UM SERVIÇO ATIVO DE DISTRIBUIÇÃO DE VÍDEO MULTIPONTO

por

Paulo André da Silva Gonçalves

UFRJ

Tese submetida para a obtenção do título de

Mestre em Ciências em Engenharia Elétrica

ao Programa de Pós-Graduação de Engenharia Elétrica

da COPPE/UFRJ

UM SERVIÇO ATIVO DE DISTRIBUIÇÃO DE VÍDEO MULTIPONTO

Paulo André da Silva Gonçalves

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Prof. José Ferreira de Rezende, Dr.

Prof. Luiz Fernando Rust da Costa Carmo, Dr.

Prof. Raimundo Sampaio Neto, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2000

GONÇALVES, PAULO ANDRÉ DA SILVA

Um Serviço Ativo de Distribuição de Vídeo Multiponto [Rio de Janeiro] 2000

XV, 82 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2000)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Distribuição de Vídeo
2. Redes Ativas
3. Multiponto
4. Serviço Adaptativo

I. COPPE/UFRJ II. Título (Série)

Aos meus pais.

Agradecimentos

Aos meus pais, Agnes e Paulo, à minha irmã Simone, e demais familiares pelos incentivos e pelo respeito às minhas decisões.

Aos meus "priminhos" Kleyton e Lucas e "priminhas" Taíssa e Suellen.

Aos meus orientadores, Professores Otto Carlos Muniz Bandeira Duarte e José Ferreira de Rezende, pela sinceridade e o apoio.

Aos demais membros da banca examinadora, professores Luiz Fernando Rust da Costa Carmo e Raimundo Sampaio Neto.

Ao amigo Flavio do GTA pela contribuição para a realização deste trabalho.

Aos demais amigos do GTA: Rubi (pelo companheirismo), Artur (pelo apoio), Baiano, Ronaldo, Renata, LuisH, Daniel, Marcos, Jonny, Belem, Prof. Mauros, Emerson, Antguria, Sílvia, Fred Sauer, Vidal, Alexandre, Bill, Hermes, Marcial, Abdalla, Célio, Faerman, Ildemar, Jucele, Valentim, Ezequiel, Miranda, Duboc, Gardel, Saulo, Roman, Andreamb, Adriana, Iniciações Científicas (Bernardo, Tadeu, Doc, Éric, Pedro, Fred, Zé Carlos, Ingrid, Renzo).

Aos Professores Valmir, Leão, Richard, Seixas, Antônio Cláudio, Aloysio e Lanza.

Ao pessoal da secretaria: Solange, Bia, Evelyn e Wilson.

À CAPES pelo apoio financeiro que tornou possível a realização deste trabalho.

A todos que me incentivaram, contribuindo de forma direta ou indireta para a minha formação profissional.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM SERVIÇO ATIVO DE DISTRIBUIÇÃO DE VÍDEO MULTIPONTO

Paulo André da Silva Gonçalves

Março/2000

Orientadores: Otto Carlos Muniz Bandeira Duarte
José Ferreira de Rezende
Programa: Engenharia Elétrica

As redes ativas proporcionam uma nova forma de se visualizar um ambiente de rede no qual as aplicações podem injetar cápsulas que realizem computações específicas às suas necessidades.

Com base neste novo paradigma, este trabalho propõe um serviço de distribuição de vídeo multiponto flexível e interativo que se adapta dinamicamente aos requisitos distintos de QoS pedidos por cada receptor. Os receptores podem se conectar/desconectar do serviço quantas vezes desejarem e a qualquer instante de uma sessão. Adicionalmente, o serviço elimina a necessidade do estabelecimento de múltiplas árvores multipontos para a distribuição do vídeo em camadas e impede que um receptor, ao tentar receber uma qualidade melhor do vídeo, interfira na qualidade do vídeo sendo recebido por outros receptores. Para evitar o efeito de implosão de cápsulas que chegam ao transmissor, o protocolo possui um mecanismo de fusão de cápsulas.

Simulações foram realizadas com o intuito de estudar o comportamento do mecanismo de fusão utilizado e verificar o seu desempenho para avaliar a escalabilidade do serviço proposto. Foram utilizadas uma topologia em árvore e outra semelhante às encontradas na Internet. Os resultados obtidos mostram índices de filtragem entre 91% e 96% que evidenciam a escalabilidade do serviço ativo proposto.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AN ACTIVE SERVICE FOR MULTICAST VIDEO DISTRIBUTION

Paulo André da Silva Gonçalves

March/2000

Advisors: Otto Carlos Muniz Bandeira Duarte
José Ferreira de Rezende
Department: Electrical Engineering

Active networks are a novel approach to network architecture in which applications can perform customized computations by injecting their programs into the network. These programs are embedded in packets called capsules.

Based on this new paradigm, this work proposes a flexible and interactive service for multicast video distribution. The service adjusts itself to individual Quality of Service (QoS) requirements posed by the receivers. The receivers can connect/disconnect from the service how many times they want and anytime during a session. Futhermore, the service eliminates the need for establishing multiple multicast trees for distributing layered video and it prevents receivers from interfering with one another when they attempt to receive a better video quality. The source learns about the supported transmission rate of the video by receiving feedback capsules from receivers. A fusion mechanism is used to prevent feedback-capsule implosion.

The scalability of the service and the behavior of the fusion mechanism were evaluated via simulation. A network topology model represented by a tree and another that correlates well with Internet structure were used. The results show a capsule filtering level between 91% and 96%, which makes evident the scalability of the service.

Palavras-chave

Distribuição de Vídeo

Redes Ativas

Multiponto

Serviço Adaptativo

Glossário

AE :	Ambiente de Execução.
ANEP :	*Active Network Encapsulation Protocol.*
ANTS :	*Active Network Transfer System.*
API :	Interface de Programação (*Application Programming Interface*).
EWMA :	*Exponential Weighted Moving Average.*
FDDI :	*Fiber Distributed Data Interface.*
IP :	*Internet Protocol.*
IPv4 :	*Internet Protocol version 4.*
LAN :	Rede Local (*Local Area Network*).
MPEG :	*Motion Picture Experts Group.*
ns :	*Network Simulator.*
QoS :	Qualidade de Serviço (*Quality of Service*).
RLM :	*Receiver-driven Layered Multicast.*
TCP :	*Transmission Control Protocol.*
TTL :	*Time-To-Live.*
UCP :	Unidade Central de Processamento.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

RECENTEMENTE, as operadoras de TV a cabo vêm se empenhando em oferecer aos seus clientes acesso direto à Internet. Novas tecnologias como o *cable modem* estão surgindo de modo a permitir o tráfego de dados na malha de cabos destas operadoras. Deste modo, em um futuro próximo, é possível que as redes das operadoras de TV a cabo estejam totalmente integradas à Internet. Esta possibilidade tem gerado novas perspectivas para estas operadoras, as quais poderão oferecer seus serviços para um número elevado de clientes em potencial. Além disto, a integração das redes de TV a cabo à Internet possui outras vantagens para estas operadoras que incluem cobertura em âmbito mundial, redução de custos e aumento de flexibilidade [1].

Os serviços de difusão de TV e de TV a cabo consistem na transmissão de um sinal de banda passante constante para receptores homogêneos. Portanto, a qualidade é fixa e a boa recepção só depende do nível de sinal (relação sinal-ruído) que chega aos receptores. Por outro lado, a distribuição de vídeo pela Internet possui características distintas: a informação de vídeo é transmitida em pacotes que são roteados, nó a nó, da fonte ao destino; a banda passante da rede é compartilhada dinamicamente pelas diversas aplicações em andamento; e os receptores são heterogêneos. Essa heterogeneidade é evidenciada pelas diferentes capacidades de processamento e de banda passante entre a fonte e os receptores. Se a taxa de transmissão for adaptada ao receptor de menor capacidade, então os receptores com uma maior capacidade serão penalizados. Sendo assim, uma única taxa de transmissão falha em acomodar a heterogeneidade dos receptores.

Outra questão importante, é com relação à tarifação dos serviços. Se os receptores são heterogêneos, eles devem receber qualidades diferentes do vídeo e portanto, devem pagar de forma diferenciada. Além disto, por ser um meio interativo, a Internet possibilita aos seus usuários o recebimento de informações de acordo com os seus próprios interesses. A resolução da imagem, o número de cores, zoom etc. poderão ser escolhidos por cada usuário receptor. Sendo assim, as novas aplicações de vídeo pela Internet devem visar diversos níveis de qualidade. Algumas das aplicações de vídeo emergentes na Internet incluem televisão, vídeo-conferência ou conferência multimídia, tele-seminário, vídeo-telefonia, e ensino à distância.

1.1 O Problema da Heterogeneidade

A Internet é formada por redes que possuem características distintas de banda passante. Os usuários podem estar conectados via modem, RDSI (Rede Digital de Serviços Integrados), *cable modem*, ADSL (*Assymetric Digital Subscriber Line*), *wireless modem* ou rede local. A figura 1.1 ilustra a banda passante associada a algumas formas de conexão utilizadas pelos usuários. Diferentes tecnologias de acesso à Internet constituem um grande obstáculo para realizar de forma satisfatória comunicações multimídias de tempo real.

Além disto, deve-se considerar o fato da banda passante da rede ser compartilhada dinamicamente entre as diversas aplicações em andamento. Deste modo, a banda passante total de um enlace pode não estar disponível para uma determinada aplicação. Associado a isto tudo, existe ainda o fato dos computadores possuírem capacidades de processamento diferentes. Assim sendo, mesmo que um receptor esteja conectado à Internet por tecnologias que permitam taxas elevadas de transferência de dados , a capacidade computacional pode ser um fator limitante.

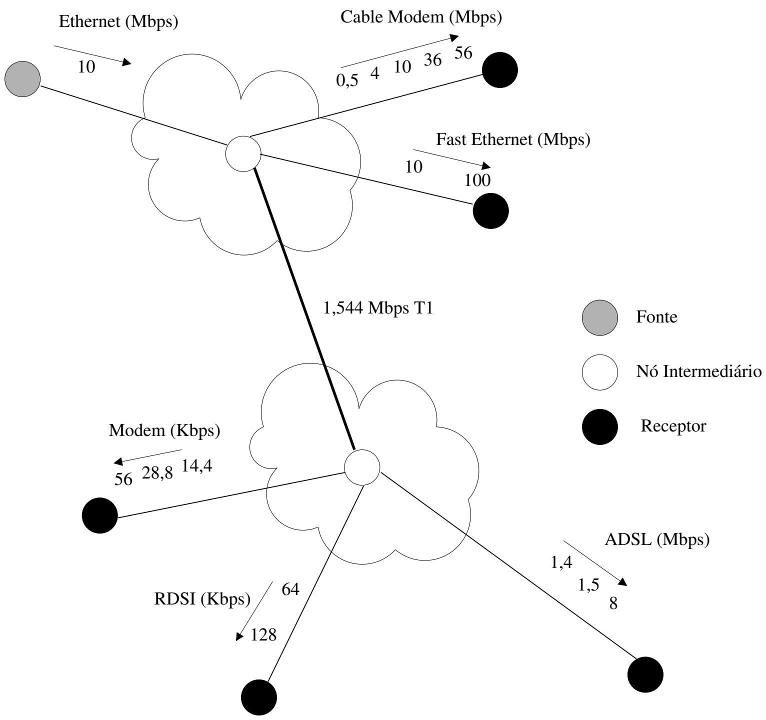


Figura 1.1: Banda passante proporcionada por diferentes formas de conexão.

1.2 A Distribuição de Vídeo na Internet

Existem várias abordagens que visam acomodar diversos níveis de qualidade na distribuição de vídeo na Internet. A primeira delas é conhecida como *simulcast* [2, 3]. Nesta abordagem, a fonte transmite vários fluxos independentes do mesmo vídeo com qualidades distintas. Um exemplo é apresentado na figura 1.2 onde cada fluxo vídeo proporciona uma resolução diferente. Cada receptor, baseado nas suas limitações de banda passante, escolhe qual dos fluxos deseja receber. Esta abordagem resolve o problema de acomodar diferentes níveis de qualidade contudo, a introdução de vários fluxos independentes do mesmo vídeo resulta no uso ineficiente da banda passante da rede.

Uma outra abordagem é a utilização de filtros em pontos específicos da rede. Yedon [4] classifica a filtragem em três tipos principais: filtragem seletiva, transformação e mistura (*mixing*). Os *filtros seletivos* decidem, com base em algum critério, se devem enviar ou descartar pacotes de um fluxo. A *filtragem seletiva* é aplicável ao descarte de sub-fluxos de um fluxo codificado hierarquicamente ou para reduzir a

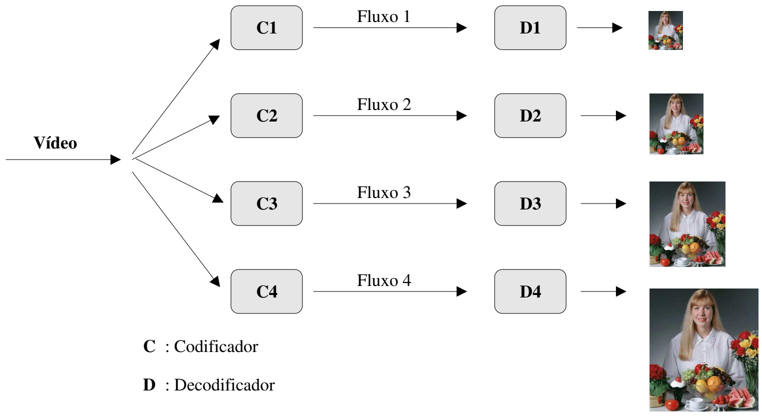


Figura 1.2: Vídeo *simulcast*.

taxa de quadros de um fluxo de vídeo. A *transformação* consiste em reduzir a vazão de um fluxo. Um fluxo de vídeo que utiliza um esquema de cores de 16 bits pode ser transformado para um esquema de cores de 8 bits. A *mistura* é aplicável quando se deseja que múltiplos fluxos de vídeo sejam incorporados a um único fluxo.

Com base na técnica de transformação, Assunção e Ghanbari [5] propõem que um único fluxo de vídeo seja codificado pela fonte e transmitido a uma taxa elevada. Os nós intermediários da rede teriam a função de transcodificar o vídeo (decodificar e recodificar) nos pontos de gargalo da rede. Embora esta solução permita a obtenção de diferentes níveis de qualidade, ela exige que transcodificadores complexos e de alto custo computacional estejam presentes na rede. Adicionalmente, haverá desperdício de banda passante entre a fonte e os pontos de gargalo da rede. Técnicas similares são apresentadas em [6, 7].

Outra abordagem para a distribuição de vídeo com diferentes níveis de qualidade é baseada na codificação de vídeo em camadas ou hierárquica. Um codificador de vídeo hierárquico comprime o vídeo em uma ou mais camadas de diferentes prioridades. A camada com a mais alta prioridade, chamada de camada base, contém as partes mais importantes do fluxo de vídeo. Camadas adicionais, cada qual com prioridade decrescente, podem ser codificadas para refinar a qualidade da camada base. Conceitualmente, cada camada é transmitida sobre diferentes árvores multipontos, identificadas por diferentes endereços. Deste modo, cada receptor pode se associar a um ou mais grupos multipontos de acordo com a disponibilidade de banda passante no caminho até a fonte.

A figura 1.3 representa o funcionamento da codificação hierárquica. As camadas geradas são combinadas para produzir um refinamento progressivo. As imagens apresentadas possuem resoluções diferentes contudo, o refinamento pode estar relacionado à taxa de quadros por segundo ou à relação sinal-ruído.

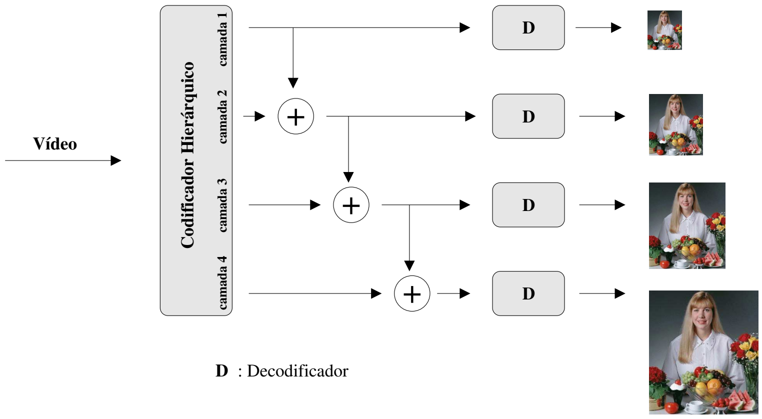


Figura 1.3: Codificação hierárquica.

Para a distribuição multiponto do vídeo em camadas, várias propostas encontram-se na literatura [8, 9, 10, 6, 11]. O serviço RLM (*Receiver-driven Layered Multicast*) [10] busca alcançar sempre o melhor nível de qualidade para os receptores contudo, eles não podem escolher uma qualidade arbitrária. Periodicamente, tenta-se aumentar a qualidade do vídeo recebido. Para este fim, é realizado um "experimento de associação" ao grupo multiponto correspondente à camada imediatamente superior a que o receptor estava recebendo e monitora-se os pacotes que chegam. Se o número de pacotes perdidos ultrapassar um certo limiar, conclui-se que não há banda passante disponível para esta nova camada e o receptor sai do grupo associado. Isto significa que este experimento pode causar congestionamentos na rede e, conseqüentemente, perda de pacotes destinados aos vários receptores. Neste caso, um receptor interferirá na qualidade do vídeo recebido por outros receptores. A realização de um experimento de associação é difundida para todos os receptores que ao receberem uma notificação de que um experimento está sendo realizado, ignoram as perdas temporariamente para tentar minimizar falsas conclusões sobre a disponibilidade de banda passante na rede. A difusão da realização de um experimento de associação é uma forma de cooperação entre os receptores.

O *ThinStreams* proposto por Wu et al. [9] utiliza um sinal de vídeo com alta taxa, dividido em vários fluxos com taxas idênticas. O objetivo é oferecer diferentes níveis de qualidade e diminuir a perda de pacotes relacionada a um experimento de associação. Tal diminuição pode ocorrer efetivamente, uma vez que, havendo muitos fluxos, a diferença entre as taxas de dois fluxos consecutivos será pequena, reduzindo o impacto causado pela introdução na rede de um tráfego que não pode ser suportado. A distribuição dos fluxos utiliza uma árvore multiponto para cada fluxo.

O LVMR (*Layered Video Multicast with Retransmissions*) proposto por Li et al. [8] utiliza os quadros I (*intraframe*), P (*predictive*) e B (*bidirectionally-predictive*) de um fluxo MPEG-2 (*Motion Picture Experts Group - 2*) para prover um serviço de distribuição de vídeo em camadas. A camada base é formada por quadros do tipo I, a primeira camada adicional é formada por quadros do tipo P e a segunda camada adicional é formada por quadros do tipo B. Nesta proposta, os pacotes perdidos por causa da realização de experimentos de associação são retransmitidos utilizando-se uma versão modificada do protocolo RMTP (*Reliable Multicast Transport Protocol*). Adicionalmente, é proposto um protocolo de controle que permite uma cooperação entre os receptores para a retransmissão de pacotes perdidos. Deste modo, um receptor pode retransmitir um pacote que não foi recebido por outro.

Em geral, as propostas existentes baseadas na distribuição do vídeo em camadas utilizam uma árvore multiponto para cada camada gerada pela fonte. Isto requer o estabelecimento de um número elevado de árvores multipontos, o que significa introduzir estados a serem mantidos pelos roteadores. Assim, mesmo que a fonte possa fornecer uma granulosidade mais elevada com o objetivo de prover uma Qualidade de Serviço (*Quality of Service* - QoS) melhor para cada receptor, este efeito limita o número de camadas que a rede pode suportar. Sendo assim, a escalabilidade está comprometida. Outro problema, é a dificuldade de se detectar a disponibilidade de banda passante, uma vez que para isto, o único mecanismo proporcionado pela a Internet, é a perda.

1.3 Objetivos e Organização do Trabalho

Recentemente, um novo modelo de redes, denominado redes ativas, foi proposto [12, 13, 14, 15]. Neste novo paradigma, os nós da rede podem executar computações específicas a cada pacote ao invés de simplesmente encaminhá-los. Os pacotes, denominados cápsulas ou pacotes ativos no contexto de redes ativas, podem conter códigos a serem executados pelos nós e desta forma, modificam e estendem as funcionalidades da rede. Assim, a rede adquire uma certa "inteligência" garantindo-lhe uma enorme flexibilidade uma vez que pode adequar-se dinamicamente às especificidades das aplicações [16, 17, 18]. Neste sentido, as redes ativas representam uma nova forma de se visualizar um ambiente de rede, proporcionando uma arquitetura aberta capaz não somente de permitir a alteração de seu comportamento funcional, como também de prover QoS para as aplicações [19]. Além das computações específicas, existe ainda a possibilidade do armazenamento de estados não-transientes (*soft-states*) nos nós ativos. Estes estados precisam ser renovados periodicamente para permanecerem ativos.

Os serviços de distribuição de vídeo multiponto que vêm sendo propostos para a Internet não tiram proveito da possibilidade da realização de computações específicas sobre pacotes individuais e nem da possibilidade de armazenamento não-transiente nos nós da rede. Tais recursos são artifícios poderosos que permitem o desenvolvimento de novos serviços que atendam sob demanda, de forma eficiente e segura, os requisitos de cada usuário.

Com base nestas potencialidades, este trabalho propõe um serviço de distribuição de vídeo multiponto baseado em redes ativas. O intuito é alcançar um serviço flexível e interativo, permitindo que os usuários do serviço sejam capazes de informar a qualidade que desejam para o vídeo e que o serviço se adapte ao pedido realizado. Sendo assim, o serviço ativo proposto provê a distribuição de vídeo multiponto orientada ao receptor. A qualquer instante, um receptor pode se conectar/desconectar do serviço e selecionar/alterar a qualidade do vídeo que deseja receber, sendo que a qualidade está limitada apenas à capacidade computacional do receptor ou à banda passante disponível no caminho individual entre a fonte e este receptor. A programabilidade dos nós da rede é explorada para superar limitações associadas aos serviços baseados em redes passivas.

Este trabalho está organizado da seguinte forma. No capítulo 2 são apresentados os conceitos de redes ativas e o estado da arte nesta área, incluindo as principais infra-estruturas propostas e suas características. O serviço ativo proposto de distribuição de vídeo multiponto, incluindo o protocolo e os algoritmos que implementam este serviço são apresentados no capítulo 3. A infra-estrutura de simulação, os resultados referentes ao funcionamento do serviço proposto e os resultados relativos ao comportamento e ao desempenho de um dos mecanismos utilizados pelo serviço são apresentados no capítulo 4. O capítulo 5 apresenta as conclusões.

Capítulo 2

As Redes Ativas

O PARADIGMA de redes ativas permite que o serviço oferecido pela rede seja programável, uma vez que os pacotes podem transportar códigos executáveis que estendam e modifiquem a infra-estrutura de rede. O objetivo do desenvolvimento de redes ativas é aumentar a flexibilidade da rede de modo que esta acomode rapidamente os requisitos específicos de QoS das novas aplicações e, também, das aplicações que estão em constante evolução.

A fim de obter um serviço de rede programável, vários grupos de pesquisa vêm propondo arquiteturas para as redes ativas, cada qual, explorando diferentes aspectos de sua programabilidade. Em algumas propostas, é comum a utilização dos paradigmas de agentes móveis e de redes programáveis. Estes paradigmas estão relacionados ao paradigma de redes ativas porém, existem várias opiniões sobre como estão relacionados.

A primeira seção deste capítulo descreve o ponto de vista particular do autor deste trabalho sobre o relacionamento entre estes três paradigmas. Na seção seguinte é apresentada uma visão geral da arquitetura de redes ativas que está sendo desenvolvida no programa de redes ativas do DARPA (*Department of Advanced Research Projects*) e que mostra vários pontos em comum entre as diversas propostas. Por último, são apresentados os ambientes de execução dos principais projetos na área de redes ativas.

2.1 Relação entre Paradigmas

A compreensão dos paradigmas de redes programáveis e agentes móveis é muito útil para o estudo das diferentes abordagens que visam oferecer infra-estruturas de rede programáveis. Estes paradigmas estão relacionados ao paradigma de redes ativas. As redes programáveis podem ser consideradas uma simplificação das redes ativas que por sua vez oferecem a base necessária à execução de agentes móveis.

2.1.1 Redes Ativas x Redes Programáveis

As redes programáveis e as redes ativas estão relacionadas porém são distintas. As primeiras propõem um controle aberto da rede através de interfaces de programação padronizadas que permitem às aplicações manipularem recursos de mais baixo nível na rede para a construção e o gerenciamento de serviços. Adicionalmente, o paradigma de redes ativas permite que, dinamicamente, pacotes ativos (cápsulas) contendo códigos de programação sejam executados em cada nó da rede. Sob este ponto de vista, as redes ativas são mais flexíveis do que as redes programáveis pois em um caso extremo, permitem que os serviços de rede sejam otimizados por pacote ao invés de serem otimizados por aplicação.

2.1.2 Redes Ativas x Agentes Móveis

Um agente móvel é um programa que atua como procurador de um usuário ou outro programa. Ele é capaz de se mover dentro da rede sob seu próprio controle. O agente decide quando e para onde migrar, podendo interromper sua própria execução ou continuá-la em qualquer outro lugar da rede. O paradigma de agentes móveis trata a rede como um conjunto de ambientes amigáveis aos agentes que por sua vez são tratados como entidades programadas para realizarem tarefas para os usuários [20]. Contudo, a idéia por de traz das redes ativas é muito mais geral. Nas redes ativas, a rede é vista como uma coleção de nós ativos capazes de realizar qualquer tipo de computação e como uma coleção de pacotes ativos que transportam programas. Sob este ponto de vista, um agente móvel pode ser considerado um tipo específico de pacote ativo, assim como um nó compatível com a tecnologia de agentes móveis pode ser considerado um tipo específico de nó ativo. A diferença fundamental entre o conceito de redes ativas e agentes móveis está no fato de no primeiro, os pacotes ativos serem executados na camada de rede, enquanto no segundo, os agentes móveis são executados como programas na camada aplicação. Sendo assim, uma rede ativa, devido a sua natureza programável, oferece funcionalidades básicas à execução dos agentes móveis.

2.2 Arquitetura Geral de Redes Ativas

Grupos de pesquisa vêm desenvolvendo uma arquitetura que descreve os componentes funcionais e a interface destes componentes para as redes ativas. Os objetivos do desenvolvimento desta arquitetura incluem a minimização de padronizações globais, a escalabilidade global, o suporte ao gerenciamento da rede, o suporte ao envio de pacotes IPv4/IPv6 (*Internet Protocol version 4/Internet Protocol version 6*) pelos nós ativos e o suporte a diferentes níveis de qualidades ou classes de serviço. A função primária da rede é a comunicação fim-a-fim e não serviços gerais de computação.

De modo a alcançar um serviço de rede programável, a abordagem empregada tem sido a especificação da arquitetura de um nó ativo. Esta arquitetura define as funcionalidades básicas que são comuns a todos os nós. Tais funcionalidades incluem a definição de como os pacotes são processados, dos recursos disponíveis no nó e da forma de acesso a tais recursos. Assim, esta arquitetura define a funcionalidade básica da interface de programação do nó ativo, embora não especifique nenhuma linguagem em particular ou codificação a ser utilizada.

A funcionalidade de cada nó de uma rede ativa é dividida entre os ambientes de execução (AE), o sistema operacional e as aplicações ativas. A organização geral destes componentes é apresentada na figura 2.1. Cada AE exporta uma API (*Application Programming Interface*) ou uma máquina virtual para uso das aplicações ativas, disponibilizando o acesso à diversas facilidades dos nós.

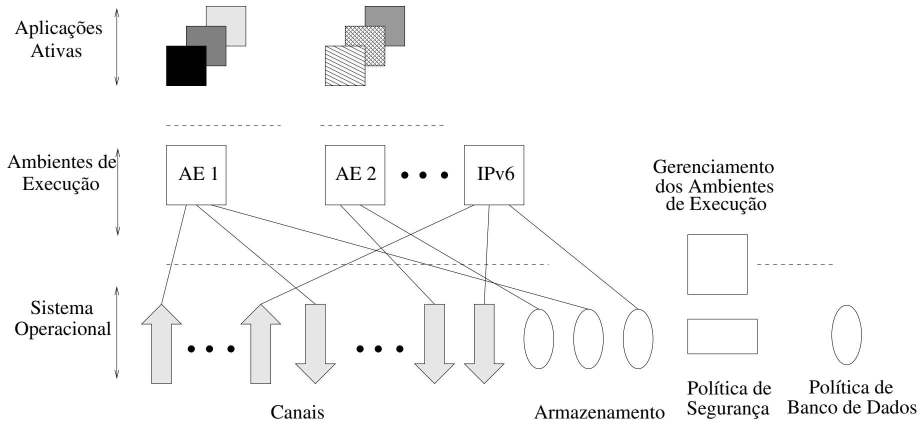


Figura 2.1: Arquitetura de um nó ativo.

O sistema operacional do nó provê funcionalidades básicas através das quais os ambientes de execução constróem as abstrações apresentadas às aplicações ativas. Entre tais abstrações, existe o fluxo de pacotes. O fluxo de pacotes é a unidade básica utilizada pelo sistema operacional para alocar e gerenciar recursos, os quais incluem banda passante, ciclos de UCP (Unidade Central de Processamento) e armazenamento não-transiente. Ao gerenciar tais recursos, o sistema operacional isola os ambientes de execução de detalhes de gerenciamento e do comportamento de outros ambientes de execução.

A arquitetura do nó é projetada de modo a suportar várias APIs de rede simultaneamente. O IPv4 e o IPv6 podem ser exemplos de APIs. Uma representação do fluxo de pacotes em um nó ativo está apresentada na figura 2.2. Ao chegar no nó, o pacote é classificado com base nas informações contidas em seu cabeçalho, determinando-se os protocolos relacionados ao seu processamento e o respectivo ambiente de execução. O protocolo ANEP (*Active Network Encapsulation Protocol*) provê o meio pelo qual os usuários controlam o roteamento dos pacotes a determinados ambientes de execução.

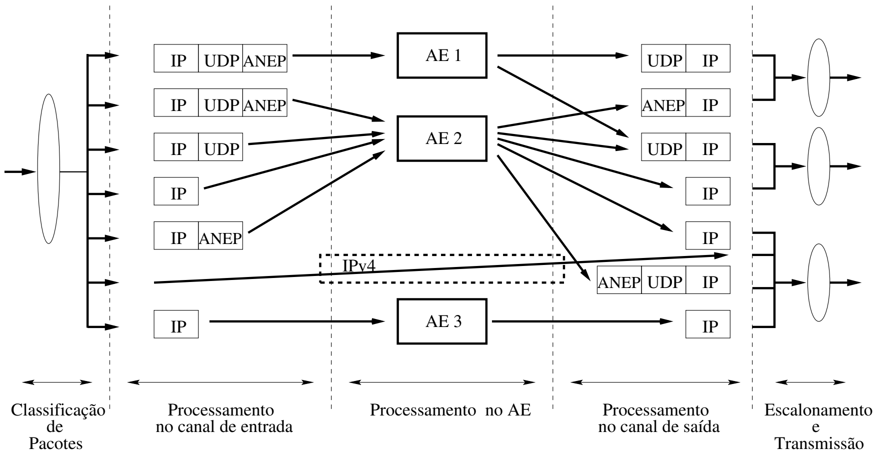


Figura 2.2: Fluxo de pacotes através de um nó ativo.

2.3 Modelos de Ambientes de Execução

Os ambientes de execução apresentados nas arquiteturas propostas para redes ativas possuem características distintas, uma vez que, um serviço de rede programável pode ser explorado de diversas maneiras. Deste modo, a flexibilidade alcançada dependerá de vários atributos que incluem [14]:

- o poder de expressividade da linguagem utilizada, isto é, o grau de programabilidade que a API de rede pode oferecer. Este grau pode variar desde um simples conjunto de escolhas pré-definidas até qualquer tipo de computação realizável que uma linguagem de programação mais abrangente é capaz de descrever;
- a capacidade de criar e referenciar estados nos nós intermediários da rede;
- a granulosidade de controle, isto é, como as alterações realizadas no comportamento funcional de um nó são vistas pelos pacotes. Por um lado, um pacote recebido pode modificar o comportamento funcional do nó visto por todos os pacotes. Em um outro extremo, o pacote recebido modifica o comportamento visto somente por ele. Entre estes dois extremos, as modificações podem ser vistas somente por pacotes de um mesmo fluxo.

2.3.1 ANTS

O ANTS (*Active Node Transfer System*) [21, 22, 18, 23], desenvolvido no MIT, é uma ferramenta baseada na linguagem Java que funciona como uma plataforma de testes para redes ativas. No ANTS, os pacotes convencionais que transportam dados são substituídos por cápsulas que contêm, além dos dados, uma descrição do procedimento a ser executado na rede. Esta rede é vista como um sistema de programação distribuído e sua API consiste em uma máquina virtual Java mais uma classe ANTS, a qual implementa métodos que permitem a decodificação e a interpretação das cápsulas. Neste sistema, um protocolo é composto por um conjunto de procedimentos transportados por tipos de cápsulas relacionadas. Cada cápsula possui um identificador para o seu tipo e o protocolo ao qual está relacionada. Quando uma cápsula chega em um nó, verifica-se se o protocolo necessário a sua execução está armazenado. Caso não esteja, os procedimentos necessários são requisitados ao nó anterior e transferidos através de um mecanismo de distribuição de códigos. Deste modo, os procedimentos de um protocolo são difundidos no caminho percorrido pelas cápsulas associadas a este protocolo.

Para a programação dos procedimentos, é disponibilizado um conjunto de primitivas as quais permitem a manipulação e o controle das cápsulas, e o acesso aos recursos disponíveis nos nós. Dentre estas primitivas encontram-se as de acesso às tabelas de roteamento, as de envio, cópia e descarte utilizadas pelas próprias cápsulas, e primitivas de manipulação do *soft-state*. O *soft-state* consiste em um espaço de memória não-transiente, ou seja, que persiste à execução de uma cápsula, onde são armazenadas as informações compartilhadas por um fluxo. O *soft-state* persiste por um período de tempo determinado, denominado tempo de expiração. Este tempo precisa ser renovado periodicamente enquanto houver necessidade da utilização deste recurso. Para limitar a quantidade de recursos que uma cápsula pode alocar na rede, cada uma delas possui em seu cabeçalho um campo que funciona como uma espécie de TTL (*Time-To-Live*). O valor deste campo é decrementado pelos nós a medida que recursos como banda passante, tempo de processamento e memória são consumidos. Quando o valor deste campo é zero, a cápsula é descartada.

2.3.2 *Smart Packets*

Como parte do programa de redes ativas do DARPA, a BBN está desenvolvendo o projeto *Smart Packets*. Este projeto visa aplicar o paradigma de redes ativas no gerenciamento de redes. Assim, os *smart packets*, contendo os procedimentos a serem executados, são encapsulados dentro de pacotes ANEP, que por sua vez são encapsulados em datagramas IP (*Internet Protocol*). Diferentemente de outras abordagens, os nós ativos não permitem a criação de *soft-states*. Deste modo, os procedimentos para a execução dos datagramas não podem ser armazenados no nó, persistindo somente enquanto durar sua execução. Para a programação dos procedimentos, utiliza-se uma linguagem denominada *Sprocket* que é compilada para uma linguagem de representação compacta e similar a uma linguagem *assembly* denominada *Spanner*. Assim, a API de rede consiste em uma máquina virtual *Spanner* implementada por um *deamon* executando em um nó ativo. A utilização da linguagem *Spanner* permite alcançar um dos objetivos do projeto que é desenvolver programas inferiores a 1k octeto. A máquina virtual *Spanner* inclui o suporte para o acesso de variáveis de gerenciamento de rede em MIBs (*Management Information Bases*), não permitindo acesso a outras chamadas do sistema ou à memória fora do escopo de execução do datagrama. Mecanismos adicionais para a autenticação dos datagramas e controle sobre a execução deles também são providos [14, 24].

2.3.3 *Liquid Software*

O projeto *Liquid Software* [25] está sendo desenvolvido por pesquisadores da universidade do Arizona. O *liquid software* é definido como sendo um código móvel, orientado à comunicação e de baixo nível que pode fluir facilmente entre os nós da rede. Como o foco principal é a transferência eficiente de dados, requisitos de qualidade de serviço e restrições de tempo real associadas à comunicação devem ser considerados pelo sistema operacional e pelo ambiente de execução. De modo a suportar tais demandas, foi desenvolvido um sistema operacional configurável através da composição de módulos, denominado *Scout*. Este sistema operacional consiste em uma máquina virtual Java modificada que permite a alocação e o gerenciamento de recursos como ciclos de UCP, memória e banda passante. Em conjunto com um compilador *just-in-time*, este sistema operacional compõe o ambiente de execução, denominado *Joust* o qual exporta a API Java para a programação dos serviços de rede. Novos serviços são adicionados como módulos do sistema operacional.

2.3.4 *SwitchWare*

O projeto *SwitchWare* em desenvolvimento na universidade da Pennsylvania propõe uma arquitetura de rede que visa alcançar a melhor relação custo-benefício entre os requisitos de segurança, que são inerentes a uma infra-estrutura compartilhada, e a flexibilidade de uma rede programável [26, 27]. Esta arquitetura é dividida em três níveis: pacotes ativos, extensões ativas, denominadas *switchlets* e infra-estrutura do roteador ativo.

Os pacotes ativos são os substitutos dos pacotes das redes passivas. Eles contêm código de programação e dados do usuário. O código de programação provê a mesma funcionalidade do cabeçalho dos pacotes das redes passivas contudo, de modo mais flexível, uma vez que os pacotes podem interagir de forma mais complexa e específica às suas necessidades. A programação dos serviços é realizada através de uma linguagem, denominada PLAN (*Programming Language for Active Networks*), bastante simples e que provê somente um conjunto mínimo de funcionalidades, reduzindo em vários casos a necessidade de verificação do código nos nós ativos [28]. Esta linguagem não permite o armazenamento de estados nos nós da rede. Desta forma, as ações de um programa são fortemente restringidas. Para compensar esta limitação, os programas utilizam as extensões ativas *switchlets* que podem implementar um conjunto arbitrário de protocolos e funcionalidades.

O nível mais baixo da arquitetura é a infra-estrutura do roteador ativo. Este nível é a base de segurança sobre a qual os pacotes ativos e os *switchlets* operam, permitindo a alocação de recursos e fazendo as verificações necessárias à inserção de novos *switchlets* no nó ativo.

2.3.5 *Netscript*

O projeto *Netscript* [29] está sendo desenvolvido na universidade de Columbia. Este projeto visa a definição de um conjunto mínimo de funcionalidades em um nó que permitam o desenvolvimento de uma ampla gama de serviços. Para isto, o *Netscript* usa como base o paradigma de agentes. Os agentes são utilizados para programar e controlar as funções dos nós da rede. Quando pacotes de um fluxo chegam a um nó, eles são processados pelos agentes apropriados para cumprir as funcionalidades específicas dos protocolos.

No *Netscript*, a rede é vista como uma coleção de máquinas de rede virtuais (*Virtual Network Engines* - VNEs) interconectadas por enlaces virtuais (*Virtual Links* - VLs). Os VNEs podem ser programados pelos agentes para processar pacotes de um fluxo e redirecioná-los para determinados VLs. Uma coleção de VNEs e VLs definem uma rede virtual *Netscript* (*Nestscript Virtual Network*- NVN). Um nó físico pode executar vários VNEs e um VL pode corresponder a um conjunto de enlaces físicos e nós que interconectam VNEs. Estas abstrações escondem detalhes relacionados aos elementos físicos da rede e simplificam o desenvolvimento dos serviços.

2.3.6 *Protocol Boosters*

O projeto *Protocol Boosters* [16, 17, 30] está sendo desenvolvido conjuntamente pelo Bellcore e pela universidade da Pennsylvania. *Protocol Boosters* representam uma metodologia para o desenvolvimento de protocolos. Eles visam evitar o processo lento de evolução e ineficiências relacionadas aos protocolos de propósito geral. Os protocolos são construídos dinamicamente, de forma independente de padronizações, a partir de um protocolo base ao qual são adicionadas novas funcionalidades de acordo com as necessidades de uma dada aplicação. Tais funcionalidades são adicionadas através de elementos denominados *boosters*, os quais são projetados de forma a melhorarem o desempenho ou características de um protocolo existente. Um elemento *booster*, por si só, não é um protocolo. Um novo protocolo é formado pela combinação das mensagens de seu respectivo protocolo base com as mensagens trocadas entre os seus *boosters*. Novos *boosters* podem ser aplicados sucessivamente ao novo protocolo criado. Um exemplo de aplicação é a utilização de *boosters* de compactação de pacotes, seguida de correção de erros (*Forward Error Correction* - FEC), entre dois pontos específicos da rede onde a taxa de erros na transmissão de pacotes seja elevada.

A plataforma de testes desenvolvida neste projeto permite somente alterações no comportamento funcional dos nós das extremidades de comunicação contudo, as metodologias propostas podem ser empregadas em uma infra-estrutura de rede totalmente programável. Os nós das extremidades executavam uma versão modificada do sistema operacional *Linux* com suporte à construção dinâmica de protocolos.

Capítulo 3

O Serviço Ativo de Distribuição de Vídeo Multiponto

A PROGRAMABILIDADE dos serviços de rede pode ser explorada sobre diferentes aspectos. A possibilidade da realização de computações específicas associada à habilidade de armazenar e acessar estados em um nó compõem um recurso extremamente importante que permite às aplicações atenderem sob demanda, de forma segura e flexível, aos diferentes requisitos de QoS de cada usuário. Deste modo, o serviço proposto explora a programabilidade da rede para superar algumas limitações dos serviços de distribuição de vídeo baseados no paradigma de redes passivas, e ainda, permitir sua adequação ao pedido de qualidade desejada por cada receptor.

A figura 3.1 ilustra o serviço de distribuição proposto. Nela, uma fonte de vídeo gera quatro camadas, as três primeiras a uma taxa de 1 Mbps e a última a uma taxa de 7 Mbps. A banda passante de cada enlace e a qualidade desejada do vídeo por cada receptor estão representadas nessa figura. Todos os receptores, exceto o quarto, recebem um número de camadas adequado à disponibilidade de banda passante na rede. O quarto receptor, apesar de poder receber uma camada a mais de vídeo, só está recebendo aquilo que deseja, ou seja, a camada base e uma camada adicional que compõem a qualidade desejada de 2 Mbps. Para tentar receber uma qualidade diferente da atual, o receptor deve se utilizar da escala de ajuste de qualidade, conforme sua própria conveniência.

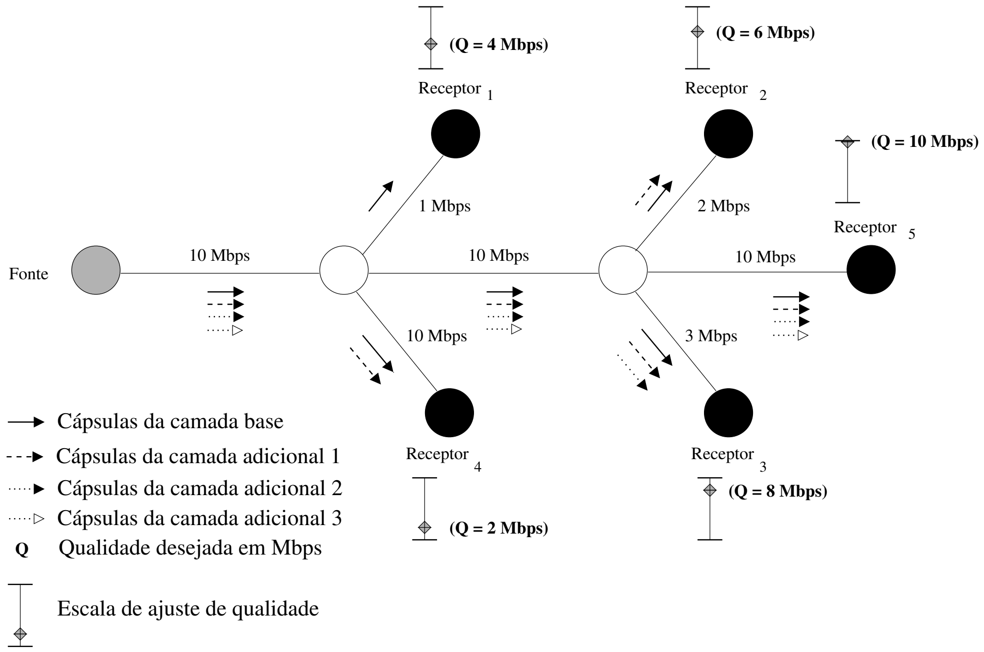


Figura 3.1: Exemplo da distribuição de vídeo.

O quinto receptor está recebendo a máxima qualidade possível (10 Mbps). Supondo que este receptor se desconecte do serviço, a fonte passará a gerar somente três camadas. Sendo assim, ela estará gerando o número de camadas necessário para melhor atender os receptores e não desperdiçar banda passante na rede.

Na seção seguinte é apresentada a infra-estrutura necessária para suportar o serviço ativo proposto. A seção 3.2 descreve o protocolo que implementa o serviço.

3.1 A Infra-estrutura Necessária

Uma infra-estrutura de redes ativas baseada no modelo de ambiente de execução do ANTS foi assumida para suportar o serviço proposto. Sendo assim, os nós da rede suportam a execução de procedimentos específicos para as aplicações e permitem o armazenamento de estados. As cápsulas utilizadas no serviço estão associadas a procedimentos que são executados a cada nó ativo e se utilizam de várias funcionalidades disponíveis nestes nós. As funcionalidades básicas necessárias para o serviço proposto são:

- Armazenamento não-transiente (*soft-state*) de cápsulas de um mesmo fluxo. Em cada nó ativo, o *soft-state* persiste por um tempo determinado que pode ser reiniciado. O período de expiração considerado é o mesmo para todos os nós ativos;

- Monitoração da banda passante consumida por um determinado fluxo e monitoração da banda passante agregada sendo consumida pelos diversos fluxos que transitam por um enlace;

- Métodos ou procedimentos básicos de envio, descarte, criação e replicação a serem utilizados pelas cápsulas.

- Fusão de cápsulas. A fusão consiste em se agregar em uma única cápsula informações transportadas por várias cápsulas que chegam a um nó. Desta forma, essas várias cápsulas podem ser suprimidas.

O serviço também requer uma fonte de vídeo em camadas que suporte o controle dinâmico de QoS. Deste modo, baseada em informações contidas nas cápsulas recebidas, ela será capaz de adicionar e remover camadas para atender os requisitos dos diferentes receptores conectados ao serviço.

3.2 O Protocolo Proposto

O protocolo proposto para este serviço ativo de distribuição de vídeo multiponto é composto por cinco tipos de cápsulas: cápsula de inscrição/manutenção, cápsula de disponibilidade, cápsula de dados de vídeo, cápsula de fim de sessão e cápsula de desligamento.

3.2.1 A Cápsula de Inscrição/Manutenção

A função da cápsula de inscrição/manutenção é a formação e a manutenção da árvore multiponto. Este tipo de cápsula é utilizada por todos os receptores que desejam receber dados do grupo multiponto. Para se associar ao grupo, o receptor envia uma cápsula de inscrição/manutenção em direção à fonte. Ao percorrer este caminho, a cápsula cria, para o respectivo grupo multiponto, um *soft-state* nos nós ativos e instala um apontador para a interface de chegada. O objetivo é marcar o caminho por onde as cápsulas de dados de vídeo devem passar para alcançarem os receptores. Esta situação está ilustrada na figura 3.2.

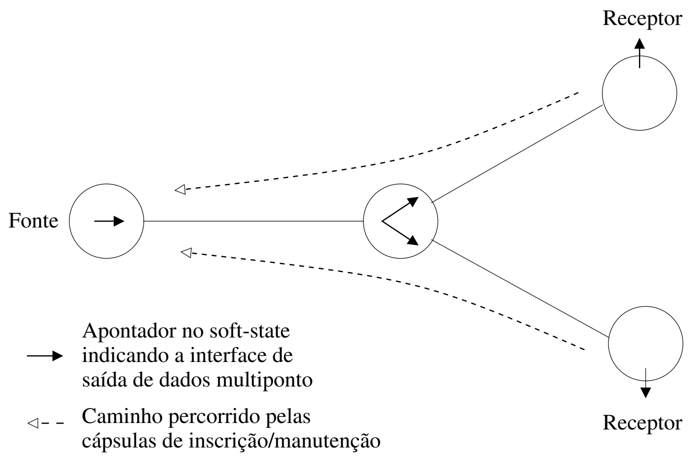


Figura 3.2: Formação da árvore multiponto.

Ao percorrer o caminho em direção à fonte, se já houver *soft-state* no nó ativo atualmente visitado, a cápsula é responsável pela manutenção desse *soft-state*. Ao verificar que o *soft-state* relacionado ao grupo multiponto ao qual pertence está próximo de expirar, ela reinicia a estampa de tempo associada a ele. Esta estampa de tempo é utilizada pelo nó ativo para verificar a existência de *soft-states* expirados e removê-los.

Além de haver uma estampa de tempo para o *soft-state* como um todo, para cada apontador instalado também existe uma estampa de tempo associada. Ao chegar por uma interface, a cápsula de inscrição/manutenção deve reiniciar a estampa de tempo associada ao apontador para esta interface. Essa estampa de tempo é utilizada por essas cápsulas para verificar a necessidade da remoção de apontadores expirados. A existência de um apontador expirado indica que um caminho não deve mais ser percorrido por cápsulas de dados de vídeo. Sendo assim, elimina-se um trecho da árvore multiponto que estava sendo desnecessariamente utilizado.

Para realizar a manutenção freqüente da árvore multiponto, todos os receptores devem enviar periodicamente cápsulas de inscrição/manutenção em direção à fonte. Contudo, nem todas as cápsulas precisam percorrer o caminho completo até a fonte. Se parte do caminho for compartilhada por dois ou mais receptores, apenas uma cápsula de inscrição/manutenção precisa percorrer o caminho até a fonte antes da expiração do *soft-state*. As cápsulas restantes se auto-descartam.

Para tomarem essa decisão, elas verificam se a estampa de tempo do *soft-state* foi reiniciada recentemente, de modo a garantir que a próxima reiniciação ocorra antes da expiração do *soft-state*. Este mecanismo de descarte reduz ao mínimo o consumo de banda passante para a manutenção da árvore multiponto. O método de formação e manutenção da árvore multiponto apresentado é similar ao utilizado no ANTS [21].

Duas características importantes desse método são a robustez quanto a mudanças de topologia e a distinção entre fonte e receptores. Para uma árvore formada, existe apenas uma fonte. Se várias fontes forem necessárias, várias árvores podem ser formadas, ou a fonte pode ser considerada a raiz de uma árvore CBT (*Core-based tree*) [31], com o roteamento de cápsulas de dados de vídeo sendo realizado para a raiz e em seguida, para os receptores.

3.2.2 A Cápsula de Disponibilidade

Para decisões de roteamento das cápsulas de dados de vídeo, é construída no *soft-state* de cada nó ativo uma tabela de disponibilidade. Esta tabela contém, para cada interface de saída marcada como pertencente ao grupo multiponto, a informação da taxa máxima permitida para o vídeo. Também é construída no *soft-state* de cada nó, exceto nos receptores, uma tabela auxiliar associada à tabela de disponibilidade. Esta tabela contém, sob o ponto de vista do nó, a qualidade do vídeo desejada pelos nós vizinhos pertencentes à sub-árvore com raiz nesse nó. A tabela auxiliar também contém uma referência para a interface que permite alcançar cada um desses vizinhos. As tabelas de disponibilidade e suas respectivas tabelas auxiliares são construídas pela cápsula de disponibilidade enviada pelo receptor, logo após o envio da primeira cápsula de inscrição/manutenção.

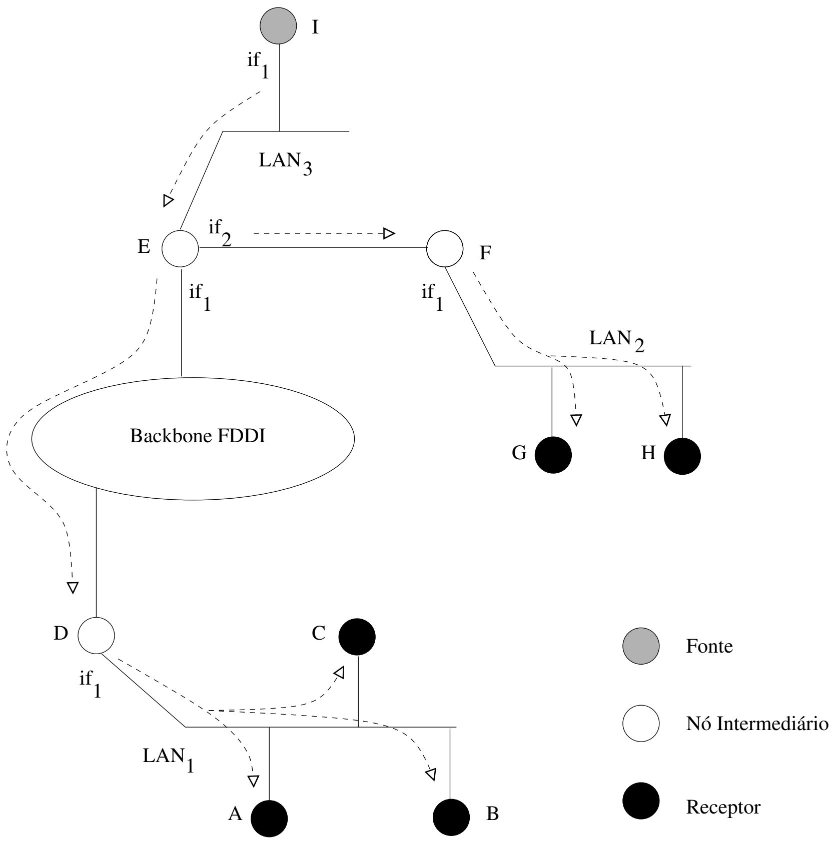


Figura 3.3: Exemplo comum de distribuição de vídeo entre sub-redes.

Para a análise das próximas figuras que ilustrarão a construção dessas tabelas, considere primeiramente a figura 3.3. Esta figura ilustra um exemplo comum de distribuição de vídeo multiponto entre sub-redes. Nela estão representadas três redes locais (LAN_1, LAN_2 e LAN_3) e um *backbone* FDDI (*Fiber Distributed Data Interface*). A fonte se encontra na rede local LAN_3 e os receptores nas redes locais LAN_1 e LAN_2. As letras de A a I representam identificadores para os nós. O nó com identificador E possui duas interfaces de saída para o vídeo e os nós cujos respectivos identificadores são D, F e I possuem apenas uma. A figura 3.4 é uma versão simplificada da figura 3.3.

A figura 3.5 mostra como as tabelas auxiliares e de disponibilidade são construídas com a entrada do primeiro receptor (receptor A) no grupo multiponto. A numeração dos nós corresponde ao ponto de vista do receptor. A banda passante

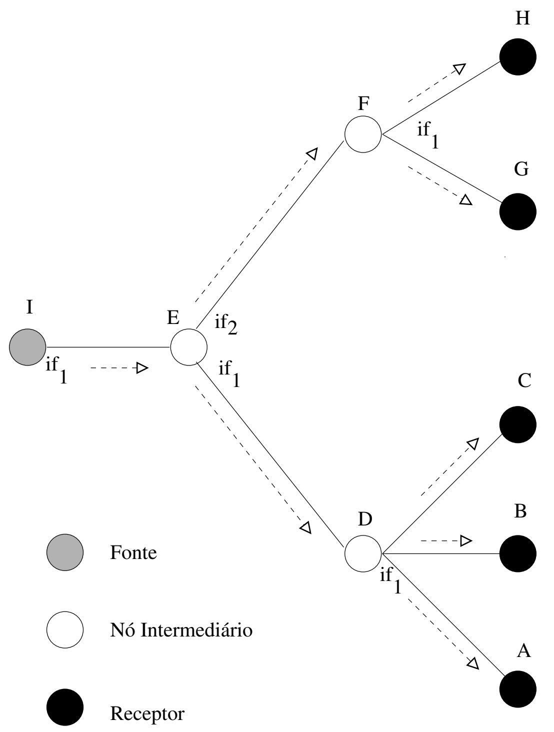


Figura 3.4: Versão simplificada da figura 3.3.

disponível em cada enlace para o vídeo está indicada sobre seu respectivo enlace. Depois da entrada do receptor no grupo multiponto, este envia uma cápsula de disponibilidade trazendo no campo TX de seu cabeçalho a informação da qualidade Q, em termos de banda passante, que ele deseja receber. Este valor também é colocado na tabela de disponibilidade construída no receptor. Ao chegar no nó n_1, essa cápsula cria uma tabela auxiliar contendo uma entrada que informa a interface por onde chegou, a identificação do nó anteriormente visitado e o valor contido no seu campo TX. Em seguida, ela verifica a disponibilidade de banda passante para vídeo no enlace que permite alcançar o último nó visitado. A cápsula então marca na tabela de disponibilidade do nó em questão o menor valor entre a banda passante disponível e o valor da qualidade na tabela auxiliar, seguindo para o próximo nó com esta informação no campo TX. Aplica-se o mesmo algoritmo ao se passar pelos nós n_2 e n_3. Ao receber a cápsula de disponibilidade, a fonte de vídeo utiliza a informação do campo TX para saber exatamente quantas camadas deve gerar de modo a atender da melhor forma possível o receptor.

A figura 3.6 mostra a entrada do segundo receptor (receptor B) no grupo multi-

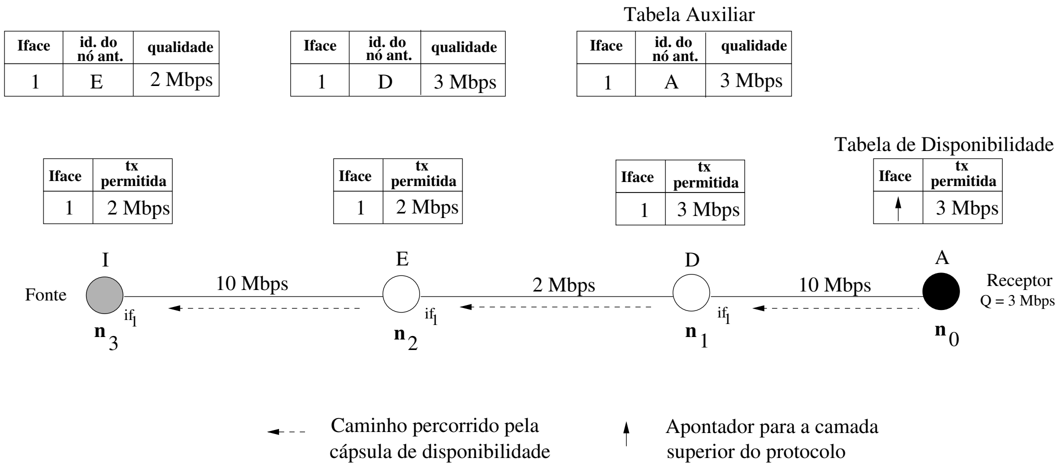


Figura 3.5: Construção das tabelas auxiliares e de disponibilidade. Entrada do primeiro receptor.

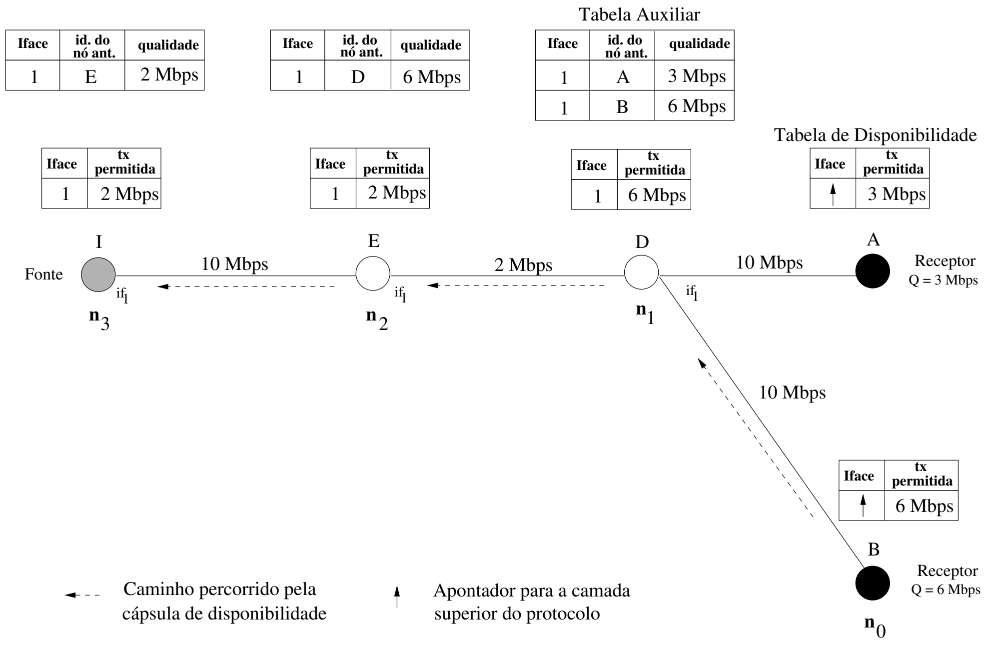


Figura 3.6: Construção e atualização das tabelas auxiliares e de disponibilidade. Entrada do segundo receptor no grupo multiponto.

ponto. Os receptores A e B estão conectados à mesma interface do nó identificado por D. A numeração dos nós apresentada corresponde ao ponto de vista do receptor B. A construção da tabela de disponibilidade desse receptor segue o mesmo algoritmo descrito anteriormente. Após aplicar este algoritmo, a cápsula de disponibilidade chega ao nó n_1 trazendo em TX a qualidade desejada por este receptor. Neste nó, é adicionada uma nova entrada na tabela auxiliar contendo a tripla $\{1,\ B,\ 6\ Mbps\}$, ou seja, a qualidade de 6 Mbps, indicada no campo TX da cápsula, desejada pelo receptor B, alcançado pela interface 1. Em seguida, a cápsula atualiza a entrada, na tabela de disponibilidade, respectiva a essa interface. O novo valor para a taxa permitida para o vídeo por esta interface é computado como sendo o menor valor entre a banda passante disponível para o vídeo no enlace que permite alcançar o receptor B e o maior valor entre todas as qualidades relacionadas à interface 1 na tabela auxiliar. O resultado dessa computação é posto no campo TX da cápsula que segue para o nó n_2. Neste nó, a entrada da tabela auxiliar é atualizada com o valor de TX. Em seguida, computa-se o menor valor entre a banda passante disponível para o vídeo e o valor contido no campo *qualidade* na tabela auxiliar. É feita a atualização na entrada da tabela de disponibilidade com este resultado e, posteriormente, a cápsula segue para o nó n_3 com esta informação em TX. O procedimento de atualização das tabelas nesse nó é o mesmo.

Sob o ponto de vista do nó cujo identificador é D, a tabela auxiliar contém a qualidade desejada por cada receptor do grupo multiponto. Sob o ponto de vista do nó cujo identificador é E, a tabela auxiliar contém a qualidade "desejada" pelo nó com identificador D. Sendo assim, de uma forma geral, a tabela auxiliar de um nó contém a qualidade desejada pelos nós vizinhos pertencentes à sub-árvore com raiz nesse nó.

A entrada do receptor G no grupo multiponto é apresentada na figura 3.7. O algoritmo aplicado para a construção das tabelas nos nós n_0 e n_1 segue os mesmos passos descritos anteriormente. O principal ponto a ser observado é com relação à inclusão da interface 2 nas tabelas do nó n_2. Ao chegar nesse nó, a cápsula de disponibilidade adiciona à tabela auxiliar uma entrada relativa à qualidade desejada pelo seu nó vizinho, cujo identificador é F, e que pode ser alcançado através da

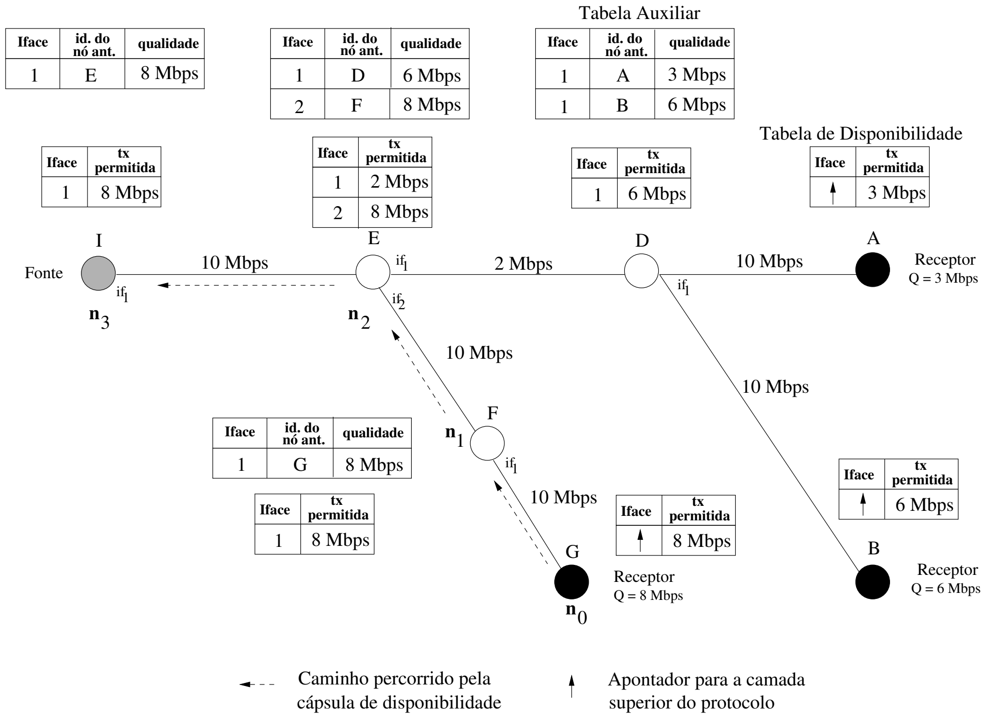


Figura 3.7: Construção e atualização das tabelas auxiliares e de disponibilidade. Entrada do terceiro receptor no grupo multiponto.

interface 2. A nova entrada na tabela de disponibilidade é computada como sendo o menor valor entre a banda passante disponível para o vídeo no enlace que permite alcançar o nó anterior visitado (F) e a entrada relacionada à interface 2 na tabela auxiliar. Ou seja, o valor computado é o mínimo entre 10 Mbps e 8 Mbps. As entradas referentes à interface 1 que porventura existam na tabela auxiliar não são consideradas. Com a nova entrada na tabela de disponibilidade, a informação a ser colocada no campo TX da cápsula de disponibilidade que seguirá para o nó n_3 é o maior valor observado nessa tabela. Neste caso, os dados referentes a todas as interfaces são considerados.

A figura 3.8 resume o relacionamento entre as entradas na tabela auxiliar, as entradas na tabela de disponibilidade e o campo TX da cápsula de disponibilidade a ser enviada para o próximo nó no caminho para a fonte. Cada entrada na tabela de disponibilidade é obtida a partir de um subconjunto de entradas na tabela auxiliar. A informação contida no campo TX da cápsula de disponibilidade é obtida a partir de todas as entradas contidas na tabela de disponibilidade.

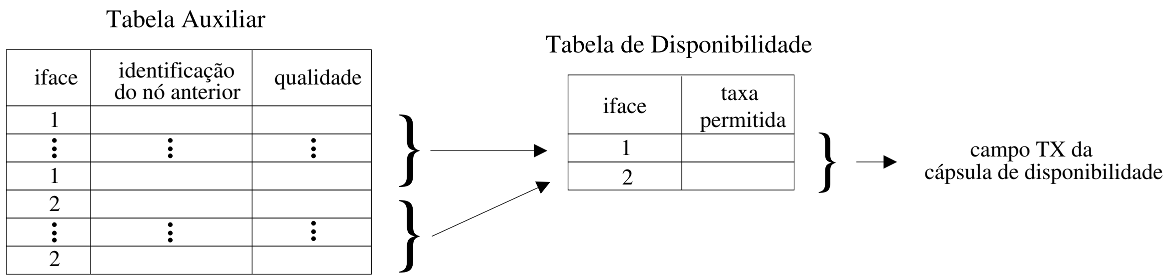


Figura 3.8: Resumo do relacionamento entre a tabela auxiliar, a tabela de disponibilidade e o campo TX da cápsula de disponibilidade a ser enviada para o próximo nó.

Sendo assim, de uma forma geral, o algoritmo aplicado para adicionar ou atualizar uma entrada na tabela auxiliar segue a equação. 3.1.

$$\Gamma^{(i)}_{n,n-1} = TX_{n-1} \text{ para } n \geq 1 \tag{3.1}$$

onde i é a interface por onde chegou a cápsula de disponibilidade, $\Gamma^{(i)}_{n,n-1}$ representa, sob o ponto de vista do nó n, a qualidade desejada pelo nó $n-1$, alcançado pela interface i. O valor do campo TX do cabeçalho da cápsula de disponibilidade ao sair do nó $n-1$ é representado por TX_{n-1}.

O algoritmo aplicado para adicionar ou atualizar uma entrada na tabela de disponibilidade segue a equação. 3.2.

$$tx^{(i)}_n = \begin{cases} \min\{\max(\Lambda^{1,(i)}_n, \ldots, \Lambda^{\sigma^{(i)},(i)}_n), b^{(i)}_n\} & \text{para } n \geq 1 \text{ e } \sigma^{(i)} \geq 1 \\ Q & \text{para } n = 0 \end{cases} \tag{3.2}$$

onde i é a interface por onde chegou a cápsula de disponibilidade, $b^{(i)}_n$ representa a banda passante disponível para o vídeo no enlace alcançado pela interface i do nó n, $tx^{(i)}_n$ representa a taxa máxima permitida para vídeo no enlace alcançado pela interface i do nó n quando $n \geq 1$ e representa a qualidade Q desejada pelo receptor quando $n = 0$. O número de entradas relativas à interface i na tabela auxiliar é representado por $\sigma^{(i)}$. Os valores $\Lambda_n^{1,(i)}$ a $\Lambda_n^{\sigma^{(i)},(i)}$ representam as entradas do campo *qualidade*, referentes à interface i, na tabela auxiliar do nó n.

O valor a ser colocado no campo TX da cápsula de disponibilidade que será enviada em direção à fonte é computado seguindo a equação. 3.3.

$$TX_n = \begin{cases} \max\left(\Omega_n^1, \ldots, \Omega_n^\theta\right) & \text{para } n \geq 1 \text{ e } \theta \geq 1 \\ Q & \text{para } n = 0 \end{cases} \tag{3.3}$$

onde, Q é a qualidade do vídeo, em termos de banda passante, desejada pelo receptor, TX_n representa o valor do campo TX do cabeçalho da cápsula de disponibilidade ao sair do nó n. O número de entradas na tabela de disponibilidade é representado por θ. Os valores Ω_n^1 a Ω_n^θ representam as entradas do campo *taxa permitida* na tabela de disponibilidade do nó n.

Estampas de Tempo na Tabela Auxiliar

Assim como as cápsulas de inscrição/manutenção mantêm uma estampa de tempo associada a cada interface de saída marcada como pertencente ao grupo multiponto, as cápsulas de disponibilidade mantêm para cada entrada na tabela auxiliar uma estampa de tempo associada. Essas estampas são utilizadas por essas cápsulas para verificarem a existência de entradas expiradas na tabela. Uma entrada expirada na tabela auxiliar de um nó indica que um nó vizinho pertencente à sub-árvore com raiz nesse nó não deve mais receber o vídeo e assim sendo, deve-se remover essa entrada da tabela para liberar recursos.

Cada estampa de tempo é reiniciada pela cápsula de disponibilidade provinda do nó associado a esta entrada. Ao atualizar ou inserir uma entrada na tabela auxiliar, a cápsula verifica para todas as entradas referentes à mesma interface por onde chegou se existem entradas expiradas. Caso existam, essas entradas são removidas.

Suponha, por exemplo, uma tabela auxiliar com apenas duas entradas, uma para a interface 1 e outra para a interface 2. Suponha ainda que cápsulas de disponibilidade e de inscrição/manutenção não estejam mais chegando pela interface 2. Portanto, por algum motivo, não devem mais existir receptores alcançados pela interface 2. As cápsulas de disponibilidade que chegam pela interface 1 não podem remover a entrada expirada, uma vez que esta interface não é a mesma interface por onde ela chegou ao nó. Contudo, se cápsulas de inscrição/manutenção não estão realizando a reiniciação da estampa de tempo associada ao apontador contido no *soft-state* para a interface 2, esta estampa de tempo expirará e será removida por uma cápsula de inscrição/manutenção provinda da interface 1. Por conseqüência, todas as entradas relativas à interface 2 nas tabelas de disponibilidade e auxiliar serão removidas. Esta forma de cooperação das cápsulas de inscrição/manutenção é um mecanismo bastante robusto pois mesmo na presença de falhas é capaz de liberar os recursos desperdiçados no nó ativo.

Os Recursos Necessários para as Tabelas

O número máximo de entradas na tabela de disponibilidade está associado à quantidade de interfaces que um nó possui. Este valor máximo é igual ao número de interfaces existentes menos um, uma vez que uma das interfaces é a interface que permite alcançar a fonte. Isto permite predizer a quantidade máxima de recursos a serem alocados no nó para a construção da tabela, independentemente do número de receptores pertencentes ao grupo multiponto. Contudo, o número máximo de entradas na tabela auxiliar de um nó depende do número de nós vizinhos pertencentes à sub-árvore com raiz nesse nó. Supondo um *backbone* FDDI, cobrindo uma distância de 200 km, o número máximo de nós que podem estar conectados a ele é 1000. Sendo assim, é possível que um nó possua 1000 vizinhos. Este valor é o número máximo de entradas na tabela auxiliar. Supondo ainda que cada entrada ocupasse 1 octeto para referenciar a interface, 4 octetos para o identificador do nó, 4 octetos para a qualidade e mais 4 octetos para a estampa de tempo, a quantidade de recursos a serem alocados para uma tabela com 1000 entradas seria de 13k octetos. Se tivéssemos 1000 fluxos, cada qual utilizando sua própria tabela, os recursos alocados seriam de 13M octetos.

O Mecanismo de Realimentação

Para manter a fonte sempre informada sobre a melhor taxa de envio do vídeo, e assim, sobre o número de camadas que ela deve gerar, se faz necessário um mecanismo de realimentação por parte dos receptores do grupo multiponto. Para isso, todos os receptores enviam periodicamente cápsulas de disponibilidade à fonte. A interatividade entre a fonte e os receptores é alcançada, uma vez que, a fonte pode se adaptar ao receber a cápsula de disponibilidade que traz consigo, de forma direta ou indireta, a qualidade desejada por cada receptor ou a qualidade suportada pela rede. Ajustes de qualidade realizados por um receptor, durante a sessão, são automaticamente incorporados às novas cápsulas de disponibilidade enviadas em direção à fonte.

O Mecanismo de Fusão

O envio periódico de cápsulas de disponibilidade por todos os receptores do grupo multiponto acarreta dois efeitos indesejáveis. O primeiro é a atualização descoordenada das tabelas auxiliares e de disponibilidade. O segundo, é a possibilidade da implosão de cápsulas de disponibilidade que chegam à fonte devido ao crescimento do número de receptores no grupo multiponto.

A figura 3.9 apresenta uma idéia do primeiro efeito. Nela, um nó ativo possui duas interfaces de chegada de cápsulas de disponibilidade. Para a interface 1 existem quatro entradas na tabela auxiliar e para a interface 2 existem três. As cápsulas de disponibilidade provindas dos nós identificados na tabela auxiliar chegam por estas interfaces. Cada cápsula atualiza uma entrada da tabela auxiliar e da tabela de disponibilidade, levando consigo para o próximo nó o maior valor observado nessa última tabela. Considerando que essas cápsulas chegam ao nó em um intervalo de tempo curto, a atualização das tabelas ocorrerá de forma descoordenada. Cada cápsula realizará sua função e seguirá para o próximo nó independentemente das outras cápsulas que estão por chegar. Além disso, considerando este intervalo de tempo curto, somente a última cápsula carregará consigo a informação mais atualizada a ser enviada.

Tabela Auxiliar

iface	id. do nó ant.	qualidade
1	J	1 Mbps
1	K	3 Mbps
1	L	2 Mbps
1	M	8 Mbps
2	N	6 Mbps
2	O	4 Mbps
2	P	1 Mbps

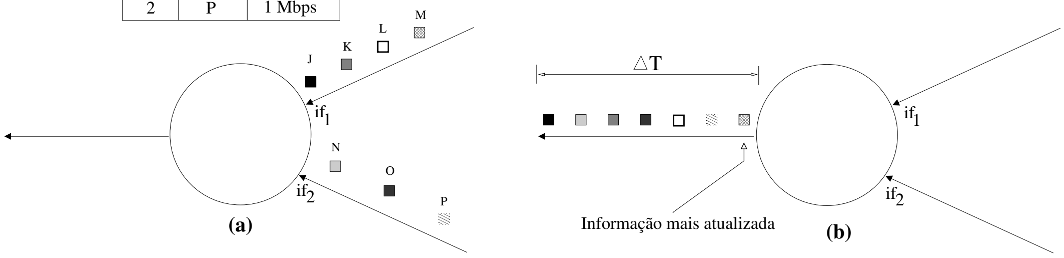


Figura 3.9: Efeito indesejável do envio periódico de cápsulas de disponibilidade à fonte.

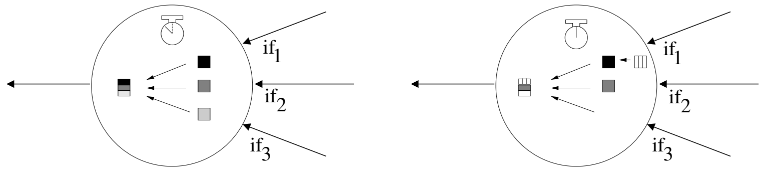
Sendo assim, para impedir que ocorra a implosão de cápsulas de disponibilidade e coordenar a atualização das tabelas, suprimindo cápsulas com informações desatualizadas, é adotado um mecanismo de fusão dessas cápsulas. Esta fusão é realizada na ocorrência de uma das seguintes condições: (1) pelo menos uma cápsula de disponibilidade referente à cada entrada na tabela auxiliar tenha chegado; (2) o temporizador de fusão tenha expirado. A figura 3.10(a) ilustra a primeira condição, onde uma cápsula de disponibilidade referente a cada entrada na tabela auxiliar chega pelas respectivas interfaces antes da expiração do temporizador de fusão. Neste caso, ao chegar a última cápsula esperada, para cada cápsula armazenada, é atualizada a respectiva entrada na tabela auxiliar com base na equação 3.1. Em seguida, com base na equação 3.2 é atualizada de uma só vez a tabela de disponibilidade e transmitida apenas uma cápsula de disponibilidade contendo em TX o resultado da equação 3.3. A atualização de um campo na tabela de disponibilidade, relativo a uma interface, ocorre somente quando existir pelo menos uma cápsula armazenada que tenha chegado por esta interface. A figura 3.10(b) ilustra a segunda condição. Neste caso, chegaram cápsulas de disponibilidade relativas somente à duas entradas na tabela auxiliar. Ao expirar o temporizador de fusão, as entradas na tabela auxiliar são atualizadas seguindo a equação 3.1. Após o término destas atualizações, com base na equação 3.2, são atualizadas as entradas na tabela de disponibilidade e é transmitida uma única cápsula contendo o resultado da equação 3.3. Se antes de ocorrer a fusão mais de uma cápsula de disponibilidade relativa à mesma entrada na tabela auxiliar chegar, a informação armazenada é descartada em favor do novo dado recebido.

Tabela Auxiliar

iface	id. do nó ant.	qualidade
1	Q	1 Mbps
2	R	3 Mbps
3	S	2 Mbps

Tabela Auxiliar

iface	id. do nó ant.	qualidade
1	Q	1 Mbps
2	R	3 Mbps
3	S	2 Mbps



(a) (b)

Figura 3.10: O mecanismo de fusão.

3.2.3 A Cápsula de Dados de Vídeo

A cápsula de dados de vídeo utiliza as informações contidas na tabela de disponibilidade dos nós e o valor contido no campo TX de seu cabeçalho para tomar decisões de roteamento. Possíveis decisões incluem o auto-descarte, seguir por uma interface ou replicar-se para várias interfaces de saída. Para auxiliar essa tomada de decisão, a fonte coloca no campo TX de cada cápsula de dados de vídeo a taxa cumulativa referente à camada à qual pertence a cápsula, ou seja, o campo TX é marcado de acordo com a seguinte regra:

$$TX = \sum_{i=1}^{c} TC_i \text{ para } 1 \leq c \leq N \tag{3.4}$$

onde c é à camada a qual pertence a cápsula, TC_i é a taxa da i-ésima camada e N é o número máximo de camadas que a fonte pode gerar.

Ao chegar em um nó, a cápsula de dados de vídeo verifica se existe um *soft-state* para o grupo multiponto ao qual ela pertence. Caso não exista, ela se auto-descarta. Havendo *soft-state* para o seu grupo, ela verifica, para cada interface de saída contida na tabela de disponibilidade, se a taxa indicada é maior ou igual a taxa informada no campo TX. Neste caso, ela segue por esta interface, devendo ser replicada quando mais de uma interface atender a esta condição. Se nenhuma interface atender a esta condição, a cápsula então se auto-descarta. Desta forma, o mecanismo de formação e manutenção da árvore multiponto associado à filtragem das cápsulas de vídeo elimina a necessidade do estabelecimento de múltiplas árvores multiponto para a distribuição do vídeo em camadas. A única árvore multiponto existente é a percorrida por cápsulas da camada base. As cápsulas das camadas adicionais podem percorrer toda essa árvore ou simplesmente trechos dela. Além disso, a quantidade de recursos alocados nos nós torna-se independente do número de camadas geradas pela fonte. Isto não ocorre quando se distribui cada camada em uma árvore multiponto diferente.

Já o mecanismo de criação e atualização das tabelas de disponibilidade associado à filtragem realizada pelas próprias cápsulas de dados de vídeo permite que cada receptor receba um número determinado de camadas, de acordo com a característica do caminho individual entre a fonte e este receptor e a qualidade Q desejada.

3.2.4 A Cápsula de Fim de Sessão

A cápsula de fim de sessão é utilizada pela fonte de vídeo e tem como função alcançar todos os receptores do grupo multiponto, informando-lhes sobre o fim da sessão. Ao chegar em um nó, esta cápsula verifica se há *soft-state* para o grupo multiponto ao qual ela pertence. Caso não haja, ela se auto-descarta. Havendo *soft-state* para o seu grupo, ela é replicada para cada uma das interfaces de saída contidas na tabela de disponibilidade.

Se após enviar uma cápsula de fim de sessão, a fonte receber cápsulas de inscrição/manutenção ou de disponibilidade, ela envia novamente a cápsula de fim de sessão. Isso é realizado devido ao fato de não existir garantia de que a cápsula anterior de fim de sessão tenha alcançado todos os receptores. As cápsulas de inscrição/manutenção ou de disponibilidade que porventura cheguem à fonte após o fim da sessão, poderiam estar em trânsito na rede enquanto a cápsula de fim de sessão estivesse tentando realizar sua função. Elas poderiam também ser provindas de receptores que não receberam a informação sobre o fim da sessão, ou ainda, de receptores que tenham tentado entrar no grupo multiponto após o término da sessão.

Ao receber a cápsula de fim de sessão, o receptor deixa de enviar cápsulas de inscrição/manutenção e de disponibilidade à fonte. Sendo assim, a árvore multiponto será desfeita devido a expiração, nos nós ativos, do *soft-state* associado ao respectivo grupo multiponto.

3.2.5 A Cápsula de Desligamento

O serviço ativo proposto prevê o comportamento dinâmico dos receptores ao longo de uma sessão. A qualquer instante, um receptor pode se conectar ou se desconectar do serviço. Para se desconectar do serviço, o receptor deve interromper o envio de cápsulas de inscrição/manutenção e de disponibilidade e deve enviar, em direção à fonte, uma cápsula de desligamento. A função dessa cápsula é alcançar a fonte ou, ao percorrer o caminho em direção à fonte, alcançar o primeiro nó ativo que possui mais de uma entrada na tabela auxiliar. Ao encontrar este nó, a cápsula de desligamento remove da tabela auxiliar a entrada referente a sua interface de chegada e ao nó anterior do qual ela foi enviada. Se existirem outras entradas referentes a sua interface de chegada, a cápsula se auto-descarta. Caso contrário, não existindo mais entradas na tabela auxiliar referentes a sua interface, a cápsula remove a respectiva entrada da tabela de disponibilidade, se auto-descartando em seguida. Sendo assim, as cápsulas de dados de vídeo deixam de se auto-rotearem em direção ao receptor que saiu da sessão.

Caso, ao longo do caminho para a fonte, não existam nós com mais de uma entrada na tabela auxiliar, a cápsula de desligamento chegará à fonte. Neste caso, ao receber este tipo de cápsula, a fonte conclui que não há mais receptores na sessão e interrompe a geração do vídeo.

Os recursos utilizados nos nós ao longo do caminho percorrido pela cápsula de desligamento são liberados tão logo ocorra a expiração do *soft-state*. Mesmo que a cápsula de desligamento seja perdida antes de realizar sua função, a robustez ainda estará garantida. Isto ocorre porque a saída do receptor do grupo multiponto ou ocorrerá pela expiração do *soft-state* nos nós ativos, ou pela remoção de apontadores expirados por parte das cápsulas de inscrição/manutenção ou por parte de outras cápsulas de disponibilidade.

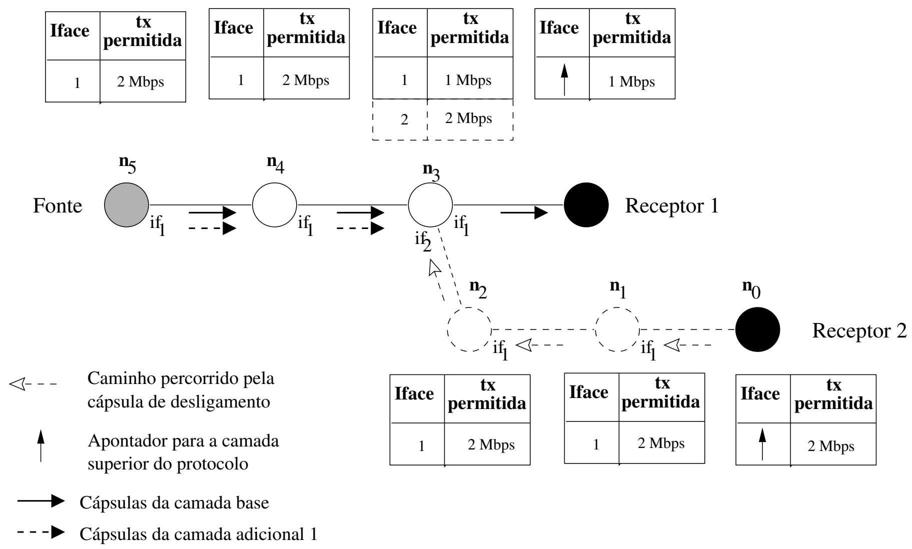


Figura 3.11: Saída de um receptor do grupo multiponto.

A figura 3.11 ilustra a saída de um receptor de um grupo multiponto. Por simplicidade, só estão representadas as tabelas de disponibilidade. A tabela auxiliar do nó n_3 possui duas entradas, uma devida ao receptor 1 e outra devida ao receptor 2. Nesta figura, uma fonte de vídeo gera duas camadas cada qual à 1 Mbps. O primeiro receptor está recebendo apenas a camada base, enquanto o segundo receptor estava recebendo a camada base e a camada adicional. Ao chegar no nó n_3, a cápsula de desligamento remove das tabelas de disponibilidade e auxiliar, a entrada relativa à interface por onde ela chegou e, em seguida, se auto-descarta. As cápsulas de dados de vídeo da camada adicional, ao chegarem neste nó, também se auto-descartarão pois a taxa contida na tabela de disponibilidade é inferior à taxa da camada adicional. Note, contudo, que a fonte só interromperá o envio da camada adicional após receber a cápsula de disponibilidade trazendo a nova informação sobre a melhor taxa de envio do vídeo. O auto-descarte das cápsulas de dados de vídeo associado com a atualização das tabelas de disponibilidade mostra que a adaptação do serviço ocorre de forma gradativa à medida que a cápsula de disponibilidade percorre o caminho até a fonte. Com base na figura 3.11, esta situação está apresentada na figura 3.12.

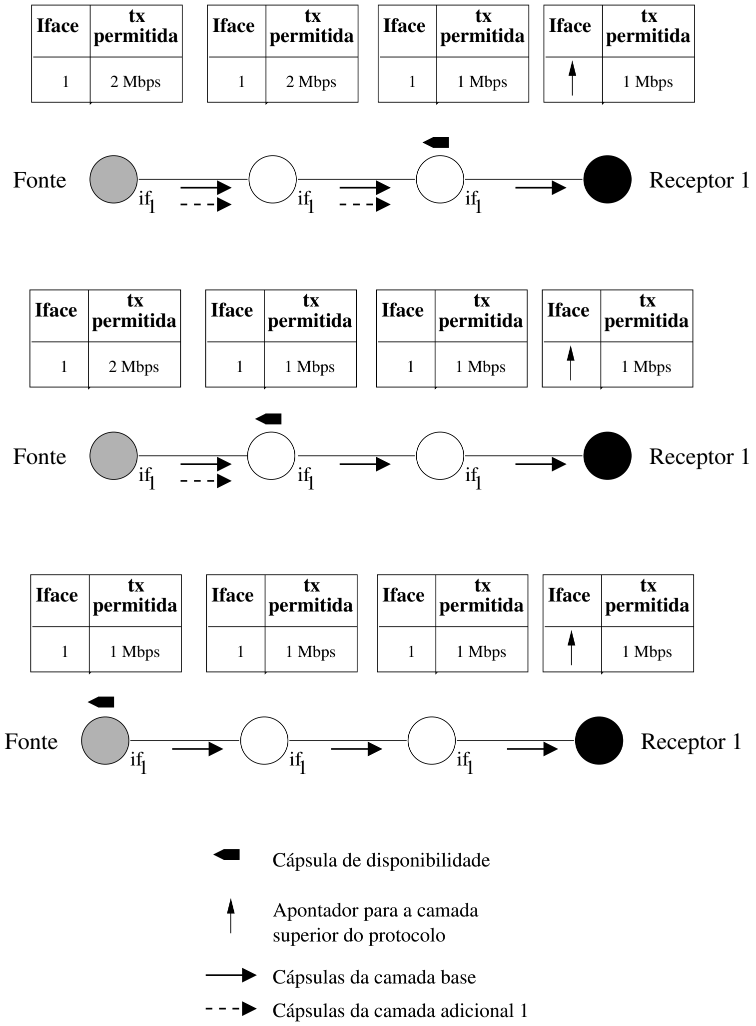


Figura 3.12: Seqüência ilustrando a adaptação gradual do serviço à medida que a cápsula de disponibilidade atualiza a tabela de disponibilidade de cada nó visitado.

3.2.6 A Máquina de Estados do Protocolo do Receptor

A máquina de estados do protocolo executado pelo receptor está representada na figura 3.13. O significado dos parâmetros apresentados está tabela 3.1. Ao desejar se conectar ao serviço, o receptor passa do estado *desconectado* para o estado *tentando conexão*. Nesse estado, são disparados os temporizadores de envio de cápsulas de inscrição/manutenção, de envio de cápsulas de disponibilidade e de silêncio. Este último, ao expirar, indica que o vídeo não está sendo recebido. Neste caso, a tentativa de conexão ao serviço falhou e o receptor passa então ao estado *desconectado* . O receptor também passa ao estado *desconectado* se receber uma cápsula de fim de sessão. Isso é possível, uma vez que ele pode estar tentando se conectar logo após o término da sessão. Se no estado *tentando conexão* o receptor começar a receber o vídeo, então ele passa ao estado *conectado*, e continua enviando cápsulas de disponibilidade e de inscrição/manutenção de acordo com as suas respectivas temporizações. Nesse estado, o receptor pode passar ao estado *desconectado* se receber uma cápsula de fim de sessão, ou se enviar uma cápsula de desligamento, ou se o temporizador de silêncio expirar. No estado *desconectado*, o receptor que receber outra cápsula de fim de sessão deve ignorá-la, permanecendo neste estado.

Tabela 3.1: Parâmetros apresentados na máquina de estados do protocolo do receptor.

$T_{I/M}$	Temporizador de envio de cápsulas de inscrição/manutenção
T_D	Temporizador de envio de cápsulas de disponibilidade
T_S	Temporizador de silêncio
V	Vídeo sendo recebido
F	Cápsula de fim de sessão recebida
De	Cápsula de desligamento enviada

O período de envio de cápsulas de inscrição/manutenção depende do tempo de expiração do *soft-state* nos nós ativos. Estas cápsulas devem ser enviadas com uma freqüência que permita sustentar a continuidade da comunicação, considerando o

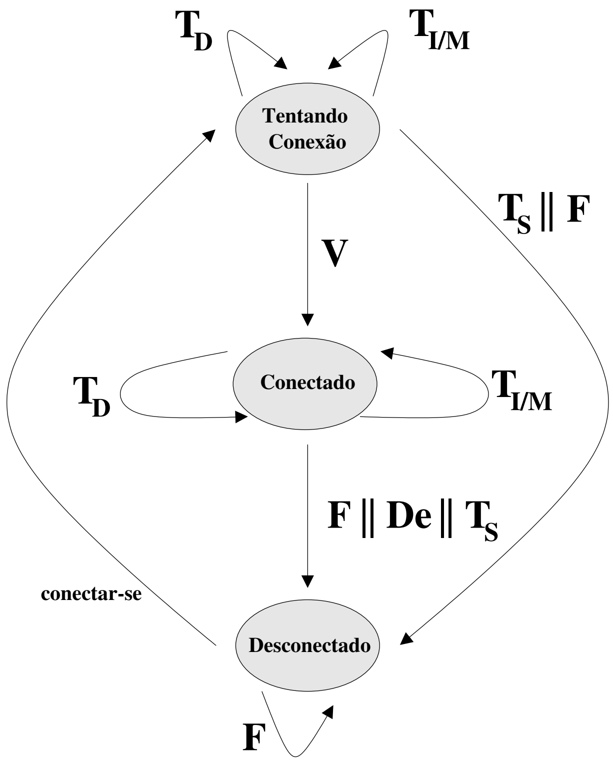


Figura 3.13: Máquina de estados do protocolo do receptor.

compromisso de não inundar a rede com este tipo de cápsula. Junqueira [32] demonstra que o período de envio de cápsulas de inscrição/manutenção, representado por $T_{I/M}$ deve respeitar a seguinte condição:

$$\mathbf{T_{I/M}} \leq \beta \times \mathbf{T_{sf}} \quad \text{onde } 0 \leq \beta < 0.5 \tag{3.5}$$

onde T_{sf} representa o período de expiração do *soft-state*.

O período de envio de cápsulas de disponibilidade, representado por T_D, deve manter o compromisso de não inundar a rede com cápsulas de disponibilidade e manter uma taxa adequada de atualização das tabelas de disponibilidade de modo que a informação que chegue à fonte melhor represente a taxa atual de envio permitida para o vídeo. Resultados de simulação que indicam como escolher este parâmetro serão apresentados no capítulo 4.

Com relação ao temporizador de silêncio, seu valor pode ser escolhido de acordo com a seguinte equação:

$$\mathbf{T_S} = \alpha \times \mathbf{T_{I/M}} \tag{3.6}$$

ou seja, o temporizador de silêncio expira após o envio de α cápsulas de inscrição/manutenção sem sucesso no recebimento do vídeo.

Capítulo 4

Resultados de Simulação

COM o intuito de verificar o comportamento global do serviço proposto e, em particular, o desempenho do mecanismo de fusão utilizado, foram realizadas simulações. O comportamento do mecanismo de fusão foi avaliado utilizando-se duas topologias em árvore e outra similar a estrutura encontrada na Internet. O estudo do desempenho do mecanismo de fusão permite avaliar a escalabilidade do serviço ativo proposto. Na seção 4.1 é apresentada a infra-estrutura utilizada na simulação. A modelagem da fonte de vídeo em camadas, o mecanismo de estimação de banda passante e as métricas de avaliação são apresentados na seção 4.2. A seção 4.3 apresenta os resultados de simulação do comportamento do serviço ativo proposto. Na seção 4.4 são apresentadas as simulações referentes ao comportamento do mecanismo de fusão. Os resultados de simulação para uma topologia em árvore são apresentados na seção 4.5. E finalmente, os resultados para uma topologia similar a estrutura encontrada na Internet são apresentados na seção 4.6.

4.1 Infra-estrutura de Simulação

A ferramenta escolhida para realizar as simulações foi o *ns* (*Network Simulator*) [33, 34]. O *ns* é um simulador de redes dirigido a eventos discretos que são controlados e configurados através da linguagem OTcl (*Object Tool Command Lan-*

guage). Ele fornece abstrações de nós e enlaces e vem sendo desenvolvido para a simulação e análise dos serviços de rede voltados principalmente para a Internet.

As simulações configuradas pelos *scripts* OTcl descrevem, por exemplo, a topologia de rede, incluindo as características dos nós e da interconexão física entre eles; os protocolos utilizados; modelos de geração de tráfego; e a modelagem de perdas de pacotes associadas à falhas em nós e enlaces da rede.

Contudo, o *ns* é baseado no paradigma de redes passivas. Deste modo, utilizou-se uma versão estendida deste simulador com suporte básico às redes ativas desenvolvido no Grupo de Teleinformática e Automação (GTA) da COPPE/UFRJ [32, 35]. Para a implementação do serviço proposto foram desenvolvidos novos agentes[1], novas cápsulas, o mecanismo de fusão, mecanismos adicionais de controle de estados, e um mecanismo de monitoração de banda passante nos enlaces. Maiores detalhes sobre o *ns* e as modificações para a implementação do serviço são apresentados no apêndice A. As simulações foram executadas em um computador com processador Pentium II-350Mhz com 64 MB de memória RAM e sistema operacional *Linux*.

4.1.1 Geração de Topologias

As topologias para a simulação no *ns* podem ser geradas através do gerador de topologias GT-ITM (*Georgia Tech - Internetwork Topology Models*). Ele é capaz de gerar aleatoriamente tanto topologias simples quanto topologias mais reais que possuem estruturas de domínios semelhantes às encontradas na Internet, e que são denominadas topologias do tipo *transit-stub*. Em um domínio *stub* só trafegam pacotes destinados ou gerados neste domínio, enquanto em um domínio *transit* não existe esta definição. Os domínios *transits* representam os *backbones* da Internet [36, 37]. Um exemplo de topologia *transit-stub* é apresentado na figura 4.1.

[1] No *ns*, um agente é um gerador e consumidor de pacotes que implementa um determinado tipo de protocolo. Alguns exemplos são os agentes TCP (*Transmission Control Protocol*), os agentes UDP (*User Datagram Protocol*) e os agentes nulos. Estes últimos, somente consomem pacotes.

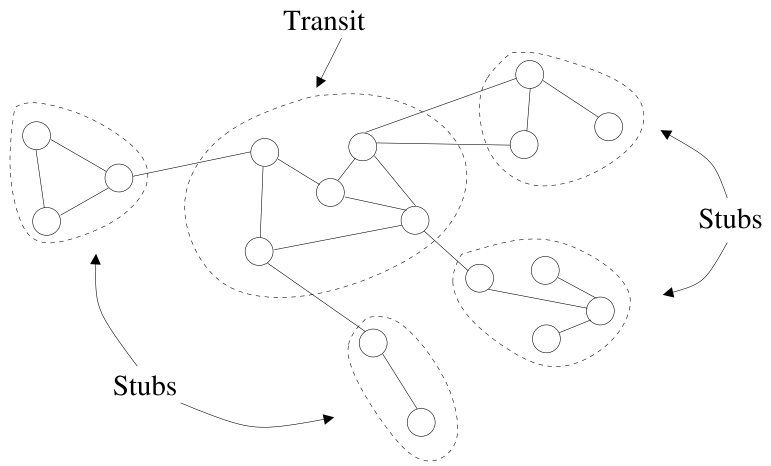


Figura 4.1: Exemplo de topologia do tipo *transit-stub*.

4.2 Modelagem, Estimação e Métricas de Avaliação

Para modelar a fonte de vídeo em camadas foi adotada a proposta apresentada em [10]. Cada fonte é composta por um conjunto de fluxos, cada qual modelado por um tráfego *jitter* CBR (*Constant Bit Rate.*) Os pacotes de um fluxo *jitter* CBR são gerados de tempos em tempos de acordo com a seguinte lei:

$$
\begin{aligned}
&T_0 = 0 \\
&T_k = T_{k-1} + \Delta + N_k \quad , \mathrm{k} > 0
\end{aligned}
\tag{4.1}
$$

onde Δ é um intervalo fixo para manutenção da taxa, N_k representa um processo de ruído com média zero cujo objetivo é modelar a variação de atraso na codificação do vídeo. Os valores $N_1, N_2, \ldots$ são independentes e identicamente distribuídos no intervalo $[-\Delta/2, +\Delta/2]$.

Como mecanismo de medida da banda passante consumida por um fluxo foi utilizado um estimador EWMA (*Exponential Weighted Moving Average*), similarmente ao utilizado pelo TCP para computar o RTT (*Round-Trip Time*) médio. O estimador é definido pela equação 4.2, onde v_n é a taxa média estimada sendo consumida por um fluxo, v^S representa a taxa média de chegada de cápsulas medida dentro do período de amostragem S. O valor anterior computado pela equação é v_{n-1} e ω é o fator de suavização que determina o quanto de peso é dado ao valor anteriormente computado.

$$v_n = (1-\omega)v_{n-1} + \omega v^S \text{ para } 0 \leq \omega \leq 1 \text{ e } n > 0 \tag{4.2}$$

A avaliação dos resultados apresentados neste capítulo referentes ao mecanismo de fusão foram feitas com base em quatro métricas distintas. A primeira delas é o índice de filtragem que é definido como o percentual de cápsulas de disponibilidade que não chegam à fonte do total de cápsulas de disponibilidade enviadas pelos receptores. Esta métrica é utilizada para a verificação do desempenho do mecanismo de fusão. Para não influenciarem nesta medida, os mecanismos de modelagem de perdas de pacotes do *ns* não foram utilizados. A segunda métrica é o intervalo médio de chegada de cápsulas de disponibilidade à fonte. O comportamento dessa métrica será estudado variando-se o intervalo de envio de cápsulas de disponibilidade pelos receptores. A terceira métrica refere-se à percentagem do número de fusões que ocorrem na rede de acordo com as duas condições possíveis para haver fusão. Deste modo, pretende-se verificar como ocorrem as fusões na rede. Finalmente, a quarta métrica é a percentagem de substituições de cápsulas de disponibilidade no primeiro nó ativo visitado por estas cápsulas que possua mais de uma entrada na tabela auxiliar. O objetivo dessa métrica é avaliar o desperdício de cápsulas de disponibilidade enviadas pelos receptores.

4.3 Comportamento do Serviço Ativo Proposto

O serviço ativo proposto foi avaliado, por partes, através de quatro conjuntos de simulações referentes, respectivamente, ao comportamento dinâmico dos receptores, à distribuição do vídeo, à robustez do serviço e ao mecanismo de fusão. Para tal fim, foram implementados no *ns* a geração de relatórios específicos para o serviço proposto. Os relatórios são gerados a partir de uma simulação e fornecem informações sobre eventos como conexão e desconexão de receptores do serviço; geração, recebimento e descarte de cápsulas; e remoção de *soft-state* dos nós ativos. Mais detalhes sobre a geração de relatórios são apresentados no apêndice A. Para visualizar os eventos ocorridos durante a simulação foi utilizada a ferramenta *nam* (*Network Animator*). O *nam* é um programa que a partir de um arquivo específico gerado por uma simulação no *ns* realiza uma "animação" da geração e da transmissão dos pacotes entre os diversos nós que formam a rede. A interface gráfica do *nam* possui controles que permitem iniciar, avançar, retroceder, e parar uma animação. Sendo assim, possibilitando ao usuário do programa observar passo a passo o comportamento de uma simulação.

No primeiro conjunto de simulações foi avaliado o funcionamento do serviço frente ao comportamento dinâmico dos receptores. O *script* de simulação do *ns* foi configurado de forma que os receptores permanecessem conectados ao serviço por um período de tempo escolhido de forma aleatória, uniformemente distribuído no interavlo [10, 50] segundos. Após a desconexão, cada receptor aguardava um período de tempo aleatório, distribuído uniformemente no intervalo [0, 20] segundos para tentar uma nova conexão. Foi verificado o correto funcionamento do serviço nas fases de conexão, de desconexão e de reestabelecimento de conexão.

O segundo conjunto de simulações visou verificar se a distribuição do vídeo estava de acordo com a qualidade desejada por cada receptor e a banda passante disponível para o vídeo no caminho entre a fonte e os receptores. Neste conjunto de simulações, a fonte gerava 7 camadas, cada qual com a respectiva taxa de 32 kbps, 64 kbps, 128 kbps, 256 kbps, 512 kbps, 1024 kbps e 2048 kbps. A qualidade desejada por cada receptor foi escolhida de forma aleatória, uniformemente distribuída no intervalo [32, 4064] kbps. A capacidade de banda passante de alguns enlaces foi escolhida de forma a criar pontos de gargalo na rede. Nas simulações, verificou-se que as cápsulas de dados de vídeo de cada camada seguiam somente para os receptores que desejavam receber a camada em questão e cuja disponibilidade de banda passante no caminho individual até a fonte permitia. Foi observada uma discrepância entre a qualidade que um receptor poderia receber e qualidade que estava realmente sendo recebida. Isso ocorre porque o receptor pode escolher uma qualidade arbitrária no intervalo [32, 4064] kbps e a taxa cumulativa das camadas sendo geradas pela fonte é fixa. Sendo assim, para reduzir essa discrepância, permitindo um ajuste mais fino da qualidade, é necessário que a fonte seja capaz de gerar um número mais elevado de camadas e que a diferença de taxa entre duas camadas consecutivas seja baixa. Também não foi observada a influência entre os receptores, ou seja, ao tentar receber uma qualidade melhor do vídeo, um receptor não influenciava na qualidade do vídeo sendo recebido por outro receptor. Este resultado já era esperado pois essa característica é intrínseca ao mecanismo de distribuição de vídeo proposto.

No terceiro conjunto de simulações foi avaliada a robustez do serviço na ocorrência de falhas. Foram criados cenários de simulação que permitiram verificar o funcionamento correto do serviço tanto na ocorrência de mudanças de topologia, uma vez que as cápsulas de inscrição/manutenção criavam novos caminhos até a fonte, quanto na manutenção coerente dos estados nos nós ativos. Esta manutenção coerente dos estados nos nós ativos foi garantida pela cooperação entre as cápsulas de inscrição/manutenção e as cápsulas de disponibilidade que removiam entradas nas tabelas que não tinham suas estampas de tempo reiniciadas. Para verificar a remoção, pelos nós ativos, de *soft-states* expirados, foram criadas duas situações. A primeira situação foi gerada a partir da desconexão de um receptor do serviço. Como apresentado na seção 3.2.5, a cápsula de desligamento não remove o *soft-state* nos nós ativos pelos quais passa. Esta cápsula desconecta um receptor do serviço removendo a devida entrada na tabela auxiliar do primeiro nó ativo no caminho para a fonte com mais de uma entrada na tabela auxiliar. Nesta primeira situação simulada, foi utilizada a topologia apresentada na figura 3.11. Após a desconexão do receptor do serviço, a medida que os *soft-states* foram expirando, os nós ativos liberavam esses recursos. Na segunda situação, foi verificada a remoção de *soft-states* expirados após a fonte enviar uma cápsula de fim de sessão para os receptores. Mais uma vez, foi verificado o correto funcionamento do mecanismo de controle de estados nos nós ativos, uma vez que os *soft-states* expirados foram devidamente removidos.

O quarto conjunto de simulações é referente ao mecanismo de fusão utilizado no serviço proposto. Este mecanismo será estudado com mais detalhes nas próximas seções deste capítulo, uma vez que seu comportamento e seu desempenho influenciam diretamente na escalabilidade do serviço proposto.

4.4 Comportamento do Mecanismo de Fusão

O comportamento do mecanismo de fusão foi avaliado por simulação utilizando-se a topologia apresentada na figura 4.2. Os enlaces possuem banda passante de 10 Mbps e atraso de 2 ms. A expiração do temporizador de fusão ocorre a cada 100 ms. O receptor 1 se conecta ao serviço no instante $t = 5$ segundos. Os gráficos apresentados foram obtidos variando-se em passos de 50 ms, dentro do intervalo [0, 10] segundos, o instante de conexão do receptor 2 ao serviço. São apresentados gráficos para diferentes intervalos de envio de cápsulas pelos receptores. O tempo total de simulação foi de 100 segundos. O índice de filtragem máximo para esta topologia é de 50%.

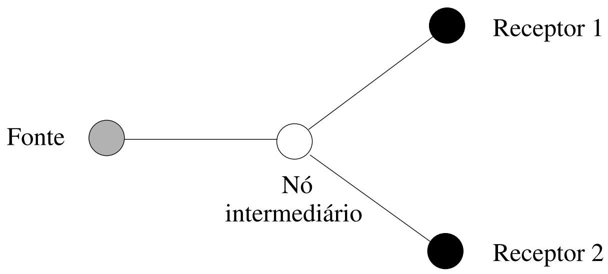


Figura 4.2: Topologia para o estudo do comportamento do mecanismo de fusão.

O comportamento do mecanismo de fusão é apresentado pelo conjunto de curvas que formam a figura 4.3. Cada subconjunto de três figuras é referente a um intervalo específico de envio de cápsulas de disponibilidade pelos receptores. Analisando as curvas das figuras 4.3(a), 4.3(b) e 4.3(c), referentes ao intervalo de envio de 120 ms, verifica-se a ocorrência de pontos de sincronismo de envio de cápsulas pelos dois receptores. Tais pontos são evidenciados pelos picos apresentados nas figuras 4.3(a) e 4.3(b). Contudo, a filtragem máxima de 50% só ocorre quando os dois receptores se conectam ao serviço ao mesmo tempo. Os outros picos de filtragem são menores que 50% devido a fusão não poder começar a atuar enquanto somente um receptor estiver conectado. Deste modo, as cápsulas enviadas pelo único receptor conectado não podem ser filtradas, contribuindo de forma negativa ao desempenho do mecanismo de fusão. A diferença entre os valores máximo e mínimo de filtragem é de aproximadamente 22%. A figura 4.3(c) mostra que não ocorrem substituições de cápsulas no nó ao qual a fonte está diretamente ligada. Isto já era esperado pois não podem existir substituições de cápsulas de disponibilidade no nó intermediário quando o intervalo de envio dessas cápsulas for maior ou igual ao tempo de expiração do temporizador de fusão. Suponha, por exemplo, que intervalo de envio seja de 110 ms e o temporizador de fusão expire a cada 100 ms. Suponha ainda que o receptor 1 tenha enviado uma cápsula de disponibilidade que tenha sido armazenada pelo nó intermediário imediatamente após a reiniação do temporizador de fusão. Sendo assim, na pior das hipóteses, a cápsula armazenada aguardará a fusão por 100 ms. Quando o receptor 1 enviar a próxima cápsula de disponibilidade, esta cápsula só chegará ao nó intermediário 10 ms após a reiniciação do temporizador de fusão e, conseqüentemente, após a cápsula armazenada ser enviada para a fonte.

As curvas apresentadas nas figuras 4.3(d), 4.3(e) e 4.3(f) referem-se ao intervalo de envio de 100 ms, ou seja, o mesmo valor do período de expiração do temporizador de fusão. Mais uma vez, o índice de filtragem máximo ocorre quando os dois receptores se conectam ao serviço ao mesmo tempo. Contudo, dois fatos importantes devem ser observados. O primeiro fato é a diferença entre os índices de filtragem máximo e mínimo que é de apenas 2,8%, um valor muito baixo quando comparado aos 22% apresentados anteriormente. E o segundo, é com relação a uma possível conclusão errônea, através da análise da figura 4.3(e), de que os receptores estão sempre sincronizados, uma vez que aparentemente 100% das fusões ocorrem pela condição 1. A figura 4.4 é uma ampliação da figura 4.3(e) ressaltando apenas os resultados referentes a esta condição. Esta figura mostra uma percentagem de fusões ocorridas pela condição 1 bastante elevada porém, existem somente alguns pontos de sincronismo. Para o intervalo de envio de 100 ms também não há substituições de cápsulas como apresentado na figura 4.3(f).

Nas figuras 4.3(g), 4.3(h) e 4.3(i) são apresentadas as curvas relativas ao intervalo de envio de 80 ms. Essas curvas são bem semelhantes as apresentadas nas figuras 4.3(j), 4.3(k) e 4.3(l) as quais são referentes ao intervalo de envio de 60 ms. A redução do intervalo de envio para estes dois valores abaixo de 100 ms mostra um aumento progressivo na quantidade de pontos de sincronismo. Também pode-se observar que começam haver substituições de cápsulas de disponibilidade que estavam

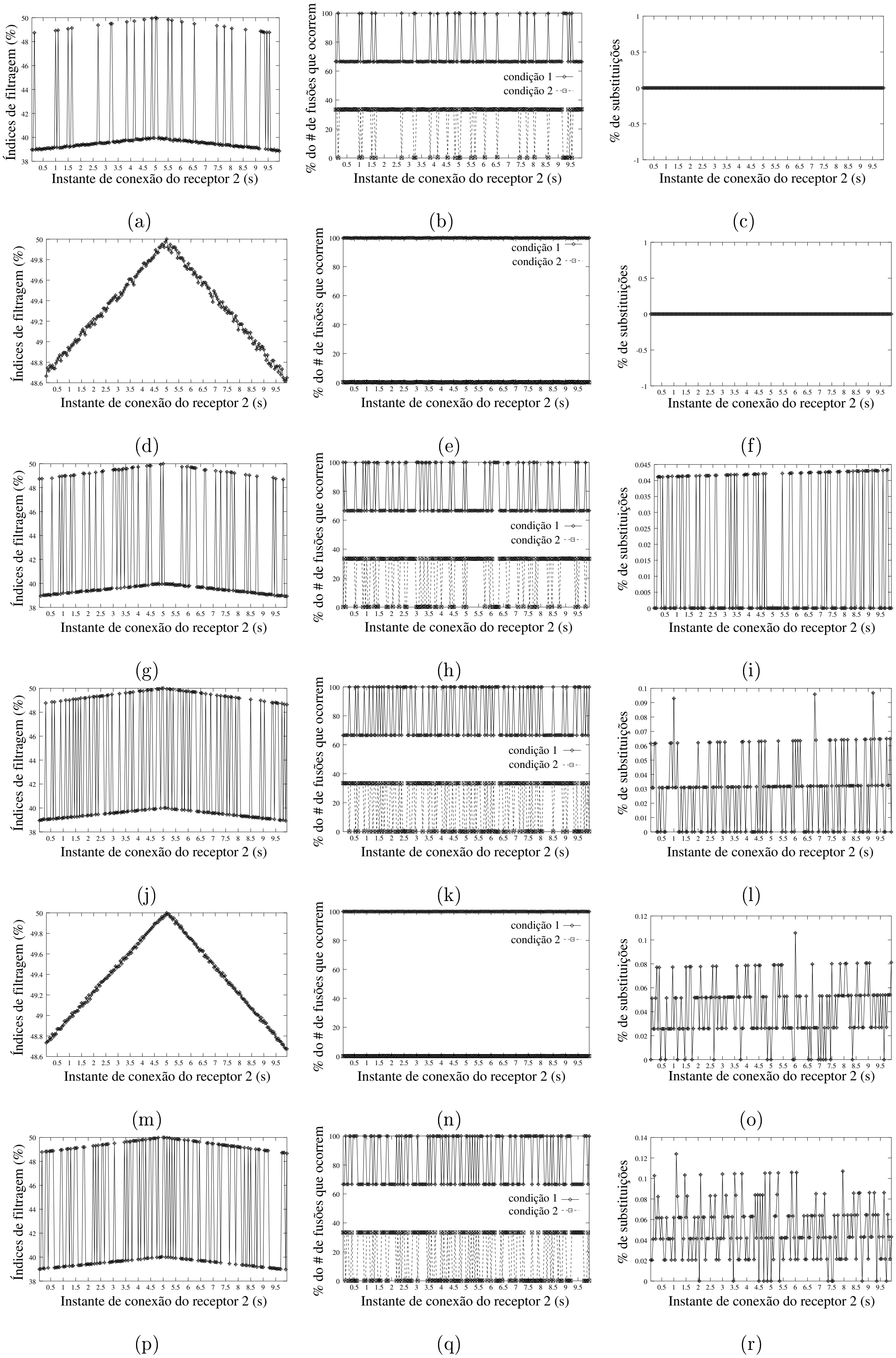


Figura 4.3: Comportamento do mecanismo de fusão.

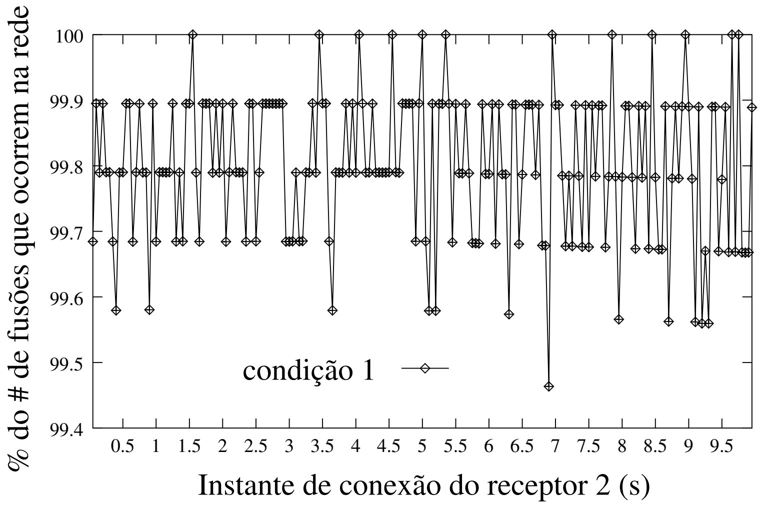


Figura 4.4: Percentagem das fusões que ocorrem devido somente à condição 1 versus instante de conexão do receptor 2 ao serviço. Intervalo de envio de cápsulas de disponibilidade igual a 100 ms.

armazenadas esperando a fusão. A figura 4.3(i) mostra que a percentagem máxima de cápsulas substituídas é de aproximadamente 0,045% quando o intervalo de envio é igual a 80 ms. O aumento desta percentagem máxima é observado na figura 4.3(l) onde este valor chega próximo dos 0,1%.

Já as figuras 4.3(m), 4.3(n) e 4.3(o) mostram um comportamento bastante interessante desse mecanismo de fusão quando o intervalo de envio é de 50 ms. As curvas obtidas são praticamente iguais as apresentadas para o intervalo de envio de 100 ms, exceto pela percentagem de substituições que em relação aos resultados anteriores aumentou para um valor máximo próximo de 0,11%. Uma versão ampliada da figura 4.3(n) ressaltando a curva referente à condição 1 é apresentada na figura 4.5. Observa-se um ligeiro aumento no valor médio desta curva com relação a apresentada na figura 4.4.

As curvas apresentadas nas figuras 4.3(p), 4.3(q) e 4.3(r) mostram o comportamento do mecanismo de fusão quando o intervalo de envio de cápsulas é de 40 ms. O valor máximo observado para a percentagem de substituições se elevou mais, passando agora para aproximadamente 0,12% como mostra a figura 4.3(r). Analisando-se todas as curvas de filtragem apresentadas, verifica-se que a envoltória superior das curvas representadas nas figuras 4.3(a), 4.3(g), 4.3(j) e 4.3(p) é a mesma das curvas apresentadas nas figuras 4.3(e) e 4.3(n). Adicionalmente, excluindo as curvas relacionadas ao intervalo de envio de 100 ms e 50 ms, vale observar que a quantidade de pontos de sincronização aumenta a medida que o intervalo de envio diminui.

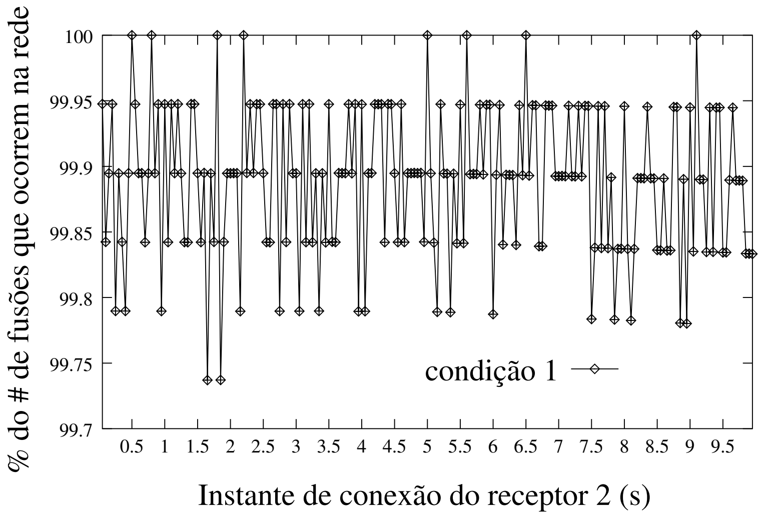


Figura 4.5: Percentagem das fusões que ocorrem devido somente à condição 1 versus instante de conexão do receptor 2 ao serviço. Intervalo de envio de cápsulas igual a 50 ms.

4.5 Resultados para Topologia em Árvore

Os resultados apresentados nesta seção consideram uma árvore básica, sobre a qual são adicionados os receptores e a fonte. Nesta árvore básica, todos os nós possuem o mesmo número de filhos, exceto as folhas as quais não possuem filhos. Para formar a topologia de estudo, são adicionados, então, os receptores, um por folha e conecta-se a fonte de vídeo à raiz desta árvore. Um exemplo deste tipo de topologia é apresentado na figura 4.6.

4.5.1 Índice de Filtragem Máximo

O índice de filtragem máximo teórico para topologias em árvore do tipo apresentado na figura 4.6 pode ser facilmente calculado fazendo-se algumas considerações:

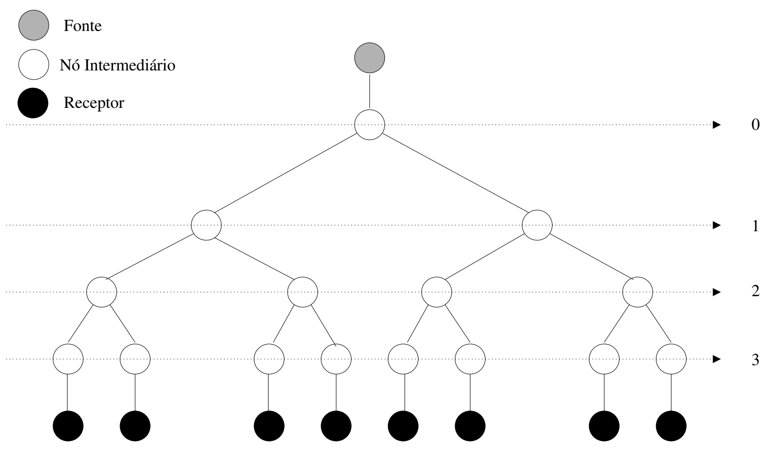


Figura 4.6: Exemplo de topologia em árvore. Árvore básica com altura $h = 3$ e número de filhos por nó $m = 2$.

- todos os enlaces são idênticos;
- os receptores estão sincronizados, ou seja, todos se conectam ao serviço ao mesmo instante e enviam cápsulas de disponibilidade em instantes de tempo iguais.

Sendo assim, as fusões ocorrerão de forma sincronizada em todos os níveis da árvore. Para tais topologias, o índice de filtragem máximo será dado pela seguinte equação:

$$F_{max} = \left(1 - \frac{1}{m^h}\right) 100\% \text{ para } m \geq 1 \text{ e } h \geq 0 \tag{4.3}$$

A figura 4.7 apresenta o comportamento do índice de filtragem máximo de acordo com a altura h da árvore básica para alguns valores de m.

4.5.2 Mecanismos de Fusão Associados

Para avaliar o comportamento do mecanismo de fusão em um ambiente onde mais de um nó pode realizar a fusão, foi utilizada a topologia em árvore apresentada na figura 4.8 onde $h = 3$ e $m = 3$. Pela equação 4.3, o índice de filtragem máximo é próximo de 96,3%. Todos os enlaces possuem banda passante de 10 Mbps. O

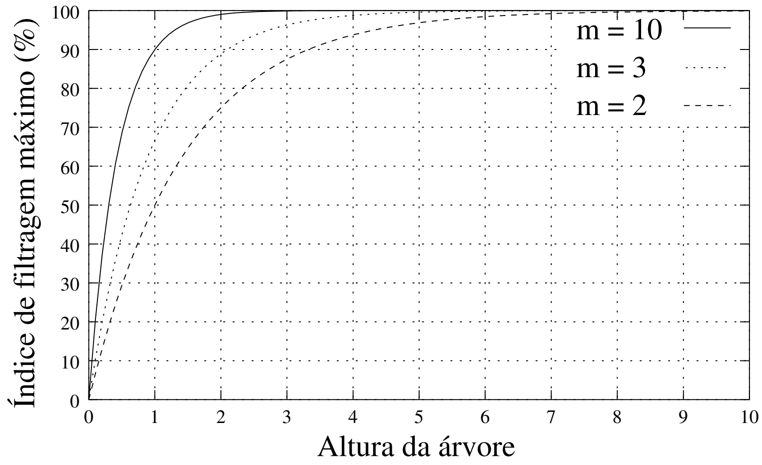


Figura 4.7: Filtragem máxima de acordo com h e m.

atraso em cada enlace foi definido de forma aleatória, uniformemente distribuído no intervalo entre [1, 10] ms. O instante de conexão ao serviço de cada um dos 27 receptores foi escolhido de forma aleatória, uniformemente distribuído no intervalo [0, 10] segundos. O tempo total de simulação foi de 100 segundos. Todos os resultados apresentados possuem um intervalo de confiança de 90%.

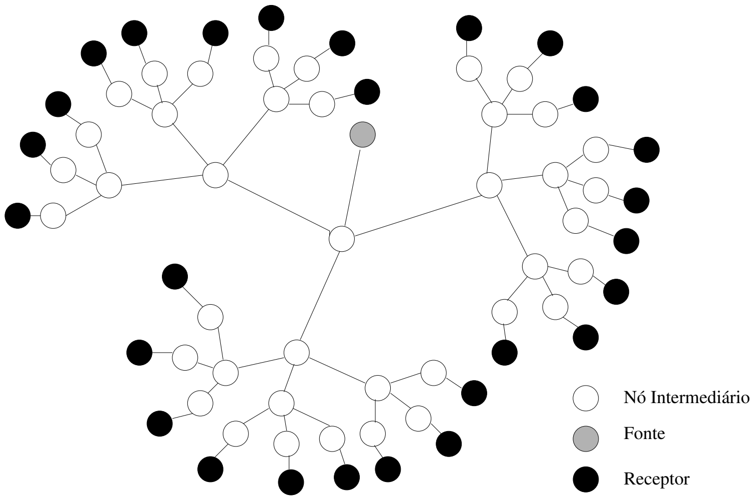


Figura 4.8: Topologia simulada.

Expiração do Temporizador de Fusão a cada 100 ms

A curva apresentada na figura 4.9 mostra os índices de filtragem de cápsulas de disponibilidade alcançados quando se varia o intervalo de envio destas cápsulas. Os índices de filtragem alcançados encontram-se no intervalo entre 91% e 96,15%.

O índice de filtragem máximo ocorre quando o intervalo de envio de cápsulas de disponibilidade é inferior a 25 ms , ou quando o intervalo de envio destas cápsulas é igual ou a metade do tempo de expiração do temporizador de fusão considerado como sendo de 100 ms. Os picos ocorrem para o intervalo de envio de 100 ms e 50 ms devido ao comportamento do mecanismo de fusão nesses pontos. Como apresentado na seção 4.4, nesses pontos, os índices de filtragem são bastante elevados, bem próximos do índice de filtragem máximo. Deste modo, a figura 4.9 mostra que o efeito agregado de vários nós realizando a fusão preserva este comportamento. Para intervalos de envio inferiores a 25 ms existem tantos pontos de sincronismo relacionados à conexão dos receptores ao serviço que acabam fazendo com que o índice de filtragem se aproxime de seu valor máximo.

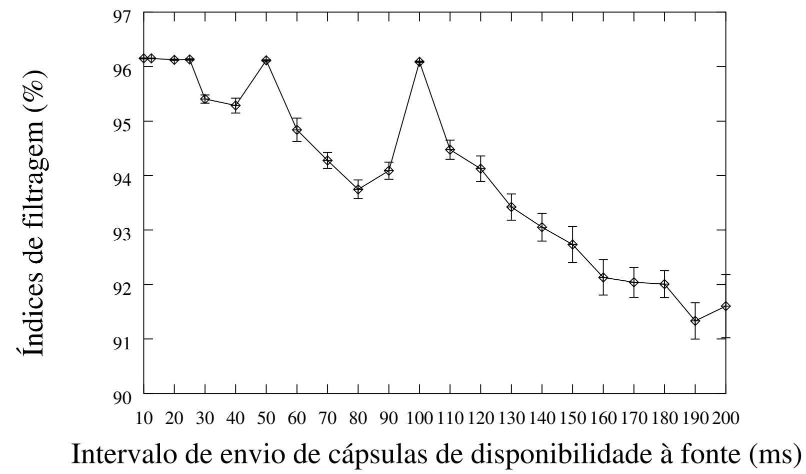


Figura 4.9: Índices de filtragem observados variando-se o intervalo de envio de cápsulas de disponibilidade à fonte.

Na figura 4.10 é apresentado o percentual de fusões que ocorrem na rede de acordo com as duas condições necessárias para haver uma fusão. O resultado mostra que quando o intervalo de envio de cápsulas de disponibilidade é inferior à 25 ms, igual ou metade do tempo de expiração do temporizador de fusão, praticamente todas as fusões ocorrem devido a chegada de uma cápsula de disponibilidade relacionada a cada entrada na tabela auxiliar no *soft-state* do nó ativo. Isso significa que as atualizações nas tabelas ocorrem de forma mais coordenada, fazendo com que a informação que chega à fonte represente melhor a disponibilidade atual de banda passante da árvore multiponto, e não apenas a disponibilidade atual de banda passante de trechos da árvore. Os picos na curva referente à condição 1, nos pontos de 100 ms e 50 ms não significam necessariamente que os receptores estejam sincronizados. Como visto na seção 4.4, nesses pontos, a percentagem de fusões que ocorrem devido a condição 1 é muito elevada, dando a ilusão de sincronismo total entre os receptores.

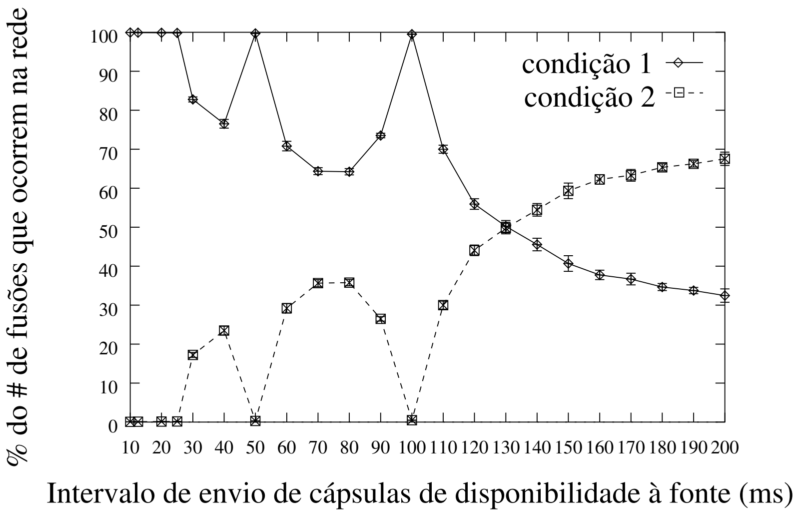


Figura 4.10: Percentagem das fusões que ocorrem na rede por cada condição versus intervalo de envio de cápsulas de disponibilidade à fonte.

A curva da figura 4.11 mostra o comportamento do intervalo médio de chegada de cápsulas de disponibilidade à fonte quando se varia o intervalo de envio destas cápsulas. Este resultado mostra que quando o intervalo de envio de cápsulas de disponibilidade é inferior à 25 ms, igual ou a metade do tempo de expiração do temporizador de fusão, o intervalo médio de chegada de cápsulas de disponibilidade à fonte se mantém praticamente igual ao intervalo de envio destas cápsulas pelos receptores.

Na figura 4.12 é apresentado o percentual de substituições de cápsulas de disponibilidade que estão armazenadas no nó ativo aguardando a fusão. Este resultado é referente ao percentual de cápsulas substituídas no primeiro nó intermediário que possui mais de uma entrada na tabela auxiliar do total de cápsulas de disponibilidade enviadas pelos receptores. Este resultado mostra que menos de 0,06% das cápsulas de disponibilidade enviadas pelos receptores são substituídas quando o intervalo de envio é de 10 ms. Quando o intervalo de envio aumenta, este percentual

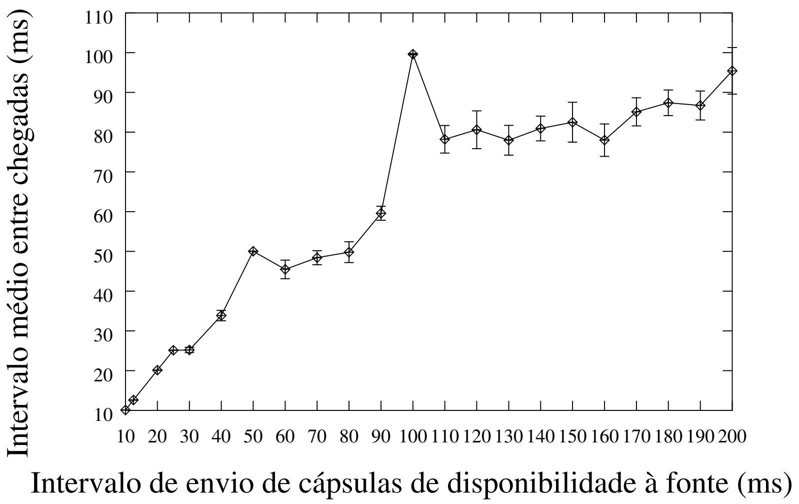


Figura 4.11: Comportamento do intervalo médio entre chegadas de cápsulas de disponibilidade à fonte de acordo com o intervalo de envio dessas cápsulas pelos receptores.

diminui, sendo zero a partir do intervalo de envio de 100 ms.

Expiração do Temporizador de Fusão a cada 150 ms

Os resultados apresentados a seguir são referentes ao intervalo de expiração do temporizador de fusão igual a 150 ms. A figura 4.13 mostra que os índices de filtragem se encontram no intervalo de 93% a 96,16%. Observa-se também que com relação aos resultados para o intervalo de expiração igual a 100 ms, existem mais pontos que se aproximam do índice de filtragem máximo teórico de 96,3%. Tais pontos incluem, por exemplo, os relativos ao intervalo de envio de 150 ms, 75 ms, 50 ms e 30 ms que são submúltiplos do intervalo de expiração do temporizador.

A percentagem das fusões que ocorrem na rede por cada condição, variando-se o intervalo de envio de cápsulas de disponibilidade é apresentada na figura 4.14. O comportamento das curvas apresentadas é similar ao comportamento das curvas da figura 4.10. Mais uma vez, observa-se que para intervalos de envio submúltiplos do intervalo de expiração do temporizador de fusão praticamente todas as fusões ocorrem devido a chegada de uma cápsula de disponibilidade relacionada a cada entrada na tabela auxiliar no *soft-state* do nó ativo.

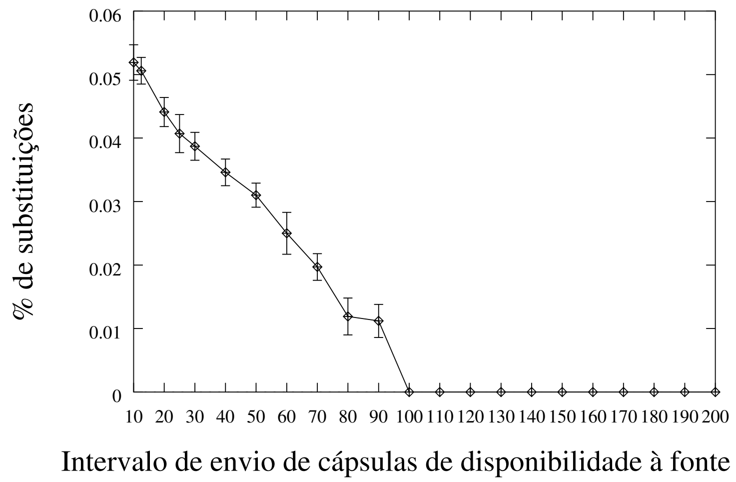


Figura 4.12: Percentagem de substituições de cápsulas de disponibilidade no primeiro nó ativo com mais de uma entrada na tabela auxiliar versus intervalo médio de envio destas cápsulas pelos receptores.

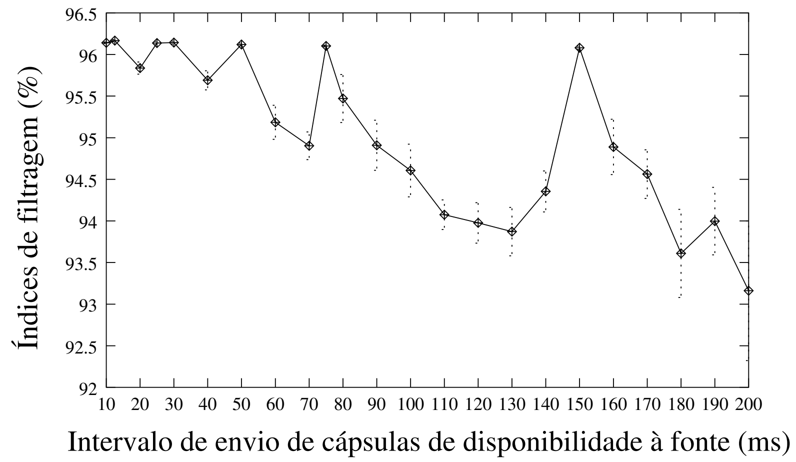


Figura 4.13: Índices de filtragem observados variando-se o intervalo de envio de cápsulas de disponibilidade à fonte.

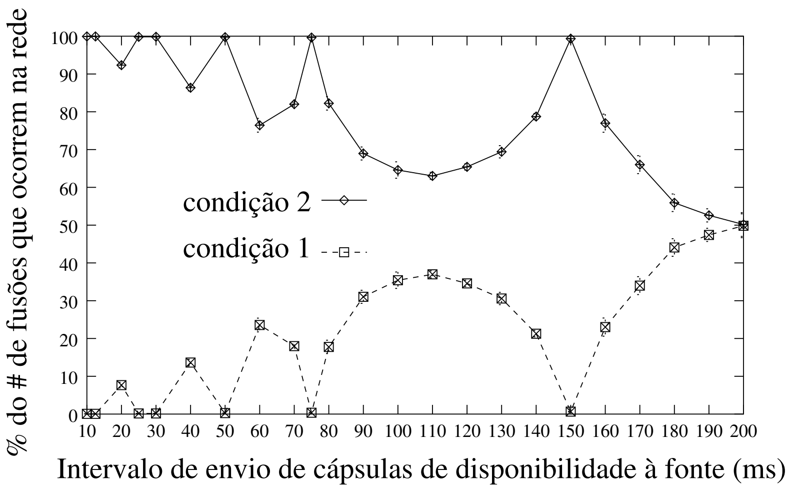


Figura 4.14: Percentagem das fusões que ocorrem na rede por cada condição versus intervalo de envio de cápsulas de disponibilidade à fonte.

A figura 4.15 apresenta o comportamento do intervalo médio de chegada de cápsulas de disponibilidade à fonte. O efeito mais observável é a manutenção do intervalo médio de chegada de cápsulas de disponibilidade à fonte praticamente igual ao intervalo de envio dessas cápsulas pelos receptores quando o intervalo de envio é de 150 ms, 75 ms ou inferior a 50 ms. Já o comportamento da percentagem de substituições apresentado na figura 4.16 mostra que para o intervalo de envio de 10 ms, menos de 0,06% das cápsulas de disponibilidade enviadas pelos receptores continuam sendo substituídas no primeiro nó ativo, que realiza fusões, no caminho para a fonte. A partir do intervalo de envio de 150 ms não há substituições pelos mesmos motivos apresentados no início da seção 4.4.

4.6 Resultados para Topologia *Transit-Stub*

Para a obtenção dos resultados apresentados nesta seção, gerou-se uma topologia *transit-stub* com 5 nós no domínio *transit* e 6 nós por domínio *stub*. Cada nó do domínio *transit* está conectado a somente um domínio *stub*. A soma do número de nós de todos os domínios do tipo *stub* é 30. Desses 30 nós, são utilizados 27 para a conexão dos receptores. Deste modo, a cada um desses 27 nós existe um receptor conectado. A fonte é conectada em um nó pertencente a um dos domínios *stub*.

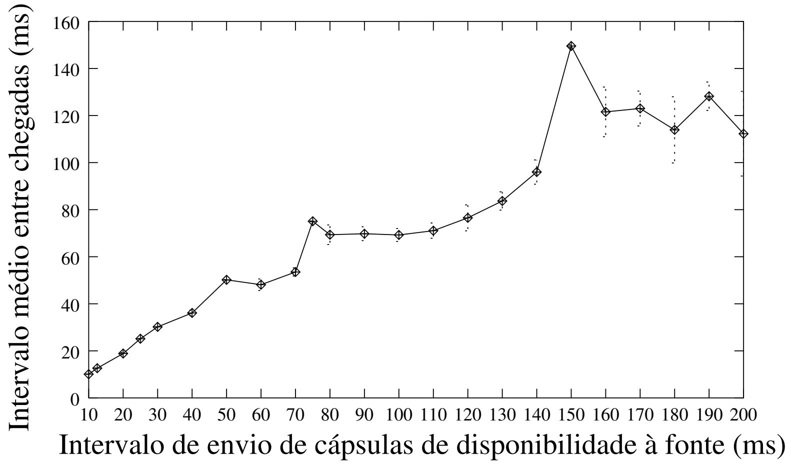


Figura 4.15: Comportamento do intervalo médio entre chegadas de cápsulas de disponibilidade à fonte de acordo com o intervalo de envio dessas cápsulas pelos receptores.

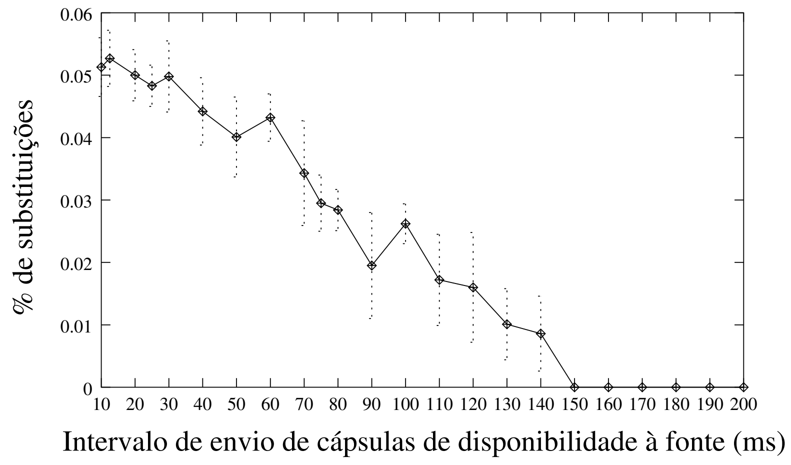


Figura 4.16: Percentagem de substituições de cápsulas de disponibilidade no primeiro nó ativo com mais de uma entrada na tabela auxiliar versus intervalo médio de envio destas cápsulas pelos receptores.

Sendo assim, a topologia estudada possui 63 nós como apresentada na figura 4.17. A fonte está localizada no nó 62. Os nós numerados de 35 até 61 representam os receptores. Mais detalhes sobre a geração da topologia são apresentados no apêndice B.

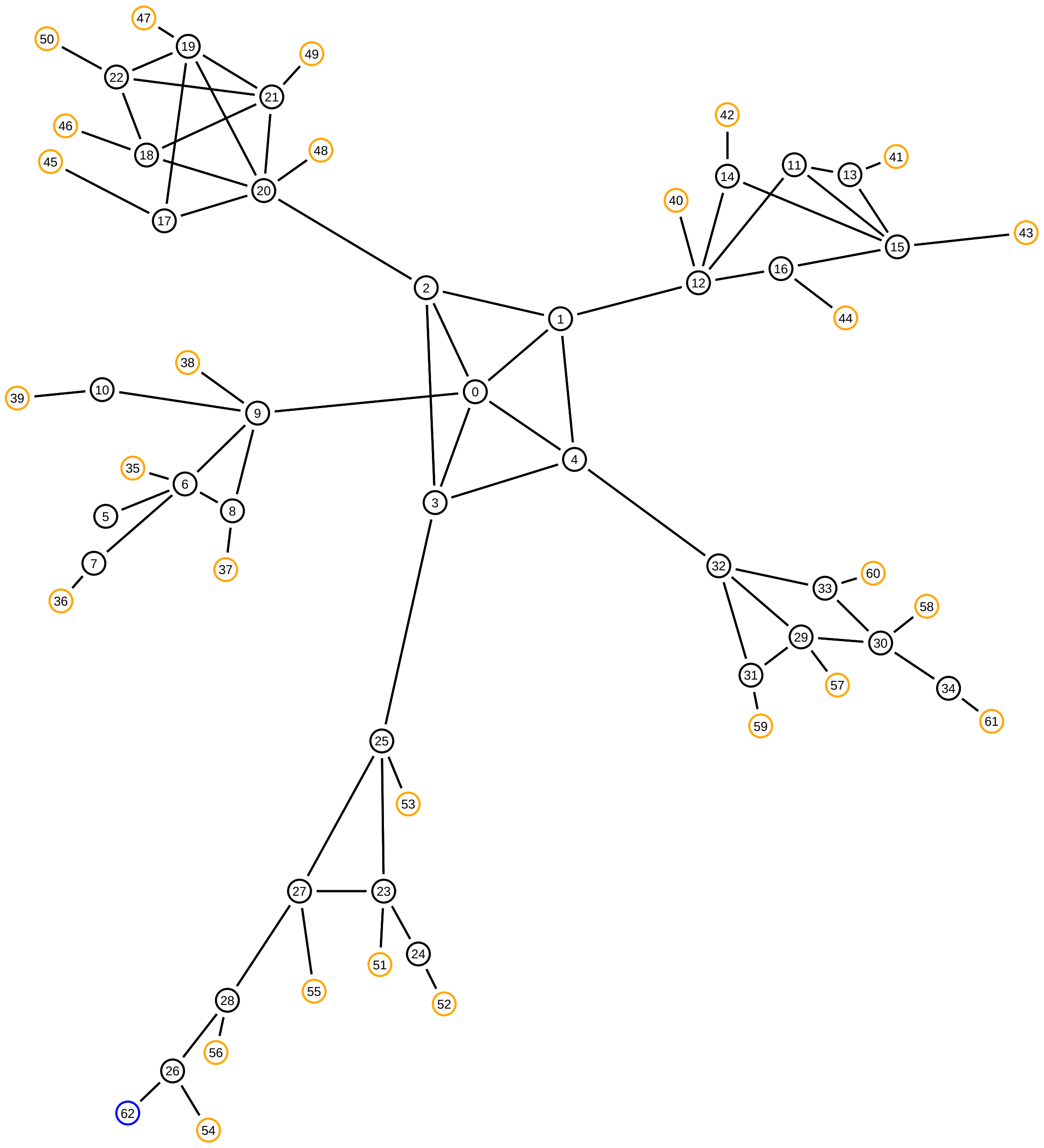


Figura 4.17: Topologia *transit-stub* simulada.

Na topologia, todos os enlaces possuem banda passante de 10 Mbps. O atraso em cada enlace foi definido de forma aleatória, uniformemente distribuído no intervalo entre [1, 10] ms. A expiração do temporizador de fusão ocorre a cada 100 ms.

O instante de conexão de cada receptor ao serviço foi escolhida de forma aleatória, uniformemente distribuída no intervalo [0, 10] segundos. O tempo total de simulação foi de 100 segundos. O intervalo de confiança dos resultados apresentados é de 90%.

As curvas obtidas possuem um comportamento semelhante às curvas apresentadas na seção 4.5.2 para o período de expiração do temporizador de fusão igual a 100 ms. O principal detalhe observado é com relação aos índices de filtragem alcançados que se encontram no intervalo entre 92% e 96,18%. Estes índices são ligeiramente superiores aos apresentados para a topologia em árvore. Isto ocorre porque, em geral, na topologia *transit-stub* existe um número maior de nós, entre um receptor e a fonte, que realizam a fusão de cápsulas de disponibilidade. Deste modo, o fator de filtragem é geralmente mais elevado. Por exemplo, o fator de filtragem dos nós 9, 12, 20 e 32 para o nó 25 é de 4:1, o que contribui significativamente para para um melhor desempenho do mecanismo de fusão. Sendo assim, executando o serviço sobre uma topologia mais próxima às encontradas na Internet, o desempenho do mecanismo de fusão continua sendo elevado.

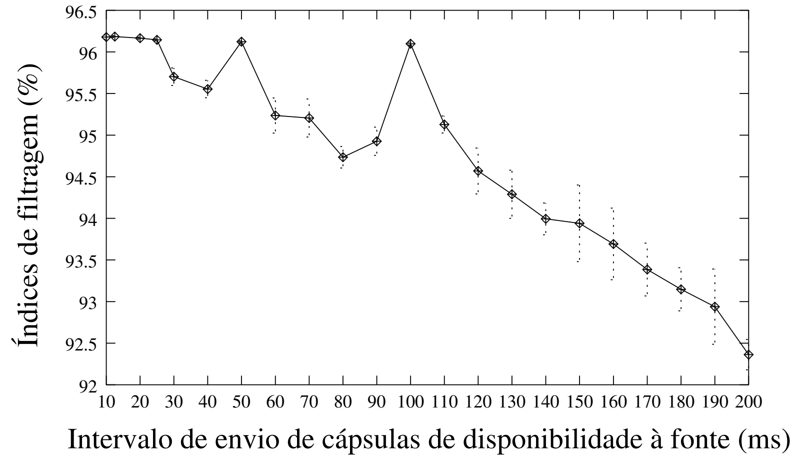


Figura 4.18: Índices de filtragem em função do intervalo de envio.

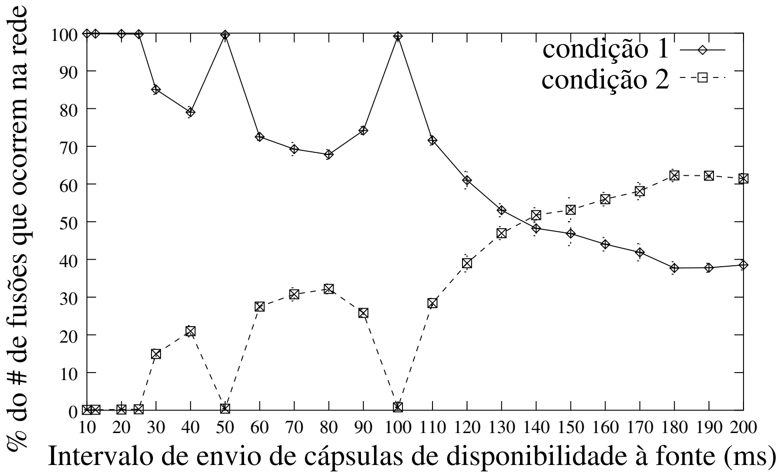


Figura 4.19: Percentagem das fusões que ocorrem na rede por cada condição versus intervalo de envio de cápsulas de disponibilidade à fonte.

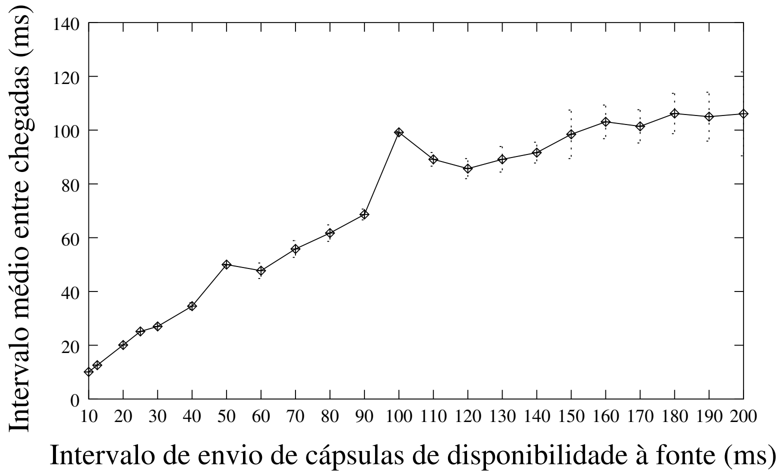


Figura 4.20: Comportamento do intervalo médio entre chegadas de cápsulas de disponibilidade à fonte de acordo com o intervalo de envio dessas cápsulas pelos receptores.

Capítulo 5

Conclusões

As redes ativas proporcionam um novo modo de se visualizar um ambiente de rede ao qual as aplicações podem injetar cápsulas que realizem computações específicas às suas necessidades. Deste modo, o comportamento funcional da rede pode ser modificado e estendido sob o ponto de vista individual de cada aplicação e os diferentes requisitos de QoS das diversas aplicações em andamento podem ser rapidamente acomodados. Além disto, a habilidade de armazenar e acessar estados no nó é extremamente importante pois permite a comunicação e a cooperação entre as cápsulas.

O protocolo ativo foi implementado no simulador de redes *ns*, utilizando uma versão com suporte básico às redes ativas desenvolvida no Grupo de Teleinformática e Automação (GTA) da COPPE/UFRJ. A esta versão do simulador foram adicionados novos controles de estado nos nós, os mecanismos de fusão e de monitoração de banda passante nos enlaces, novos agentes e novas cápsulas. As simulações foram realizadas com dois objetivos principais: estudar o comportamento global do serviço ativo proposto e, em especial, o comportamento do mecanismo de fusão que permitiu avaliar a escalabilidade do serviço proposto. Para isto foram utilizados dois tipos de topologias, uma em árvore e outra mais real, do tipo *transit-stub*, amplamente encontrada na Internet.

O serviço ativo de distribuição de vídeo multiponto proposto neste trabalho explorou os diferentes aspectos de programação dos nós ativos. Deste modo, o intuito de oferecer aos usuários/receptores um serviço flexível, interativo e eficiente foi alcançado. Os usuários estão habilitados a escolher a qualidade do vídeo desejada quantas vezes desejarem e a qualquer instante da sessão. O mecanismo de formação e manutenção da árvore multiponto associado à filtragem das cápsulas de dados de vídeo eliminou a necessidade do estabelecimento de múltiplas árvores multipontos para a distribuição do vídeo em camadas e tornou a quantidade de recursos alocados nos nós independente do número de camadas sendo geradas pela fonte. O mecanismo de criação e atualização das tabelas de disponibilidade associado à filtragem realizada pelas próprias cápsulas de dados de vídeo permitiu aos receptores o recebimento de um número determinado de camadas de vídeo, de acordo com a característica do caminho individual entre a fonte e este receptor e a qualidade Q desejada. Sendo assim, a tentativa de um receptor receber uma qualidade melhor do vídeo não influenciava na qualidade do vídeo recebido por outro receptor. As características acima citadas são inerentes ao serviço proposto e foram observadas nas simulações realizadas.

As simulações também demonstraram que o serviço é bastante robusto, tanto quanto a mudanças de topologia de rede, quanto a manutenção coerente dos estados nos nós ativos. Estes dois fatos são referentes, respectivamente, ao mecanismo de formação e manutenção da árvore multiponto, e à cooperação entre as cápsulas de inscrição/manutenção e as cápsulas de disponibilidade para a eliminação de entradas nas tabelas que não tinham suas estampas de tempo reiniciadas. Adicionalmente, a remoção, pelos nós ativos, de *soft-states* expirados garantiu a liberação de recursos alocados pelo serviço a partir do momento em que tais recursos não fossem mais necessários.

Foram apresentadas análises sobre a quantidade máxima de recursos necessários em um nó para as tabelas auxiliares e de disponibilidade. Estas análises mostraram que com apenas 13k octetos de memória é possível armazenar uma tabela auxiliar com 1000 entradas. A quantidade máxima de recursos para a tabela de disponibilidade é praticamente desprezível, uma vez que ela é igual ao número de interfaces que um nó possui, menos um.

Através de análise, foram apresentados os índices de filtragem máximos para topologias em árvore do tipo estudado neste trabalho. Os índices apresentados são bastante elevados, podendo chegar próximo de 100% dependendo da árvore. Por outro lado, os resultados obtidos por simulação para a topologia em árvore estudada demonstraram que índices de filtragem entre 91% e 96% podem ser obtidos. Para a topologia do tipo *transit-stub*, os índices de filtragem ficaram no intervalo de 92% e 96%. Este desempenho do mecanismo de fusão evidencia a escalabilidade do serviço proposto.

Os resultados obtidos por simulação para as topologias estudadas também demonstraram que escolhendo o intervalo de envio de cápsulas de disponibilidade igual ao intervalo de expiração do temporizador de fusão é possível alcançar um índice de filtragem bastante elevado, manter as atualizações das tabelas de forma mais coordenada e manter a taxa média de chegada de cápsulas de disponibilidade à fonte praticamente igual a taxa de envio dessas cápsulas. A escolha desse intervalo de envio tem o compromisso de não inundar a rede com cápsulas de disponibilidade e manter uma taxa adequada de atualização das tabelas de disponibilidade de modo que a informação que chegue à fonte represente melhor a taxa atual de envio permitida para o vídeo.

Como sugestão para trabalhos futuros, pode-se destacar:

- o estudo da utilização de um mecanismo de reserva de banda passante associado ao serviço;
- a extensão do serviço para suportar comunicações multiponto-a-multiponto;
- a implementação de uma plataforma de testes (*testbed*) para avaliar visualmente a qualidade do vídeo recebido pelos usuários/receptores.

Referências Bibliográficas

[1] Najafi, K., and Leon-Garcia, A. Active video: A novel approach to video distribution. In *Proc. of Second IFIP/IEEE International Conference on Management of Multimedia Networks and Services* (Paris, França, novembro de 1998).

[2] Cheung, S. Y., Ammar, M. H., and Li, X. On the use of destination set group to improve fairness in multicast video distribution. In *IEEE INFOCOM'96* (São Francisco, CA, março de 1996), vol. 2, pp. 553–560.

[3] Li, X., and Ammar, M. Bandwidth control for replicated-stream multicast video distribution. In *5th Internetional Symposium on High Performance Distributed Computing, HPDC-5* (Syracuse, NY, agosto de 1996), pp. 356–363.

[4] Yeadon, N., et al. Filters: QoS support mechanisms for multipeer communications. *IEEE Journal on Selected Areas in Communications 14*, 7 (setembro de 1996), 1245–1262.

[5] Assunção, P. A. A., and Ghanbari, M. Multi-casting of MPEG-2 video with multiple bandwidth constraints. In *7th Int'l Workshop on Packet Video* (1996), pp. 235–238.

[6] Amir, E., McCanne, S., and Zhang, H. An application level video gateway. In *ACM Multimedia'95* (São Francisco, CA, 1995), pp. 255–265.

[7] Turletti, T., and Bolot, J.-C. Issues with multicast video distribution in heterogeneous packet networks. In *6th International Workshop on Packet Video* (Portland, OR, setembro de 1994).

[8] Li, X., Paul, S., and Ammar, M. H. Layered video multicast with retransmission (LVMR): Evaluation of hierarchical rate control. In *IEEE INFOCOM'98* (São Francisco, CA, abril de 1998), pp. 1062–1072.

[9] Wu, L., Sharma, R., and Smith, B. ThinStreams: An architecture for multicasting layered video. In *NOSSDAV'97* (St. Louis, MO, maio de 1997), pp. 173–182.

[10] McCanne, S., Jacobson, V., and Vetterli, M. Receiver-driven layered multicast. In *Proceedings of SIGCOMM'96* (Stanford, CA, agosto de 1996), pp. 117–130.

[11] Shacham, N. Multipoint communication by hierarchically encoded data. In *IEEE INFOCOM'92* (Florença, Itália, maio de 1992), vol. 3, pp. 2107–2114.

[12] Tennenhouse, D. L., et al. A survey of active network research. *IEEE Communications Magazine* (janeiro de 1997), 80–86.

[13] Tennenhouse, D. L., and Wetherall, D. J. Towards an active network architecture. *Computer Communication Review 26*, 12 (abril de 1996).

[14] Calvert, K. L., Bhattacharjee, S., and Sterbenz, J. Directions in active networks. *IEEE Communications Magazine 36*, 10 (outubro de 1998), 72–78.

[15] Biswas, J., et al. The IEEE P1520 standards initiative for programmable network interfaces. *IEEE Communications Magazine 36*, 10 (outubro de 1998), 64–70.

[16] Feldmeier, D., et al. Protocol boosters. *IEEE Journal on Selected Areas in Communications 16*, 3 (abril de 1998), 437–444.

[17] Marcus, W. S., et al. Protocol boosters: Applying programmability to network infrastructures. *IEEE Communications Magazine 36*, 10 (outubro de 1998), 79–83.

[18] Legedza, U., Wetherall, D., and Guttag, J. Improving the performance of distributed applications using active networks. In *IEEE INFOCOM'98* (São Francisco, CA, abril de 1998), pp. 590–599.

[19] Calderón, M., et al. Active network support for multicast applications. *IEEE Network 12*, 3 (junho de 1998), 46–52.

[20] Pham, V. A., and Karmouch, A. Mobile software agents: An overview. *IEEE Communications Magazine 36*, 7 (julho de 1998), 26–37.

[21] Wetherall, D. J., Guttag, J. V., and Tennenhouse, D. L. ANTS: A toolkit for building and dynamically deploying network protocols. In *IEEE OPENARCH'98* (São Francisco, CA, abril de 1998).

[22] Wetherall, D., Legedza, U., and Guttag, J. Introducing new internet services: Why and how. *IEEE Network 12*, 3 (junho de 1998), 12–19.

[23] Wetherall, D., Guttag, J., and Tennenhouse, D. ANTS: Network services without the red tape. *IEEE Computer 32*, 4 (abril de 1999), 42–48.

[24] Smith, J., et al. Activating networks: A progress report. *IEEE Computer 32*, 4 (abril de 1999), 32–41.

[25] Hartman, J. J., et al. Joust: A plataform for liquid software. *IEEE Computer 32*, 4 (abril de 1999), 50–56.

[26] Alexander, D. S., et al. The SwitchWare active network architecture. *IEEE Network 12*, 3 (junho de 1998), 29–36.

[27] Alexander, D. S., et al. A secure active network environment architecture: Realization in SwitchWare. *IEEE Network 12*, 3 (junho de 1998), 37–45.

[28] Hicks, M., Kakkar, P., and Moore, J. T. PLAN: A packet language for active networks. In *International Conference on Functional Programming* (1998), pp. 86–93.

[29] Yemini, Y., and da Silva, S. Towards programmable networks. In *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management* (L'Aquila, Itália, outubro de 1996).

[30] Marcus, W., McAuley, T., and Raleigh, T. Protocol boosters: a kernel-level implementation. In *Proceedings of GLOBECOM'98* (Sydney, Austrália, novembro de 1998), vol. 3, pp. 1619–1623.

[31] Ballardie, T., Francis, P., and Crowcroft, J. Core-based trees (CBT) in a architecture for scalable inter-domain multicast routing. In *Proceedings of SIGCOMM'93* (São Francisco, CA, setembro de 1993), pp. 13–17.

[32] Junqueira, F. P. *Comunicação Multiponto para Estações Móveis em Redes Ativas.* M.Sc. thesis, COPPE/PEE/UFRJ, agosto de 1999.

[33] Fall, K., and Varadhan, K. NS notes and documentation. Tech. rep., The VINT Project, julho de 1999.

[34] Bajaj, S., et al. Improving simulation for network research. Tech. rep., University of Southern California, Departament of Computer Science, março de 1999.

[35] Junqueira, F. P., and Duarte, O. C. M. B. Comunicação multiponto confiável para estações móveis em redes ativas. In *XVII Simpósio Brasileiro de Redes de Computadores SBRC'99* (Salvador, BA, maio de 1999), pp. 389–404.

[36] Zegura, E. W., Calvert, K., and Doar, M. B. Modeling internet topology. *IEEE Communications Magazine 35*, 6 (junho de 1997), 160–163.

[37] Zegura, E. W., Calvert, K., and Donahoo, M. J. A quantitative comparison of graph-based models for internet topologies. *IEEE/ACM Transactions on Networking 5*, 6 (dezembro de 1997), 770–783.

[38] Huitema, C. *Routing in the Internet.* Prentice-Hall, 1995.

Apêndice A

Simulador *ns*

O SIMULADOR *ns* [33, 34] é um simulador de redes dirigido a eventos discretos que são controlados e configurados através da linguagem OTcl (*Object Tool Command Language*). Ele fornece abstrações de nós e enlaces e vem sendo desenvolvido para a simulação e análise dos serviços de rede voltados principalmente para a Internet. O desenvolvimento deste simulador está associado ao projeto *Virtual InterNet Testbed* (VINT), uma colaboração entre pesquisadores do laboratório LBNL (*Lawrence Berkeley National Laboratory*), do laboratório Xerox PARC, do ISI (*Information Sciences Institute*) da USC (*University of Southern California)* e do grupo de pesquisa MASH da UCB (*University of California at Berkeley*).

Para a realização das simulações apresentadas neste trabalho, utilizou-se uma versão modificada deste simulador, com suporte básico às redes ativas, desenvolvida no Grupo de Teleinformática e Automação (GTA) da COPPE/UFRJ [32, 35]. Contudo, para implementar o serviço ativo de distribuição de vídeo multiponto proposto, parte do trabalho consistiu na extensão dos módulos existentes e no desenvolvimento de novos módulos.

A.1 A estrutura Básica do *ns*

O *ns* foi escrito nas linguagens C++ e OTcl. O núcleo do simulador é implementado em C++, uma linguagem compilada que permite obter um melhor desempenho quando comparada com uma linguagem interpretada. As simulações são configuradas pelos *scripts* OTcl e descrevem, por exemplo, a topologia de rede, incluindo as características dos nós e da interconexão física entre eles; os protocolos utilizados; modelos de geração de tráfego; e a modelagem de perdas de pacotes associadas à falhas em nós e enlaces da rede. As topologias para a simulação no *ns* podem ser geradas manualmente ou através do gerador de topologias GT-ITM (*Georgia Tech - Internetwork Topology Models*). Mais detalhes sobre o GT-ITM são apresentados no apêndice B.

A figura A.1 ilustra a estrutura dos nós no *ns*. Esta estrutura é composta por agentes, um ponto de entrada no nó, um classificador de endereços e um classificador de portas. No *ns*, um agente é um gerador e consumidor de pacotes que implementa um determinado tipo de protocolo. Alguns exemplos são os agentes TCP (*Transmission Control Protocol*), os agentes UDP (*User Datagram Protocol*) e os agentes nulos. Estes últimos, somente consomem pacotes. Um pacote gerado por um agente é entregue inicialmente ao nó ao qual o agente está ligado, através do ponto de entrada que recebe não somente os pacotes dos agentes, mas também os pacotes destinados àquele nó. Após passar pelo ponto de entrada do nó, o pacote é recebido pelo classificador de endereços que verifica se o pacote deve ser entregue a um agente pertencente ao nó ou transmitido para um enlace de saída. Caso o pacote seja destinado a um agente conectado ao próprio nó, o pacote é repassado para o classificador de porta, que de acordo com o endereço de destino entrega o pacote ao devido agente.

Cada agente possui um endereço de 16 bits, no qual 8 bits correspondem ao identificador do nó e os outros 8 bits identificam o agente no nó. Os classificadores de endereço resolvem um endereço com base no tipo de estratégia de roteamento utilizada, que pode ser: estática, por sessão e dinâmica. As estratégias estática e por sessão usam o algoritmo SPF (*Shortest Path First*) de Dijkstra. Enquanto na

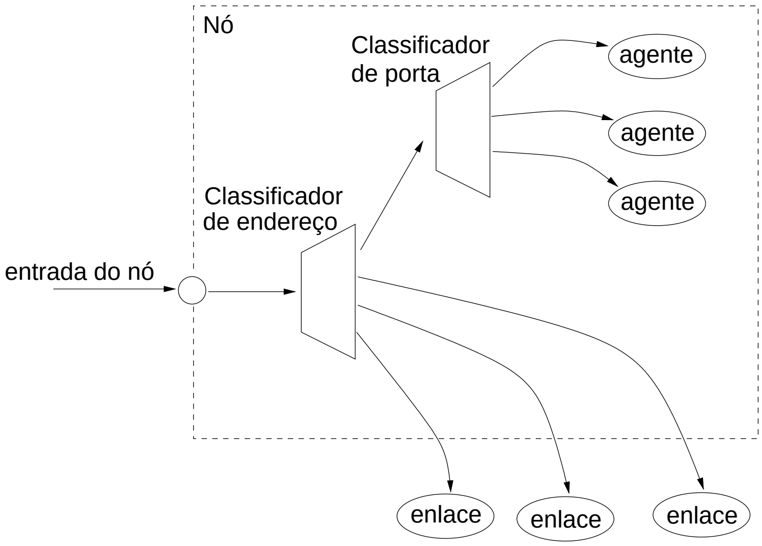


Figura A.1: Estrutura dos nós no *ns*.

estratégia estática este algoritmo é executado apenas uma vez no início da simulação, na estratégia por sessão, executa-se este algoritmo sempre que ocorrer uma mudança de topologia. Já na estratégia dinâmica, a algoritmo utilizado é o vetor-distância de Bellman-Ford. Neste algoritmo, são enviadas periodicamente atualizações da tabela de rotas para todos os nós vizinhos. Deste modo, as atualizações periódicas refletem de forma dinâmica alterações na topologia de rede [38, 33].

O escalonamento de eventos no *ns* pode ser feito utilizando-se um dos quatro tipos de escalonadores de eventos existentes: lista simplesmente encadeada, *heap*, *calendar queue* e um outro denominado "tempo real". As simulações apresentadas neste trabalho utilizaram o escalonador *calendar queue* que possui uma estrutura de dados similar a um calendário anual, em que eventos em um mesmo mês/dia de anos distintos podem ser mapeados em um único dia. A função do escalonador é selecionar o próximo evento, executá-lo até o fim e retornar para executar o próximo evento. Eventos escalonados para executarem ao mesmo tempo são ordenados e executados em série.

A.2 Implementação do Serviço Ativo

Para a implementação do serviço proposto sobre a versão ativa do *ns* foram desenvolvidos dois novos agentes, cinco novas cápsulas, o mecanismo de fusão, mecanismos adicionais de controle de estados, e um mecanismo de monitoração de banda passante nos enlaces. As cápsulas que implementam o protocolo ativo proposto são apresentadas na seção 3.2.

Os dois novos agentes implementados são, respectivamente, o agente que executa o protocolo dos receptores do serviço e outro agente que atua como uma fonte de vídeo em camadas com controle dinâmico de QoS. Os agentes implementados herdam métodos da classe *ActiveAgent* disponibilizada na versão ativa do *ns*. As cápsulas enviadas pelos agentes possuem um cabeçalho fixo contendo os 8 parâmetros apresentados na tabela A.1. Os 6 primeiros parâmetros já faziam parte da versão para redes ativas. Deste modo, foram introduzidos dois novos campos e as funcionalidades dos outros campos foram utilizadas de acordo com as necessidades do serviço.

Tabela A.1: Campos do cabeçalho de uma cápsula.

caps_ID	Identificador do tipo de cápsula
previous_	Endereço do nó anterior
source_	Endereço da fonte da cápsula
resources_	Quantidade de recursos a ser utilizada
fid_	Identificador do fluxo
params_	Parâmetros para os procedimentos
TX_	Taxa cumulativa da camada a que pertence a cápsula de dados de vídeo
layer_	Camada a que pertence a cápsula de dados de vídeo

O mecanismo de fusão foi implementado no classificador de endereços do nó. Deste modo, as cápsulas de disponibilidade que chegam de diferentes origens são armazenadas por este classificador para a posterior realização da fusão. A cápsula resultante de uma fusão se utiliza dos métodos de resolução de endereços deste classificador com o intuito de seguir para o próximo nó no caminho para a fonte. Mecanismos adicionais foram implementados para permitir que o classificador de endereços controlasse os estados relacionados ao mecanismo de fusão.

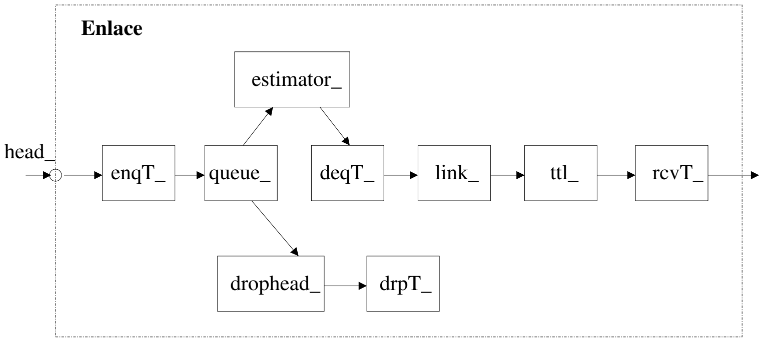


Figura A.2: Módulos que compõem um enlace no *ns*.

A implementação do mecanismo de monitoração de banda passante nos enlaces foi realizada adicionando-se um elemento estimador, referenciado por *estimator_*, entre os elementos *queue* e *deqT*, referenciados respectivamente por *queue_* e *deqT_*. Estes elementos fazem parte de um enlace no *ns*. A figura A.2 mostra a organização dos elementos que compõem um enlace. O significado das respectivas variáveis que referenciam a cada elemento é apresentado a seguir:

- *head_* - Ponto de entrada de cápsulas no enlace;
- *queue_* - Referência para a fila de cápsulas. Cada enlace possui uma fila;
- *estimator_* - Referência ao estimador EWMA que estima a banda passante agregada dos fluxos que transitam no enlace ou a banda passante consumida por fluxos específicos;
- *link_* - Referência ao elemento que modela o enlace, incluindo características de atraso e banda passante;
- *ttl_* - Referência ao elemento que manipula o TTL das cápsulas;

- *drophead_* - Referência ao objeto que representa o início da fila que processa cápsulas perdidas no enlace;
- *enqT_* - Referência ao elemento que gera o relatório de cápsulas que entram na fila referenciada por *queue_*;
- *deqT_* - Referência ao elemento que gera o relatório de cápsulas que saem da fila referenciada por *queue_*;
- *drpT_* - Referência ao elemento que gera o relatório de cápsulas descartadas da fila referenciada por *queue_*;
- *rcvT_* - Referência ao elemento que gera relatórios de cápsulas recebidas pelo próximo nó.

O *ns* proporciona elementos que permitem a geração de relatórios de simulação. Para a geração de relatórios específicos ao serviço proposto, alguns desses elementos foram modificados. Basicamente, foram utilizados relatórios dos eventos relacionados aos receptores, à fonte, aos classificadores de endereço, e ao tráfego de pacotes entre os nós ativos.

A seguir, são apresentados alguns exemplos dos relatórios gerados.

```
# RELATÓRIO DE EVENTOS DOS RECEPTORES
# <INSTANTE DO EVENTO> <IDENTIFICAÇÃO DO NÓ> <EVENTO>
# EVENTOS:
# JOIN - Tentando conexão
# JOIN_SUCCESS - Conexão realizada
# HIGHEST_RECEIVED_LAYER - Camada mais alta sendo recebida
# LEAVE - Desconexão do serviço. Dados informados: <camada mais alta recebida>
# <# de cápsulas recebidas> <perdas computadas>
# FINAL_STATS - estatísticas após recebimento de uma cápsula de fim de sessão
# Dados informados: <camada mais alta recebida> <# de cápsulas recebidas>
# <perdas computadas>
# RECV_EOS - Cápsula de fim de sessão recebida
```

```
1.056748 3 JOIN
1.076153 3 JOIN_SUCCESS
1.076953 3 HIGHEST_RECEIVED_LAYER 2
1.077753 3 HIGHEST_RECEIVED_LAYER 3
1.078553 3 HIGHEST_RECEIVED_LAYER 4
1.079353 3 HIGHEST_RECEIVED_LAYER 5
1.080153 3 HIGHEST_RECEIVED_LAYER 6
1.080953 3 HIGHEST_RECEIVED_LAYER 7
1.492298 2 JOIN
1.502597 2 JOIN_SUCCESS
1.502597 2 HIGHEST_RECEIVED_LAYER 6
1.504239 2 HIGHEST_RECEIVED_LAYER 7
1.698369 1 JOIN
1.708849 1 JOIN_SUCCESS
1.708849 1 HIGHEST_RECEIVED_LAYER 7
1.760999 4 JOIN
1.778101 4 JOIN_SUCCESS
1.778101 4 HIGHEST_RECEIVED_LAYER 7
4.000000 1 LEAVE 7 1162 0
10.010708 3 RECV_EOS
10.010708 3 FINAL_STATS 7 4543 0
10.010708 2 RECV_EOS
10.010708 2 FINAL_STATS 7 4325 0
10.010708 4 RECV_EOS
10.010708 4 FINAL_STATS 7 4184 0
10.019006 3 RECV_EOS
10.019006 3 FINAL_STATS 7 4543 0
10.019006 2 RECV_EOS
10.019006 2 FINAL_STATS 7 4325 0
10.019006 4 RECV_EOS
```

```
10.019006 4 FINAL_STATS 7 4184 0


# TRECHO DE UM RELATÓRIO DE TRÁFEGO DE PACOTES
# <evento> <instante do evento> <nó origem> <nó destino> capsule
# <tamanho da cápsula> ------- <identificação do fluxo> <tipo de cápsula>
# <# de seqüência para cada tipo de cápsula> <# de seqüência global>
#
# EVENTOS:
#   + : Cápsula na fila para a transmissão
#   - : Cápsula transmitida
#   r : Cápsula recebida

r 1.0629 4 1 capsule 64 ------- 1 41 1 -1 0
+ 1.0629 1 0 capsule 64 ------- 1 41 1 -1 0
- 1.0629 1 0 capsule 64 ------- 1 41 1 -1 0
r 1.06495 1 0 capsule 64 ------- 1 41 1 -1 0
+ 1.06495 0 1 capsule 1000 ------- 1 20 1 0 1
- 1.06495 0 1 capsule 1000 ------- 1 20 1 0 1
+ 1.06495 0 1 capsule 1000 ------- 1 20 2 0 2
+ 1.06495 0 1 capsule 1000 ------- 1 20 3 0 3
+ 1.06495 0 1 capsule 1000 ------- 1 20 4 0 4
+ 1.06495 0 1 capsule 1000 ------- 1 20 5 0 5
+ 1.06495 0 1 capsule 1000 ------- 1 20 6 0 6
+ 1.06495 0 1 capsule 1000 ------- 1 20 7 0 7


# RELATÓRIO DE ESTATÍSTICAS DA FONTE
#

----------Estatísticas Finais da Fonte----------
```

```
No. total de cápsulas recebidas: 34

No. de cápsulas de disponibilidade recebidas: 31

No. de cápsulas de inscrição/manutenção recebidas: 3

Camada mais alta enviada: 7

-----------------------------------------------
```

Apêndice B

Parâmetros para o GT-ITM

O GERADOR de topologias GT-ITM (*Georgia Tech - Internetwork Topology Model*) pode ser utilizado para gerar topologias planas, hierárquicas de nível n e *transit-stub*. Estas topologias são baseadas no modelo padrão de geração de grafos que distribui os vértices em posições aleatórias em um plano, havendo uma probabilidade p de uma aresta ser adicionada entre um par de vértices. Este modelo é denominado puramente aleatório. Tal modelo de geração de grafos não reflete a estrutura das redes reais, sendo por isso, propostos outros modelos que alteram a função de probabilidade de um vértice para representarem de forma mais fiel a estrutura dessas redes [36, 37].

As topologias *transit-stub* são grafos hierárquicos construídos através da composição de domínios *transit* e *stub*. Primeiramente, é construído um grafo aleatório conexo com cada nó representado um domínio *transit* completo. Em seguida, cada nó é substituído por outro grafo aleatório conexo que representa o *backbone* de um domínio *transit*. Para cada nó deste domínio é gerado um número de grafos aleatórios conexos que representam domínios do tipo *stub* conectados ao nó. E por último, são feitas ligações adicionais entre nós de um domínio *transit* e um domínio *stub*, ou de dois domínios *stub* diferentes.

O *script* a seguir apresenta os dados utilizados para a geração da topologia *transit-stub* simulada. A primeira linha após os comentários indica, respectivamente, o tipo de grafo a ser gerado, o número de grafos e a semente a ser utilizada para a geração dos números aleatórios. A linha *1 0 0* significa que existirá somente um domínio *stub* conectado a um nó do domínio *transit*, não devendo haver arestas adicionais *transit-stub* ou *stub-stub*. A linha com os parâmetros *1 20 3 0.5* fornece os valores para a geração dos domínios *transit*. O primeiro parâmetro informa a existência de somente um domínio *transit*. Os outros valores, comuns para as três últimas linhas, indicam respectivamente a escala, o modelo de probabilidade de arestas a ser utilizado, e um parâmetro para este modelo. A escala é correspondente ao número de sub-divisões em cada nível da hierarquia. Este número serve para o cálculo do tamanho das arestas. O parâmetro para modelo de probabilidades de arestas indica a probabilidade de haver um enlace conectando dois nós de um mesmo domínio. As duas linhas seguintes são referentes, respectivamente, às características dos domínios do tipo *transit* e do tipo *stub*. Deste modo, a linha *5 20 2 0.6* indica que cada domínio *transit* possuirá, em média, 5 nós, existindo uma probabilidade de 0,6% de dois nós estarem conectados. A linha *6 10 3 0.4* indica que todo domínio *stub* terá, em média, 6 nós, havendo uma probabilidade de 0,4% de dois nós estarem conectados.

```
# Comments :
# <method keyword> <number of graphs> [<initial seed>]
# <# stubs/trans node> <#rand. t-s edges> <#rand. s-s edges>
# <n> <scale> <edgemethod> <alpha> [<beta>] [<gamma>]
# number of nodes = 1x5x(1+1x6) = 35

ts 1 47
1 0 0
1 20 3 0.5
5 20 3 0.6
6 10 3 0.4
```

A partir desta topologia, cuja soma do número de nós de todos os domínios do tipo *stub* é 30, foi gerada a topologia estudada. Desses 30 nós, são utilizados 27 para a conexão dos receptores. Deste modo, a cada um desses 27 nós existe um receptor conectado. A fonte é conectada em um nó pertencente a um dos domínios *stub*. Sendo assim, a topologia estudada possui 63 nós