

# Desenvolvimento de uma Ferramenta para Comutação de Pacotes Aplicada ao Ensino de Redes na Engenharia

Alexandre Timm Vieira, Antônio Kieling, Fábio Duarte, Flávia L. Caminha, Jorge Silveira Guedes, Lourenço Basso, Ricardo Balbinot

Programa de Pós-Graduação em Engenharia Elétrica Faculdade de Engenharia  
Pontifícia Universidade Católica do Rio Grande do Sul  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{ atvieira, akieling, fduarte, fcaminha, jguedes, lbasso, rbalbinot }  
@gparc.org

**Resumo.** *Este artigo apresenta uma proposta de construção de uma ferramenta educacional, específica para o ensino de redes de computadores, com a finalidade principal de demonstrar o funcionamento de comutadores de pacotes em redes Ethernet 802.3. Trata-se de uma ferramenta de domínio público com os códigos fontes disponíveis, direcionado para sistemas UNIX. Sua utilização é voltada para ambiente reais em computadores com múltiplas interfaces de rede onde o seu usuário pode determinar o número de interfaces participantes da comutação e acompanhar o processo de aprendizagem da tabela de endereçamento. Esta é a dinâmica, sendo conduzida por seu usuário através de instruções do teclado em uma interface por linha de comandos.*

**Palavras-Chave.** *Comutador de pacotes, ponte, switch, ethernet, educacional*

## 1. Introdução

A quantidade de informações geradas pela era do conhecimento é exponencial. Levando em consideração a área de tecnologia de informação, com suas inúmeras ramificações, teremos um sem-fim de novos conceitos e técnicas sendo geradas e/ou agregadas as anteriores.

Uma das áreas que apresentam desafios constantes é a de tecnologia de redes locais, a qual será abordada neste trabalho com enfoque na técnica de comutação de pacotes de redes Ethernet.

O presente artigo propõe um sistema que possibilite acrescentar facilidades ao processo de ensino/aprendizagem da área de redes de comunicação, possibilitando abstração do processo de comutação de pacotes sobre redes Ethernet. Para tanto um sistema comutador de pacotes foi desenvolvido, disponibilizando códigos fonte, com documentação de codificação, e documentos complementares acerca da tecnologia de redes trabalhada. É um sistema funcional, desempenhando tarefas reais, que trabalha como um dispositivo de comutação.

Com o objetivo de uma melhor organização, o artigo está dividido em três (3) seções que apresentam: os padrões especificados pelo grupo 802.1D do IEEE para a comutação de pacotes e pelo grupo 802.3 para a tecnologia de rede local Ethernet

(seção II); os aspectos relacionados à implementação das técnicas necessárias ao projeto e programação do sistema em questão (Seção III) e conclusão com a finalização do trabalho, comentando as vantagens a serem conseguidas, dificuldades encontradas, e possíveis melhorias a serem efetuadas no presente trabalho (Seção IV).

## 2. A Tecnologia Ethernet e a Comutação de Pacotes

As redes com tecnologia Ethernet são normatizadas com o padrão 802.3 do IEEE (*Institute of Electrical and Eletronic Engineers*) que define a sinalização elétrica e a forma de acesso ao meio da rede local - LAN (*Local Area Network*). Esse padrão utiliza o protocolo de acesso ao meio CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), operando a sinalização de 10, 100 e 1000 Mbps, em modo de transmissão half-duplex ou full-duplex e com controle de erros tipo CRC no bloco transmitido. Este artigo não tem como objetivo concentrar-se na tecnologia Ethernet, abordando somente suas regras básicas, como pré-requisito para compreensão dos dispositivos de comutação de pacotes (Bridges e Switches).

No protocolo CSMA/CD, as estações que desejam transmitir esperam até que o meio esteja livre para iniciar a transmissão de seus dados (*carrier sense*). Se várias estações tentam transmitir em um mesmo instante (*multiple access*), simultaneamente, uma colisão ocorre e então cada estação espera por um intervalo de tempo randômico para tentar transmitir novamente (*collision detection*). O acesso ao meio é, portanto, aleatório e assíncrono, ocorrendo por competição. Quando os dados chegam ao destino, a estação receptora não envia nenhuma confirmação de que recebeu o pacote.

Um segmento de rede Ethernet é definido como um domínio de colisão, arbitrado pelo protocolo CSMA/CD, formando uma topologia lógica em barramento e fisicamente uma topologia em barramento ou estrela.

Um domínio de colisão pode ser estendido com o uso de repetidores (*repeaters*) na topologia em barra e com o uso de hubs (nó central) na topologia em estrela, mas não haverá segmentação da rede, pois as estações dos dois trechos de rede conectados pelo repetidor ou hub irão competir pelo acesso ao mesmo local, constituindo ainda, um único domínio de colisão.

Os repetidores e hubs são classificados como dispositivos da camada 1, no modelo RM-OSI, porque eles atuam apenas no nível do bit e não consideram nenhuma outra informação. A função básica é estender o alcance da rede, copiando bits de um trecho para outro. São usados em casos em que o cabo que interliga estações distantes apresenta um comprimento maior que o limite máximo estipulado para a tecnologia.

Em casos em que a rede é estendida usando-se mais de um repetidor ou hub, deve-se ficar atento a regra "5-4-3". Essa regra afirma que pode-se conectar cinco segmentos de rede ponto a ponto usando quatro repetidores/hubs, mas apenas três segmentos podem ter computadores. O motivo para esta limitação está relacionado ao intervalo mínimo entre os quadros (*frames*). O padrão Ethernet de 10 Mbps define o intervalo entre os quadros de 9,6 microsegundos (0,0000096 segundos), que significa que as estações não podem transmitir quadros na rede com um espaço inferior a 9,6 $\mu$ s. O intervalo entre os quadros é de 0,96 $\mu$ s em Fast Ethernet e 0,096 $\mu$ s no sistema Gigabit Ethernet. Essa presença de intervalo entre os quadros ajuda a estabelecer o tempo de recuperação para uma interface Ethernet, após o qual ela deve estar pronta para aceitar

um novo quadro. No entanto, deve-se prever algo chamado "encolhimento de intervalo entre quadros". A medida que cada quadro passa por um repetidor/hub de 10 Mbps, o repetidor/hub regenera os bits perdidos do preâmbulo. Se o primeiro quadro tiver experimentado mais perda de bits do que o segundo, então o intervalo entre eles encolherá enquanto saem do repetidor/hub. Conseqüentemente, os quadros podem acabar sendo separados por menos do que o intervalo de 9,6  $\mu$ s. O encolhimento do intervalo é um comportamento esperado e permitido no padrão. No entanto, se o intervalo entre quadros sucessivos for muito pequeno devido a travessia pelos vários repetidores/hubs, então a interface poderá não ser capaz de recuperar-se em tempo de ler o próximo quadro. O resultado poderia ser uma fonte de quadros perdidos à medida que as interfaces descobrem que não podem acompanhar.

## 2.1 Comutação de Pacotes

Para se montar grandes sistemas Ethernet que se estendem além dos limites de um único domínio de colisão, seja em topologias físicas em barra ou estrela, é necessário o uso de dispositivos de comutação de pacotes (encaminhamento de pacotes). O termo comutação veio originalmente das redes telefônicas, em que o equipamento que trata de chamadas telefônicas é conhecido como um chaveador. O termo chaveador é agora aplicado a equipamentos eletrônicos usados para redes de dados (usando tecnologia Ethernet, por exemplo), e tais redes são chamadas de redes de dados comutadas.

Dispositivos de comutação, como Pontes - *Bridges* e Hubs de Comutação - *Switches*, (topologias em barramento e estrela, respectivamente) ligam segmentos Ethernet que operam em diferentes velocidades e controlam o fluxo de tráfego através de um sistema, além de melhorar a confiabilidade e aumentar bastante a quantidade de largura de banda Ethernet disponível para uso.

As pontes e switches são chaveadores de pacotes que operam na camada 2 do modelo de referência OSI, em nível de quadros Ethernet, tendo como principal função isolar segmentos de rede Ethernet compartilhados, de modo a evitar congestionamentos. Para a expansão de redes cuja taxa de utilização é superior a 40%, ou que já possuem 4 repetidores ou hubs no enlace, ou ainda, que apresentem um retardo de propagação próximo a 51,2 microssegundos, a solução imediata é a segmentação em sub-redes através da utilização de uma ponte ou switch.

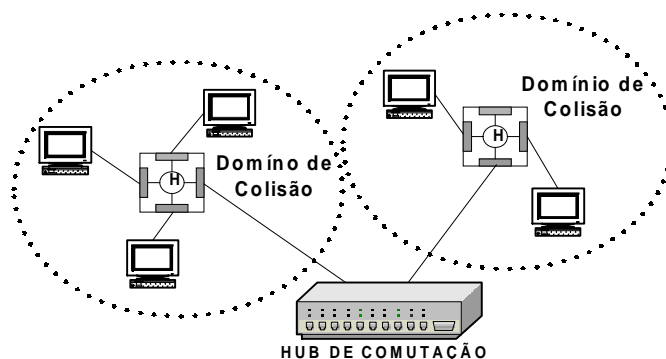


Fig. 1. Switches interrompem o domínio de colisão

As primeiras pontes Ethernet (início da década de 80) eram dispositivos de duas portas que podiam unir dois segmentos Ethernet. Mais tarde, foi possível projetar e

vender pontes com muitas portas, que foram usadas no hub de um sistema de cabeamento, e por isso passaram a ser conhecida como hub de comutação ou switch, conforme vemos na Figura 1. Neste artigo, usamos a palavra "ponte" e "switch" para indicar a mesma coisa ao descrevermos como esses dispositivos funcionam.

Os switches em específico, também podem ser usados para conexão direta de estações terminais, onde cada segmento de LAN compartilhada conecta uma estação apenas em específico. Nesse extremo temos a microsegmentação. Em um microsegmento, cada estação tem uma banda dedicada para operação, sendo as taxas de dados utilizadas em cada estação, independente umas das outras. A prioridade maior de um microsegmento é não existir colisões entre estações terminais, possível, contudo, ter colisões entre o terminal MAC do Switch e da estação. Pode-se eliminar as colisões via uso da operação em full-duplex.

Observe que a operação das pontes e switches não é especificada no padrão IEEE 802.3 Ethernet. Ao invés disso, esses dispositivos são baseados no padrão 802.1D, que oferece regras para encaminhamento (comutação) de um quadro Ethernet de uma porta para outra, com base no endereço de destino do quadro. Embora o padrão 802.1D ofereça regras para a passagem de quadros entre as portas de uma ponte e para alguns outros aspectos da operação da ponte, o padrão não especifica o desempenho ou a embalagem da ponte.

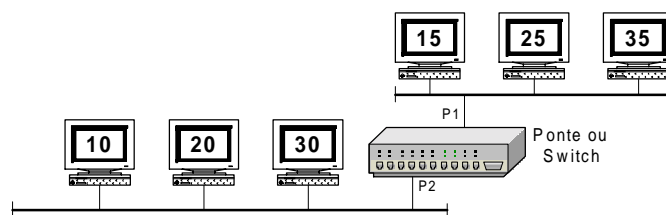
## 2.2 Funcionamento

A ponte só permite que um quadro passe de um segmento para o outro se o endereço de destino (endereço MAC - *Media Access Control*) for de uma das estações do outro segmento. Desta maneira, um segmento da rede vai impactar o tráfego do outro segmento apenas quando houver tráfego comum entre eles. A ponte controla este fluxo de tráfego entre os segmentos com um mecanismo de encaminhamento automático de tráfego, descrito no padrão de pontes IEEE 802.1D.

### Processo de Auto-aprendizagem

O encaminhamento de tráfego é baseado no aprendizado de endereço, e as pontes tomam decisões de encaminhamento com base nos endereços dos quadros Ethernet. Para fazer isso, a ponte aprende quais estações estão em quais segmentos da rede (Figura 2) verificando os endereços de origem em todos os quadros que a ponte recebe (quando uma estação envia um quadro Ethernet, ela coloca dois endereços nesse quadro, o destino a qual está enviando o quadro e o endereço de origem da estação que envia o quadro).

Ao contrário de uma estação normal, que só lê quadros endereçados diretamente a ela, a interfaces Ethernet em cada porta de uma ponte trabalha no modo "promíscuo". Nesse modo, a interface lê todos os quadros que encontrar na LAN conectada, e não apenas os quadros que estão sendo enviados para o próprio endereço MAC da ponte. À medida que cada quadro é lido em cada porta da ponte, o software da ponte verifica o endereço de origem do quadro e inclui o endereço em uma tabela de endereço mantida pela ponte (tabela SAT - *Source Address Table*), como o que aparece na Tabela I. É assim que a ponte descobre quais estações podem ser atingidas em quais portas (processo de self-learning). Há algum tempo, as pontes exigiam que o usuário programasse manualmente as tabelas internas com os endereços MAC.



**Fig. 2. Aprendizado de endereço de uma ponte/switch**

As pontes também devem desaprender os endereços. A ponte registra a idade de cada entrada de endereço no banco de dados de endereços (tabela SAT) e apaga o item após um certo período de tempo (normalmente de dois a cinco minutos) se nenhum quadro for recebido com esse endereço de origem (Tabela I). Isso permite que as estações sejam alteradas de um segmento para outro sem o problema da ponte manter as tabelas de endereços permanentemente, o que não refletiria a nova realidade.

**TABELA 1. Banco de dados de encaminhamento mantido pela ponte/switch**

Estação – Endereço MAC	Porta	Idade
15	P1	38s
25	P1	15s
35	P1	2s
10	P2	84s
20	P2	7s
30	P2	119s

### Encaminhamento dos Quadros

Para encaminhar os quadros, as pontes consultam a tabela interna SAT e analisam o endereço de destino do quadro. Se o endereço de destino pertencer a uma estação no mesmo segmento que a estação origem, a ponte despreza o quadro, não o repassando para o outro segmento. Caso contrário, a ponte copia o quadro para o outro segmento. Se o endereço destino do quadro, não estiver na tabela, a ponte envia, de qualquer maneira, o quadro para todos os outros segmentos (enchente ou flooding). Todas as pontes 802.1D precisam realizar o flooding de quadros desconhecidos.

### Pontes e Switches Transparentes

A ponte copia o quadro para o outro segmento, colocando-o na fila para transmissão da porta de saída que liga a este outro segmento. A ponte contém uma memória de buffer de curta duração para cada porta, caso o segmento de saída esteja ocupado quando o quadro de transmissão chegar para a transmissão. Durante este processo, uma ponte transmitindo um quadro Ethernet não faz mudanças nos campos de dados, de endereço ou tipo de quadro, o quadro é transmitido intacto para a porta de saída, exatamente como foi recebido na porta de entrada, portanto, no que se refere à entrega do quadro, a operação da ponte é transparente a todas as estações da rede.

## Tratamento de Colisões

Quando vai fazer a transmissão de um quadro de um segmento a outro, a ponte deve seguir as regras de acesso definidas no método CSMA/CD. Se o outro segmento estiver ocupado, a ponte deve aguardar a sua liberação para tentar a transmissão. Caso ocorra uma colisão, a ponte é responsável pelo processo de reversão e transmissão como qualquer estação da rede Ethernet. A estação de envio original não é notificada da ocorrência da colisão e considera que o quadro foi enviado de forma correta. Caso a ponte não seja capaz de enviar o quadro ao seu destino final, a estação de envio original, aguardando uma resposta do dispositivo que estava tentando conectar, irá observar que o tempo máximo de espera foi atingido (*time-out*) e, dependendo do protocolo das camadas superiores, irá tentar a retransmissão. Caso, contudo, a interface considerada seja full-duplex, o envio do frame é direto e sem maiores delongas.

## Métodos de Encaminhamento

Pode-se identificar quatro modos distintos para encaminhamento de quadros, o modo armazena-e-encaminha (*store-and-forward*), o modo direto (*cut-through*), o modo direto modificado (*modified cut-through*) e o modo adaptativo (*error sensing*).

O modo de processamento interno no dispositivo de comutação pode diminuir a latência para os quadros, conforme o modo de operação para encaminhamento. As pontes usam o processamento *store-and-forward*, o que significa que todo o quadro é recebido antes do primeiro bit do quadro ser reenviado.

Os switches podem usar a lógica do *store-and-forward* assim como o processamento *cut-through*. Com o processamento *cut-through*, os primeiros bits do quadro são reenviados na porta de saída antes do último bit do quadro de chegada ser recebido, em vez de esperar todo o quadro ser recebido. Em outras palavras, tão logo a porta de entrada receba o suficiente do quadro para que veja o endereço de destino, o quadro é transmitido pela porta de saída apropriada para a estação de destino. O "efeito colateral" é que como o campo de detecção de erros FCS (*Frame check sequence*) está no cabeçalho Ethernet, e o quadro reenviado pode ter erros de bit que o switch teria detectado com a lógica *store-and-forward*, na qual não propaga quadros com erro.

A diferença entre o modo *cut-through* e o modo *modified cut-through* é que, no modificado, o switch chaveia a operação de acordo com o tamanho do quadro. O switch opera no modo direto para quadros maiores de 64 octetos e descarta quadros menores. Esta técnica também chamada de - *Fragment Free* - filtra fragmentos de colisão, visto que elas são detectadas nos primeiros 64 octeto. Finalmente, no modo adaptativo - *error sensing*, o switch inicia a operação no modo *cut-through*, apenas monitorando os FCS dos quadros. Se o número de quadros com erro se tornar elevado, segundo um limite pré-estabelecido o switch automaticamente altera para o modo *store-and-forward*. Assim que a "tempestade" de quadros errados passar, o switch retorna ao modo *cut-through* com monitoração FCS.

## Transmissão Full-Duplex

A operação em full-duplex não se realiza com o uso de meios Ethernet compartilhados, como um barramento físico ou em uma topologia em estrela usando hubs (barramento lógico). Para haver condições de existir uma operação em full-duplex, devem existir

apenas dois dispositivos na LAN Ethernet. Apenas um switch microsegmentado atende esse critério de duas estações logicamente conectadas diretamente, possibilitando a operação em full-duplex.

A operação Ethernet full-duplex é, na realidade, bastante simples. O protocolo CSMA/CD virtualmente deixa de existir, ou seja, a implantação em si é mais simples do que no caso do modo half-duplex. Uma interface de rede em modo full-duplex não utiliza nenhum protocolo de contenção, daí a impossibilidade de seu uso em meio compartilhados.

### **Domínios de Broadcast e Multicast**

Um endereço de multicast é um endereço de grupo que várias interfaces podem ser configuradas para receber. Portanto, um único quadro com um endereço de destino de multicast pode ser recebido por um conjunto de estações ouvindo esse endereço multicast. Broadcast é um caso especial de multicast, e é o grupo de todas as estações. Portanto, um pacote enviado ao endereço de broadcast (o endereço com todos os bits 1) é recebido por cada estação da LAN. As pontes e switches são projetados para ligar segmentos em uma determinada LAN, de modo que todas as pontes e switches enviem pacotes de broadcast para todas as portas pertencentes a uma LAN - exceto a porta por onde foi recebido o broadcast. Dessa forma, os pacotes de broadcast podem chegar a todas as estações da LAN.

A maioria das pontes e switches não tem como descobrir onde podem estar localizadas as estações ouvindo um determinado endereço multicast. Portanto, elas também causarão flooding de todos os pacotes de multicast por todas as portas, menos aquela onde o quadro foi recebido. Switches mais sofisticados podem usar protocolos de descoberta de grupo multicast para limitar a propagação de pacotes de multicast.

As estações enviam pacotes de broadcast e multicast por diversos motivos. Alguns protocolos de rede de alto nível usam quadros de broadcast ou multicast como parte de seu processo de descoberta de endereço (protocolo ARP). Broadcasts e multicasts também são usados para a atribuição dinâmica de endereço, o que normalmente acontece quando uma estação é ligada inicialmente e precisa encontrar um endereço de rede de alto nível para iniciar a comunicação (DHCP e Bootp).

### **Recursos Avançados**

Em sua forma mais básica, as pontes e switches realizam as funções descritas acima. Novas funcionalidades foram incorporadas pelos fabricantes e organismos de padronização ao longo dos anos, sejam para aumentar o controle sobre o tráfego encaminhado ou para reduzir limitações da tecnologia Ethernet.

### **Filtragem Personalizada de Quadros**

Os filtros personalizados permitem que um administrador de rede especifique a filtragem de quadros com base em diversos parâmetros. Usando estes filtros é possível configurar uma ponte ou switch para controlar o tráfego na rede como base nos endereços do quadro Ethernet, no campo de tipo do quadro Ethernet e etc.

## **VLAN - LANs Virtuais**

Outro recurso opcional, especificamente para switches é a capacidade de agrupar fluxo de tráfego em LANs virtuais, ou VLANs. Em sua forma mais simples, uma VLAN é um grupo de portas do switch que se comportam como se estivessem em um switch independente. Isso é feito manipulando-se o software de encaminhamento de quadros.

Por exemplo, pode-se configurar um switch de 8 portas de modo que as portas de 1 a 4 estejam em um VLAN e as outras portas de 5 a 8 em outra VLAN. Essas VLANs atuam com domínios de broadcast separados, e um broadcast ou um multicast enviado na VLAN 1 não será transmitido em porta alguma que pertença à VLAN 2.

Dois endereços MAC em VLANs distintas só poderão trocar informações através da camada 3 do modelo de referência OSI/ISO. Ou seja, eles estão isolados pela camada de enlace de dados (camada 2).

## **Algoritmo de Spanning Tree**

Uma dificuldade do algoritmo de encaminhamento de quadros da ponte, descrito anteriormente, é que é possível haver dois segmentos, ambos conectados a duas pontes, de modo que as pontes estejam em paralelo. Sem algum meio de evitar o tráfego, as pontes paralelas ligando os mesmos segmentos entrarão em um looping de encaminhamento. Nesse tipo de loop, o tráfego de broadcast e multicast circula infinitamente e continua a crescer à medida que novo tráfego é transmitido, até que a taxa de tráfego fique tão alta que a rede se sature.

Para evitar loops de encaminhamento, o padrão de ponte 802.1D oferece um algoritmo de árvore dividida (*Spanning Tree*). A finalidade do algoritmo de árvore dividida é permitir que pontes e switches em um determinado sistema Ethernet criem dinamicamente um conjunto de percursos sem que haja loop.

A operação do algoritmo de Spanning Tree é baseada em mensagens de configuração enviadas por cada ponte, usando um endereço de multicast que foi reservado para a operação de Spanning Tree. Todas as pontes compatíveis com o IEEE 802.1D escutam quadros enviados para esse endereço, de modo que cada ponte possa enviar e receber mensagens de configuração de Spanning Tree.

As mensagens de configuração contêm informações que permitem a qualquer conjunto de pontes elegerem uma ponte raiz. A ponte raiz prossegue então para enviar mensagens de configuração. Cada ponte usa as informações nas mensagens de configuração recebidas para calcular o melhor percurso a partir dela até a ponte raiz, sem loops.

## **3. Implementação**

Esta seção descreve a implementação do sistema de comutação de pacotes para uso educacional que implementa os recursos de um switch de redes Ethernet, descritos na seção II, com funcionamento em um ambiente real.

Para o ambiente de desenvolvimento foi adotado o sistema operacional UNIX - GNU/Linux, por ser de livre distribuição. O código fonte foi totalmente desenvolvido em linguagem C, orientado para UNIX. O compilador utilizado foi o GCC (*GNU project C and C++ Compiler*), que é amplamente divulgado e faz parte das mais



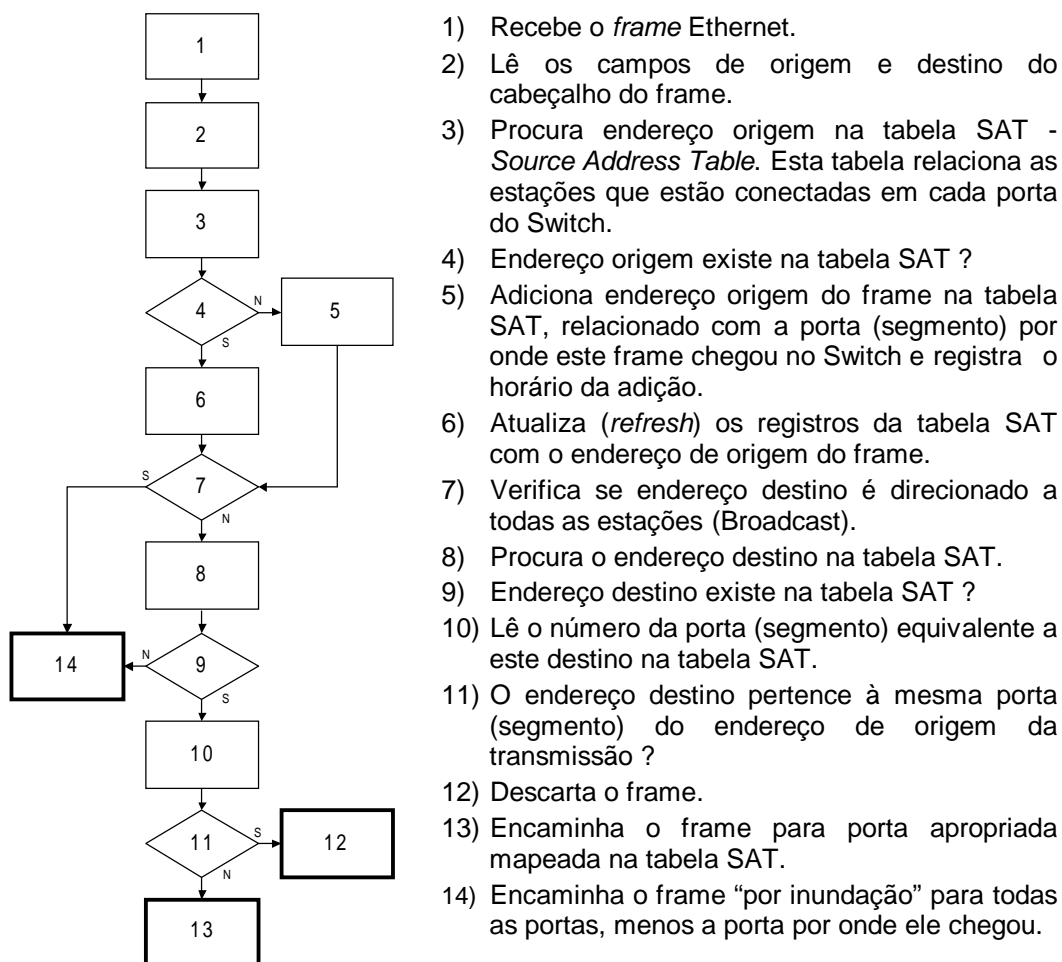
variadas distribuições de GNU/Linux, incluindo as distribuições Mandrake 9 e Conectiva 9, utilizadas durante o desenvolvimento.

O hardware da implementação consiste em um PC (i86) com duas ou mais interfaces de rede do tipo Ethernet, cada uma conectada a diferentes segmentos de rede.

Dentre as ferramentas utilizadas, destacam-se duas bibliotecas para manipulação de pacotes que desempenham um papel decisivo na implementação. São elas: a LibPcap [10] e a LibNet [11]. A primeira, no ambiente GNU/Linux, dedica-se exclusivamente a captura de pacotes das interfaces de rede do sistema. A segunda, dedica-se ao envio de pacotes através destas interfaces.

Através do uso destas bibliotecas, foi possível a criação de um software a nível de usuário que capture os pacotes Ethernet de forma bem simplificada e clara, passando-os para processamento do sistema computador e, em seguida, os envie à interface apropriada.

Para melhor ilustrar o funcionamento da ferramenta, a captura dos pacotes na porta de entrada, o mecanismo de comutação e o envio dos pacotes na porta de saída, conforme detalhado na seção II, é ilustrado na Figura 3.



- 1) Recebe o *frame* Ethernet.
- 2) Lê os campos de origem e destino do cabeçalho do frame.
- 3) Procura endereço origem na tabela SAT - *Source Address Table*. Esta tabela relaciona as estações que estão conectadas em cada porta do Switch.
- 4) Endereço origem existe na tabela SAT ?
- 5) Adiciona endereço origem do frame na tabela SAT, relacionado com a porta (segmento) por onde este frame chegou no Switch e registra o horário da adição.
- 6) Atualiza (*refresh*) os registros da tabela SAT com o endereço de origem do frame.
- 7) Verifica se endereço destino é direcionado a todas as estações (Broadcast).
- 8) Procura o endereço destino na tabela SAT.
- 9) Endereço destino existe na tabela SAT ?
- 10) Lê o número da porta (segmento) equivalente a este destino na tabela SAT.
- 11) O endereço destino pertence à mesma porta (segmento) do endereço de origem da transmissão ?
- 12) Descarta o frame.
- 13) Encaminha o frame para porta apropriada mapeada na tabela SAT.
- 14) Encaminha o frame "por inundação" para todas as portas, menos a porta por onde ele chegou.

**Fig. 3. Fluxograma de funcionamento do Switch**

Visto que o programa roda á nível de usuário, e não como um *device driver* do Linux, todas as funções da biblioteca LibC estão disponíveis, assim como outras bibliotecas, como a Libpthread, também muito importante para o desenvolvimento.

Na intenção de deixar clara a implementação do comutador, dividimos o mesmo em algumas atividades principais distintas: Inicialização, Manipulação da tabela SAT, Recepção, Aprendizagem, Comutação, Transmissão e Acesso ao Sistema.

Quando o programa é **inicializado**, este inicia uma *thread* para cada interface de rede do switch. Além destas *n threads*, também é disparada uma que trata do gerenciamento da tabela SAT, sendo sua principal função a de apagar as entradas que estão inativas, sem enviar pacotes, a um tempo pré-determinado. Este mecanismo denomina-se envelhecimento.

Para a **manipulação da tabela SAT** de endereços MAC é criado um *array* de estruturas alocado estaticamente, que contém: um *array* de bytes que representa um endereço Ethernet (também conhecido como endereço MAC), um byte identificando a qual das portas do switch este endereço pertence, um byte identificando se esta entrada da tabela esta ocupada ou vaga e, por fim, uma *int* que informa o tempo decorrido desde o último pacote enviado por este endereço. Através da consulta à tabela SAT é tomada à decisão se um pacote será descartado, enviado a apenas uma das portas, ou enviados a todas as portas por *broadcast*.

Quanto à **recepção** de quadros, quando a *thread* responsável pela captura de pacotes de uma determinada interface é iniciada, esta trata de inicializar o descritor de captura da biblioteca através da função *pcap\_open\_live()*. A seguir, com o uso da função *pcap\_loop()* indicamos à biblioteca, qual função deve ser chamada cada vez que um pacote Ethernet for recebido ou enviado pela interface em questão. Tal função, que é escrita pelo usuário, quando chamada, recebe como parâmetros, dentre outros, um ponteiro para uma estrutura de dados com informações sobre o pacote capturado (*struct pcap\_pkthdr \*header*).

O código fonte original da LibPcap, não provê informação ao usuário sobre distinção se o pacote capturado foi recebido ou enviado pela interface. Na sua implementação para Linux, é possível agregar tal funcionalidade através da aplicação de um *patch* de modificações simples. Porém, em outras plataformas esta alteração não é tão trivial. Com a intenção de a tornar o comutador portátil a outras plataformas, foi decidido por contornar esta limitação da LibPcap de uma forma diferente. Este mecanismo é detalhado a seguir, junto do processo de aprendizagem.

Para a **auto-aprendizagem** da tabela, quando um pacote é recebido, primeiro sua consistência é verificada (endereço origem multicast ou broadcast é descartado). Após, parte-se para análise do endereço de origem, que consiste na aprendizagem. Nesta etapa, basicamente o que é feito consiste em varrer-se a tabela de endereços à procura do endereço de origem, se o endereço existir, é atualizado o contador de envelhecimento, se não, adiciona-o na tabela como sendo pertencente à porta que chegou. Dada a limitação da LibPcap descrito acima, a processo de aprendizagem agregou uma funcionalidade a mais que corrige este problema, detectando se o endereço de origem consta na tabela, mas está associado à outra porta. Isto significa que a *thread* responsável pela mesma é que está enviando o pacote, ou seja, o pacote na verdade foi transmitido pela interface, portanto, deve ser ignorado. Para que este mecanismo

funcione, é necessário que apenas uma *thread* esteja manipulando a tabela de endereços por vez. Para garantir esta premissa, foi necessário o uso de *mutexes* para a proteção do acesso à tabela.

O processo de **comutação** consiste em decidir a qual interface o pacote deve ser enviado. Para tal, uma consulta à tabela de endereços, desta vez à procura do endereço de destino, deve ser executada. Caso o endereço de destino seja do tipo broadcast ou multicast, não se faz necessária a consulta à tabela, pois o pacote deve ser enviado a todas as portas, semelhante ao caso do endereço destino não existir na tabela. Se o endereço existe na tabela, e está associado a uma outra porta, neste caso, encaminha o pacote apenas para a porta ao qual o endereço está associado. Finalmente, se o endereço de destino existe, mas está associado a própria porta, neste caso, não faz nada.

Para a **transmissão**, cada vez que um pacote vai ser enviado, uma sessão de envio da LibNet é criada através da função *libnet\_init()*. Esta biblioteca permite variados tipos de injeção de pacotes na rede, mas visto que apenas queremos replicar o pacote recebido sem manipulação, utilizamos o tipo de injeção *LIBNET\_LINK\_ADV*. Depois de iniciada a sessão, a função *libnet\_adv\_write\_link()* é chamada, para o envio do pacote. Esta função recebe como parâmetros, dentre outros, um ponteiro para o pacote a ser enviado. Cada sessão de envio de pacote permite que apenas uma porta de destino seja especificada: a porta de destino é também um argumento da função *libnet\_init()*. Ou seja, quando o pacote deve ser enviado a mais de uma interface de rede, várias sessões devem ser criadas, uma para cada interface de destino.

O **acesso ao sistema** ocorre via linha de comando local ou remotamente por terminal virtual (*Telnet*). Ao ser executado, podem ser informados, via parâmetros, o número de interfaces de rede e o tempo de expiração do registro na tabela; se tais informações forem omitidas, o sistema rodará com os parâmetros padrões (duas interfaces de rede e 120 segundos para expiração). O sistema rodando exibe a tabela de comutação com suas informações atualizadas: identificador do registro, endereço MAC, o nome da interface associada e o tempo que o registro está na tabela. Para acesso de múltiplos usuários (alunos) em laboratórios (aulas), é possível acesso remoto, via um cliente *Telnet* padrão ao endereço IP do computador onde roda o software comutador.

#### 4. Conclusão

A utilização deste trabalho, como ferramenta auxiliar, por parte dos envolvidos no processo ensino/aprendizagem, traz vantagens consideráveis. A prática torna mais concreta a assimilação dos conceitos estudados, o que acarreta em uma abstração mais objetiva.

Uma das dificuldades encontradas, diz respeito à utilização das técnicas e ferramentas de programação (incluindo pesquisa das bibliotecas Libpcap e Libnet), recursos do sistema operacional (como tratamento de acessos concorrentes) e conceitos da tecnologia Ethernet (especificação de cabeçalhos, comutação, aprendizagem e montagem da tabela SAT, etc).

A fim de preencher lacunas deixadas em branco, funcionalidades que não foram implementadas terão especial atenção em uma próxima edição. Pode-se citar VLANs e filtragem de pacotes. Tais adições poderão acrescentar substanciais conteúdos.

## Referências

- [1] Spurgeon, Charles E. *Ethernet: The Definitive Guide*. O'Reilly, 2000.
- [2] Kurose, James F, Ross, Keith W. *Computer Networking: A Top-Down Approach Featuring the Internet* (2 ed) Pearson Addison Wesley, 2002.
- [3] The IEEE 802.1 Working Group, <http://grouper.ieee.org/groups/802/1/>
- [4] Tanenbaum, Andrew S. *Computer Networks* (Fourth Edition). Prentice Hall, 2003.
- [5] Peterson, Larry, Davie, Bruce. *Computer Networks: A Systems Approach* (3rd Ed). Morgan Kaufmann, 2003.
- [6] Stevens, W. Richard. *UNIX Network Programming* (2nd ed). Prentice Hall, 1998.
- [7] Stallings, W. *Data and Computer Communications* (6th ed). Prentice Hall, 1999.
- [8] Perlman, R. *Interconnection: bridges, routers, switch, and internetworking protocols* (2th edition). Addison Wesley Professional Computing Series, 1999.
- [9] *Designing Switched LAN Internetworks*, 1998,  
<http://www.cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2012.htm>
- [10] *LibpCap* - TCPDUMP Public Directory, <http://www.tcpdump.org/>
- [11] *LibNet Packet Construction Lib*, <http://www.packetfactory.net/libnet>
- [12] *Switching vs. Routing*. Network Magazine,  
<http://www.networkmagazine.com/article/NMG20000724S0038>

**Abstract.** *This article presents a construction proposal of an educational tool, specific for the teaching of computer networks. It's main purpose is to demonstrate the operation of packet switching in Ethernet 802.3 networks. It is a public domain tool with open source code, addressed to UNIX systems. Its use is facing a real environment of computers with multiple network interfaces where the user can determine the number of interfaces that will take part in the switching mechanism and also to observe the learning process of the address table. This is its dynamics, being driven by the user through keyboard instructions in a command line interface.*

**Key-words.** *Packet switching, bridge, Ethernet, learning tool*