

Bridge para Interligação de uma Rede *Ethernet* e de uma Rede Sem Fios

Alberto Teixeira Queirós, Manuel Gomes de Oliveira
Rui Tomaz Valadas, A. Manuel de Oliveira Duarte

Resumo- Este artigo descreve o projecto de fim-de-curso designado por "Bridge para interligação de uma Rede *Ethernet* e de uma Rede sem Fios".

O trabalho decorreu em duas fases. Na primeira fase desenvolveu-se o *software* da *bridge* tendo como suporte um PC. Este *software* foi testado e optimizado utilizando uma *package* comercial para análise de tráfego em redes. Na segunda fase projectou-se, desenvolveu-se e testou-se o *hardware* da *bridge*.

Abstract- This paper describes the work carried out as part of the final year project designated by "Bridge for the interconnection of Ethernet and Wireless Local Area Networks".

The work was developed in two distinct phases. In the first phase the software of the Bridge was developed assuming a PC based platform. This software was tested and optimised using a commercially available software package for LAN traffic analysis. In the second phase the hardware of the Bridge was designed, developed and tested.

I. INTRODUÇÃO*

O trabalho de investigação no qual se inseriu este projecto tem como objectivo o desenvolvimento de uma *bridge* para interligar uma rede *Ethernet* a uma rede sem fios a 1Mbps compatível com o *standard* IEEE802.3 [2], como representado na Fig.1.

No sentido de facilitar as fases de desenvolvimento e teste optou-se por desenvolver inicialmente uma *bridge* entre duas redes por cabo *Ethernet*. O trabalho efectuado neste ano lectivo dividiu-se em duas partes. Na primeira implementou-se um sistema tendo como suporte um PC IBM compatível e duas placas de rede, de modo a interligar duas redes *Ethernet* a 10 Mbps. Este sistema permitiu projectar, desenvolver e testar o algoritmo de funcionamento da *bridge* e o respectivo *software*. Na segunda parte do projecto desenvolveu-se e testou-se um sistema *stand-alone* em *hardware* que permitirá a implementação de todas as funções da *bridge* desenvolvida na primeira parte, para além de outras, nomeadamente de gestão de redes.

O trabalho aqui apresentado decorreu no âmbito do projecto Europeu ESPRIT.6892 POWER ("Portable Workstation for Education in Europe").

Na secção II descreve-se a *bridge* desenvolvida bem como os seus princípios de funcionamento. Na secção III apresenta-se a *bridge* desenvolvida por *software*. Na secção IV apresenta-se o sistema *stand-alone*

desenvolvido em *hardware*. Finalmente na secção V apresentam-se as conclusões do projecto.

II. DESCRIÇÃO GERAL DO FUNCIONAMENTO DA BRIDGE

Uma *bridge* é um sistema que permite interligar LANs (Local Area Networks). Opera no nível 2 do modelo OSI (camada lógica), ao contrário dos *routers* que operam no nível 3 (camada física) e dos repetidores que operam no nível 1 (camada física). A *bridge* desenvolvida interliga duas LANs do tipo IEEE 802.3 (redes *Ethernet*).

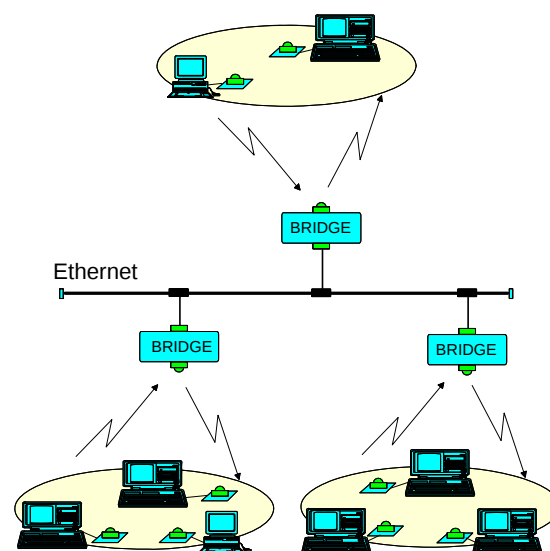


Fig. 1: Diagrama geral da rede.

A. A rede *Ethernet* (IEEE802.3)

O *standard* IEEE 802.3 define uma família de sistemas baseados no protocolo de acesso ao meio CSMA/CD para velocidades de transmissão entre 1Mbps e 10Mbps. O modo de operação do protocolo de acesso ao meio, CSMA/CD (Carrier Sense Multiple Access with Collision Detection) pode descrever-se da seguinte forma: quando uma estação deseja transmitir, ela "observa" primeiro o canal de transmissão para verificar se há actividade neste. Se o canal estiver livre a estação transmite de imediato. Caso contrário espera até este ficar desocupado e tenta imediatamente transmitir. No caso de duas ou mais

* Trabalho realizado no âmbito da disciplina de projecto

estações comecem a transmitir simultaneamente ocorrerá uma colisão. Nesta situação as estações deverão interromper as suas transmissões e esperar um intervalo de tempo aleatório antes de voltar a repetir todo o processo.

Na implementação em banda base é usado o código de linha *Manchester* o qual facilita a extracção do sinal de relógio por parte do receptor. A interface entre o cabo e a estação é constituída tipicamente por um *transceiver* e um controlador de rede. O *transceiver* faz a interface entre o controlador e o cabo. É ainda responsável pela detecção da portadora (*carrier sense*) e pela detecção de colisões (*collision detection*). Por seu lado, o controlador é responsável pela implementação do protocolo de acesso ao meio e pela interface com o CPU (*Central Processor Unit*).

É possível estender um troço de cabo acima do valor especificado pela topologia usada, fazendo uso de repetidores. No entanto uma trama enviada por uma dada estação nunca poderá passar por mais de quatro repetidores.

B. A trama Ethernet

Designa-se por trama, ao conjunto de bits enviados por uma dada estação. Ao nível da subcamada MAC (*Medium Access Control*), a trama é vista como um conjunto de campos. O protocolo IEEE 802.3 especifica o formato da trama como mostra a Fig.2.

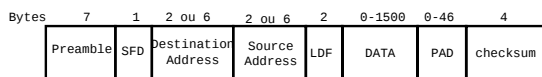


Fig. 2: Formato da trama Ethernet.

Seguidamente descreve-se o significado de cada um dos campos:

íPreamble: sequência de uns e zeros consecutivos que permitem ao relógio do receptor sincronizar com o do emissor.

íSFD (*Start of Frame Delemiter*): indica o início da trama.

íEndereços: identificam o destinatário (*Destination Address*) e remetente (*Source Address*) da trama. Esses campos têm um comprimento de 2 ou 6 bytes.

íLDF (*Length of Data Field*): indica o número de bytes que ocupa o campo de dados (DATA).

íDATA: campo de dados, o seu comprimento pode variar entre 0 e 1500 bytes.

íPAD: campo de comprimento variável que só é inserido nas tramas mais pequenas de maneira a terem um comprimento mínimo de 64 bytes (comprimento mínimo da trama Ethernet). Assim, qualquer trama com comprimento inferior a 64 bytes terá um campo PAD.

íChecksum: campo que permite a detecção de erros.

A Fig. 3 relaciona o protocolo IEEE 802.3 com o modelo de referência OSI. Por definição uma *bridge* implementa apenas as camadas física e lógica.

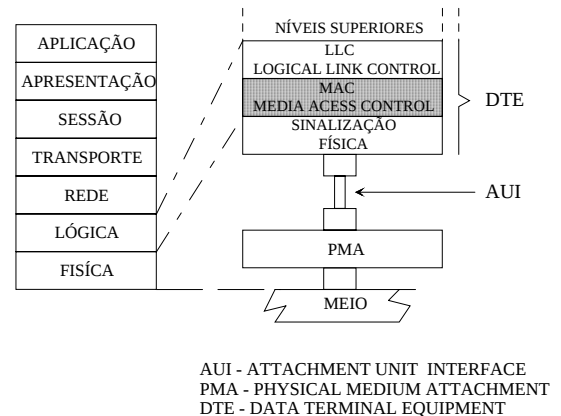


Fig. 3: Relação entre o protocolo CSMA/CD (Std IEEE 802.3) com o modelo de referência OSI.

C. A rede sem fios

A rede sem fios proposta [5] difere de uma rede Ethernet apenas no nível físico, utilizando o mesmo protocolo de acesso ao meio (CSMA/CD) desta. Esse meio consiste no espaço livre de uma sala, não num cabo, a transmissão no meio passa a ser sinais ópticos e não de sinais eléctricos. Esta solução permite facilitar o *interworking* com o backbone Ethernet uma vez que *backbone* e células operam com o mesmo protocolo de acesso ao meio [5].

D. A bridge

A *bridge* é constituída por dois portos sendo cada um deles sendo constituído pelos elementos representados na Fig. 4.

Um dos portos está ligado à rede Ethernet - Backbone (Port_B) e o outro à rede sem fios - Célula (Port_C).

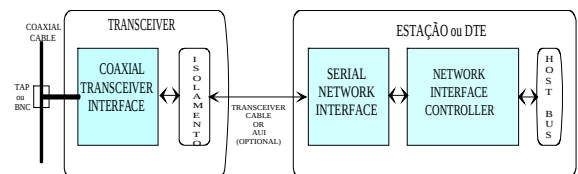


Fig. 4: Diagrama de blocos de um porto da bridge.

A rede Backbone pode ser encarada como a espinha dorsal do sistema de comunicação. As redes Célula podem ser encaradas como ramificações dessa espinha dorsal. Assim, a *bridge* pode ser vista como um dispositivo que estende a rede principal.

Uma característica importante da rede Célula em relação à rede Backbone é que estas somente diferem relativamente

à velocidade de transmissão. Assim o *software* a desenvolver para os *drivers* da *bridge* associados a cada um dos portos é praticamente o mesmo.

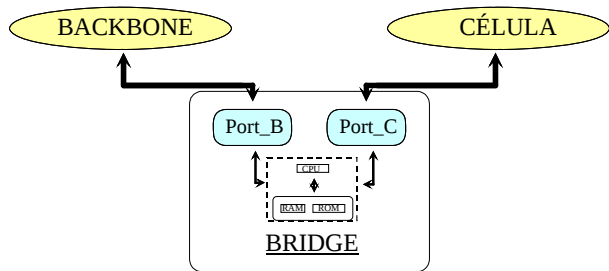


Fig. 5: Esquema básico da *bridge*

Em termos físicos, além dos dois Portos, a *bridge* contém ainda a *hardware* necessário para armazenar e processar a informação (processador, memória RAM, memória ROM, etc.).

A Fig. 5 mostra o esquema básico da *bridge*.

E. Funcionamento da *bridge*

Como já se referiu acima, a *bridge* é um dispositivo que interliga duas LANs e difere dum simples repetidor nos seguintes aspectos:

1 Filtragem:

A *bridge* processa toda a informação que viaja nas redes que interliga de modo a seleccionar as tramas que têm seguimento para a rede oposta àquela em que foram transmitidas e as que não têm seguimento. Esta operação de filtragem é executada de acordo com os dados armazenados em duas tabelas de endereços (uma associada a cada porto) e é uma das principais funções da *bridge*. Esta particularidade permite evitar uma sobrecarga de tráfego nas redes que são interligadas pela *bridge*. A *bridge* inicializa e actualiza as tabelas de endereços automaticamente e também adaptativamente através de um processo de um processo designado por *bridge learning*.

1 Buffering:

No eventual seguimento das tramas de uma rede para a outra, a *bridge* não poderá retransmiti-las ao mesmo ritmo com que chegam, devido à diferença de velocidades de transmissão, ao tempo de processamento e ao eventual "congestionamento" das redes. Assim ela terá que as armazenar num buffer para posterior retransmissão.

A *bridge* projectada é uma *bridge* transparente - *bridge* inteligente que não necessita de inicialização externa e "aprende" ao longo do tempo de funcionamento a localização das máquinas ligadas a cada uma das redes que interliga (*bridge learning*).

A *bridge* funciona com os seus controladores em modo promíscuo. Neste modo a *bridge* analisa todas as tramas que viajam nas redes a ela ligadas.

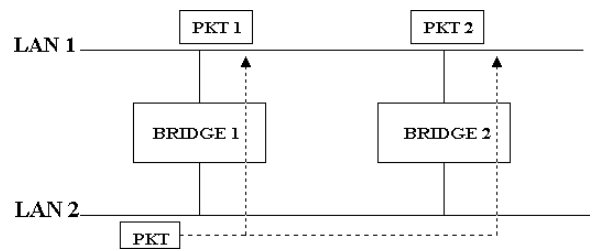


Fig. 6: Exemplo de uma duplicação de tramas.

No caso da topologia considerada, o acesso a uma rede Célula será feito por uma única *bridge*. Esse aspecto permite que não se gerem eventuais duplicações de tramas que circulam na rede, além de eliminar possíveis situações de *dead lock*- existência de loops fechados que podem provocar a circulação permanente de um pacote entre a rede Célula e a rede *Backbone* [1]. Esta situação acontece quando uma rede pode ser acedida por várias *bridges* que não têm conhecimento do destinatário desse pacote. A Fig. 6 exemplifica esta situação. Sendo assim não é necessário neste caso implementar os algoritmos de *spanning tree* que visam, precisamente, evitar as situações de *dead lock* [1,2].

III. DESENVOLVIMENTO DO SOFTWARE

A. Objectivos

Nesta parte do trabalho desenvolveu-se uma *bridge* suportada por um PC IBM compatível para interligar duas redes *Ethernet* funcionando a 10 Mbps.

Este *software* servirá de ponto de partida para o desenvolvimento de uma *bridge* que interligará uma rede *Ethernet* a 10 Mbps tendo um cabo coaxial como meio, com uma rede sem fios e que terá uma velocidade de transmissão de, aproximadamente, 1Mbps.

B. Constituição

As placas de rede que fazem a interface entre o PC, que suporta a *bridge*, e as redes são placas WD8003 da SMC (*Standard Microsystems Corporation*) [3]. Os drivers para essas placas foram criados tendo como base funções cedidas pela SMC o que permitiu uma fácil optimização do *software* para os drivers.

O *software* desenvolvido é constituído por três grandes partes:

- ✦ os drivers para as placas WD8003;
- ✦ as funções que implementam o processo de bridging;
- ✦ as funções que suportam o interface com o utilizador.

As funções que constituem estas três partes foram desenvolvidas em Microsoft C600, com excepção das rotinas enviadas pela SMC que tinham sido desenvolvidas

em Microsoft C500, segundo informações da própria SMC.

C. Algoritmo da bridge

O funcionamento da *bridge* pode-se resumir nos seguintes passos:

- 1 - Inicialização das tabelas de endereços para os dois portos;
- 2 - Recepção das tramas de um porto;
- 4 - Processamento das tramas;
 - 4.1 Verificar se uma determinada trama tem que seguir para o porto oposto, se sim então armazená-la em memória.
 - 4.2- Actualização das tabelas de endereços.
- 5 - Transmissão das tramas que tenham como destino uma máquina do porto oposto.
- 6 - Repetir os passos 2, 3, 4 e 5 para o outro porto.

Os passos 2, 3, 4, 5 e 6 são repetidos até que uma tecla seja primida.

Interrompendo o processo de *bridging* pode-se alterar a configuração das placas de rede, introduzir ou remover os endereços permanentes nas tabelas de endereços e consultar as estatísticas referentes ao último bridging.

D. Os testes da bridge

Depois de desenvolvido o software era necessário testá-lo para verificar o seu funcionamento e tentar otimizar o desempenho da *bridge*, principalmente no que se referia à sua velocidade de processamento. Era ainda necessário fazer as medidas relacionadas com o desempenho.

Para os testes da *bridge* usou-se o analisador de tráfego *NetSight Analyst v.1.1* da Intel [4]. Este analisador tem a facilidade de gerar tráfego, permitindo controlar o número de pacotes a enviar por segundo e o seu tamanho.

A base de trabalho usada para testar a *bridge* e medir as *performances* é a representada na Fig. 7. A referida base de trabalho é constituída por:

- ☛ Um PC + 2 placas de rede (WD8003 da SMC). Este PC suporta o *software* desenvolvido.
- ☛ Um PC + 1 placa de rede para simular a rede Célula.
- ☛ Um PC + 1 placa de rede para simular a rede Backbone.

Nestes testes instalou-se no PC que simulava uma das redes o *NetSight Analyst* a gerar tráfego, e no PC que simulava a outra rede o *NetSight Analyst* a capturar esses pacotes (que tinham como endereço de destino o endereço físico da placa instalada nesse PC). Pode-se desta forma verificar a velocidade a partir da qual a *bridge* não conseguia passar todos os pacotes transmitidos para o porto oposto e a percentagem de pacotes perdidos.

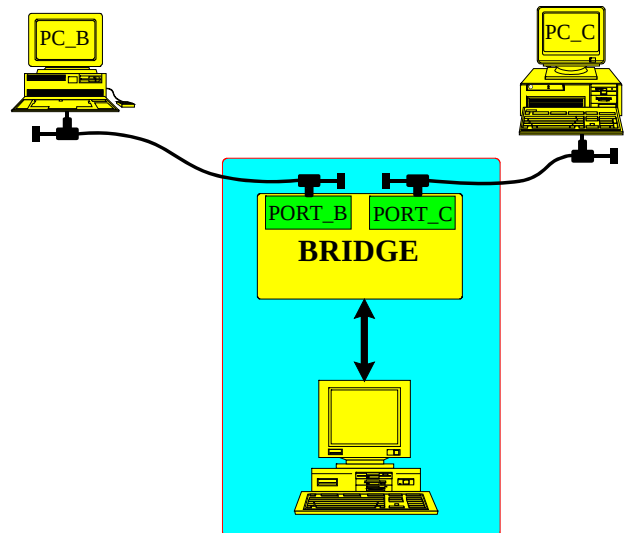


Fig. 7: Base de trabalho usada para testar a *bridge* e medir as suas *performances*.

Fizeram-se testes para pacotes *Ethernet* com comprimento de 60, 1520 e 650 *bytes*. Usaram-se estes comprimentos porque na rede *Ethernet*, 60 *bytes* é o tamanho mínimo de uma pacote, 1520 *bytes* é o tamanho máximo e 650 *bytes* é um valor intermédio.

Como exemplo, pode-se dizer que nas primeiras versões da *bridge* uma transferência de 190 ficheiros num total de 2.312.894 *bytes* de informação, usando o *ftp*, entre uma máquina *Unix* e um PC situado na Célula, demorou cerca de 410 segundos, em horas de pouca carga na rede, o que dá uma média de 45462 bps (5682 *bytes* por segundo). Na última versão uma transferência de 218 ficheiros num total de 2416789 *bytes* de informação demorou cerca de 123 segundos (157189 bps - 19648 *bytes* por segundo), em condições idênticas. Esta última transferência realizada logo de seguida mas sem a *bridge* a interligar a fonte e o destino (estavam na mesma rede) demorou cerca de 68 segundos (284328 bps - 35541 *bytes* por segundo), cerca de metade do tempo gasto com a *bridge* a interligar os dois pontos extremos (fonte e destino da informação). Pode-se ver que a *performance* mais do que triplicou o que mostra uma evolução significativa no refinamento do software. Este teste foi realizado com 50 entradas na tabela de endereços associada à rede Backbone e 20 entradas na tabela de endereços associada à rede Célula.

Verificou-se que o número de entradas nas tabelas de endereços têm uma grande influência no desempenho da *bridge*. Esse número não deve ser muito pequeno para garantir que só atravessam a *bridge* as tramas que realmente tenham como destino uma máquina do porto oposto, ou as tramas cujo endereço destino seja do tipo *broadcast*. Também não deve ser exageradamente grande para não se perder muito tempo com o processamento (*bridging*), o que implica velocidades baixas para a *bridge*. E, o número de entradas na tabela de endereços associadas à rede Backbone deve ser bastante maior que o

número de entradas na tabela associada à rede Célula, visto as dimensões físicas serem bastante diferentes.

Estes testes permitiram no entanto verificar que existem algumas limitações de performance devidas ao facto da *bridge* ser baseada num PC, como aliás era esperado.

III. DESENVOLVIMENTO DO *HARDWARE*

Nesta segunda parte do trabalho desenvolveu-se um sistema *stand-alone* em *hardware* que permitirá implementar todas as funções da *bridge* desenvolvida na primeira parte, para além de outras, nomeadamente de gestão [2].

Esta segunda parte decorreu em duas fases:

- ↳ projecto do hardware da *bridge*.
- ↳ montagem e teste da placa por etapas, de maneira a mostrar o bom funcionamento do sistema projectado.

A *bridge* que se pretende implementar é basicamente constituída por:

- dois portos, cada um ligado a uma rede. Esses portos fazem a interface entre a rede a qual estão ligados e o *bus* do sistema, sendo o seu principal componente um controlador de rede 82586 da Intel.

- Um microprocessador que faz a gestão dos recursos do sistema e processa a informação envolvida durante o funcionamento da *bridge*. A utilização dum CPU da família do 80186 simplificou muito o desenho da placa, permitindo um elevado desempenho do sistema numa configuração com um mínimo de hardware.

- Memória ROM onde é armazenado o programa que implementa as várias funções da *bridge* e ainda, informação para a inicialização dos controladores.

- Memória RAM para armazenar a restante informação (tabelas de endereços, tramas, comandos e registos).

- *Latches* de endereços que fazem a captura dos endereços lançados pelo microprocessador, permitindo aos controladores e ao CPU o acesso às memórias.

- *Data trasceivers* que servem de buffer e de driver ao varios componentes ligados ao *bus* do sistema.

- Lógica para descodificação dos endereços lançados pelo microprocessador que permite activar os sinais de *chip select* das memórias e eliminar o conflito na inicialização dos controladores que acedem, por defeito, a uma posição predeterminada da memória ROM.

- Lógica suplementar que garanta a exclusividade de posse do *bus* por parte dos controladores quando estes lhe querem aceder.

- Gerador de *wait state* que permite programar o número de *wait state* no acesso dos controladores às memórias.

A Fig. 8 mostra o diagrama de blocos do sistema que se implementou.

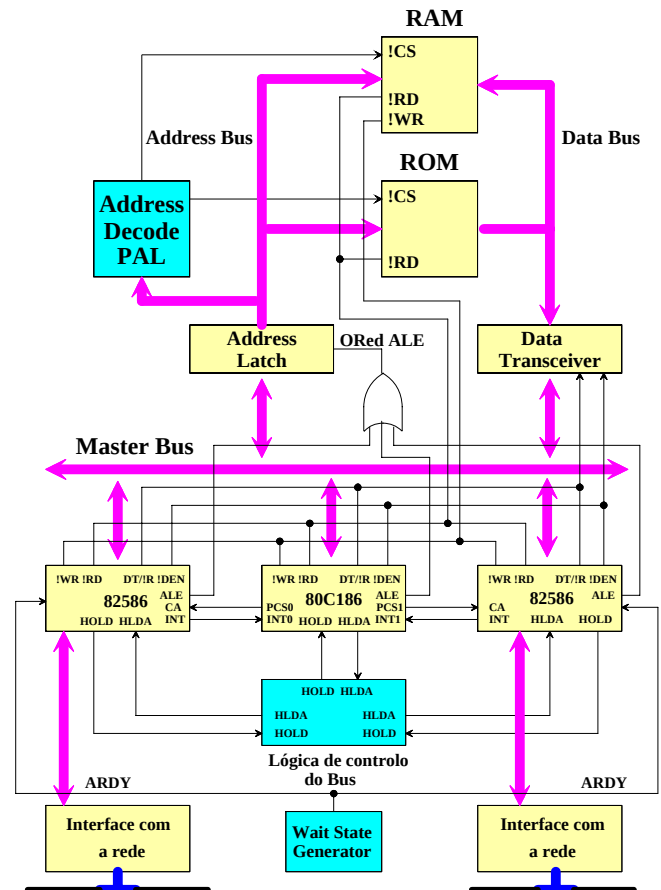


Fig. 8: Diagrama de blocos da *bridge*

A montagem e os testes da placa efectuaram-se em três etapas, tendo como objectivos a verificação do funcionamento das seguintes interfaces:

- entre o CPU e as memórias, verificando as operações de leitura nas memórias ROM e RAM, escrita na RAM, inicialização do processador e activação dos sinais de *chip select* programáveis;

- entre o CPU e um controlador, conferindo os vários passos na inicialização do controlador e o correcto acesso às memórias;

- entre o CPU e os dois controladores, confirmando os endereços e os dados de inicialização do sistema e a resolução do conflito no acesso ao *bus* por parte dos controladores.

Todos os testes foram efectuados com sucesso.

CONCLUSÕES

No que se refere à primeira parte do trabalho (secção II), o objectivo principal era desenvolver uma *bridge* transparente que servisse de base ao desenvolvimento de uma *bridge* que assentará sobre o *hardware* apresentado na segunda parte (secção III). Esse objectivo foi atingido, pois o algoritmo é eficaz no que se refere ao processo de *bridging*. Como comprovação desse facto temos que as velocidades da *bridge* são bastante boas tendo em

consideração que é uma *bridge* suportada por um PC. Outras melhorias em termos de velocidade de processamento poderiam ainda ser feitas se algumas das rotinas responsáveis pelo *bridging* fossem desenvolvidas em *assembly*.

O trabalho desenvolvido na segunda parte consistiu no projecto, na montagem e no teste da placa sobre a qual se pretende desenvolver a *bridge*. Os resultados obtidos são muito animadores, tendo-se conseguido uma montagem simples (com um mínimo de *hardware*) e funcional, nomeadamente no que respeita a resolução do conflito no acesso ao *bus* (pela lógica de controlo do *bus*) e na inicialização dos controladores de rede. A maneira como foi concluído o trabalho permitirá o prosseguimento do projecto sem necessidade do conhecimento específico das várias interfaces, isto é a plataforma *hardware* básica para a *bridge* ficou concluída. Torna-se agora necessário

aumentar a velocidade de relógio para a frequência prevista, montar as interfaces séries de rede e montar a interface de comunicação série com o computador. Note que esquemas detalhados destas interfaces já existem, não devendo portanto criar nenhum problema grave.

AGRADECIMENTOS

Agradecemos à SMC as rotinas enviadas. Aos docentes do GSBL pelo apoio dispensado.

REFERÊNCIAS

- [1] Tanenbaum, Andrew S, *Computer Network*, Prentice Hall Software.
- [2] *Std IEEE 802.1D 1990*, IEEE.
- [3] *WD8003E Hardware Installation Guide*, Western Digital.
- [4] *Intel NetSight AnalystTM Reference Guide*, Intel.
- [5] Rui T. Valadas, Adriano C. Moreira, A.M. de Oliveira Duarte "Hybrid (Coaxial/Infrared) Ethernet Local Area Networks" *Proceedings IEEE Conference on Wireless LAN Implementation*, Dayton, Ohio, USA, 17-18 September 1992.