Universidade Estadual de Campinas

Faculdade de Engenharia Elétrica e de Computação

Um Protocolo Multicast Bi-direcional para Educação a Distância

Autor: Edison de Queiroz Albuquerque

Orientador: Prof. Dr. Hugo Enrique Hernández Figueroa

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Telecomunicações e Telemática**.

Banca Examinadora

Hugo Enrique Hernandez Figueroa, Ph. D. ..... DMO/FEEC/Unicamp
Rafael Dueire Lins, Ph. D. ..... DES/UFPE
André Francheschi de Angelis, Ph. D. ..... CESET/Unicamp
Rui Fragassi Souza, Ph. D. ..... FEEC/Unicamp
Aldário Chrestani Bordonalli, Ph. D. .....FEEC/Unicamp

Campinas, SP

Setembro/2005

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

AL15p	Albuquerque, Edison de Queiroz Um protocolo multicast bi-direcional para educação a distância / Edison de Queiroz Albuquerque. --Campinas, SP: [s.n.], 2005. Orientador: Hugo Enrique Hernández Figueroa. Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 1. Intranet (Redes de computação). 2. Redes de computação – Protocolos. 3. TCP/IP (Protocolo de rede de computação. I. Figueroa, Hugo Enrique Hernández. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Titulo em Inglês: A bidirectional multicast protocol for distance learning.
Palavras-chave em Inglês: Intranets (Computer networks), Computer network protocols, TCP/IP (Computer network protocol).
Área de concentração: Telecomunicações e Telemática.
Titulação: Doutor em Engenharia Elétrica
Banca examinadora: Rafael Dueire Lins, André Franceschi de Angelis, Rui Fragassi Souza e Aldário Chrestani Bordonalli.
Data da defesa: 8/9/2005

AGRADECIMENTOS

Em primeiro lugar quero agradecer a DEUS por ter colocado o Doutorado em meu caminho, um velho sonho que eu pensava não mais poder realizar.

Em seguida meus sinceros agradecimentos ao meu orientador, Prof. Dr. Hugo Figueroa, por me aceitar como orientando e por me incentivar nos momentos em que o peso de fazer um Doutorado longe de casa, ao mesmo tempo que trabalhava na EMBRATEL e tinha meus compromissos familiares, fazia-me fraquejar.

Meu profundo reconhecimento ao Prof. Dr. Shusaburo Motoyama, pela inestimável colaboração com a qual me honrou.

A Fabiano Bizinelli, M.Sc., meu reconhecimento pelo suporte que tantas vezes me deu nas Linguagens de Programação C++ e Otcl, o que abreviou sobremaneira o tempo de elaboração da simulação.

Não posso deixar de agradecer à EMBRATEL, que me liberou tantas vezes para viajar de Curitiba a Campinas, de modo a assistir às aulas. Particularmente à pessoa do meu, então Diretor, o Engenheiro Breno Bina Kessler.

A Alexandre de Souza Falcão, por ter colocado sua vasta experiência em instalação e gerência de Redes IP (entre outras), pela leitura deste trabalho, pela troca profícua de idéias e pelos testes que realizou, validando meus experimentos iniciais, meu muito obrigado!

Agradeço ao Prof. Dr. Eduardo Parente Ribeiro, da UFPR, pelas vezes em que me cedeu seu tempo, ocasião em que me deu sugestões valiosas.

Aos membros da Banca Examinadora, por me terem cedido seu tempo que, tenho certeza, anda escasso. Muito Obrigado.

Quero agradecer, do fundo do meu coração, o apoio da minha esposa, Targélia, que pacientemente saiu com as crianças tantas vezes para que eu pudesse ficar em casa estudando e trabalhando no texto e na simulação, pesquisando e instalando os *softwares* necessários, além das várias férias não usufruídas. A meus filhos, Natália (agora com 18 anos) e Lucas (agora com 9 anos), que ficaram sem a presença do pai e que, mesmo assim, sempre me encorajaram a continuar nestes quase 5 anos de esforço.

RESUMO

Este trabalho está baseado em uma experiência real, vivida no dia-a-dia do nosso trabalho na Empresa Brasileira de Telecomunicações S.A, EMBRATEL, quando da busca de ferramentas capazes de viabilizar o permanente treinamento dos Consultores Técnicos da empresa, localizados em diferentes partes do nosso país, em face da escassez de recursos financeiros e de tempo.

Para tanto, iniciamos com uma retrospectiva histórica da Educação a Distância no mundo, seu presente e suas tendências visíveis. Depois, apresentamos uma descrição dos protocolos multicasting existentes, seguido de nossa proposta.

Apresentamos, também, uma simulação usando o Simulador ns-2 de maneira a comprovar o correto funcionamento do protocolo proposto e avaliar seu desempenho.

O núcleo desta tese é a proposta de um novo protocolo multicasting, elemento essencial para a otimização da rede de suporte, um imperativo para a eficiência de seu funcionamento e, principalmente, da diminuição de seu custo, tendo em vista a situação ideal de se transmitir voz e vídeo para garantir que uma aula a distância será o mais próximo possível de uma aula presencial, conforme aponta o resultado de pesquisas referenciadas no presente texto, uma vez que vídeo é uma aplicação que demanda grande largura de banda.

ABSTRACT

This work is based on a real life situation, experienced in our daily work at Empresa Brasileira de Telecomunicações S.A, EMBRATEL, when pursuing tools that would allow the continuous training of its staff of Technical Consultants. The Technical Consultants were spread all over Brasil, a country of continental dimensions. Time and cost were obstacles to achieve the above mentioned training.

This thesis starts providing a short account of the historical milestones on Distance Learning. We added considerations on the state-of-the-art and scenarios of future developments, which were drawn taking into account social and economical policies.

Routing and existing multicasting protocols are described and compared with the bi-directional multicast protocol proposed in this thesis.

The main purpose of this thesis is the proposal of a new Multicasting Protocol, to support a new concept on which a teacher can be a student on a different subject by taking into account that we aim a Distance Learning environment inside a corporation.

The ns-2 Simulator is used to simulate the perfect operation of the proposed protocol and evaluate its performance.

The use of video is mandatory in order to achieve a maximum efficiency at class. The rapid switching of video transmission from the teacher to a remote student that asks a question (so that all other students spread around the country can also see who is asking) and back to the teacher is mandatory and is what motivated the development of this new Multicast Protocol.

SUMÁRIO

Lista de Figuras

Capítulo 1

Capítulo 3

Capítulo 4

Trabalhos Publicados pelo Autor

1. E.Q. Albuquerque. “Testes de conformidade em protocolos modelos OSI”. *7º Simpósio Brasileiro de Redes de Computadores,* Porto Alegre, Rio Grande do Sul, Brasil, Março 1989.

2. E.Q. Albuquerque. “Approximate Solution of Closed Queueing Networks Using Mathematical Formulae”. *SBT / IEEE International Telecommunications Symposium* (ITS´ 90), Rio de Janeiro, Rio de Janeiro, Brasil, Setembro 1990.

3. E.Q. Albuquerque. “Multi-redes Digitais”. *I Seminário de Estudos e Perspectivas das Telecomunicações em Pernambuco*, Recife, Pernambuco, Brasil, Março 1994.

4. E.Q. Albuquerque. “ Communication Technologies for Distance Learning”. *International Conference on Engineering and Computer Education* (ICECE2000), São Paulo, São Paulo, Brasil, Agosto 2000.

5. E.Q. Albuquerque, H.E. Hernandez-Figueroa, S. Motoyama. “A Switched Multicast Protocol for Intranets”. *International Workshop on Telecommunications* (IWT2004), Santa Rita do Sapucaí, Minas Gerais, Brasil, Agosto 2004.

6. E.Q. Albuquerque, H.E. Hernandez-Figueroa, S. Motoyama. “A bi-directional Multicast Protocol”. *5th. Conference on Telecommunications* (CONFTELE2005), Tomar, Portugal, Abril 2005.

INTRODUÇÃO

O término do regime comunista soviético representou o fim da força que se opunha ao capitalismo, resultando na expansão deste último. Além disto, a globalização abriu as fronteiras, introduzindo a competição desenfreada e desigual com os países desenvolvidos. Isto obrigou as empresas a mudarem a equação **Preço = Custo + Lucro** para a nova equação **Lucro = Preço – Custo** que, exigindo a diminuição dos Custos para maximizar os Lucros, levou ao enxugamento dos quadros de pessoal (entre outras medidas), movimento mais conhecido pelos nomes de *downsizing* e "reengenharia". Esta situação coloca aos demais funcionários uma grande carga que os impede de se afastar por tempos prolongados do seu posto de trabalho para treinamento, impondo a necessidade de novas maneiras de se conduzir a capacitação do corpo funcional.

Atualmente, as empresas têm se defrontado com a necessidade, sempre crescente, de capacitar seus funcionários de maneira contínua [1][2]. O antigo modelo de capacitação nas empresas, de permitir de dois a três treinamentos por ano, com durações de uma semana até seis meses cada um, acrescido dos custos (que se quer minimizar) de passagem aérea e hospedagem, não é mais viável, dado que as empresas têm diminuído o seu orçamento para esta finalidade e efetuado cortes sucessivos em orçamentos já aprovados.

Em várias áreas do conhecimento, a demanda por capacitação aumenta vertiginosamente em virtude da velocidade com que a inovação tecnológica se impõe, o que conduz à impossibilidade de parar os programas de capacitação (pois, hoje, o capital intelectual apresenta-

se como o maior patrimônio de qualquer empresa e constitui seu diferencial competitivo). Faz-se, então, imprescindível a utilização das ferramentas propiciadas pelas tecnologias de telecomunicações e informática (TI) para substituir e/ou complementar os cursos presenciais tradicionais existentes.

Dentro desse cenário, nas empresas, há, por um lado a necessidade de capacitação intensa e continuada do corpo técnico, e, por outro, as restrições orçamentárias e de disponibilidade de tempo do pessoal. Por um lado, há o desejo de dotar a empresa de um grupo de excelência que seja um diferencial importante nas licitações, de modo a vencer a concorrência (feroz) e, por outro lado, a necessidade de maximizar o lucro diminuindo o quadro de funcionários e cortando gastos considerados “menos importantes”, dentre os quais, o treinamento de qualidade.

A solução é lançar mão das modernas tecnologias de informática e telecomunicações para ministrar treinamentos à distância e assíncronos, uma vez que os funcionários têm compromissos com os clientes e não há como reservarem sempre o mesmo horário por vários dias para assistir aulas, mesmo que a distância.

Ora, na EMBRATEL há um corpo de técnicos altamente capacitados que podem transmitir para os colegas seus conhecimentos específicos (estes trabalham, principalmente, no Rio e em São Paulo). Além do mais, a empresa é proprietária de uma excelente plataforma de comunicações, não precisando contratar com terceiros. Logo, o uso de EaD (Educação a Distância) é o caminho natural para resolver o problema.

Da perspectiva do ensino tradicional (curso de graduação e pós-graduação) [3][4], existe uma enorme demanda não atendida de alunos em cidades do interior que não têm possibilidade de se mudar para as metrópoles (onde estão as melhores faculdades e universidades) devido ao alto custo imposto aos pais, tais como moradia, alimentação, etc.. Se eles tivessem em suas cidades programas de EaD, com salas de aula atendidas por sistemas de comunicação terrestre ou via satélite, poderiam, por exemplo, fazer um curso de Direito, juntando-se à população de estudantes de nível superior, uma vez que só teriam que pagar a mensalidade e os livros.

Em qualquer caso, a mudança de atitude de alunos e professores é fundamental para o sucesso dessa modalidade de ensino [5][6][7][8]. O aluno passa de um passivo receptor de conteúdo a um ativo pesquisador, crítico e condutor de seu próprio aprendizado (sob a supervisão e orientação do professor). O professor tem que abandonar a postura de repositório do saber, senhor do destino dos alunos, infalível, para a de um facilitador e provocador dos alunos a estudar e elaborar seus próprios raciocínios, mais voltado para tirar suas próprias conclusões do que, meramente, decorando a matéria.[1][4][9].

A importância do vídeo durante as aulas, de modo que professores e alunos possam se ver (aumentando a interatividade), exige que os meios de comunicação tenham uma largura de banda apropriada para tal finalidade, hoje não menor que 128 kbps. Felizmente, as redes ópticas vêm ao encontro dessa necessidade (mais banda por menos custo) se apresentando, num futuro não muito distante, como o meio preferencial de interligação dos computadores das empresas, e mesmo domésticos, a preços acessíveis. Complementando esta rede óptica, temos as tecnologias

wireless dando mobilidade ao aluno e ao professor, de modo que se possa estar aprendendo, ensinando em qualquer lugar e a qualquer momento. Dentre essas tecnologias, sobressaem-se o BlueTooth, para distâncias de até 10 metros e o IEEE 802.11x, também conhecido como WLAN (*Wireless* LAN), para distâncias maiores (até 300 metros), além de satélite, etc.

Por um lado, pesquisadores têm se esforçado por desenvolver dispositivos que ofereçam banda larga a preços cada vez menores. Por outro lado há um esforço de se comprimir os sinais, principalmente de vídeo e áudio, de modo a se poder transmiti-los através de meios com largura de banda cada vez menores. A convergência destes dois trabalhos vai acelerar a chegada ao ponto em que se possa transmitir os sinais de vídeo/áudio, necessários à EaD, com qualidade e preços acessíveis a uma parcela bem maior da população.

O uso de meios multimídia de apresentação da aula e a perda do medo de lidar com o computador são habilidades as quais professores e alunos devem se apropriar (bem como qualquer elemento da moderna sociedade)[10][11]. Neste particular os estudantes de cursos técnicos, tais como Engenharia, Física, etc., poderiam se beneficiar por já terem intimidade com o uso dessas ferramentas[12].

Não se pode falar em um sistema eficiente de Educação a Distância (EaD) sem as necessárias considerações sobre capacidade da rede instalada (ou a instalar). Desta maneira, é indispensável o entendimento dos mecanismos presentes em uma rede de dados, predominantemente IP, e o dimensionamento destas redes, de modo a se obter a melhor relação custo/benefício para uma implementação sólida de EaD, seja em uma corporação ou de caráter público. Estes métodos devem também predizer o comportamento da rede estudada em caso de

aumento de carga (uma vez que, normalmente, as redes são usadas para aplicações outras que não só EaD). Esta atividade é conhecida como *Capacity Planning* [13]. Outra maneira de se predizer o desempenho de uma rede é através do uso de simuladores.

Um país com as características geográficas e sociais como o Brasil revela grandes áreas não atendidas por meios de telecomunicações de alta qualidade e confiabilidade. Nestes casos é preciso lançar mão da solução via satélite para alcançar as populações mais distantes dos grandes centros.

Antes de apresentar a maneira como está organizado este trabalho, vale salientar que uma plataforma de EaD requer toda uma infra-estrutura necessária para que se viabilize um curso adequadamente [14], além do protocolo, na qual se destacam:

- **as salas do professor e dos alunos, sua dimensão, iluminação, condicionamento de ar, instalação elétrica, tratamento acústico, etc.**
- **os periféricos usados em ambos tipos de salas, tais como câmeras de vídeo, monitores ou projetores, microfones, alto-falantes, quadros-brancos, videocassete *players*, micros, notebooks, controladores, mesas misturadoras de vídeo e áudio, etc.**
- **o meio de transmissão adequado ao uso pretendido, ou seja, RDSI, Internet, circuitos determinísticos alugados, Frame Relay, ATM, TV a cabo [15], etc.**
- **o software necessário para autoria das aulas, protocolos de comunicação, *groupware*, administração das tele-aulas, *browsers*, sistemas operacionais, agentes**

de ensino (uso de inteligência artificial, IA, para personalizar o conteúdo ao aluno [16]), etc.

- **melhores práticas pedagógicas/andragógicas específicas para EaD.**
- **métodos para determinação do custo/benefício, tais como simulação, algoritmos de dimensionamento da rede necessária, levantamento dos gastos fixos e variáveis, avaliação do resultado do treinamento à distância tais como quantidade de alunos treinados, notas obtidas, questionários de satisfação dos alunos com os resultados alcançados, quantidade de horas que os alunos estiveram conectados à plataforma de EaD, etc.**

Esse trabalho está estruturado como segue:

O **Capítulo 1** inicia com a descrição de uma experiência vivida na EMBRATEL, que visava a suprir uma necessidade concreta onde nos defrontamos com o desafio de manter a equipe técnica atualizada ao mesmo tempo em que reduzíamos os custos de treinamento e a ausência do funcionário do seu posto de trabalho. Neste capítulo relatamos o problema encontrado, uma síntese da solução adotada e a *feature* que percebemos faltar para incrementar a interação entre alunos e professor aproximando, desta forma, o ambiente EaD de uma sala de aula convencional.

O **Capítulo 2** faz uma relação dos fatores que afetam o processo de Ensino/Aprendizagem e relata as novas tentativas de uso dos meios eletrônicos na prática pedagógica/comunicacional.

O **Capítulo 3** introduz o conceito de Multicasting, bem como fornece uma visão dos protocolos já existentes que implementam esta facilidade.

O **Capítulo 4** apresenta a nossa proposta, que chamamos de XMP (sigla derivada da expressão *Switched Multicast Protocol*). Aqui detalhamos o funcionamento do protocolo proposto e apresentamos os passos do seu desenvolvimento no simulador ns-2. Neste capítulo, também são apresentados os resultados obtidos da verificação, tanto do ponto de vista de funcionalidade como de carga de tráfego na rede, valores de utilização de banda, retardo, etc.

O XMP visa prover as sessões multicast com a facilidade da bidirecionalidade, objetivando aumentar a interação aluno-professor e aluno-aluno. Assim sendo, o ambiente virtual da EaD se aproximaria de uma situação real de sala de aula presencial, beneficiando o processo de Ensino/Aprendizagem.

Nossa proposta leva em conta as dificuldades financeiras das instituições (este é o dia-a-dia na EMBRATEL junto aos clientes), de modo que o protocolo foi concebido para causar o menor impacto possível na plataforma IP das empresas, tanto do ponto de vista de meios de comunicação (economizando banda) como do custo dos equipamentos envolvidos, notadamente os roteadores da rede corporativa.

CAPÍTULO 1

UMA EXPERIÊNCIA de EaD CORPORATIVA: da PRÁTICA à REFLEXÃO

No decorrer da minha vida profissional na EMBRATEL vivi dois momentos bem distintos. A EMBRATEL empresa estatal e a privatizada.

Durante o tempo de estatal esta empresa chegou a ter 12 engenheiros encarregados de elaborar os treinamentos internos, tanto no que tange ao conteúdo das matérias, quanto à elaboração das apostilas. Nossas apostilas, para uso interno, eram tão boas que, não raro, foram publicados como livros por conceituadas editoras multinacionais, tendo sido adotados em várias universidades brasileiras. Na EMBRATEL privada, esta estrutura foi desmontada e, nos cursos ministrados, eram fornecidas apostilas que continham tão somente a cópia das transparências mostradas.

1 – Nesse ambiente privado, eu era responsável por uma equipe de 12 Consultores Técnicos espalhados de Brasília a Porto Alegre, que era encarregada de elaborar soluções de telecomunicações para os clientes classificados como *Top Account*, ou seja, clientes que traziam grandes receitas para a empresa. A área de atuação do meu grupo está identificada na Figura 1, pelas cores amarela e laranja. O consultor técnico é um engenheiro responsável por entender os problemas de telecomunicações do cliente, projetar e precificar a solução mais adequada. O posto de trabalho de consultor técnico foi criado antes da privatização, quando a empresa entendeu que devia deixar de ser uma empresa técnica para uma empresa prestadora de serviços

à comunidade. Eles estão lotados em grandes cidades e trabalham junto aos vendedores, dando-lhes suporte para aumentar suas chances de êxito para convencer o cliente a comprar os produtos ofertados.

Figura 1 – Área de atuação de nossa equipe de CTs (amarelo e laranja).

Premido por restrições orçamentárias para gastar com treinamento, era meu dever (entre outros) capacitar os consultores nos mais variados serviços de telecomunicações de voz, dados e imagem, tanto via meios terrestres como via satélite.

A competitividade da empresa no mercado exige que a capacitação deve ser continuada, uma vez que a TI (tecnologia da informação) é extremamente dinâmica, com o surgimento de novas modalidades e protocolos de comunicação com grande velocidade, e os manter

capacitados passa a ser um elemento chave.

A EMBRATEL já possuía uma estrutura de EaD através do uso de um pacote comprado, o UNIVERSITE (http://www.mhw.com.br/universite/). No entanto, seu uso ficava limitado a treinamentos de inglês e outros voltados para a área de vendas e administrativas, sendo seu uso na área técnica praticamente inexistente. Esse pacote oferecia acesso ao *site* do treinamento, tanto de dentro da empresa quanto de fora, visando que os funcionários pudessem acessar os cursos de suas casas à noite ou em finais de semana. O acesso de casa apresentava dificuldades, pois a maioria dos funcionários, à época, só possuía conexão discada, o que tornava extremamente cansativo o seu uso pelo tempo que demorava para carregar as páginas.

Por outro lado, com o enxugamento do quadro de funcionários, após a privatização, já não podíamos dispensar o consultor para treinamentos fora do seu local de trabalho e em tempo integral, como ocorria no tempo de estatal, com funcionários se ausentando para treinamento por períodos de até 9 meses ou mesmo 1 ano.

A solução para esta capacitação, com menores custos se tornou uma meta de alta prioridade. A falta de verbas e a dificuldade de dispensar o consultor do seu posto de trabalho não podiam impedir que o mesmo recebesse a capacitação necessária para o exercício de suas funções de maneira satisfatória.

Tendo a empresa uma rede corporativa de telecomunicações de grande porte, o pensamento natural foi de aproveitar esta infra-estrutura para suportar uma plataforma de EaD de maneira que o consultor pudesse receber sua capacitação sem sair, sequer, de sua mesa. Assim, poderia ter 1 hora de aula por dia e, fora deste horário, dar continuidade aos projetos para os

clientes. Semelhantemente, buscava-se que o professor, de dentro ou de fora da empresa, pudesse ministrar sua aula de onde quer que estivesse, mantendo suas atividades normais fora deste período.

O pressuposto básico é que o professor de uma determinada matéria pode, para um outro assunto, ser o aluno e, vice-versa, o aluno de uma determinada disciplina vir a ser o professor em alguma outra. Isto de fato acontece na empresa.

Procurei o auxílio do departamento de TI da empresa para implementar uma plataforma de vídeo, onde todos os alunos pudessem ver o professor. Imediatamente fomos barrados pela equipe encarregada da rede corporativa, uma vez que o uso de vídeo demanda muita banda e isto iria congestionar os meios de comunicação e os equipamentos da rede da empresa. Impedidos de usar vídeo partimos para uma solução só com áudio (além de material escrito), ou seja, os alunos iriam ouvir a voz do professor explicando o conteúdo. Naturalmente, as máquinas dos alunos deveriam estar equipadas com placa de som e alto-falantes. Esta foi uma situação fácil de se conseguir, uma vez que não se compra mais PCs sem a placa de som, ainda que seja *on-board*. Ao invés de alto-falantes é preferível usar fones de ouvido porque como, normalmente, os consultores trabalham em uma mesma sala sem divisórias, o som proveniente dos alto-falantes atrapalha os colegas vizinhos de mesa que não estão participando do curso. Além disto, há um problema inerente da tecnologia que faz com que o som não saia nos alto-falantes de todos ao mesmo tempo, uma vez que o processador central pode estar ocupado com outras tarefas e entrega o som no alto-falante frações de segundo depois, ou antes, em máquinas distintas,

provocando um efeito desagradável.

Realizei uma pesquisa entre os consultores para saber em que tema(s) eles sentiam ter conhecimentos insuficientes e desejosos de ampliar suas competências. Esta pesquisa consistiu em um questionário, via e-mail, que enviei para os consultores sob minha responsabilidade onde eles podiam assinalar, para cada tópico, o grau de conhecimento que tinham tais como “bom”, “razoavel”, “deficiente” e “não conheço”. Alguns (em torno de 10%) tinham domínio da parte de satélite mas careciam de conhecimentos na parte de centrais telefônicas e PABX. Outros (em torno de 40%) dominavam as transmissões via rádio, mas precisavam de treinamento na parte de comunicações de dados, principalmente sistemas baseados no protocolo IP, da Internet [17].

Em seguida fomos atrás de especialistas em cada uma destas matérias, funcionários ou não da empresa, que pudessem colaborar. Montamos um calendário de treinamento e, antecipadamente, mandávamos e-mail para os consultores listados onde indicávamos o dia e hora do treinamento, orientações para instalar o programa necessário, telefone e e-mail de uma pessoa encarregada de tirar dúvidas, a apostila do curso (que podia conter a foto do professor) com a orientação de que eles deveriam imprimir a mesma em suas unidades (isto diminuía o custo de confecção centralizada de apostilas uma vez que usavam a impressora, cartucho de tinta e papel do material de escritório de cada aluno), estudar antes da aula e usar estas apostilas durante a aula para que pudessem acompanhar e fazer anotações. Além disto, eles eram orientados a entrar no *site* da disciplina 15 minutos antes do previsto, quando deveriam ouvir uma música qualquer. Caso não ouvissem a música deveriam entrar em contato imediatamente com a pessoa indicada

para o suporte técnico de modo a sanarem os problemas em tempo. As perguntas deveriam ser encaminhadas ao professor pelo próprio correio eletrônico da empresa. Nas aulas-teste ministradas, só um colega em uma das aulas não entrou conforme orientado, pois estava acontecendo uma reforma na rede local da sua sala.

O computador usado pelo professor nas aulas-teste foi um Notebook Extensa 390 com 166MHz de *clock*, da Texas Instrument (Acer), com 32M de RAM e um disco rígido de 2G. O sistema operacional foi o Windows 98. Esta máquina transmitia a fala do professor, além de receber os e-mails com as perguntas dos alunos. Apesar dos poucos recursos deste Notebook, não houve problema. Os alunos usavam seus próprios PCs (normalmente Pentium rodando Windows 95 ou 98). Só posteriormente, a EMBRATEL alienou todos os PCs com menos de 200Mhz de *clock*. Todos os micros possuiam placa de som (ainda que fosse *on board*) e alto-falantes, o que era bastante comum na época.

Os consultores inscritos para o treinamento recebiam um e-mail, como dito anteriormente, onde constavam as orientações necessárias, um arquivo anexo com a apresentação para ser impressa (normalmente Powerpoint feito sem muitos recursos de cores para que pudessem ser impressos sem perder a legibilidade) e um arquivo anexo com o Winamp que deveria ser instalado na máquina do aluno com um simples duplo clique em seu arquivo executável, que se auto instalava.

O professor recebia, além do Winamp, o Shoutcast (ambos podem ser descarregados do *site* http://www.winamp.com). Este, na forma de um arquivo executável, auto instalável, de apenas 400kbytes, evitava obrigar que o professor necessitasse de uma máquina grande e cara

para ministrar sua aula (o que acontecia com o programa da Real pois testes realizados na EMBRATEL mostraram que mesmo a versão mais leve, o Real Basic, rodou com dificuldade na transmissão de vídeo). A plataforma usada para os testes era composta de 2 PCs Celeron com 500M de *clock*, 64 Mega de RAM e um HD de 10G, rodando Windows 98. Estes computadores estavam interligados por um barramento Ethernet de 10Mbps.

Vários programas foram pesquisados, como por exemplo o Real (server e client), o Lipstream, o Hearme e o Shoutcast (server, http://www.shoutcast.com) com o WinAmp (client, http://www.winamp.com). O Real (a versão Basic era gratuita), se mostrou pesado, demandando máquinas mais possantes, o que não poderia ser garantido na maioria dos casos. O Lipstream e o Hearme não apresentaram um funcionamento convincente além de apresentarem dificuldades na obtenção de literatura sobre seu funcionamento. O escolhido foi o Shoutcast e o WinAmp, este último é o *player* de MP3 mais comum. Ambos são feitos pela Nullsoft, são leves, gratuitos, possuem uma boa literatura sobre sua instalação, configuração e funcionamento e, importante, têm versão para rodar em Linux. Uma vantagem adicional, mas não menos importante, é que este Software (SW) comprime o áudio usando mp3 de comprovada eficiência, uma vez que comprime a banda exigida pelo áudio em até 12 vezes. Em medições realizadas em bancada na empresa, com profissionais da área de TI, não foi possível medir o tráfego adicional gerado por esta aplicação. Isto significa, na prática, que nossa aplicação de EaD não exige aumento de banda da rede corporativa (e banda é dinheiro).

Este estudo despertou o interesse de aprofundamento no uso destas tecnologias para levar ensino/treinamento para pessoas que não podem dedicar todo o seu tempo em aula e nem

possuem os recursos materiais para gastar com transporte e hospedagem. A própria empresa, ao ver o sucesso desta plataforma resolveu (não sei se por coincidência ou não) desenvolver algo semelhante, agora, com o uso de vídeo também. Foi instalado e testado o programa *Microsoft Media Server* na máquina onde o professor iria dar aula e o *Microsoft Media Player,* que já vem instalado normalmente em todas as máquinas, para os alunos. Na realidade foi instalado o *Media Server* em um computador em São Paulo (que apresenta a maior concentração de equipamentos de informática da empresa), e o *Media Encoder* na máquina de onde o professor ministraria a aula. Como, desta vez, a ordem veio de superiores, a banda adicional exigida pelo vídeo não foi mais problema.

Como interessado, participei dos testes (durante o ano 2000) com o aplicativo da Microsoft, apontando falhas e necessidades de reconfiguração do SW e dos equipamentos envolvidos (basicamente roteadores e *switches*). No início, as transmissões eram feitas no modo unicast, o que significa que uma amostra de vídeo/áudio é encapsulada em um pacote IP e transmitida para a máquina de cada aluno. Assim, se tivéssemos 20 alunos, a mesma amostra teria que ser transmitida 20 vezes, ou seja, necessitaria de uma banda 20 vezes maior. A empresa resolveu instalar em todos os seus roteadores um protocolo *multicasting*, escolhendo o PIM-SM (*Protocol Independent Multicasting - Sparse Mode*) por ser o mais atual, na época, e o modo SM o mais apropriado para uma rede de abrangência geográfica nacional, como era o nosso caso.

Com o protocolo Multicasting instalado o que passou a acontecer é que a amostra de vídeo/áudio era transmitida uma única vez e entregue nas máquinas de cada aluno através do uso de endereços IP especiais (este processo será explicado adiante). A limitação do número de

alunos que havia no início (no máximo 25 alunos) não existia mais. Hoje, transmissões usando esta tecnologia permitem que todos os funcionários da empresa (mais de 7.000) possam assistir a um pronunciamento, por exemplo, do presidente, simultaneamente e sem sair de suas mesas.

Há, ainda, uma limitação que precisa ser superada que é o fato de que o professor é visto e ouvido por todos os alunos, mas nenhum aluno é visto em momento algum da aula. Para solucionar tal problema estamos propondo, nesta tese de Doutorado, um protocolo contemplando a facilidade de que a transmissão de vídeo que se inicia no professor pode ser passada para um aluno de modo que esse possa fazer sua intervenção enquanto é visto e ouvido pelo professor e por todos os outros alunos. O aluno que pretende fazer a intervenção solicita, por e-mail ou outro mecanismo, a um monitor (que pode ser o próprio professor) que habilita a máquina do aluno a atuar como transmissora e não mais como receptora, e a máquina do professor como receptora e não mais como transmissora. Ao fim da intervenção o monitor comanda a volta da transmissão ao professor ou a outro aluno que tenha se inscrito para tal.

Na época da elaboração inicial desta tese (no ano de 2001), não tínhamos encontrado em nenhum lugar, referência a um protocolo que possibilitasse isto, apesar de uma extensa pesquisa. No entanto, recentemente, tomamos conhecimento de que há um *draft* (*paper*) na IETF (*Internet Engineering Task Force*, http://www.ietf.org) para ser apreciado pela comunidade de modo a se tornar um padrão (uma RFC ou *Request For Comments*).[1]

Há um outro meio de se conseguir que todos os alunos se vejam, bem como ao professor. É um equipamento chamado de MCU (*Multipoint Control Unit*). Ele pode ser usado tanto em

[1] Em julho de 2005 o site da IETF retirou este trabalho da lista de "drafts" sem convertê-lo para RFC.

redes IP como RDSI (Rede Digital de Serviços Integrados) ou Ponto-a-Ponto, usando a interface V.35 (http://picturephone.com). Neste caso, todos os participantes se conectam à MCU e esta faz a distribuição de todos os vídeos e áudios recebidos, para todos os participantes, de modo que a tela do monitor de todos é dividida em vários quadrados cada um com a imagem de um participante. Este equipamento é muito caro (em torno de US$35.000,00 para um com 32 portas e 4 sessões simultâneas, a preços de 2005) e necessita de um especialista para configurar e operar as transmissões.

CAPÍTULO 2

A EaD como Instrumento de Qualidade Educacional:

Questões para Debate

Historicamente, a EaD não é algo recente. Porém, nos dias de hoje, ganhou impulso, encontrando os meios propícios para sua difusão em larga escala, tanto por imposição da nova Sociedade do Conhecimento (que requer aprendizado contínuo dada a velocidade com que as inovações surgem), quanto pela dificuldade de se ausentar do posto de trabalho ou pelas novas facilidades tecnológicas no campo das telecomunicações e informática que fornecem meios e dispositivos cada vez mais variados e baratos.

Entre vários temas relacionados à questão educacional da EaD, três focos, mesmo não sendo, diretamente, objetos da nossa pesquisa, serão balisadores de muitos momentos desta investigação, pois os consideramos "pano de fundo" de estudos sobre novas tecnologias educacionais.

Consideramos que:

(a) do ponto de vista de quem vai investir em plataformas para EaD, faz-se necessário avaliar a relação custo/benefício do ensino à distância, comparado (ou não) ao ensino convencional e várias pesquisas já foram feitas nesse sentido, porém são não conclusivas [18], tornando-se um campo fértil de investigação;

(b) há, também, o aspecto de que a pedagogia convencional precisa ser revista para propiciar um ensino de qualidade. Neste sentido a questão do currículo e da inovação educacional precisam ocupar o centro dos debates sobre qualidade educativa em EaD. Em decorrência disso, novas técnicas pedagógicas precisam ser elaboradas [8]. Acrescente-se a isto que a retenção do conhecimento tem uma eficiência de 25% se o aluno só ouvir, 45% se ouvir e ver e 70% se ouvir, ver e fizer [19].

(c) É preciso problematizar, ou seja, questionar sobre a eficiência da EaD é questionar sobre a qualidade da participação do aluno no processo de aquisição do conhecimento. Este, como principal sujeito da dinâmica educativa, deve ser co-autor de sua aprendizagem e produtor de novas interações [20].

A grande maioria dos estudiosos de novas tecnologias para fins educacionais apontam para o fato de que a EaD só será eficiente se o sistema puder transferir a maior parte do trabalho para o aluno. E isto, justamente, é o que faz a EaD.

Um exemplo desta prática pode ser constatado em algumas empresas multinacionais que oferecem um bônus periódico para aqueles funcionários que tiverem registrado um mínimo de horas de EaD por mês, variando de 8 a 32 horas, dependendo da empresa e do cargo do funcionário.

Por ser considerado de interesse nacional, cursos em instituições públicas não levam em grande consideração a auto-suficiência financeira, uma vez que os alunos pagam pequenas taxas de inscrição que, nem de longe, cobrem os custos necessários ao seu aprendizado. Fica por conta do Estado.

Nas Universidades particulares, com fins lucrativos, as taxas mensais são elevadas e investimentos em EaD devem proporcionar um aumento da margem de lucro destas instituições.

No treinamento corporativo, a EaD não é medida, de maneira exata, quanto ao seu custo/beneficio. Neste caso os benefícios decorrem de 3 fatores:

1) custo do treinamento, propriamente dito, que é fácil de ser medido se for feito fora da empresa, pois se resume à sua taxa de inscrição. Se for feito dentro da empresa, empregando pessoal e recursos tais como salas, despesas com iluminação, zeladoria, *coffee-break*, etc., fica mais difícil quantificar.

2) custo de passagem aérea e hospedagem para os funcionários de outras cidades.

3) custo relativo ao fato do funcionário não estar no seu posto de trabalho e, portanto, não estar produzindo para a empresa [18].

Quanto ao item 1), a experiência tem demonstrado que o custo de produzir um treinamento à distância tem ficado na mesma ordem de grandeza de um curso convencional, ainda que um mesmo curso possa ficar disponível ou ser repetido várias vezes sem despesas adicionais de elaboração. Os ganhos provêm da eliminação, ou minimização, dos itens 2) e 3).

O item 1) se beneficia da EaD se puder alcançar uma maior audiência, ganhando em escala.

Na impossibilidade prática de se mensurar os benefícios de um curso e apoiado em pesquisas que afirmam serem os cursos à distância tão eficientes quanto os presenciais convencionais (na realidade, estudos apontam que a EaD reduz o tempo necessário de

aprendizado para 2/3 do tempo de um curso convencional porque aumenta o *time-on-task* do aluno que, agora, acessa seu curso a qualquer hora de qualquer dia da semana, incluindo os fins-de-semana), resta focalizar nos custos. Mesmos esses são difíceis de se quantificar, o que nos remete à prática de trabalhar para minimizá-los. Dentro desse espírito, fica clara a importância de se fazer projetos precisos para dimensionar bem os meios de telecomunicações alocados à essa finalidade.

Uma tentativa de quantificar o custo/benefício foi feita em [18], com a criação da figura do *contact time,* que é o tempo que um aluno fica ligado ao meio que lhe permite alcançar o conteúdo. Calcula-se, então, o "Custo por Hora de Contato" que depende do tipo de aula que se dá.

Têm-se notícia de que a MCI WorldCom, por meio do ensino eletrônico, conseguiu treinar 7.000 técnicos que estavam em 800 locais diferentes, economizando US$ 5,6 milhões com deslocamentos [21]. A Caixa Econômica Federal (CEF) concluiu que "os cursos que custavam, em média, R$ 300,00 por profissional passaram a custar R$ 25,00" [21]. A Brasil Telecom economizou R$ 500mil em um treinamento da nova ferramenta SAP implantada, para 2.500 profissionais espalhados do Acre ao Rio Grande do Sul [21]. A TIGRE, fabricante de tubos e conexões, afirma que "o treinamento convencional custa, em média, R$ 24,00 por aluno, enquanto o treinamento on line sai por R$ 2,00". A TIGRE pretende treinar, anualmente, 40.000 profissionais em estações de capacitação instaladas em 600 revendas [22].

A nova dinâmica sócio-econômica impõe novos olhares sobre a questão da Educação. Um deles é a diluição das fronteiras entre disciplinas de conhecimento. Outro é da necessidade de se aprender por toda a vida (*continuous learning*). Não há, também, a distinção clara entre o que é escola e o que não é escola (desaparecimento dos muros, ou escola além das paredes) uma vez que se pode aprender na empresa, em outras organizações e em casa. É interessante constatar que tais concepções já estavam esboçadas por Ivan Illich [23] em 1976. Illich defendia que um bom sistema educacional deve ter três propósitos:

- dar a todos que queiram aprender, acesso aos recursos disponíveis, em qualquer lugar e a qualquer momento (a própria essência de EaD, via Internet);
- capacitar a todos os que queiram partilhar o que sabem a encontrar os que queiram aprender algo deles (a Internet, sabemos, possibilita isto);
- dar oportunidade a todos os que queiram tornar público um assunto a que tenham possibilidade de que seu desafio seja conhecido (outra vez a Internet é o veículo).

Para Illich, o papel das novas instituições de ensino é o de facilitar, ao máximo, o acesso ao conhecimento, para todo aquele que deseja dele se apropriar, seja em que campo for, possa fazê-lo sem distinção de classe social, idade, etc. Os logradouros públicos ofereceriam pontos através dos quais colegas e pessoas mais idosas poderiam acessar conteúdos e encontrar-se com parceiros.

Illich escreveu:

> *"Usarei o termo teia de oportunidades em vez de rede para designar modalidades específicas de acesso a cada um dos quatro conjuntos de recursos. A palavra "rede" é muitas vezes usada erroneamente para designar os canais reservados ao material selecionado por outros para doutrinação, instrução e diversão. Mas também pode ser usada para os serviços telefônicos e postais que são principalmente utilizados pelos indivíduos que desejam enviar mensagens uns aos outros. Oxalá tivéssemos outra palavra com menos conotações de armadilha, menos batida pelo uso corrente e mais sugestiva pelo fato de incluir aspectos legais, organizacionais e técnicos. Não encontrando tal palavra, tentarei redimir a que está disponível, usando-a como sinônimo de teia educacional".*

Teria sido Illich o inspirador do termo Web para designar a grande rede mundial que é a Internet, a World Wide Web (www ou w3)?.

Ele classifica o uso desta "teia" em quatro grandes grupos, como nos mostra o texto abaixo:

> *"Quatro redes: os recursos educacionais são geralmente rotulados de acordo com as metas curriculares dos programas educacionais. Proponho fazer o contrário, rotular quatro diferentes abordagens que permitam ao estudante ter acesso a todo e qualquer recurso educacional que poderá ajudá-lo a definir e obter suas próprias metas:*
>
> ***Serviço de consultas a objetos educacionais**: que facilitem o acesso a coisas ou processos que concorrem para a aprendizagem formal. Algumas coisas podem ser totalmente reservadas para este fim,*

> *armazenadas em bibliotecas, agências de aluguéis, laboratórios e locais de exposição tais como museus e teatros; outras podem estar em uso diário nas fábricas, aeroportos ou fazendas, mas devem estar à disposição dos estudantes, seja durante o trabalho ou nas horas vagas.*
>
> ***Intercâmbio de habilidades**: que permite as pessoas relacionarem suas aptidões, dar as condições mediante as quais estão dispostas a servir de modelo para outras que desejem aprender essas aptidões e o endereço em que podem ser encontradas.*
>
> ***Encontro de colegas**: uma rede de comunicações que possibilite as pessoas descreverem a atividade de aprendiz em que desejam engajar-se, na esperança de encontrar um parceiro para essa pesquisa.*
>
> ***Serviço de consultas a educadores em geral**: que podem ser relacionados num diretório dando o endereço e a autodescrição de profissionais, não profissionais, "free-lancers", juntamente com as condições para ter acesso a seus serviços. Tais educadores, como veremos, podem ser escolhidos por votação ou consultando seus clientes anteriores".*

Ao invés de tentar definir EaD, prefere-se entendê-la como a saída possível para a enorme demanda de ensinar e capacitar milhões de pessoas em um cenário onde não há professores qualificados em número suficiente, onde há poucos recursos financeiros para pagar viagens, hospedagem e prover toda a infra-estrutura necessária para acomodar os alunos (tais como salas de aula, gastos com energia, limpeza, etc.) e não há tempo suficiente para treinar esse contingente, sempre crescente, de indivíduos em busca de conhecimento teórico e prático, conhecimento este que se faz novo mais rápido que podemos apreendê-lo pelos métodos tradicionais.

Torna-se necessário, portanto, discutir a EaD, tomando-se por base a questão da produção e apropriação do conhecimento. Para viabilizar EaD é preciso conhecer a multipla relação entre os vários componentes da dinâmica educacional, entre eles, o processo de ensino e aprendizagem, as políticas de capacitação e treinamento corporativas e a conjuntura político-econômica deste momento histórico. Para tanto ressaltaremos, a seguir, alguns fatores básicos para compreender EaD.

2.1 – Aspectos Didáticos e Pedagógicos da EaD.

A EaD precisa ser compreendida no contexto das transformações sociais, econômicas e políticas em relação ao desenvolvimento das novas tecnologias e avanços na área educacional. Coloca-se aqui a necessidade de analisar alguns elementos pedagógicos que integram e dinamizam a EaD.

Na aula presencial, que é usada como referência, a transmissão do conhecimento ocorre no mesmo lugar, ou seja, professor e alunos estão no mesmo espaço físico. Também ela ocorre ao mesmo tempo, pois o professor está falando e escrevendo (e gesticulando, etc.) ao mesmo tempo que os alunos estão recebendo estes estímulos. E, finalmente, ocorre no mesmo ritmo (o ritmo imposto pelo professor e pelas leis que dispõe sobre conteúdo e carga horária semestral).

O uso da aula presencial, usada como referência na maioria dos trabalhos sobre este assunto, não nos parece adequado. Em nossa opinião, a EaD não é a maneira ideal de transmissão de conhecimento, assim como uma sala de aula com 40 (ou mais) alunos também não o é. O melhor sistema seria o de um professor para cada aluno. Melhor ainda, um professor

que domina o que ensina e tem habilidade para transmitir este conhecimento somado a um aluno que precisa (ou está de alguma forma motivado) do que está sendo ensinado.

Na impossibilidade econômica e prática desta modalidade de ensino, a EaD (fortemente embasada pelo que há de mais moderno em termos de tecnologia) aparece como um meio "sub-ótimo" de ensino/aprendizagem.

Na educação à distância, podemos reproduzir este ambiente e ir mais além. As novas tecnologias de telecomunicações, informática e da eletrônica nos permitem quebrar os paradigmas de "no mesmo lugar", do "ao mesmo tempo" e do "no mesmo ritmo".

A Tabela 2.1, adiante, relaciona as atividades típicas em um curso tradicional, classifica-os pelo tipo de aprendizagem e aponta as tecnologias que as viabilizam em um ambiente EaD.

Atividade	***Tipo de Aprendizagem***	***Tecnologia de Suporte***
Comunicação de Fatos	L, I	Páginas *Web*, *RealVideoAudio*
Discussão/Debate	L, I, E, P-P	*e-mail*, *Conferencing*, *Mailing List*
Solução de problemas	E, A	*Groupware*, *Web*, *Conferencing*
Pensamento Crítico	E	*Conference Call*, *Chat*
Elaboração	A	*Web*, portfolios

Tabela 2.1 – Soluções EaD para atividades de cursos tradicionais

Onde:

L = Literal;

I = Inferencial;

A = Aplicativo;

E = Avaliativo;

P-P = Colega-a-Colega (*peer-to-peer*).

Também aqui percebemos o grande potencial das novas tecnologias no auxílio do aprendizado, seja ele presencial ou à distância. Por exemplo, em [24] encontramos uma plataforma desenvolvida para o ensino da Física onde os autores classificam 4 abordagens:

- livros eletrônicos via páginas *Web* (*courseware*);
- laboratório baseado em microcomputador (já existem ferramentas para fazê-lo à distância [15]);
- simuladores;
- laboratório baseado em vídeos digitais.

2.2 – A Questão da Aquisição do Conhecimento e a EaD.

Usamos, como premissa, a divisão da obtenção do conhecimento através de 4 modalidades. Fica claro que existem modalidades intermediárias, mas preferimos classificá-las em 4 grandes categorias por motivo de clareza e facilidade de raciocínio.

2.2.1 – Disponibilizar literatura previamente selecionada para os alunos de modo que eles possam, sem ajuda de professor ou monitor, obter o conhecimento de que necessitam através de sua própria iniciativa e do seu próprio esforço. Aqui temos a situação de "não no mesmo lugar", "não ao mesmo tempo" e "não no mesmo rítmo". Não há interatividade entre professor e aluno. A ferramenta de TI mais apropriada para produzir e armazenar estes "livros eletrônicos" é a Web através de páginas em html. Eles podem ser encarados como livros tradicionais organizados de forma mais dinâmica e interativa, uma vez que podem lidar com uma maior variedade de recursos tais como imagens, animações, simulações, ambientes virtuais, sons, vídeos, etc [25]. Além disto devem permitir ao leitor imprimir, ou gravar, somente a parte que lhe interessa no momento. Este tipo de "biblioteca" não fecha à noite nem nos feriados e fins de semana.

2.2.2 – Dispor de meios que permitam aos alunos obter o conhecimento de que necessitam via meios tecnológicos eficazes, com o apoio de monitores para tirar dúvidas, aplicar e corrigir testes, entre outros. Aqui temos a situação de "não no mesmo lugar", "não ao mesmo tempo" e "não no mesmo ritmo", apesar de que o fator "tempo" não pode ser deixado completamente solto sob pena de alguns alunos levarem um tempo proibitivo para concluir seu aprendizado. Há uma interatividade limitada entre professor e aluno.

2.2.3 – Dispor de meios, propiciados pela tecnologia, que permitam aos alunos obter o conhecimento de que necessitam em aulas onde vêm e ouvem o professor (ou seja

que nome se dê ao indivíduo que está transmitindo o conhecimento que detém, tais como palestrante, consultor, etc.) em tempo real, mas as perguntas são feitas via fax, e-mail ou *chat*. É o que chamamos de aula "quase-presencial". Há uma interatividade quase igual à uma aula presencial, convencional, entre professor e aluno, ou seja, o professor pode ver os alunos e vice-versa, as perguntas e comentários são formuladas e respondidas durante a aula com o aluno que fez a intervenção sendo visto pelo professor e pelos demais colegas tal como ocorre em uma aula presencial.

2.2.4 – Dispor de meios, propiciados pela tecnologia, que permitam aos alunos obter o conhecimento de que necessitam em tempo real, vendo e ouvindo o professor, e fazendo perguntas com total interatividade. Aqui temos a situação de "mesmo tempo" e "mesmo ritmo". Só o "mesmo lugar" não se concretiza da maneira convencional, mas acontece através da "presença-eletrônica" usando os recursos de videoconferência. Atualmente, a Lei de Diretrizes e Bases da Educação Nacional, 9394/96, através do título VIII, "Das disposições gerais", no seu art. 80, explicita com clareza que "O poder público incentivará o desenvolvimento e a veiculação de programas de ensino a distância, em todos os níveis e modalidades de ensino, e de educação continuada". Neste sentido, hoje, a EaD amplia cada vez a sua importância nos processos de democratização da educação brasileira, seja em nível público ou corporativo/privado. Fontes relativas à legislação pertinente a EaD estão indicadas em [26]. O site http://www.abed.org.br contém vários *hiperlinks* ricos em informação sobre legislação relacionada à questão em pauta.

2.3 - Topologias

Do ponto de vista de abrangência as aulas podem ser classificadas como abaixo:

2.3.1 - Ponto-a-ponto, onde pode haver um professor de um lado e uma classe inteira na outra ponta.

2.3.2 - Ponto-multiponto, onde pode haver um professor de um lado e várias classes (com vários alunos cada) em localidades remotas distintas.

2.3.3 - Multiponto-multiponto, o que se assemelha mais a uma mesa de debates do que a uma aula clássica.

2.4 – Ferramentas e Aplicabilidade

Há uma multiplicidade de situações em que as ferramentas de EaD são valiosos instrumentos pedagógicos, ampliando novos horizontes de comunicação/interação.

Aqui vale mencionar algumas ferramentas que podem ser usadas para construir EaD:

a) ferramentas de autoria, que são editores de texto e gráfico, bem como páginas Web;
b) ferramentas de conferência, onde o *Lotus Notes* aparece com destaque;
c) ferramentas de Multimídia, em que o *RealVídeo* é um dos mais conhecidos, e para quem tem o Windows 2000, o MS *Media Server*;

d) ferramentas de simulação e aprendizado interativo, que pode ser obtido através de páginas Web com o uso de *JavaScript*, *Java* e controles *ActiveX*; além desses, o *Visual Basic* oferece grande poder de programação, assim como o ASP (*Active Server Page*);

e) ferramentas de avaliação que, normalmente, precisam ser desenvolvidas pois não existem pacotes prontos;

f) ferramentas de gerência.

Entre as várias situações em que elas podem ser utilizadas destacamos as seguintes, salientando que cada ferramenta tem aplicações específicas e, geralmente, precisamos lançar mão de várias delas para atingir os objetivos educacionais:

- viabilizar conferências (usando *e-mail*, listas de discussão, *News Groups* e *chats*);
- discutir questões da matéria ensinada;
- tirar dúvidas com o professor e colegas;
- praticar monitoria (que também pode ser via telefone);
- disponibilizar materiais e informações (via hipertextos em *sites* Web);
- organizar horários de aula e eventos;
- realizar e avaliar as tarefas de casa;
- disponibilizar literatura;
- proporcionar banco de informações com palestras;
- disponibilizar *Streaming* de vídeo/áudio de aulas pré-gravadas;

- gerenciar os cursos;
- acompanhar o progresso e participação dos estudantes;
- ministrar aulas ao vivo;
- transmitir aulas via meios terrestres ou satélite, com vídeo/áudio bidirecional;
- realizar aulas "quase-presenciais" com vídeo unidirecional e perguntas por *chat.*

2.5 – Implementação de uma Estrutura de EaD

Deve-se resistir à tentação de considerar o meio mais importante que o fim, ou seja, a tecnologia mais importante que a finalidade a que se destina, que é a aquisição de conhecimento por parte dos alunos.

A modelagem instrucional de cada curso deve ser desenvolvida a partir das necessidades dos clientes (alunos). Devem ser analisados, cuidadosamente, o objetivo que se pretende, o tipo do conhecimento que vai ser ministrado, a filosofia da instituição, a legislação pertinente e o orçamento de que se dispõe. O diagnóstico final é que vai apontar qual infra-estrutura tecnológica será a mais adequada, isto é, que trará melhor relação custo/benefício.

Uma questão chave é qual a natureza do curso em mente. Se for um curso de direito, por exemplo, não serão necessários grandes recursos de vídeo. No entanto, se for um curso de medicina ou odontologia, por exemplo, a resolução de imagem deverá ser alta. Há informações sobre um curso de astronomia em que estudantes no Japão observavam o telescópio de Mount

Wilson, em Pasadena-USA, através de um circuito de 45 Mbps, pois era necessária uma grande resolução de imagem [27].

O curso à distância pretende ser de FORMAÇÃO ou de INFORMAÇÃO? Será um curso voltado para a Graduação, Extensão Universitária, Mestrado, Doutorado, um curso de Capacitação ou uma simples palestra? Será para uma instituição de ensino, um órgão de governo ou uma empresa, seja ela pública ou privada? Os alunos são adultos ocupando postos de trabalho, com suas obrigações profissionais, familiares e sociais?

Seja qual for o seu uso e a sua natureza, é conveniente partir dos usuários (professores e alunos) e adequar a vasta gama de soluções que a tecnologia nos propicia, hoje em dia, às suas necessidades. Não o que achamos saber como necessidade, mas o que eles (usuários) manifestam. Um exemplo desta questão foi a pesquisa realizada no Campus da Universidade de Washington [28] para a implementação de uma plataforma de EaD, levando-se em consideração as necessidades de alunos, professores, bibliotecários etc. Eis algumas sugestões encontradas em [28]:

- **Planeje o uso da tecnologia de modo a atender os objetivos instrucionais.**
- **Crie os recursos e materiais baseados em Web.**
- **Conecte-se à rede do campus.**
- **Aprenda as lições ensinadas pelo uso da tecnologia de modo a melhorar sempre. Para isto disponibilize facilidades para que críticas sejam enviadas, tais como um site de pesquisas sobre o valor que as ferramentas tecnológicas agregam ao aprendizado, ou não.**

Ainda em [28] encontramos algumas lições já aprendidas:

- tecnologia é, apenas, mais uma ferramenta para ajudar professores a ensinar e alunos a aprender;
- recursos são limitados, portanto pense estrategicamente e use bem o que tem, procurando idéias, projetos e oportunidades que tenham o potencial de alavancar os recursos disponíveis (*sic*);
- aceite a incerteza e permita-se experimentar;
- mantenha uma comunicação transparente com os que dão suporte ao grupo, entendendo que o processo está sempre se aperfeiçoando e a interação contínua mantém todos engajados;
- colabore sempre, trocando idéias com a comunidade usuária e agindo como um "corretor" de relacionamento entre os grupos usuários.

Neste sentido, reafirma-se a idéia de que, por detrás de qualquer utilização de uma ferramenta de EaD, existem decisões de caráter político, pedagógico/curricular e institucional, entre outras.

CAPÍTULO 3

Revisão Bibliográfica sobre *Multicasting*

De modo a melhor transmitir o conteúdo multimídia na rede, do ponto de vista de carga na mesma, lança-se mão de protocolos *multicasting*. Temos, desta maneira, um campo importante do ponto de vista da engenharia de tráfego e, a despeito de vários protocolos já existentes, da necessidade de elaboração de novos protocolos mais eficientes, ou modificações nos atuais, para torná-los mais adequados aos novos usos.

Outros temas de pesquisa devem reclamar por atenção, mas pela sua importância, generalidade e aplicabilidade à realidade Brasileira, esta tese enfoca a elaboração de protocolos para *multicasting* mais eficientes e adequados que os que atualmente existem.

Quando falamos da transmissão de um-para-muitos, que é típico de EaD, os protocolos de *multicasting* aparecem como capazes de propiciar esta otimização reduzindo, por conseqüência, a largura de banda necessária e, em decorrência desta redução, os custos necessários para se implantar uma rede que sirva de plataforma para EaD. A Figura 3.1, (http://www.cisco.com/warp/public/732/Tech/multicast/docs/intromulticast.pdf), ilustra a economia de banda obtida.

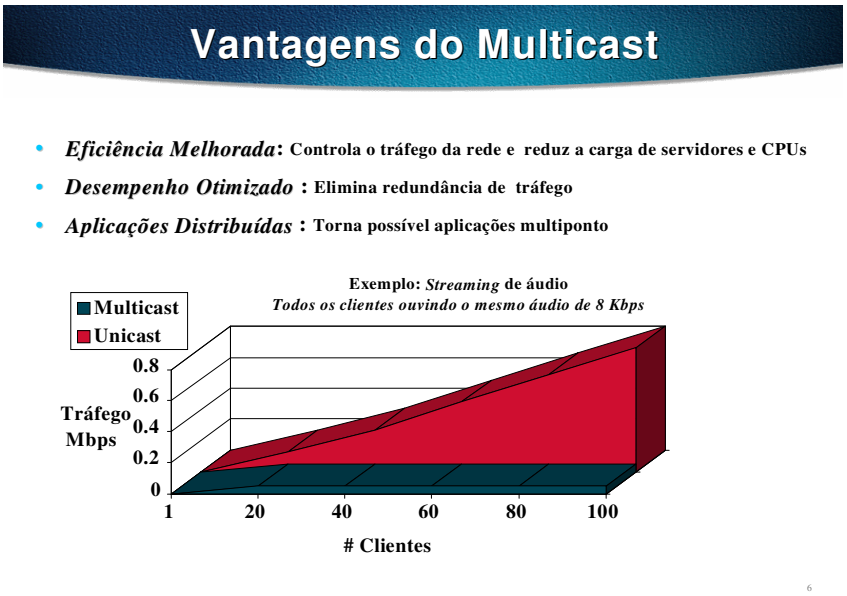


Figura 3.1 – Vantagem do uso de *Multicast.*

Do ponto de vista social, isto implica que mais pessoas podem ser alcançadas por redes de EaD, dado que os investimentos (tais como compra de equipamentos de comunicação de menor capacidade) e custos mensais (tais como aluguel de meios de transmissão) ficam diminuídos. Desta forma, o mesmo orçamento possibilita a construção de redes maiores e mais abrangentes.

Protocolos *Multicasting* é um tema em constante evolução, sendo alvo de pesquisa por boa parte da comunidade que se dedica à Internet.

Encontrava-se no IETF (*Internet Engineering Task Force*), o rascunho de um protocolo *Multicast* Bi-direcional à disposição da comunidade científica para críticas e sugestões. Este rascunho tinha a data de dezembro/2004 como “data de expiração”. Ele foi removido sem ser aceito como RFC (*Request For Comments*).

3.1 - PREMISSAS BÁSICAS

- Estaremos focalizando *e-learning* em corporações, assumindo que os treinandos são profissionais em pleno exercício de suas funções, adultos e com sua formação básica concluída.
- A corporação possui uma rede baseada no protocolo IP, ou seja, possui uma Intranet, ou quer construir uma.
- Esta rede pode ser exclusiva para EaD ou pode ser compartilhada por todas as aplicações que a companhia necessita rodar, o que é mais razoável.
- É imprescindível a utilização de vídeo [29].
- Possibilidade de se usar simuladores remotos e, provavelmente, realidade virtual [19].
- Dadas as considerações no item anterior, a rede deve ter capacidade para um acréscimo importante de largura de banda quando se estiver realizando uma EaD. É importante lembrar que os aplicativos de gestão, operação, correio eletrônico, *webpages*, e outros da corporação, estarão compartilhando os mesmos meios. Alta velocidade de transmissão (ou seja, grande largura de banda) é necessária na intranet. Como isto é caro, um cuidadoso planejamento de capacidade deve ser elaborado no dimensionamento da rede.
- Como em qualquer rede, busca-se minimizar os custos apesar das exigências de altas velocidades de transmissão.

- Numa empresa, existem vários profissionais com algum conhecimento necessário a outros grupos de funcionários. Desta maneira, tomaremos como princípio que um determinado funcionário será aluno para algum tema e poderá ser professor para outro.
- Busca-se, aqui, estabelecer que um aluno possa assistir uma determinada aula sem sair de sua mesa. Da mesma maneira, e dentro do princípio do item anterior, o professor pode dar aula sem sair de sua mesa de trabalho.
- Caso o conhecimento necessário não esteja disponível dentro da própria corporação, buscar-se-á um professor externo. A aula poderá ser ministrada de fora da corporação ou o professor poderá se deslocar para algum ambiente apropriado, dentro da empresa, para ministrar seu curso. Aqui, precisamos fazer algumas considerações sobre acessos externos à intranet, devido ao cuidado que se precisa tomar com invasões, vírus, etc. Normalmente os acessos externos estão bloqueados no *Fire-Wall*.
- Para atender à premissa, acima, assumimos que as corporações terão um micro–computador para cada funcionário, ligado em rede e equipado com placa de som e fones de ouvido (os fones de ouvido são melhores do que caixas acústicas no sentido de que o aluno pode ouvir a aula sem incomodar os colegas que se sentam por perto dele). Na pior das hipóteses, haverá uma sala próxima, equipada com micro-computador ligado em rede que será usado para assistir/ministrar aulas. Neste último caso, deve-se substituir os fones de ouvido por caixas acústicas, projetando o vídeo em um telão, via canhão ou *data-show*.
- A grande maioria dos eventos de *e-learning* são do tipo ponto-multiponto, ou seja, professor (em algum lugar da empresa) para alunos (em qualquer lugar da empresa). Daí a importância

de se usar o protocolo *multicasting*, habilitando esta facilidade em todos os roteadores, e *switches*, da intranet.

3.2 – CONCEITOS

Normalmente, os datagramas IP trafegam, na Internet, de uma única origem para um único destino (*unicast*), o que atende a um número muito grande de aplicações. No entanto, há outras aplicações que precisam que o datagrama seja enviado de uma origem para múltiplos destinos. Podemos citar, para este tipo de transmissão, a Educação a Distância (EaD), atualização de bancos de dados distribuídos, vídeo - conferência, atualização de *news group*, informações financeiras, etc. Neste caso, quando uma mensagem deve ser enviada de uma origem para múltiplos destinos, o que ocorre é a transmissão do mesmo datagrama várias vezes na rede, tantas vezes quantos forem os destinatários. Isto acarreta uma ocupação de banda desnecessária, desperdiçando recursos da rede para trafegar pacotes idênticos.

A modalidade de transmissão *multicasting* vem corrigir esta deficiência proporcionando que uma mesma mensagem seja transmitida uma única vez, ocupando a banda só uma vez.

3.3 – FUNCIONAMENTO

Podemos fazer *multicasting* em uma LAN e em uma WAN.

- No caso LAN, o início de uma recepção *multicast* se dá com o aplicativo do *host*, que quer participar de um determinado grupo, informando às camadas inferiores de protocolo (UDP e

IP) um determinado endereço IP *multicast*. A camada IP, por sua vez, notifica ao *Device Driver* que deve permitir a captura dos quadros Ethernet com endereço MAC mapeados pelo endereço IP de destino enviado pela fonte.

A placa de rede do *host* membro do grupo, já informada sobre o endereço de grupo, captura e envia para cima o quadro para ser processado pelas camadas intermediárias até chegar à aplicação. A Figura 3.2 mostra como um pacote pode ser recebido e passado para a camada superior ou descartado em uma das camadas. Os descartes podem ocorrer por causa do quadro não vir com o endereço MAC da placa, o quadro vir com erro ou não haver porta ativada para a aplicação entrante.

Do lado da transmissão, a estação de origem monta o pacote IP com o endereço *multicast* do grupo. A placa de interface de rede mapeia o endereço IP classe D no endereço MAC correspondente e o quadro é enviado.

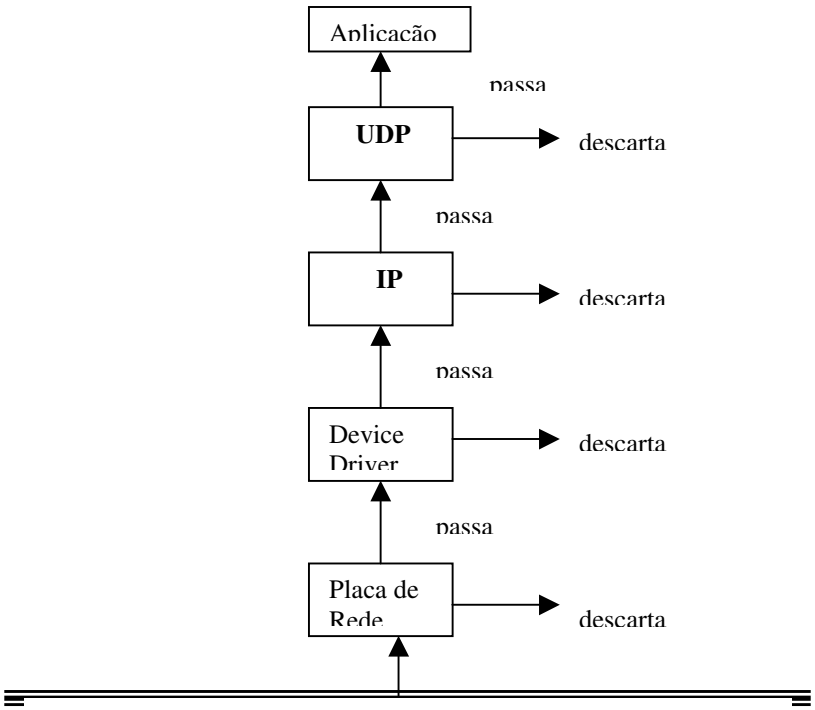


Figura 3.2 – Tratamento de um quadro Ethernet

- No caso da WAN, torna-se necessário preparar os roteadores e *switches* para trabalhar nesta modalidade. Os roteadores intermediários precisam saber se fazem parte da árvore *multicasting* e os roteadores designados (ligados diretamente às LANs) precisam saber se algum *host* de sua rede local está querendo participar de algum grupo. Além disto, o roteador deve ser capaz de encaminhar os datagramas *multicast* que receber, para o(s) *host*(s) pertencente(s) a um determinado grupo. Para o roteador designado falar com os *hosts* ligados na sua LAN, usa-se o protocolo IGMP. Para otimizar o tráfego dos datagramas *multicast* nos s*witches* usa-se o protocolo ICMP (RFC 0792) [30]. Para construir a árvore do grupo usa-se, principalmente, o protocolo PIM [31], como se verá mais adiante.

3.4 – PROTOCOLOS *MULTICASTING*

Os protocolos que implementam *multicasting* mais conhecidos são o DVMRP (*Distance Vector Multicast Routing Protocol*), o MOSPF (*Multicast Open Shortest Path First*, ou Multicast OSPF), o PIM (*Protocol Independent Multicast*) e o IGMP (*Internet Group Management Protocol*), sendo este último usado pelos protocolos mencionados acima com a função de fazer a comunicação de grupos *multicast* entre o roteador da LAN e os *Hosts* ligados a ele pelo barramento Ethernet, como se verá mais adiante.

3.4.1 - DVMRP - *Distance Vector Multicast Routing Protocol* (RFC 1075)

A métrica deste protocolo é o número de *hops* que ele considera para escolher uma determinada rota.

Um componente importante do DVMRP é o RPM (*Reverse Path Multicasting*) que faz com que os pacotes oriundos da fonte sigam a árvore de distribuição *multicast* até todos os membros de um grupo, repetindo o pacote só nos ramos necessários. Quando um datagrama chega a uma interface, o caminho reverso até a origem do datagrama é determinado examinando-se uma tabela de roteamento DVMRP de redes de origem conhecida. Se o datagrama chegar em uma interface que seria usada para transmitir datagramas de volta para a origem, então ele será encaminhado para a lista apropriada de interfaces subseqüentes. Caso contrário, ele não estará em uma árvore de distribuição ótima e deve ser descartado. Esta construção evita que se formem *loops* na árvore. A árvore muda de topologia "podando" ramos nos quais não haja mais *hosts* participando de grupo. De modo semelhante, quando um *host* resolve aderir a um grupo, é acrescentado um novo ramo à árvore de modo a transmitir os datagramas para aquele *host*. Diz-se que foi "enxertado" um novo ramo. Uma facilidade interessante do DVMRP (não existente no MOSPF) é a criação de túnel. Esta facilidade foi implementada uma vez que nem todos os roteadores suportam roteamento *multicast*. Através do túnel, um roteador DVMRP pode alcançar um outro roteador idêntico, em uma outra rede, passando por redes não capazes de *multicast*. Isto é conseguido através do encapsulamento de datagramas *multicast* IP em pacotes IP *unicast*, os quais são endereçados para os roteadores apropriados.

O DVMRP inicia com um *broadcast* geral e, em seguida, vai podando os ramos onde não existem roteadores (e *hosts*) interessados em participar do grupo. Ele monta árvores de *broadcast* por origem, com base nas trocas de tabela de roteamento e, depois, cria dinamicamente árvores de distribuição de *multicast* por grupo de origem, cortando os ramos da árvore.

Roteadores DVMRP “descobrem” seus vizinhos (outros roteadores DVMRP) através do envio de mensagens de investigação. Estas mensagens são enviadas periodicamente para todos os endereços de *multicast* IP de roteadores DVMRP. Cada mensagem-sonda contém a lista de roteadores DVMRP vizinhos que receberam as mensagens nessa interface. Dessa forma, esses roteadores DVMRP podem ter certeza de que um vê o outro. Uma vez que um roteador tenha recebido uma mensagem-sonda que contenha um endereço de roteador na lista de vizinhos, o roteador estabelece uma adjacência bidirecional.

Para localizar a origem da transmissão *multicast*, um roteador procura, na tabela de roteamento DVMRP, a rede de origem. A interface na qual a melhor rota de origem do datagrama é determinada é chamada de *upstream*. Quando um datagrama chega na interface *upstream* correta, ele é encaminhado para as interfaces *downstream*. Se o datagrama não tiver chegado na interface esperada ele, então, é descartado. Essa verificação é conhecida como Teste de Encaminhamento de Caminho Reverso e precisa ser realizada por todos os roteadores DVMRP.

Para garantir que os roteadores DVMRP tenham uma visão consistente do caminho de volta para uma origem, uma tabela de roteamento é propagada para todos os roteadores DVMRP como parte integrante do protocolo. Cada roteador anuncia o endereço da rede e a máscara das interfaces às quais ele está conectado diretamente, bem como transmite em cadeia as rotas recebidas dos roteadores vizinhos.

Quando mais de um roteador está ligado a uma LAN, é necessário estabelecer qual deles será o DESIGNADO para executar as funções de roteamento *multicast* para cada origem. O

critério é eleger aquele roteador que tenha menor métrica em relação à origem. Se houver empate, será tomado aquele com menor endereço IP.

3.4.2 - MOSPF - *Multicast* OSPF (RFC 1585)

Este protocolo é, na verdade, uma extensão do protocolo OSPF para atender a demanda de redes *multicast*. Novas facilidades foram incluídas para capacitar os bancos de dados de topologia OSPF de modo a computar as interfaces da árvore de caminho mais curto a partir da origem. O que se faz é acrescentar ao BD normal o "Estado de Enlace" para membros de grupos *multicast*. A partir daí, é calculada a árvore de caminho mais curto (*Shortest Path Tree*-SPT) com raiz na origem. Estas árvores de caminho mais curto são formadas a partir da recepção do primeiro datagrama. Os resultados são, então, armazenados em *cache* para serem usados por datagramas subseqüentes que tenham a mesma origem e o mesmo destino (*Multicast* ID). Como o MOSPF não implementa túnel, podem ocorrer problemas quando a origem está em outro AS (*Autonomous System*).

3.4.3 - PIM – *Protocol Independent Multicast* (SM: RFC 2362 e DM: RFC 3973)

Existem dois tipos de PIM, um para situações em que os *hosts* estão concentrados (*dense*) e outro em que estão dispersos (*sparse*) geograficamente.

O modo denso pressupõe que há largura de banda em abundância. O modo esparso é otimizado para larguras de banda não tão generosas.

O protocolo PIM-DM (*Dense Mode*) utiliza o algoritmo RPM (*Reverse Path multicasting*) e pressupõe a existência de um protocolo de roteamento *unicast* para se adaptar a mudanças na topologia.

O PIM-DM simplesmente encaminha o tráfego *multicast* em todas as interfaces *downstream* até que receba mensagens explícitas de podagem. No caso de membros de grupo resolverem aderir ao grupo no meio da transmissão *multicast*, em interfaces que tenham seu ramo associado podado, o PIM-DM emprega mensagens para incorporar uma determinada interface (que estava inativa para *multicasting*) à árvore de transmissão.

O PIM-SM (*Sparse Mode*) é o modo mais apropriado dentro do que pretendemos neste trabalho, pois pressupõe um professor sendo ouvido e visto por dezenas de alunos espalhados por todo o território onde a empresa atua.

O PIM-SM foi projetado para limitar o tráfego *multicast* somente para os roteadores que têm *hosts* participantes de algum grupo associados a eles. Tais roteadores se anexam à "Árvore de Distribuição *Sparse-Mode*" através do envio de mensagens de anexação. Caso contrário, eles não receberão tráfego *multicast* endereçado ao grupo. Ao contrário do modo denso, o modo esparso bloqueia todo o tráfego que não seja explicitamente requisitado.

O PIM-SM emprega o conceito de *Rendez-vous point* (RP), que é um roteador onde os *hosts* receptores encontram a fonte de transmissão, o *host* emissor. Um rede *multicast* pode ter mais de um RP, mas só um por grupo.

Cada *host* que deseje se juntar a um grupo *multicast* deve contatar seu roteador, na mesma LAN, usando o protocolo IGMP.

Por sua vez, o roteador a quem foi solicitado entrar na árvore de distribuição (pelo *host*) deve enviar uma mensagem de anexação ao RP do grupo.

A fonte da transmissão usa o RP para anunciar sua presença e para encontrar um caminho para os membros que se juntaram ao grupo. Fica claro que os roteadores da rede devem estar capacitados para trabalhar PIM-SM.

Quando houver mais de um roteador PIM em uma LAN, o que será designado para a função (*Designated Router*, ou DR) será o que tiver maior endereço IP.

O DR é o responsável por enviar mensagens de Anexação/Podagem para o RP. Quando ele (DR) recebe uma mensagem IGMP *Report* para um novo grupo, faz uma busca na Lista RP-grupo para determinar o RP para aquele grupo. Depois, cria um *cache* de roteamento *multicast* para o par (*,grupo) e envia uma mensagem PIM-*Join*, *unicast*, para o RP. A notação (*,grupo) indica que pode ser qualquer fonte (*).

Os roteadores intermediários encaminham a mensagem PIM-Join e criam um entrada no *cache* de roteamento para o par (*,grupo). Desta maneira os roteadores intermediários sabem como rotear o tráfego endereçado para o par (*,grupo) *downstream* para o DR que originou a mensagem PIM-*Join*.

Quando uma fonte transmite um pacote *multicast* pela primeira vez para um grupo, o roteador ao qual a fonte está anexada deve enviar um datagrama para o RP que, por sua vez, o repassará à árvore de distribuição daquele grupo.

O roteador da fonte encapsula o pacote *multicast* em um pacote PIM-SM-*Register* e o envia, em modo *unicast*, para o RP do grupo.

O pacote PIM-SM-*Register* informa ao RP que uma nova fonte deseja transmitir, obrigando o RP ativo a transmitir uma mensagem PIM-*Join* de volta ao roteador da fonte. Os roteadores entre o roteador da fonte e o RP devem manter o status das mensagens PIM-*Join* recebidas de modo a saber como reencaminhar os pacotes *multicast* subseqüentes, não encapsulados, da fonte para o RP. O Roteador da fonte pára de encapsular pacotes quando recebe a mensagem *Join/Prune* do RP. A partir deste ponto, o tráfego originado na fonte é roteado pelo roteador da fonte, no formato *multicast* nativo, para o RP.

Quando o RP recebe os pacotes *multicast* da fonte, ele reenvia os datagramas na árvore para todos os membros do grupo. Esta árvore recebe o nome de RP-*Shared*. Isto ocorre porque o tráfego da fonte deve sempre, passar pelo RP. Esta não é uma maneira otimizada de usar a rede. A melhor maneira é aquela em que a fonte envia os pacotes *multicast* direto para os *hosts*, sem passar pelo RP, formando uma árvore SPT (*Shortest Path Tree*). Apesar da SPT diminuir o atraso dos pacotes e otimizar os enlaces que conectam os roteadores, esta facilidade não é usada na EMBRATEL em favor da simplificação da administração da rede.

3.4.4 – IGMP - *Internet Group Management Protocol* (v2: RFC 2236)

O IGMP roda entre os *hosts* e seu roteador *multicast* imediatamente vizinho. Através deste protocolo, um *host* informa ao seu roteador local que quer receber transmissões de um determinado grupo *multicast*. Por sua vez, este roteador se comunica com os roteadores acima

dele (*up stream*), vindo a fazer parte da árvore *multicast*. Quando receber datagramas *multicast,* saberá se deve retransmití-los para os *hosts* da LAN ao qual está conectado.

Periodicamente, o roteador envia mensagens *Query* na sua LAN para determinar se há *hosts* querendo participar de algum grupo. As mensagens *Query* são endereçadas para todos os *hosts* ligadas à LAN e tem *Time to Live* = 1, de modo que não possa ser retransmitido para nenhuma outra sub-rede. Quando um *host* recebe a mensagem *Query* ele reponde com uma mensagem *Report* para cada grupo do qual deseje participar. Para evitar uma enxurrada de *Reports*, cada *host* inicializa um temporizador e, se durante esta temporização, algum *Report* (daquele grupo) é percebido, o *host* não precisa mais responder ao *Query* e zera seu temporizador. Basta que um dos *hosts* ligados à LAN participe de um grupo para que o roteador envie os datagramas daquele grupo no barramento da LAN. Se nenhum *host* da LAN responder ao *Query* após um intervalo de tempo, o roteador entende que os *hosts* que estavam ouvindo a transmissão *multicast* se retiraram e os datagramas não precisam mais ser enviados, retirando-se da árvore *multicast*. Quando um *host* junta-se a um grupo, ele, logo de início, envia um *Report* independentemente do roteador enviar um *Query*.

Atualmente, o IGMP está na versão 2 que implementa um procedimento para determinação do roteador designado em uma LAN com mais de um roteador no barramento. O roteador escolhido é o que tem o menor endereço IP. Além disto, a versão 2 possibilita o *host* enviar uma mensagem *leave* quando se retira de um grupo, quando na versão 1 era necessário esperar 3 *Queries* não respondidos para sinalizar esta ação.

O IGMP só trata *multicasting*, em uma LAN, entre os *hosts* e o Roteador Designado.

3.5 – PROTOCOLOS DE ROTEAMENTO

Para entendermos *multicasting,* se faz necessário entender o que são protocolos de roteamento em uma rede IP. A capacidade de roteamento é o que dá à Internet sua principal vantagem, que é a conectividade, encaminhando pacotes de maneira independente uns dos outros para destinos diferentes. A função de rotear pode ser desempenhada por máquinas que executam, por exemplo, um programa Servidor e, ao mesmo tempo, roteiam pacotes para outras máquinas. Contudo, existem máquinas que são dedicadas à função de rotear, que são os roteadores, de uso generalizado.

O processo que executa o roteamento é chamado, em Unix, de *daemon* (*disk access and execution monitor*). O *daemon* é, normalmente, inicializado quando o *boot* (a carga do Sistema Operacional ao ligar o computador) é efetuado e roda durante todo o tempo em que a máquina estiver ligada. A Figura 3.3, adiante, ajuda-nos a entender o que acontece [30].

Vemos o *daemon* atualizando regularmente a Tabela de Roteamento que é usada para apontar a melhor interface pelo qual deve ser enviado o datagrama recebido. Os comandos **route** e **netstat** são digitados no *prompt* do sistema operacional e são usados pelo operador para obter informações da tabela de roteamento de sua máquina. O driver de placa envia para esta camada (IP) o datagrama que é colocado, inicialmente, em uma fila. Depois o *Header* do pacote é processado e caso haja o campo *Options* no datagrama, é acionada a rotina pertinente, que no desenho está mostrada a opção de **roteamento pela origem**. Caso o pacote entrante tenha o

mesmo endereço desta máquina ou seja um broadcast, então o pacote é enviado para a camada superior seja UDP , TCP ou mesmo ICMP.

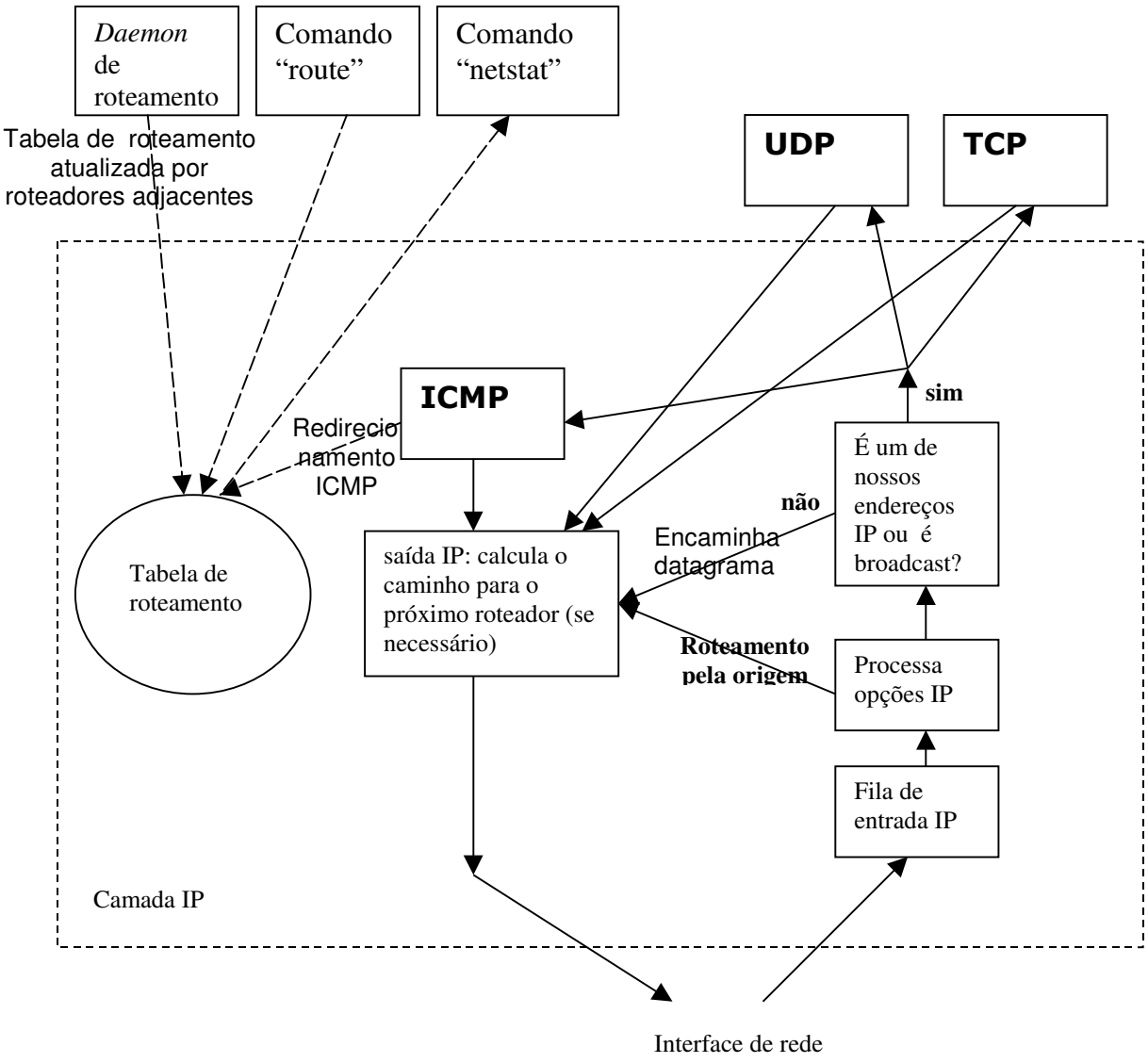


Figura 3.3 – Processamento na camada IP.

Um roteador pode ter várias interfaces por onde entram e saem pacotes. Uma rotina fica varrendo, permanentemente, as interfaces configuradas como de entrada e, ao receber um pacote entrante, consulta as opções IP deste pacote, verifica se este pacote é destinado a ele mesmo, caso contrário é enviado em alguma interface de saída indicada pela Tabela de Roteamento.

Caso o pacote seja para o próprio roteador seu cabeçalho IP é removido e o *pay-load* enviado para ser tratado por um processo ICMP, UDP ou TCP, conforme indicado no cabeçalho.

Para ser eficiente, a Tabela de Roteamento deve refletir as condições das vizinhanças do roteador, motivo pelo qual ela é freqüentemente atualizada após a primeira carga. A atualização manual é extremamente ineficiente para redes grandes, como as que vemos hoje em dia. Algoritmos foram desenvolvidos para que esta atualização seja automática e freqüente.

Tendo em mente que um endereço IP é composto de uma parte que aponta para o *host* e outra que aponta para a rede, um roteador procura na sua tabela local, primeiro, um casamento do endereço completo do *host*. Caso não tenha sucesso, procura um casamento do endereço de rede. Se nenhuma destas procuras achar o endereço pesquisado, é feita uma procura pelo endereço do roteador *default*. Uma vez encaminhado o pacote, o próximo roteador fica encarregado de enviar o pacote para frente, e assim por diante, até o pacote alcançar seu destino.

Vale observar que uma máquina, *host* ou roteador, pode encaminhar um pacote para si próprio, daí a utilização da facilidade de *loop-back interface*. Por convenção, a maioria dos sistemas atribui o endereço IP 127.0.0.1 a esta interface, à qual dão o nome de *localhost*. A Figura 3.4, a seguir, ilustra este comportamento [30]. Esta é uma facilidade desejável uma vez

que numa transmissão de voz e/ou vídeo o transmissor também deve receber sua própria emissão de modo a poder monitorar seu sinal. Na figura vemos que o datagrama que está saindo é colocado na fila de entrada da camada IP se o endereço de destino for o seu próprio, broadcast ou multicast de seu interesse.

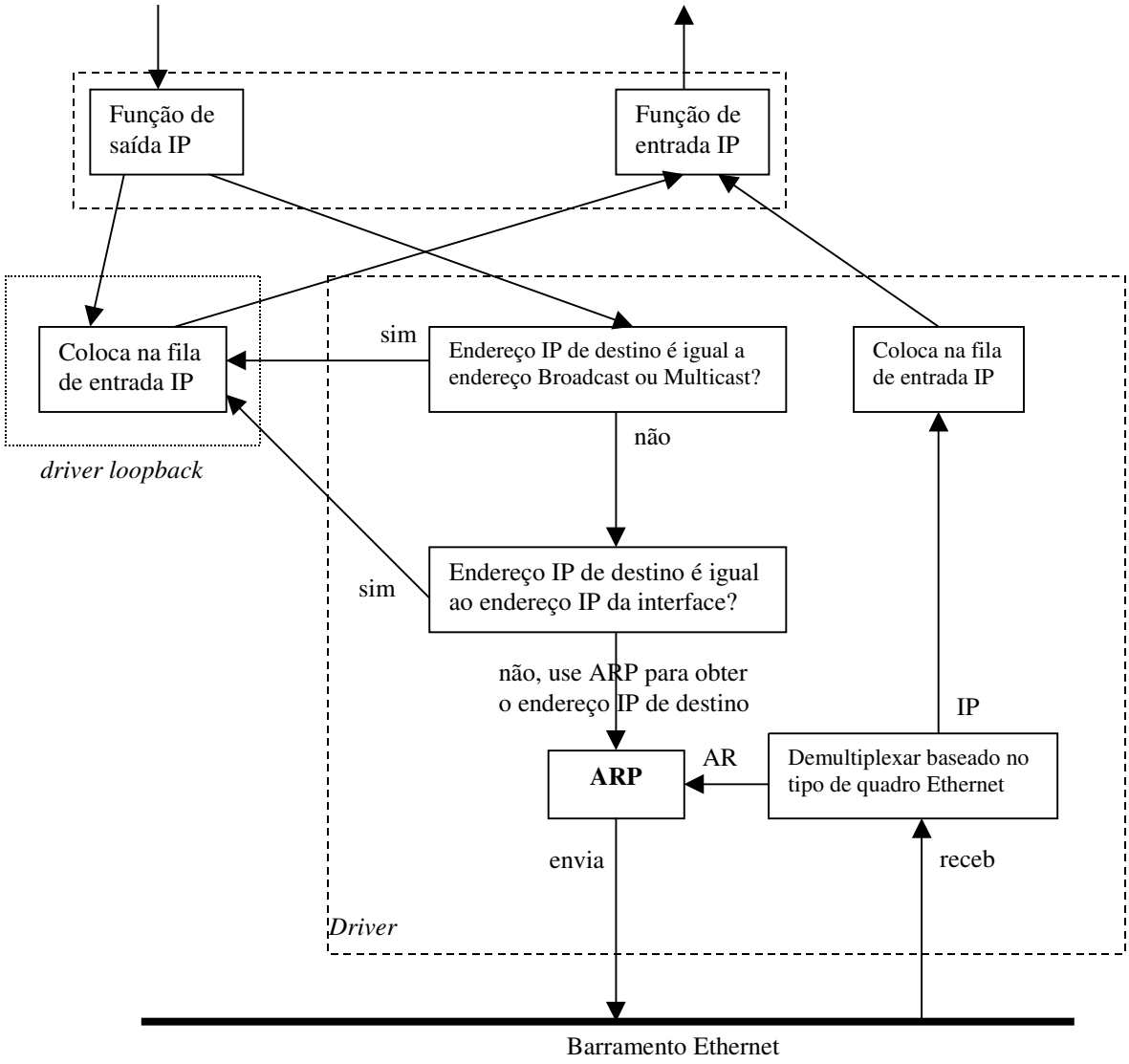


Figura 3.4 – O funcionamento da *loop-back interface.*

Um pacote IP enviado pela máquina local para o barramento Ethernet pode ser copiado para a própria máquina através da facilidade *loop-back,* caso o endereço IP de destino seja de *broadcast*, *multicast* ou seu próprio endereço.

Uma tabela de roteamento típica precisa conter pelo menos os itens abaixo:

- endereço IP de destino;
- endereço IP das interfaces;
- a interface na qual está ligado o roteador *default* (também chamado *gateway*);
- *flags* que indicam:
  - U = a rota, em questão, está ativa e funcionando (*up*);
  - G = a rota aponta para um roteador default (*gateway*) - se este *flag* não estiver ligado, então o destino é uma rede diretamente conectada, ou seja, o endereço IP do roteador é da Interface (atenção, o endereço IP de destino pode ser de um *host* ou de outra rede);
  - H = a rota aponta para um *host*, ou seja, o endereço IP de destino é de um *host*. Se este *flag* não estiver ligado então a rota é para uma rede e o endereço é de rede - quando ligado, significa que o endereço IP de destino é o endereço de um *host*;
  - D = a rota foi criada por um redirecionamento;
  - M = a rota foi modificada por um redirecionamento.

Na Tabela 3.1 vemos que as interfaces assinaladas estão ativas (*up*). Vemos a sempre presente interface Lo0 (*localhost*), bem como a interface *default* que é para onde se manda o

pacote quando não se sabe o que fazer com ele. A primeira linha nos informa que o destino está além do *host* apontado na tabela (*flag* G ligado). A última linha nos informa que o destino é a LAN diretamente ligada ao roteador cuja tabela estamos analisando, pois como esta LAN está, também, diretamente conectada o *flag* G não está ligado.

Destino	Roteador	Interface	*Flag* (U/G/H/D/M)				
140.252.13.65	140.252.13.35	Emd0	1	1	1	0	0
127.0.0.1	127.0.0.1	Lo0	1	0	1	0	0
Default	140.252.13.33	Emd0	1	1	0	0	0
140.252.13.32	140.252.13.34	Emd0	1	0	0	0	0

Tabela 3.1 – Exemplo de Tabela de Roteamento.

Para manter a clareza de exposição, omitimos a máscara que vai permitir saber qual é a porção *netID* e a porção *hostID*. Por exemplo, na linha que aponta para a LAN (*netID*) a máscara seria 0xffffffe0 (255.255.255.224 em decimal), com os 5 bits mais à direita do endereço IP colocados em 0 (*hostID* = 0).

3.6 – CONSTRUINDO E MANTENDO A TABELA DE ROTEAMENTO

Os protocolos que fazem o roteamento dinâmico, ou seja, atualizam dinamicamente as tabelas de roteamento, são divididos em IGP (*Interior Gateway Protocol*, também chamado de *Intradomain Routing Protocol*) [32] e EGP (*Exterior Gateway Protocol*, também chamado de *Interdomain Routing Protocol*) [32]. A primeira família de protocolos provê o anúncio das tabelas de roteamento dos roteadores vizinhos dentro de um mesmo *Autonomous System* (AS). A

segunda se encarrega do roteamento entre AS's. Dos protocolos IGP, temos o HELLO (que foi abandonado), o RIP (*Routing Information Protocol*) e o OSPF (*Open Shortest Path First*, atualmente o mais usado). Dos protocolos EGP, temos o EGP (o uso da mesma sigla não é erro de digitação, abandonado) e o BGP (*Border Gateway Protocol*, atualmente usado em sua forma *light* ou *full*).

3.6.1. - RIP – *Routing Information Protocol* (RFC 1058)

O protocolo RIP (*Routing Information Protocol*), nos ajuda a compreender os mecanismos envolvidos em um processo de roteamento dinâmico.

A mensagem RIP é transportada em datagramas UDP que, por sua vez, é claro, são formatados como pacotes IP.

A Figura 3.5, ajuda a entender a descrição que se segue [30].

Comando	Versão	ID do daemon
Família de Endereço		ID do AS
Endereço IP (32 bits ou 4 bytes, quando IP) da rota pesquisada		
Máscara de Sub-rede (32 bits ou 4 bytes, quando IP)		
Endereço IP do roteador *next-hop*		
Métrica (1 a 16)		
Pode conter mais 24 rotas, no formato anterior (menos a primeira linha)		

Figura 3.5 – Formato da Mensagem RIP versão 2.

Este “pacote” tem o comprimento mínimo de 24 bytes. O campo **Comando** especifica se é um *Request* (neste caso recebe o valor 1) ou um *Reply* (neste caso recebe o valor 2). Existem outros comandos: 3 e 4, obsoletos, 5 e 6, *Poll* e *Poll-entry*, não documentados.

Um *Request* pede ao outro roteador para enviar toda, ou parte de sua tabela de roteamento. Um *Reply* contém a resposta conforme requisitado.

O campo **Versão** é preenchido com o valor 1, para a versão 1 do RIP ou 2 para RIP 2.

O campo **ID do *daemon*** fornece uma identificação do processo que gerou a mensagem (também chamada de *routing domain*). Isto permite rodar múltiplas instâncias do RIP em um único roteador. No RIP 1, este campo é preenchido com zeros.

O campo **Família de Endereço** deve ser 2, caso o endereço associado seja IP.

O campo **ID do AS** (também chamado de *route tag*) permite trabalhar com protocolos EGP (p.ex. BGP), pois identifica o AS ao qual pertence. Este campo é preenchido com zeros no RIP 1.

O campo **Máscara de Sub-rede** se refere ao endereço IP. Deve ser preenchido com zeros no RIP 1, visto que o mesmo só opera nas classes A, B e C, desconsiderando sub-classes produzidas pelo emprego de máscaras.

Quando o campo **Endereço IP do roteador *next-hop*** contiver o valor 0 (zero), indica que a mensagem RIP deve ser enviada ao roteador que gerou a solicitação. Caso contrário devem ser enviados ao endereço constante neste campo. No RIP 1, este campo deve ser preenchido com zeros.

O campo **Métrica** deve ser preenchido com um valor que reflita a quantidade de *hops* que existem entre o roteador local e o roteador distante de quem se quer baixar a tabela de roteamento.

Daí para frente pode-se adicionar outros campos de forma a anunciar mais 24 rotas. Este limite é fixado para atender a MTU (*Maximum Transmission Unit*) de 512 bytes (20 x 25 + 4). Fica claro que serão necessários várias mensagens RIP para enviar uma tabela de roteamento inteira.

O protocolo RIP tem uma "porta bem conhecida", de número 520.

O funcionamento do RIP segue os passos abaixo:

1) Quando é dado a partida no roteador, são identificadas as interfaces que estão ativas e enviada, em cada uma delas, uma mensagem RIP de *Request*, solicitando tabelas de roteamento completas. Esta mensagem *Request* tem o campo Comando igual a 1, mas o campo Família de Endereço igual a 0 e o campo Métrica igual a 16 (que representa infinitos *hops*, ou seja, o outro roteador está inalcançável).
2) Ao receber o *Request*, o receptor envia toda sua tabela de roteamento ao solicitante.
3) Quando o roteador solicitante recebe a resposta, esta é validada e a tabela de roteamento local é atualizada. Isto significa que pode-se introduzir novas entradas na tabela, apagar entradas existentes ou modificá-las.

A cada 30 segundos, cada roteador envia sua tabela atual para todos os roteadores vizinhos a ele, a qual pode ser enviada de forma parcial ou total. A tabela também é enviada quando existe

alguma modificação nela, que deve ser comunicada aos roteadores vizinhos. Esta forma é chamada de *triggered update*. Neste caso, só o que foi mudado é enviado.

Cada rota tem um *time-out* associado, e se uma determinada rota não é atualizada após 3 minutos, então a métrica daquela rota é colocada em 16 (infinito, rota não existente) e marcada para eliminação da tabela. É a mesma coisa que dizer que após 6 atualizações automáticas não foi recebida nenhuma atualização daquela rota determinada. O roteador espera mais 60 segundos para retirar aquela rota da tabela de modo a garantir que a invalidação se propagou para os roteadores vizinhos.

Quando o campo Família de Endereço tiver o valor 0xFFFF (isto só pode ocorrer no primeiro pacote) e o campo ID do AS o valor 2, os próximos 16 bits da mensagem conterão uma senha (*password*), composta de caracteres ASCII ajustados à esquerda e preenchidos com nulos à direita, de modo a prover algum nível de autenticação ao protocolo (esta facilidade não existe no RIP 1). Adicionalmente, o RIP 2 suporta *multicasting* de modo a reduzir a carga em roteadores que não estão "escutando" mensagens RIP. O RIP 1 só suporta *broadcast* ou ponto-a-ponto.

O protocolo RIP é lento para estabilizar na rede quando ocorre uma mudança repentina tal como a queda de uma rota. Durante este tempo podem ocorrer *loops*. Por este motivo, é um protocolo de difícil operação. Além disto, o uso da métrica de contagem de *hops* não leva em consideração outros fatores importantes, como vazão, velocidade do enlace, retardo e custo (indicado pelo administrador de rede).

3.6.2 - OSPF – *Open Shortest Path First Protocol* (RFC 2328)

Este é um protocolo muito usado para redes grandes, desde que estejam dentro de um único AS. Ao invés da contagem de *hops*, como no protocolo RIP, usa o conceito de "Estado de Enlace".

Para entendê-lo, devemos conhecer como ele foi construído, ou seja, sua arquitetura. Vale lembrar que este trabalho não é sobre OSPF e o que queremos é apenas compreender o mecanismo sobre o qual ele se baseia, para poder entender o funcionamento dos protocolos de *multicast*.

Para que o OSPF possa ser usado, devemos dividir a rede em áreas. Uma delas, a principal, será o *backbone*. A rede deve ter, pelo menos, uma área que, é claro, será o próprio *backbone*. Cada área recebe uma designação que tem a forma de um endereço IP mas, atenção, não é endereço IP. Cada área é composta por uma quantidade de roteadores ligados entre si de forma lógica, dependendo da topologia da rede onde se pretende implantar o roteamento OSPF. As áreas podem ser definidas a partir de um planejamento que contemple conveniências geográficas, administrativas, etc.

É importante ressaltar uma característica fundamental do OSPF, que é a de que todos os roteadores mantem a mesma tabela de roteamento. Para isto, os roteadores de uma mesma área trocam, freqüentemente, informações de estado de enlace, de modo a manter o sincronismo.

Estes roteadores, dentro de uma mesma área, são chamados de ROTEADORES INTRA-ÁREA (IAR, *Intra Area Router*).

Os roteadores que interligam o *backbone* com as demais áreas, caso existam, são chamados de ROTEADORES DE FRONTEIRA DE ÁREA (ABR, *Area Border Router*). Estes podem, também, interligar duas áreas sem passar pelo *backbone* (neste caso, a área intermediária, entre uma dada área e o *backbone*, é chamada de Área de Trânsito). Estes roteadores mantém bancos de dados da topologia da área à qual pertencem e trocam informações de estado de enlace com outros roteadores da área adjacente.

Os roteadores que interligam uma rede OSPF com outras redes são chamados de ROTEADORES DE FRONTEIRA DE AS (ASBR, AS *Border Router*) . O protocolo usado nesta tarefa é um EGP (como, por exemplo, o BGP-4). Os roteadores ASBR também são encarregados de anunciar, para dentro de sua rede, o estado dos enlaces externos ao AS de que fazem parte.

Quando não é possível colocar todas as áreas adjacentes ao *backbone*, lança-se mão do conceito de Enlace Virtual, que nada mais é que um enlace ponto-a-ponto, com métricas de estado idênticas às da situação intra-áreas, ligando dois roteadores ABR de uma área de trânsito, de modo a ligar o *backbone* à área não contígua. Na Figura 3.6, a área C é de trânsito [32].

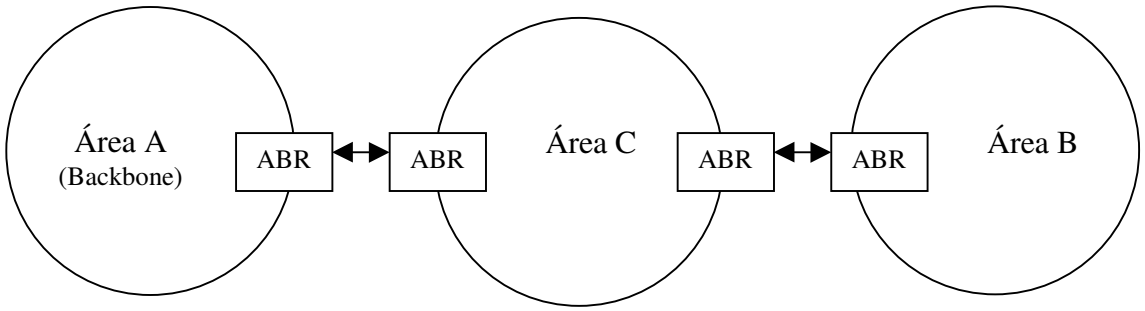


Figura 3.6 – Área de Trânsito entre duas regiões OSPF.

Quaisquer dois roteadores que possuam interfaces para uma rede comum são chamados de ROTEADORES VIZINHOS. Estes roteadores podem estar classificados de uma das maneiras que vamos ver, de acordo com seus estados. Dois roteadores vizinhos podem tornar-se ADJACENTES se, depois da troca de mensagens entre si, chegarem à condição de possuírem tabelas de estado de enlace iguais, ou sincronizadas. Só roteadores adjacentes trocam informações de estado de enlace.

Todas as redes multi-acesso devem ter determinados um ROTEADOR DESIGNADO e um ROTEADOR DESIGNADO RESERVA. Os roteadores designados geram anúncios de enlaces de rede que relacionam os roteadores anexados a uma rede multi-acesso. Além disso, eles formam adjacências com outros roteadores tornando-se, portanto, o ponto-focal para o envio de todos os anúncios de estado de enlace. O roteador designado reserva é idêntico ao principal e assume em caso de falha deste.

Um roteador, pode ter conexão com um (ponto a ponto) ou mais roteadores (multi-acesso). Estas conexões são efetuadas através das INTERFACES dos roteadores que, por sua vez,

também estão descritas pelos seus estados. Resumidamente podemos listar os Estados de uma Interface como sendo [32]:

- *Down* – a interface não está disponível (estado inicial).
- *Loopback* – é o modo de enviar os pacotes para si próprio.
- *Waiting* – o roteador está tentando determinar a identidade do roteador designado ou do seu reserva.
- Ponto a ponto – o roteador forma adjacência com um único roteador na outra ponta.
- DR *Other* – a interface está em um estado de multi-acesso mas o roteador não é um DR nem seu reserva.
- *Backup* – este é o DR reserva.
- DR – é o próprio roteador designado.

Cada roteador cria e mantém um BANCO DE DADOS DE ESTADO DE ENLACE, também chamado de GRAFO DIRECIONADO e, para tanto, executa o algoritmo SPF (*Shortest Path First*) sobre o Grafo de modo a obter a árvore de caminho mais curto, que é empregada na construção da Tabela de Roteamento.

Apesar do Banco de Dados de Estado de Enlace ser idêntico em todos os roteadores de uma mesma área, cada roteador constrói uma árvore diferente, pois vê a rede a partir de si próprio.

As Tabelas de Roteamento contêm entradas para cada destino, seja ele uma Rede, uma parte de uma rede ou um *host*.

Cada área é identificada por um número de 32 bits, sendo o endereço 0.0.0.0 atribuído à área *backbone*.

Cada roteador possui, um ID que pode ser o endereço IP mais baixo que ele possui. Também possui uma PRIORIDADE, que é um número inteiro de 8 bits e indica a prioridade de um dado roteador na sua eleição para DR. Se a Prioridade for igual a 0 (zero) isto significa que o roteador em questão não pode ser um DR.

Os Estados de Enlace são compostos por 5 tipos, que descrevem completamente uma rede OSPF e seu ambiente externo:

- Enlaces de Roteador, que são produzidos por todos os roteadores, dentro de uma área e descrevem o estados de suas interfaces (são anunciados em uma única área).
- Enlaces de Rede, que são produzidos pelos DRs em uma rede multi-acesso e relacionam os roteadores conectados à rede, dentro de sua área (são anunciados em uma única área).
- Enlaces de Resumo, que são produzidos pelos ABRs, descrevendo as rotas destino para outras áreas e/ou as rotas para roteadores da fronteira AS (são anunciados em uma única área).
- Enlaces de AS Externos, que são produzidos por ASBRs e descrevem as rotas para destinos externos à rede OSPF (são anunciados em todas as áreas da rede, em questão).

3.6.2.1 - DESCRIÇÃO DO FUNCIONAMENTO DO OSPF

Os pacotes OSPF são transmitidos diretamente em datagramas IP e não sobre UDP ou TCP, como no caso do protocolo RIP. Os datagramas IP que contém mensagens OSPF são

identificados pelo número 89 (decimal) no campo Identificador de Protocolo, pelo número 0 (zero) no campo Tipo de Serviço, e o valor "Controle de Rede" (todos os 3 bits no estado 1) no campo Precedência. Desta forma as mensagens OSPF têm preferência sobre o tráfego IP normal[2].

Os pacotes OSPF usam a técnica de *multicasting*, sendo enviados para um endereço *multicast* "bem conhecido" (224.0.0.5) também conhecido como "AllOSPFRouters".

Cada roteador OSPF deve executar as seguintes funções:

- Descobrir vizinhos.
- Selecionar o DR.
- Inicializar os vizinhos.
- Propagar as informações de enlace.
- Calcular as tabelas de roteamento.

É importante relacionar os estados que o relacionamento entre dois roteadores vizinhos podem assumir:

- *Down* – é o estado inicial da conversação entre dois roteadores vizinhos, que indica que nenhuma informação recente (*timeout* configurável entre 40s e 120 s) foi recebida do vizinho.
- *Attempt* – é o estado em que um roteador tenta estabelecer conversação com um roteador vizinho que parece estar caído.

[2] Estamos sempre nos referindo ao IPv4, que é o que a literatura consultada aborda.

- *Init* – este estado indica que um pacote *Hello* foi recebido de um vizinho, mas uma comunicação bidirecional ainda não foi estabelecida (o que acontece quando o roteador vê a si próprio na tabela recebida do vizinho).
- *2-way* – neste estado, a comunicação entre os vizinhos é bidirecional e, a partir daqui, pode-se estabelecer uma adjacência e algum dos vizinhos pode ser eleito como DR, ou seu reserva.
- *ExStart* – este estado indica que os dois vizinhos estão prestes a criar uma adjacência.
- *Exchange* – este estado indica que os dois vizinhos estão trocando informações de Banco de Estado de Enlace.
- *Loading* – neste estado, os vizinhos estão sincronizando seus Banco de Estado de Enlace.
- *Full* – este estado indica que os vizinhos são, agora, adjacentes e seus Bancos de Estado de Enlace estão sincronizados.

Antes de continuarmos vamos rever conceitos importantes para a compreensão do que vai ser abordado a seguir:

- roteadores que estão conectados a um único roteador (enlace ponto a ponto, virtual ou não) serão sempre adjacentes.
- roteadores que estão conectados a mais de um roteador (multi acesso) formarão adjacência apenas com os roteadores eleitos como DR ou DR reserva.

- só roteadores adjacentes trocam informações de estado de enlace e, para isto, é necessário que eles, primeiro, tenham o mesmo Banco de Dados de Topologia e que os mesmos estejam sincronizados.

Para isto são definidos 5 tipos de pacotes OSPF:

- Pacote *Hello* – é o pacote responsável por descobrir, estabelecer e manter os roteadores vizinhos. Estes pacotes são enviados periodicamente em todas as interfaces de um roteador. Através da troca de pacotes *Hello* são executadas as tarefas de descobrir e inicializar vizinhos e selecionar o DR. Assim que envia o pacote *Hello* em uma interface, o estado do relacionamento do roteador naquela interface muda de *Down* para *Point-to-Point*, DR *Other* ou *Waiting*, dependendo da situação. Quando recebe um pacote *Hello* em uma interface, o estado de relacionamento daquela interface muda para *Init*. Se o pacote *Hello* recebido contiver dados do roteador receptor, o estado muda de *Init* para *2-way*, pois um relacionamento verdadeiramente bidirecional foi estabelecido entre os vizinhos. Após a escolha do DR, o estado de relacionamento daquela interface muda de *Waiting* para DR, *Backup* ou *DROther*, dependendo se o roteador escolhido for classificado como designado, designado-reserva ou nenhum destes.
- Pacote Descrição de Banco de Dados OSPF - para sincronizar BDs, usa-se um mecanismo chamado de Troca entre Banco de Dados, que é obtido pelo intercâmbio de vários pacotes de Descrição de BD, o qual define o conjunto de informações de estado de

enlace presente no BD de cada roteador. Neste processo estabelece-se um relacionamento Mestre/Escravo, sendo o Mestre o primeiro a transmitir e a enviar os pacotes de descrição de BD, forçando o escravo a descrever o seu BD, o que pode requerer o envio de vários destes pacotes onde o último tem o *flag* M (*more*) desligado, indicando que não há mais nada a transmitir.

- Pacote Solicitação de Estado de Enlace OSPF – após o encerramento do processo anterior, cada roteador solicita o anúncio de estados mais atualizados usando o presente pacote. Durante a troca entre BDs, o estado da interface passa de *2-way* para *ExStart* (quando o Mestre concordar com a adjacência criada), *Exchange* (quando os BDs já tiverem sido descritos), *Loading* (durante o tempo em que as solicitações de estado de enlace estiverem sendo enviadas e recebidas) e *Full* (quando a adjacência estiver completada, ou seja, BDs sincronizados).
- Pacote Atualização Estado de Enlace – uma vez estabelecido o sincronismo dos BDs, é necessário mantê-los assim devido a mudanças nos estados dos enlaces da rede. Os pacotes ora descritos são os encarregados desta tarefa. Como já vimos, existem 5 tipos de estados de enlace a serem anunciados. As atualizações ocorrem como conseqüência dos pacotes de solicitação, ou por ocorrência de uma mudança em algum estado na rede. O processo usado para transmitir pacotes de atualização é o de *flooding* com alcance de um *hop*, somente.
- Pacote Reconhecimento de Estado de Enlace – este pacote é usado para reconhecer, separadamente, cada anúncio de estado de enlace, de modo a tornar o procedimento de

propagação do anúncio confiável. Um anúncio é descartado se (i) o *checksum* do estado de enlace estiver incorreto, (ii) o tipo do estado de enlace não for válido, (iii) a validade atingir seu limite, (iv) o anúncio for mais antigo (ou de mesma idade) que um que já esteja no BD. Se um anúncio for considerado válido, o roteador envia um Pacote de Reconhecimento de Estado de Enlace ao originador do anúncio e o propaga para outras interfaces até que tenha sido recebido por todos os roteadores daquela área. Caso o originador não receba o Pacote de Reconhecimento, ele retransmite o anúncio após um certo tempo. Após a atualização, o roteador recalcula o grafo de topologia e a tabela de roteamento.

A última etapa após a troca dos pacotes acima descritos é a do Cálculo da Tabela de Roteamento. Cada roteador, em uma dada área, constrói um BD da topologia dos anúncios de estado de enlace válidos e o utiliza para calcular o mapa de rede da área. A partir deste mapa, o roteador pode determinar a melhor rota para cada destino e inseri-la em sua tabela de roteamento. Um anúncio permanece em um BD, no máximo, por uma hora, após o que ele é retransmitido como se fosse um anúncio novo.

3.7 – MULTICASTING NA CORPORAÇÃO: O CASO EMBRATEL

Para montar nossa rede *multicast*, foram usados os protocolos:

- PIM-SM entre os roteadores.
- IGMP e CGMP (*Cisco Group Management Protocol*) dos roteadores para os *hosts*, e vice-versa.

A escolha do PIM-SM foi tomada por ser um protocolo que, se mostrou adequado à topologia da rede da empresa, que é dispersa sobre uma grande área geográfica, o Brasil e São Paulo se apresenta como um local ideal para a convergência de todas as demais localidades, sendo a escolha natural para a instalação do roteador *Rendez-vous*. O IGMP foi adotado por ser a solução padrão. O CGMP foi adotado porque, uma vez que toda a rede da EMBRATEL é composta por roteadores e *switches* CISCO, propicia uma melhor utilização das redes.

Na configuração adotada na EMBRATEL, *Shared Tree* usando PIM-SM, é atribuído a um dos roteadores da rede o papel de Ponto-de-Encontro (RP). Este roteador se anuncia aos demais roteadores.

Os pacotes *multicasting*, recebidos da Fonte pelo RP, são enviados para os roteadores que solicitaram inclusão em sua árvore e, daí, são retransmitidos para os *hosts* que pertencem ao grupo em questão. A cada grupo é atribuído um endereço classe D. Quando houver um *switch* entre o Roteador Designado (DR) e os *hosts,* é aconselhado usar o protocolo CGMP para evitar que o *switch* replique o pacote *multicasting* em todas as suas saídas. Com este protocolo, o

switch só encaminha os pacotes recebidos para as saídas onde existem *hosts* pertencentes ao grupo respectivo.

Sabemos que, em uma LAN Ethernet, os pacotes são vistos pelas Placas de Rede de todos os *hosts* ligados ao barramento. A Placa de Rede tem que decidir se captura o pacote e o envia para o *Device Driver*. Normalmente, a Placa vai capturar somente os quadros[3] Ethernet que têm o endereço MAC dela ou o endereço Broadcast, que é da forma ff:ff:ff:ff:ff:ff, em hexadecimal.

Além disto, as Placas podem ser configuradas para receber endereços *multicast*. Este endereço, na Ethernet, é da forma 01:00:5E:xx:xx:xx, em hexadecimal.

Quando o endereço MAC é reconhecido pela placa, o quadro é passado ao *Device Driver*, que verifica se o protocolo especificado é suportado (IP, ARP, RARP). Depois, é verificado se o *host* em questão é membro do grupo, ou seja, se responde ao endereço *multicast* do quadro entrante.

Se tudo estiver correto, o datagrama IP, retirado de dentro do quadro Ethernet, é passado ao nível imediatamente superior. Aqui, pode-se efetuar uma filtragem baseada no endereço IP.

Quando os protocolos UDP ou TCP recebem o pacote, efetuam uma nova filtragem, agora baseada no número da porta de destino. Se não houver um processo rodando com aquele número de porta, o pacote é descartado e uma mensagem ICMP (*host unreachable*) é enviada de volta à origem.

[3] Os Sniffers e Analisadores de Protocolo têm suas Placas de Rede colocadas em modo promíscuo para que possam capturar todos os quadros Ethernet que passam por ele.

3.7.1 – Endereçamento

Usam-se endereços da classe D para transmissões *multicasting* [32] (Figura 3.7).

De 224.0.0.0 até 239.255.255.255

Formato do endereço IP, classe D

Abaixo relacionamos os primeiros endereços usados, como exemplo:

1) o endereço 224.0.0.1 indica que o datagrama deve ser enviado a “todos os sistemas nesta parte da rede”.
2) O endereço 224.0.0.2 indica que o datagrama deve ser enviado a “todos os roteadores nesta parte da rede”.
3) O endereço 224.0.0.3 não atribuído.
4) O endereço 224.0.0.4 indica todos os roteadores DVMRP
5) O endereço 224.0.0.5 indica todos os roteadores OSPF
6) O endereço 224.0.0.13 indica todos os roteadores PIM, etc.
7) .....

Figura 3.7 – Exemplo de endereços classe D atribuídos.

O IANA (*Internet Assigned Numbers Authority*) detém os endereços MAC de 00:00:5e:00:00:00 até 00:00:5e:ff:ff:ff. Destes, os endereços 01:00:5e:00:00:00 até 01:00:5e:7f:ff:ff são usados para endereços *multicast* MAC [32].

A Figura 3.8 mostra o mapeamento entre os endereços IP e MAC.

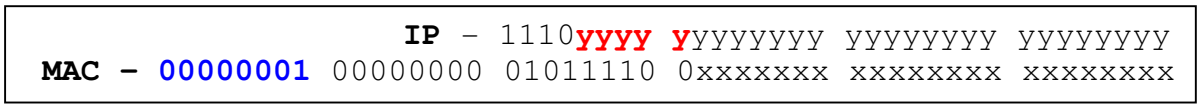


Figura 3.8 – Mapeamento IP x MAC.

Os bits em vermelho não são usados para formar o endereço MAC.

O byte em azul indica para a placa de rede que este é um endereço (MAC) *multicast*. Este byte está em formação *little endian* (bit mais significativo mais à direita) onde o bit 7 indica *unicast* (se for igual a 0) ou *broadcast/multicast* (se for igual a 1). O bit 6 indica se é Universalmente Administrado (se for igual a 0) ou Localmente Administrado (se for igual a 1).

O fato de se ter 5 bits ignorados pode causar alguns problemas, uma vez que o endereço IP não é unívoco, ou seja, pode haver até 32 (2^5) endereços IP mapeados no mesmo endereço MAC e, portanto, devem ser evitados.

3.7.2 - Endereçamento Administrativamente Agrupado

O confinamento dos pacotes *multicast* através do TTL (*Time to Live*) às vezes acarreta problemas de descarte pelo roteador de pacotes *multicast* quando há *hosts downstream* querendo receber a transmissão.

A configuração de roteadores de borda, que não propagam pacotes *multicast* para além dos seus limites, permite confinar as transmissões de maneira eficiente, otimizando os recursos de rede e garantindo a privacidade.

Este mecanismo é particularmente interessante para redes corporativas. Dada a importância desta utilização de *multicasting,* foi definida uma faixa de endereços específica que é 239.180.0.0/14 (*Organization-local scope*).

3.7.3 – Exemplo de uma conexão m*ulticast*

Vale lembrar que a ponta receptora é, sempre, um *host* ligado em uma LAN com, pelo menos, um roteador na LAN que vai se encarregar de ligar o *host* ao grupo do qual ele deseja participar, recebendo os pacotes *multicast* da fonte de programa.

Quando um *host* deseja receber uma transmissão *multicast,* é ativado o seu aplicativo o qual instrui a Placa de Rede para receber o endereço IP *multicasting* para aquela transmissão específica (grupo). O *host* comunica ao seu *default gateway* que deseja participar de um determinado grupo e, para tanto, utiliza o protocolo IGMP. Ao saber que um *host* sob sua área de abrangência deseja entrar em um grupo, este roteador solicita, na WAN, sua inclusão na árvore daquele grupo através do protocolo *multicast*.

De tempos em tempos, o roteador pergunta aos *hosts* de sua área se ainda estão no grupo, usando o protocolo IGMP. Caso nenhum *host* se pronuncie como participando de nenhum grupo, o roteador solicita sua exclusão da árvore. Os pacotes *multicast* são enviados para os roteadores que solicitaram inclusão em sua árvore e, daí, são retransmitidos para os *hosts*. Quando ocorre de

haver um *switch* entre o *default gateway* e os *hosts* é aconselhado usar o protocolo CGMP para evitar que o *switch* replique o pacote em todas as saídas, encaminhando-os só para as saídas de interesse.

Na Figura 3.9, vemos que a fonte de programa (F) se anuncia aos roteadores (R) da rede (seta 1, azul). Se algum *host* desejar participar da sessão ele deve avisar seu *default gateway* através do protocolo IGMP (seta 2, vermelha). O roteador *default gateway* encaminha o pacote *multicast*, -o para os *hosts* do grupo (seta 3, verde).

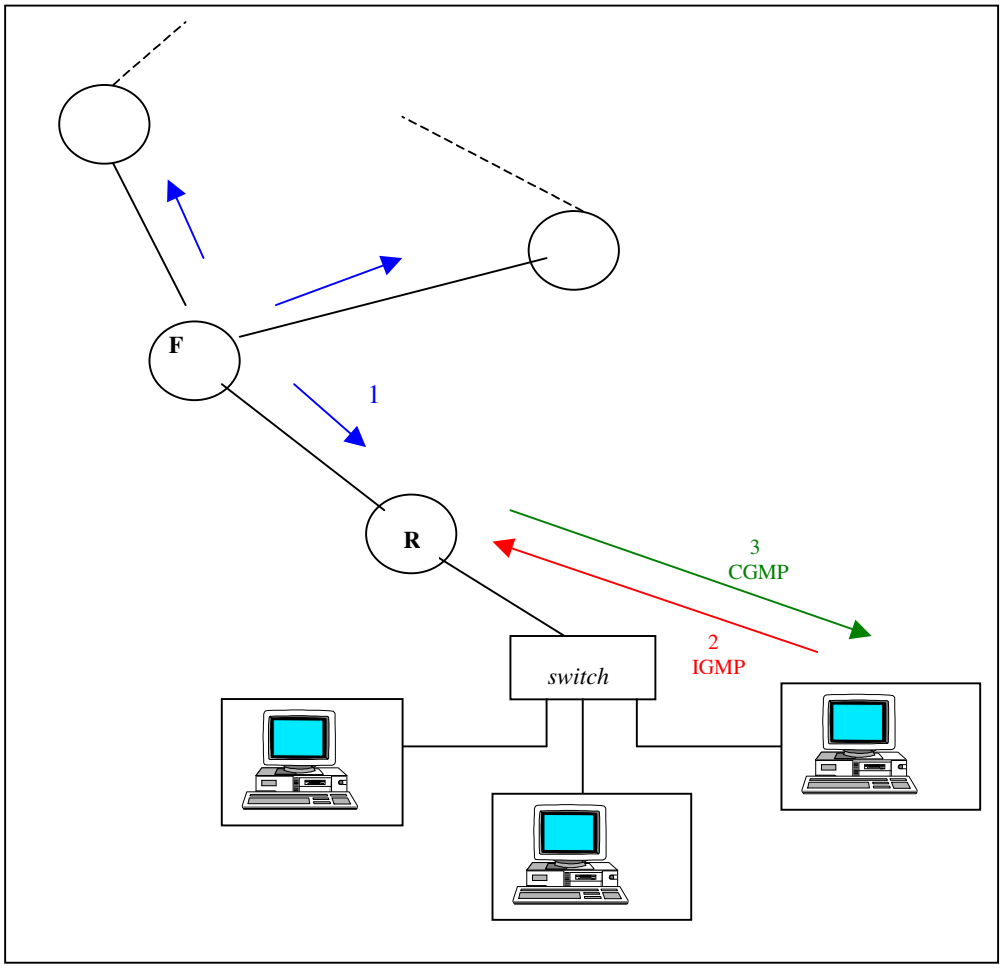


Figura 3.9 – Ilustração do funcionamento do IGMP e CGMP.

CAPÍTULO 4

Protocolo Multicast Bi-Direcional, XMP

O nosso foco será em uma configuração que responde pela grande maioria das redes instaladas no mundo, ou seja, *hosts* (PCs) conectados a redes locais (LAN) Ethernet ligadas a roteadores que, por sua vez, se ligam a outros roteadores (via LAN ou WAN).

4.1 - NOSSA PROPOSTA

Considerando:

- a necessidade de mudar o paradigma de "um professor e vários alunos" para "vários professores e vários alunos" e, ainda, um professor em um determinado momento pode ser um aluno e vice-versa, um indivíduo que é aluno para um determinado tema poderá ser professor para outro tema. Além disto, também existe o caso em que uma única aula é ministrada por mais de um professor (que podem estar geograficamente distantes).

- que queremos introduzir o conceito de que um professor pode ministrar a aula de sua própria mesa de trabalho, o mesmo se aplicando para os alunos, no sentido de assistir à aula.

- que, dentro desta visão, precisamos de uma ferramenta que suporte esta diversidade de localização e de papéis, além de propiciar a transmissão de vídeo e áudio, ora a partir do professor, ora de algum aluno ou de uma sala com vários alunos, comutando rapidamente o áudio e o vídeo recebidos de um para outro.

Dadas as considerações acima sugerimos o protocolo para *multicast* descrito a seguir.

O protocolo tem que atender várias fases no estabelecimento, manutenção e comutação da transmissão entre pontos diversos, além do encerramento da sessão.

É premissa, ainda, que o professor tenha em sua máquina uma câmera de vídeo, placa de som e algum software que codifique e transmita áudio/vídeo (A/V). Para que as pontas remotas (alunos ou salas) possam fazer suas intervenções na fala do professor, é necessário que estas pontas também tenham um conjunto compatível de hardware e software com o do professor. É evidente que um software específico precisa estar instalado de modo a poder fazer funcionar todas as facilidades de processamento de A/V e de comunicação. O uso de placas de vídeo dedicadas melhoram consideravelmente a qualidade das imagens geradas, aliviando a CPU (*Central Processing Unit*) do trabalho pesado de capturar, digitalizar e comprimir o sinal de vídeo.

As fases a serem implementadas são as seguintes:

Fase 0 – início da sessão, quando os *hosts* envolvidos obtêm os parâmetros necessários para o estabelecimento das conexões.

Fase 1 - registro dos participantes nos roteadores envolvidos.

Fase 2 – a transmissão de A/V, propriamente dita.

Fase 3 – mudança da origem da transmissão (quando algum aluno faz uma intervenção, por exemplo).

Fase 4 – retorno para a condição normal (que é o professor transmitindo).

Fase 5 – encerramento da sessão.

Passamos ao detalhamento das várias fases:

Fase 0 - Nesta fase, por exemplo, professor e alunos acessam um Servidor Web, onde está localizada a chamada para o curso. Este acesso pode ser feito através de um *browser* e permite a todos visualizarem a data e hora, bem como o tema da palestra, dados sobre o palestrante, etc. Pode ser necessário, ou não, o fornecimento de uma senha (previamente distribuída aos participantes por e-mail) de modo a evitar que pessoas estranhas participem. Outra maneira é fazer uso do ASP (*Active Server Page*) onde as pessoas só conseguem ver na página aquilo que seu *logon* na rede permite. Ao localizar a palestra de seu interesse, o participante pode clicar com o mouse sobre ela e recebe, via um *applet*, os endereços IP do Roteador *Focal Point* (ou FP, que veremos adiante), do grupo (endereço da classe D) e da máquina do professor (F, de fonte) que será, também, o moderador da sessão, permitindo a quem solicitar que faça a intervenção com a conseqüente transmissão de seu A/V.

Fase 1 - Nesta etapa do processo, o receptor (R) e a fonte (F) se registram no FP através do envio de uma mensagem que chamaremos START(G) conforme mostra a Figura 4.1.

Número do Grupo	Quem sou	Tipo
Carga: amostra A/V		

Obs.: O Campo Carga só existe em pacotes do tipo A/V

Figura 4.1 – Formato Geral de um Pacote XMP.

No campo número do grupo é colocado o número do grupo atribuído pelo administrador de rede.

O campo “Quem sou” informa a origem da fonte da mensagem que pode ser um F ou R (ou outro tipo de participante que venha a ser criado).

No campo “Tipo” consta a natureza do pacote que pode ser um comando ou uma amostra A/V. Só haverá o campo Carga se o campo “Tipo” contiver o indicador de que o pacote carrega uma amostra A/V. Os tipos podem ser *Start*, *Stop*, A/V.

A mensagem START(G) é do tipo *unicast*, enviada diretamente ao FP, passando através dos vários roteadores que possam estar no trajeto até o FP. O endereço do FP, obtido na fase 0, é colocado no campo Endereço de Destino do cabeçalho IP do datagrama que encapsula nossa mensagem.

Cada roteador no trajeto da mensagem constrói uma tabela que vamos chamar de MRT (*Multicasting Routing Table*). Ver o exemplo na Tabela 4.1.

Interface	Quem sou
1	F
2	R

Tabela 4.1 – Exemplo de MRT

Para construir a MRT, o roteador examina o campo HEADER LENGTH do cabeçalho do datagrama IP e, caso ele indique a presença do campo OPTIONS (*Header Length* > 5) neste

cabeçalho, examina este campo. Se for uma indicação de que o datagrama é deste protocolo que estamos descrevendo, chama-se uma rotina que trata de processar o datagrama, montando a tabela e retransmitindo-o para as interfaces assinaladas e/ou na interface para FP, apontada pela Tabela de Roteamento.

Para preencher o campo “Interface”, basta o roteador colocar ali a interface por onde entrou o datagrama, ou a interface na direção de FP, apontada pela Tabela de Roteamento.

O campo “Quem sou” indica o papel que a máquina origem da mensagem desempenha no grupo, que pode ser de F, R.

A MRT tem associada a si contadores para controlar quantos Rs enviaram START(G) em uma mesma interface. Cada interface deve possuir um contador (outra maneira de monitorar a população de um determinado grupo em uma interface de um dado roteador é através de uma mensagem tipo *KeepAlive* enviada a cada 3 minutos, mesma temporização do IGMP, por cada um dos participantes do grupo e caso um roteador não receba esta mensagem por mais de 3 minutos em uma dada interface, ele remove esta interface da MRT).

Fase 2 - Uma vez que FP e os roteadores envolvidos estão com suas tabelas montadas, F começa a enviar pacotes com as amostras A/V, colocando no endereço de destino o endereço do grupo (este procedimento propicia que os Rs que estão na mesma LAN de S recebam a amostra A/V diretamente sem precisar ir a FP e voltar).

Os roteadores envolvidos no caminho da amostra até FP, ao receberem o datagrama com o campo OPTIONS, assinalando que é um datagrama XMP, consultam sua MRT, confirmam que o pacote entrou pela interface cadastrada para F, confirmam se o emitente é um F (só um F pode

enviar amostras de A/V), confirmam a qual grupo pertence o datagrama, e enviam o pacote nas interfaces registradas como R para aquele grupo e na interface para FP, menos na interface por onde este tipo de pacote entrou. O endereço que o roteador coloca como de destino é o do grupo.

O FP, ao receber o datagrama, consulta sua tabela para ver se a interface é a que aponta para F e replica o pacote em todas as interfaces assinaladas como "R", menos aquela por onde entrou o pacote (o mecanismo é o mesmo do item anterior). O endereço que FP coloca no campo DESTINATION ADDRESS é o do grupo.

Os roteadores do trajeto *down-stream* replicam em todas as interfaces, registradas na sua MRT, o pacote recebido do FP, de novo, menos na interface por onde entrou o A/V.

Fase 3 - Se algum receptor quiser fazer uma intervenção, ele deve solicitar, através de facilidades proporcionadas pela aplicação, à fonte. A fonte executa um comando que faz com que sua aplicação autorize a aplicação do receptor solicitante a entrar em modo de transmissão, ou seja o receptor assume a categoria F.

A solicitação do receptor vai para a fonte através de uma mensagem *unicast* (lembrar que o endereço da fonte foi passado para o receptor no início da sessão).

A autorização da fonte para o receptor também é feita em *unicast,* utilizando o endereço IP que veio no campo SOURCE ADDRESS do cabeçalho do pacote IP na qual veio a solicitação do receptor.

O receptor interessado em virar fonte envia a mensagem START(G) com F no campo "Quem Sou". Em cada roteador, incluindo FP, este pacote faz com que a interface configurada como F e a interface por onde este pacote foi recebido invertam seus papéis, ou seja, F vira R e R

vira F. Este pacote é, então, encaminhado somente por esta interface que estava configurada como F. Em outras palavras, esta mensagem é enviada do receptor à fonte, podendo passar por FP.

A aplicação da fonte, ao receber o START(G) do receptor, entra em modo R. Desta forma, temos um mecanismo de *acknowledgement* do receptor para a fonte.

O receptor, agora F, envia amostra A/V para FP usando o endereço *multicast* do grupo a que pertence. FP replica o pacote com a amostra em todas as interfaces cadastradas em sua tabela, menos na interface por onde entrou o pacote.

Fase 4 - A fonte deseja retomar a transmissão (o receptor pode ter acabado de fazer a pergunta e a fonte vai, agora, responder). Ele comanda sua aplicação para a retomada. Esta comanda a aplicação do receptor para voltar ao modo R, enquanto ela própria retorna ao modo F. A fonte anuncia a alteração aos roteadores envolvidos e a FP através da mensagem START(G) os quais atualizam suas tabelas MRT. Todas estas mensagens são *unicast*. A fonte envia pacote com amostra A/V para FP em modo *multicast* e tudo se repete.

Fase 5 - Para sair da conferência, o participante envia mensagem *unicast* STOP(G) para FP. Os roteadores envolvidos decrementam seus contadores e, caso o contador fique zerado, o roteador em questão retira aquela interface da MRT e, portanto, pára de enviar os pacotes recebidos de FP naquela interface. FP faz o mesmo com sua própria MRT.

4.2 – Diagramas de Funcionamento

Em seguida vamos mostrar nossa proposta de uma forma gráfica através de diagramas.

4.2.1 – Roteadores Intermediários (X é o início)

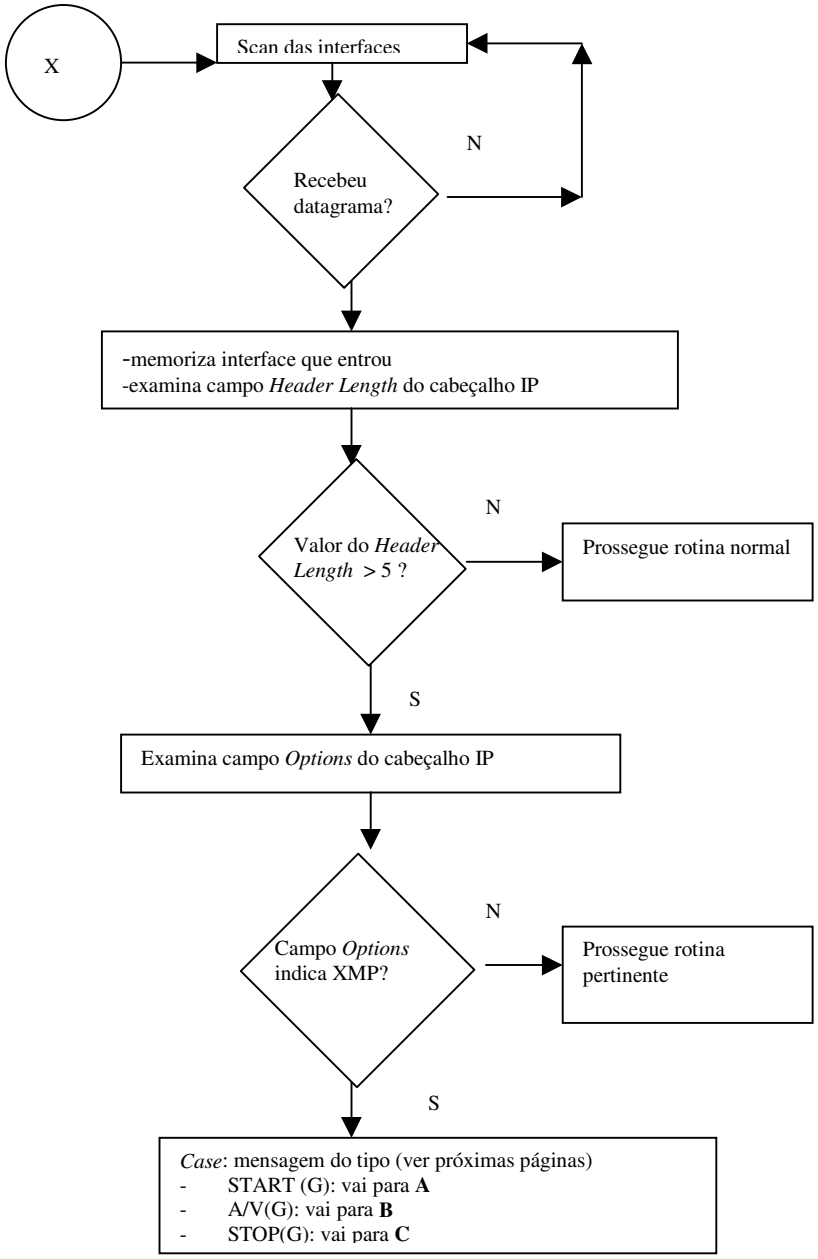


Figura 4.2 – Diagrama de funcionamento dos Roteadores Intermediários (1/4).

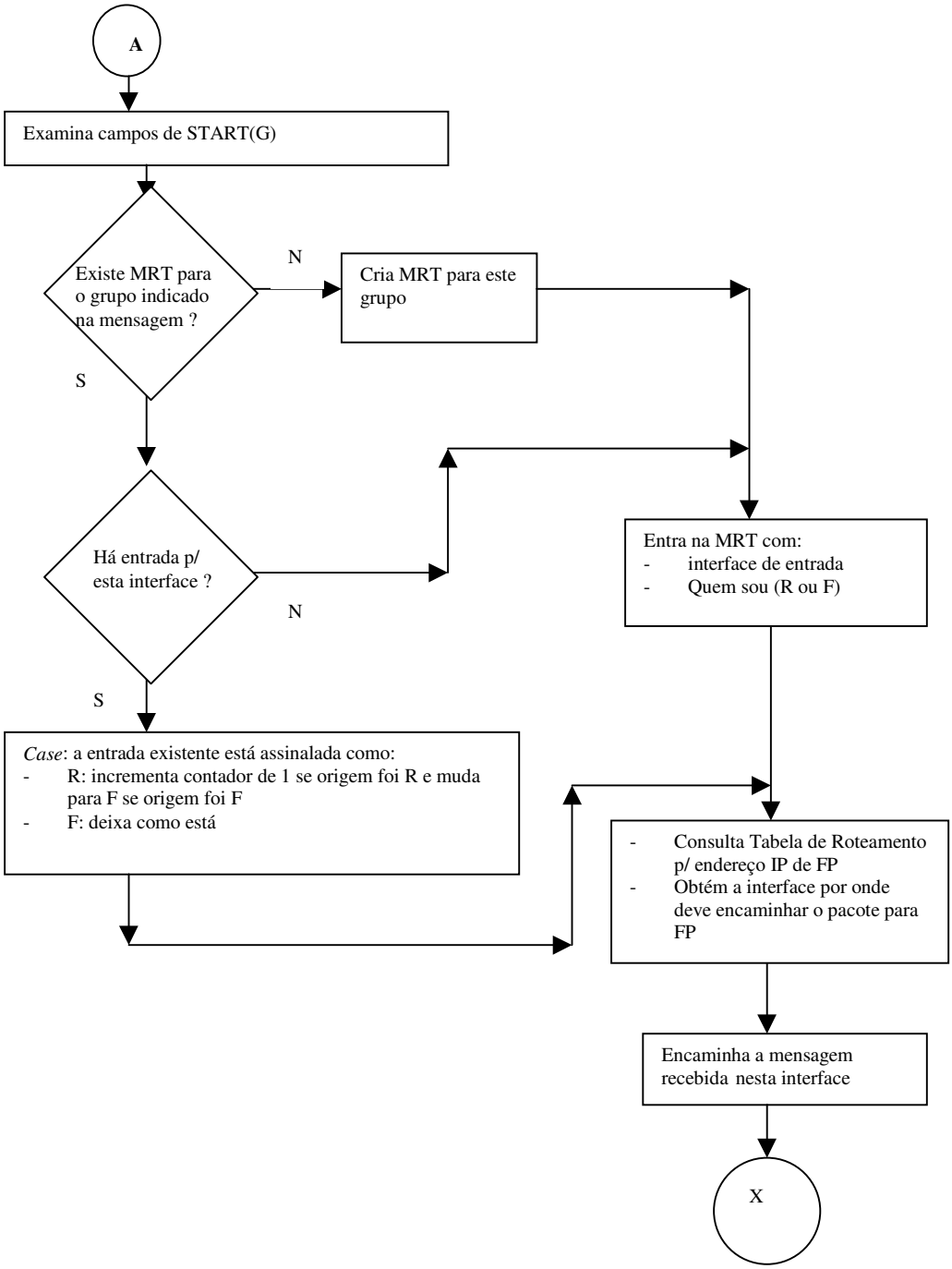


Figura 4.3 - Diagrama de funcionamento dos Roteadores Intermediários (2/4).

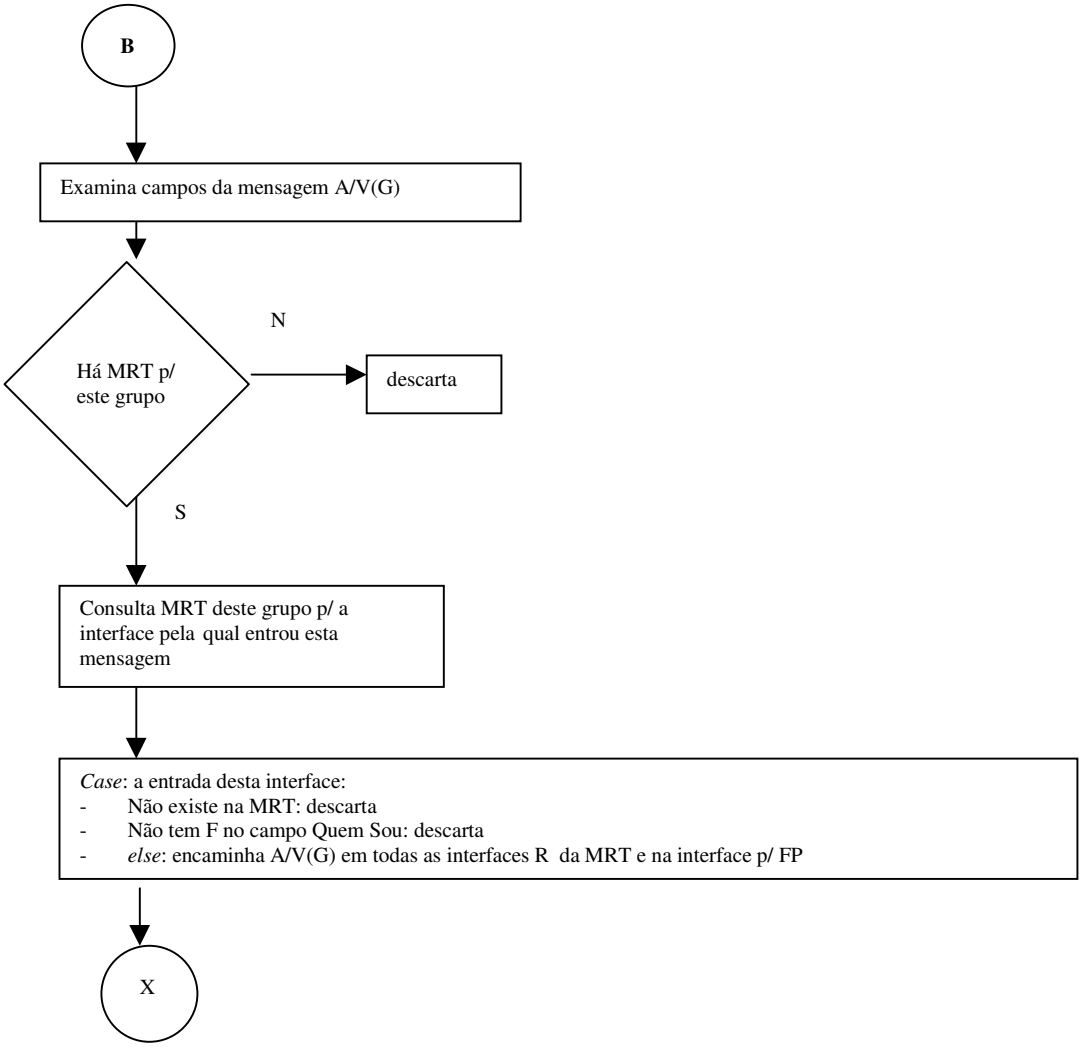


Figura 4.4 - Diagrama de funcionamento dos Roteadores Intermediários (3/4).

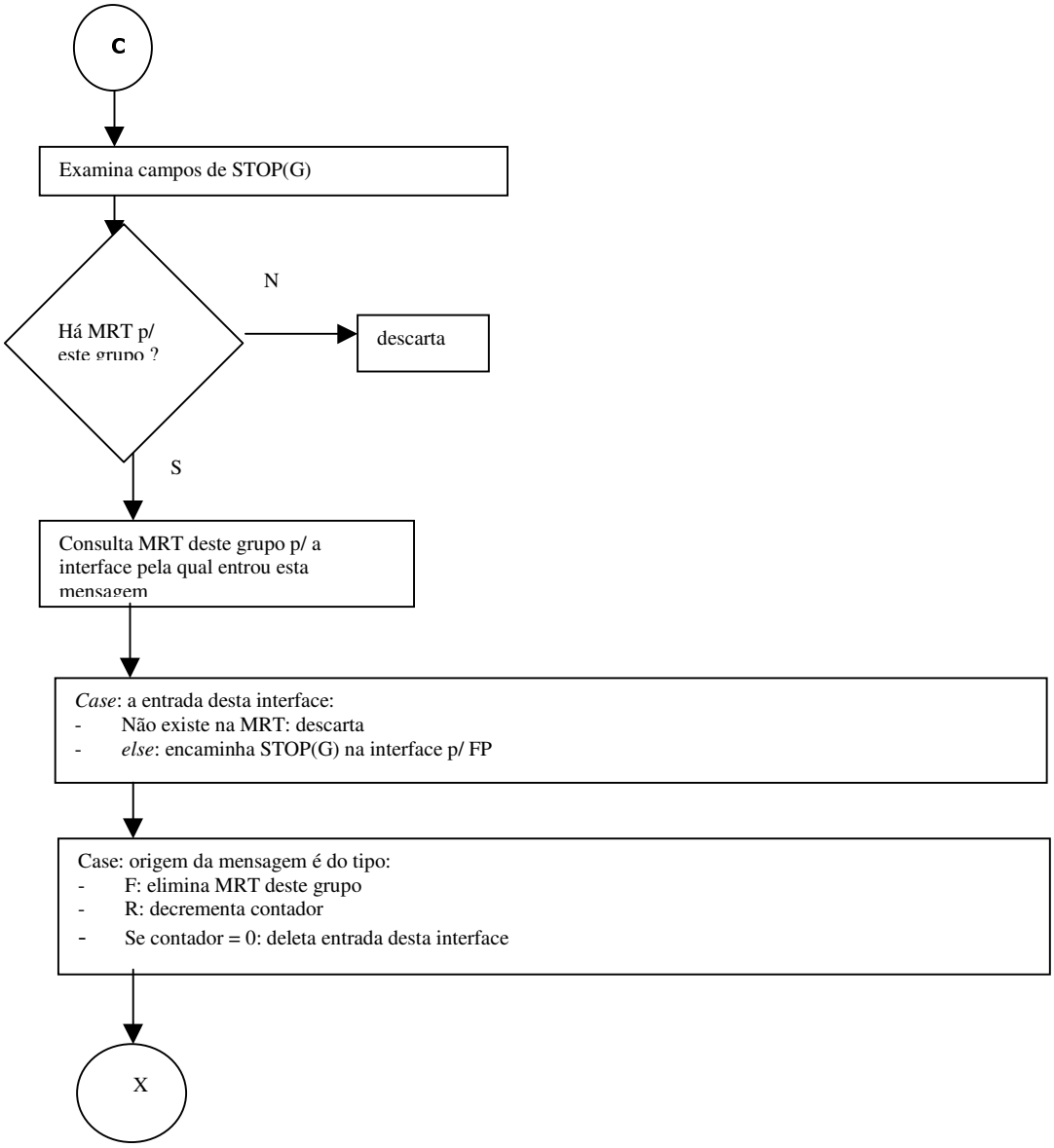


Figura 4.5 - Diagrama de funcionamento dos Roteadores Intermediários (4/4).

4.2.2 – Roteador *Focal Point* (FP)

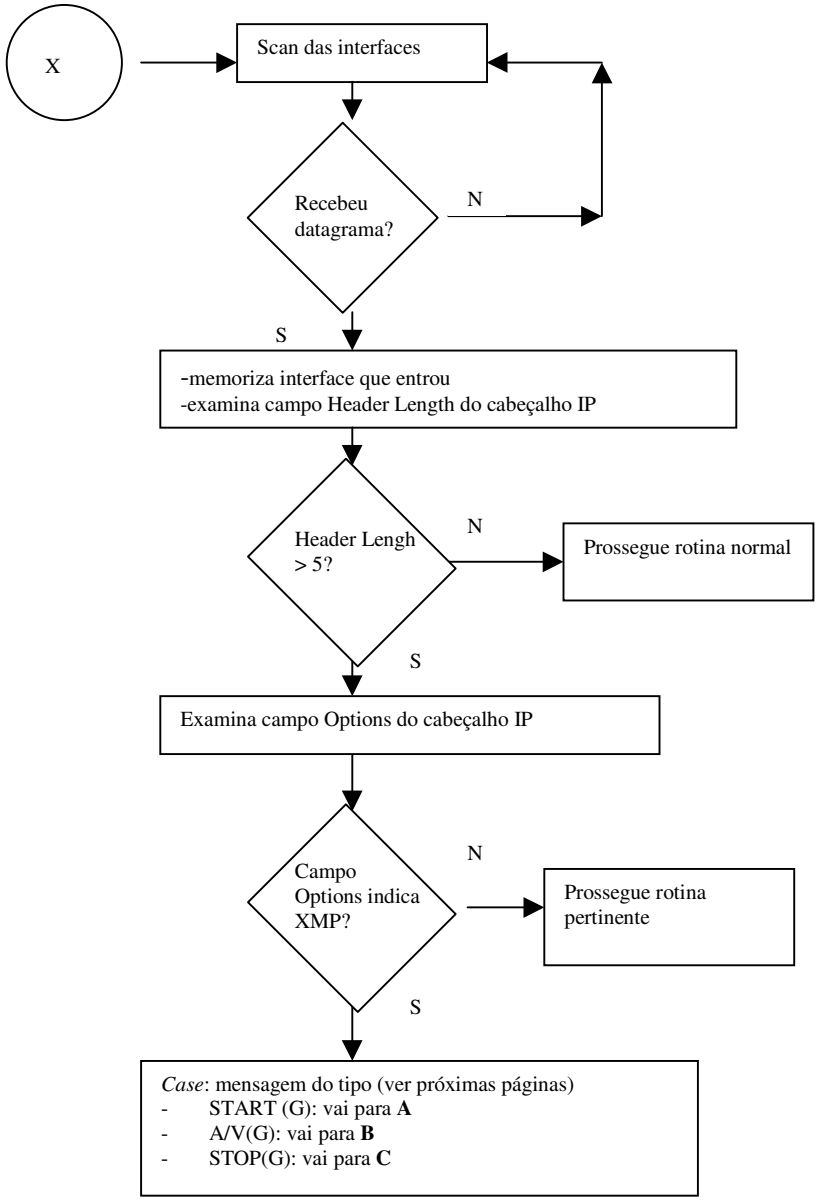


Figura 4.6 - Diagrama de funcionamento do Roteador *Focal Point* (1/4).

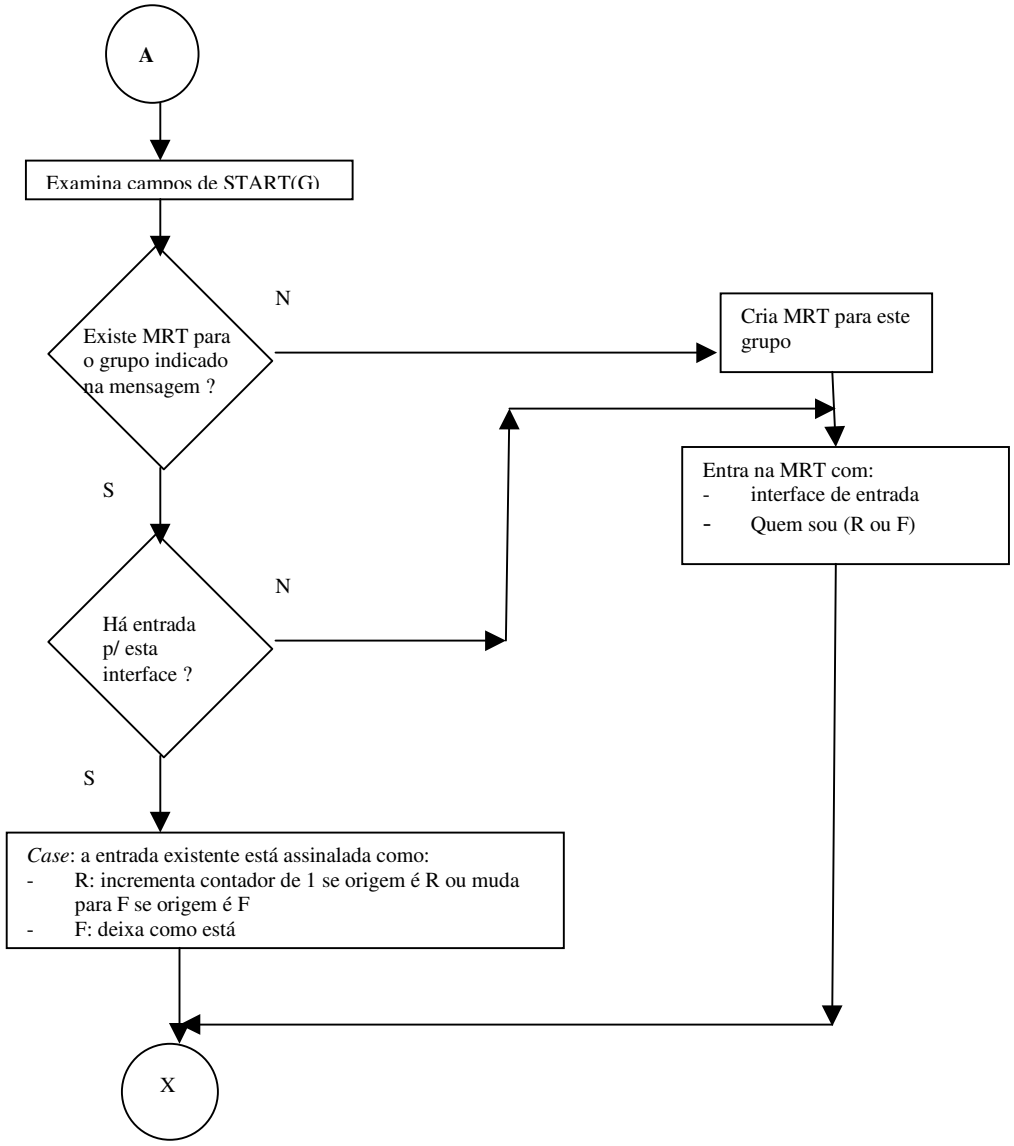


Figura 4.7 - Diagrama de funcionamento do Roteador *Focal Point* (2/4).

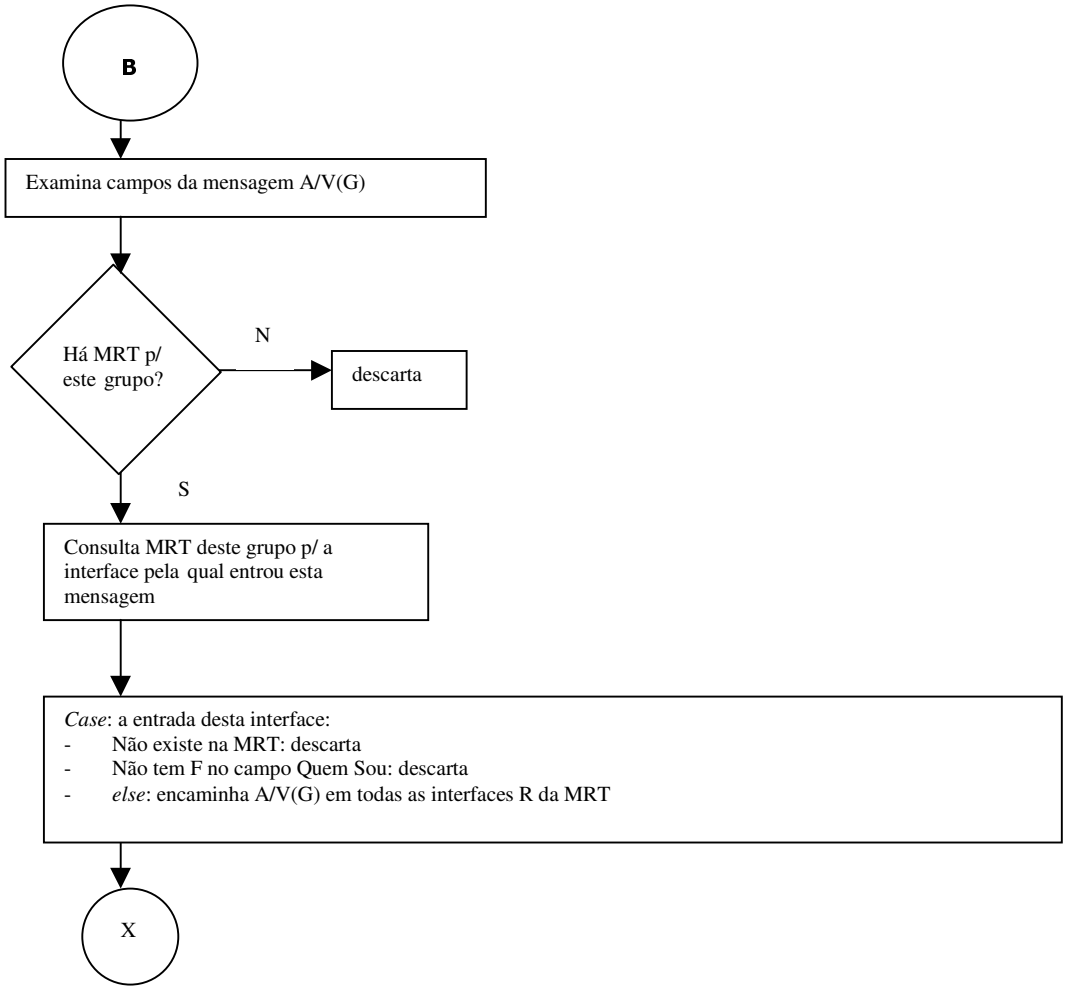


Figura 4.8 - Diagrama de funcionamento do Roteador *Focal Point* (3/4).

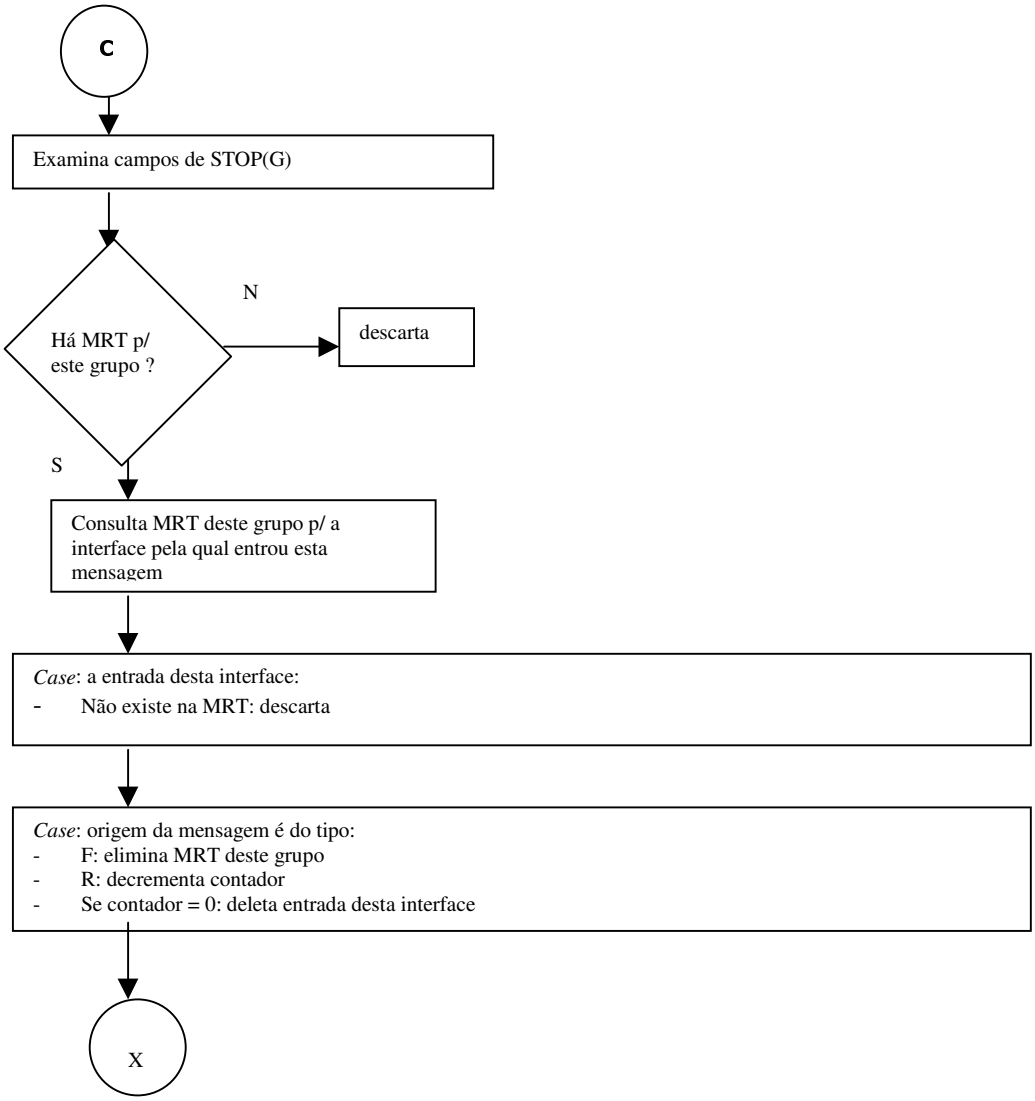


Figura 4.9 - Diagrama de funcionamento do Roteador *Focal Point* (4/4).

4.3 – Verificação do XMP

Na impossibilidade de obter um laboratório com roteadores e *switches* para implementar o XMP, verificar seu funcionamento e observar seu desempenho, optamos por utilizar uma simulação, uma vez que métodos matemáticos poderiam fornecer o desempenho da rede rodando o XMP, mas não verificariam sua funcionalidade. Foram elaboradas as simulações de redes com variadas topologias, cargas e tipos de tráfego. Nesta tese apresentamos somente a simulação mais significativa, que é a da rede corporativa da EMBRATEL por ser nossa conhecida e onde poderiamos obter medições do seu tráfego usual. Essa metodologia foi adotada porque, além da verificação do funcionamento do protocolo, é possível estudar o impacto que a rede receberá com a nova implementação, em relação à utilização e atraso de pacotes. O simulador foi desenvolvido sobre o pacote de *software* ns-2 [33].

4.3.1 - O Simulador ns-2

O ns-2 é um simulador de eventos discretos que objetiva a pesquisa de Redes de Computadores, notadamente Redes IP. O ns provê um suporte substancial para a simulação de protocolos, roteamento, e protocolos *unicast* e *multicast* tanto sobre redes convencionais como sobre redes *wireless* , sejam elas locais ou via satélite.

O ns-2 começou a ser difundido em 1989 como uma variação do simulador de redes REAL e tem evoluído substancialmente nos últimos anos. Em 1995, o desenvolvimento do ns-2 foi patrocinado pelo DARPA (*Defense Advanced Research Projects Agency*), através do projeto VINT (*Virtual InterNetwork Testbed*) na LNBL (*Lawrence Berkeley National Laboratory*), Xerox PARC (*Palo Alto Research Center*), UCB (*University of California at Berkeley*) e

USC/ISI (*University of Southern California / Information Sciences Institute*). Atualmente o desenvolvimento do ns-2 é patrocinado pelo DARPA / SAMAN (*Simulation Augmented by Measurement and Analysis for Networks*) e através do NSF (*National Science Foundation*) com a CONSER (*Collaborative Simulation for Education and Research*), e em colaboração com outros pesquisadores [33].

Apesar de existirem outros simuladores como, por exemplo, o OPNET que é muito caro, o ns-2 tem sido o preferido pelas comunidades acadêmicas em todo o mundo pelos motivos abaixo:

- Documentação: o ns-2 tem um manual bastante detalhado que é mantido pelos desenvolvedores e atualizado regularmente, além de vários tutoriais disponíveis na Internet.

- Grande número de usuários: uma vez que é usado nas universidades, é fácil perceber que o ns-2 tem a colaboração de muitos profissionais que estão constantemente publicando novas simulações. Já estão disponíveis *add-ons* para *wireless* (802.11b), satélite, MPLS e LAN.

- Lista de discussão: o ns-2 *Mailing List* transforma inúmeros usuários do ns-2 em todo o mundo em colaboradores que contribuem para trocar idéias, tirar dúvidas e solucionar problemas.

- Integração entre OTCL (linguagem interpretada) e C++ (linguagem compilada): a parte da simulação que processa um número muito grande de pacotes, e requer que isto seja feito rapidamente, é codificada em C++. A parte de estabelecimento das estruturas e da topologia é feita em OTCL que, por ser uma linguagem de mais fácil uso que o C++, permite uma grande

flexibilidade de alteração de protocolos e topologias, simulando vários cenários com rapidez. Se um código em OTCL ficar crítico para o desempenho da simulação, é conveniente migrar esta parte do código para C++. Há um exemplo na literatura que aponta um ganho de 50% na velocidade de processamento da simulação quando uma parte crítica do código em OTCL foi convertida para C++ [33].

Depois que começamos a trabalhar com o ns-2, ficou evidente que simular um roteador seria uma tarefa excessivamente consumidora de recursos uma vez que uma rede de, digamos, 10 roteadores, necessitaria da capacidade de computação de 10 máquinas rodando simultaneamente. Simular uma rede usando um computador pessoal de preço acessível requer uma representação que proporcione uma aproximação aceitável da realidade sem, contudo, executar suas funções na íntegra.

O "pacote" (datagrama) do simulador não é o mesmo do mundo real. Por exemplo, o cabeçalho IP contém somente os endereços IP de origem e destino e o TTL (*Time to Live*). Além dos cabeçalhos usuais, ou seja, IP, TCP ou UDP, RTP, etc., o simulador trabalha com todos os outros tipos de cabeçalho, ainda que não sejam usados. Isto é, a menos que nós explicitamente os retiremos da simulação. Há, entretanto, um cabeçalho que não pode ser retirado, que é o *Common Header*, que contém informações importantes para o *trace* da simulação, tais como Tipo de Pacote, Tamanho do Pacote, um identificador único para cada pacote (atribuído automaticamente na criação do mesmo) e o campo *time-stamp,* que é usado para medir atrasos nas filas. Há também o campo *Label* do *Link*, no caso de simulação *multicast*.

No entanto, o ns-2 é poderoso quando queremos introduzir novas funcionalidades.

a) Componentes

Os 3 principais componentes do ns-2 são: O *Node*, o *Link* e o *Agent*. Os *Nodes* representam os *hosts* e os roteadores. Os *Nodes* são conectados uns aos outros através dos *Links*, formando a topologia desejada. Os *Agents* representam os Terminais da Rede, onde os pacotes são produzidos e/ou consumidos. Os *Agents* podem efetuar, dentro dos Roteadores, funções de controle. Os *Agents* são incorporados aos *Nodes* (sejam eles Terminais ou Roteadores) e identificados por um endereço que consiste de um identificador do nós (*Node* ID) e um número de Porta. Normalmente, um *Node* roteador não possui agentes sendo seu papel, tão somente, de encaminhar os pacotes recebidos de acordo com alguma lógica de roteamento preestabelecida.

Hosts se conectam a roteadores, que se conectam a outros roteadores que, ao final,

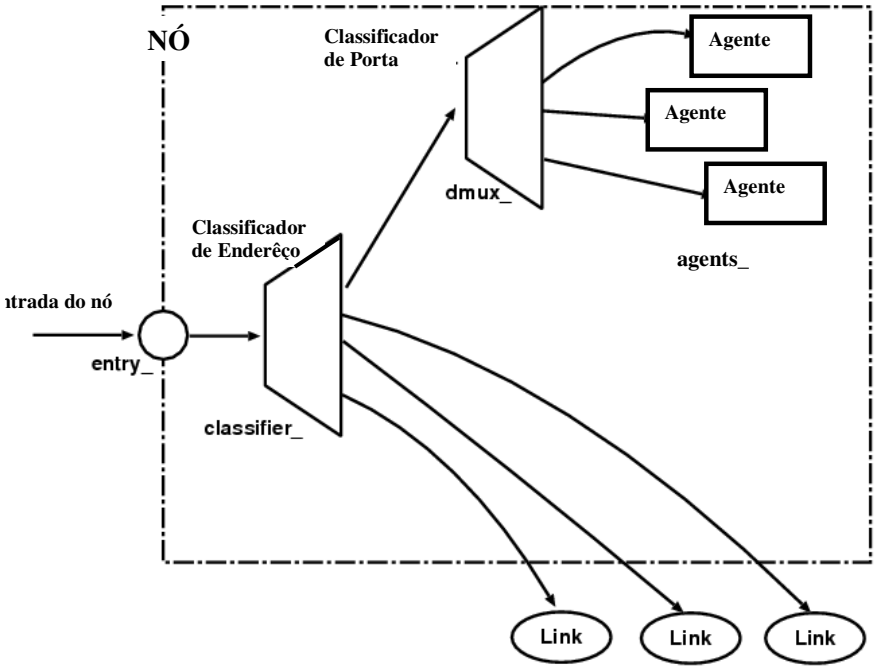


Figura 4.10 - Diagrama em blocos de um *Node Unicast* no ns-2.

se conectam a outros *hosts*.

O ***Node*** é composto por *Classifiers* e *Agents* (e *Replicators* no caso de *multicast*), conforme ilustrado nas Figuras 4.10 e 4.11.

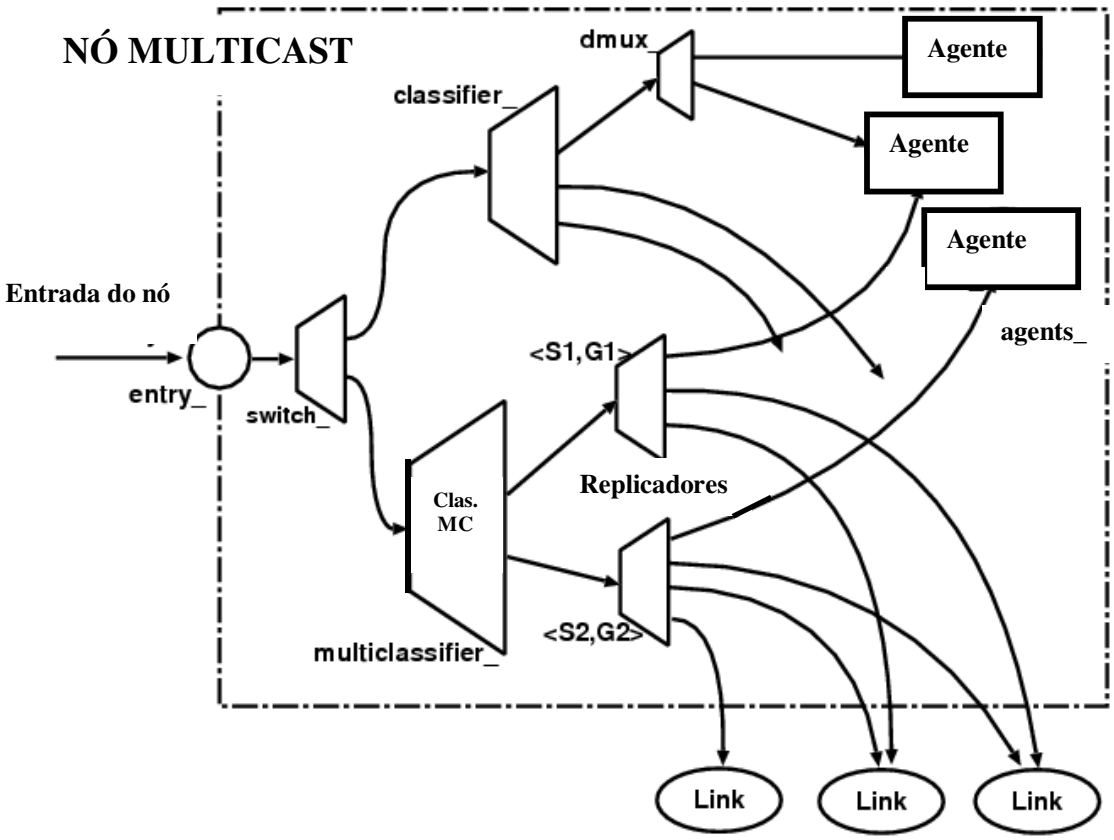


Figura 4.11 - Diagrama em blocos de um *Node Multicasting* no ns-2.

O ***Link*** é composto pelos objetos *queues*, *links*, *ttl* e *drophead*, além dos objetos responsáveis pela coleta de dados sobre a passagem do pacote no *Link* (Fig. 4.12).

Notar que *Port Classifier* é o demux_ que envia o pacote para o Agente pertinente.

O *Addr Classifier* (classifier_) envia o pacote para o *Node* seguinte no sentido *downstream* da transmissão, ou para o demux_ (Fig 4.11).

Note-se a presença do classificador *Switch* na entrada do *Node*. Se o endereço vier com o bit mais significativo igual a 1 é porque é um endereço *multicasting* e o *Switch* envia o pacote para o *Multicast Classifier*. Caso contrário é uma transmissão *unicast* e o pacote é enviado para o *Address Classifier* (classifier_). O demux_ se encarrega de enviar para o *Agent* pertinente. O objeto Gerenciador de Roteamento (rtObject) não está representado para não sobrecarregar a figura. <S1, G1> são representações do protocolo PIM que indicam a fonte S1 transmitindo no grupo G1.

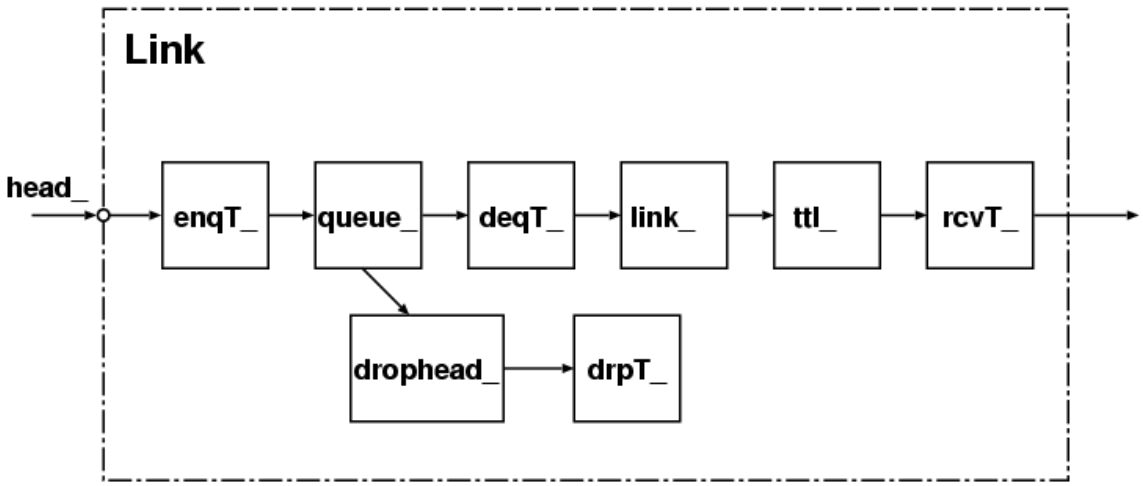


Figura 4.12 - Diagrama em blocos do Objeto *Link* no ns-2.

b) Roteamento no ns-2

As funcionalidades do roteador são implementadas em partes diferentes da rede. As tabelas de roteamento são mantidas por objetos de roteamento dentro da classe Simulador (classe RouteLogic). No caso de roteamento dinâmico, estas tabelas estão dentro de cada nó (classe rtObject).

Dentro do nó, um pacote é recebido e processado em um objeto *Classifier*. *Classifiers* são componentes com uma única entrada de pacotes e uma ou mais saídas. Eles analisam os pacotes entrantes e os passam para o próximo objeto *downstream* que pode ser outro Classificador, um *Link* ou um Agente.

Alguns tipos de Classificadores são:

- *Address Classifiers*, que examinam o campo Destino do pacote e são usados, principalmente, para encaminhamento *unicast*.

- *Port Classifiers*, que são semelhantes ao anterior mas destinam os pacotes recebidos para agentes e não para *Links*.

- *Multicast Classifiers*, que são instalados nos *Nodes* para simulação de transmissões *multicast*. Fazem uso dos *Replicators* (REPs). No mais, sua função é bastante semelhante ao *Address Classifier*, ou seja, encaminhar o pacote recebido baseado no endereço de destino do cabeçalho do pacote.

- *Replicators*, que simplesmente enviam cópias do pacote em todas suas saídas sem análise.

- *Multipath Classifiers*, que não examinam nenhum campo dos pacotes entrantes. Os pacotes são encaminhados para todas as suas saídas em um modo *round-robin*. Isto é útil quando se quer simular um roteador que tem múltiplas rotas para o destino, dividindo a carga da transmissão entre todas as rotas.

- *Hash Classifiers*, que são usados quando os pacotes devem ser encaminhados de acordo com o valor de um ou mais campos do cabeçalho do pacote.

Apesar de já haver nós *unicast* e *multicast* pré-definidos é possível alterar a estrutura interna de um *Node* através de métodos existentes nas classes.

c) *Tracing*

Em sua execução, o ns-2 pode acumular uma série de informações em um arquivo de *trace*. Esta facilidade deve ser habilitada na elaboração do *script* que chama o *simulator* e monta a topologia, os agentes e aplicações que se deseja.

O ns-2 registra cada pacote, quando ele entra, quando é encaminhado ou quando é retirado da fila em um *link* ou *queue*.

Para se processar os dados contidos no arquivo de *trace*, é recomendado utilizar o programa AWK que varre o arquivo, linha por linha, separando e executando alguma função, de acordo com o conteúdo de cada linha.

Um exemplo de arquivo *trace* pode ser visto na Listagem 1 (Anexo I).

d) Nam

O nam é a parte do ns que proporciona uma interface gráfica para visualização da topologia e do trânsito dos pacotes na rede. É uma ferramenta de animação desenvolvida em Tcl/Tk (*Tools Command Language/Tools Kit*). O nam também precisa ser, explicitamente, habilitado no *script* da simulação (em OTCL). Ele gera um arquivo de *trace*, diferente do ns-trace mencionado anteriormente, pois sua finalidade é fornecer dados para a geração da animação gráfica e não fazer medições e estatísticas.

Os comandos que habilitam tanto o *trace* do ns-2 quanto do nam, bem como as chamadas ao nam para rodar após o simulador ter executado a simulação, podem ser vistas na Listagem 2, em negrito, no Anexo I.

No ns-2, as funções básicas de uma rede IP já estão desenvolvidas (UDP, TCP, MPLS, satélite, w*ireless*), e podem ser incorporadas à simulação pretendida com um mínimo de esforço. O ns-2 usa C++ para construir o núcleo do pacote de *software* (para operações que são executadas muitas vezes e têm que ser rápidas) e usa Otcl (*Object Tools Command Language*) para implementar as funções menos críticas e para construir a topologia da rede em estudo, bem como estabelecer várias conexões entre nós da rede, gerando um arquivo (*trace*) que é usado para se medir parâmetros usuais em redes de computadores como atraso na propagação dos pacotes (W – *waiting time*), utilização (U – *utilization*), tamanho de fila (n - população) e vazão (X - *throughput*).

O trabalho necessário para a inclusão de um novo protocolo no ns-2 é intenso, sendo necessário elaborar programas em C++ e em Otcl para construir novos cabeçalhos, novos tipos de *Nodes* e de *Agents*. Após esta fase, usa-se Otcl para escrever os *scripts* que irão compor a topologia desejada, os *hosts* geradores de pacotes e os *hosts* que os vão consumir. Redes podem ser construídas com grande facilidade a partir desses elementos.

Para verificação do protocolo proposto, foram simuladas 5 redes de complexidade crescente, com variados tráfegos e topologias. Decidimos apresentar nesta Tese a Rede Corporativa da EMBRATEL (não a Rede Pública na qual esta empresa comercializa seus serviços) por sua complexidade e por ter sido possível efetuar medições na rede real que ajudaram na simulação e proporcionaram comparação.

A topologia desta Rede será mostrada mais adiante.

4.3.2 – A Construção do XMP no ns-2

Antes de começar a codificação, é necessário ter uma perfeita visão de todos os estados do protocolo que podem ocorrer nos *hosts* e roteadores.

Por este motivo, foi construida a TABELA DE ESTADOS, a seguir, onde foram retratadas todas as possíveis transições de estado para cada tipo de pacote entrante. Com base nesta tabela foi, então, codificado o XMP. A princípio, foram escritas linhas de programa para cada tipo de Node (*host* receptor, *host* emissor, roteador focal e roteador intermediário). Numa etapa seguinte, foi elaborado um único código que pode ser usado para cada tipo acima. Este código pode ser visto na Listagem 5 do Anexo I.

Em seguida, foi escrito o código de Xmp.h e Xmp.cc. O código gerado em C++ foi para capturar os pacotes XMP passando-os para serem processados em Otcl uma vez que as mensagens de comando e a construção das tabelas MRT eram tarefas executadas poucas vezes e não requeriam grande velocidade de execução. Os pacotes XMP, com amostras de áudio/vídeo (tráfego CBR), ficou sendo executada por código em C++, já oferecido pelo simulador.

TABELA DE ESTADOS DO PROTOCOLO XMP

(PARA CADA GRUPO)

Nó de Roteador (R ou FP):

REP – Replicador, MC – Multicast Classifier, IF – interface, IFFP – Interface na direção de FP
OLDIIF – memória do estado da interface (F ou R) na transição da Fonte
? – estado indefinido da Interface até que chegue o primeiro pacote

Estado da IF (OFF/F/R)	Existe Rep? (S/N)	Tipo de Pacote (Start/Stop, F/R)	R Counter	Ação
OFF	N	Start + F	=0	- Cria REP - Põe REP no MC para grupo G e interface IF - Se R: - Coloca IFFP na saída do REP - Incrementa contador p/ IFFP
OFF	N	Start + F	>0	- Ignora
OFF	N	Start + R	=0	- Cria REP - Se R: - Põe REP no MC para grupo G e interface IFFP - Se FP: - Põe REP no MC para grupo G e interface **?** - Coloca IF na saída de REP - Incrementa contador p/ IF
OFF	N	Start + R	>0	- Ignora
OFF	N	Stop + F	=0	- Ignora

Estado da IF (OFF/F/R)	Existe Rep? (S/N)	Tipo de Pacote (Start/Stop, F/R)	R Counter	Ação
OFF	N	Stop + F	>0	- Ignora
OFF	N	Stop + R	=0	- Ignora
OFF	N	Stop + R	>0	- Ignora
OFF	S	Start + F	=0	- Redireciona REP no MC para grupo G e interface IF - Se OLDIIF != ?: - Redireciona OLDIIF para a saída do REP - Incrementa contador p/ OLDIIF
OFF	S	Start + F	>0	- Ignora
OFF	S	Start + R	=0	- Coloca IF na saída de REP - Incrementa contador p/ IF
OFF	S	Start + R	>0	- Ignora
OFF	S	Stop + F	=0	- Ignora
OFF	S	Stop + F	>0	- Ignora
OFF	S	Stop + R	=0	- Ignora
OFF	S	Stop + R	>0	- Ignora
F	N	Start + F	=0	- Ignora
F	N	Start + F	>0	- Ignora
F	N	Start + R	=0	- Ignora
F	N	Start + R	>0	- Ignora
F	N	Stop + F	=0	- Ignora
F	N	Stop + F	>0	- Ignora
F	N	Stop + R	=0	- Ignora

Estado da IF (OFF/F/R)	Existe Rep? (S/N)	Tipo de Pacote (Start/Stop, F/R)	R Counter	Ação
F	N	Stop + R	>0	- Ignora
F	S	Start + F	=0	- Ignora
F	S	Start + F	>0	- Ignora
F	S	Start + R	=0	- Incrementa contador p/ IF
F	S	Start + R	>0	- Incrementa contador p/ IF
F	S	Stop + F	=0	- Remove REP
F	S	Stop + F	>0	- Remove REP
F	S	Stop + R	=0	- Ignora
F	S	Stop + R	>0	- Decrementa contador p/ IF
R	N	Start + F	=0	- Ignora
R	N	Start + F	>0	- Ignora
R	N	Start + R	=0	- Ignora
R	N	Start + R	>0	- Ignora
R	N	Stop + F	=0	- Ignora
R	N	Stop + F	>0	- Ignora
R	N	Stop + R	=0	- Ignora
R	N	Stop + R	>0	- Ignora
R	F	Start + F	=0	- Ignora
R	S	Start + F	>0	- Redireciona REP no MC para grupo G e interface IF - Decrementa contador p/ IF - Se OLDIIF != ?: - Redireciona OLDIIF para a saída do REP

Estado da IF (OFF/F/R)	Existe Rep? (S/N)	Tipo de Pacote (Start/Stop, F/R)	R Counter	Ação
				- Incrementa contador p/ OLDIIF - Propaga pacote por OLDIIF
R	S	Start + R	=0	- Ignora
R	S	Start + R	>0	- Incrementa contador p/ IF
R	S	Stop + F	=0	- Ignora
R	S	Stop + F	>0	- Ignora
R	S	Stop + R	=0	- Ignora
R	S	Stop + R	>0	- Decrementa contador p/ IF - Se contador = 0, remove IF da saída de REP

TABELA DE ESTADOS DO PROTOCOLO XMP

(PARA CADA GRUPO)

Nó de Usuário (Fonte ou Receptor):

REP – Replicador, MC – Multicast Classifier, IF – interface, IFFP – Interface na direção de FP

Estado da IF (OFF/F/R)	Existe Rep? (S/N)	Tipo de Pacote (Start/Stop, F/R)	Ação
OFF	N	Start + F	- Cria REP - Põe REP no MC para grupo G e interface –1 (ele mesmo) - Insere interface IFFP na saída do REP
OFF	N	Start + R	- Cria REP - Põe REP no MC para grupo G e interface IF - Insere agente de áudio/vídeo na saída do REP
OFF	N	Stop + F	- Não transmite pacote
OFF	N	Stop + R	- Não transmite pacote
OFF	S	Start + F	- Ignora
OFF	S	Start + R	- Ignora
OFF	S	Stop + F	- Ignora
OFF	S	Stop + R	- Ignora
F	N	Start + F	- Ignora
F	N	Start + R	- Ignora
F	N	Stop + F	- Ignora
F	N	Stop + R	- Ignora

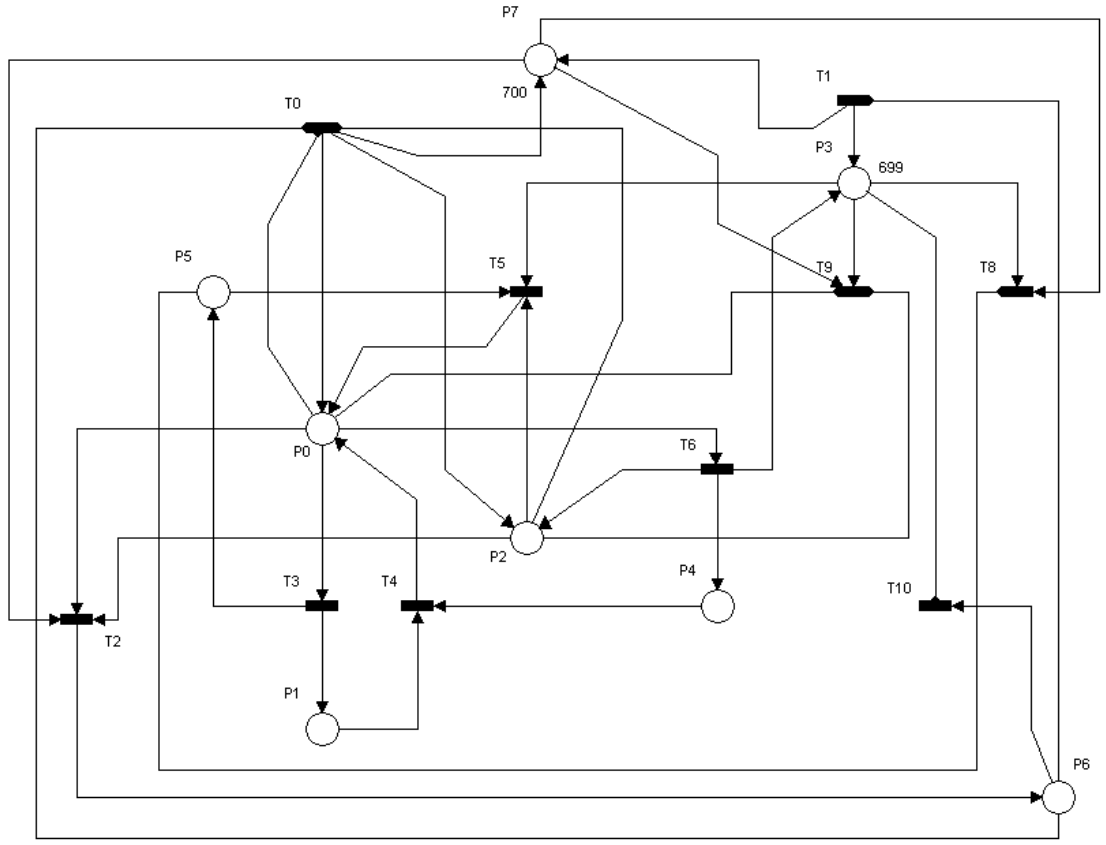
F	S	Start + F	- Não transmite pacote
F	S	Start + R	- Não transmite pacote
F	S	Stop + F	- Remove REP
F	S	Stop + R	- Não transmite pacote
R	N	Start + F	- Ignora
R	N	Start + R	- Ignora
R	N	Stop + F	- Ignora
R	N	Stop + R	- Ignora
R	S	Start + F	- Redireciona REP no MC para grupo G e interface -1 - Remove agente de áudio/vídeo da saída do REP - Insere interface IFFP na saída do REP
R	S	Start + R	- Não transmite pacote
R	S	Stop + F	- Não transmite pacote
R	S	Stop + R	- Remove REP

Quando ocorre mudança de Fonte, o pacote START+F segue em *unicast* o caminho determinado pela antiga interface de entrada IIF de cada replicador. Por fim a Fonte original recebe este pacote e deve alterar sua configuração interna (MC e REP) para a configuração de Receptor, ou seja, redireciona REP no MC para grupo G e interface IFFP:

- Remove interface IFFP da saída do REP

- Insere agente de áudio/vídeo na saída do REP

Os estados das interfaces foram, também, representados por uma Rede de Petri, conforme Figura 4.13, adiante. O simulador usado foi o **HPetriSim,** de Henryk Anschuetz**,** que testa automaticamente todos as situações possíveis e não detetou qualquer anormalidade.



Transição	Ação
T0	Chegada do fonte/moderador
T1	Chegada de receptores
T2	Saída do moderador
T3	Moderador vai para estado receptor
T4	Moderador volta para estado fonte
T5	Receptor vai para estado fonte
T6	Receptor volta para estado normal
T7	Saída de receptores durante conferência

T8	Esvazia receptores
T9	Habilita nova conferência

Estado	Descrição
P0	Fonte
P1	Moderador em estado receptor
P2	Saída do moderador ou entrada de receptor em estado fonte habilitada
P3	Receptores
P4	Retorno do moderador para estado fonte habilitada
P5	Entrada de novo receptor no estado fonte habilitada / Desabilitada a saída de receptores até que o receptor que solicitou ser fonte assuma a posição de fonte
P6	Entrada de novos receptores/fonte proibida até esvaziamento completo da tabela (estado P3)
P7	Contador de usuários na interface

Figura 4.13 - Rede de Petri para os estados das interfaces.

4.3.3 - A REDE CORPORATIVA DA EMBRATEL

Na Figura 4.16 vemos a topologia da Rede Corporativa da EMBRATEL aparecendo, inclusive, os *links* de contingência. Eles têm capacidade menor (são geralmente de 2 Mbps) e só são usados em caso de falha no enlace principal.

As localidades servidas pela Rede Corporativa estão divididas em três níveis de acordo com o volume de tráfego:

Nível 1: São Paulo, Rio de Janeiro, Curitiba, Brasilia e Recife.

Nível 2: Florianópolis, Porto Alegre, Belo Horizonte, Campinas, Vitória, Salvador, Fortaleza e Campo Grande.

Nível 3: demais localidades onde a EMBRATEL tem presença, tais como Uruguaiana, Santo Angelo, Passo Fundo, Pelotas, Santa Maria, Novo Hamburgo e Caxias do Sul (que concentram em Porto Alegre); Criciuma, Joaçaba, Lages e Chapecó (que concentram em Florianópolis); e Londrina, Maringa, Cascavel, Foz do Iguaçu, Ponta Grossa e Paranaguá (que concentram em Curitiba), por exemplo. Estas localidades estão representadas por um único nó cujo tráfego é semelhante ao tráfego agregado de todas as localidades de sua região.

Nas localidades de Nível 1 são empregados roteadores Cisco 7507, com 64 Mbytes de RAM.

Nas localidades de Nível 2 são empregados roteadores Cisco 4700, com 32 Mbytes de RAM.

Nas localidades de Nível 3 são empregados roteadores Cisco 2500, com 8 Mbytes de RAM.

A disciplina de tratamento dos pacotes configurado nas filas dos roteadores é FIFO (*First In First Out*).

O tráfego nesta rede é composto por:

Transferência de arquivos

Aplicações *Web*

Correio eletrônico *Lotus Notes*

Datamining (SAP-R3, SIEBEL e outros desenvolvidos internamente)

Protocolos de controle da rede (DHCP, WINS, DNS, etc.)

Os servidores de *Datamining* ficam no Rio de Janeiro e os Servidores *Web* ficam em São Paulo.

Os Servidores *Notes* ficam nas cidades de São Paulo, Rio de Janeiro, Curitiba, Brasília, Recife e Belo Horizonte. Além disto, o Rio de Janeiro abriga, ainda, o MTA (*Message Transport Agent*) e um *Hub* para onde convergem todos os servidores deste serviço. O MTA se conecta à Internet através de um *Firewall*.

Estes servidores atendem as cidades de sua região. Por exemplo, Curitiba atende a Florianópolis e Porto Alegre, bem como as localidades do interior dos estados de Paraná, Santa Catarina e Rio Grande do Sul, onde a EMBRATEL tem presença.

O MTU (*Maximum Transmission Unit*) usado é de 1500 bytes.

O tráfego foi medido através do IBM *Resource Usage Monitoring*, gerando gráficos da Vazão de entrada e saída de cada roteador, além de valores de utilização e vazão máximo, médio e corrente. Foram gerados gráficos diários, semanais, mensais e anuais.

As Figuras 4.14 e 4.15, adiante, ilustram os gráficos mencionados que, sendo em grande número, não são apresentamos neste trabalho, mas tão somente as médias de utilização que foram usadas na simulação.

O tráfego da Rede Corporativa EMBRATEL é predominantemente sobre TCP sendo que DNS, WINS e NETBIOS são transportados sobre UDP. O protocolo de roteamento usado é o OSPF. A Área 0 (zero) deste protocolo abrange as localidades de Nível 1. A Área 1 abrange as localidades de Belo Horizonte e Vitória, e adjacências, com borda na localidade Rio de Janeiro. A Área 2 abrange as localidades de Campinas, e adjacências, com borda em São Paulo e Curitiba. A Área 3 abrange a localidade de Campo Grande, e adjacências, com borda em Recife e Brasília. Há também o serviço de DHCP trafegando na rede.

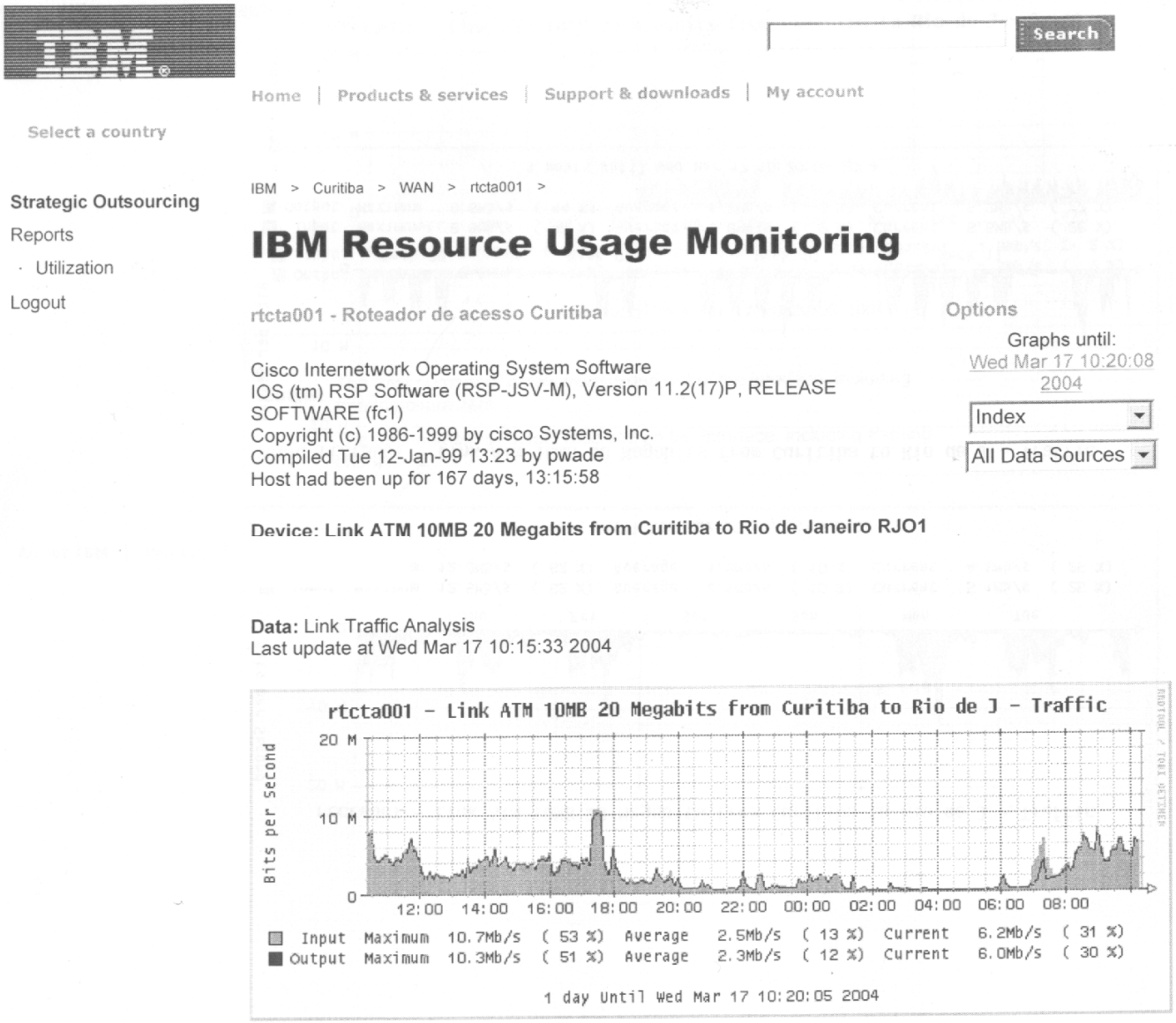


Figura 4.14 – Exemplo de Gráfico de Utilização.

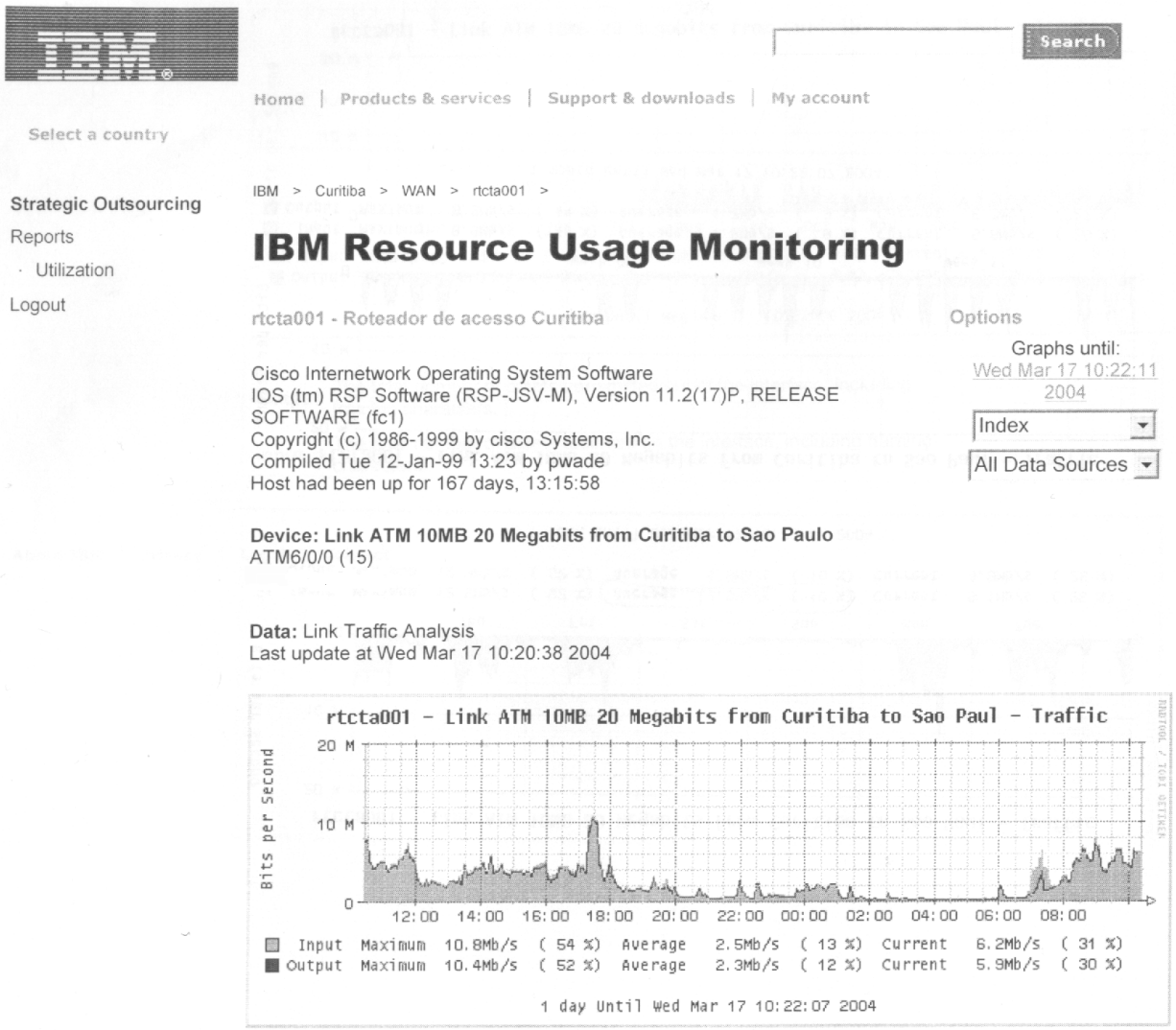


Figura 4.15 – Exemplo de Gráfico de Utilização

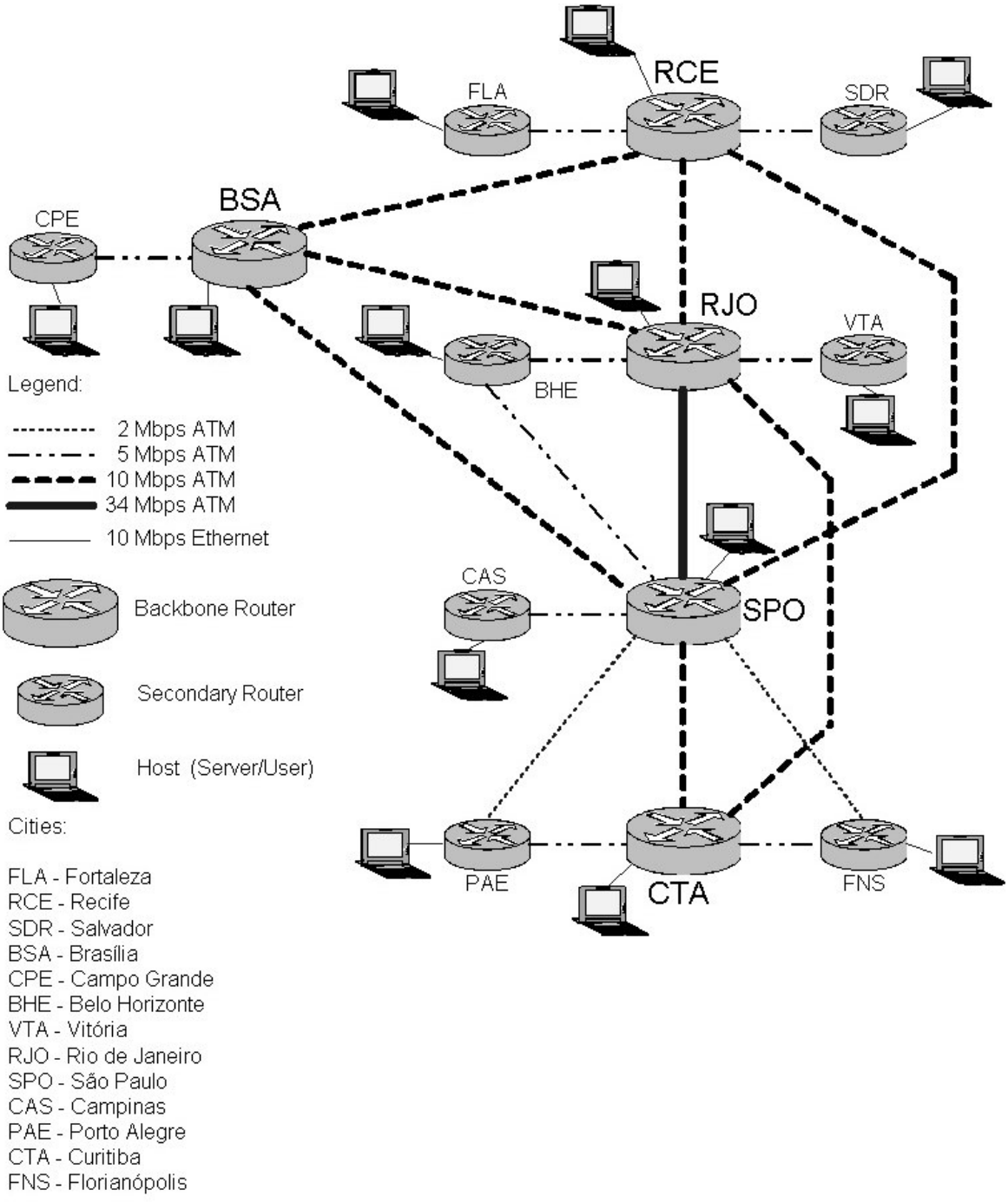


Fig. 4.16 - Topologia da Rede Simulada.

4.3.4 – RESULTADOS DA SIMULAÇÃO

Com a simulação, foi possível testar cada uma das 5 fases (de 1 a 5) do XMP, tendo sido possível observar seu correto funcionamento através do cenário de animação propiciado pelo NAM.

A simulação teve dois objetivos básicos:

- a) verificar o funcionamento do XMP do ponto de vista qualitativo, ou seja, se ele se comporta da maneira para a qual foi projetado, com o tráfego CBR sendo gerado a partir de uma máquina qualquer (Professor) e recebido por diversas máquinas (ALUNOS); verificar se o mecanismo de comutação do sentido de transmissão (aluno toma a transmissão de áudio/vídeo) está funcionando corretamente e se esta comutação é executada rapidamente, ou seja, que não prejudique a sessão *multicast*.

- b) verificar se o XMP provoca impacto na Rede existente e qual a extensão deste impacto, uma vez que ele foi concebido tendo em vista a finalidade de não causar sobrecarga nos enlaces e roteadores.

Estamos apresentando os resultados da simulação da Rede Corporativa da EMBRATEL por ser possível comparar com valores medidos em condições reais de funcionamento.

Usando o gerador de tráfego EXPO *on-off* do próprio simulador, carregamos a rede, *link* por *link*, com um tráfego que resultou em uma utilização muito próxima àquela apresentada pelas medições da rede real. Este tipo de tráfego é adequado uma vez que gera rajadas (o que é típico do tráfego em rede IP) e que se pode configurar o *burst-time*, o *idle-time* e o *burst-rate*. Tanto o tráfego *Web* quanto o de *Datamining* foram simulados assim. Uma vez que a predominância é da arquitetura Cliente/Servidor, foram gerados tráfegos EXPO das pontas remotas (Clientes) para os Servidores, só que com volumes menores uma vez que as solicitações dos Clientes são compostos de poucos *bytes*, enquanto as respostas dos Servidores é volumosa.

Cada localidade tem um roteador onde se conectam, do lado da LAN, vários usuários. Estas LANs operam a 10 Mbps, a menos dos Servidores de Rio de Janeiro e São Paulo que operam a 100 Mbps. O tráfego gerado por todos os usuários foi representado pelo tráfego gerado por um único nó em cada localidade.

O gerador de tráfego EXPO ficou configurado como segue:

- *burst time* = 100 ms para os servidores e 10 ms para os usuários
- *idle time* = 100 ms para os servidores e 100 ms para os usuários
- tamanho do pacote = 1500 bytes para Servidores e 210 bytes para usuários (uma vez que os Servidores respondem com grande volume às solicitações dos Clientes e que o MTU, *Maximum Transmission Unit*, da rede corporativa da EMBRATEL é de 1500 bytes)
- *burst rate* = configurado, no *script* Otcl, individualmente para cada localidade (um valor típico é em torno de 3500 kbps) uma vez que cada localidade gera um volume de tráfego diferente.

Em seguida, geramos um tráfego XMP composto da transmissão dos pacotes de controle (START) e de áudio/vídeo (tráfego CBR com pacotes de 200 bytes e taxa de transmissão de 448 kbps)[4]. Para a finalidade de EaD, a qualidade de vídeo de 384 kbps (com os cabeçalhos dos protocolos envolvidos vai a 448 kbps) tem apresentado um custo/benefício atraente, sendo o preferido dos clientes EMBRATEL. Alguns preferem 128 kbps devido ao preço menor, mas a experiência mostrou que a qualidade de imagem é regular, aparentando um efeito estroboscópico quando o orador move a cabeça, os olhos e a boca. A velocidade de 256 kbps é boa, apesar de mais cara, mas ainda apresenta algumas falhas quando há movimento mais acentuado. A velocidade de 384 kbps não seria adequada para um vídeo onde houvesse intenssos movimentos, tais como um jogo ou corrida de carro. Esta velocidade se mostrou satisfatória para a finalidade de vídeo-conferência. O Simulador foi executado com este tráfego e os valores de utilização e vazão resultantes estão transcritos na Tabela 4.3.4. Naturalmente, a vazão (e, portanto, a utilização) teve seus valores aumentados naqueles *links* que receberam a transmissão *multicast*. Isto se deve a que o tráfego CBR contribuiu de maneira significativa. Um modo de diminuir este impacto é utilizando uma qualidade de vídeo inferior, o que acarreta uma taxa de transferência menor. É preciso notar que a carga de tráfego imposto pelo tráfego de áudio/vídeo (CBR) não depende do protocolo *multicast* empregado, mas tão somente da qualidade de vídeo pela qual se optou.

A geração de eventos está na Listagem 2, do Anexo I, onde os tempos de execução foram da ordem de 10 segundos, pois uma simulação de horas geraria arquivos de *trace* muito grandes

[4] Foi usado CBR uma vez que voz e vídeo precisam ser gerados a taxas constantes.

(tanto do ns quanto do nam), o que não caberia na plataforma por nós usada, conforme consta no Anexo III. Além disto, o tempo para finalizar a simulação seria demasiado longo, a menos que usassemos máquinas de grande capacidade. Não há perda de sua validade, uma vez que todos os eventos usuais em uma sessão deste tipo tiveram suas ocorrências exercitadas, só que numa escala de tempo comprimida. Na realidade, esta compressão no tempo comprova que o protocolo responde bem mesmo em condições de rápido chaveamento entre fontes. Em suma, os eventos simulados são representativos de uma situação real que tem uma duração em torno de uma hora e meia a duas horas.

Pela Listagem 2, vemos que o *script* de funcionamento dos eventos foi escrito para que um Professor no Rio de Janeiro iniciasse a transmissão de CBR (instante 2 segundos após o início da simulação), enquanto Fortaleza e Porto Alegre entravam no grupo aos 2,1 segundos, Salvador entrou aos 2,2 segundos e Campo Grande entrou aos 2,3 segundos. Aos 2,5 segundos, Porto Alegre toma a transmissão enquanto Rio de Janeiro passa à condição de receptor (aluno). Aos 2,6 segundos, Campinas entra já recebendo a transmissão de Porto Alegre, assim como as demais localidades que já estavam assistindo Rio de Janeiro. Aos 3 segundos, Porto Alegre pára de transmitir abruptamente e Rio de Janeiro comanda a retomada da transmissão para si. Aos 3,2 segundos, Porto Alegre abandona o grupo (a aula), enquanto Vitória entra aos 3,5 segundos, assim fazendo também Belo Horizonte e Recife no mesmo instante. Aos 4 segundos, Recife toma a transmissão fazendo, de novo, com que o Rio passe a ser receptor. Campinas abandona aos 4,4 segundos, Salvador aos 4,7 e, aos 4,8 segundos, Recife deixa de transmitir, ao mesmo

tempo em que Rio de Janeiro retoma a condição de transmissor. Belo Horizonte e Vitória abandonam a sessão aos 5,5, segundos seguidos de Campo Grande aos 5,6 segundos e de Rio de Janeiro aos 6 segundos. Sendo transmissor, o Rio de Janeiro está encerrando a sessão. Fortaleza e Recife abandonam a sessão tardiamente.

O próximo passo foi retirar o tráfego CBR (áudio/vídeo), deixando somente o tráfego dos pacotes de controle START e STOP.

Após nova rodada do simulador e do programa BWutil (programa em AWK que mede a utilização dos *links*), verificou-se que não houve aumento mensurável da mesma, que era o que se esperava.

Foi possível comprovar:

1 – o correto funcionamento do XMP sob complexas condições de topologia, tráfego, quantidade de alunos (receptores) e situações de comutação da transmissão. Este funcionamento fica bem visível ao se rodar o nam, a animação do ns-2, quando podemos observar a passagem dos pacotes entre os *Nodes*, tanto os resultantes do tráfego EXPO quanto os resultantes do tráfego CBR. Como os pacotes contendo os comandos do XMP são muito pequenos (3 bytes) foi preciso aumentá-los para 10 bytes de modo que pudessem ser vistos na janela aberta pelo nam.

2 - o impacto na rede existente se deve tão somente ao tráfego CBR (o que acontecerá com qualquer protocolo utilizado), sendo que o XMP tem um impacto desprezível.

Os trabalhos sobre o arquivo de *trace*, gerado pela execução do ns-2, foram feitos utilizando a linguagem AWK. Os principais valores obtidos podem ser vistos na Tabela 7.3.4 onde constam, também, o valor da capacidade de cada *link*.

		Com CBR		**Sem CBR**	
Link	BW(Mbps)	Vazão(Mbps)	Utilização(%)	Vazão(Mbps)	Utilização(%)
PAE-CTA	5	0,67	13,4	0,65	12,9
CTA-PAE	5	3,96	79,32	3,55	71,1
FNS-CTA	5	0,65	13,03	0,65	13,03
CTA-FNS	5	3,90	77,85	3,46	69,16
CTA-SPO	10	1,37	13,70	1,35	13,48
SPO-CTA	10	7,3	73,32	6,92	69,20
SPO-CAS	5	0,71	14,28	0,68	13,56
CAS-SPO	5	3,62	72,31	3,38	67,62
SPO-BSA	10	4,88	48,83	4,74	47,35
BSA-SPO	10	1,08	10,78	1,08	10,78
RJO-BHE	5	3,84	76,72	3,75	74,93
BHE-RJO	5	0,69	13,75	0,69	13,75
SPO-RJO	34	17,86	52,53	17,8	52,36
RJO-SPO	34	16,5	48,40	16,1	47,28

SPO-RCE	10	7,34	73,36	6,9	69,0
RCE-SPO	10	0,88	8,86	0,89	8,86
BSA-CPE	5	3,21	64,22	3,06	61,3
CPE-BSA	5	0,62	12,30	0,62	12,3
RJO-BSA	10	4,58	45,77	4,58	45,8
BSA-RJO	10	0,95	9,48	0,95	9,5
RJO-RCE	10	7,29	72,96	7,29	72,96
RCE-RJO	10	1,51	15,11	1,51	15,11
RJO-VTA	5	3,28	65,56	3,19	63,8
VTA-RJO	5	0,65	13,10	0,66	13,1
RCE-SDR	5	3,93	78,52	3,49	69,7
SDR-RCE	5	0,71	14,20	0,71	14,2
RCE-FLA	5	4,12	82,41	3,68	73,6
FLA-RCE	5	0,59	11,80	0,59	11,8
CTA-RJO	10	1,39	13,87	1,39	13,9
RJO-CTA	10	7,18	71,82	7,18	71,8

TABELA 4.3.4 – Medidas comparativas com o tráfego usual da rede

Obs.: tráfego simulado é, por construção, idêntico ao medido

Como era de se esperar, a saída do tráfego CBR acarretou uma diminuição da Utilização. Lembrar que estamos usando vídeo de alta qualidade (pelo menos em se tratando de uma

conferência) pois usamos 384 kbps que, com o *overhead* do protocolo RTP + UDP + IP, resulta nos 448 kbps mencionados acima.

Em seguida, retiramos o tráfego das mensagens de controle do XMP, Start (G) e Stop (G), tanto dos transmissores quanto dos receptores. Refeita a simulação e efetuadas as medidas da Tabela 4.3.4, os novos valores de Utilização resultaram idênticos aos da coluna "Utilização Atual". A carga adicionada pelo protocolo XMP à rede se revelou desprezível, como se pretendia.

O tempo medido no simulador referente ao intervalo de tempo em que o emissor pára de enviar sinal de áudio/vídeo e passa a recebê-lo, de algum receptor que tenha comutado para transmissor, foi de 21 ms. Apesar deste tempo depender da quantidade de roteadores no caminho do sinal XMP/CBR, ele está bastante satisfatório.

O atraso dos pacotes no *link* mais sobrecarregado (82,41% de Utilização), foi de 5,3 ms (mínimo), 27,8 ms (médio) e de 170,5 ms (máximo). A latência média, medida na rede corporativa foi de 17,8 ms.

A perda de pacotes foi 0%, o que se aproxima das medidas realizadas na rede, que acusou uma perda de pacotes de 0,018% (o máximo admitido pela EMBRATEL em sua rede corporativa e nos serviços que oferece aos clientes é de 1%).

CONCLUSÕES

O desenvolvimento do protocolo *multicast* foi uma enorme evolução no uso da Internet. Em março de 1992 foi feita a primeira reunião do IETF na qual os membros participaram a distância utilizando só o recurso de voz [34]. O grande interesse, a partir daí despertado, exigiu o surgimento de protocolos *multicast* mais elaborados, tais como o PIM-SM e o PIM-DM, pois o DVMRP, então usado, não atendia o caso de muitos usuários em grandes redes, devido ao fato de que era necessário computar uma árvore para cada participante do grupo. Em uma rede de grandes dimensões, com participantes constantemente entrando e saindo do grupo, o DVMRP, bem como o MOSPF, são computacionalmente ineficientes [35].

Na EMBRATEL, no começo dos nossos trabalhos de implantação de uma plataforma para EaD, não havia esta facilidade (*multicast*), o que limitava o número de alunos a algo em torno de 20, devido à sobrecarga imposta pelo *unicast* à rede.

Após as primeiras aulas bem sucedidas, a empresa se interessou pela plataforma e tomou a iniciativa de implantar em todos os roteadores de sua rede corporativa, o protocolo *multicast* PIM-SM. Este fato alavancou sobremaneira o uso de EaD, pois, agora, não havia mais limite para o número de alunos. De fato a mesma plataforma era usada para pronunciamentos da Presidência e Diretores aos cerca de 7.000 empregados da firma.

Entretanto, em nossa necessidade concreta de treinamento corporativo, sentíamos a falta de um protocolo multicast que possibilitasse melhorar a interação entre os participantes,

de modo que todos pudessem ver a todos, ainda que não ao mesmo tempo, conforme já mencionado.

Com isto em mente, resolvemos propor um protocolo *multicast* que suprisse a necessidade por nós sentida na empresa. Durante a elaboração desse protocolo, detectamos a existência do "rascunho" de um protocolo *multicast* bidirecional, o BIDIR-PIM, que estava no IETF para avaliação por parte da comunidade científica. Este protocolo ficou nesta condição até 15 de julho do presente ano, quando foi removido, e não virou RFC. Por ser elaborado com vistas a uma Internet ampla, onde nem todos os roteadores estão capacitados para processar os seus datagramas, há uma sobrecarga para identificar os roteadores vizinhos que seriam usados na formação da rede *multicast*. Além disto, ele trabalha com 3 tabelas para poder tratar as árvores compartilhadas, o que impõe um esforço computacional grande por parte dos roteadores envolvidos.

O protocolo proposto nesse trabalho foi verificado através do uso de um simulador amplamente difundido no meio acadêmico em todo o mundo, o ns-2. A dificuldade de implantar esse protocolo em uma rede real se deu por não ser possível parar uma rede em funcionamento para alterar o software dos roteadores, devido ao grande risco de paralisar a operação da empresa. As instituições que poderiam disponibilizar estes recursos estavam comprometidos com outras pesquisas e não foi possível utilizá-los. A saída foi utilizar simulação. O simulador da própria UNICAMP e do CPqD estavam, também, comprometidos, de modo que foi usado o simulador ns-2. Este oferece um alto grau de confiabilidade dos resultados pois este simulador

tem sido provado e aprovado por muitos pesquisadores, inclusive para redes *wireless* e satélite em universidades do mundo inteiro.

Os resultados da simulação proporcionaram duas informações importantes, ou seja, que o protocolo proposto executa as funções para as quais foi projetado e que o seu funcionamento não sobrecarrega a rede, além do que o chaveamento entre fontes funciona com precisão mesmo em seqüências muito rápidas. Ademais, os códigos gerados aproveitam as funcionalidades já existentes nos roteadores usuais e, por isto, foram codificados em um pequeno número de linhas de código, o que sugere que seu código objeto final é reduzido, fator este essencial para não obrigar o aumento das memórias dos roteadores e/ou a migração para máquinas de maior capacidade, reduzindo os custos para sua adoção.

Dado que só foi possível verificar nossa proposta através do uso de um simulador, trabalhos futuros poderiam incluir a inserção de código nos Sistemas Operacionais dos roteadores para o tratamento deste protocolo proposto, com vistas a o aprimorar em situações reais.

Trabalhos futuros deverão contemplar a implementação do protocolo proposto, em redes baseadas em IPv6, dado que já está ocorrendo uma migração para esta nova versão de plataforma IP.

Um outro campo fascinante de pesquisa que se vislumbra é o de investigar a utilização do protocolo para o uso com realidade virtual, onde, através de óculos especiais ou das novas

técnicas de projeção tri-dimensional que estão sendo desenvolvidas [36], possa-se ter uma sala de reunião, ou de aula, onde os participantes possam se sentir próximos uns dos outros.

O uso do *multicast*, bidirecional ou não, encontra grande uso em transmissões de TV, por exemplo, em lugar (ou complementarmente) à TV a cabo e aberta. Imagine-se que cada emissora de televisão do Brasil tivesse um endereço classe D atribuído, de modo que pudéssemos receber a sua transmissão em nossos computadores domésticos simplesmente selecionando um aplicativo que orientasse a placa de rede a receber aquele endereço classe D particular. Evidentemente, será necessário que o uso de Banda Larga na Última Milha seja mais difundido e ofereça velocidades maiores do que as atuais. Mas, pelo que se lê na literatura especializada, este é um serviço que brevemente estará em nossos lares.

Tendo em vista a importância do *multicasting*, poder-se-ia pesquisar variações do protocolo *multicast* de modo a tirar o máximo proveito das especificidades de determinadas plataformas, tais como meios ópticos (DWDM), MPLS, WiFi, WiMax, etc.

Devido à grande agilidade com a qual este protocolo consegue atuar sobre os componentes de uma rede de computadores é possível usá-lo como um meio de se reconfigurar toda uma rede em tempo mínimo.

Reafirmo a relevância da presente tese, pela significativa colaboração que pode dar aos trabalhos de EaD na empresa e em práticas pedagógicas dentro e fora das instituições educacionais, e pelos caminhos investigativos que podem ser abertos, para ampliar e consolidar novas propostas em EaD.

ANEXO I – LISTAGENS DOS CÓDIGOS FONTE

LISTAGEM 1

Exemplo de arquivo gerado pelo comando *trace-all*, quando da execução do Simulador

Normalmente os arquivos de *Trace* são muito grandes. Por este motivo apresentamos, abaixo, somente as primeiras linhas dele para dar uma idéia de sua construção

Legenda de op:
+ enfileiramento do pacote entrante no *Link*
- saída do pacote da fila
r chegada do pacote no fim do *Link* (entrada do próximo *Node*)

(a linha, abaixo, foi colocada com o propósito de identificar cada coluna, não existindo no *trace*)

op	time	src id	dst id	type	pkt size	flags	flow id	Src addr.port	dst addr.port	seq.	pkt id
+	0.12	1	3	start	3	------	31	1.0	3.0	-1	6
-	0.12	1	3	start	3	------	31	1.0	3.0	-1	6
r	0.1210	1	3	start	3	------	31	1.0	3.0	-1	6
+	0.1210	3	0	start	3	------	31	3.0	0.0	-1	7
-	0.1210	3	0	start	3	------	31	3.0	0.0	-1	7
r	0.1220	3	0	start	3	------	31	3.0	0.0	-1	7
+	0.1225	1	3	cbr	210	------	0	1.1	*.0	6	8
-	0.1225	1	3	cbr	210	------	0	1.1	*.0	6	8
r	0.1236	1	3	cbr	210	------	0	1.1	*.0	6	8
+	0.1236	3	0	cbr	210	------	0	1.1	*.0	6	8
-	0.1236	3	0	cbr	210	------	0	1.1	*.0	6	8
r	0.1248	3	0	cbr	210	------	0	1.1	*.0	6	8
+	0.12625	1	3	cbr	210	------	0	1.1	*.0	7	9

```
-  0.12625 1   3  cbr   210 ------  0    1.1      *.0    7   9
r  0.12741 1   3  cbr   210 ------  0    1.1      *.0    7   9
+  0.12741 3   0  cbr   210 ------  0    1.1      *.0    7   9
-  0.12741 3   0  cbr   210 ------  0    1.1      *.0    7   9
r  0.12858 3   0  cbr   210 ------  0    1.1      *.0    7   9
+  0.13    1   3  cbr   210 ------  0    1.1      *.0    8  10
-  0.13    1   3  cbr   210 ------  0    1.1      *.0    8  10
r  0.13116 1   3  cbr   210 ------  0    1.1     *.0    8  10
+  0.13116 3   0  cbr   210 ------  0    1.1     *.0    8  10
-  0.13116 3   0  cbr   210 ------  0    1.1     *.0    8  10
r  0.13233 3   0  cbr   210 ------  0    1.1     *.0    8  10
+  0.13375 1   3  cbr   210 ------  0    1.1     *.0    9  11
-  0.13375 1   3  cbr   210 ------  0    1.1     *.0    9  11
r  0.13491 1   3  cbr   210 ------  0    1.1     *.0    9  11
+  0.13491 3   0  cbr   210 ------  0    1.1     *.0    9  11
-  0.13491 3   0  cbr   210 ------  0    1.1     *.0    9  11
r  0.13608 3   0  cbr   210 ------  0    1.1     *.0    9  11
+  0.1375  1   3  cbr   210 ------  0    1.1     *.0   10  12
-  0.1375  1   3  cbr   210 ------  0    1.1     *.0   10  12
r  0.13866 1   3  cbr   210 ------  0    1.1     *.0   10  12
+  0.13866 3   0  cbr   210 ------  0    1.1     *.0   10  12
-  0.13866 3   0  cbr   210 ------  0    1.1     *.0   10  12
```

- é igual a –2147483648

Obs.: O valor acima é atribuido, automaticamente, pelo ns-2. A cada grupo criado o ns-2 atribui um número uma unidade maior (já que o número é negativo) que o anterior.

LISTAGEM 2
(*Script* que configura a Rede Corporativa da EMBRATEL)

```
set trfile out.tr
set namfile out.nam
set layfile layout
# Inclui o arquivo ns-xmp, que contem a definicao do protocolo Xmp
source ../ns-xmp.tcl
source ../ns-xmp-aux2.tcl

# Cria o objeto responsavel pelo gerenciamento da simulacao, informando-o
# que havera trafego multicast
set ns [new Simulator -multicast on]

# Configura arquivos de tracing
set f [open $trfile w]
$ns trace-all $f
set nf [open $namfile w]
$ns namtrace-all $nf

# Pacotes de dados do professor = VERMELHO
$ns color 1 red
# Pacotes de dados dos alunos = VERDE ESCURO
$ns color 2 darkgreen
# Pacotes XMP STOP = ROXO
$ns color 30 purple
# Pacotes XMP START = AZUL
$ns color 31 blue
# Pacotes FTP = CINZA ESCURO
$ns color 10 darkgray
# Trafego Exponencial = AZUL SUAVE
$ns color 11 HotPink

### ROTEADORES ###
# Cria os roteadores
puts "Criando roteadores..."
array set cidade "0 CTA 1 SPO 2 RJO 3 BSA 4 RCE \
                  5 PAE 6 FNS 7 CAS 8 BHE 9 VTA \
                       10 SDR 11 FLA 12 CPE"
for { set i 0 } { $i < 13 } { incr i } {
          set j $cidade($i)
```

```
        set router($j) [$ns node]
        # Roteadores Nivel 2 - forma de quadrado
        $router($j) shape square
        $router($j) label $j
        $router($j) color red
        puts "Roteador em $j = [$router($j) id]"
}
# Roteadores Nivel 1 - formato hexagonal
$router(SPO) shape hexagon
$router(CTA) shape hexagon
$router(RJO) shape hexagon
$router(BSA) shape hexagon
$router(RCE) shape hexagon
# SPO sera o FP - cor VERDE
$router(SPO) color green

# Cria os links entre roteadores
puts "Cria enlaces entre roteadores..."
$ns duplex-link $router(PAE) $router(CTA) 5Mb 5ms DropTail
#$ns duplex-link $router(PAE) $router(SPO) 2Mb 5ms DropTail
$ns duplex-link $router(FNS) $router(CTA) 5Mb 5ms DropTail
#$ns duplex-link $router(FNS) $router(SPO) 2Mb 5ms DropTail
$ns duplex-link $router(CTA) $router(SPO) 10Mb 5ms DropTail
$ns duplex-link $router(CTA) $router(RJO) 10Mb 5ms DropTail
$ns duplex-link $router(SPO) $router(RJO) 34Mb 5ms DropTail
$ns duplex-link $router(SPO) $router(BSA) 10Mb 5ms DropTail
$ns duplex-link $router(SPO) $router(CAS) 5Mb 5ms DropTail
$ns duplex-link $router(SPO) $router(RCE) 10Mb 5ms DropTail
$ns duplex-link $router(RJO) $router(BHE) 5Mb 5ms DropTail
$ns duplex-link $router(RJO) $router(BSA) 10Mb 5ms DropTail
$ns duplex-link $router(RJO) $router(RCE) 10Mb 5ms DropTail
$ns duplex-link $router(RJO) $router(VTA) 5Mb 5ms DropTail
$ns duplex-link $router(BSA) $router(CPE) 5Mb 5ms DropTail
#$ns duplex-link $router(BSA) $router(RCE) 10Mb 5ms DropTail
$ns duplex-link $router(RCE) $router(SDR) 5Mb 5ms DropTail
$ns duplex-link $router(RCE) $router(FLA) 5Mb 5ms DropTail

#$ns duplex-link-op $router(RJO) $router(RCE) queuePos 0.5
#$ns duplex-link-op $router(RJO) $router(CTA) queuePos 0.5
#$ns duplex-link-op $router(SPO) $router(RCE) queuePos 0.5
#$ns duplex-link-op $router(SPO) $router(CAS) queuePos 0.5
#$ns duplex-link-op $router(SPO) $router(CTA) queuePos 0.5
```

```
$ns duplex-link-op $router(PAE) $router(CTA) color "blue"
#$ns duplex-link-op $router(PAE) $router(SPO) color "blue"
$ns duplex-link-op $router(FNS) $router(CTA) color "blue"
#$ns duplex-link-op $router(FNS) $router(SPO) color "blue"
$ns duplex-link-op $router(CTA) $router(SPO) color "blue"
$ns duplex-link-op $router(CTA) $router(RJO) color "blue"
$ns duplex-link-op $router(SPO) $router(RJO) color "orange"
$ns duplex-link-op $router(SPO) $router(BSA) color "blue"
$ns duplex-link-op $router(SPO) $router(CAS) color "blue"
$ns duplex-link-op $router(SPO) $router(RCE) color "blue"
$ns duplex-link-op $router(RJO) $router(BHE) color "blue"
$ns duplex-link-op $router(RJO) $router(BSA) color "blue"
$ns duplex-link-op $router(RJO) $router(RCE) color "blue"
$ns duplex-link-op $router(RJO) $router(VTA) color "blue"
$ns duplex-link-op $router(BSA) $router(CPE) color "blue"
#$ns duplex-link-op $router(BSA) $router(RCE) color "blue"
$ns duplex-link-op $router(RCE) $router(SDR) color "blue"
$ns duplex-link-op $router(RCE) $router(FLA) color "blue"


### USUARIOS ###
# Cria usuarios e links entre usuarios e roteadores
puts "Cria um usuario em cada roteador..."
for { set i 0 } { $i < 13 } { incr i } {
        set j $cidade($i)
        set user($j) [$ns node]
        puts "Usuario $j = [$user($j) id]"
        $user($j) label $j
        $ns duplex-link $user($j) $router($j) 10Mb 5ms DropTail
        $ns duplex-link-op $user($j) $router($j) color "darkgray"
}
$ns bandwidth $user(SPO) $router(SPO) 100Mb duplex
$ns bandwidth $user(RJO) $router(RJO) 100Mb duplex

# Angulos dos Links
$ns duplex-link-op $router(SPO) $router(RJO) orient right
$ns duplex-link-op $router(SPO) $router(BSA) orient 60deg
$ns duplex-link-op $router(RJO) $router(BSA) orient 120deg
$ns duplex-link-op $router(SPO) $router(CAS) orient left-up
$ns duplex-link-op $user(SPO) $router(SPO) orient right
$ns duplex-link-op $router(CTA) $router(SPO) orient right-up
```

```
$ns duplex-link-op $router(SPO) $router(RCE) orient 345deg
$ns duplex-link-op $router(CTA) $router(RJO) orient 15deg
$ns duplex-link-op $router(RJO) $router(RCE) orient right-down
$ns duplex-link-op $router(RJO) $router(VTA) orient right-up
$ns duplex-link-op $router(RJO) $router(BHE) orient up
$ns duplex-link-op $user(RJO) $router(RJO) orient left
$ns duplex-link-op $router(PAE) $router(CTA) orient right-down
$ns duplex-link-op $user(CTA) $router(CTA) orient right
$ns duplex-link-op $router(FNS) $router(CTA) orient right-up
$ns duplex-link-op $router(BSA) $router(CPE) orient up
$ns duplex-link-op $user(BSA) $router(BSA) orient right
$ns duplex-link-op $router(RCE) $router(SDR) orient right-down
$ns duplex-link-op $user(RCE) $router(RCE) orient left
$ns duplex-link-op $router(RCE) $router(FLA) orient right-up
$ns duplex-link-op $user(PAE) $router(PAE) orient right
$ns duplex-link-op $user(FNS) $router(FNS) orient right
$ns duplex-link-op $user(CAS) $router(CAS) orient right
$ns duplex-link-op $user(BHE) $router(BHE) orient left
$ns duplex-link-op $user(VTA) $router(VTA) orient left
$ns duplex-link-op $user(SDR) $router(SDR) orient left
$ns duplex-link-op $user(FLA) $router(FLA) orient left
$ns duplex-link-op $user(CPE) $router(CPE) orient right

# Comprimentos dos links
$ns duplex-link-op $router(SPO) $router(RJO) length 40
$ns duplex-link-op $router(SPO) $router(BSA) length 40
$ns duplex-link-op $router(SPO) $router(CAS) length 20
$ns duplex-link-op $user(SPO) $router(SPO) length 10
$ns duplex-link-op $router(CTA) $router(SPO) length 20
$ns duplex-link-op $router(SPO) $router(RCE) length 68
$ns duplex-link-op $router(CTA) $router(RJO) length 68
$ns duplex-link-op $router(RJO) $router(RCE) length 20
$ns duplex-link-op $router(RJO) $router(VTA) length 20
$ns duplex-link-op $router(RJO) $router(BHE) length 30
$ns duplex-link-op $user(RJO) $router(RJO) length 10
$ns duplex-link-op $router(RJO) $router(BSA) length 40
$ns duplex-link-op $router(PAE) $router(CTA) length 20
$ns duplex-link-op $user(CTA) $router(CTA) length 10
$ns duplex-link-op $router(FNS) $router(CTA) length 20
$ns duplex-link-op $router(BSA) $router(CPE) length 20
$ns duplex-link-op $user(BSA) $router(BSA) length 10
$ns duplex-link-op $router(RCE) $router(SDR) length 20
```

```
$ns duplex-link-op $user(RCE) $router(RCE) length 10
$ns duplex-link-op $router(RCE) $router(FLA) length 20
$ns duplex-link-op $user(PAE) $router(PAE) length 10
$ns duplex-link-op $user(FNS) $router(FNS) length 10
$ns duplex-link-op $user(CAS) $router(CAS) length 10
$ns duplex-link-op $user(BHE) $router(BHE) length 10
$ns duplex-link-op $user(VTA) $router(VTA) length 10
$ns duplex-link-op $user(SDR) $router(SDR) length 10
$ns duplex-link-op $user(FLA) $router(FLA) length 10
$ns duplex-link-op $user(CPE) $router(CPE) length 10

# Limites das filas
$ns queue-limit $router(BSA) $user(BSA) 360
$ns queue-limit $router(BSA) $router(CPE) 360
#$ns queue-limit $router(BSA) $router(RCE) 360
$ns queue-limit $router(BSA) $router(RJO) 360
$ns queue-limit $router(BSA) $router(SPO) 360
$ns queue-limit $router(CTA) $user(CTA) 360
$ns queue-limit $router(CTA) $router(FNS) 360
$ns queue-limit $router(CTA) $router(PAE) 360
$ns queue-limit $router(CTA) $router(RJO) 360
$ns queue-limit $router(CTA) $router(SPO) 360
$ns queue-limit $router(SPO) $user(SPO) 360
$ns queue-limit $router(SPO) $router(BSA) 360
$ns queue-limit $router(SPO) $router(CAS) 360
$ns queue-limit $router(SPO) $router(CTA) 360
#$ns queue-limit $router(SPO) $router(FNS) 360
#$ns queue-limit $router(SPO) $router(PAE) 360
$ns queue-limit $router(SPO) $router(RCE) 360
$ns queue-limit $router(SPO) $router(RJO) 360
$ns queue-limit $router(RJO) $user(RJO) 360
$ns queue-limit $router(RJO) $router(BHE) 360
$ns queue-limit $router(RJO) $router(BSA) 360
$ns queue-limit $router(RJO) $router(CTA) 360
$ns queue-limit $router(RJO) $router(RCE) 360
$ns queue-limit $router(RJO) $router(SPO) 360
$ns queue-limit $router(RJO) $router(VTA) 360
$ns queue-limit $router(RCE) $user(RCE) 360
#$ns queue-limit $router(RCE) $router(BSA) 360
$ns queue-limit $router(RCE) $router(FLA) 360
$ns queue-limit $router(RCE) $router(RJO) 360
$ns queue-limit $router(RCE) $router(SDR) 360
```

```
$ns queue-limit $router(RCE) $router(SPO) 360

# Leitura do endereco do grupo
set group [Node allocaddr]
# Define o node 0 como FP
XmpMcast set FP_($group) $router(SPO)
# Cria objetos do protocolo multicast XmpMcast em cada um dos nodes
set mproto XmpMcast
$ns mrtproto $mproto
# Configura os nodes dos roteadores como roteadores
for { set i 0 } { $i < 13 } { incr i } {
        [[$router($cidade($i)) getArbiter] getType XmpMcast] set isUser_ false
}

puts "Configuracao dos agentes..."
# Configuracao dos agentes de AV
#foreach j { array names user } {
for { set i 0 } { $i < 13 } { incr i } {
        set j $cidade($i)
        #create-av-rcvr [ append j 1 ]
        create-av-rcvr $j
}
# Cria trafegos AV
create-av-sndr RJO $group 1
#create-av-sndr PAE $group 2
#create-av-sndr FNS $group 2
#create-av-sndr CTA $group 2
#create-av-sndr SPO $group 2
#create-av-sndr VTA $group 2
#create-av-sndr BSA $group 2
#create-av-sndr CPE $group 2
#create-av-sndr BHE $group 2
#create-av-sndr CAS $group 2
#create-av-sndr RCE $group 2
#create-av-sndr FLA $group 2
#create-av-sndr SDR $group 2



# Cria trafegos exponenciais
create-exp-s SPO20 BHE1 3500k 11
create-exp-s SPO21 BSA1 7000k 11
```

```
create-exp-s SPO22 CAS1 3500k 11
create-exp-s SPO23 CPE1 3500k 11
create-exp-s SPO24 CTA1 7000k 11
create-exp-s SPO25 FLA1 3500k 11
create-exp-s SPO26 FNS1 3500k 11
create-exp-s SPO27 PAE1 3500k 11
create-exp-s SPO29 RCE1 7000k 11
create-exp-s SPO30 RJO1 23800k 11
create-exp-s SPO28 SDR1 3500k 11
create-exp-s SPO31 VTA1 3500k 11

create-exp-s RJO20 BHE2 3500k 10
create-exp-s RJO21 BSA2 7000k 10
create-exp-s RJO22 CAS2 3500k 10
create-exp-s RJO23 CPE2 3500k 10
create-exp-s RJO24 CTA2 7000k 10
create-exp-s RJO25 FLA2 3500k 10
create-exp-s RJO26 FNS2 3500k 10
create-exp-s RJO27 PAE2 3500k 10
create-exp-s RJO28 RCE2 7000k 10
create-exp-s RJO29 SDR2 3500k 10
create-exp-s RJO30 SPO2 23800k 10
create-exp-s RJO31 VTA2 3500k 10

create-exp-u BHE20 SPO3 3500k 11
create-exp-u BSA20 SPO3 7000k 11
create-exp-u CAS20 SPO3 3500k 11
create-exp-u CPE20 SPO3 3500k 11
create-exp-u CTA20 SPO3 7000k 11
create-exp-u FLA20 SPO3 3500k 11
create-exp-u FNS20 SPO3 3500k 11
create-exp-u PAE20 SPO3 3500k 11
create-exp-u RCE20 SPO3 3500k 11
create-exp-u RJO20 SPO3 23800k 11
create-exp-u SDR20 SPO3 3500k 11
create-exp-u VTA20 SPO3 3500k 11

create-exp-u BHE21 RJO3 3500k 11
create-exp-u BSA21 RJO3 7000k 11
create-exp-u CAS21 RJO3 3500k 11
create-exp-u CPE21 RJO3 3500k 11
create-exp-u CTA21 RJO3 7000k 11
```

```
create-exp-u FLA21 RJO3 3500k 11
create-exp-u FNS21 RJO3 3500k 11
create-exp-u PAE21 RJO3 3500k 11
create-exp-u RCE21 RJO3 7000k 11
create-exp-u SDR21 RJO3 3500k 11
create-exp-u SPO21 RJO3 23800k 11
create-exp-u VTA21 RJO3 3500k 11


####################
# Eventos no tempo #
####################
puts "Listando eventos no tempo..."

foreach ag $agents {
        $ns at 0.01 "$ag start"
}

#$ns at 0.1 "$cbr(RJO) start"

$ns at 2.0 "$user(RJO) join-group-tx $group"
$ns at 2.1 "$user(FLA) join-group-rx $rcvr(FLA) $group"
$ns at 2.1 "$user(PAE) join-group-rx $rcvr(PAE) $group"
$ns at 2.2 "$user(SDR) join-group-rx $rcvr(SDR) $group"
$ns at 2.3 "$user(CPE) join-group-rx $rcvr(CPE) $group"
$ns at 2.5 "$user(PAE) join-group-tx $group"
#$ns at 2.5 "$cbr(PAE) start"
$ns at 2.6 "$user(CAS) join-group-rx $rcvr(CAS) $group"
#$ns at 3.0 "$cbr(PAE) stop"
$ns at 3.0 "$user(RJO) join-group-tx $group"
$ns at 3.2 "$user(PAE) leave-group-rx $rcvr(PAE) $group"
$ns at 3.5 "$user(VTA) join-group-rx $rcvr(VTA) $group"
$ns at 3.5 "$user(BHE) join-group-rx $rcvr(BHE) $group"
$ns at 3.5 "$user(RCE) join-group-rx $rcvr(RCE) $group"
$ns at 4.0 "$user(RCE) join-group-tx $group"
#$ns at 4.0 "$cbr(RCE) start"
$ns at 4.4 "$user(CAS) leave-group-rx $rcvr(CAS) $group"
$ns at 4.7 "$user(SDR) leave-group-rx $rcvr(SDR) $group"
#$ns at 4.8 "$cbr(RCE) stop"
$ns at 4.8 "$user(RJO) join-group-tx $group"
$ns at 5.5 "$user(BHE) leave-group-rx $rcvr(BHE) $group"
$ns at 5.5 "$user(VTA) leave-group-rx $rcvr(VTA) $group"
```

```
$ns at 5.6 "$user(CPE) leave-group-rx $rcvr(CPE) $group"
$ns at 6.0 "$user(RJO) leave-group-tx $group"
$ns at 6.0 "$user(FLA) leave-group-rx $rcvr(FLA) $group"
$ns at 6.0 "$user(RCE) leave-group-rx $rcvr(RCE) $group"

#$ns at 7 "$cbr(RJO) stop"

foreach ag $agents {
        $ns at 7 "$ag stop"
}

# Fim da simulacao
#A rotina “finish” encerra a simulação e inicia o nam (ver Listagem 6, ns-xmp-aux.tcl)
$ns at 7.1 "finish"

#Roda o Simulador
puts "Inicia simulacao..."
$ns run
```

LISTAGEM 3
Xmp.h

```
#ifndef ns_xmp_h
#define ns_xmp_h
#include "agent.h"
#include "rtmodule.h"

enum xmp_dir { TRANSMITTER, RECEIVER };

// constroi o cabecalho (Header) dos pacotes XMP
struct hdr_xmp {
        int grupo_;
        xmp_dir dir_; // S, R
        char ptype_[15]; // start, stop
        int args_;
        int& grupo() { return grupo_; }
        xmp_dir& dir() { return dir_; }
        char* type() { return ptype_; }
        int& args() { return args_; }
        int maxtype() { return sizeof(ptype_); }
        // metodos de acesso ao Header
        static int offset_; //necessario para o PacketHeader Manager
        inline static int& offset() { return offset_; }
        inline static hdr_xmp* access (const Packet* p) {
                return (hdr_xmp*) p->access (offset_);
        }
};

#define OFF 0
#define TX -1

struct if_status {
        int iface_;
        int status_;
        if_status* next_;
        int& iface() { return iface_; }
        int& status() { return status_; }
        void next(if_status* n) { next_ = n; }
        if_status* next() { return next_; }
```

```
};

#define MAXGROUP 15

class XmpControlAgent : public Agent {
        public:
                XmpControlAgent();
                ~XmpControlAgent();

        protected:
                void recv(Packet*, Handler*);
                int command(int, const char*const*);
                if_status* insert_if(int, int, int);
                if_status* lookup_if (int, int);

                if_status* first_if_[MAXGROUP];
                int isUser_;
                int isProf_;
};

class XmpClassifier : public DestHashClassifier {
        public:
                XmpClassifier ();
                int command (int argc, const char* const* argv);
                void recv (Packet*, Handler*);
        protected:
                Classifier* dmux_;
                int FP_;
};

class XmpRoutingModule : public BaseRoutingModule {
        public:
                XmpRoutingModule() : BaseRoutingModule() { }
                const char* module_name() const { return "Xmp"; }
                //virtual int command(int argc, const char*const* argv);
                //virtual void add_route(char *dst, NsObject *target);
};

/*
class XmpUser : public Agent {
        public:
                XmpUser ();
```

```
                int command (int argc, const char* const* argv);
                void recv (Packet*, Handler*);
};

class XmpRouter : public Agent {
        public:
                XmpRouter ();
                int command (int argc, const char* const* argv);
                void recv (Packet*, Handler*);
        protected:
};
*/

#endif
```

LISTAGEM 4
Xmp.cc

```
// rotina em C++ para criar o Header do Xmp

#include "Xmp.h"
#include "packet.h"

int hdr_xmp::offset_;

static class XmpHeaderClass : public PacketHeaderClass {
public:
        XmpHeaderClass() : PacketHeaderClass("PacketHeader/Xmp",
                                             sizeof(hdr_xmp)) {
                bind_offset(&hdr_xmp::offset_);
        }
} class_xmphdr;

static class XmpControlClass : public TclClass {
public:
        XmpControlClass() : TclClass("Agent/Mcast/Control/Xmp") { }
        TclObject* create(int, const char*const*) {
                return (new XmpControlAgent());
        }
} class_xmp_ctrl;

static class XmpClassifierClass : public TclClass {
        public:
                XmpClassifierClass() : TclClass("Classifier/Hash/Dest/Xmp") { }
                TclObject* create(int, const char* const*) {
                        return (new XmpClassifier());
                }
} class_xmp_classifier;

static class XmpRoutingModuleClass : public TclClass {
public:
        XmpRoutingModuleClass() : TclClass("RtModule/Xmp") {}
        TclObject* create(int, const char*const*) {
                return (new XmpRoutingModule);
        }
} class_xmp_routing_module;
```

```
/*************************************************************/
// XmpControlAgent
/*************************************************************/

XmpControlAgent::XmpControlAgent() : Agent(PT_NTYPE), isUser_(0), isProf_(0)
{
        for (int i = 0; i < MAXGROUP; ++i)
        {
                first_if_[i] = 0;
        }
        bind("packetSize_", &size_);
        bind_bool("isUser_", &isUser_);
        bind_bool("isProf_", &isProf_);
}

XmpControlAgent::~XmpControlAgent()
{
        for (int i = 0; i < MAXGROUP; ++i)
        {
                while (0 != first_if_[i])
                {
                        if_status* next_if = first_if_[i]->next();
                        delete first_if_;
                        first_if_[i] = next_if;
                }
        }
}

void
XmpControlAgent::recv(Packet* pkt, Handler*)
{
        hdr_xmp* xmph = hdr_xmp::access(pkt);
        hdr_cmn* ch = hdr_cmn::access(pkt);
        Tcl::instance().evalf("%s recv %s %d %d", name(),
                                xmph->type(), ch->iface(), xmph->args());
        Packet::free(pkt);
}

int
XmpControlAgent::command(int argc, const char*const* argv)
{
```

```
if (argc == 2)
{
	if (strcmp(argv[1], "isUser") == 0)
	{
		isUser_ = 1;
		return (TCL_OK);
	}
	if (strcmp(argv[1], "isRouter") == 0)
	{
		isUser_ = 0;
		return (TCL_OK);
	}
	if (strcmp(argv[1], "isProf") == 0)
	{
		isUser_ = 1;
		isProf_ = 1;
		return (TCL_OK);
	}
	if (strcmp(argv[1], "isAluno") == 0)
	{
		isUser_ = 1;
		isProf_ = 0;
		return (TCL_OK);
	}
}
else if (argc == 4)
{
	/*
	 * $proc send $type $dir $group
	 * $proc send $type $msg_index
	 */
	if (strcmp(argv[1], "send") == 0)
	{
		// Analisa o tipo de pacote a ser transmitido
		if (strcmp(argv[2], "start") == 0)
		{
			type_ = PT_START;
		}
		else if (strcmp(argv[2], "stop") == 0)
		{
			type_ = PT_STOP;
		}
```

```
                        else
                        {
                                Tcl& tcl = Tcl::instance();
                                tcl.result("invalid control message");
                                return (TCL_ERROR);
                        }
                        // Analisa o estado que o usuario quer ingressar
/*                      xmp_dir dir;
                        if (strcmp(argv[3], "tx") == 0)
                        {
                                dir = TRANSMITTER;
                        }
                        else if (strcmp(argv[3], "rx") == 0)
                        {
                                dir = RECEIVER;
                        }
                        else
                        {
                                Tcl& tcl = Tcl::instance();
                                tcl.result("invalid control message");
                                return (TCL_ERROR);
                        }*/
                        // Create a new packet
                        Packet* pkt = allocpkt();
                        // Access the XMP header for the new packet:
                        hdr_xmp* xmph = hdr_xmp::access(pkt);
                        strcpy(xmph->type(), argv[2]);
                        xmph->args() = atoi(argv[3]);
                        send(pkt, 0);
                        // return TCL_OK, so the calling function knows that the
                        // command has been processed
                        return (TCL_OK);
                }
        }
        return (Agent::command(argc, argv));
}

if_status*
XmpControlAgent::insert_if (int group, int iface, int status)
{
        if_status* new_if = new if_status;
        new_if->iface() = iface;
```

```
        new_if->status() = status;
        new_if->next(first_if_[group]);
        first_if_[group] = new_if;

        return new_if;
}

if_status*
XmpControlAgent::lookup_if (int group, int iface)
{
        if_status* cur_if;
        for (cur_if = first_if_[group]; cur_if->next() != 0; cur_if = cur_if->next())
        {
                if (cur_if->iface() == iface)
                {
                        break;
                }
        }
        if (iface != cur_if->iface()
        && 0 == cur_if->next())
        {
                return 0;
        }
        return cur_if;
}


/*************************************************************/
// XmpClassifier
/*************************************************************/
XmpClassifier::XmpClassifier() : dmux_(0), FP_(0)
{
        bind_bool("FP_", &FP_);
}

int
XmpClassifier::command (int argc, const char* const* argv)
{
        Tcl& tcl = Tcl::instance();
        if (argc == 3)
        {
                if (strcmp(argv[1], "demux") == 0)
```

```
                {
                        dmux_ = (Classifier *) TclObject::lookup(argv[2]);
                        if (0 == dmux_)
                        {
                                tcl.result("ERRO: Handle invalido");
                                return (TCL_ERROR);
                        }
                        return (TCL_OK);
                }
        }
        return (DestHashClassifier::command(argc,argv));
}

void
XmpClassifier::recv (Packet* p, Handler*)
{
        hdr_cmn* hcmn = hdr_cmn::access(p);
        if (hcmn->ptype_ == PT_XMP)
        {
                if (0 == dmux_)
                {
                        printf("ERRO: dmux_ deve ser passado ao classifier\nUse o procedure
\"$classifier demux $portclassifier\"\n");
                        exit(1);
                }
                dmux_->recv(p->copy(),NULL);
        }
        // Se roteador nao for FP, repassa pacote para proximo roteador
        if (!FP_)
        {
                DestHashClassifier::recv(p,NULL);
        }
        else
        {
                Packet::free(p);
        }
}

/*
static class XmpUserClass : public TclClass {
        public:
                XmpUserClass() : TclClass("Agent/Xmp/User") { }
```

```
		TclObject* create(int, const char* const*) {
			return (new XmpUser());
		}
} class_xmp_user;

static class XmpRouterClass : public TclClass {
	public:
		XmpRouterClass() : TclClass("Agent/Xmp/Router") { }
		TclObject* create(int, const char* const*) {
			return (new XmpRouter());
		}
} class_xmp_router;
*/

/*****************************************************************/
// XmpUser
/*****************************************************************/
/*XmpUser::XmpUser() : Agent(PT_NTYPE)
{
	bind("packetSize_",&size_);
}

int XmpUser::command (int argc, const char* const* argv)
{
	if (argc == 4)
	{
		// $xmpuser start $grupo $quemsou
		if (strcmp(argv[1], "start") == 0)
		{
			xmp_dir dir;
			if (strcmp(argv[3], "tx") == 0)
			{
				dir = SENDER;
			}
			else if (strcmp(argv[3], "rx") == 0)
			{
				dir = RECEIVER;
			}
			else
			{
				return(TCL_ERROR);
			}
```

```
		type_ = PT_START;
		// Create a new packet
		Packet* pkt = allocpkt();
		// Access the XMP header for the new packet:
		hdr_xmp* xmph = hdr_xmp::access(pkt);
		xmph->grupo_ = atoi(argv[2]);
		xmph->dir_ = dir;
		// Send the packet
		printf("Pacote tipo START transmitido pelo usuario %s\n", name());
		send(pkt, 0);
		// return TCL_OK, so the calling function knows that the
		// command has been processed
		return (TCL_OK);
	}
	// $xmpuser start $grupo $quemsou
	else if (strcmp(argv[1], "stop") == 0)
	{
		xmp_dir dir;
		if (strcmp(argv[3], "tx") == 0)
		{
			dir = SENDER;
		}
		else if (strcmp(argv[3], "rx") == 0)
		{
			dir = RECEIVER;
		}
		else
		{
			return(TCL_ERROR);
		}
		type_ = PT_STOP;
		// Create a new packet
		Packet* pkt = allocpkt();
		// Access the XMP header for the new packet:
		hdr_xmp* xmph = hdr_xmp::access(pkt);
		xmph->grupo_ = atoi(argv[2]);
		xmph->dir_ = dir;
		// Send the packet
		printf("Pacote tipo STOP transmitido pelo usuario %s\n", name());
		send(pkt, 0);
		// return TCL_OK, so the calling function knows that the
		// command has been processed
```

```
                        return (TCL_OK);
                }
        }
        return (Agent::command(argc,argv));
}

void XmpUser::recv (Packet* p, Handler*)
{
        // O agente do usuario NUNCA deveria receber pacote de controle
        printf("ERRO: Agente XMP do usuario NUNCA deveria receber pacote de controle\n");
        exit(1);
        Packet::free(p);
}
*/
/*************************************************************/
// XmpRouter
/*************************************************************/
/*XmpRouter::XmpRouter() : Agent(PT_NTYPE)
{
}

int XmpRouter::command (int argc, const char* const* argv)
{
        return (Agent::command(argc,argv));
}

void XmpRouter::recv (Packet* p, Handler*)
{
        hdr_cmn* ch = hdr_cmn::access(p);
        if (ch->ptype() == PT_START)
        {
                printf("Pacote tipo START recebido pelo roteador %s\n", name());
        }
        else if (ch->ptype() == PT_STOP)
        {
                printf("Pacote tipo STOP recebido pelo roteador %s\n", name());
        }
        else
        {
                printf("ERRO! Tipo de pacote XMP inexistente\n");
        }
        Packet::free(p);
```

```
}
*/
```

LISTAGEM 5

ns-xmp.tcl

```
### INICIALIZACAO DE VARIAVEIS EM TCL E C++ ###
### (ver funcao bind em C++)
Agent/Mcast/Control/Xmp set packetSize_ 10;#3 bytes XMP (+RTP+UDP+IP)
Agent/Mcast/Control/Xmp set isUser_ true
Agent/Mcast/Control/Xmp set isProf_ true
#####################################################################
# P R O C S   &   I N S T P R O C S
#####################################################################

#####################################################################
### Modificacoes na Classe Node
#####################################################################

#Node instproc direction { dir group } {
#       if { $dir != "tx" && $dir != "rx" } {
#               puts "AVISO: Argumento invalido. Deve ser 'tx' ou 'rx'"
#               return ""
#       }
#       set arbiter [ $self getArbiter ]
#       if { $arbiter != "" } {
#               set proto [ $arbiter getType XmpMcast ]
#               if { $proto != "" } {
#                       set iface [$proto set iface_]
#                       $proto set if_status_($iface:$group) $dir
#               } else {
#                       puts "AVISO: Nao existe instancia do protocolo Xmp"
#               }
#       } else {
#               puts "AVISO: Nao existe instancia do arbitro (mrtObject)"
#       }
#       return ""
#}

Node instproc disable-agents { group { src "" } } {
```

```
        $self instvar Agents_
          set group [expr $group] ;# use expr to get rid of possible leading 0x

          if { $src == "" } {
                  set reps [$self getReps "*" $group]
          } else {
                  set reps [$self getReps $src $group]
          }

          if [info exists Agents_($group)] {
                  foreach a $Agents_($group) {
                        foreach rep $reps {
                                $rep disable $a
                           }
                           set k [lsearch -exact $Agents_($group) $a]
                           set Agents_($group) [lreplace $Agents_($group) $k $k]
                  }
          } else {
                  warn "cannot erase an agent from a group without joining it"
          }
}

Node instproc join-group-tx { group { src "" } } {
        $self instvar replicator_ mrtObject_
        set group [expr $group] ;# use expr to convert to decimal

          set xmp [$mrtObject_ getType XmpMcast]
          if { $xmp != "" } {
                  if [$xmp is-valid? $group "start" "tx"] {
                        $mrtObject_ join-group-tx $group $src
                  }
          }
}

Node instproc join-group-rx { agent group { src "" } } {
        $self instvar replicator_ mrtObject_
        set group [expr $group] ;# use expr to convert to decimal

          set xmp [$mrtObject_ getType XmpMcast]
          if { $xmp != "" } {
                  if [$xmp is-valid? $group "start" "rx"] {
                          $self join-group $agent $group $src
```

```
                }
        }
}

Node instproc leave-group-tx { group { src "" } } {
        $self instvar replicator_ Agents_ mrtObject_
        set group [expr $group] ;# use expr to get rid of possible leading 0x

          set xmp [$mrtObject_ getType XmpMcast]
          if { $xmp != "" } {
                  if [$xmp is-valid? $group "stop" "tx"] {
                          $mrtObject_ leave-group $group $src
                  }
          }
}

Node instproc leave-group-rx { agent group { src "" } } {
        $self instvar replicator_ mrtObject_
        set group [expr $group] ;# use expr to convert to decimal

          set xmp [$mrtObject_ getType XmpMcast]
          if { $xmp != "" } {
                  if [$xmp is-valid? $group "stop" "rx"] {
                          $self leave-group $agent $group $src
                  }
          }
}

#####################################################################
### Modificacoes na Classe mrtObject
#####################################################################
# similar to membership indication by igmp..
mrtObject instproc join-group-tx { gfp src } {
        eval $self all-mprotos join-group $gfp "tx"
}

#####################################################################
### Modificacoes na Classe Classifier/Replicator/Demuxer
#####################################################################
Classifier/Replicator/Demuxer instproc reset {} {
        $self instvar nactive_ active_
        foreach { target slot } [array get active_] {
```

```
                if { $slot >= 0 } {
                        $self clear $slot
                }
        }
        set nactive_ 0
        if [info exists active_] {
                unset active_
        }
}

#####################################################################
###       Modificacoes na Classe Classifier   ############
#####################################################################
Classifier instproc clear slot {
        if { $slot < 0 } {
                warn "ERRO: Tentativa de apagar slot invalido em classificador $self"
                exit 1
        }
        $self cmd clear $slot
        $self unset slots_($slot)
}

#####################################################################
##     Classe XmpMcast - executa o protocolo do Xmp   #######
#####################################################################
Class XmpMcast -superclass McastProtocol

XmpMcast instproc init { sim node } {
        XmpMcast instvar FP_ ; # FPs by group

        if ![info exists FP_] {
                error "XmpMcast: 'XmpMcast instvar FP_' must be set"
                exit 1
        }

        $self next $sim $node

        $self instvar id_ mctrl_ if_status_ isUser_ isProf_ if_rxcounter_ iface_

        set id_ [$node id]
        set isUser_ true
        set isProf_ true
```

```
        set mctrl_ [new Agent/Mcast/Control/Xmp $self]
        $node attach $mctrl_

        foreach gfpx [array names FP_] {
                set gfp [expr $gfpx]
                if [string compare $gfp $gfpx] {
                        set FP_($gfp) $FP_($gfpx)
                        unset FP_($gfpx)
                }
        }
}

#XmpMcast instproc start {} {
#       $self instvar ns_

        #interconnect all agents, decapsulators and encapsulators
        #this had better be done in init()
#       foreach gfp [array names encaps_] {
#               foreach e $encaps_($gfp) {
#                       $ns_ simplex-connect $e $decaps_($gfp)
#               }
#       }
#       $self next
#}

XmpMcast instproc join-group { group {dir "rx"} } {
        set src "x"
        if { $src != "x" } {
                $self dbg "ERRO: Nao passar src!!!"
                return
        }

        $self instvar node_ ns_ if_status_ iface_
        XmpMcast instvar FP_
        if ![info exists iface_] {
                set iface_ [$node_ from-node-iface $FP_($group)]
        }

        if ![info exists if_status_($iface_:$group)] {
                set if_status_($iface_:$group) "off"
        }
```

```
        set r [$node_ getReps "x" $group]
        if {$r == ""} {
                if { $dir == "rx" } {
                        set iif $iface_
                        set oif ""
                } elseif { $dir == "tx" } {
                        set iif -1
                        set oif [$node_ iif2oif $iface_]
                } else {
                        puts "AVISO: Valor errado do argumento dir"
                        exit 1
                }
                set if_status_($iface_:$group) $dir
                $self dbg "********* start $dir: adding <x, $group, $iif>"
                $node_ add-mfc "x" $group $iif $oif
                #set r [$node_ getReps "x" $group]
        } elseif { $if_status_($iface_:$group) == "rx" && $dir == "tx" } {
                set if_status_($iface_:$group) $dir
                set oldiif $iface_
                $node_ change-iface "x" $group $oldiif -1
                $node_ disable-agents $group
                $r insert [$node_ iif2oif $oldiif]
                $self dbg "********* start $dir: adding <x, $group, -1>"
        } else {
                warn "Nao transmite pacote"
                return
        }
        $self send-ctrl "start" $FP_($group) $group $dir
        $self next $group ; # annotate
}

XmpMcast instproc leave-group { group {src "x"} } {
        if { $src != "x" } return

        $self instvar node_ if_status_ iface_
        XmpMcast instvar FP_
        if ![info exists iface_] {
                set iface_ [$node_ from-node-iface $FP_($group)]
        }

        if { $if_status_($iface_:$group) == "off" } {
                warn "cannot leave a group without joining it"
```

```
                return
        }

        set dir $if_status_($iface_:$group)
        #set if_status_($iface_:$group) "off"
        $self next $group
        # check if the rep is active, then send a prune
        $node_ clearReps "x" $group
        unset if_status_
        $self dbg "********* stop $dir: removing <x, $group, ...>"
        $self send-ctrl "stop" $FP_($group) $group $dir
}

#XmpMcast instproc handle-wrong-iif { srcID group iface } {
#       $self instvar node_
#       $self dbg "!!!!!XmpMcast: wrong iif $iface, src $srcID, gfp $group"
#       return 0
#}

# There should be only one mfc cache miss: the first time it receives a
# packet to the group (FP). Just install a (x,G) entry.
#XmpMcast instproc handle-cache-miss { srcID group iface } {
#       $self instvar node_
#       $self dbg "!!!!!XmpMcast: wrong iif $iface, src $srcID, gfp $group"
#       return 0
        #XmpMcast instvar FP_

        # FP gets packets from the local decapsulator, so iface must be "?" (-1)
        #if { $iface != -1 } {$self dbg ".....invalid cache miss" }
        #$self dbg "cache miss, src: $srcID, group: $group, iface: $iface"
        #$self dbg "********* miss: adding <x, $group, $iface, >"
        #$node_ add-mfc "x" $group $iface ""
        #return 1
#}

XmpMcast instproc drop { replicator src dst iface } {
        $self instvar node_ ns_
        XmpMcast instvar FP_

        # No downstream listeners
        # Send a prune back toward the FP
        #$self dbg "drops src: $src, dst: $dst, iface: $iface, replicator: [$replicator set srcID_]"
```

```
        #if {$iface != -1} {
                # so, this packet came from outside of the node
                #$self send-ctrl "stop" $FP_($dst) $dst
        #}
}

XmpMcast instproc recv-stop { from src group dir iface } {
        $self instvar node_ ns_ id_ if_status_ if_rxcounter_
        XmpMcast instvar FP_

        # Testa se o estado (stop, dir [tx/rx], if_status_ [tx/rx], if_rxcounter_ [>0/==0])
        #   eh um estado possivel
        if ![$self is-valid? $group "stop" $dir $iface] {
                #$self dbg "ignoring 'stop' from: $from, dir: $dir. Current interface status:
$if_status_($iface:$group)"
                return
        }
        #$self dbg "received a 'stop $dir' from $from ==> Current interface status:
$if_status_($iface:$group)"

        set r [$node_ getReps "x" $group]
        if { $r == "" || $if_status_($iface:$group) == "off" } {
                # it's a cache miss!
                #$self dbg "recvd a stop, do nothing....."
                warn "node $id_, got a stop from $from, trying to stop a non-existing interface?"
                return
        } elseif { $if_status_($iface:$group) == "tx" } {
                if { $dir == "tx" } {
                        #set if_status_($iface:$group) "off"
                        $node_ clearReps "x" $group
                        unset if_status_
                        unset if_rxcounter_
                } elseif { $dir == "rx" && $if_rxcounter_($iface:$group) > 0 } {
                        incr if_rxcounter_($iface:$group) -1
                } else {
                        warn "node $id_, got a stop from $from, trying to stop a non-existing
interface?"
                        return
                }
        } elseif { $if_status_($iface:$group) == "rx" } {
                if { $dir == "rx" && $if_rxcounter_($iface:$group) > 0 } {
```

```
                        incr if_rxcounter_($iface:$group) -1
                        if { $if_rxcounter_($iface:$group) == 0 } {
                                set if_status_($iface:$group) "off"
                                set oif [$node_ iif2oif $iface]
                                if ![$r exists $oif] {
                                        warn "node $id_, got a stop from $from, trying to stop a
non-existing interface?"
                                } else {
                                        if [$r is-active-target $oif] {
                                                $r disable $oif
                                        }
                                }
                        }
                } else {
                        warn "node $id_, got a stop from $from, trying to stop a non-existing
interface?"
                        return
                }
        } else {
                warn "ERROR: This line should never be run"
                exit 1
        }
        $self send-ctrl "stop" $FP_($group) $group $dir
}

XmpMcast instproc recv-start { from to group dir iface } {
        $self instvar node_ ns_ id_ if_status_ if_rxcounter_ isUser_
        XmpMcast instvar FP_

        if { ![$self is-valid? $group "start" $dir $iface] && \
            !($isUser_ && $if_status_($iface:$group) == "tx" && $dir == "tx") }  {
                #$self dbg "ignoring 'start' from: $from, dir: $dir. Current interface status:
$if_status_($iface:$group)"
                return
        }
        #$self dbg "received a 'start $dir' from $from ==> Current interface status:
$if_status_($iface:$group)"

        set r [$node_ getReps "x" $group]
        if { $r == "" } {
                set if_status_($iface:$group) $dir
                if { $dir == "tx" } {
```

```
                set iif $iface
                if { $node_ != $FP_($group) } {
                        set iffpf [$node_ from-node-iface $FP_($group)]
                        set if_status_($iffpf:$group) "rx"
                        set if_rxcounter_($iffpf:$group) 1
                        set oif [$node_ iif2oif $iffpf]
                } else {
                        set oif ""
                }
        } elseif { $dir == "rx" } {
                if { $node_ != $FP_($group) } {
                        set iif [$node_ from-node-iface $FP_($group)]
                        set if_status_($iif:$group) "tx"
                } else {
                        set iif ?
                }
                set oif [$node_ iif2oif $iface]
                incr if_rxcounter_($iface:$group)
        } else {
                warn "ERROR: Invalid direction"
                exit 1
        }
        #$self dbg "********* start $dir: adding <x, $group, $iif>"
        $node_ add-mfc "x" $group $iif $oif
} elseif { $if_status_($iface:$group) == "off" } {
        set if_status_($iface:$group) $dir
        if { $dir == "tx" } {
                set oldiif [$node_ lookup-iface "x" $group]
                $node_ change-iface "x" $group $oldiif $iface
                if { $oldiif >= 0 } {
                        $r insert [$node_ iif2oif $oldiif]
                        set if_status_($oldiif:$group) "rx"
                        if [info exists if_rxcounter_($oldiif:$group)] {
                                incr if_rxcounter_($oldiif:$group)
                        } else {
                                set if_rxcounter_($oldiif:$group) 1
                        }
                        # Mudanca de TX ao longo de uma sessao multicast:
                        # Envia pacote START TX ate o professor,
                        # ignorando a existencia do FP.
                        # Para isso, o pacote START TX deve ser encaminhado
                        # para a antiga interface de entrada
```

```
                        set oldlink [$node_ iif2link $oldiif]
                        set oldnode [$oldlink src]
                        $self send-ctrl "start" $oldnode $group $dir 1
                        return
                }
        } elseif { $dir == "rx" } {
                $r insert [$node_ iif2oif $iface]
                incr if_rxcounter_($iface:$group)
        } else {
                warn "ERROR: Invalid direction"
                exit 1
        }
} elseif { $if_status_($iface:$group) == "tx" } {
        if { $dir == "rx" } {
                incr if_rxcounter_($iface:$group)
        } elseif { $isUser_ && $dir == "tx" } {
        # Mudanca de TX ao longo de uma sessao multicast:
        # Unico caso em que usuario recebe pacote.
        # Usuario estava no estado "PROF" e recebe "START TX"
        # Node deve alterar sua configuracao interna (multiclassifier
        # e replicator) para configuracao "ALUNO"
                set if_status_($iface:$group) "rx"
                $node_ change-iface "x" $group -1 $iface
                $r disable [$node_ iif2oif $iface]
                if [$node_ check-local $group] {
                # Se ja existir agente de recepcao de dados,
                # basta inseri-lo na saida do replicador
                        foreach a [$node_ set Agents_($group)] {
                                $r insert $a
                        }
                } else {
                # Se nao existir nenhum agente de recepcao de dados,
                # eh necessario criar um novo agente, atualizar
                # a lista de agentes do node e, finalmente,
                # inserir o agente na saida do replicador
                        set a [new Agent/LossMonitor]
                        $ns_ attach-agent $node_ $a
                        $node_ set Agents_($group) $a
                        #$node_ set Agents_($group) [concat [$node_ set
Agents_($group)] $a]
                        $r insert $a
                }
```

```
                        return
                }
        } elseif { $if_status_($iface:$group) == "rx" } {
                set if_status_($iface:$group) $dir
                if { $dir == "tx" } {
                        incr if_rxcounter_($iface:$group) -1
                        $r disable [$node_ iif2oif $iface]
                        set oldiif [$node_ lookup-iface "x" $group]
                        $node_ change-iface "x" $group $oldiif $iface
                        if { $oldiif >= 0 } {
                                $r insert [$node_ iif2oif $oldiif]
                                set if_status_($oldiif:$group) "rx"
                                if [info exists if_rxcounter_($oldiif:$group)] {
                                        incr if_rxcounter_($oldiif:$group)
                                } else {
                                        set if_rxcounter_($oldiif:$group) 1
                                }
                                # Mudanca de TX ao longo de uma sessao multicast:
                                # Envia pacote START TX ate o professor,
                                # ignorando a existencia do FP.
                                # Para isso, o pacote START TX deve ser encaminhado
                                # para a antiga interface de entrada
                                set oldlink [$node_ iif2link $oldiif]
                                set oldnode [$oldlink src]
                                $self send-ctrl "start" $oldnode $group $dir 1
                                return
                        }
                } elseif { $dir == "rx" } {
                        incr if_rxcounter_($iface:$group)
                } else {
                        warn "ERROR: Invalid direction"
                        exit 1
                }
        } else {
                warn "ERROR: This line should never be run"
                exit 1
        }
        $self send-ctrl "start" $FP_($group) $group $dir
}

#
# send a graft/prune to dst/group up the FPF tree towards dst
```

```
#
XmpMcast instproc send-ctrl { which dst group dir { toProf 0 } } {
	$self instvar mctrl_ ns_ node_

	if { $toProf || $node_ != $dst} {
		set nbr [$node_ fpf-nbr $dst]
		$ns_ simplex-connect $mctrl_ \
				[[[$nbr getArbiter] getType [$self info class]] set mctrl_]
		if { $which == "stop" } {
			$mctrl_ set class_ 30
		} else {
			$mctrl_ set class_ 31
		}
		$mctrl_ send $which [$node_ id] -1 $group $dir
	}
}

### Helpers
XmpMcast instproc is-valid? { group which dir { iif "" } } {
	$self instvar isUser_ if_status_ node_ if_rxcounter_ iface_
	XmpMcast instvar FP_

	if { $isUser_ } {
		if ![info exists iface_] {
			set iface_ [$node_ from-node-iface $FP_($group)]
		}
		if ![info exists if_status_($iface_:$group)] {
			set if_status_($iface_:$group) "off"
		}
		switch $if_status_($iface_:$group) {
			off {
				if { $which == "start" } {
					return 1
				} else {
					return 0
				}
			}
			tx {
				if { $which == "stop" && $dir == "tx" } {
					return 1
				} else {
					return 0
```

```
                        }
                    }
                    rx {
                        if { $which == "start" && $dir == "tx" || \
                            $which == "stop" && $dir == "rx" } {
                            return 1
                        } else {
                            return 0
                        }
                    }
                    default
                        { return 0 }
                }
            } else {
                if { $iif == "" } {
                    warn "'iif' argument is mandatory for Router"
                    exit 1
                }
                if ![info exists if_status_($iif:$group)] {
                    set if_status_($iif:$group) "off"
                    set if_rxcounter_($iif:$group) 0
                }
                switch $if_status_($iif:$group) {
                    off {
                        if { $which == "start" && $if_rxcounter_($iif:$group) == 0 } {
                            return 1
                        } else {
                            return 0
                        }
                    }
                    tx {
                        if { $which == "start" && $dir == "rx" || \
                            $which == "stop" && $dir == "tx" || \
                            $which == "stop" && $dir == "rx" &&
$if_rxcounter_($iif:$group) > 0 } {
                            return 1
                        } else {
                            return 0
                        }
                    }
                    rx {
                        if { $if_rxcounter_($iif:$group) == 0 || \
```

```
                                $which == "stop" && $dir == "tx" } {
                                        return 0
                                } else {
                                        return 1
                                }
                        }
                        default
                                { return 0 }
                }
        }
}

XmpMcast instproc dbg arg {
        $self instvar ns_ node_ id_
        puts [format "At %.4f : node $id_ $arg" [$ns_ now]]
}
##################################################################
## Agent/Mcast/Control/Xmp
##################################################################
Agent/Mcast/Control/Xmp instproc send {type from src group dir args} {
        Agent/Mcast/Control instvar mcounter messages
        set messages($mcounter) [concat [list $from $src $group $dir] $args]
        $self cmd send $type $mcounter
        incr mcounter
}

Agent/Mcast/Control/Xmp instproc recv2 {type iface from src group dir args} {
      $self instvar proto_
      eval $proto_ recv-$type $from $src $group $dir $iface $args
}
##################################################################
############# CODIGO QUE NAO ESTA SENDO UTILIZADO ########
##################################################################
#
# Base routing module
#

# Use standard naming although this is a pure OTcl class
#
# In fact, all functionalities of Base are already implemented in RtModule.
# We add this class only to provide a uniform interface to the RtModule
# creation process, where a single line [new RtModule/$name] is used.
```

```
#
# Defined in ~ns/rtmodule.{h,cc}

#RtModule/Xmp instproc register { node } {
#       $self next $node
#
#       $self instvar classifier_
#       set classifier_ [new Classifier/Hash/Dest/Xmp 32]
#       $classifier_ set mask_ [AddrParams NodeMask 1]
#       $classifier_ set shift_ [AddrParams NodeShift 1]
#       # XXX Base should ALWAYS be the first module to be installed.

#       $node install-entry $self $classifier_
#}
#
#Classifier/Hash/Dest/Xmp set FP_ false
```

LISTAGEM 6
ns-xmp-aux.tcl

```
proc create-ftp { serv cli { cl 0 } } {
        global ns tcp tcpsink ftp user server
        regsub "\[0-9\]*$" $serv {} snode
        regsub "\[0-9\]*$" $cli {} dnode
        # Cria receptor de dados TCP
        set tcpsink($serv:$cli) [new Agent/TCPSink]
        # Insere agente TCPSink no cliente
        $ns attach-agent $user($dnode) $tcpsink($serv:$cli)
        # Cria agente TCP para transmitir dados
        set tcp($serv:$cli) [new Agent/TCP]
        # Insere agente TCP no servidor
        $ns attach-agent $user($snode) $tcp($serv:$cli)
        # Conecta o servidor ao cliente
        $ns connect $tcp($serv:$cli) $tcpsink($serv:$cli)
        # Cria aplicacao FTP
        set ftp($serv:$cli) [new Application/FTP]
        lappend agents $ftp($serv:$cli)
        # Insere o gerador FTP no agente TCP
        $ftp($serv:$cli) attach-agent $tcp($serv:$cli)
        # Identifica fluxo
        if { $cl != 0 } {
                $tcp($serv:$cli) set class_ $cl
        }
}

proc create-exp-s { serv cli rate { cl 0 } } {
        global ns udp sink expo user server agents
        regsub "\[0-9\]*$" $serv {} snode
        regsub "\[0-9\]*$" $cli {} dnode
        # Cria agente receptor de dados
        set sink($serv:$cli) [new Agent/LossMonitor]
        # Insere agente no node cli
        $ns attach-agent $user($dnode) $sink($serv:$cli)
        # Cria agente UDP para transmitir dados
        set src($serv:$cli) [new Agent/UDP]
        # Insere agente UDP no node serv
        $ns attach-agent $user($snode) $src($serv:$cli)
        # Conecta agentes
```

```
        $ns connect $src($serv:$cli) $sink($serv:$cli)
        # Insere o gerador Exponencial
        set expo($serv:$cli) [new Application/Traffic/Exponential]
        lappend agents $expo($serv:$cli)
        # Insere o gerador exponencial no agente UDP
        $expo($serv:$cli) attach-agent $src($serv:$cli)
        # Identifica fluxo
        if { $cl != 0 } {
                $src($serv:$cli) set class_ $cl
        }
        # Configura gerador
        $expo($serv:$cli) set packetSize_ 1500
        $expo($serv:$cli) set burst_time_ 100ms
        $expo($serv:$cli) set idle_time_ 100ms
        $expo($serv:$cli) set rate_ $rate
}

proc create-exp-u { serv cli rate { cl 0 } } {
        global ns udp sink expo user server agents
        regsub "\[0-9\]*$" $serv {} snode
        regsub "\[0-9\]*$" $cli {} dnode
        # Cria agente receptor de dados
        set sink($serv:$cli) [new Agent/LossMonitor]
        # Insere agente no node cli
        $ns attach-agent $user($dnode) $sink($serv:$cli)
        # Cria agente UDP para transmitir dados
        set src($serv:$cli) [new Agent/UDP]
        # Insere agente UDP no node serv
        $ns attach-agent $user($snode) $src($serv:$cli)
        # Conecta agentes
        $ns connect $src($serv:$cli) $sink($serv:$cli)
        # Insere o gerador Exponencial
        set expo($serv:$cli) [new Application/Traffic/Exponential]
        lappend agents $expo($serv:$cli)
        # Insere o gerador exponencial no agente UDP
        $expo($serv:$cli) attach-agent $src($serv:$cli)
        # Identifica fluxo
        if { $cl != 0 } {
                $src($serv:$cli) set class_ $cl
        }
        # Configura gerador
        $expo($serv:$cli) set packetSize_ 210
```

```
        $expo($serv:$cli) set burst_time_ 10ms
        $expo($serv:$cli) set idle_time_ 100ms
        $expo($serv:$cli) set rate_ $rate
}

proc create-av-rcvr { node } {
        global rcvr ns user
        # Cria agente receptor de dados
        set rcvr($node) [new Agent/LossMonitor]
        # Insere agente no node 2
        $ns attach-agent $user($node) $rcvr($node)
}

proc create-av-sndr { node group { cl 0 } } {
        global udp ns cbr user
        # Cria agente UDP para transmitir dados
        set udp($node) [new Agent/UDP]
        # Insere agente UDP no node 8
        $ns attach-agent $user($node) $udp($node)
        # Agente esta configurado para enviar pacotes multicast
        $udp($node) set dst_addr_ $group
        $udp($node) set dst_port_ 0
        # Cria gerador CBR (trafego de taxa constante)
        set cbr($node) [new Application/Traffic/CBR]
        # Configura o fluxo cbr
        $cbr($node) set packetSize_ 200
        #$cbr($node) set rate_ 40kb
        #$cbr($node) set interval_ 0.25
        #$cbr($node) set random_ 1
        # Insere o gerador CBR no agente UDP
        $cbr($node) attach-agent $udp($node)
        # Identifica fluxo
        if { $cl != 0 } {
                $udp($node) set class_ $cl
        }
}

#################################################################
# Procedure finish:
# - Fecha arquivos de tracing
# - Executa nam
# - Sai do simulador
```

```
#
proc finish {} {
        global ns f nf namfile layfile
        $ns flush-trace
        close $f
        close $nf

#       puts "changing nam layout..."
#       change-layout $namfile $layfile
        puts "running nam..."
        exec ~/ns-allinone-2.1b8a/bin/nam $namfile &
        exit 0
}
#################################################################

#################################################################
# Procedure insert-layout:
# - Altera arquivo out.nam, substituindo as definicoes
#   de node e link pelas definicoes dadas em um arquivo
#   (ex. arquivo de layout salvo no proprio nam)
##proc change-layout { namfile layfile } {
##      exec cp $namfile $namfile.old
##      exec cat $layfile > temp.tmp
##      exec awk {
##              ! /^[nl]/ {
##                      print $0
##              } } $namfile >> temp.tmp
##      exec cp temp.tmp $namfile
##      exec rm temp.tmp
##}
#################################################################

SimpleLink instproc dump-namconfig {} {
        # make a duplex link in nam
        $self instvar link_ attr_ fromNode_ toNode_

        if ![info exists attr_(COLOR)] {
                set attr_(COLOR) "black"
        }

        if ![info exists attr_(ORIENTATION)] {
                set attr_(ORIENTATION) ""
```

```
        }

        if ![info exists attr_(LENGTH)] {
                set attr_(LENGTH) ""
        }

        set ns [Simulator instance]
        set bw [$link_ set bandwidth_]
        set delay [$link_ set delay_]

        $ns puts-nam-config \
                "l -t * -s [$fromNode_ id] -d [$toNode_ id] -S UP -r $bw -D $delay -c
$attr_(COLOR) -o $attr_(ORIENTATION) -l $attr_(LENGTH)"
}

Link instproc length { len } {
        $self instvar attr_
        set attr_(LENGTH) $len
        [Simulator instance] register-nam-linkconfig $self
}
```

LISTAGEM 7
Programa de cálculo da Utilização
(por *Link* e por direção de transmissão)

```
#!/bin/sh
#
# Calcula a utilizacao de determinado link em um sentido (simplex)
#
# Sintaxe:
# BWutil <ininode> <endnode> <initime> <endtime> <bandwidth>
#

echo "Calcula a utilizacao de determinado link em um sentido (simplex):"
echo "No fonte.....: " $1
echo "No destino.....: " $2
echo "Tempo Inicial....: " $3
echo "Tempo Final......: " $4
echo "Largura de banda.: " $5

if [[ $3 > $4 || $3 == $4 ]]; then
        printf "ERRO: tempo final (%i) deve ser maior que tempo inicial (%i)" $4 $3
        exit 1
fi
if [[ $5 == 0 ]]; then
        printf "ERRO: largura de banda (%i) deve ser diferente de zero" $5
        exit 1
fi
if [ ! -f out.tr ]; then
        printf "ERRO: arquivo out.tr nao existe"
        exit 1
fi

awk -v ininode=$1 -v endnode=$2 -v initime=$3 -v endtime=$4 -v bw=$5 '
BEGIN {
        sum = 0
};
$3 == ininode && $4 == endnode && $2 > initime {
        txtime = 8 * $6 / bw
        if ($1 == "r" && $2 < (txtime + initime)) {
                printf("t = %5f ==> Packet %6i with %4i bytes from %2i to %2i leaving buffer\r",
$2, $12, $6, $3, $4)
```

```
                size = ($2 - initime) * bw
                printf("\n    Only last %i bits of the packet is in the link ***\n", size)
                sum = sum + size
        } else if ($1 == "r" && $2 > endtime && $2 < (txtime + endtime)) {
                printf("t = %5f ==> Packet %6i with %4i bytes from %2i to %2i leaving buffer\r",
$2, $12, $6, $3, $4)
                size = 8 * $6 - ($2 - endtime) * bw
                printf("\n    Only first %i bits of the packet is in the link ***\n", size)
                sum = sum + size
        } else if ($1 == "r" && $2 <= endtime) {
                printf("t = %5f ==> Packet %6i with %4i bytes from %2i to %2i leaving buffer\r",
$2, $12, $6, $3, $4)
                sum = sum + 8 * $6
        }
};
END {
        perc_util = sum / (bw * (endtime - initime))
        util = perc_util * bw
        printf("\nPorcentagem de utilizacao do enlace entre %i e %i = %f%%\n", ininode,
endnode, perc_util*100)
        printf("Vazao do enlace entre %i e %i = %7.3e\n", ininode, endnode, util)
};' out.tr
```

LISTAGEM 8
Programa de cálculo da latência (*delay*)
(por origem e destino)

```
#!/bin/sh
#
# Calcula a latencia media dos pacotes que circulam pela rede
#
# Sintaxe:
# latency <pkttype> <sournode> <destnode> <initime> <endtime>
#

echo "Calcula a latencia media dos pacotes que circulam pela rede:"
echo "Tipo de pacote.: " $1
echo "No fonte.....: " $2
echo "No destino.....: " $3
echo "Tempo Inicial..: " $4
echo "Tempo Final....: " $5

if [[ $4 > $5 ]]; then
        printf "ERRO: tempo final (%i) deve ser maior que tempo inicial (%i)\n" $4 $3
        exit 1
fi
if [ ! -f out.tr ]; then
        printf "ERRO: arquivo out.tr nao existe\n"
        exit 1
fi

awk -v pkttype=$1 -v sournode=$2 -v destnode=$3 -v initime=$4 -v endtime=$5 '
BEGIN {
        sum = 0
        pkts = 0
};
initime < $2 && $2 < endtime && $5 == pkttype {
        # Se pacote do tipo especificado esta saindo de um node,
        # o instante em que ele foi transmitido nao existe (ou seja,
        # o node atual eh o node transmissor) e o tamanho do pacote
        # eh maior que 40 (para excluir os pacotes de handshake do TCP)
        if ($1 == "+" && tx_time[$12] == "" && $6 > 40 ) {
                sub(/.[0-9]$/,"", $9) # Exclui o id da porta do socket-fonte,
                                      # isto e, isola o endereco da fonte
                                          # em $9
```

```
                # Se o node atual eh a fonte do pacote e se eh o node
                # fonte solicitado, grava o instante de tempo de tx
                if ($3 == $9 && $3 == sournode) {
                        tx_time[$12] = $2
                }
        }
        # Se pacote do tipo especificado acabou de ser recebido por um
        # node e o instante em que ele foi transmitido existe
        if ($1 == "r" && tx_time[$12] != "") {
                sub(/.[0-9]$/,"", $10) # Exclui o id da porta do socket-dest,
                                # isto e, isola o endereco do destino
                                        # em $10
                # Se o node atual eh o node-destino solicitado para o caso
                # do multicast, ou se o node atual eh o destino unicast,
                # o pacote foi recebido e podemos calcular a latencia
                if ($4 == destnode && ($10 < 0 || $4 == $10)) {
                        if (rx_time[$12] != "") {
                                print "ERROR: " pkttype " packet must be received by the client
only once"
                                exit 1
                        }
                        rx_time[$12] = $2
                        latency[$12] = rx_time[$12] - tx_time[$12]
                        sum = sum + latency[$12]
                        ++pkts
                        sub(/.[0-9]$/,"", $9)
                        printf("\r%i) %i %i %i %f %f %f   ", pkts, $9, $10, $12, tx_time[$12],
rx_time[$12], latency[$12])
                }
        }
};
END {
        avg = pkts != 0 ? sum / pkts : 0
        printf("\n=====> Pacotes validos = %i\n", pkts)
        printf("=====> Latencia media  = %f\n", avg)
};' out.tr
```

LISTAGEM 9
Programa para cálculo do atraso do pacote, por *Link*

```
#!/bin/sh
#
# Calcula a latencia media dos pacotes que circulam em um Link
#
# Sintaxe:
# LinkDelay <pkttype> <sournode> <destnode> <initime> <endtime>
#

echo "Calcula o atraso, medio, dos pacotes em um Link Simplex:"
echo "Tipo de pacote.: " $1
echo "No fonte.......: " $2
echo "No destino.....: " $3
echo "Tempo Inicial..: " $4
echo "Tempo Final....: " $5

#if [[ $4 > $5 ]]; then
#       printf "ERRO: tempo final (%i) deve ser maior que tempo inicial (%i)\n" $4 $3
#       exit 1
#fi
#if [ ! -f out.tr ]; then
#       printf "ERRO: arquivo out.tr nao existe\n"
#       exit 1
#fi
#
awk -v pkttype=$1 -v sournode=$2 -v destnode=$3 -v initime=$4 -v endtime=$5 '
BEGIN {
        sum = 0
        pkts = 0
};
$3 == sournode && $4 == destnode && \
$2 >= initime && $2 <= endtime && \
($5 == pkttype || pkttype == "all") {
        # Se pacote do tipo especificado esta saindo de um node,
        # o instante em que ele foi transmitido nao existe (ou seja,
        # o node atual eh o node transmissor) e o tamanho do pacote
        # eh maior que 40 (para excluir os pacotes de handshake do TCP)
        if ($1 == "+" && tx_time[$12] == "" && $6 > 40 ) {
                tx_time[$12] = $2
        }
        # Se pacote do tipo especificado acabou de ser recebido por um
```

```
        # node e o instante em que ele foi transmitido existe
        if ($1 == "r" && tx_time[$12] != "") {
                if (rx_time[$12] != "") {
                        print "ERROR: " pkttype " packet must be received by the client only
once"
                        exit 1
                }
                rx_time[$12] = $2
                latency[$12] = rx_time[$12] - tx_time[$12]
                sum = sum + latency[$12]
                ++pkts
                printf("\r%6i) %6i %5s %8.6f %8.6f %8.6f   ", pkts, $12, $5, tx_time[$12],
rx_time[$12], latency[$12])
        }
};
END {
        avg = pkts != 0 ? sum / pkts : 0
        printf("\n=====> Pacotes validos = %i\n", pkts)
        printf("=====> Latencia media  = %f\n", avg)
};' out.tr
```

LISTAGEM 10
Programa para cálculo da Perda de Pacotes na Rede como um todo

```
#!/bin/sh
#
# Calcula a perda de pacotes na rede
#
# Sintaxe:
# PacketLoss
#

echo "Calcula a perda de pacotes na rede."

awk '
$1 == "d" || $1 == "+" {
        if ( $1 == "d" ) {
                ++drops
                printf("Pacote %6i descartado.                          \n", $12)
        }
        else if ( $1 == "+" && isTx[$12] == 0 ) {
                ++pkts
                isTx[$12] = 1
                printf("Pacote %6i transmitido. Total de pacotes transmitidos = %6i\r", $12, pkts)
        }
}
END {
        printf("\n=====> Pacotes transmitidos = %i\n", pkts)
        printf("=====> Pacotes perdidos  = %i\n", drops)
        printf("=====> Perda de Pacotes  = %f%%\n", ((drops / pkts) * 100))
}' out.tr
```

ANEXO II – CRIAÇÃO de um NOVO PROTOCOLO no ns-2

A "receita" para criar um novo protocolo no ns-2 segue as seguintes etapas:

1) criação de uma nova estrutura de cabeçalho (escrever Xmp.h e Xmp.cc)

- criação de classe estática para "linkar" com Otcl (incluir em packet.h)
- habilitação do novo cabeçalho em Otcl (incluir em tcl/lib/ns-packet.tcl)

2) criação de um novo agente (subclasse da classe Agent), em Otcl e C++

- decidir a posição do agente na hierarquia (p.ex., acima do IP)
- "Linkar" com uma classe Otcl
    - através da TclClass e TclObject
    - através do TclObject::bind()
- Agent::command (int argc, const char* const* argv) em Xmp.cc
- Agent::recv (Packet* pkt, Handler*) em Xmp.cc
- comunicação Otcl → C++, através de command() e tcl.result()
- comunicação C++ → Otcl, através de otcl::tcl.eval ("string Otcl")

No presente caso é necessário uma lógica de roteamento ainda não existente que é a do protocolo aqui proposto. Portanto, temos que codificar um novo Objeto gerenciador do roteamento nos nós (ns-xmp.tcl).

Alguns arquivos devem ser alterados de modo a que o Simulador reconheça o novo cabeçalho:

- no arquivo packet.h introduzir a linha PT_XMP ao final, mas antes de PT_TYPE em enum packet_t{ ... }

- ainda no mesmo arquivo introduzir a linha name_[PT_XMP] = “Xmp” em class p_info { ... }
- em tcl/lib/ns-default.tcl introduzir “Agent/Xmp set packetSize_ 3
- em tcl/lib/ns-packet.tcl introduzir Xmp em foreach prot { ... }

ANEXO III

PLATAFORMA UTILIZADA

Notebook Acer Extensa 390

- 96M RAM
- 2 Gbytes HD
- 166 MHz clock

Sistema Operacional Linux
- Mandrake 9.0 (dolphin) for i586
- Kernel 2.4.19-16mdk

Compilador
- gcc 2.5

Sistema gráfico
- xgraph 12.2

Linguagens
- tcl 8.3.2
- tclcl-1.0b11
- otcl-1.0 a7
- c/c++
- awk

Simulador
Ns-2.1b8a
Nam-1.0 a10

Editor
- Kate

REFERÊNCIAS BIBLIOGRÁFICAS:

[1] - MARTINS, O. B., **A Educação Superior à Distância e a Democratização do Saber**. Vozes, 1991.

[2] - FISHER G. (1999)., **Lifelong Learning: Changing Mindsets**. In: ICECE99 1 CD-ROM.

[3] - ALMEIDA, F. J. de, *et al*., **Educação a Distância**. PUC-SP, 2001.

[4] - VALENTE, J. A., **Criando oportunidades de aprendizagem continuada ao longo da vida**. Pátio Revista Pedagógica, ano IV, No. 15, pp.8-12, 2000, Nov 2000/Jan 2001.

[5] - BRANDÃO, Z. *et al*., **A Crise dos Paradigmas e a Educação**. São Paulo: Cortez, 2000.

[6] - MORAES, M. C. (2001)., **Tecendo a rede: mas com que paradigma?**. In M.C. Moraes (org.) *Fundamentos e práticas em educação a distância*. Campinas: UNICAMP/NIED. No prelo.

[7] - MORIN, E., **Os sete saberes necessários à uma educação do futuro**. São Paulo: Cortez, 2000.

[8] - DEMO, P., **Curso Superior de Formação de Professores para Educação Inicial, à Distância**. UnB, 1988.

[9] - NISKIER, A., **Virtualmente Correto.** Ensino Superior, no. 25, pp 16-17, 2000.

[10] - HARRIS, D. A., **Online Distance Education in the United States**. IEEE Communications Magazine, v. 37, pp. 87-91. 1999.

[11] - NARDI, B. A. (Ed.), **Context and consciousness: activity theory and human-computer interation**. Cambridge. MIT Press, 1996.

[12] - UBELL, R., **Engineers turn to e-learning.** IEEE SPECTRUM Online, Vol.37, no. 10, October 2000.

[13] - MENASCÉ, D. A.; ALMEIDA, V. A. F., **Capacity Planning for Web Performance, Metrics, Models, & Methods.** Prentice Hall PTR, 1998.

[14] - SYED, M. R., **Diminishing the Distance in Distance Education**. IEEE MULTIMIDIA – July/September 2001, pg. 18-20.

[15] - ANIDO-RIFÓN L. *et al*., **Internet-Based Access to a Microprocessor Laboratory using the Java/Corba Paradigm**. ICECE2000.

[16] - LÉVY, P., **As Tecnologias da inteligência**. Rio de Janeiro: Editora 34, 1993.

[17] – BOURNE, J.R. *et al*., **Paradigms for On-Line Learning: A Case Study in the Design and Implementation of an Asynchronous Learning Networks (ALN) Course**. JALN, Nashville, v. 1, Issue 2, August 1997.

[18] – MOONEN J., **The Efficiency of Telelearning**. JALN**,** v. 1, Issue 2, August 1997.

[19] – HANSEN, E., **The role of interactive video technology in higher education: Case study and proposed framework**. Education Technologyk, pps 13-21, September 1990.

[20] – RAMOS, E. M. F.; ROSATELLI, M. C. e WAZLAWICK, R. S., **Informática na Escola – um Olhar Multidisciplinar**. Fortaleza: Editora UFC, 2003.

[21] –UTIYAMA, C. H., **E-Learning: Vai ou não Vai?.** Revista VOCÊ S.A., pps 48-51, Maio 2002.

[22] –PADUAN, R., **E aí, parceiro?**. Revista EXAME, pps 80-82, Maio 2002.

[23] - ILLICH, I., **Sociedade sem Escolas**. Trad. Lúcia M.E.Orth, Petrópolis: Vozes, 1976.

[24] - RODRIGUEZ, F. G. *et al*., **Creating a High School Physics Video-Based Laboratory.** IEEE Multimedia-July/September 2001, pg. 78-86.

[25] - FERNANDEZ, D. *et al*., **Multimedia Services for Distant Work and Education in na IP/ATM Environement**. IEEE Multimedia 2001, pg.68-77.

[26] – **FÓRUM EM DEFESA DA ESCOLA PÚBLICA**. Vários autores. Lei de Diretrizes e Bases da Educação Nacional- 9394/96. Curitiba: APP-Sindicato, 1997.

[27] – CARELESS, J., **Distance Learning via Satellite: Celestial Curriculum**. VIA SATELLITE, September 2001, pag. 16 – 24.

[28] – DONOVAN, M., Ph.D; MACKLIN S., **One Size Doesn't Fit All Designing Scaleable, Client-Centered Support for Technology in Teaching**. University of Washington, Seattle, Washington. December 1998.

[29] – KOMIYA, R., **Application of Virtual Reality Technologies for Future Telecommunications Systems**. GlobeCom2000.

[30] – STEVENS, W. R., **TCP/IP Illustrated**. vol 1, Addison-Wesley, 2000.

[31] – Estrin D., *et al*., **Request for Comments: RFC 2117 – Protocol Independent Multicast-Sparse Mode (PIM-SM)**: Protocol Specification. Disponível em: <http://www.ietf.org/rfc/rfc2362.txt>. Acesso em: 22 Jun, 2001.

[32] –MURHAMMER, M. W. *et al*., **TCP/IP Tutorial and Technical Overview**. MAKRON Books, 2000.

[33] – FALL K., VARADHAN K., **The ns Manual**. The VINT Project, Disponível em: <http://www.isi.edu/nsnam/ns-documentation.html.>. Acesso em: 22 Jun, 2001.

[34] – McCANNE, S., **Scalable Multimedia Communication with Internet Multicast, Light-weight Sessions, and the Mbone**. Disponível em: <http://lecs.cs.ucla.edu/lecs-reading/spring2001/mccanne-ieee98.pdf> Acesso em: 13 Jul, 2005.

[35] – MACEDONIA, M., BRUTZMAN, D. MBONE, **The Multicast BackbONE**. Disponível em: <http://www-mice.cs.ucl.ac.uk/multimedia/projects/mice/mbone-review.html>. Acesso em: 13 Jul, 2005.

[36] – SULLIVAN, A., **3-DEEP**. IEEE Spectrum, April 2005, pag. 22 – 27.

GLOSSÁRIO

A

Administrador de rede
Pessoa responsável pela operação e pela manutenção diária de uma rede de computadores.

Aplicação cliente-servidor
Aplicações que compõem um sistema distribuído. Composta de pelo menos um computador que atue como servidor e um ou mais computadores atuando como cliente.

Applet
Um programa que é escrito na linguagem Java e é inserido em páginas HTML. Para que um *applet* seja lido é preciso que o *browser* interprete a linguagem Java.

ARP
Address Resolution Protocol. Protocolo do conjunto TCP/IP, utilizado para vincular um endereço IP a um endereço físico de *hardware*. Refere-se à camada Internet do modelo de 4 camadas do TCP/IP.

ARPA
Advanced Research Projects Agency. Agência de Projetos de Pesquisa Avançada. Subordinada ao Departamento de Defesa Americano.

ARPANET
ARPAnetwork (Rede da ARPA). Uma das redes que deu origem à Internet.

Arquitetura 3-tier
Arquitetura de três camadas. Arquitetura cliente/servidor com três processos separados, normalmente rodando em plataformas diferentes: a interface que roda no computador do usuário (um *browser*, por exemplo), os módulos que processam os dados (em conjunto com um servidor Web) e um sistema de bancos de dados.

Arquitetura cliente-servidor
É a arquitetura de desenvolvimento de sistemas, precursora da Arquitetura de 3 camadas. Na cliente-servidor há um módulo responsável pela armazenagem de dados, o servidor, e um módulo responsável pelo processamento e apresentação, o cliente.

Arquitetura de rede
Conjunto de definições sobre a hierarquização e organização dos componentes de uma rede. Um

bom projeto de arquitetura garante desempenho, manutenção baixa, necessidades atendidas, segurança, entre outras coisas, sendo fundamental para uma rede funcionar bem.

B

Backbone
Tronco principal de uma rede, ao qual podem estar ligados diversos roteadores, cada um com um segmento de rede específico. É composto por linhas de conexão de alta velocidade, que se conectam às linhas de menor velocidade.

Banda de rede
Largura de banda. Termo que designa a quantidade de informação passível de ser transmitida por unidade de tempo, num determinado meio de comunicação (fio, onda rádio, fibra óptica etc.). Normalmente medida em *bits*, *kilobits* ou *megabits* por segundo.

Base de dados
Conjunto de dados organizados de forma específica que permite fácil acesso por parte dos programas de aplicação. Banco de dados.

BGP-4
Border Gateway Protocol. Protocolo usado entre roteadores de *backbones* e demais sistemas autônomos em toda a Internet.

Binário
Aquele que usa apenas dois dígitos para representar caracteres, o 0 (zero) e o 1 (um). É a base numérica sob a qual programas de computador são modelados.

Bit
Dígito binário. Um *bit* é a menor unidade de armazenamento em um computador. Um *bit* pode armazenar dois valores: 0 ou 1. Todos os dados usados nos computadores são armazenados a partir da combinação de bits.

Bloco de Endereços IP
Conjunto de endereços IP, designados por um *backbone* à uma rede conectada. Um bloco pode ser composto de 8, 16, 32, 64, 128 ou mais endereços IP, desvinculado do conceito "antigo" de classes IP (A, B, C).

Blocos IP Reservados
Conjuntos de endereços IP reservados para uso em redes privadas, cujo roteamento não é anunciado na Internet. Os blocos reservados são três: redes tipo classe A (10.0.0.0); redes tipo classe B (de 172.16.0.0 até 172.31.0.0) e redes tipo classe C (de 192.168.0.0 até 192.168.255.0).

Broadcast
Enviar uma mensagem a todas as estações ou a todas as estações pertencentes a uma mesma classe.

Browser
Aplicação cliente utilizada para acesso a páginas na Web. Os *browsers* que dominam o mercado são o *Netscape Navigator* e o *Internet Explorer*. Os *browsers* são capazes de se comunicar com um servidor Web por meio do protocolo HTTP e interpretam documentos contendo HTML, JavaScript e Java, dentre outras tecnologias.

Byte
Seqüência de oito *bits* que forma a unidade básica de informação à memória do computador. Um *byte* poder ser representado por um número decimal entre 0 e 255.

Bytecode
Código de *bytes*. Instruções de código *byte*. Formato de programas Java gerados após o processo de compilação através do compilador Java.

C

C++
Linguagem de programação orientada a objeto.

Cabeamento
Infra-estrutura que liga fisicamente os computadores.

Cabeçalho
Informações de controle incluídas no ínicio de um pacote do TCP/IP: endereço IP de origem, endereço IP de destino, porta TCP de origem, porta TCP de destino e *flags* TCP.

Camada de aplicação
Sétima camada do modelo OSI, quarta do modelo TCP/IP. Trata da transferência de informações entre a rede e os aplicativos. O HTTP, o FTP, SMTP e POP3 são exemplos de protocolos de aplicação usados na Internet.

Camada de rede
Terceira camada do modelo OSI, segunda do modelo TCP/IP. Trata do estabelecimento de rotas para o transporte de dados e da interconexão de redes. Na Internet, o IP é o protocolo padrão usado nesta camada.

Camada de transporte
Quarta camada do modelo OSI, terceira do TCP/IP. Trata da transmissão confiável dos dados. O TCP é o principal protocolo envolvido nesta camada, na Internet.

Camada física
Primeira camada do modelo OSI e do TCP/IP. Define protocolos para *hardware* de interface, cabeamento, meio de comunicação. O exemplo mais comum para esta camada é o ethernet.

CGI
Common Gateway Interface. Interface padrão que permite aos servidores Web executar aplicativos externos, possibilitando a criação de *sites* dinâmicos.

Classe de endereço
Bloco de endereços IP. Uma classe pode ter um número máximo definido de endereços, assim como na Internet podem existir um número limitado de classes. São amplamente difundidas as classes do tipo A, B e C.

Cliente
Software que requisita serviços de um servidor e são normalmente utilizados pelos usuários (exemplo: *browser* Web). O termo "cliente" também pode ser usado para se referir a estações de trabalho, computadores que são usados por usuários e que requisitam serviços de um ou mais servidores em uma rede.

Comitê Gestor (CG)
Órgão que recomenda as regras gerais da Internet, fomenta o desenvolvimento de seus serviços, coordena a atribuição de endereços Internet, registro de nomes de domínio e a interconexão de *backbones* e coleta, organiza e dissemina informações sobre a Internet em âmbito nacional.

Compilador
Um programa que traduz código-fonte em código a ser executado por um computador.

Componente de uma aplicação
Parte de uma aplicação que executa uma função específica.

Conectividade
Ligação entre computadores, entre redes. Um computador que possui conectividade Internet está ligado à Internet.

Conexão
Ligação de um computador com outro computador ou uma rede, de forma a que eles possam enviar e receber dados.

Conexão dedicada
Se refere à uma conexão que é alugada para uso exclusivo, estando funcional 100% do tempo.

Conexão discada
Ligação do seu computador a um computador remoto ou a uma rede, através de uma ligação telefônica normal.

D

DHCP
Dynamic Host Configuration Protocol. Centraliza e gerencia a alocação dinâmica de endereços IP e configurações TCP/IP automáticas a todos os computadores de uma rede.

DHTML
Dynamic HyperText Markup Languagem. Aperfeiçoamento da linguagem HTML e utilização conjunta da mesma com JavaScript, permitindo a criação de páginas com animações e outros efeitos.

DNS
Domain Name System. É um serviço família TCP/IP para o armazenamento e consulta de informações sobre os nomes dos computadores (nomes de domínio) na Internet.

Domínio
O nome de uma rede integrante da Internet.

Download
Transferência de um arquivo de um computador-servidor para um computador-cliente.

DVMRP
Protocolo *Multicast* baseado no protocolo *Distance Vector Routing Protocol*

E

Encapsular
Incorporar, a um determinado protocolo, um outro protocolo de nível mais elevado.

Endereçamento
Processo que permite que dispositivos (computadores, roteadores, placas de rede etc.) possam ser encontrados por outros. Na camada física, o endereçamento está ligado ao endereço de *hardware* das interfaces de rede. Na camada de rede, a uma numeração definida por *software*, como o endereço IP.

Endereço físico
Endereço do *hardware* de um *host* utilizado pelas redes.

Endereço Internet
Pode ser considerado como endereço de *e-mail* como também o endereço de um *site*.

Endereço IP
Número de 32 bits que identifica cada computador (*host*) na Internet ou em uma intranet. Um computador consegue se conectar e trocar informações com outro graças à existência de um endereço IP único para cada um.
Exemplo de endereço IP: 130.213.41.251.

Endereço IP CIDR
Classless Interdomain Routing. Forma de organização dos endereços IP que estabelece o roteamento entre redes através de máscaras de rede, ao invés do uso das classes IP originais. Com o CIDR, a distribuição dos endereços IP é bem mais flexibilizada e o desperdício de endereços é muito menor.

Endereço IP Válido
Endereço necessário para que um computador passe a ter acesso à Internet.

Escalabilidade
Facilidade de expansão do sistema para um maior volume de trabalho.

Estação de Trabalho
Workstation. Computador utilizado pelos usuários em uma rede. Uma estação, ou estação de trabalho, é um computador tipicamente cliente, usado pelos usuários para trabalhar com programas cliente, acessando serviços de computadores servidores.

Espinhas dorsais
Backbones. Tronco principal de uma rede, ao qual podem estar ligados diversas redes. É composto por linhas de conexão de alta velocidade. Um dos primeiros *backbones* Internet do país é o da Embratel, com o qual a maioria das redes de empresas e dos provedores de acesso está conectada.

Ethernet
Tecnologia ou tipo de rede, ou tipo de arquitetura de rede local de computadores (LAN), capaz de interconectar computadores próximos com altas taxas de transmissão (10 Mb/seg). Originalmente desenvolvido pelo *Palo Alto Research Center* (PARC), da Xerox, nos EUA.

Extranet
Rede semi-privada, baseada na mesma tecnologia da Internet, usada para a comunicação entre empresas parceiras, entre empresas e clientes, ou com fornecedores.

F

FAPESP
Fundação de Amparo à Pesquisa do Estado de São Paulo <www.fapesp.br>. Órgão responsável pelo registro e manutenção dos domínios no Brasil. A URL específica para o registro de domínios é <registro.fapesp.br>.

Fax-modem
Equipamento usado para possibilitar a um computador usar uma linha telefônica para conexão com outro computador, e que também permite ao computador enviar e receber *faxes* comuns.

Feedback
Retorno de informação sobre alguma ação executada

Fibra óptica
Meio de transmissão de dados que emprega pulsos de luz enviados através de cabos de fibra de vidro.

Filtro de pacote
Packet filter. Conceito existente em um *firewall* que descarta seletivamente pacotes de dados baseando-se em uma tabela de configuração definida pelo administrador, na qual há regras de decisão a partir de informações TCP/IP dos pacotes.

Firewall
Software (instalado em computador dedicado) que gerencia a ligação de uma rede com outra, filtrando e controlando o acesso à Internet. Existem soluções de *firewall* que integram *software* e *hardware*.

Frame
Uma espécie de moldura, usada para subdividir em uma página Web em um *browser*.

Freeware
Modalidade de *software* gratuito.

FTP
File Transfer Protocol. Protocolo de transferência remota de arquivos. Bastante usado para a transferência de programas entre computadores da Internet.

G

Gateway
Termo utilizado tanto para designar dispositivos que interligam duas ou mais redes, como também para aplicações que fazem a tradução entre padrões diferentes (por exemplo, de um correio eletrônico proprietário para o correio eletrônico Internet).

Groupware
Trabalho colaborativo através de rede de computadores. Permite que os diversos usuários da rede, usando programas apropriados, troquem informações e se comuniquem

H

Hardware
Termo utilizado para designar os elementos materiais de um computador e seus periféricos. Engloba a parte mecânica, magnética, eletromecânica, elétrica e eletrônica.

HDLC
High-Level Data Link Control. Protocolo padrão do nível de enlace de dados, tipicamente utilizado em conexões entre roteadores.

Hipertexto
Texto não linear, com ligações, ou *links*, entre documentos. Trata-se de um conceito antigo, dos anos 60, implementado com grande sucesso através da WWW.

Hit
Arquivo enviado por um servidor Web. Se um *site* teve 1000 *hits*, 1000 elementos entre imagens, páginas HTML e outros, foram enviados para usuários.

Home-page
Página incial de um conjunto de páginas (*site*). Muitas vezes um *site*, também é chamado de *home-page*.

Host
Equipamento conectado à Internet.

HTML
Hyper Text Markup Language. Linguagem de Marcação Hipertexto. Trata-se da linguagem básica para a criação de documentos para a Web. É interpretada pelos *browsers* e tem como importantes características a portabilidade e a alta flexibilidade.

HTTP
Hyper Text Transfer Protocol, ou Protocolo de Transferência de Hipertexto. Trata-se do protocolo criado especificamente para a *World Wide Web*, sendo fundamental para o seu funcionamento. É utilizada pelos *browsers* e servidores Web.

HTTP-NG
HTTP - New Generation. Versão mais recente, em desenvolvimento, do protocolo HTTP.

HTTPS
HTTP seguro. Protocolo HTTP oferecendo segurança para transações pela Internet, via Web, funcionando em conjunto com o SSL, com garantias de autenticação, confidencialidade e integridade.

I

IAB
Internet Architecture Board, órgão ligado ao ISOC. Responsável por diretrizes e padrões referentes ao protocolo TCP/IP.

IANA
Internet Assigned Numbers Authority, órgão ligado ao ISOC. Responsável pela organização dos nomes de domínio e endereços IP.

ICMP
Internet Control Message Protocol. Protocolo de manutenção que trata as mensagens de erro a serem enviadas. Protocolo referente a camada Internet.

IETF
Internet Engeneering Task Force. Comitê responsável por construir os padrões da Internet. Órgão ligado ao ISOC.

IGMP
Protocolo usado para comunicação entre os *hosts* de uma LAN e o roteador ligado a esta LAN de modo a especificar quais *hosts* pertencem a qual grupo (sigla de *Internet Group Management Protocol*).

Interface
Qualquer dispositivo que intermedia informações, seja entre componentes de um computador, seja entre computadores interligados, seja entre computadores e usuários.

Internet
"Rede das redes", originalmente criada nos EUA, se tornou um conjunto internacional de redes interligadas, que utilizam protocolos da família TCP/IP. Provê serviços como correio eletrônico, grupos de discussão, execução remota e transferência de arquivos de dados.

InterNic
Internet Network Information Center. Órgão responsável pelo registro e manutenção dos domínios *top level* ".com", ".edu", ".net" e ".org", todos dos EUA.<www.internic.net>.

Intranet
Rede privada baseada na mesma tecnologia da Internet (TCP/IP), usada tipicamente para a comunicação dentro de uma empresa, com serviços como *e-mail* e Web internos.

IP
Internet Protocol. Responsável pela identificação das máquinas e redes e encaminhamento correto das informações entre elas.

IPv4
IP versão 4. A atual versão do protocolo IP.

IPv6
Também chamdo de IPng. Nova especificação do *Internet Protocol*, que permitirá um maior número de servidores conectados à Internet e maior otimização do tráfego de dados.

ISO
International Standards Organization. Organização internacional sediada na Suiça, para a definição de normas e padrões.

ISOC
Internet Society. Órgão que coordena o processo de padronização na Internet.

ISP
Internet Service Provider. Provedor de serviços Internet, que fornece acesso, hospedagem de *sites* e/ou outros serviços.

J

Java
Linguagem orientada a objeto, criada pela Sun Microsystems. Sua principal característica é a portabilidade, que permite aos programas em Java funcionarem em quase todas as plataformas. Pequenos programas Java (*applets*) podem ser transmitidos juntamente com as páginas Web.

JavaScript
Linguagem de *script*, utilizada em conjunto com o HTML, para tornar páginas da Web mais interativas. Desenvolvida pela Netscape, foi orginalmente criada com o nome de LiveScript. É interpretada pelo *Netscape Navigator* e, parcialmente, pelo *Internet Explorer*.
JavaScript é uma marca registrada.

JPG
JPEG. *Joint Photographers Experts Groups*. Formato de arquivos de imagens. Tem alta capacidade de compactação e funciona bem com imagens fotográficas. Muito utilizado na Web.

K

Kbps
Kilobits por segundo. Taxa de transferência de dados de um computador a outro, de mais de mil (1.024) bits (dígitos binários) por segundo.

Kbyte
Kilobyte. Medida de capacidade de memória que corresponde a 1.024 *bytes*.

L

LAN
Local Area Network. Rede local. Dois ou mais computadores conectados por uma rede, usualmente com cabos.

Largura de banda
Bandwidth. Termo que designa a quantidade de informação passível de ser transmitida por unidade de tempo, num determinado meio de comunicação (fio, onda rádio, fibra óptica etc.). Normalmente medida em *bits*, *kilobits* ou *megabits* por segundo.

Layer
Camada. Em aplicativos de desenho e edição de imagens, um dos níveis de uma ilustração. Também é um conceito utilizado no HTML Dinâmico.

Layout de rede
Topologia de rede. A topologia física descreve de que maneira os cabos e equipamentos estão dispostos.

Ligação discada
Conexão feita através de uma ligação normal, por um aparelho telefônico.

Linha analógica
Linha convencional. Utiliza métodos de transmissão desenvolvidos para sinais de voz. Como foi criada apenas para a faixa de freqüencia da voz, é limitada em sua capacidade de transportar sinais digitais de alta velocidade.

Linha digital
Em termos genéricos, uma linha que em vez de usar frequências analógicas moduladas para a transmissão de informações, utiliza seqüências binárias, isto é, dados no formato digital.

Linha privativa
Circuito de comunicações reservado para a utilização permanente de um cliente. Também chamada de LP. Uma linha privativa liga dois pontos, que podem ser filias de uma empresa, empresas diferentes, uma empresa e um *backbone*, um usuário e um provedor de acesso etc.

Link
Referência para outra página, ou outro documento, dentro de um documento hipertexto. É também utilizada para denominar uma conexão Internet.

Linux
Implementação do sistema operacional Unix, disponível para processadores Intel. Nome derivado do nome do autor do núcleo deste sistema, Linus Torvalds. É distribuído gratuitamente.

Load balancing
Recurso que distribui o tráfego de maneira otimizada entre vários servidores.

Login
É o identificador do usuário no sistema. Através dele, um sistema é capaz de saber seus direitos de acesso.

LPCD
Linha privativa para comunicação de dados. Numa LPCD, a empresa operadora da linha garante sua qualidade. Isto é, quando se contrata uma linha que transmita, por exemplo, 128Kbits/segundo, há a garantia de funcionamento nesta velocidade. Isto não ocorre em linhas privativas comuns, chamadas de LPs.

M

Mainframe
Computador de grande porte, geralmente acessado por terminais burros (nos quais não há processamento). Um *mainframe* centraliza todo o processamento e o armazenamento de dados.

Máscara de rede
Subnet mask. Endereço de 32 bits cujo objetivo é separar a porção do endereço IP que identifica o *network ID* da que identifica o *host ID*. Usada para determinar se um endereço IP qualquer pertence à rede local ou se é externo.

Mbps
Megabits por segundo. Taxa de transferência de dados de mais de um milhão (1.048.576) de *bits* (dígitos binários) por segundo.

Memória RAM
Random Access Memory. Memória de acesso aleatório. Também conhecida como memória de leitura e gravação, é a memória utilizada para executar programas aplicativos quaisquer. Quanto mais memória um computador tiver, mais programas pode executar simultaneamente.

Modem
MOdulador/DEModulador. Equipamento usado para possibilitar a um computador usar uma linha telefônica para conexão com outro computador.

MRT
Tabela construida pelos roteadores para encaminhar os pacotes XMP.

Multi homed
Redes que possuem mais de um *link* com a Internet, visando ter maior disponibilidade e velocidade, não ficando na dependência de uma única ligação.

Multicast
Protocolo que permite que um pacote seja enviado de um ponto para muitos (*one-to-many*) com a mesma carga de tráfego na rede, independente do número de receptores.

Multimídia
Sistema que utiliza várias formas de mídia. Mensagens e programas que podem incluir voz, som, imagem e/ou dados.

N

NAT
Network Address Translation. Traduz endereços válidos na Internet para endereços reservados numa intranet e vice-versa. Permite que uma rede local use um conjunto de endereços IP para o tráfego interno e um outro para o tráfego externo, protegendo-a e permitindo melhor gerência dos endereços.

Navegador
Browser. Aplicação cliente utilizada para acesso a páginas na Web. Os navegadores que dominam o mercado são o *Netscape Navigator* e o *Microsoft Internet Explorer*.

NetBios
Network Basic Input/Output System. Um padrão desenvolvido pela IBM para acessar os serviços da rede por meio do sistema operacional do computador.

Nome de domínio
O nome de uma rede integrante da Internet.

Número IP
O mesmo que endereço IP. Endereço numérico que a Internet necessita para enviar fluxos de pacotes entre computadores. Todo computador *host* ligado à Internet possui um endereço, chamado de endereço IP, que é um número único de 32 bits.

O

Oak
Nome originário da linguagem Java, em sua fase de desenvolvimento.
Oak é uma marca registrada.

Objetos
São os blocos principais da programação orientada a objetos. Cada objeto é uma unidade de programação consistindo em dados e funcionalidade.

Online
Conectado com algum serviço remoto no momento. Pode-se dizer também de fatores ligados a Internet. Por exemplo: uma versão *online* de um jornal, publicidade *online*.

OSI
Open Systems Interconnection. É um modelo conceitual de protocolo com sete camadas definido pela ISO, para a compreensão e o projeto de redes de computadores. Trata-se de uma padronização internacional para facilitar a comunicação entre computadores de diferentes fabricantes.

OSPF
Protocolo usado para manter atualizada a tabela de roteamento dos roteadores que compõe grandes redes (sigla de *Open Shortest Path First*).

P

Pacotes
Blocos de dados enviados pela rede, que incluem tanto informações básicas quanto a codificação específica do protocolo usada para identificar e processar o bloco. A informação é dividida em pacotes para que a sua transmissão na rede seja mais eficiente.

Padrão aberto
Um padrão de domínio público, ou seja, que qualquer pessoa ou empresa pode ter acesso e implementar, sem ter que ser feito o pagamento de *royalties*.

Padrão fechado
Padrão proprietário. Somente a empresa que o criou pode utilizá-lo ou licenciá-lo para terceiros.

Página dinâmica
Uma página que é construida a partir de uma aplicação Web, normalmente com informações armazenadas em bases de dados.

Periférico
Dispositivo exterior à unidade de processamento, podendo ser de entrada/saída (impressoras, terminais, teclados) ou de memória (discos, tambores ou fitas magnéticas).

PIM
Protocolo *Multicast* atualmente usado em grandes redes que apresenta duas modalidades: o SM (*Sparse Mode*) e o DM (*Dense Mode*). O primeiro é usado para redes de grande abrangência geográfica e pouca largura de banda, enquanto o segundo é usado para redes mais concentradas onde largura de banda não é um recurso escasso. (sigla de *Protocol Independent Multicast*).

PIR
Ponto de Interconexão de Redes. Ponto de troca de tráfego entre *backbones* distintos.

Plataforma
Conjunto que abrange a combinação do tipo de CPU com o sistema operacional usado (*hardware* e *software*) de um determinado computador.

Plug-in
Programa que é instalado adicionalmente, ou integrado, para acrescentar comandos ou funções a outro programa. No caso da Internet, utiliza-se *plug-ins* para complementar os *browsers*.

Portabilidade
Capacidade de um programa ser executado, sem apresentar mudança de comportamento, em computadores com diferentes plataformas ou sistemas operacionais.

Portal
Sites com um vasto conjunto de benefícios e serviços gratuitos de forma a se congregar uma enorme quantidade de usuários e auferir lucros através da venda de espaço para marketing. *Sites* de busca, notícias, personalização são alguns exemplos.

PPP
Point-to-Point Protocol. Um dos protocolos mais conhecidos para acesso telefônico. É considerado o sucessor do SLIP por ser confiável e mais eficiente. Protocolo referente à camada de rede na Internet.

Programa cliente
Um *software* qualquer usado para se comunicar e obter dados de um determinado programa servidor, que normalmente está em outro computador. Por exemplo, um *browser* é um programa cliente que se comunica com um servidor Web.

Programação estruturada
É uma forma de programação, a mais utilizada até hoje. A programação orientada a objetos surgiu como uma opção para a programação estruturada.

Programação orientada a objetos (OOP)
Uma metodologia de desenvolvimento de *software* que se baseia numa analogia dos programas com o mundo físico. Surgiu como uma opção para a programação estruturada.

Programa servidor
Um *software* que provê determinado serviço a um programa cliente rodando em outro computador. Por exemplo, um servidor Web provê páginas Web para *browsers*, que são programas cliente.

Protocolo
Conjunto de padrões e procedimentos que possibilitam a comunicação entre computadores.

Provedor de acesso
Empresa especializada em fornecer acesso à Internet.

Proxy
Software utilizado, freqüentemente como parte de uma solução de *firewall*, que pode trazer alguns dos seguintes benefícios à rede: Maior segurança, maior agilidade na obtenção das páginas da Internet e economia de banda.

Push
Também conhecido como *webcasting*. É uma tecnologia que permite a atualização freqüente de informações no *desktop* do usuário, a partir de uma conexão do mesmo com a Internet.

R

Rede
Dois ou mais computadores conectados com o propósito de trocar mensagens e/ou compartilhar dados e recursos do sistema. Termo também usado para designar a Internet, quando escrito com a primeira letra maiúscula.

Rede Ethernet
Tipo de rede local de computadores, desenvolvida pela Xerox em 1976 e muito difundida hoje. Segue o padrão IEE 802.3. Possui velocidade de transmissão 10 Mbps.

Rede fechada
Rede privada. Rede específica de uma determinada empresa, fechada ao público em geral.

Rede interna
Rede privada de uma determinada empresa fechada ao público em geral. Se estiver baseada no protocolo TCP/IP, pode ser considerada como uma Intranet.

Rede local
Dois ou mais computadores em rede e relativamente próximos entre si, que podem trocar dados por meio de cabos ou de dispositivos de comunicação.

Rede privada
Rede específica de uma determinada empresa, fechada ao público em geral.

Registro de nome de domínio
Ato de adquirir um endereço Web, junto ao órgão responsável. Em âmbito nacional, o órgão responsável é a FAPESP.

Relaying
Prática do uso de servidores de correio eletrônico de terceiros para o envio de *e-mails*. Muito usado para a prática de *spam* (envio não autorizado de *e-mails*).

RFC
Request For Comments ou Solicitação Para Comentários. Os RFCs são documentos de trabalho que a comunidade Internet emprega para desenvolver e registrar padrões.

RIP
Protocolo de roteamento usado em pequenas redes que tem sido substituido pelo OSPF (sigla de *Routing Internet Protocol*).

Roteador
Computador especial, que tem a tarefa de controlar o tráfego na conexão entre duas ou mais redes. Diferentemente das pontes (*bridges*), os roteadores são dispositivos inteligentes, sendo capazes de determinar quais as melhores rotas para se enviar pacotes entre redes.

S

Script
Programas que realizam operações específicas.

Server side
Parte referente ao servidor numa aplicação cliente/servidor.

Servidor
Computador dedicado que armazena dados e processa as solicitações para acessar ou transmitir os dados armazenados.

Servidor de acesso
Hardware que provê acesso remoto a um computador. Muito utilizado em provedores de acesso, para fornecer acesso Internet a seus clientes.

Servidor de DHCP
Tipo de servidor que facilita e permite a alocação dinâmica de endereços IP e configurações TCP/IP a todos os computadores de uma rede.

Servidor de DNS
Servidores que traduzem nomes de domínio em endereços IP.

Servidor *proxy*
Software utilizado, frequentemente como parte de uma solução de *firewall*, que traz diveros benefícios à rede: Maior segurança, maior agilidade na obtenção das páginas da Internet e economia de banda.

Servidores *root*
Root servers. Servidores centrais de DNS.

Servidor Web
Servidor WWW. Aplicação servidora utilizada para a disponibilização de documentos na Web. Todas as páginas da WWW ou de uma intranet ficam depositadas em um servidor Web, que sabe como atender aos pedidos dos *browsers* através do uso do protocolo HTTP.

Servlets
Pequenos programas em Java que ficam rodando permanentemente no servidor e que funcionam em conjunto com estes.

S-HTTP
Secure HTTP. Versão do HTTP que permite segurança, com encriptação, autenticação e integridade das transmissões entre um *browser* e um servidor Web. Trata-se de um padrão anterior ao SSL e que tipicamente não é usado.

Sistema distribuído
Também chamada de aplicação distribuída, refere-se a uma aplicação que não está restrita a um único computador.

Sistema operacional
Conjunto de programas básicos, responsável pelo controle e supervisão de todas as operações internas de um computador e seu interfaceamento com os periféricos.

Site
Termo associado à presença de uma empresa ou organização na Internet. Não compreende apenas um conjunto de páginas na Web, mas abrange todos os arquivos mantidos por aquela empresa na Internet (o que inclui programas e documentos de todos os tipos) e ainda serviços como *Newsgroups* e as mais diversas aplicações.

SMTP
Simple Mail Transfer Protocol. Trata-se do protocolo encarregado do roteamento de mensagens de correio eletrônico através da Internet. Quando uma mensagem é enviada a alguém, é através do SMTP que ela chega ao seu destinatário.

Streaming
Tecnologia que permite que um arquivo de áudio ou vídeo seja executado por um *browser* ao mesmo tempo em que *download* está sendo feito, ou seja, o usuário vê o resultado do arquivo enquanto ele está chegando.

Subrede
Subnet. Subdivisão de uma rede.

T

Tag
Marcação de formatação do HTML. Através de *tags*, pode-se formatar o texto (mudar a cor, o tamanho, o tipo de letra, efeitos como negrito e itálico etc.), inserir *links*, imagens e diversas outras funcionalidades em páginas da Web.

TCP
Transmission Control Protocol. A parte do conjunto de protocolos TCP/IP que controla o transporte de dados. Assegura que os dados transmitidos pelas redes TCP/IP alcancem seu destino, detectando e reenviando pacotes perdidos ou adulterados.

TCP/IP
Transmission Control Protocol / Internet Protocol. Conjunto de protocolos padrão na Internet.

Telefonia via IP
Serviço que permite que um usuário da Internet possa se comunicar, como se estivesse num telefone, com qualquer pessoa.

Telnet
Serviço que permite conexão com computadores remotos de um sistema ligado à Internet, como num terminal.

Terminal burro
Dumb terminal. Equipamento para comunicação com computadores de grande porte, limitado apenas às operações de entrada de dados por teclado e apresentação de informações no vídeo.

Timeout
Tempo máximo em que uma conexão é mantida sem resposta da outra parte.

Top-Level Domain
São as letras que vêm após o último ponto de um nome de domínio. No Brasil, são sempre ".br".

Topologia de rede
O *layout* de uma rede. A topologia descreve de que maneira os cabos e equipamentos estão dispostos.

U

UDP
User Datagram Protocol. Um dos protocolos de transporte, utilizado normalmente em aplicações de áudio e vídeo, ao invés do protocolo TCP.

URL
Uniform Resource Locator. Endereço que permite identificar e acessar um serviço na Web. Por exemplo, a URL abaixo aponta para o site na WWW da Embratel:
< http://www.embratel.com.br>.

V

VPN
Virtual Private Network ou Rede Privada Virtual. Tipo de ligação entre redes intranet que utiliza a Internet como meio de conexão.

VRML
Virtual Reality Modeling Language. Linguagem utilizada para desenvolvimento de páginas que geram ambientes de realidade virtual.

W

W3C
World Wide Web Consortium. Organização responsável pelo desenvolvimento dos padrões da Web.

WAN
Wide Area Network. Redes de computadores de larga abrangência geográfica.

Web
A Web é um serviço desenvolvido para a publicação de informações em hipertexto na Internet. Em intranets, empresas podem criar uma Web interna, usando a mesma tecnologia.

Windows
Sistema operacional gráfico desenvolvido pela Microsoft.
Windows é uma marca registrada.

World Wide Web
WWW. O serviço mais popular da Internet. De uso simples, trouxe milhares de usuários para a Internet e incentivou o comércio eletrônico.

X

X-Windows
Interface gráfica utilizada em sistemas operacionais Unix.

XMP
Protocolo *Multicast* que proporciona bidirecionalidade aos participantes do grupo. Sigla de *Switched Multicast Protocol*.

Z

ZIP
Padrão de compactação de arquivos universalmente utilizado, desde a época do MS-DOS, amplamente usado na Internet, pois arquivos compactados ficam menores e podem ser transmitidos mais rapidamente.