ser. SENSORCOMM '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 101–106. [Online]. Available: http://dx.doi.org/10.1109/SENSORCOMM.2007.23

[12] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using an ant colony optimization (aco) router chip," *Sensors*, vol. 9, no. 2, pp. 909–921, 2009. [Online]. Available: http://www.mdpi.com/1424-8220/9/2/909/

[13] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multi-resolution storage for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 89–102. [Online]. Available: http://doi.acm.org/10.1145/958491.958502

[14] J. Gao, L. Guibas, N. Milosavljevic, and J. Hershberger, "Sparse data aggregation in sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 430–439. [Online]. Available: http://doi.acm.org/10.1145/1236360.1236414

[15] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi, "Holistic aggregates in a networked world: distributed tracking of approximate quantiles," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '05. New York, NY, USA: ACM, 2005, pp. 25–36. [Online]. Available: http://doi.acm.org/10.1145/1066157.1066161

[16] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, "Medians and beyond: new aggregation techniques for sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 239–249. [Online]. Available: http://doi.acm.org/10.1145/1031495.1031524

[17] M. Wu, J. Xu, X. Tang, and W.-C. Lee, "Monitoring top-k query in-wireless sensor networks," *Data Engineering, International Conference on*, pp. 143–147, 2006.

[18] ——, "Top-k monitoring in wireless sensor networks," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, pp. 962–976, July 2007. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2007.1038

[19] K. Ammar and M. A. Nascimento, "On the use of histograms for processing exact aggregate queries in wireless sensor networks," *International Workshop on Data Management for Sensor Networks*, vol. 8, 2011.

[20] A. Manjeshwar and D. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, apr 2001, pp. 2009 –2015.

[21] A. Manjeshwar and D. P. Agrawal, "Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, ser. IPDPS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 48–55. [Online]. Available: http://dl.acm.org/citation.cfm?id=645610.662036

[22] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, "Tina: a scheme for temporal coherency-aware in-network aggregation," in *Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access*, ser. MobiDe '03. New York, NY, USA: ACM, 2003, pp. 69–76. [Online]. Available: http://doi.acm.org/10.1145/940923.940937

[23] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, may 2003, pp. 139 – 148.

[24] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, pp. 7:1–7:40, April 2008. [Online]. Available: http://doi.acm.org/10.1145/1340771.1340773

[25] S. Croce, F. Marcelloni, and M. Vecchio, "Reducing power consumption in wireless sensor networks using a novel approach to data aggregation," *Comput. J.*, vol. 51, pp. 227–239, March 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1349558.1349559

[26] J. Al-Karaki, R. Ul-Mustafa, and A. Kamal, "Data aggregation in wireless sensor networks - exact and approximate algorithms," in *High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on*, 2004, pp. 241 – 245.

[27] A. Dimakis, A. Sarwate, and M. Wainwright, "Geographic gossip: efficient aggregation for sensor networks," in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, 2006, pp. 69–76.

[28] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, oct. 2003, pp. 482 – 491.

[29] R. Sarkar, X. Zhu, and J. Gao, "Hierarchical spatial gossip for multi-resolution representations in sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, april 2007, pp. 420–429.

[30] K.-W. Fan, S. Liu, and P. Sinha, "On the potential of structure-free data aggregation in sensor networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1–12.

[31] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508 – 2530, june 2006.

[32] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: distributed randomized algorithms and analysis," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147741

[33] C. Olston, J. Jiang, and J. Widom, "Adaptive filters for continuous queries over distributed data streams," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '03. New York, NY, USA: ACM, 2003, pp. 563–574. [Online]. Available: http://doi.acm.org/10.1145/872757.872825

[34] C. Olston, B. T. Loo, and J. Widom, "Adaptive precision setting for cached approximate values," in *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '01. New York, NY, USA: ACM, 2001, pp. 355–366. [Online]. Available: http://doi.acm.org/10.1145/375663.375710

[35] C. Olston and J. Widom, "Best-effort cache synchronization with source cooperation," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '02. New York, NY, USA: ACM, 2002, pp. 73–84. [Online]. Available: http://doi.acm.org/10.1145/564691.564701

[36] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, "Hierarchical in-network data aggregation with quality guarantees." in *EDBT*, ser. Lecture Notes in Computer Science, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Bhm, and E. Ferrari, Eds., vol. 2992. Springer, 2004, pp. 658–675.

[37] S. Shah, A. Bernard, V. Sharma, K. Ramamirtham, and P. Shenoy, "Maintaining temporal coherency of cooperating dynamic data repositories," in *In Proc. VLDB*, 2002.

[38] C. Olston and J. Widom, "Offering a precision-performance tradeoff for aggregation queries over replicated data," in *Proceedings of the 26th International Conference on Very Large Data Bases*, ser. VLDB '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 144–155. [Online]. Available: http://dl.acm.org/citation.cfm?id=645926.671877

[39] L. Rossi, B. Krishnamachari, and C.-C. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, oct. 2004, pp. 460–469.

[40] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, ser. IPSN '04. New York, NY, USA: ACM, 2004, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/984622.984624

[41] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '05. New York, NY, USA: ACM, 2005, pp. 402–413. [Online]. Available: http://doi.acm.org/10.1145/1062689.1062739

[42] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, ser. HICSS '00, vol. 8. Washington, DC, USA: IEEE Computer Society, 2000, pp. 8020–. [Online]. Available: http://dl.acm.org/citation.cfm?id=820264.820485

[43] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, ser. VLDB '04. VLDB Endowment, 2004, pp. 588–599. [Online]. Available: http://dl.acm.org/citation.cfm?id=1316689.1316741

[44] A. Jindal and K. Psounis, "Modeling spatially-correlated sensor network data," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, oct. 2004, pp. 162 – 171.

[45] M. C. Vuran and I. F. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 316–329, April 2006. [Online]. Available: http://dx.doi.org/10.1109/TENT.2006.872544

[46] S. Yoon and C. Shahabi, "The clustered aggregation (cag) technique leveraging spatial and temporal correlations in wireless sensor networks," in *ACM Trans. Sen. Netw.*, vol. 3. New York, NY, USA: ACM, March 2007. [Online]. Available: http://doi.acm.org/10.1145/1210669.1210672

Architecture for orchestration of M2M services

Gustavo Pires, Mário Antunes, Daniel Corujo, Diogo Gomes, João Paulo Barraca, Rui Aguiar

Instituto de Telecomunicações, Universidade de Aveiro, Aveiro, Portugal
{gmpp, mario.antunes, dcorujo, dgomes, jpbarraca}@av.it.pt, ruilaa@det.ua.pt

***Abstract*—The past few years, miniaturization has allowed us to imbue computers into everyday devices. This in turn has enabled these devices to communicate with each other, and in doing so, allows us to collect a wealth of information, more accurately and with greater availability than ever before. This phenomenon is known as the Internet of Things. It allows smart environments to truly behave in an intelligent manner by using information collected from the devices mentioned above. However, it's necessary to model how the gathered data will influence the behavior of a smart environment. This open problem can be approached as a machine to machine (M2M) orchestration.**

In this paper we present a new architecture for M2M orchestration. This new architecture will be based around a platform that creates orchestration processes through a graphical interface. Through this interface a business process execution language (BPEL) service will be made and deployed on an enterprise service bus (ESB). Alongside this, we are also developing a collection of services that will be used for the purposes of implementing a smart environment.

***Index Terms*—Orchestration, Machine to Machine, Smart Environments**

I. Introduction

In the past few years, we have seen a massive increase in the number of smart devices. In fact, according to the ICT KTN [1], the number of these devices is expected to increase worldwide from 4.5 billion in 2011 to 50 billion by 2020. These devices are able to communicate with each other, sending relevant information about their environment and coordinating their actions. This trend is known as the Internet of Things (IoT).

These connected devices can be seen as an unused source of context information. In this work we will consider context as defined by Abowd and Dey [2],

> Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

According to the definition, context information is virtually any type of information as long it is related to some entity presented in the considered environment.

One computational area that will benefit from this new source of context information is the area of Smart Environments. Smart Environments have lacked the autonomy and adaptability necessary for truly intelligent actions. This was in part caused by the absence of reasoning regarding the user's actions and status, in spite of the entire environment being flooded by huge amounts of information. However, context information cannot be obtained just by gathering this information. It first must be processed and seen what relation each information source has with each other so that the context related to that information can be inferred.

One way to look at this problem is as an M2M orchestration issue. M2M is the networking of intelligent small devices that manage themselves and exchange information between them without human intervention. Orchestration is the automated coordination of a group of systems, organized by a central entity. As seen in Figure 1, a smart environment is made of a group of independent services. These services can be organized as an orchestration process.

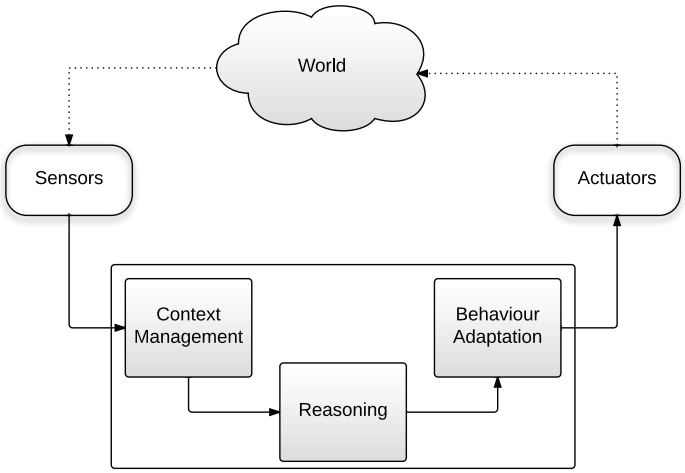


Fig. 1. Conceptual Architecture of a Smart Environment

In this paper we will present a novel architecture to enhance M2M orchestration. The proposed architecture is based around a platform that creates orchestration processes through a graphical interface. Through this interface it's possible to generate BPEL services by graphically connecting existing services. These services will then be deployed on an ESB and can also be used to compose new services. New services are also being developed for the purpose of supporting smart environments. Among others, services that extract information from a low level layer and services that reason about inhabitants' context.

A new reasoning services is also being developed to extract relevant knowledge from non-structured context information. By relevant knowledge, we mean the correct behavior of the environment. This knowledge will then be used by the smart

environment's actuators to enforce those actions. The novelty of this service is the ability to use machine learning techniques as opposed to static ontologies or manually defined rules. This allows it to dynamically adapt to its present situation. For example, differences in the deployed environment would not mean that the system would need to be reconfigured, but instead it would dynamically adapt to the new environment.

The APOLLO project main objective is the development of a platform that supports new services in the area of machine-to-machine (M2M) communications. The project aims to develop a transversal technological platform that supports management, control and monitoring of an heterogeneous network of sensors and actuators. APOLLO exports a service layer to third parties willing to develop next generation M2M applications in various areas including Utilities, Transports, Health, Agriculture, Distribution and Consumer Electronics. The project platform will support from its start a vast set of M2M Smart Services & Applications such as Smart Metering, Smart Grids, m-Health (remote monitoring of patients), Smart Cities, Smart Home and Smart Buildings according to a Portuguese Government policy for the deployment of next generation networks.

In Section II we will discuss the current state of the art related to orchestration of M2M services and smart environments. Section III will expose the architecture, its advantages and drawbacks. In Section IV we will discuss the conceptual advantages of the proposed architecture and present future work related to this project.

II. State of the art

A Service Oriented Architecture (SOA) [3] supplies resources as self-contained, independent services that can then be reused. By doing so, these supplies can be maintained in a more strict fashion and be used by any number of other higher level services, even from an external entity to the one that created them.

Currently, in SOA implementations, orchestration has gained more popularity than choreography. The reason for this is that choreography presents several implementation problems, regarding communication overheads, even though it brings some advantages over orchestration [4]. In orchestration, only a central coordinator has to know the process, and the surrounding services can just worry about handling requests, not being involved in the management of process, and behave like any ordinary service. Choreography on the other hand, needs every process to know how it will intervene in the process, requiring information regarding other services' statuses. This means that choreography has potential to be more robust at the cost of added complexity.

The defacto standard language used to describe orchestration processes is BPEL [5]. This language describes each participating service's role and presents the work flow for the interactions between them. However, this language depends purely on services described through WSDL [1].

[1]http://www.w3.org/TR/wsdl

Barricelli et al. [6] proposed an architecture that allows domain experts to compose services through a graphical editor. The editor uses a semantic search engine to locate relevant services and an orchestration engine to handle the orchestration between these same services. The end product is a BPEL workflow document which can then be used to create services that execute the described process.

The typical scenario between participating services in a BPEL process is by definition an M2M scenario and so we can model any number of M2M scenarios as a BPEL process. As mentioned before, smart environments can be modeled as an M2M orchestration. The focus of this paper is on M2M orchestration of smart environments. Currently smart environments are developed as tightly coupled services that can only be orchestrated in a single manner. Below we present the most relevant smart environment projects.

Mozer [7], [8] applied neural networks [9] in home automation. The author developed a system that was able to control air, heating, lighting, ventilation and water heating. The main goal of the project was to anticipate inhabitants needs and conserve energy at the same time. For that, it applied reinforcement learning [9]. During a learning period, inhabitants indicated their preferences whenever predictions were incorrect by selecting themselves what they would expect the system to do (for instance, if they want the lights on, and the system did not turn them, users simply turned them on). This way, the system would adjust itself to its inhabitants preferences.

Vainio et al [10] proposed home-control system that uses fuzzy logic rules [11]. Initial rules were given manually and, through reinforcement learning, the home adapted by replacing old rules with new ones. This method was applied to control a lighting system in a smart-home laboratory environment. The author concluded that users did not care if the rules generated by the system didn't match exactly what they wanted. It also concluded that, after a certain amount of time, if a rule had a big weight, it was likely to keep its importance, and new rules, that correspond to sporadic actions, were quickly eliminated.

These smart environment systems use tightly coupled components. We hope that our proposal sparks interest in loosely coupled smart environment systems, and in doing so make it easier to build and customize these kinds of systems. Loosely coupled smart environments systems would also make it easier to add/remove/modify components during runtime.

III. Conceptual architecture

The proposed architecture can easily, through a graphic interface, orchestrate smart environments related services. These services can be split into two categories: reasoning services and context awareness services.

Reasoning services are services that process context information and imbue the environment with the necessary behavior to improve the inhabitants' quality of life. **Context awareness services** provide a bridge between the environment and the system. Through the sensors it can perceive the environment's

context information, while at the same time it can use actuators to enforce changes onto the environment.

This section will be divided into three subsections relating to each of these kinds of services and to the orchestration application behind them. In Subsection III-A we integrate the necessary concepts for a flexible M2M communication framework. In Subsection III-B we present the reasoning mechanisms that will be used by the smart environment. Finally in Subsection III-C we show the orchestration application that will be able to use both of these in order to actually make the smart environment work.

A. Context Awareness

In order to not only access the information provided by different kinds of devices in Smart Environments, but to also allow their control, a flexible communication and interfacing framework needs to be set in place. Considering as well that such environments are of a widely heterogeneous nature, where involved devices support disparate sets of resources and are accessed through a myriad of networking technologies, the underlying communication framework is furthermore placed under stringent requirements: not only must it be able to simultaneously cater to the processing restrictions of devices with greater or lesser CPU and memory capabilities, but it has also to have the ability to provide information with a meaningful degree of usefulness despite those capabilities.

In this sense, our framework exploits the information gather and control capabilities of MINDiT [12], a flexible cross-layer interfacing architecture. This architecture leverages Media Independent concepts, such as the ones presented in IEEE 802.21 [13] and IEEE P1905 [14], where an abstraction layer allows the interexchange of information between lower layers (e.g., link access interfaces) and higher layers (e.g., application and service processes) under a common interface. In this sense, the conception and deployment of applications (and associated operating system mechanisms) aiming to obtain information and operate the different links, is simplified, with the middleware layer abstracting commands, events and information definition, not only locally but also remotely towards other agents located elsewhere on the network.

However, such media independent concepts are tailored concerning a well defined scope, where the commands, events and information elements supported by IEEE 802.21 concern handover optimization and, for the IEEE P1905 case, hybrid home networking. Achieving a flexible approach, MINDiT goes beyond the static nature of the interfacing capabilities of the previously mentioned technologies, and provides the means for dynamic interface definition and retrieval in true heterogeneous Smart Environments. Fig. 2 illustrates the deployment of the generic interfacing mechanisms provided by MINDiT on our framework, based on the concept proposed in [12].

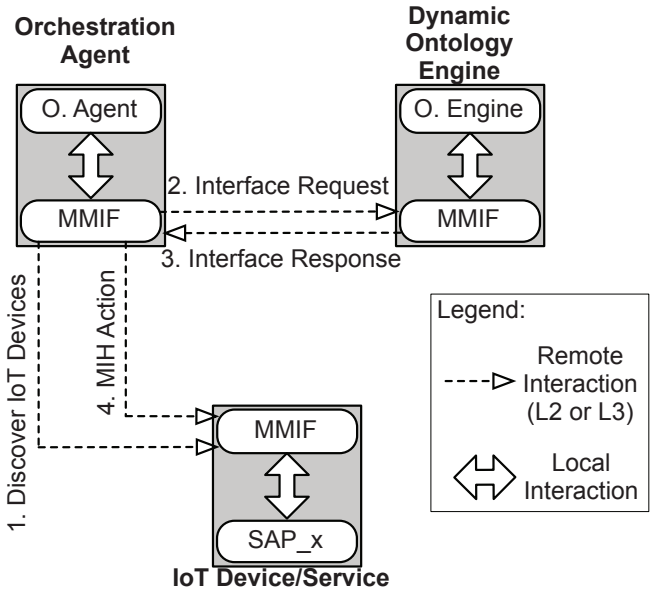


Fig. 2. MINDiT Interfacing Mechanism

The interfacing architecture of our framework relies on information exchanges between three different entities, composing an IoT Device or Service, an Orchestration Agent and an Ontology Engine.

- **IoT Device or Service:** The flexibility of our framework allows to reach for information, and execute control primitives, not only over physical aspects of a device (e.g., activate a mechanical arm to execute a precise movement, command a wireless interface to scan for access points), but also in services running in those devices or in network elements as well (e.g., switch video codec to reduce required bandwidth, trigger mobility handover procedures). The supported control and information gathered procedures are defined in the Service Access Point (SAP_x), composing the description of the list of supported commands by that device or service. To access a SAP, the node which composes the device or contains the service, is coupled with a MINDiT Media Independent Function (MMIF). This entity is a core component of this information interexchange architecture, aggregating a series of interfacing abstraction and control mechanisms:
  1) *Local and remote information exchange:* the MMIF acts as an intermediate abstraction control point for IoT communications, by receiving commands sent from agents (internal or external to the device), and collecting informational events from the SAPs. In this sense, it manages transaction state regarding these interactions, and is able to use Layer 2 and Layer 3 transport. Remote interaction is always executed between the MMIFs of the two separate entities involved.
  2) *Agent discovery and capability exchange:* the MMIF, when requested by a local or external agent,

is able to respond with the supported capabilities of the local node, regarding existing SAPs and respective supported commands. Likewise, it is also able to proactively broadcast this information in a beacon-like nature, allowing its discovery.

3) *Event registration:* agents are able to register for the reception of information events, when these are generated by the SAPs and conveyed to their local MMIF. Therein, the MMIF analyses if any other entity has requested for their reception and sends that information accordingly.

- **Orchestration Agent:** This agent composes a service or application entity implementing behaviour able to consume information and control from other devices or services. Through the knowledge of existence of other MINDiT-enabled entities (both remote and local), coupled with the knowledge of their supported commands and events provided by their SAP, the agents are able to execute the operations associated to their service logic, with support form MINDiT. It is important to notice that, in order to use these mechanisms, the agents need to be coupled with an MMIF to manage this information exchange. Moreover, the MMIF has to be seen as a complement component contained by the agent, and not as the core component itself (i.e., the agent uses MINDiT and not the other way around). In Fig. 2, the Orchestration Agent's logic establishing the bridge between its service logic and the operation with the MINDiT framework is pictured as a MINDiT Media Independent User (MMI-User).
- **Dynamic Ontology Engine:** The Ontology Engine composes the interaction with the orchestration and context structuring components of our framework, in terms of device and service interfacing. Concretely, the different capabilities of devices and services in IoT environments not only generate a plethora of different SAPs with different capabilities, but are also subject to different perceptions and interpretations of the raw data elements and parameters with which they interact. As an example, a temperature sensor can provide information in Celsius or in Fahrenheit. An Orchestration Agent needs to be aware that the numerical input sent by these devices can vary widely in definition. Moreover, the controlling capabilities also share this concern. As such, MINDiT utilizes the knowledge extracted from non structured information capabilities of our framework, to bridge the commands and parameters advertised by the SAPs of services and devices, to the necessities identified by the intended agent's behaviour.

The interactions involving these three entities are also identified in Fig. 2, following the following process:

1) **Discover IoT Devices:** Agents, motivated by their service behaviour, wish to interact with different IoT devices and/or services. Under our framework, the agents use MINDiT mechanisms to discover other MINDiT-enabled IoT entities. This discovery provides initial information indications that allow the agent to analyse which IoT entities fall within its scope of interest. For example, the announcement of the tags *#sensor* and *#temperature* provide insight to the agent that this entity deals with temperature sensing.
2) **Interface Request:** With this information, the agent is able to query the Dynamic Ontology Engine providing the received tags, alongside any meaningful information for the task belonging to its service logic (i.e., requirements, network domain, identifiers, used access technology, amongst others).
3) **Interface Response:** The Dynamic Ontology Engine extracts the necessary SAP interface from the available non structured information repository, which has been previously populated by service provides and device manufacturers with the necessary MINDiT SAP information. In this way, the MMI_User is able to gain knowledge of the SAPs for interfacing with the intended devices.
4) **MIH Action:** Empowered with the knowledge of the commands and parameters supported by the target IoT device or service, the agent is able to send a Media Independent Handover (MIH) Action command. The structure of this message is based on the one defined by the IEEE 802.21 standard regarding MIH commands. Unlike that standard, where each command has a separate structure defined by Type, Length and Value (TLV), in MINDiT we have leveraged a generic TLV structure able to identify the different commands made available by the different SAPs, as seen in Fig. 3. In this way, not only is the agent able to retain the MIH management mechanisms (e.g., transaction management) via the MIH Header as well as through indicating the MMIF Source and Destination identifiers, it is also able to specify the identifier for the target SAP, which is the intended Action and the resulting value. This ensures a more flexible approach in terms of defining the intended behaviour to be executed in the target IoT Device or Service, as verified in [12].

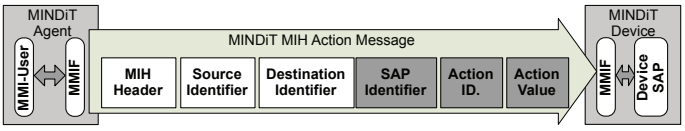


Fig. 3. MINDiT Generic MIHF Action Message

Through these mechanisms, our framework uses the MINDiT architecture to achieve the following capabilities:

1) To discover interfaceable IoT devices and services, along with their media independent link access connectivity information (e.g., IP address, MAC address, network domain, transport layer port, protocol, etc.);
2) To register agents into those IoT devices and services, allowing them to exchange connection establishment

data (e.g., to support authentication and other security processes) as well as to manifest interest in collecting informational events when triggered;

3) To dynamically request the interfacing primitives and parameters supported by the IoT device or service's SAP, by interacting with the non structured information repository from our framework;
4) To send those commands, and collect back information or command results, towards the target IoT device or service.

All this behaviour is achieved through a single generic message exchange definition, which can occur at either Layer 2 or Layer 3, according to the network and devices capabilities, which places a low impact on application service design, but providing still meaningful information content and capabilities to high-level processing. Contrary to other solutions employing resource-consuming service-oriented communication procedures that place stringent requirements over IoT devices (e.g., Devices Profile for Web Services (DPWS), Efficient XML Interchange (EXI) or Web Service Definition Language (WSDL), or on the other end, solutions that focus on wireless sensor link optimization aspects [15], MINDiT empowers our framework with a flexible, generic single protocol for IoT device interfacing.

B. Intelligent Systems

The APOLLO project's main objective is to export a service layer that allows third parties to develop next generation M2M applications. Smart environments are important applications that can be developed using the APOLLO service layer. In order to develop smart environments it is necessary to provide reasoning services. These reasoning services are necessary to comprehend the environment context through the sensors. Due to the heterogeneous nature of the information provided by the sensors and the dynamic ambient associated with smart environments it is difficult to develop efficient reasoning services. In this subsection we will discuss the difficulties of developing reasoning systems for smart environments and proposed a method that is able to cope with heterogeneous environments.

As previously mention, context information is any information that can be used to characterize an entity's situation. It is important to notice that this definition of context information does not specify any structure to share context information. This is strongly related with the heterogeneity associated with context information. Due to this heterogeneity, it is difficult to store context information into a relational database, as show in Figure 4. Context is better modeled as a continuous stream of information [16]. This approach has the advantage of facilitating the storage of the context information. On the other hand this approach does not enforce a relational model. The majority of knowledge extraction techniques rely on the relational model and the relations defined by it.

The most common approach to minimize this problem is to define an ontology. In computer science, an ontology is a description of the concepts and relations that can exist

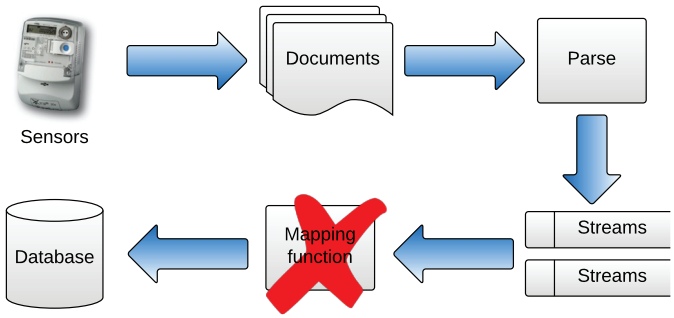


Fig. 4. Due to the heterogeneity associated to smart environments there is no dynamic function to correctly map an unstructured group of streams into a relational database.

between them [17]. Currently several context-aware systems use ontologies to map the continuous streams of data into a relation model. After that, conventional techniques can be applied in the model in order to extract meaningful knowledge. However, the use of ontologies implies a tradeoff between the quantity of represented concepts and the number of different scenarios the system is able to support. Another important issue about the use of ontologies is that users have to define all the relations between concepts. In other words the reasoning process becomes limited to the relations previously defined by users.

Currently we are developing a method to extract knowledge from unstructured context information that does not require static ontologies. The main objective of the proposed method is to learn the underlying model of unstructured information through machine learning techniques. The underlying model can be perceived as a dynamic ontology that is defined by the system itself.

Figure 5 shows the conceptual architecture of the platform. At this stage of development i is not relevant for the platform how the sensors send information. Document, in this context, represents a semi structured file (i.e. xml, json or cvs files) that contains data sent from some sensors. The platform analyses data from sensors and computes the underlying model, i.e. a dynamic ontology. The data is rearranged according to the underlying model. After this step conventional knowledge extraction methods are used to find relevant patterns. The majority of knowledge extraction frameworks provide support and confidence measures. These metrics are used by the model extraction process to determine the quality of the underlying model.

Figure 6 shows an expansion of the platform. The received documents are parsed and stored into a group of streams of data. Each stream is identified by the names extracted from the entity contained in the documents. E.g. if a document contains an entity named temperature and an associated value of 25, this value will be stored in a stream named temperature.

Two different analysis methods are used on the streams: statistic analysis and semantic analysis. First correlations matrices (statistic analysis) are computed based on the values on the streams. A strong mathematical correlation can be

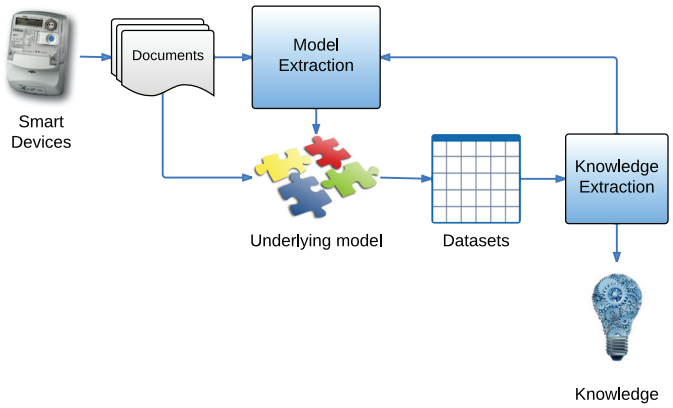


Fig. 5. Architecture of the knowledge extraction platform.

perceived as a relation on the underlying model.

After semantic analysis is used to discover relations between streams' names. The meaning of each streams' name is retrieved from a semantic web service. Text mining techniques are used to detect relations in the meanings of the streams' names. A graph of concepts is built based on the relations discovered. Each node in the graph represents a stream's name, while each edge represents a relations between two concepts.

After clustering algorithms are used to group the related streams. The results of the statistic (correlations matrices) and semantic analysis (graph of concepts) are the distance metrics used by the clustering algorithms. The underlying model is composed by the groups of concepts and relations returned by the clustering algorithm.

These type of analysis are subject to ambiguities in the concepts and relations. Association rules are used to detect relevant patterns in the rearranged data. The model extraction process uses the support and confidence framework to solve ambiguities in the concepts and relations. This feedback loop can be perceived as reinforcement learning.

Conceptually this approach offers some interesting advantages:

1) The proposed platform does not require the definition of static ontologies.
2) The dynamic model can cope with the evolution of concepts and relations.
3) The platform has the potential to discover unknown relations.

Currently we are finishing some details in the architecture of the knowledge extraction service. There are some open issues, e.g. how the platform should optimize based on the knowledge extraction feedback, what clustering algorithm should be used, what semantic services offers relevant information. These concerns are being addressed in a PhD Thesis. After solving these open issues the platform will be implemented as part of APOLLO project.

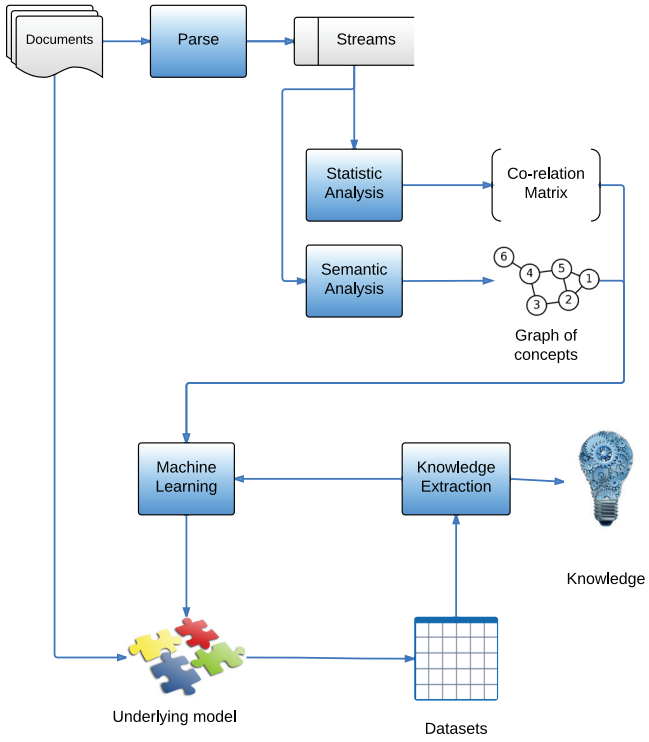


Fig. 6. Architecture of the model extraction process.

C. Orchestration

Smart environments need to cope with a diversity of underlying hardware and different user needs. One way to cope with this diversity is through the use of a SOA. A SOA allows components to be modules as independent services, creating a loosely-coupled system. It also allows the use of external modules that would be otherwise unavailable, such as proprietary services in which the implementer is not interested in revealing the internal logic. While this last one is beyond the scope of a typical smart environment, when one considers a city-wide deployment, having such capabilities starts having interesting applications.

One problem users face in a smart environment is that they currently might not be able to configure the behaviors they want the system to take. Since they may not be able to program, either time constraints or skills, the use of services together with a way of providing orchestration capabilities for them is one of the most promising options. As such, we decided to allow users with little programming skill, as well as domain experts to orchestrate services as they please through a supposedly easy to use graphical interface.

The orchestration architecture can be seen in Figure 7. We can divide it into four major components:

- Orchestration Creator Service
- Graphical Orchestration Interface
- Enterprise Service Bus
- Services

The first two are this paper's contribution to this orches-

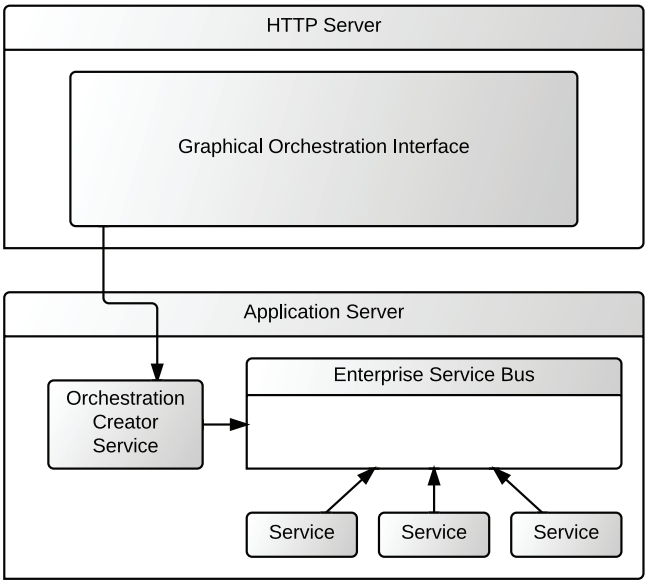


Fig. 7. The orchestration architecture

tration architecture. In fairly simple terms, the Graphical Orchestration Interface will obtain a list of available services. It will then provide a graphical representation of these services for the user to compose higher level services with. Afterwards, it creates a process description from the users input and call the Orchestration Creator Service. This service will then create and deploy a new service based on the obtained process description.

The Graphical Orchestration Interface is being implemented in HTML/JavaScript using Three Node JS[2], thus ensuring that most devices will be able to use it. The process description language to be used is most likely going to be BPEL. However, the current BPEL specification can only handle services described through WSDL. This limitation restricts the ability to orchestrate services that do not naturally use such description language, such as those using REST. These kinds of services are often much simpler and can be implemented using a wider variety of devices, increasing the scope of the Internet of Things. A number of BPEL extensions have been proposed over the years to address this restriction [18]–[22]. Regarding this issue, we are still in the process of researching how we should make BPEL processes interact with WSDLless services such as REST.

As for the ESB, we'll use is JBoss's Switchyard[3] due to a business requirement and as such the we'll also use JBoss AS[4].

IV. Conclusions

Current smart environments are not flexible enough to be considered truly intelligent. They use tightly coupled components and cannot cope with changes in the smart environment, e.g. adding new sensors to the system. In this paper we introduced a new conceptual architecture for M2M orchestration that conceptually solves these issues. With it, we proposed a new intelligent system to extract context from non-structured information and a new orchestration mechanism to allow users of little programming skill to create processes based on deployed services.

We hope this architecture will facilitate the proliferation loosely coupled smart environments that do not rely on static ontologies or only user defined rules. Our orchestration platform enables the use of a set of services that ranges from user defined rules to truly intelligent systems. Additionally, the move to a SOA will help standardize the way smart environment are controlled and allow the collaboration of several smart environments, potentially enabling smart environments of arbitrary sizes to be built with ease.

Currently, we're finishing an implementation of the context management component based on the XMPP protocol and key-value databases. Afterwards, we'll design the reasoning services provided by the APOLLO platform. As previously mentioned, we are still trying to make BPEL processes interact with WSDLless services such as REST. After analysing Switchyard's framework, we decided to implement a new component that enable services to access the service registry.

[2]https://github.com/idflood/ThreeNodes.js
[3]http://www.jboss.org/switchyard.html
[4]http://www.jboss.org/jbossas

V. Acknowledgement

This work has been partially funded by the Portuguese Innovation Agency/ National Strategic Reference Framework (AdI/QREN) under grant agreement No. 2011/021580 (APOLLO project).

References

[1] U. F. I. S. Group, "Future internet report," ICT Knowledge Transfer Network, Tech. Rep., May 2011.
[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, ser. HUC '99. London, UK,: Springer-Verlag, 1999, pp. 304–307. [Online]. Available: http://dl.acm.org/citation.cfm?id=647985.743843
[3] M. Bell, *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Wiley Publishing, 2008.
[4] S. Quinton, I. Ben-Hafaiedh, and S. Graf, "From orchestration to choreography: Memoryless and distributed orchestrators," in *FLACOS'09 Workshop Proceedings*, 2009.
[5] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, "Bpel4ws, business process execution language for web services," IBM, Tech. Rep., 2003.
[6] B. R. Barricelli, P. Mussio, S. Valtolina, M. Padula, P. L. Scala, and A. Piccinno, "Visual workflow composition through semantic orchestration of web services," in *Proceedings of the International Conference on Advanced Visual Interfaces*, ser. AVI '10. New York, NY, USA: ACM, 2010, pp. 405–405. [Online]. Available: http://doi.acm.org/10.1145/1842993.1843079
[7] M. Mozer, "The neural network house: An environment that adapts to its inhabitants," in *Proceedings of the American Association for Artificial Intelligence*, A. Press, Ed., 1998, pp. 110–114.
[8] ——, "Lessons from an adaptive home," in *Smart Environments: Technology, Protocols and Applications*, D. J. Cook and S. K. Das, Eds. John Wiley & Sons, Inc., 2005, pp. 271–294.
[9] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[10] A.-M. Vainio, M. Valtonen, and J. Vanhala, "Proactive fuzzy control and adaptation methods for smart homes," *IEEE Intelligent Systems*, vol. 23, pp. 42–49, 2008.

[11] J. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345 –377, mar 1995.

[12] D. Corujo, M. Lebre, D. Gomes, and R. Aguiar, "Mindit: A framework for media independent access to things," *Computer Communications*, vol. 35, no. 15, pp. 1772 – 1785, 2012, smart and Interactive Ubiquitous Multimedia Services. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412000734

[13] IEEE, "Ieee standard for local and metropolitan area networks- part 21: Media independent handover," *IEEE Std 802.21-2008*, 21 2009.

[14] "Ieee draft standard for a convergent digital home network for heterogeneous technologies," *IEEE P1905.1/D06, September 2012*, pp. 1 –125, 11 2012.

[15] W. Masri and Z. Mammeri, "Middleware for wireless sensor networks: A comparative analysis," in *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on*, sept. 2007, pp. 349 –356.

[16] N. Santos, . O. M. Pereira, and D. Gomes, "Context storage using nosql," in *Proc. 2011 Conferência sobre Redes de Computadores*, Coimbra, Portugal, November 2011.

[17] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal Human-Computer Studies*, vol. 43, no. 5-6, pp. 907–928, November 1995.

[18] C. Pautasso, "Bpel for rest," in *Proc. of the 6th International Conference on Business Process Management (BPM 2008*, 2008.

[19] ——, "Restful web service composition with bpel for rest," *Data Knowl. Eng.*, vol. 68, no. 9, pp. 851–866, September 2009.

[20] T. A. S. Foundation, "Restful bpel, part i," 9 2012. [Online]. Available: http://ode.apache.org/restful-bpel-part-i.html

[21] ——, "Restful bpel, part ii," 9 2012. [Online]. Available: http://ode.apache.org/restful-bpel-part-ii.html

[22] J. Nitzsche, T. van Lessen, D. Karastoyanova, and F. Leymann, "Bpel light," in *5th International Conference on Business Process Management (BPM 2007)*. Springer, Sep. 2007.

Adaptive Multimedia Transmission in Wireless Sensor Networks

Daniel G. Costa[1,2,3], Luiz Affonso Guedes[2], Francisco Vasques[3], Paulo Portugal[3]
[1]Department of Technology, State University of Feira de Santana, Brazil
[2]Department of Computing and Automation, Federal University of Rio Grande do Norte, Brazil
[3]IDMEC/ISR, Faculty of Engineering, University of Porto, Portugal
Email: {danielgcosta@uefs.br, affonso@dca.ufrn.br, vasques@fe.up.pt, pportugal@fe.up.pt}

***Abstract* — Multimedia transmissions over wireless sensor networks may provide valuable information of the monitored field, enhancing monitoring and control applications. However, multimedia streaming can considerably shorten the expected lifetime of battery-operated sensor networks due to the large amount of information to be transmitted from multimedia-enabled source nodes. On the other hand, multimedia data may have different relevancies for the application, depending on the performed monitoring tasks and the application requirements. In such context, we propose an adaptive context-aware energy-efficient multimedia transmission approach, where intermediate nodes may consider its current residual energy level and preconfigured energy thresholds to decide if multimedia packets must be relayed to the next hop or silently discarded. Doing so, energy is saved and the expected network lifetime is enlarged, with controlled impact on the overall monitoring quality.**

***Keywords* — Wireless sensor networks; Energy-efficient adaptive transmission; Multimedia streaming; Packet relaying.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been employed in a large series of monitoring and control applications, changing the way information is harvest and retrieved from the monitored field [1][2]. Besides scalar information such as humidity, pressure, temperature and luminosity, sensor nodes may be endowed with multimedia sensing units, enriching monitoring applications with audio, still images and video streams [3][4]. The resulting Wireless Multimedia Sensor Networks (WMSNs) present new possibilities for monitoring and control applications, but also bring new challenging issues that need to be addressed.

In typical wireless sensor networks, nodes are constrained in energy, processing and memory resources due to the effort to reduce its cost, in order to allow its massive deployment [5]. Energy constraints bound the wireless communication range and impose controlled use of transmission and processing functions. Furthermore, processing and memory constraints can impact the use of buffering techniques by communication protocols, as long as the execution of optimization codes. In addition to the fact that wireless sensor networks may be deployed in regions with lack of infrastructure, or even in hazard or hard access areas, sensors nodes are expected to be autonomous and battery-operated. It is also expected that monitored data will flow upstream through ad hoc multihop communication links [6].

Transmitting multimedia data in WSNs is indeed a challenging task, either due to the large amount of data to be transmitted or due to the real-time nature of some multimedia monitoring applications. In fact, it is expectable that much more energy will be consumed in the transmission of packets over wireless links than in storing and processing operations [7]. In other words, the network lifetime directly depends on the energy consumption rate, and multimedia streaming is expected to consume much more energy than transmission of scalar data [4].

Wireless multimedia sensor networks may be employed to retrieve any combination of audio, still images, video streams and scalar data. Specialized source nodes may be deployed to retrieve a particular type of data or multipurpose sensor nodes equipped with a low-power camera and a microphone may be employed, resulting in different configurations of homogeneous or heterogeneous networks. Whatever the case, packets carrying pieces of audio, image, video or scalar data from different source nodes may be crossing the network toward the sink, whether sources' transmissions are continuous or triggered-based. As packet relaying consumes most energy and transmission paths may be disabled due to energy depletion of intermediate nodes, the total amount of information transmitted over the network should be minimized as much as possible, potentially prolonging the network lifetime. Such energy depletion is more harming in braided-paths [8], where a single intermediate "hub" node may relay packets from more than one source. As these intermediate nodes belong to more than one path, they are expected to receive more combined upstream traffic than other intermediate nodes, thereby spending more energy in average.

Transmission of multimedia data presents different energy consumption patterns. Depending on the nature of the monitoring applications, they may also have different relevancies. For example, in wildlife observation, visual

information might be more relevant than audio. On the other hand, audio streaming may be more desirable for an underwater whale monitoring system. Furthermore, multimedia data may be only retrieved to complement traditional scalar information. Whatever the case, we expect that each application defines a monitoring profile, indicating the relevance of different types of data, concerning multimedia and scalar data alike. As different levels of relevance may be identified, data packets may be prioritized according to their payloads.

We propose in this paper an adaptive multimedia transmission approach where each non-mobile intermediate node decides to relay multimedia data packets to the next hop according to its residual energy level and to the packets' priority. Each priority is associated to an energy threshold to be preconfigured in the desired intermediate nodes. As the energy level of intermediate nodes will decrease along the time, the overall amount of data to be relayed will also decrease, potentially saving energy and prolonging the network lifetime. We performed simulations over the proposed approach in order to highlight the expect energy saving over traditional relaying mechanisms.

The remainder of this paper is organized as follows. Section II presents some related works. Section III brings the statements and definitions of the proposed approach. Simulation results are presented in Section IV, followed by conclusions and references.

II. RELATED WORKS

Many works in recent years have been investigating multimedia streaming in wireless sensor networks [4]. Issues as energy consumption, real-time data delivery and error and congestion control have been major concerns [6] [9], leading research in this area. In such context, we are especially interested in research works that have influenced our investigation.

The work in [10] exploits multipath routing along with the relevance of multimedia data for efficient path selection. Authors define the MPMPS (Multi-Priority Multi-Path Selection) algorithm to find paths with lower end-to-end delay for multimedia streaming, considering a set of available node-disjoint paths [11]. As in [12], the source video stream (72 kbps) is split into an image stream (48 kbps) and an audio stream (24 kbps), giving to each resulting substream a particular priority according to the current monitoring being performed. The node-disjoint paths with lower delay are assigned to the higher priority substreams, leaving the remaining paths to the lower priority substreams. In a different way, the work presented in [13] proposes a transmission mechanism where each intermediate node decides to relay image packets to the next hop according to its residual energy and the packets' priority. Each transmitted packet has a priority level according to the relevance of Discrete Wavelet Transform (DWT) subbands. Doing so, the quality of images transmitted from any source node will be reduced but energy saving is achieved, specially benefiting communications over braided-paths. Similarly, we presented in [14] an energy-efficient packet relaying approach where intermediate nodes forward packets to the next hop toward the sink according to their residual energy level and predefined energy thresholds. However, in that work, packets' priorities are established based on the sensing relevancies of visual source nodes.

We indeed propose herein an energy-efficient packet relaying approach, but in a different way of [13] [14], the type of the retrieved data is considered when setting the packets' priorities, as in [10][12]. Basing the forwarding decision in the current monitoring tasks, we achieve an adaptive context-aware relaying approach, offering a new possibility for multimedia packet transmission over wireless sensor networks.

III. ADAPTIVE ENERGY-EFFICIENT MULTIMEDIA TRANSMISSION

After deployment, many nodes will be employed to compose a multihop ad hoc wireless communication infrastructure. Such intermediate nodes will relay data packets from source nodes toward the sink, employing some MAC and routing protocol. The actual energy consumption in each node due to the packet relaying operations depends on many factors, as the employed radio, the transmission power and the physical and MAC protocols. In this last case, duty-cycle MAC protocols are often used in wireless sensor networks to avoid idle listening, which play an import role in energy wasting [15]. Completing such complex scenario, synchronization messages may be transmitted among the nodes to optimize the sleeping time.

During the multihop communication, each packet may be acknowledged by a 1-hop ACK message. When a packet is successfully received and acknowledged by an intermediate node, it may decide to silently drop the incoming packet in order to save energy. Although the monitoring quality may be somehow negatively impacted when some packets do not reach the sink, such approach may turn the network active for a longer time.

We propose a threshold-based packet relaying mechanism where the current energy level of the relaying nodes (referred as *e*) indicates what type of data packets may flow over the network. As long as applications define a monitoring profile, each data packet will be dynamically assigned to a priority level according to the packet's payload. The type of the packets' payloads is identified employing a 2-bit Data Type (DT) field, which is expected to be inserted in every packet's header. Table I presents a typical configuration for the DT field of multimedia packets.

Depending on the intended level of differentiation of data packets, different values for DT can be used, being this field enlarged. For example, image snapshots, infrared and thermal images might represent different contents for the applications, requiring new values for DT.

TABLE I
Values for the DT.

Type of Data	DT
Scalar	0
Audio	1
Image	2
Video	3

Intermediate nodes that are implementing the proposed relaying approach will rely on the values of DT to assign a priority level to data packets. Each priority level will be directly mapped to an energy threshold, which will be considered when deciding if incoming packets must be forwarded to the next hop or silently dropped. Adopting energy thresholds create an adaptive behavior for the network, where the overall application quality decreases along the time but the network lifetime is enlarged.

Three different energy thresholds are defined: e_1, e_2 and e_3. When e is below one of the thresholds, only a subset of data packets must be relayed, while the remaining packets are silently dropped. While the DT of data packets depends exclusively on the packets' payloads, the monitoring profile will indicate what data contents are relevant for the application depending on the current energy level of intermediate nodes. Table II presents four different monitoring profiles, each one with a particular prioritization policy. However, users could create and employ any monitoring profile, just defining the type of data that must be relayed depending on the current energy level and thresholds. Note that for each monitoring policy, there is a type of data that will be always delivery to the sink, whatever is the current energy configuration of intermediate nodes.

TABLE II
Relaying strategy depending on the monitoring profile.

Energy level	Packets that must be relayed
Profile 1	
$e \geq e_1$	Scalar, audio, image and video
$e_2 \leq e < e_1$	Scalar, audio and image
$e_3 \leq e < e_2$	Scalar and audio
$e < e_3$	Scalar
Profile 2	
$e \geq e_1$	Audio, image, video and scalar
$e_2 \leq e < e_1$	Audio, image and video
$e_3 \leq e < e_2$	Audio and image
$e < e_3$	Audio
Profile 3	
$e \geq e_1$	Image, audio, video and scalar
$e_2 \leq e < e_1$	Image, audio and video
$e_3 \leq e < e_2$	Image and audio
$e < e_3$	Image
Profile 4	
$e \geq e_1$	Video, audio, image and scalar
$e_2 \leq e < e_1$	Video, audio and image
$e_3 \leq e < e_2$	Video and audio
$e < e_3$	Video

We consider that the initial energy level of the intermediate nodes is $e = 1$, and that they become inoperative when they run out of energy, $e = 0$. We can also state that $0 \leq e_3 \leq e_2 \leq e_1 \leq 1$. For $e > e_1$, all incoming packets will be forwarded to the next hop toward the sink. For the remaining possibilities of e and energy thresholds, packets will be relayed according to the considered monitoring profile. Considering that the application profile is already known, intermediate nodes should be preconfigured before deployment in the monitored field. However, a dynamical configuration mechanism might be employed to automatically configure the intermediate nodes, potentially adapting the network if the considered monitoring profile needs to be changed. As we are concerned with energy saving results when employing the proposed relaying approach, we assume that intermediate nodes are already configured when source nodes start transmission and that such configuration is steady.

Each intermediate node takes the decision of relaying or dropping packets based only on its residual energy, in an open-loop approach. Doing so, intermediate nodes do not need to know the energy status of neighbor nodes, reducing complexity and avoiding the transmission of feedback messages. For networks composed of braided-paths, the proposed relaying approach should be adopted by only hub nodes, which are more energy-critical. Therefore, employing the proposed optimization approach in only critical intermediate nodes turns open-loop processing a proper option.

As the monitoring quality will be directly associated with the monitoring profile, the proposed relaying approach is expected to have a reduced impact on the monitoring quality when compared with optimization algorithms based only on the relevancies of packets' payloads.

Based on several energy consumption models designed in the last few years [7] [9] [13], there is a basic notion that more packets to be transmitted (or received) consume more energy of the nodes, potentially impacting the network lifetime. In general words, more energy is expected to be consumed over the network when packets have to be relayed through more intermediate nodes. Thus, in theory, energy saving should be achieved when some packet relaying is avoided. However, energy consumption in wireless sensor networks depends mostly on the time that wireless radio is turned on, and thus idle listening and sleeping time have a major role when computing the expected energy consumption. Therefore, the proposed approach will be validated through realistic discrete event simulations, as presented in section IV.

IV. SIMULATION RESULTS

The proposed context-aware energy-efficient multimedia streaming approach can be employed to save energy in real-world wireless sensor networks, turning the network active during longer time periods. As different applications may have different monitoring profiles, the application quality is expected to be not severely harmed, since data that is most

relevant for applications is always preserved. Thus, the performed simulations are concerned with the energy efficiency of the proposed approach over a traditional transmission mechanism, where all data packets are always relayed to the next hop whatever is the current energy level of intermediate nodes.

We conducted a series of simulations where different source nodes are deployed, retrieving different type of monitoring data. The simulations were performed employing the framework Castalia [16], which was adapted to incorporate packet relaying based on the relevance of packets' payloads. Castalia is a C++ discrete event simulator based on the OMNet++ platform.

Source nodes will transmit data packets according to their expected sensing functionality. We assume a specific configuration for each type of data, as described in Table III, although other configurations are also possible. For data packets sizing 125 bytes, with an effective payload area of 110 bytes, we expect that every scalar source node will transmit a single data packet (informing e.g. temperature, pressure or humidity) every second. Audio and video source nodes will continuously stream data packets, while image sources will transmit a single compressed still image per second. In fact, this continuous monitoring approach is highly energy-consuming for real-world wireless multimedia sensor networks, but it is valuable to present the expected energy saving when employing the proposed relaying mechanism. Note, however, that energy-efficient packet relaying based on the relevance of packets' payloads can also be achieved for triggered-based monitoring applications, since intermediate nodes will employ the same prioritization and relaying approach.

TABLE III
Codec and transmission rate for each data type.

Media	Codec	Rate
Scalar	Raw data	1 kbps
Audio	G.723.1 speech or noise	6.3 kbps
Image	JPEG 10:1 128 x 128 grayscale	13.1 kbps
Video	H.263 QCIF 15fps	48 kbps

We initially intended to assess the energy consumption of individual source nodes, for a very simple linear multihop wireless sensor network composed of four nodes (source, sink and two intermediate nodes). The assessment of the energy consumption pattern for the transmissions of a single source node is presented in Fig 1, where data packets have to be relayed through two ordinary intermediate nodes (without packet prioritization) to reach the sink. As each of the considered data types expressed in Table III imposes a particular transmission rate, we can note substantial variations in the average energy consumption of the network for the performed tests.

Scalar data is usually represented by few bytes, requiring transmission of few data packets along the time. On the other hand, even compressed video streaming may consume much more energy, significantly reducing the network lifetime. It is expectable that wireless multimedia sensor networks will employ most of camera-enabled source nodes for image monitoring, which is a less stringent approach for visual monitoring. However, video streaming can be desirable for many applications, fostering the adoption of optimizations approaches as the proposed energy-efficient packet relaying mechanism.

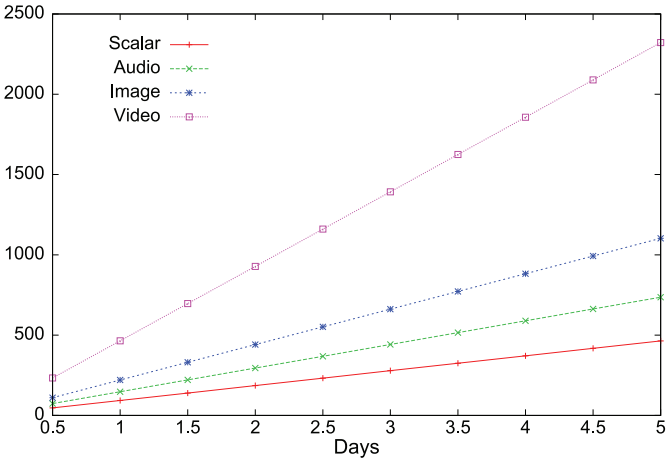


Fig 1. Average energy consumption (J) of the network.

In order to assess the performance of the proposed relaying mechanism, consider a typical heterogeneous communication scenario, as depicted in Fig 2. There are six source nodes transmitting different data contents to the sink, where each node is communicating using a duty-cycle MAC protocol [15]. The intermediate node n1 receive most data traffic and thus should relay data packets according to the packets' payloads in order to enlarge the expected network lifetime.

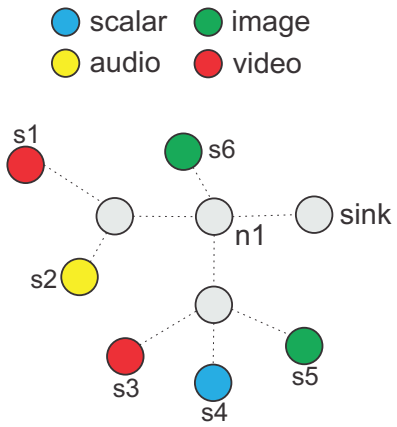


Fig 2. A heterogeneous communication scenario.

Assuming the maximum distance between neighbor nodes as 10 meters and that every node has a transmission power of -5dBm, we can assess the energy consumption of node n1 along the time. The four monitoring profiles described in Table II were considered, as well as a traditional relaying

mechanism where every incoming packet is always relayed to the next hop. The initial energy level of the nodes was established in 28000J.

Fig 2 presents the energy consumption of node n1, assuming transmission of monitoring data during 5 days. We also consider $e_1 = 0.9$, $e_2 = 0.6$ and $e_3 = 0.3$.

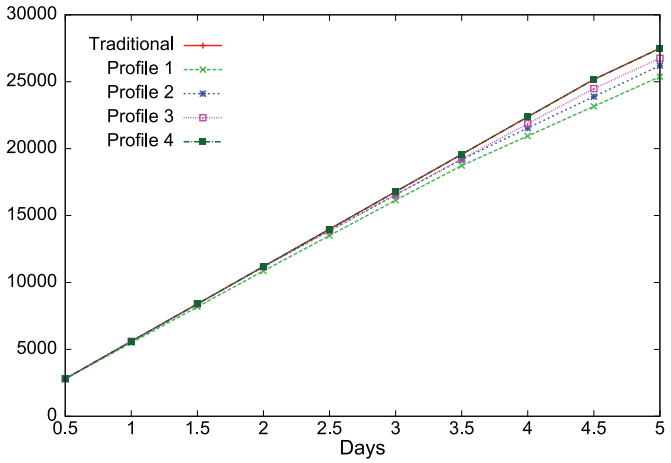


Fig 3. Energy consumption for the 1st configuration of energy thresholds.

As can be noted in Fig 3, Profile 4 and the traditional transmission approach consume almost the same energy. It is because video streams are the most significant data for the application with Profile 4, and video consumes most energy. On the other hand, transmission following Profile 1 consumes least energy since scalar data is prioritized over the other data types, and scalar data transmission consumes less energy.

The energy saving of the proposed context-aware relaying approach is highly dependent on the transmission configuration and the predefined energy thresholds. Fig 4 presents the energy consumption of node n1 for the thresholds $e_1 = 0.95$, $e_2 = 0.8$ and $e_3 = 0.65$.

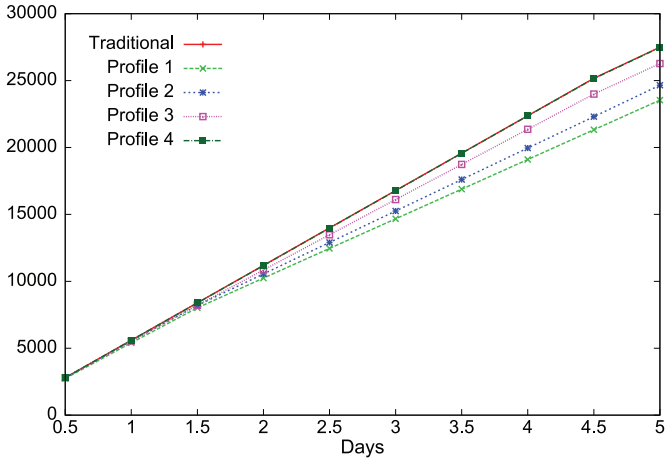


Fig 4. Energy consumption for the 2nd configuration of energy thresholds.

For a configuration of energy thresholds with higher values, packets will be silently dropped in intermediate nodes in early stages of the network lifetime. Doing so, energy saving will be achieved as less data packets will have to be relayed, potentially enlarging the network lifetime. Note that changing the energy thresholds only alter the total expected amount of energy saving, but do not impact the proportion of energy consumption of each monitoring profile.

In order to highlight the expected energy saving after 5 days of transmissions, we isolated the final energy consumption presenting this subset of results in the graphic of Fig 5. We also assumed the energy thresholds $e_1 = 0.95$, $e_2 = 0.8$ and $e_3 = 0.65$.

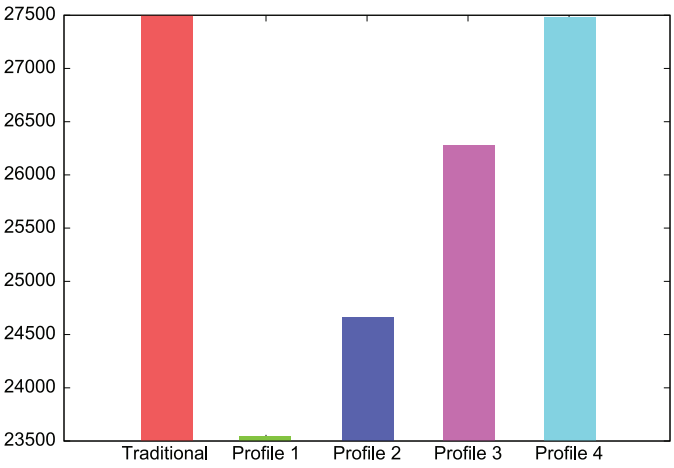


Fig 5. Energy consumption after 5 days of transmission.

Video streaming in wireless multimedia sensor networks is expected to consume most energy of the nodes. As an example of how stringent video streaming can be, we redefined the communications scenario on Fig 2 considering that three of the camera-enabled source nodes will transmit still images, instead of two. In other words, only one source node will stream video.

The energy consumption in node n1 was assessed again for 5 days of transmission, as depicted in Fig 6. We considered the same energy thresholds of the previous simulations, which were $e_1 = 0.95$, $e_2 = 0.8$ and $e_3 = 0.65$.

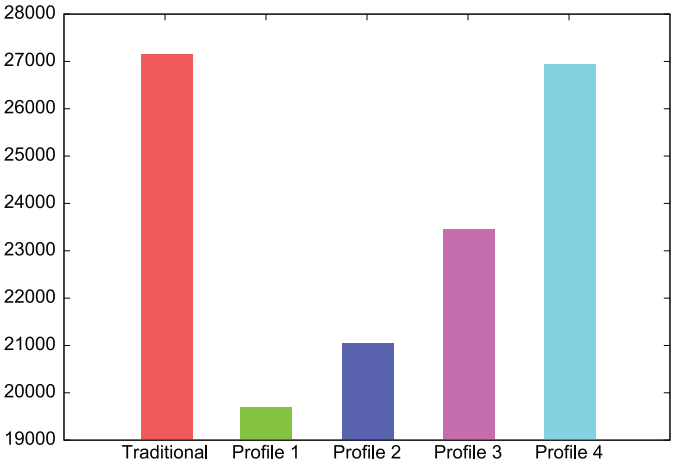


Fig 6. Energy consumption reducing the number of video streams.

The same energy consumption pattern can be identified in the graphics on Fig 5 and Fig 6. However, less energy is consumed for all configurations when there is only one source node streaming video, though differing in the absolute value of energy saving. Thus, although video streaming can be harmful for real-world wireless sensor network, transmission optimizations as proposed in this paper can be valuable to achieve energy savings while providing video content for an extended period of the network lifetime.

It is interesting to note that higher values of energy thresholds imply in energy savings that can be reflected in the increasing of the network lifetime. However, dropping low-relevant packets in early stages of the network may harm the overall monitoring quality of the application. This tradeoff between energy savings and monitoring quality require a clear understanding of the actual monitoring tasks of the network, the expected quality and intended network lifetime.

As a final comment, multimedia coding and compression tasks consume energy, memory and processing resources of the source nodes. In our simulations, however, we considered only the energy costs for packet transmission, since whatever the employed relaying approach, the energy consumption in source nodes will be the same.

V. CONCLUSIONS

We have proposed an adaptive energy-efficient multimedia streaming transmission where intermediate nodes relay data packets to the next hop based on their residual energy level and pre-configured energy thresholds. As the final monitoring quality depends on the context of the performed monitoring tasks, each application defines a monitoring profile directly indicating the relevance of each data type. The defined monitoring profiles will then be employed in critical intermediate nodes to define the relaying policy of data packets that must be adopted, saving energy when less relevant packets are silently dropped.

Some simulations were performed in order to assess the performance of the proposed approach in terms of energy consumption. The initial results indicated that the proposed adaptive multimedia streaming mechanism was suitable for optimizations in wireless multimedia sensor networks.

Future works will address simulations for different communication scenarios, relating network topologies with the resulting energy consumption. Furthermore, some numerical approach will be designed to assess the resulted monitoring quality when some packets do not reach the sink because of dropping in intermediate nodes. Doing so, a numerical resource will be available to verify the final monitoring quality depending on the considered energy thresholds, providing a valuable mechanism to adjust these parameters according to the expected QoS of monitoring applications in wireless sensor networks.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of research agencies CAPES (grant no. 9014-11-0) and FCT (project ref. PTDC/EEA-TEL/104185/2008), that partially funded this work.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. “Wireless sensor networks: a survey”. Computer Networks, vol. 38, pp. 393-422, 2002.

[2] J. Yick, B. Mukherjee and D. Ghosal. “Wireless sensor network survey”. Computer Networks, vol 52, pp. 2292-2330, 2008.

[3] I. Akyildiz, T. Melodia and K. Chowdhury, “A survey on wireless multimedia sensor networks”, Computer Networks, vol. 51, pp. 921-960, 2007.

[4] I. Almalkawi, M. Zapata, J. Al-Karaki and J. Morillo-Pozo, “Wireless multimedia sensor networks: current trends and future directions”, Sensors, vol. 10, pp. 6662-6717, 2010.

[5] J. Ai and A. Abouzeid, “Coverage by directional sensors in randomly deployed wireless sensors networks”, Journal of Combinatorial Optimization, vol. 11, pp. 21-41, 2006.

[6] D. Costa and L. Guedes, “A survey on transport protocol for wireless multimedia sensor networks”, KSII Transactions on Internet and Information Systems, vol 6, pp. 241-69, 2012.

[7] S. Qaisar and H. Radha, “Multipath multi-stream distributed reliable video delivery in wireless sensor networks”, in *Proc*. 43rd Annual Conference on Information Sciences and Systems, Baltimore, USA, March 2009.

[8] D. Ganesan, R. Govindan, S. Shenker and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks”, Mobile Computing and Communications Review, vol. 38, pp. 10-24, 2001.

[9] D. Costa and L. Guedes. “A survey on multimedia-based cross-layer optimization in visual sensor networks”. Sensors, vol. 11, pp. 5439-5468, 2011.

[10] L. Zhang, M. Hauswirth, L. Shu, Z. Zhou, V. Reynolds and G. Han, “Multi-priority multi-path selection for video streaming in wireless multimedia sensor networks”. Lectures Notes in Computer Science, vol. 5061, pp. 439-452, 2008.

[11] L. Shu, Z. Zhou, M. Hauswirth, D. Phuoc, P. Yu and L. Zhang, “Transmitting streaming data in wireless multimedia sensor networks with holes”. in *Proc*. 6th International Conference on Mobile and Ubiquitous Multimedia, Oulu, Finland, pp. 24-33, December 2007.

[12] L. Shu, Y. Zhang, Z. Yu, L. Yang, M. Hauswirth and N. Xiong, “Context-aware cross-layer optimized video streaming in wireless multimedia sensor networks”. Journal of Supercomputing, vol. 54, pp. 94-121, 2010.

[13] V. Lecuire, C. Duran-Faundez and N. Krommenacker, “Energy-efficient transmission of wavelet-based images in wireless sensor networks”. Journal of Image Video Processing, pp. 1-11, 2007.

[14] D. Costa, L. Guedes, F. Vasques and P. Portugal, “Energy-efficient packet relaying based on the sensing relevancies of source nodes in visual sensor networks”, in *Proc*. IEEE Workshop on Factory Communication Systems, Lemgo, Germany, 2012.

[15] T. van Dam and K. Langendoe. “An Adaptive energy-efficient MAC protocol for wireless sensor networks”, In *Proc*. ACM Conference on Embedded Networked Sensor Systems, Los Angeles, USA, pp. 1-10, 2003.

[16] Y. Tselishchev, A. Boulis and L. Libman, “Experiences and lessons from implementing a wireless sensor network MAC protocol in the Castalia simulator”. In *Proc.* IEEE Wireless Communications & Networking Conference, Sydney, Australia, pp. 1-6, 2010.

Estratégia centralizada energeticamente eficiente para RSSF heterogêneas utilizando Lógica Fuzzy

M. Christiano, M. Alexandre, C. Tassio, J. Jailton, T. Carlos

***Resumo* — As restrições de energia impostas pela capacidade limitada da bateria interna dos nós sensores estimulam o desenvolvimento de algoritmos energeticamente eficientes para aumentar o período de estabilidade e o tempo de vida útil das RSSF. Neste artigo, é proposto um controle centralizado na Estação Base para eleger *Cluster Heads* mais eficientes, considerando três tipos de nós sensores com diferentes níveis de energia. O controle centralizado utiliza o algoritmo *k-meas*, responsável pela divisão dos *clusters,* e Lógica *Fuzzy* pra a eleição dos *Cluster Heads*. As simulações indicam que um controle centralizado e a inserção de três níveis de heterogeneidade permitem aumentar o período de estabilidade e o tempo de vida útil em RSSF.**

.

***Palavras-chave* — RSSF heterogêneas, Cluster Head, Lógica Fuzzy, k-means.**

I. INTRODUÇÃO

Os avanços nas áreas de microprocessamento, sistemas micro eletromecânicos (MEMS – Micro Electro Mecanical System) e comunicação sem fio, estimulam o desenvolvimento de equipamentos sensores que podem ser utilizados em diversas áreas de aplicação.

Representando uma subclasse das redes *ad hoc* sem fio, as RSSF (Redes de Sensores Sem Fio) são consideradas como uma nova geração de sistemas embarcados de tempo real com recursos computacionais, energia e memória limitados. Estas redes podem apresentar grande número de nós sensores sem fio cooperando entre si para realizar tarefas substanciais, que simplificadamente, reportam a informação sobre um fenômeno monitorado para um ponto de coleta, denominado BS (*Base Station*).

Um dos principais entraves apresentados nestas redes está relacionado à restrição de energia devido à capacidade limitada das baterias internas dos nós sensores. Vários fatores são culminantes para o desgaste da bateria dos nós, sendo o modulo de rádio um dos principais consumidores de energia dos nós sensores no processo de transmissão de dados, [9].

O consumo de energia pode ser reduzido, admitindo que apenas alguns nós possam enviar dados para a BS. RSSF baseada em uma estrutura hierárquica organiza seus nós em agrupamentos (*clusters*) e elege um nó líder do grupo, denominado *cluster head* (CH). O CH é responsável por coletar todos os dados dos nós de seu *cluster*, informações provenientes de sensoriamento, podendo agrega-los e posteriormente os encaminha para a BS, [1].

A estrutura hierárquica pode ser formada por dois tipos de redes, homogêneas ou heterogêneas. Em redes homogêneas, os nós apresentam as mesmas características, como por exemplo, capacidade energética, processamento e rádio, [6]. Em estruturas heterogêneas, alguns nós sensores podem apresentam requisitos de hardware diferenciados, como melhor capacidade energética. Estes nós sensores, dão a rede um maior período de estabilidade [11, 10]. Trabalhos que consideram a heterogeneidade dos nós podem ser encontrados em [11, 8, 6, 2, 10, 12].

O algoritmo LEACH (*Low Energy Adaptative Clustering Hierarchy*) proposto por [5], é a solução mais popular encontrada na literatura para formação de *clusters* e serve como base para inúmeros trabalhos voltados para clusterização. Como no LEACH, o grande problema destes algoritmos é a utilização de informações locais com base em cálculos de probabilidade para eleição dos lideres dos *clusters*. Esse tipo de seleção pode gerar CHs muito próximos da borda da rede aumentando a dissipação de energia devido à distância de transmissão dos nós para o CH. Outro grande problema com a forma de eleição utilizada pelo algoritmo LEACH, é a falta de tratamento discriminatório sobre as discrepâncias energéticas dos nós que formam a rede, uma vez que os CHs selecionados devem possuir recursos energéticos suficientes para suportar as cargas de transmissão dos nós associados a ele.

Considerando os entraves relacionados à eleição de CHs com base em informações locais, este artigo propõe uma estratégia para eleição do CH ideal em RSSF heterogêneas utilizando Lógica *Fuzzy* com base em informações

M. Christiano. Author is with the Federal University of Pará (UFPA) Institute of Electrical and Computer Engineering, CO 66.075-110 Belém, Brazil. phone: 55-91-32017634; fax: ; e-mail: christiano@ufpa.br.

M. S. Alexandre, Author, is with the Department of Computer of Science, Federal University of Pará, Belém, Brazil, CO 66.075-110 Brazil. Fax: 55 – 91- 3201-7405; e-mail: amelo@ufpa.br.

C. Tassio. Author is with the Federal University of Pará (UFPA) Institute of Electrical and Computer Engineering, CO 66.075-110 Belém, Brazil. phone: 55-91-32017634; fax: ; e-mail: tassioccarvalho@yahoo.com.br

J. Jailton. Author is with the Federal University of Pará (UFPA) Institute of Electrical and Computer Engineering, CO 66.075-110 Belém, Brazil. phone: 55-91-32017634; fax: ; e-mail: josejailtonjunior@yahoo.com.br

T. Carlos. Author is with the Federal University of Pará (UFPA) Institute of Electrical and Computer Engineering, CO 66.075-110 Belém, Brazil. phone: 55-91-32017634; fax: ; e-mail: cartav@ufpa.br

centralizadas na BS. Este controle centralizado define o CH com base em informações adquiridas no momento de formação da rede. As informações coletadas são utilizadas para carregar o algoritmo *k-means*, responsável pela divisão dos *clusters*, como também, o sistema *fuzzy*, que se encarrega de selecionar o líder de cada grupo formado pelo algoritmo *k-means*. Os critérios adotados para de seleção de CHs são: nível de energia, centralidade e proximidade para a BS. A inserção de três tipos de nós com recursos energéticos diferenciados e a utilização de informações centralizadas na BS, permitem eleger *cluster heads* bem posicionados e com níveis adequados de energia para suportar a carga de transmissão de seu cluster, aumentando o período de estabilidade e vida útil da rede.

II. Trabalhos Correlatos

O Algoritmo LEACH (*Low Energy Adaptative Clustering Hierarchy*), proposto por [5], elege o CH com base em informações locais a cada novo ciclo (*round*). Para que o nó seja eleito, o numero escolhido deve ser menor que o limiar *T*. Após a eleição, o novo CH envia mensagens de anuncio para todos os nodos da rede. Os demais nós associados decidem a que CH devem se conectar.

Embora o LEACH apresente uma estrutura hierárquica que permite reduzir o consumo de energia em RSSF, o mesmo não adota nenhum critério de posicionamento no momento de eleição do CH, podendo selecionar CH próximos à borda da rede. Outro problema do algoritmo é descrito no trabalho apresentado em [12] SEP, comprovando que o LEACH não é eficiente em estruturas heterogêneas, por não considerar a discrepância de energia dos nós que formam a rede.

Semelhante ao LEACH, o algoritmo DEEC, proposto por [10] utiliza informações locais para eleição do CH. Entretanto, o algoritmo proposto pelos autores é capaz de tratar a heterogeneidade da rede. Os CHs são selecionados utilizando uma probabilidade com base na energia residual dos nós e a energia média da rede. A heterogeneidade inserida pelos autores de DEEC diz respeito apenas à capacidade energética diferenciada de um conjunto de nós de que formam a rede. Como a maioria dos algoritmos baseados no algoritmo LEACH, DEEC utiliza de informações locais para seleção dos CHs. As imagens apresentadas pelos autores ilustram claramente a eleição do lideres próximos da borda da rede.

Baseado no algoritmo DEEC, os algoritmos E-DEEC proposto por [11] e LEACH-HPR proposto por [6] também considera a energia residual dos nós no processo de eleição para o CH da rede. O diferencial dos algoritmos está na inserção de três tipos de nós com diferentes níveis de energia: nós normais, nós avançados e super nós. Os algoritmos propostos pelos autores permitem prolongar o período totalmente funcional da rede com a adição dos super nós. Tal como em DEEC, os trabalhos apresentados pelos autores utilizam informações locais para eleição do CH. Para minimizar a dissipação de energia na fase de comunicação entre os CHs mais afastados e a BS, os autores de LEACH-HPR propõem um algoritmo de múltiplos saltos entre os CHs eleitos, semelhante ao ACHTLEACH proposto por [3].

Em [13] os autores propõem um algoritmo que considera uma estratégia hibrida móvel com o objetivo de distribuir o consumo de energia por toda a rede com o CH movimentando-se para um local de maior concentração de energia quando ocorrer um evento e com possibilidade de controlar sua potência de transmissão. Entretanto, este deslocamento pode maximizar o consumo de energia gasto no envio de dados para a BS. O algoritmo BS-CH *hybrid mobile strategy* permite a movimentação da estação base para minimizar a distância para o CH e o consumo na transmissão de dados, desconsiderando a energia consumida e supondo que a estação base é móvel, sem informação se a fonte de alimentação é continua.

III. Lógica Fuzzy e Algoritmo k-means

Proposto por [14] a Lógica *Fuzzy* utiliza métodos com o objetivo de controlar a linguagem vaga e a imprecisão utilizada diretamente pelo homem, por meio de um conjunto de valores representados por variáveis linguísticas. Cada conjunto de valores tem um intervalo diretamente associado a regras semânticas. Ao contrario da Lógica booleana, na lógica *fuzzy*, a avaliação de uma determinada preposição pode compreender valores, graus de pertinência, que variam no intervalo de [0,1].

O algoritmo *k-means* é uma técnica de agrupamento de dados por k-médias muito popular por sua facilidade de implementação. Normalmente os algoritmos de clusterização são amplamente utilizados em aplicações que necessitem gerar padrões, dividindo os objetos em grupos úteis ou significativos, [7]. Para a proposta apresentada neste trabalho o algoritmo k-means é carregado com as coordenadas de todos os nós que foram à rede. Desta forma o algoritmo gera um padrão, dividindo os clusters, estes, formados pelos nós mais próximos.

IV. Modelagem

Para a solução proposta neste trabalho, a escolha do *cluster head* se da a cada *round*. Um *round* termina no final do processo de agregação e envio de dados para a BS. Basicamente o processo é dividido em três etapas: (i) A primeira etapa consiste no inicio do processo de formação dos *clusters*. A divisão dos *clusters* é feita utilizando o algoritmo *k-means*; (ii) A segunda etapa consiste na seleção do CH para cada *cluster* formado pelo *k-means*. Os dados dos de cada *cluster* são carregados no sistema *fuzzy*. Após o resultado de seleção do CH, a estação base envia mensagens em *broadcast* para os nós da rede informando o ID do líder do grupo para que os nós possam enviar dados para seu respectivo CH; (iii) A terceira etapa concerne no processo de agregação dos dados pelo CH. Este processo consiste em comprimir os dados e envia-los a BS.

A. Modelo de dissipação de energia

O modelo de dissipação de energia adotado é semelhante ao modelo utilizado pelo LEACH. O modelo consiste na energia dissipada na transmissão e recepção de *k-bit* de mensagem em uma distância ***d,*** o radio consome:

$$E_{Tx}(k,d) = E_{elec} * k + \varepsilon_{fs} * k * d^2 \quad para\ d < d_0$$

$$E_{Tx}(k,d) = E_{elec} * k + \varepsilon_{amp} * k * d^4 \quad para\ d \geq d_0 \qquad (1)$$

$$E_{Rx}(k,d) = E_{elec} * k \qquad (2)$$

A energia dissipada na transmissão e recepção do radio é representada por $E_{elec} = 50\,nJ/bit$. Dois modelos são utilizados para que o amplificador de transmissão alcance um nível aceitável, dependendo da distância entre o transmissor e o receptor. Se esta distância não ultrapassar o limiar d_0 a energia dissipada pelo amplificador de transmissão é dada por $E_{fs} = 10\ pJ/bit/m^2$, caso o limiar seja ultrapassado o modelo utilizado $E_{amp} = 0.0013\,pJ/bit/m^4$. Onde $d_0 = \sqrt{E_{fs}/E_{amp}}$, [11, 5].

B. Critérios para eleição do CH

Os critérios utilizados para carregar o sistema *fuzzy* são: (i) Energia – O nível de energia de cada nó da rede é representado pela variável linguística Bateria e possui os valores linguísticos Baixa, Moderada e Alta. Consideramos que a rede possui características heterogêneas, no que diz respeito à energia dos nós. Logo, os super nós e nós avançados representam maior probabilidade para eleição do CH, Fig. 1.

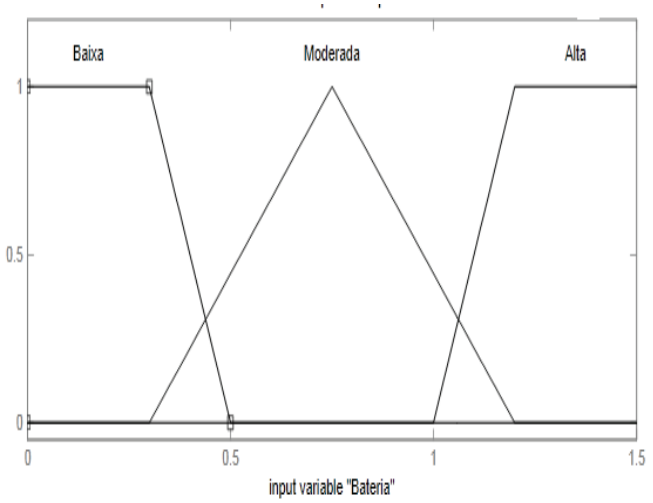


Fig. 1. Variável bateria com os valores linguísticos correspondentes aos níveis de energia dos três tipos de nós sensores disponíveis na rede.

(ii) Centralidade – A variável centralidade diz respeito ao posicionamento do nó em relação ao centro do *cluster*. Os valores linguísticos que representam a variável centralidade são: Perto, Moderado e Longe.

Quanto menor o valor de centralidade mais próximo o nó está do centro do cluster, Fig. 2.

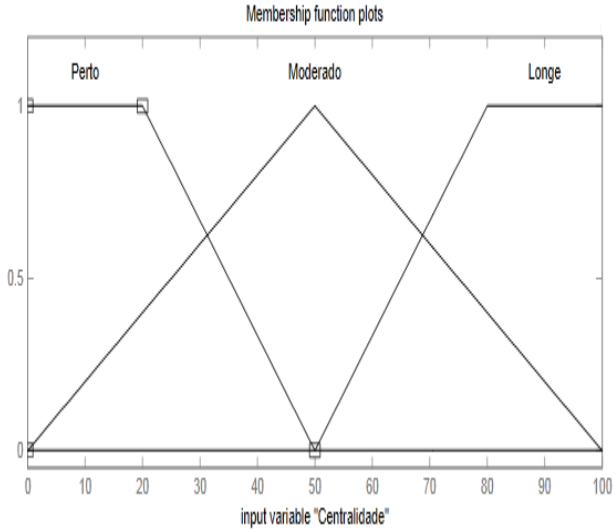


Fig. 2. Variável linguística centralidade.

(iii) Distância para BS – A variável é representada no sistem como DistBS. Semelhante a variável centralidade, utiliza c valores linguísticos: Perto, Moderado e Longe. O critério d distância para a BS é utilizado para gerar uma aproximação d CH com a BS, objetivando minimizar o consumo de energi do CH na fase de transmissão de dados, Fig. 3

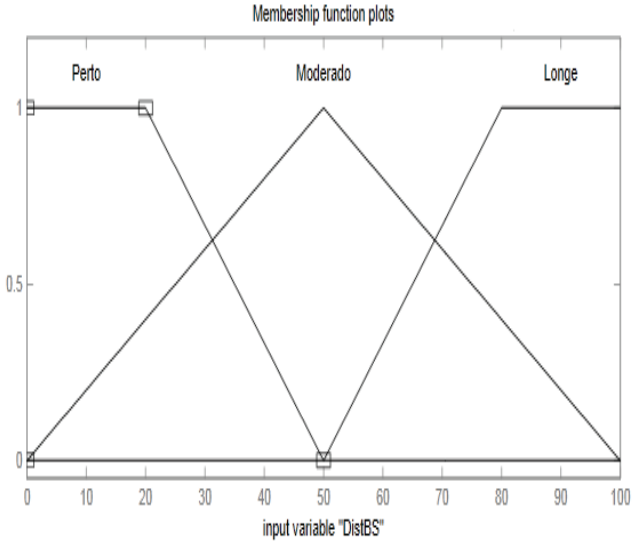


Fig. 3. Variável linguística Distância para BS.

As variáveis descritas acima correspondem ac antecedentes do sistema *fuzzy* e dão entrada ao processo d fuzzyficação. Cada valor de entrada é mapeado para ur conjunto *fuzzy* de entrada com seu determinado grau d pertinência.

Os consequentes do sistema, conjuntos *fuzzy* de saída d sistema que vão determinar o CH ideal em um *cluster x*, usar os seguintes valores linguísticos: *muito fraco*, *fraco*, *médi forte e muito forte*, Fig. 4.

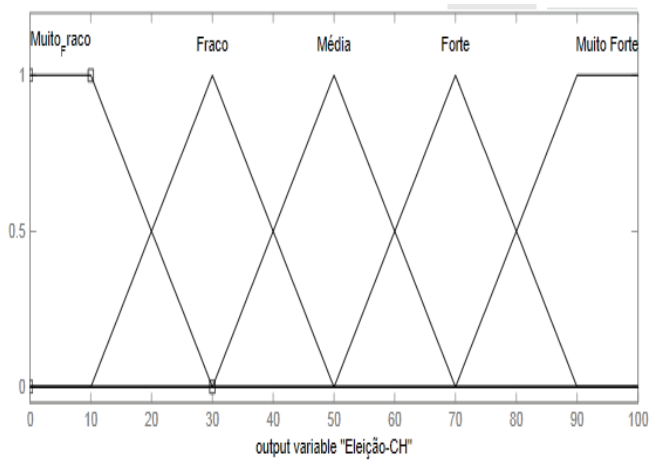


Fig. 4. Saída do sistema *fuzzy*, consequentes do sistema.

No processo de mapeamento dos conjuntos *fuzzy* de saída para valores *crisp* (defuzzificação), cada nó sensor apresenta seu respectivo valor de saída do sistema *fuzzy*. Os maiores valores de saída indicam os *cluster heads* selecionados para o *round* atual.

A formação da Base de Regras do sistema é compreendida por vinte e sete regras 3^3=27, Tabela I. A melhor condição para a eleição do *cluster head* é dada pela seguinte regra: Se *Centralidade é* PERTO e *Bateria é* ALTA e *DistBS é* PERTO então *Eleição cluster head* ou *Saída é* MUITO FORTE.

C. Centralidade dos nós e distância para BS

Assumimos que a BS detém o conhecimento de nível de energia e posicionamento dos nós. Estas informações são enviadas no início de formação da rede. Os nós dissipam energia neste processo. Para determinar os valores de centralidade, a BS seleciona cada nó e utiliza o cálculo da distância euclidiana destes nós para o centro dos seus respectivos *clusters*, definidos pelo algoritmo *k-means*, Eq. (3). O nó que apresentar maior centralidade, como CH, permitirá que a dissipação de energia na comunicação seja de forma mais homogênea em relação aos demais nós associados.

$$d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

TABELA I
BASE DE REGRAS DO SISTEMA FUZZY

Regras	Entrada	Entrada	Entrada	Saída
1.	Centralidade é Perto	Bateria é Baixa	DistBS é Longe	Eleição-Muito Fraco
2.	Centralidade é Perto	Bateria é Moderada	DistBS é Moderado	Eleição- Forte
3.	Centralidade é Perto	Bateria é Alta	DistBS é Perto	Eleição- Muito Forte
4.	Centralidade é Moderado	Bateria é Baixa	DistBS é Longe	Eleição-Muito Fraco
5.	Centralidade é Moderado	Bateria é Moderada	DistBS é Moderado	Eleição- Forte
6.	Centralidade é Moderado	Bateria é Alta	DistBS é Perto	Eleição-Forte
7.	Centralidade é Longe	Bateria é Baixa	DistBS é Longe	Eleição-Muito Fraco
8.	Centralidade é Longe	Bateria é Moderada	DistBS é Moderado	Eleição-Fraco
9.	Centralidade é Longe	Bateria é Alta	DistBS é Perto	Eleição-Média
10.	Centralidade é Perto	Bateria é Alta	DistBS é Longe	Eleição-Forte
11.	Centralidade é Perto	Bateria é Moderada	DistBS é Longe	Eleição-Média
12.	Centralidade é Perto	Bateria é Moderada	DistBS é Perto	Eleição-Forte
13.	Centralidade é Perto	Bateria é Baixa	DistBS é Perto	Eleição-Muito Fraco
14.	Centralidade é Perto	Bateria é Alta	DistBS é Moderado	Eleição-Muito Forte
15.	Centralidade é Moderado	Bateria é Baixa	DistBS é Moderado	Eleição-Fraco
16.	Centralidade é Moderado	Bateria é Alta	DistBS é Moderado	Eleição-Forte
17.	Centralidade é Moderado	Bateria é Baixa	DistBS é Perto	Eleição-Muito Fraco
18.	Centralidade é Moderado	Bateria é Moderada	DistBS é Perto	Eleição-Controlador é Média
19.	Centralidade é Moderado	Bateria é Moderada	DistBS é Longe	Eleição- Fraco
20.	Centralidade é Longe	Bateria é Baixa	DistBS é Perto	Eleição-Muito Fraco
21.	Centralidade é Longe	Bateria é Moderada	DistBS é Perto	Eleição-Muito Fraco
22.	Centralidade é Longe	Bateria é Baixa	DistBS é Moderado	Eleição-Muito Fraco
23.	Centralidade é Longe	Bateria é Alta	DistBS é Moderado	Eleição-Fraco
24.	Centralidade é Longe	Bateria é Moderada	DistBS é Longe	Eleição-Muito Fraco
25.	Centralidade é Longe	Bateria é Alta	DistBS é Longe	Eleição-Fraco
26.	Centralidade é Perto	Bateria é Baixa	DistBS é Moderado	Eleição-Muito_Fraco
27.	Centralidade é Moderado	Bateria é Alta	DistBS é Longe	Eleição-Média

O mesmo processo ocorre no calculo da distância de cada n para a BS. Entretanto, vale resaltar que, este critério é muit importante quando a BS não está localizada demasiadament longe do cluster em questão. Logo, a disposição da rede dev ser considerada para que o critério distância para BS sej válido. Com isso, definimos regras seguras para não gerar CH próximo da borda da rede, já que a centralidade é mai importante.

D. Modelo de Sistema Fuzzy

Para o modelo de Lógica *Fuzzy*, foi utilizado *Fuzzificador Singleton*, máquina de inferência *Mamdani defuzzificador* centro de gravidade. Para cada entrad (x_1, x_2, x_3), a Saída do sistema é calculada, como mostra Eq. (4).

$$y(x_1, x_2, x_3) = \frac{\sum_{l=1}^{27} \mu_{F_l^1}(x_1)\mu_{F_l^2}(x_2)\mu_{F_l^3}(x_3)c_{avg}^l}{\sum_{l=1}^{27} \mu_{F_l^1}(x_1)\mu_{F_l^2}(x_2)\mu_{F_l^3}(x_3)} \quad (4)$$

E. Modelagem da rede

Para o modelo de rede, assumimos que *N* sensores estão distribuídos em uma área N x M. Três tipos de nós sensores, sensores normais, sensores avançados e super sensores, apresentando diferentes níveis de energia inicial, representam a heterogeneidade da rede. O cálculo que determina a quantidade de sensores normais, sensores avançado e super sensores na rede é semelhante ao utilizado por E-DEEC:

$$N.(1 - mf) \quad (5)$$

$$N.mf(1 - mp) \quad (6)$$

$$N.mf.mp \quad (7)$$

Onde mf é a fração do numero total de N nós sensores e mp a percentagem para o numero total de nós sensores que apresentam *e* mais energia que o nó sensor normal na rede. O cálculo de energia inicial total da rede adotado é o mesmo apresentado em [11].

$$E_{total} = N.(1 - mf).E_o + N.mf(1 - mp).(2).E_o + N.mf.mp.E_o(1 + e) = N.E_o(1 + mf(2 + mp.e) \quad (8)$$

Para a proposta apresentando três níveis de heterogeneidade, a energia total da rede é acrescida, considerando a maior capacidade energética dos sensores avançados e super sensores. Esta diferença é dada pelo fator $1 + mf(2 + mp.e)$.

F. Propriedades da rede

No modelo de rede descrito anteriormente, apresentamos as características dos nós que formam a rede, quanto ao seu nível de energia. Para o cenário de rede proposto, assumimos algumas propriedades: (i) No que concerne à distribuição de N nós da rede, esta é feita de forma aleatória, e não possuem nenhuma mobilidade; (ii) Os nós enviam sua localização para a BS utilizando GPS; (iii) Os nós dissipam energia para o envio de informação; (iv) Todos os nós detêm a mesma capacidade de transmissão e processamento, a heterogeneidade é aplicada a níveis de energia, já que alguns nós possuem recurso energético aumentado, o que difere dos sensores normais. (v) A BS é fixa e sua localização é pré-definida no algoritmo; (vi) Os nós sempre tem dados para transmitir para o CH.

VI. SIMULAÇÃO E RESULTADOS

A simulação é dividida em *rounds*, a cada *round* obtém-se um valor de saída com base nós parâmetros de entrada do sistema e um novo CH é eleito para cada *k cluster*. Estes valores são atualizados para entrada do *round* seguinte. Cada *round* executa as fases descritas na seção 4.

Cada nó é distribuído aleatoriamente em uma área de 100 m^2, onde á classificação do numero de nós normais, avançados e super com seus respectivos níveis de energia, é calculada utilizando as Eq. 5 e Eq. 6 e 7, sendo $e = 1$, $mf = 1$ e $mp = 0.6$. O nível de energia inicial para os nós normais é de 0.5J e para os nós avançados e de 1.0J e para super nós 1.5J. A BS é previamente definida com as coordenadas x= 50 e y=50 O modelo de rádio que foi utilizado segue a descrição em (1) (2).

100 nós estão distribuídos, divididos em *k clusters*. Cada n envia 4000 Bits de mensagem por *round* para o *cluster hea* da rede. A taxa de compressão dos dados é de 5%. A simulação e gerada em 5000 rounds.

Neste cenário é aplicada a métrica FND (*First Node Dies* para determinar o período de estabilidade da rede.

Na primeira fase da simulação é obtida a coordenada e níve de energia de cada nó que compõe a rede. O consumo d energia dissipada no envio das informações de cada nó calculado utilizando a Eq. (1). A energia inicial de cada nó decrementada neste processo. Após o envio das coordenadas o algoritmo *k-means*, implementado na BS, estipula u padrão com base na coordenada dos nós, calculando posicionamento de cada nó e divide os nós que formam a red em *k clusters*. O algoritmo também calcula o centro de cad cluster, informação utilizada posteriormente para o process de eleição do CH, descrito na subseção *C*. O número d *clusters* utilizados para simulação é *k=5*.

Cada nó, com seu respectivo valor de centralidad proximidade para BS e nível de energia, terá um consequent Um valor com grau de pertinência y, determinando a chanc deste nó se tornar *cluster head*. No processo d *defuzzyficação*, o nó que apresentar maior valor de saída, ser eleito como CH ideal no *round* atual.

A Fig. 5, exibe a divisão dos *clusters*, em uma área de 10 m , e o final da eleição de CHs no *round* 0, sendo representad por '◊' o *cluster head* eleito pelo sistema, 'O' o centro do cluster e □ representa a BS.

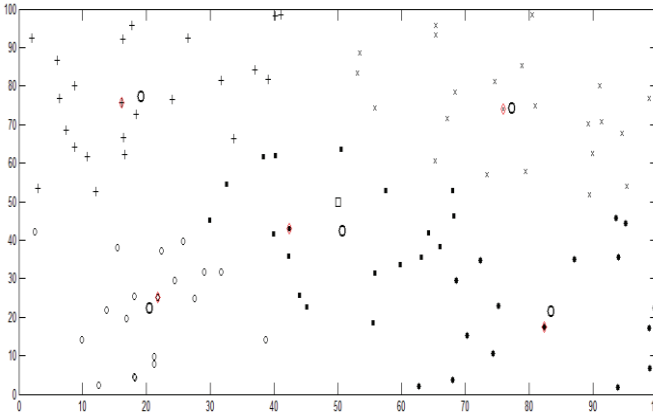


Fig. 5. CHs selecionados pelo sistema *fuzzy* no *round* 0.

Comparamos nossa proposta com os algoritmos LEACH DEEC. Para avaliação de desempenho utilizamos o final d período de estabilidade da rede e o tempo de vida útil. A escolha dos algoritmos para comparação se dá principalment pela utilização de informações locais para eleição dos CHs Além do método de escolha do líder, o algoritmo LEACH, nã trata as discrepâncias de energia dos nós que compõem a red Diferente de LEACH, o algoritmo DEEC, considera heterogeneidade dos nós para eleição do CH. Entretant utiliza informações locais para eleição do líder e apenas do

níveis de heterogeneidade enquanto nossa proposta utiliza três níveis.

A Fig. 6 exibe a quantidade de nós sensores ativos no tempo de vida útil da rede. Esta medida reflete o número total de nós que ainda não esgotaram toda sua energia em um determinado período de simulação.

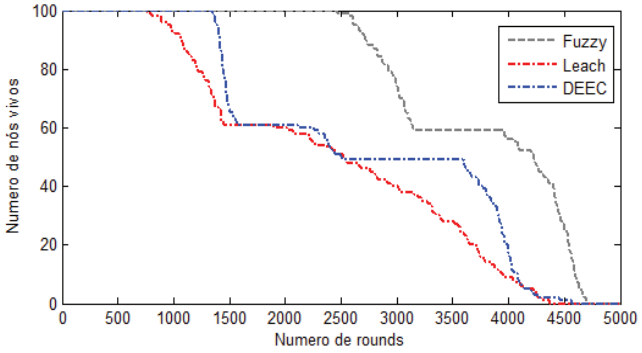


Fig. 6. Numero de sensores ativos no período de simulação

Os resultados indicam claramente que a inserção de novos níveis de heterogeneidade, a abordagem da lógica *fuzzy* como ferramenta de seleção e informações centralizadas na BS, permitem eleger *cluster heads* mais eficientes, aumentando o período de estabilidade e o tempo de vida da rede.

A proposta resulta em melhores resultados, quando comparado com os algoritmos LEACH e DEEC. O algoritmo LEACH apresentou o menor período de estabilidade, com a FND ocorrendo no *round* 765. O algoritmo DEEC apresentou melhor desempenho sobre o LEACH, com o período de instabilidade da rede iniciado por volta 1350 *rounds*. Nossa proposta, como observado na Fig. 6, mostra um melhor desempenho sobre os algoritmos comparados, aumentando o período de estabilidade da rede até aproximadamente 2470 *rounds*, quando ocorre a primeira inatividade de um nó por falta de energia.

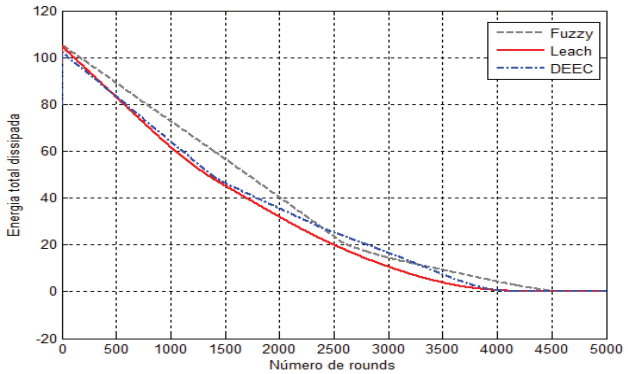


Fig. 7. Energia total residual de LEACH, DEEC e FUZZY.

A Fig. 7 exibe a energia dissipada pela rede ao longo dos 5000 rounds de simulação para cada algoritmo comparado. O total de energia para cada rede é de 104.5J. A dissipação de energia apresenta um declínio linear ao longo de 2500 *rounds*, para nossa proposta e para o algoritmo DEEC por volta de 1500 *rounds* , mudando a partir do momento em que o primeiro nó na rede fica inativo, quebrando o período de estabilidade. Ambos os algoritmos, *Fuzzy* e DEEC, permitem tratar as discrepâncias energéticas de cada nó na rede para eleição do CH, enquanto o algoritmo LEACH apresenta maior dissipação de energia por *round*, consequência dos problemas descritos anteriormente sobre a forma de eleição de CH pelo algoritmo e a falta de tratamento discriminatório das discrepâncias energéticas dos nós que formam a rede.

VII. Conclusões

Os resultados indicam que a proposta apresentada em nosso trabalho, oferece grandes vantagens, permitindo selecionar os nós mais adequados para lideres do grupo a cada *round*, com base nos valores de defuzificação do sistema *fuzzy*. A inserção de três níveis de heterogeneidade correspondente aos sensores normais, avançados e super sensores contribui consideravelmente para o aumento do período de estabilidade da rede, logo que, com esta inserção, a rede passa a possuir maiores recursos energéticos. Entretanto, os resultados indicam que, se esta discrepância não for considerada no momento de seleção do CH, não influencia de forma considerável no aumento do período de estabilidade.

Outra grande vantagem que contribui para os resultados obtidos neste trabalho é a utilização de um cotrole central na BS. Por não possuir severas limitações de energia, processamento e armazenamento, como os nós que formam a rede, a BS apresenta vantagens sobre o processamento local de informações em cada nó, processo este, encontrado nos algoritmos tradicionais para eleição do CH. O envio de atualizações das informações de nível de energia dissipada dos nós em cada *round*. Entretanto, mesmo com esta atualização, a proposta ainda apresenta melhorias sobre os outros modelos apresentados. Outra vantagem do controle central na BS concerne no momento de seleção do CH, por ter o papel de informar a rede sobre os lideres selecionados para cada *cluster*, previamente dividido pelo algoritmo *k-means*.

Este processo difere-se do processo encontrado nos algoritmos que utilizam informações locais para seleção de seu líder, cabendo ao próprio CH eleito enviar mensagens em *broadcast* para a rede, gerando dissipação de energia no momento da propagação. Finalmente, o trabalho apresentado tem a principal contribuição na eleição do CH mais eficientes, considerando sua localização e discrepâncias de níveis de energia, como também, na inclusão de novos níveis de heterogeneidade, permitindo, desta forma, aumentar o período de estabilidade da rede, ou seja, o período que a rede é totalmente funcional, aumentando consideravelmente o tempo de vida útil em redes de sensores sem fio heterogêneas.

Referencias

[1] Akyildiz, I.F.; Weilian Su; Sankarasubramaniam, Y.; Cayirci, E.; , "A survey on sensor networks," *Communications Magazine, IEEE* , vol.40, no.8, pp. 102- 114, Aug 2002.

[2] Badi, A., Mahgoub, I., Slavik, M., Ilyas, M. , "Investigation of the effects of network density on the optimal number of clusters in hierarchical Wireless Sensor Networks (WSNs)," High-Capacity Optical Networks and Enabling Technologies (HONET), 2010 , vol., no., pp.171-177, 19-21 Dec. 2010.

[3] Guo, L., Xie, Y., Yang, C., Jing, Z., "Improvement on LEACH by combining Adaptive Cluster Head Election and Two-hop transmission," *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on* , vol.4, no., pp.1678-1683, 11-14 July 2010.

[4] Gupta I., Riordan, D.; Srinivas S., "Cluster-head election using fuzzy logic for wireless sensor networks," *Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual* , vol., no., pp. 255- 260, 16-18 May 2005.

[5] Heinzelman, W., Chandrakasan A., and H. Balakrishnan, **"**Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd Annual Hawaii International Conference on System Sciences(HICSS)*, Maui, HI, Jan. 2000, pp. 3005 – 3014.

[6] Han, L., "LEACH-HPR: An energy efficient routing algorithm for Heterogeneous WSN," Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on , vol.2, no., pp.507-511, 29-31 Oct. 2010.

[7] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. ACM Comput. Surv., 31(3):264–323.

[8] Mubarak, T.M., Sattar, S. A., Rao, G. A., Sajitha, M., "Intrusion detection: An energy efficient approach in heterogeneous WSN," Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on , vol., no., pp.1092-1096, 23-24 March 2011.

[9] Pottie, G. J., Kaiser, W. J., "Wireless integrated network sensors (WINS)". Communications of the ACM. 43, 5 (maio de 2000), 51-58.

[10] Quing, L., Zhu, Q., Wang, M., "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks". ELSEVIER, Computer Communications 29, 2006, pp 2230- 2237.

[11] Saini, P., Sharma, A. K., "E-DEEC- Enhanced Distributed Energy Efficient Clustering scheme for heterogeneous WSN," Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on , vol., no., pp.205-210, 28-30 Oct. 2010.

[12] Smaragdakis, G., I. M., Bestavros A., "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks", in: Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004), 2004.

[13] Yan, B., Wu, X., Zhou, X. "A Improved Base Station Cooperative Mobile Strategy for WSN with Finite Powered Cluster Heads" Wireless Communications Networking and Mobile Computing (WiCOM), 6th International Conference, vol., no., p.1-4, 23-25 set. 2010.

[14] Zadeh, L. A. "Fuzzy Sets. Information and Control", 1965.