Comunicação Multiponto Confiável para Estações Móveis em Redes Ativas

por

Flavio Paiva Junqueira

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

COMUNICAÇÃO MULTIPONTO CONFIÁVEL PARA ESTAÇÕES MÓVEIS EM REDES ATIVAS

Flavio Paiva Junqueira

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. Orlando Gomes Loques Filho, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

AGOSTO DE 1999

JUNQUEIRA, FLAVIO PAIVA

Comunicação Multiponto Confiável para Estações Móveis em Redes Ativas [Rio de Janeiro] 1999

XIV, 84 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 1999)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Protocolos em Redes Ativas
2. Mobilidade
3. Comunicação Multiponto Confiável

I. COPPE/UFRJ   II. Título (Série)

Aos meus familiares e amigos.

Agradecimentos

À Renata, pela paciência.

À minha família.

Ao meu orientador, Professor Otto Carlos Muniz Bandeira Duarte, pela dedicação e apoio.

Ao professor José Ferreira de Rezende, pelo apoio e pelos conselhos.

Ao professor Orlando Loques, membro da banca examinadora.

Aos amigos do GTA: Rubi, Ronaldo, APaulo, Artur, Vidal, Emerson PS, Belem, Hermes, LuisH, Antguria, Andreamb, Adriana, Daniel, Fred, Bill, Alexszt, Marcos Tadeu, Sílvia, Marcial, Jonny, Abdalla, Célio, Faerman, Ildemar, Prof. Mauros, Iniciações Científica (Bernardo, Tadeu, Doc, Eric, Pedro e Fred), Guilherme, Lemos, Jucele, Ezequiel, Duboc e Miranda.

Aos professores Valmir, Paulo Aguiar, Aloysio, Leão, Lanza e Gelson.

Aos amigos do GEP: Evandro, Tony, Rosas, Marcelo, Pedro e Luis Oscar.

Aos funcionários: Wilson, Wanderlei, Paulo Roberto, Bia, Solange e Evelyn.

Aos amigos e professores da Graduação.

À CAPES, pelo apoio financeiro.

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

COMUNICAÇÃO MULTIPONTO CONFIÁVEL PARA ESTAÇÕES MÓVEIS EM REDES ATIVAS

Flavio Paiva Junqueira

Agosto/1999

Orientador: Otto Carlos Muniz Bandeira Duarte
Programa: Engenharia Elétrica

O paradigma de redes ativas propõe que o serviço de rede seja programável e, como conseqüência, capaz de ser moldado para atender aos mais diferentes requisitos das aplicações. Assim, computações necessárias às aplicações são inseridas nos nós para alterar o comportamento funcional da rede, ou seja, o serviço se adequa aos requisitos através de procedimentos inseridos nos nós pelas próprias aplicações.

Este trabalho propõe um serviço em redes ativas de comunicação multiponto confiável e escalável para estações móveis. O serviço utiliza algumas facilidades disponíveis nos nós ativos para compensar os principais problemas tanto da comunicação multiponto quanto da mobilidade, que são: as altas taxas de erro nos enlaces sem fio, as perdas devido ao movimento intercelular, a localização das estações móveis para o roteamento e a implosão de cápsulas. Entre as funções realizadas pelos nós ativos têm-se a marcação de caminhos por estados flexíveis (*soft-state*), a filtragem de cápsulas e a utilização de caches para retransmissões. A alocação de espaço de armazenamento para o cache é feita através de um mecanismo proposto, chamado de cache adaptativo, que é capaz de variar o tamanho deste espaço de acordo com a incidência de pedidos de retransmissão.

Simulações foram realizadas utilizando topologias em árvore e topologias similares às encontradas na Internet. Os resultados obtidos mostram que o serviço permite elevadas taxas de movimento intercelular, mesmo para um grande número de usuários. Além disto, constatou-se que a utilização do cache adaptativo possibilita uma economia de recursos dos nós quando há banda passante suficiente e um aumento no desempenho do serviço quando há escassez de banda passante.

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

RELIABLE MULTICAST COMMUNICATION FOR MOBILE HOSTS IN ACTIVE NETWORKS

Flavio Paiva Junqueira

August/1999

Advisor: Otto Carlos Muniz Bandeira Duarte
Department: Electrical Engineering

The active networking paradigm proposes a programmable network interface, in order to provide a service capable of accomodating the great variety of application requirements. Thus, applications inject specific computations into the network to modify its functional behavior, i.e., to modify functionalities seen by users. In this manner, applications have the opportunity to satisfy their requirements, and to experience better performance.

This work proposes a scalable and reliable multicast service for mobile hosts in cellular environments based on active networking. This service uses some facilities provided by active nodes to compensate the main mobility and multicast communication problems, as: high bit-error rates, losses due to handoffs, host position for routing service, and capsule implosion. Among active node functions are path indication using soft-state, capsule filtering and cached retransmissions. To allocate memory to cache, an adaptive mechanism was proposed to maintain its size in accordance with retransmission incidence.

Service evaluation was accomplished through simulations using tree topologies, as well as random topologies that represent Internet real ones. To carry out simulations, extensions to the *ns* (Network Simulator) were implemented, including active networking support, and service specific procedures. Simulation results show that this service performs well for a large number of users along with high handoff frequencies. Moreover, adaptive cache allows memory saving whether enough bandwidth is available, and better performance whether available bandwidth is not enough.

Palavras-chave

Redes Ativas

Redes Celulares

Mobilidade

Confiabilidade

Escalabilidade

Acrônimos

QoS : Qualidade de Serviço (*Quality of Service*);

ERB : Estação Rádio-base (*Base Station*);

SO : Sistema Operacional (*Operating System*);

UCP : Unidade Central de Processamento (*Central Processing Unit*);

CBR : Taxa Constante de Bits (*Constant Bit Rate*);

IP : *Internet Protocol*;

TCP : *Transmission Control Protocol*;

UDP : *User Datagram Protocol*;

ACK : Reconhecimento Positivo (*ACKnowledgement*);

NAK : Reconhecimento Negativo (*Negative AcKnowledgement*);

NAC : Nó Ativo Celular;

ANTS : *Active Node Transfer System*;

RTT : *Round-trip Time*;

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

COM o sucesso da Internet, pessoas com os mais variados perfis e expectativas são hoje usuárias desta grande rede mundial, interessadas principalmente no livre acesso à informação e na comunicação de dados global. Assim, freqüentemente surgem novas idéias de aplicações para atender às necessidades de comunicação de um determinado tipo de usuário. Como exemplo de novas classes de aplicações que vêm surgindo, pode-se mencionar as que envolvem o uso de múltiplas mídias como áudio, vídeo, texto etc, denominadas aplicações multimídias, e as que envolvem a participação simultânea de um grupo de usuários, denominadas aplicações multipontos ou multidestinatárias. Estas aplicações, em geral, possuem variados requisitos de qualidade de serviço (QoS - *Quality of Service*) que exigem diferentes características da rede em relação à banda passante, ao atraso e à variação do atraso (*jitter*), e que não são atendidas pelo serviço de melhor esforço (*best-effort*) oferecido atualmente na Internet [1]. Além disto, as aplicações são freqüentemente beneficiadas pela implantação de funcionalidades específicas nos nós da rede, o que é dificultado pelo processo custoso e demorado adotado atualmente para a alteração do serviço de rede [2].

Visando uma rede flexível e adequada às necessidades de cada aplicação, o paradigma de redes ativas propõe que o serviço de rede seja programável, ao invés de definir um serviço diferente para cada aplicação ou classe de aplicações. Para isto, as redes ativas permitem a injeção e a execução de programas específicos nos nós do interior da rede com dois objetivos básicos: possibilitar serviços adequados às aplicações, aumentando a flexibilidade, e a rápida implantação de novos serviços. Desta forma, as redes assumem um papel “ativo” na execução das aplicações, que por sua vez não estariam mais restritas a computações nas entidades de comunicação de extremidade, fornecendo uma arquitetura aberta capaz não só de prover QoS para as aplicações, mas também de permitir a alteração do comportamento funcional do serviço de rede [3, 4, 5].

Entre as aplicações beneficiadas pela adequação do serviço de rede estão as que necessitam ou desejam o suporte à mobilidade, isto é, aplicações que permitem a movimentação dos usuários durante as suas execuções. Diferentes das estações fixas que, geralmente, estão conectadas através de meios físicos com baixas taxas de erro, as estações móveis ligadas através de enlaces sem fio enfrentam altas taxas de perda devido às interferências nos enlaces e a necessidade de atualização constante das informações de localização, utilizadas para o roteamento. Um outro problema surge quando supõe-se que as estações móveis encontram-se em redes celulares. Uma rede celular divide uma área geográfica em células, estando cada célula sob o domínio de um roteador que possui interfaces aéreas para comunicação sem fio com as estações móveis, denominado estação rádio-base (ERB). Conforme as estações móveis se movimentam ocorrem trocas de célula, denominadas handoffs, ocasionando a perda dos dados em trânsito durante as trocas [6, 7, 8].

Se a aplicação visada for ainda multidestinatária, exigindo confiabilidade, como por exemplo *whiteboard* compartilhado, simulação interativa distribuída (DIS - *Distributed Interactive Simulation*) e computação distribuída, surgem outros problemas além dos colocados para a mobilidade [9]. Considerando-se um grupo multiponto com muitos usuários, deve-se evitar que a utilização de recursos da rede e das estações participantes cresça na mesma proporção do número de usuários para evitar a sobrecarga de processamento no transmissor e o uso excessivo de banda passante.

Para a distribuição dos dados em uma comunicação multiponto é necessária a formação de uma árvore, sendo os participantes as folhas desta árvore. Se os usuários forem estações móveis em redes celulares, a atualização das árvores multipontos deve ser rápida o suficiente para permitir altas taxas de movimento intercelular. Desta forma, a atualização realizada nos nós mais próximos da localização corrente da estação seria mais rápida e, portanto, mais eficiente [7]. Outro requisito diz respeito às retransmissões, que não devem ser realizadas para todo o grupo multiponto, pois como as taxas de perda são elevadas, seria grande o desperdício de banda passante, assumindo que grande parte das perdas ocorrem somente para um sub-conjunto de um grupo multiponto. Além disto, a responsabilidade pelas retransmissões deve ser distribuída, para evitar uma sobrecarga no transmissor pela manipulação de um grande número de pedidos de retransmissão quando o número de usuários é grande. Assim, retransmissões realizadas próximas à localização das estações móveis seriam mais eficientes, por serem mais rápidas e somente para o sub-conjunto dos usuários que efetivamente sofreu a perda [10].

As seções seguintes deste capítulo descrevem com mais detalhes os problemas relacionados à comunicação multiponto e à mobilidade na Internet, apresentando também algumas propostas que serviram de base para o trabalho desenvolvido nos próximos capítulos desta tese.

1.1 Comunicação Multiponto Confiável na Internet

As duas abordagens mais populares para protocolos de comunicação multipontos confiáveis são a orientada ao transmissor e a orientada ao receptor [10, 11, 9]. Na abordagem orientada ao transmissor, o estado de todos os receptores é mantido pelo transmissor, que guarda o que foi difundido aos receptores e o que foi recebido corretamente por cada receptor. Cada transmissão (ou retransmissão) realizada pela fonte é difundida para todos os integrantes do grupo multiponto e a recepção correta deve ser confirmada por cada um dos receptores através do envio de um reconhecimento positivo (ACK) para o transmissor. Este procedimento implica em uma sobrecarga no transmissor devido à grande quantidade de reconhecimentos positivos que devem ser tratados. Este fenômeno é conhecido como implosão de ACKs.

Na abordagem orientada ao receptor, o transmissor, por sua vez, difunde os pacotes, para o grupo multiponto e cada receptor envia reconhecimentos negativos (NAKs) quando detectarem que ocorreram perdas ou dados foram recebidos incorretamente. Como o número de pacotes não recebidos ou recebidos incorretamente é, normalmente, bem menor que o de recebidos corretamente, a sobrecarga de processamento no transmissor e o número de pacotes de reconhecimento diminui bastante. No entanto, se um pacote não é recebido corretamente por um grande número de receptores, o fenômeno de implosão de NAKs ocorre. A implosão de reconhecimentos é o grande limitante da escalabilidade de um protocolo multiponto confiável.

Uma análise comparativa realizada por Pingali *et al.* [9] demonstrou quantitativamente a superioridade da abordagem orientada ao receptor quando um grande número de usuários está envolvido. Esta análise mostrou também que para aplicações ponto-a-multiponto é preferível a utilização de um protocolo orientado ao receptor como foi apresentado anteriormente, em que um NAK é enviado para cada perda ou erro detectado. Por outro lado, se a aplicação de interesse é multiponto-a-multiponto, um melhor desempenho é obtido quando atrasos aleatórios são introduzidos entre a detecção de uma perda e o envio do NAK [9, 12].

1.2 Mobilidade na Internet

Algumas características importantes diferenciam a comunicação com estações móveis: a perda de dados em trânsito devido ao movimento intercelular, a necessidade de localizar as estações móveis para o serviço de roteamento de pacotes devido ao movimento e as altas taxas de erro nos enlaces sem fio [7].

Algumas propostas foram feitas para contornar os efeitos da perda de dados nos enlaces sem fio. O Snoop TCP [6] propõe o armazenamento temporário de pacotes na estação rádio-base para a retransmissão local dos pacotes perdidos devido ao movimento intercelular e aos erros nos enlaces sem fio com o objetivo de evitar que estas perdas sejam consideradas como sendo devido à congestionamentos. Desta forma, o tempo de recuperação é diminuido, e evita-se o acionamento das políticas de controle de fluxo do TCP. O I-TCP (*Indirect-TCP*) [13] divide uma conexão TCP em duas: uma conexão entre o transmissor e a estação rádio-base, e outra entre a estação rádio-base e a estação móvel. Assim, consegue-se uma separação entre os controles de fluxo e congestionamento da rede fixa e os da rede sem fio.

Uma proposta para o roteamento de datagramas IP (*Internet Protocol*) para estações móveis é dada por Perkins [14, 15]. A proposta para IP móvel (*Mobile IP*) considera o uso de agentes intermediários para a comunicação de dados com estações móveis, como está ilustrado na figura 1.1. Um agente localizado na rede de origem da estação móvel, denominado agente de origem (*home agent*), tem como responsabilidade o envio de datagramas IP destinados à estação móvel através de um túnel IP para um agente localizado na rede em que a estação está visitando no momento, denominado agente estrangeiro (*foreign agent*). O agente estrangeiro recebe os datagramas IP através do túnel e os remete para o terminal móvel. Outra função do agente de origem é manter a localização do terminal móvel, que deve ser atualizada sempre que houver a passagem do domínio de um agente estrangeiro para o domínio de outro. Entretanto, esta proposta não permite taxas elevadas de movimento intercelular, pois a ocorrência de uma troca de célula implica na atualização das informações no agente de origem. Como o agente de origem pode estar distante da localização corrente da estação móvel, o tempo para atualização pode ser elevado. Se o período médio entre trocas de célula for maior que o tempo para atualização do agente de origem, a troca de agentes estrangeiros ocorrerá antes desta atualização. Assim, para esta proposta, os usuários estão limitados a baixas taxas de movimento intercelular.

Mais recentemente, Campbell *et al.* [8] propuseram o uso de nós especiais, denominados nós celulares, para redes IP celulares. Uma rede IP celular consiste de vários nós celulares interconectados e um *gateway* que interliga a rede com o restante da Internet, como está ilustrado na figura 1.2. Os nós celulares realizam o roteamento dos pacotes IP dentro da rede celular e comunicam-se com as estações móveis através de interfaces aéreas. Os nós celulares com interfaces aéreas são denominadas estações rádio-base e enviam periodicamente um sinal de orientação (*beacon signals*) para permitir que as estações móveis localizem a estação rádio-base mais próxima,

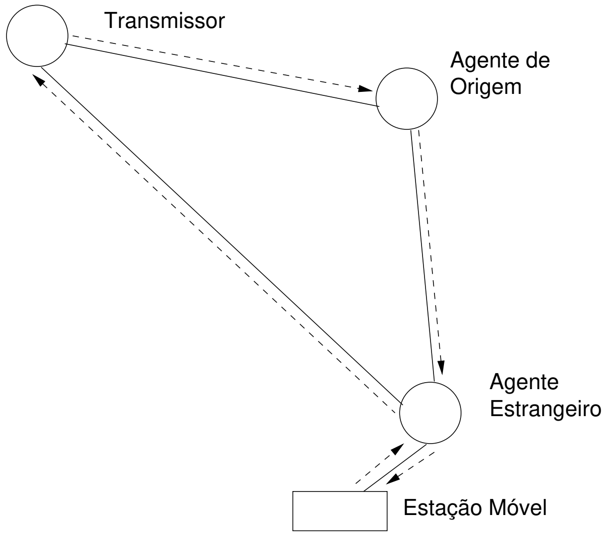


Figura 1.1: Triangulação no IP móvel.

a fim de realizar a troca de células. Com o objetivo de realizar o roteamento para as estações móveis, os nós celulares possuem caches para armazenar o caminho temporário para os terminais móveis. Estes caches são atualizados através da observação do endereço da fonte nos datagramas IP que passam pelos nós celulares. Isto permite que a atualização do caminho para uma estação móvel seja feita nos nós celulares próximos de sua localização, e desta forma a rede se adapta rapidamente ao movimento das estações móveis, possibilitando altas taxas de movimento intercelular. A proposta para IP celular atua em conjunto com a proposta para IP móvel, sendo que o *gateway* da rede celular faz o papel do agente estrangeiro de uma estação móvel, enquanto esta estação estiver no domínio de sua rede celular. Uma troca de agente estrangeiro ocorre somente quando uma estação móvel muda de uma rede celular para outra.

Algumas propostas foram feitas por Xylomenos e Polyzos [16] para a integração do IP móvel com o IP multiponto [17]. O IP multiponto baseia-se no conceito de grupo de estações, que é um conjunto dinâmico de estações identificado por um único endereço IP de classe D. Uma estação pode aderir e se desligar de um grupo em qualquer instante para começar ou parar de receber mensagens. A adesão e o desligamento são realizados através de mensagens do protocolo IGMP (*Internet Group Management Protocol*) [18] trocadas entre as estações e roteadores multipon-

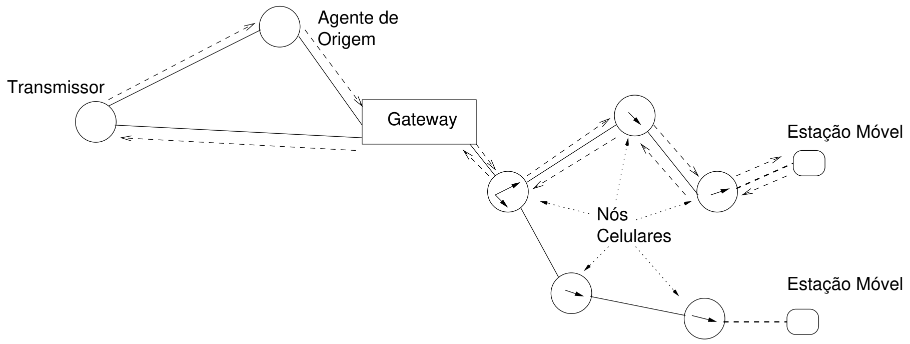


Figura 1.2: Comunicação com estações móveis no IP celular.

to. Para a entrega correta de datagramas multipontos, os roteadores devem trocar informações entre si, existindo vários mecanismos de roteamento com esta finalidade, como o DVMRP (*Distance Vector Multicast Routing Protocol*), o MOSPF (*Multicast Open Shortest Path First*) e o CBT (*Core-based Tree*) [19, 20].

Para determinar as interfaces para as quais um datagrama IP multiponto deve ser replicado, alguns mecanismos baseiam-se na interface de chegada deste datagrama e no endereço da fonte. Desta forma, se as estações se movimentam e os datagramas de uma determinada fonte chegam em uma interface diferente a cada instante, então erros de roteamento poderiam ocorrer. Por isto, Xylomenos e Polyzos propõem que os datagramas multipontos enviados por estações móveis sejam entregues através de túneis para os respectivos agentes de origem, e somente a partir destes agentes ocorreria o roteamento multiponto.

Tratando do problema de recepção de datagramas multipontos, três propostas foram feitas: o roteamento através do agente de origem, o roteamento através do agente estrangeiro e o roteamento combinado. O roteamento através do agente de origem propõe que este agente trate do roteamento multiponto, enviando através de um túnel os datagramas destinados à estação móvel e recebendo desta estação as informações pertinentes aos grupos de interesse. Este esquema é transparente para os agentes estrangeiros e interoperável com as redes existentes, mas não é eficiente com relação ao roteamento devido a triangulação. No roteamento através do agente estrangeiro, este agente possui o papel de roteador multiponto, sendo responsável pela manipulação das informações de cada estação móvel e pela entrega dos datagramas. Apesar de ser mais eficiente com relação ao roteamento, existe um custo associado à função de roteador multiponto para o agente estrangeiro em termos de entrega de datagramas e utilização de recursos. Por último, o roteamento combinado propõe que seja criado um túnel único para cada grupo multiponto de interesse sob o domínio de um agente estrangeiro. Cada túnel ligaria um agente de origem ao agente estrangeiro, sendo o agente estrangeiro responsável pela replicação para todos as estações pertencentes ao grupo. Esta abordagem também possui o problema de eficiência no roteamento devido à triangulação, mas possui a vantagem do agente estrangeiro não desempenhar o papel de roteador multiponto.

1.3 Organização da Tese

Os capítulos seguintes desta tese estão divididos da seguinte forma. O capítulo 2 apresenta os conceitos e o estado da arte das pesquisas em redes ativas, incluindo as principais propostas para infra-estruturas e as suas características. Ainda neste capítulo são descritas algumas propostas relacionadas à comunicação multiponto em redes ativas e que foram utilizadas no serviço proposto nesta tese. O serviço de comunicação multiponto para estações móveis é definido no capítulo 3, onde são apresentados o protocolo que suporta este serviço e os seus algoritmos. O capítulo 4 mostra os resultados obtidos por simulações, comentando brevemente a ferramenta de simulação utilizada e as modificações implementadas. Por último, o capítulo 5 apresenta as conclusões desta tese.

Capítulo 2

As Redes Ativas

O PARADIGMA de redes ativas propõe que o serviço de rede seja programável, permitindo que computações adequadas às necessidades das aplicações sejam inseridas e executadas na rede, e que desta forma seja possível melhorar a qualidade do serviço oferecido para as aplicações. Para isto, vários grupos de pesquisa vem propondo infra-estruturas para as redes ativas, visando objetivos diferentes e explorando diversos aspectos da programabilidade da rede.

A primeira seção deste capítulo descreve uma arquitetura geral que vem sendo desenvolvida no programa de redes ativas do DARPA (*Department of Advanced Research Projects*) e que mostra os principais pontos comuns entre as diversas propostas. Na seção seguinte, são caracterizados os principais modelos de ambiente de execução, sendo que em cada um são definidos a linguagem para a programação da rede, a forma de inserção de novos serviços e os recursos disponíveis nos nós para a programação do serviço. Por último, são apresentadas algumas propostas para comunicação multiponto em redes ativas nas quais o serviço desta tese baseou-se.

2.1 Arquitetura Geral de Redes Ativas

A comunidade científica vem desenvolvendo uma arquitetura que descreve os componentes funcionais e a interface destes componentes para as redes ativas. Esta arquitetura assume uma rede onde a comunicação é realizada através de comutação de pacotes, sendo que o objetivo desta infra-estrutura é prover um serviço para comunicação fim-a-fim, e não serviços gerais de computação [3, 21].

Como o objetivo é o fornecimento de um serviço de rede programável, a abordagem empregada tem sido a especificação da arquitetura de um nó ativo. Esta arquitetura está ilustrada na figura 2.1. Neste arquitetura, existem três componentes principais: as aplicações, os ambientes de execução e o sistema operacional do nó.

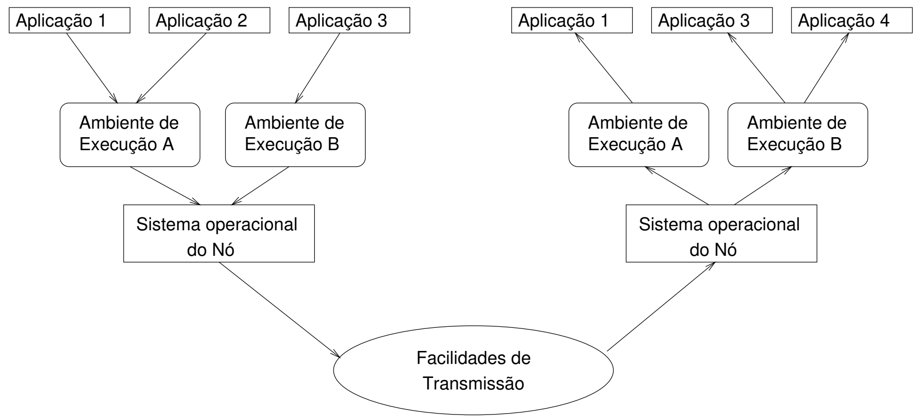


Figura 2.1: Arquitetura geral de um nó ativo.

As aplicações obtêm um serviço interagindo com um ambiente de execução, o qual tem uma instância a cada nó ativo e também nas entidades finais de comunicação. Por sua vez, o ambiente de execução é responsável por todos os aspectos da interface entre a aplicação e a rede, incluindo a sintaxe e a semântica dos pacotes que são enviados, o modelo de programação e as abstrações suportadas, e ainda o endereçamento. Desta forma, é necessário que cada ambiente de execução exporte uma API (*Application Programming Interface*) para uso das aplicações, disponibilizando o acesso às diversas facilidades dos nós.

O suporte a vários ambientes de execução é necessário para permitir que os diversos ambientes em desenvolvimento sejam testados e novas versões sejam adicionadas, mantendo as antigas por questões de compatibilidade. Entretanto, como os recursos disponíveis em um nó são compartilhados entre os ambientes de execução, existe a necessidade de um componente para controlar o uso dos recursos, sendo este o papel principal do sistema operacional (SO) do nó.

Entre as abstrações definidas para o SO encontra-se o fluxo. O fluxo é uma abstração utilizada para a contabilidade, o controle de admissão e o escalonamento agregado no sistema. Recursos do sistema operacional tais como ciclos de UCP (Unidade Central de Processamento), memória utilizada e banda passante são alocados para cada fluxo. Assim, os pacotes ativos são processados pelo ambiente de execução utilizando os recursos reservados para o fluxo ao qual estão associados.

2.2 Modelos de Ambiente de Execução

Os principais objetivos das redes ativas são a adequação do serviço de rede aos requisitos das aplicações e a facilidade de inserção de novos serviços. Assim, as escolhas do modelo de programação e dos recursos a serem disponibilizados são feitas visando o atendimento destes objetivos, sendo possível a caracterização das diversas propostas através dos seguintes atributos [3] :

- expressividade da linguagem de programação, isto é, o grau de programabilidade da rede pode variar de um simples conjunto de escolhas pré-definidas, até uma linguagem de programação abrangente capaz de descrever qualquer computação realizável;
- capacidade de criar e referenciar estados nos nós intermediários da rede;
- granulosidade do controle, ou seja, o escopo de um nó que pode ser modificado pela recepção de um pacote. Assim, em um extremo, um simples pacote pode alterar o comportamento de um nó, sendo a alteração aplicada a todos os pacotes. Em um outro extremo, um único pacote modifica o comportamento visto somente por ele. Em um nível intermediário, as modificações seriam aplicadas a um fluxo.

2.2.1 ANTS

O ANTS (*Active Node Transfer System*) [22, 23, 24, 25] foi desenvolvido no MIT por Tennenhouse *et al.*, sendo empregada a linguagem Java para a programação dos serviços. No ANTS, os pacotes convencionais que transportam dados do usuário são substituídos por cápsulas que contêm, além dos dados, uma descrição do processamento a ser executado na rede. Esta descrição é utilizada a cada nó ativo para identificar o procedimento de envio que deve ser executado e que caracteriza o tipo da cápsula. Neste sistema, um protocolo é definido por um conjunto de tipos de cápsula, sendo os procedimentos de envio de um protocolo transferidos entre os nós através de um mecanismo de distribuição de código. Quando uma cápsula chega a um nó ativo, o identificador do procedimento da cápsula é lido e o procedimento buscado em um cache. Se o conjunto de procedimentos do protocolo não for encontrado, um pedido é feito ao nó anterior. Desta forma, os procedimentos de um protocolo são difundidos pelo caminho onde são utilizados.

Para a programação dos procedimentos, são disponilizadas primitivas que fornecem acesso aos recursos e às facilidades disponíveis nos nós. Entre estas primitivas, encontram-se as utilizadas para a manipulação do armazenamento não-transiente (*soft-state*) e para o controle de cápsulas. O *soft-state* consiste em um espaço de memória não-transiente, isto é, que persiste à execução de uma cápsula, onde são guardadas informações de um fluxo por um período de tempo determinado, chamado de tempo de expiração. Assim, as cápsulas de um mesmo fluxo podem colocar e alterar informações necessárias para a execução dos procedimentos de envio das cápsulas seguintes do seu fluxo. Estas informações podem ser úteis, por exemplo, para decidir o próximo nó para o qual uma cápsula deve ser enviada, ou ainda se uma cápsula deve ser descartada ou replicada para várias interfaces. As ações de envio, descarte e replicação são realizadas através de chamadas às primitivas de controle de cápsulas nos procedimentos de envio.

2.2.2 *SwitchWare*

O projeto *SwitchWare* da universidade da Pennsylvania tem como objetivo atingir a melhor relação possível entre a flexibilidade de uma rede programável e os requisitos de segurança inerentes a uma infra-estrutura compartilhada [26, 27]. Com este objetivo, a arquitetura definida neste projeto é dividida em três níveis: pacotes ativos, extensões ativas e infra-estrutura do roteador ativo.

Os pacotes ativos são programas que substituem os pacotes tradicionais e consistem de ambos código e dados. O código de um pacote ativo desempenha a função de controle antes realizada pelo cabeçalho, mas de uma forma mais flexível, pois o processamento de um pacote não está mais restrito a uma simples consulta à tabela de rotas. Para a programação dos pacotes ativos, foi criada uma linguagem, denominada PLAN (*Programming Language for Active Networks*), simples e que fornece o mínimo de funcionalidades, evitando em vários casos a verificação do código nos nós ativos [28]. Como exemplo de funcionalidade não disponível para a linguagem PLAN, encontra-se a manipulação do estado de um nó. Outra linguagem que está sendo experimentada para a programação dos pacotes é a Caml. A Caml é uma linguagem de uso geral e, sendo assim, não possui as restrições de funcionalidades existentes para a PLAN, mas exige a verificação do código.

A limitação de funcionalidades imposta à linguagem é compensada pelo uso de extensões ativas. Estas extensões são carregadas nos roteadores e fornecem serviços para vários pacotes ativos, aumentando desta forma as possibilidades para o processamento. Assim, as extensões são consideradas funcionalidades básicas carregadas dinamicamente, ao invés de código móvel, necessitando de verificação do código somente ao serem carregadas.

O primeiro nível da arquitetura é a infra-estrutura do roteador ativo seguro. Enquanto os dois níveis superiores focalizam a mobilidade e a alteração dinâmica de código, este nível é basicamente estático e tem como objetivo a formação de uma fundação segura sobre a qual são construídos os dois outros níveis.

2.2.3 *Smartpackets*

No projeto *Smartpackets* da BBN, o principal enfoque é a utilização do conceito de redes ativas para o auxílio no gerenciamento de redes. Assim, cada datagrama IP carrega um procedimento a ser executado nos nós ativos em que passam, sendo que os procedimentos não são guardados nos nós ativos, existindo somente durante o período de tempo das suas execuções. Para a programação das rotinas, utiliza-se uma linguagem chamada *Sprocket* que é compilada para uma linguagem de representação compacta e similar a uma linguagem assembly genérica denominada *Spanner*. O ambiente de execução para a *Spanner* inclui o suporte para o acesso de variáveis de gerenciamento de rede em MIBs (*Management Information Bases*), não existindo, entretanto, acesso a outras chamadas do sistema ou à memória fora do escopo de execução do datagrama. Um suporte adicional existe para a autenticação dos pacotes que visa o controle e a monitoração seguros da rede [3, 21].

2.2.4 *Software* Líquido

Pesquisadores da universidade do Arizona definem *software* líquido (*liquid software*) [29] como sendo um código móvel, orientado a comunicação e de baixo nível, que pode fluir facilmente entre os nós de uma rede. Como a meta do *software* líquido é o transporte eficiente de dados, diferentes características são demandadas do sistema operacional e do ambiente de execução que o suportam, como garantias de qualidade de serviço e restrições de tempo real associadas a comunicação. Com este objetivo, foi utilizado um sistema operacional configurável através da composição de módulos, denominado *Scout*, para construir uma plataforma para redes ativas chamada *Joust*.

A plataforma *Joust* possui um módulo interpretador para a linguagem Java, utilizada para programar o serviço de rede. No entanto, como a Java é uma linguagem interpretada e, normalmente, os interpretadores desta linguagem restringem o acesso aos recursos do sistema, foram inseridos módulos para a tradução do código interpretado para código objeto em tempo de execução (compiladores *just-in-time*) e módulos para introduzir funcionalidades de comunicação na API da linguagem Java, ambos visando o aumento do desempenho.

2.2.5 *Netscript*

O projeto *Netscript* [30] da universidade de Columbia procura definir um conjunto mínimo de funcionalidades em um nó, a partir das quais é possível desenvolver uma grande variedade de serviços. Com esta finalidade, utiliza-se o conceito de agente, que é definido como um programa que pode ser enviado para a execução em um nó qualquer da rede, adicionando funções primitivas a este nó para o processamento dos pacotes dos fluxos e para a alocação de recursos. O principal objetivo dos agentes neste projeto é a programação de funções de controle e de gerenciamento da rede, sendo possível tanto a implementação de protocolos padronizados, quanto de protocolos não padronizados. Os pacotes, chegando aos nós, são processados pelos agentes apropriados para cumprir as funcionalidades estabelecidas para os protocolos.

2.3 Comunicação Multiponto em Redes Ativas

A possibilidade de armazenar temporariamente estados nos nós ativos é um recurso poderoso para as computações na rede [4, 5, 24]. A criação e a manutenção de estados nos nós permite a tomada de decisões pelos nós ativos, possibilitando, muitas vezes, a economia de recursos de banda passante e de processamento nas entidades de comunicação de extremidade [3]. Estas características são especialmente importantes para a comunicação multiponto confiável com um grande número de usuários, onde é desejado um consumo de recursos distribuído ao máximo para não sobrecarregar nenhuma entidade de extremidade envolvida na comunicação e nenhum nó em particular da rede de comunicação.

Tennenhouse *et al.* [25] propõem um protocolo para a formação de árvores multidestinatárias baseado no ANTS e composto por dois tipos de cápsula cooperantes.

Uma cápsula, denominada cápsula de inscrição, utilizada por todos os receptores interessados no grupo multiponto, e outra, denominada cápsula de dados multiponto, que carrega as mensagens para os membros do grupo. Aplicações interessadas em receber as mensagens de um grupo em particular devem enviar periodicamente cápsulas para o transmissor. Estas cápsulas instalam apontadores no *soft-state* dos nós ativos para marcar os caminhos do transmissor para os receptores. As cápsulas de dados multipontos simplesmente percorrem o caminho guardado no *soft-state* dos nós ativos. Esta situação está ilustrada na 2.2.

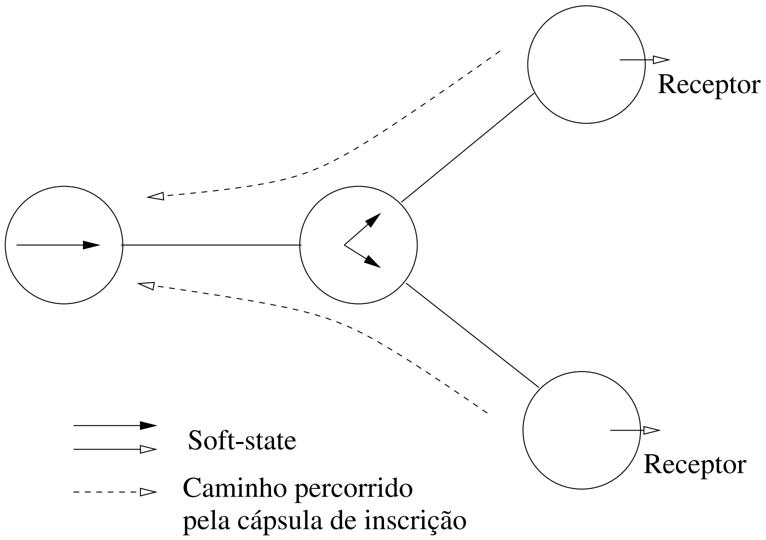


Figura 2.2: Formação de um grupo multiponto.

No entanto, nem todas as cápsulas de inscrição enviadas precisam percorrer o caminho completo até o transmissor. Se parte do caminho for compartilhada por dois ou mais receptores, somente uma cápsula de inscrição precisa percorrer o caminho até o transmissor entre períodos de expiração do *soft-state*, sendo que o restante das cápsulas pode ser descartado. O objetivo deste mecanismo de descarte do excesso de cápsulas de inscrição é consumir o mínimo de banda passante para a manutenção da árvore multiponto, sendo bastante robusto com relação às mudanças de topologia. Outra característica importante deste mecanismo é a distinção entre transmissores e receptores. Para uma árvore formada, existe apenas um transmissor. Se vários transmissores forem necessários, ou várias árvores devem ser formadas, ou utiliza-se a estratégia de roteamento CBT (*Core-based tree*) [31], com as mensagens sendo enviadas primeiro para a raiz e depois difundidas para os receptores.

Tennenhouse *et al.* [10] propõem também um esquema para a recuperação de perdas em comunicação multiponto denominado ARM (*Active Reliable Multicast*). O protocolo proposto para este esquema é orientado ao receptor, isto é, NAKs são enviados pelo receptor para o transmissor quando perdas são detectadas [9]. Para resolver o problema da implosão de NAKs, o ARM propõe o uso do *soft-state* para registrar a passagem de reconhecimentos negativos, com o objetivo de descartar as mensagens duplicadas. Outra característica importante do ARM é a utilização de caches de dados para manter temporariamente as mensagens enviadas para o grupo. Os pedidos de retransmissão enviados para o transmissor consultam os nós ativos intermediários buscando a mensagem perdida. Caso a mensagem perdida seja encontrada em algum dos caches dos nós intermediários, a retransmissão é feita e o pedido de retransmissão é descartado. Assim, diminui-se o tempo de recuperação de uma mensagem e mantém-se a retransmissão dentro do escopo de perda. Por escopo de perda, entende-se o sub-conjunto de usuários que não recebeu o pacote enviado ou recebeu incorretamente. Esta abordagem de retransmissões por escopo é similar à abordagem hierárquica adotada pelo RMTP [32]. O ARM, no entanto, é mais robusto com relação à mudanças na topologia, pois não depende de nenhum nó em especial ou receptor para realizar as retransmissões.

Os mecanismos de formação de árvores de distribuição e de recuperação de perdas apresentados foram utilizados no serviço de comunicação multiponto confiável para estações móveis proposto nesta tese. A contribuição deste trabalho está no atendimento aos requisitos tanto de escalabilidade quanto de altas velocidades para os participantes, mesmo considerando as altas taxas de perda encontradas nos ambientes celulares. Trabalhos anteriores, que foram mostrados no capítulo 1, não abordam em uma só proposta ambos os requisitos.

Capítulo 3

O Serviço de Comunicação Multiponto Móvel

O DESENVOLVIMENTO de um serviço de comunicação para estações móveis em redes celulares deve considerar as principais características que diferenciam este tipo de comunicação, tais como: as altas taxas de perdas e a necessidade de atualização das informações de posição. Além disto, em um serviço voltado para aplicações multidestinatárias que necessitam de confiabilidade, o requisito de escalabilidade também deve ser observado, ou seja, a quantidade utilizada de recursos da rede e dos transmissores não deve crescer na mesma proporção do número de usuários.

Assim, o serviço proposto neste capítulo se utiliza de computações realizadas nos nós da rede para: permitir rápidas atualizações nas árvores de distribuição multipontos, evitar o problema de implosão de mensagens no transmissor, possibilitar recuperações rápidas e limitar o escopo de retransmissão para o sub-conjunto de usuários que efetivamente sofreram as perdas no transmissor [33]. Para isto, algumas suposições existem para a infra-estrutura de rede sob a qual o serviço é executado, que serão apresentadas na próxima seção. O serviço e o protocolo são apresentados nas seções seguintes.

3.1 A Infra-estrutura de Rede

Uma infra-estrutura de redes ativas baseada no modelo de ambiente de execução do ANTS foi assumida para suportar o serviço proposto. Assim, esta rede é composta por nós ativos que suportam a execução de procedimentos específicos das aplicações, sendo utilizadas cápsulas para transportar mensagens de dados e de controle entre as entidades de comunicação de extremidade. As cápsulas estão associadas a procedimentos que são executados a cada nó ativo e se servem de um conjunto de facilidades disponibilizadas por estes nós. As facilidades necessárias para o serviço proposto são:

- o armazenamento não-transiente (*soft-state*) disponível para as cápsulas de um mesmo fluxo. O *soft-state* de um fluxo persiste sem atualização por um período de tempo determinado a cada nó ativo. Para esta infra-estrutura, considerou-se que o período de expiração é o mesmo para todos os nós ativos;
- o controle de cápsulas que engloba as funções de: envio, descarte, criação e replicação.

Estas facilidades estão disponíveis em todos os nós da rede, pois todos foram considerados ativos para as simulações realizadas no capítulo 4. Entretanto, considerando-se um caso real em que somente alguns nós escolhidos estrategicamente são ativos, o serviço não estaria impedido de funcionar, mas poderia vir a ter um desempenho inferior, dependendo da distribuição e do número de nós ativos [10].

Desta forma, o interior da rede, ou o *backbone*, é formado por nós ativos e na periferia encontram-se as redes celulares, onde cada célula possui um nó ativo responsável por ela denominado nó ativo celular (NAC). Um exemplo de uma estrutura de rede seguindo o modelo definido é apresentada na figura 3.1. O NAC assume o papel da estação rádio-base e, sendo assim, possui interfaces aéreas para comunicação com as estações móveis. Para permitir a decisão de troca de célula, supõe-se que cada nó celular emite periodicamente sinais de orientação (*beacon signals*) que permitem a identificação do NAC mais próximo através da intensidade do sinal [6, 8].

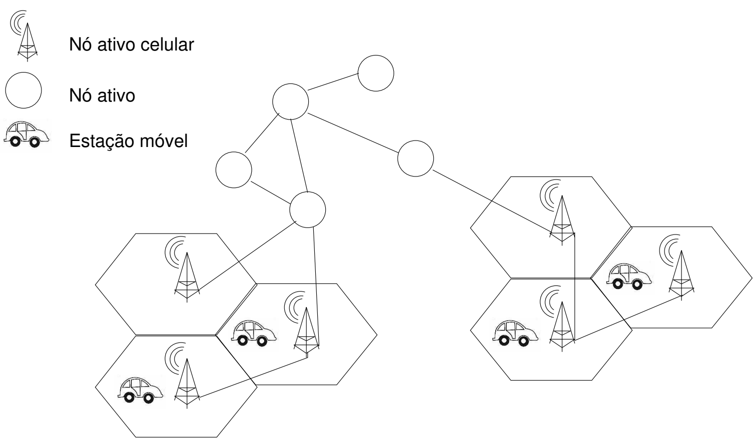


Figura 3.1: Exemplo de estrutura de rede.

As estações móveis também são consideradas como sendo ativas, pois não só possuem a capacidade de gerar e consumir cápsulas, mas também de processá-las.

3.2 O Serviço Proposto

A principal característica do serviço proposto é a formação da árvore multiponto através do uso de marcações realizadas no *soft-state* dos nós ativos. As marcações (ou apontadores) são manipuladas por cápsulas de inscrição que percorrem os caminhos das estações móveis até o transmissor do grupo multiponto. Este mecanismo foi utilizado devido às rápidas alterações permitidas na configuração de uma árvore multiponto, pois, neste caso, a configuração deve ser alterada não somente quando usuários entram ou saem de um grupo, mas também a cada handoff de uma estação participante. Para isto, a cápsula de inscrição percorre o caminho até o transmissor, verificando nos nós ativos a existência de marcações referentes ao grupo multiponto de interesse. Ao encontrar uma referência do grupo em um dos nós ativos, a adesão ao grupo está completa e este receptor passa a receber as mensagens. Este nó pode descartar a cápsula de inscrição ou deixá-la percorrer o restante do caminho até o transmissor com a finalidade de atualizar o tempo de expiração do *soft-state* dos próximos nós ativos.

As cápsulas de dados multipontos são enviadas para os receptores seguindo o caminho deixado pelas cápsulas de inscrição. Assim, o destino das cápsulas de dados multipontos não é determinado pelo endereço da máquina e sim pelo caminho marcado no *soft-state*, permitindo o envio direto dos dados para o grupo multiponto sem depender da existência de agentes intermediários como é proposto para o IP móvel.

A confiabilidade do serviço é conseguida através de retransmissões realizadas pelo transmissor ou pelos nós ativos intermediários, como é proposto no ARM [10]. Caches de cápsulas de dados multipontos existem nos nós ativos para uma recuperação mais rápida, uma vez que o pedido de retransmissão não precisaria subir toda a árvore multiponto para pedir ao transmissor a retransmissão de uma cápsula de dados. Além disto, a retransmissão é feita dentro do escopo de perdas, isto é, é feita somente para o sub-conjunto dos usuários que não recebeu ou recebeu incorretamente a cápsula de dados. Neste contexto, os nós ativos mais próximos das redes celulares desempenham um papel fundamental. Supondo que a maior parte das perdas ocorre devido aos handoffs e às altas taxas de erro dos enlaces sem fio, estes nós ativos mais próximos realizarão a maior parte das retransmissões para as estações móveis. Isto é necessário porque o tempo de recuperação de perdas causadas pela mobilidade deve ser menor que o período médio entre handoffs, senão ocorreria um acúmulo de perdas e a eficiência do serviço seria baixa.

Entretanto, o ARM propõe a utilização do cache segundo uma política de melhor esforço, em que existe um espaço de armazenamento disponível para o cache, compartilhado por todos os grupos multipontos em atividade. Como nas propostas para as futuras redes ativas é assumido que cada fluxo aloca o seu espaço de armazenamento, existe a necessidade de um mecanismo eficiente para determinar a quantidade de armazenamento não-transiente necessária em um nó ativo para um grupo multiponto. Assim, um mecanismo foi proposto para a negociação e a alocação de memória destinada ao cache. A quantidade alocada depende das condições da rede e da disponibilidade do nó [34]. Este mecanismo é apresentado com mais detalhes na seção 3.4.

3.3 O Protocolo Proposto

O protocolo proposto para este serviço é composto por quatro cápsulas: cápsula de inscrição, cápsula de dados multiponto, cápsula de pedido de retransmissão e cápsula de fim de sessão.

3.3.1 A Cápsula de Inscrição

```
Seja p a cápsula que disparou o procedimento
Procedimento de Inscrição (p)
    Configurar o tamanho do cache de dados
    Ler soft-state do grupo
    Se soft-state não existe então
        Criar soft-state para o grupo
        Inserir apontador para a interface de chegada
    Senão
        Retirar apontadores expirados
        Se apontador para interface de chegada não existe então
            Inserir apontador
        Senão
            Se t_e ≤ (t_s × β) então
                Descartar (p)
                Retornar
            Fim-Se
        Fim-Se
    Fim-Se
    Enviar p para o transmissor
Fim-Procedimento
```

As cápsulas de inscrição são enviadas periodicamente para o transmissor de um grupo multiponto por todas as estações móveis interessadas em receber os dados deste grupo. Além disto, uma nova cápsula de inscrição deve ser enviada sempre que houver um handoff, para atualizar a topologia da árvore multiponto, como está ilustrado na figura 3.2. Para que ocorra este envio a cada handoff, deve haver uma sinalização a nível de enlace indicando a ocorrência de um movimento intercelular.

Supõe-se que existe um mecanismo semelhante ao utilizado para o IP celular, em que as estações móveis decidem pela troca de célula a partir da intensidade dos sinais de orientação enviados periodicamente pelos nós ativos celulares.

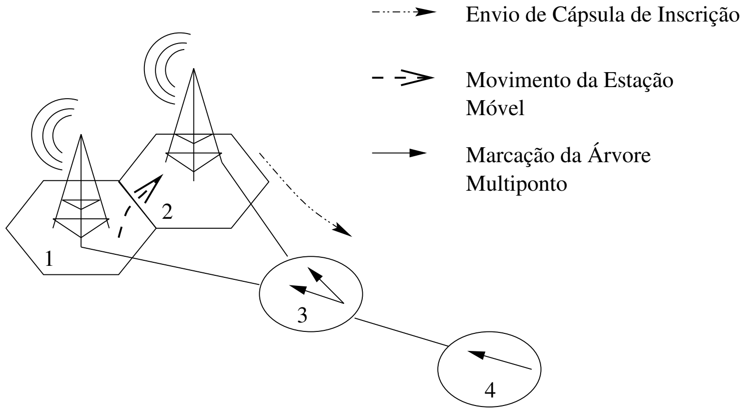


Figura 3.2: Envio de inscrição após troca de célula.

O período de envio de cápsulas de inscrição depende do tempo de expiração do *soft-state* nos nós ativos. Estas cápsulas devem ser enviadas dentro de um período de tempo que permita sustentar a continuidade da comunicação, mas que mantenha o compromisso de não inundar a rede com cápsulas de inscrição. Além disto, este mesmo período é utilizado como parâmetro nos nós para filtrar o excesso de cápsulas de inscrição, pois como parte dos caminhos em uma árvore multiponto pode ser compartilhada entre dois ou mais usuários, somente as inscrições necessárias para manter o *soft-state* dos nós ativos atualizados devem passar. O procedimento de inscrição verifica através da estampa de tempo colocada pelo nó ativo a cada vez que o *soft-state* é atualizado. Além desta estampa para o controle do *soft-state* de um grupo como um todo, cada apontador no *soft-state* do grupo deve conter uma estampa de tempo. Uma das funções do procedimento de inscrição é verificar os apontadores expirados e removê-los. Se isto não fosse feito no *soft-state* de um grupo em que existem várias indicações e algumas delas são válidas, não seria possível a remoção de apontadores para usuários que não participam mais do grupo ou que trocaram de célula. Desta forma, diminui-se a quantidade de cápsulas de dados multipontos enviadas desnecessariamente e, como conseqüência, economiza-se banda passante.

Assim, o período de envio (t_e) deve ser menor que o período de expiração do *soft-state* (t_s) para compensar atrasos em filas. Outra característica que deve ser observada ao determinar t_e é a perda de cápsulas de inscrição, ou por erros nos enlaces, ou por descarte nas filas de saída devido a congestionamentos. Para exemplificar a perda de cápsulas de inscrição, três casos podem ser observados a partir da árvore multiponto dada pela figura 3.3, supondo que $2 \times t_e \geq t_s$ e também que o período mínimo sem comunicação é dado pela equação 3.1, sendo n o número de perdas consecutivas de cápsulas de inscrição em um mesmo nó ativo. No primeiro caso, se cápsulas de inscrição enviada por um dos nós $(5, 6, 7$ ou $8)$ para um dos nós (3 ou 4) forem perdidas, o nó que enviou permanecerá sem comunicação com o transmissor. Esta perda de comunicação temporária ocorre devido a expiração do *soft-state* do grupo. Um caso diferente seria se cápsulas de inscrição repassadas por um dos nós (3 ou 4) para o nó (2) fossem perdidas, pois assim dois receptores permaneceriam sem comunicação. Mesmo se um dos receptores sem comunicação enviasse uma cápsula de inscrição durante este período, a comunicação não poderia ser restabelecida, pois no *soft-state* do nó ativo onde cápsulas de inscrição foram perdidas e dos nós ativos anteriores até os receptores, o tempo de atualização foi alterado e, portanto, até o fim do período t_e todas as cápsulas de inscrição são descartadas. O pior caso ocorreria se cápsulas de inscrição fossem perdidas entre o nó (2) e o nó (1), pois todos os receptores permaneceriam sem comunicação e a retransmissão não poderia ser realizada pelos nós intermediários, obrigando o transmissor a reenviar todos os dados gerados neste período de tempo.

$$t_{s/com} \geq ((n+1) \times t_e) - t_s \tag{3.1}$$

onde,

$$\begin{aligned} n &= \text{número de perdas consecutivas e } n \geq 1 \\ t_{s/com} &= \text{Período de tempo sem comunicação} \end{aligned}$$

A figura 3.4 ilustra duas situações onde $t_e < t_s$. Para o caso 1, em que a condição é dada por 3.2, se a cápsula de inscrição for perdida no seu percurso, haverá perda de continuidade na comunicação. Para o caso 2, em que a condição é dada por 3.3,

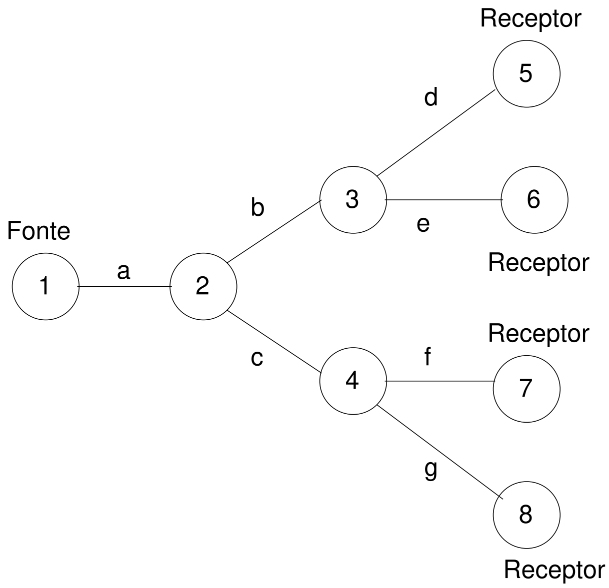
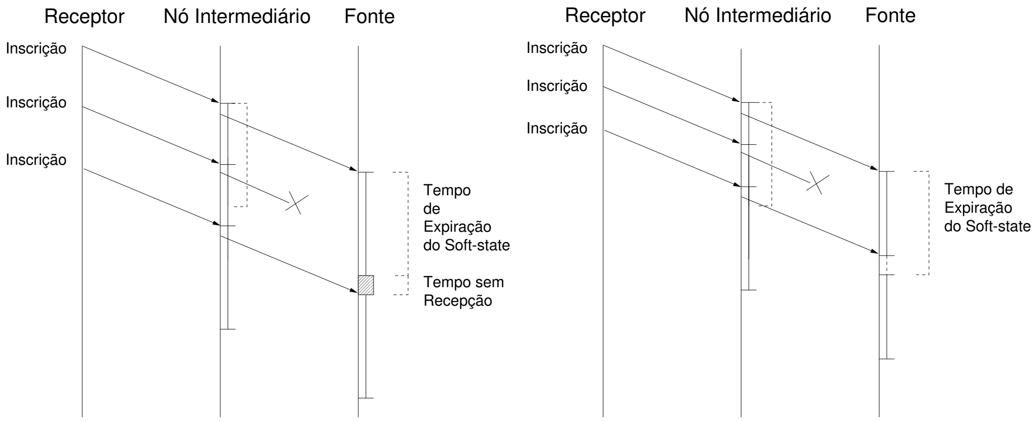


Figura 3.3: Exemplo de árvore multiponto.

a ocorrência de uma perda não acarreta em descontinuidade na comunicação, pois a segunda cápsula de inscrição enviada percorre o caminho a tempo de atualizar o *soft-state*. Só haveria descontinuidade na comunicação para a situação 2 em caso de congestionamento no caminho do receptor para a fonte. Desta forma, utiliza-se neste serviço o período de tempo dado pela condição 3.4, que corresponde à situação 2, para gerar novas cápsulas de inscrição e também no procedimento de inscrição para decidir pelo envio ou pelo descarte de uma cápsula de inscrição.

O valor percentual de t_s utilizado como período de envio pode ser calculado supondo um número maior de perdas de cápsulas de inscrição. No entanto, o período de tempo conferido para permitir a perda de uma cápsula garante robustez ao mecanismo, pois considera-se que a probabilidade de perda de duas ou mais cápsulas de inscrição consecutivas é pequena. Além disto, se duas ou mais perdas consecutivas de cápsulas de inscrição ocorrerem, é provável que somente uma parte do grupo seja afetada, dependendo da dispersão dos usuários e da probabilidade de perda nos enlaces.

$$t_s > t_e > 0.5 \times t_s \tag{3.2}$$



(a) Caso 1 (b) Caso 2

Figura 3.4: Envio de inscrição.

$$0.5 \times t_s > t_e \tag{3.3}$$

$$\beta \times t_s \geq t_e, \quad \text{onde} \quad 0.5 > \beta \geq 0 \tag{3.4}$$

3.3.2 A Cápsula de Dados Multiponto

`Seja` *p* `a cápsula que disparou o procedimento`
Procedimento de Dados (*p*)
  `Ler soft-state do fluxo de` *p*
  Para `cada apontador no soft-state`
    `Enviar uma cópia de` *p*
  Se `existe cache de dados` **então**
    `Copiar dados para o cache`
Fim-Procedimento

As cápsulas de dados multipontos seguem as marcações deixadas pelas cápsulas de inscrição até atingirem os receptores. Assim, as funções principais destas cápsulas são: ler as indicações contidas no *soft-state* do grupo multiponto e encaminhar uma cópia da cápsula de dados para cada um dos caminhos. Outra função das cápsulas de dados multipontos é verificar a cada nó ativo se existe um cache para o seu grupo multiponto e, caso exista, o nó ativo ao executar uma cápsula de dados multiponto deve copiar o seu conteúdo para o cache.

3.3.3 A Cápsula de Pedido de Retransmissão

Cápsulas de pedido de retransmissão são enviadas sempre que perdas forem detectadas. Isto é feito através da verificação dos números de seqüência das cápsulas de dados multipontos. Qualquer salto no seqüenciamento é considerado como uma perda e uma cápsula de pedido de retransmissão é enviada ao transmissor para cada número de seqüência não recebido. No entanto, como os pedidos de retransmissão e as cápsulas de dados retransmitidas podem ser perdidas, cada receptor possui um temporizador associado a cada número de seqüência não recebido para gerar novos pedidos de retransmissão caso os dados não sejam recebidos. Desta forma, novos pedidos são gerados periodicamente caso a cápsula de dados com o número de seqüência desejado não seja recebida. O período de estouro do temporizador pode ser determinado de forma estática no início da simulação ou de forma dinâmica, baseando-se no RTT (*Round-trip Time*) que corresponde ao período de tempo desde o envio do pedido de retransmissão pelo receptor até a chegada da cápsula de dados retransmitida [35]. Para as simulações realizadas, foram considerados valores estáticos, determinados a partir do RTT observado sem a ocorrência de congestionamentos. A cada estouro de um temporizador de retransmissão ocorre a reinicialização deste temporizador com um período de estouro acrescido de metade do valor anterior, aumentando o período de envio de um novo pedido de retransmissão. Desta forma, evita-se o agravamento de congestionamentos na rede, pois se o número de seqüência desejado não foi recebido mesmo após o envio de um pedido de retransmissão, existe uma alta probabilidade das perdas estarem associadas a congestionamentos e, sendo assim, um número excessivo de retransmissões só agravaria o problema.

A cápsula de pedido de retransmissão caminha pelos nós ativos no sentido receptores-transmissor. Assim, em cada nó ativo é verificado se existe uma cópia da cápsula de dados requisitada no seu cache e, se existir, a retransmissão é realizada e a cápsula de pedido de retransmissão é descartada.

Seja p a cápsula que disparou o procedimento
Procedimento de Pedido de Retransmissão (p)
    Procura n^o de seqüência no cache de dados
    Se n^o requisitado está no cache **então**
        Se requisição já foi atendida **então**
            Descartar(p)
            Retornar
        Senão retransmite
        Fim-Se
    Senão
        Se requisição já foi realizada **então**
            Descartar (p)
            Retornar
        Senão enviar p para o transmissor
        Fim-Se
    Fim-Se
Fim-Procedimento

Ao passar por cada nó ativo, o procedimento desta cápsula verifica se existe o número de seqüência desejado no cache do grupo. Caso exista, o pedido de retransmissão é feito e a cápsula é descartada. Caso contrário, o procedimento deve marcar no *soft-state* do grupo o número de seqüência requisitado e enviar a cápsula para o transmissor. As próximas cápsulas de pedido de retransmissão requisitando o mesmo número de seqüência serão descartadas. Entretanto, a indicação de números de seqüência deve existir por um tempo limitado, para impedir que novos pedidos de retransmissão de um mesmo número de seqüência sejam descartados para sempre, privando membros do grupo de receber algum número de seqüência caso ocorram múltiplas perdas deste número. Assim, as indicações são marcadas com estampas de tempo, e estas estampas são verificadas no procedimento de pedido de retransmissão antes de decidir pelo envio ou pelo descarte da cápsula. O período de validade de cada indicação é baseado também no RTT entre os receptores e o transmissor.

3.3.4 A Cápsula de Fim de Sessão

A cápsula de fim de sessão existe para indicar o término de uma sessão para um grupo multiponto, o que significa que todos os dados já foram enviados pelo transmissor. Devido às próprias características do mecanismo utilizado para formação da árvore multiponto, o transmissor pode não receber cápsulas de inscrição de todos os usuários do grupo. Sendo assim, o transmissor de um grupo multiponto não possui informações de controle sobre os usuários do seu grupo e nem do que foi recebido por eles. A função desta cápsula, portanto, é informar o último número de seqüência que foi transmitido. Desta forma, os receptores que não receberam as últimas cápsulas de dados multipontos podem requisitar a sua retransmissão. Para evitar que algum receptor não receba a cápsula de fim de sessão, cada receptor possui um temporizador de silêncio. Após um determinado tempo sem receber nenhuma cápsula do transmissor, o receptor envia um pedido de retransmissão para o último número de seqüência esperado, isto é, o último número recebido mais um. Ao receber este pedido, se o número de seqüência requisitado for maior do que o último enviado, o transmissor interpreta como sendo um pedido para a cápsula de fim de sessão e a retransmite.

```
Seja p a cápsula que disparou o procedimento
Procedimento de Fim de Sessão (p)
    Ler soft-state do fluxo de p
    Para cada apontador no soft-state
        Enviar uma cópia de p
Fim-Procedimento
```

O período de estouro do temporizador de silêncio é baseado no intervalo observado entre chegadas de cápsulas de dados. Assim, a cada cápsula de dados recebida, o temporizador de silêncio é reiniciado e seu valor de estouro é recalculado. O cálculo é feito considerando-se a média dos intervalos entre chegadas de cápsulas de dados. No entanto, para evitar que rajadas tenham uma grande influência no período de estouro, a média é calculada dando-se maior peso a média anterior, conforme está apresentado na equação 3.5.

$$\text{Média} = \frac{\Gamma \times \text{Média} + \Delta t}{\Gamma + 1} \tag{3.5}$$

onde,

$$\begin{aligned}
\text{Média} &= \text{Média ponderada dos intervalos entre chegadas} \\
\Gamma &= \text{Fator de ponderação} \\
\Delta t &= \text{Último intervalo entre chegadas medido}
\end{aligned}$$

3.4 Configuração do Cache de Cápsulas

O tamanho do espaço de armazenamento destinado ao cache de um fluxo em um nó ativo é negociado pelas cápsulas de inscrição. Esta negociação envolve o estabelecimento de dois valores, que são: o tamanho máximo e o tamanho mínimo. Inicialmente é alocado um espaço com o tamanho mínimo e conforme o número de pedidos de retransmissão não atendidos no nó aumenta, o tamanho do espaço de armazenamento para o cache também aumenta, sendo o valor máximo o limite.

Para o cálculo deste valores, a cápsula de inscrição transporta dois parâmetros que são: o instante de tempo em que foi enviada e o período médio de estouro dos temporizadores. O instante de tempo em que a cápsula foi enviada é utilizado para estimar a distância do receptor (Lat), que é o valor utilizado nos cálculos. Além destes parâmetros, são necessárias as informações de banda passante disponível para a aplicação e a taxa de perda de cápsulas no enlace pelo qual chegou a cápsula de inscrição.

As equações 3.6 e 3.7 apresentam, respectivamente, o cálculo dos tamanhos mínimo e máximo do cache em um nó ativo.

$$Min = Lat \times BW \times (1 + \frac{E[Data_{size}]}{Subscribe_{size}}) + TP \times BW + 2 \times E[Data_{size}] \tag{3.6}$$

onde,

$$
\begin{aligned}
Lat &= \text{Período de tempo entre o envio por um receptor e a chegada no nó ativo} \\
BW &= \text{Banda passante total disponível} \\
E[Data_{size}] &= \text{Tamanho médio de uma cápsula de dados} \\
Subscribe_{size} &= \text{Tamanho de uma cápsula de inscrição} \\
TP &= \text{Tempo de processamento}
\end{aligned}
$$

O tamanho mínimo corresponde a quantidade de bits que podem ser armazenados no período de tempo correspondente ao intervalo entre os receptores detectarem a perda de uma cápsula de dados e o instante da chegada do pedido de retransmissão no nó ativo. Este intervalo de tempo corresponde ao envio da cápsula de dados seguinte pelo nó ativo, seguido da sua recepção pelo membro do grupo, somado ao tempo de envio de um pedido de retransmissão e a subseqüente recepção deste pedido pelo nó ativo em questão. Além disto, deve-se considerar o espaço para armazenar a cápsula perdida e a seguinte, e também o tempo de processamento (TP). Desta forma, a equação para o tamanho mínimo contém quatro parcelas que consideram:

- O intervalo de tempo entre o envio de um pedido de retransmissão e a recepção deste pedido no nó ativo;
- O tempo de envio de uma cápsula de dados e a recepção no membro do grupo;
- O tempo de processamento;
- O espaço para duas cápsulas de dados: a cápsula perdida e a cápsula seguinte.

$$Max = Min + ((K - 1) \times ((Min - 2 \times E[Data_{size}]) + (t_{ret} - Lat) \times BW)) \tag{3.7}$$

$$P_{perda}^{K} \leq Const \tag{3.8}$$

onde,

$$
\begin{aligned}
t_{ret} &= \text{Tempo de estouro dos temporizadores de retransmissão} \\
P_{perda} &= \text{Probabilidade de perda em um enlace} \\
Const &= \text{Probabilidade de perda máxima definida pelo usuário}
\end{aligned}
$$

O cálculo do tamanho máximo permitido pelo cache é dado pelo número de vezes consecutivas (K) que um mesmo número de seqüência pode ser retransmitido por um nó ativo e perdido. Assim, o valor máximo depende do tamanho mínimo do cache, do intervalo entre pedidos de retransmissão (t_{ret}) para um mesmo número de seqüência e do valor de K. O valor de t_{ret} é trazido como parâmetro nas cápsulas de inscrição e deve ser diminuído de Lat para não sobrepor com o tempo considerado pelo mínimo. O valor de K depende da probabilidade de perda de uma cápsula de dados (P_{perda}) em uma determinada interface de saída de um nó. A equação 3.8 calcula o valor de K baseada na probabilidade de perda desejada e determinada pela aplicação ($Const$).

O valor do tamanho máximo e mínimo do cache de um grupo é calculado todas as vezes que uma cápsula de inscrição é recebida em um nó ativo. Os novos valores do máximo e do mínimo são comparados com os existentes e se forem maiores substituem os anteriores. Desta forma, usuários mais distantes do nó ativo não são prejudicados se aderirem mais tarde ao grupo multiponto. Outra observação importante diz respeito a variação dos valores de máximo e mínimo de acordo com a variação dos tráfego no interior da rede, pois com o aumento deste tráfego, os atrasos utilizados para os cálculos também aumentam.

$$
\begin{aligned}
RetBW &= (\#rets \times E[Data_{size}])/Intervalo \qquad (3.9) \\
Se &\quad (RetBW > (F \times DispBW)) \ e \ (Cache_{size} < Max) \\
&\quad Cache_{size} = Cache_{size} + (RetBW - (F \times DispBW)) \times Intervalo
\end{aligned}
$$

onde,

$$
RetBW = \text{Banda passante utilizada para retransmissões}
$$

$$\begin{aligned}
\#rets &= \text{Número de retransmissões em um período igual a } Intervalo \\
F &= \text{Fator de ponderação} \\
DispBW &= \text{Banda passante disponível} \\
Cache_{size} &= \text{Tamanho corrente do cache}
\end{aligned}$$

Durante a transmissão de dados para o grupo multiponto, o tamanho do cache de um nó ativo varia de acordo com as condições da rede entre os tamanhos mínimo e máximo correntes. Esta variação depende do número de pedidos de retransmissão que não são atendidos no nó ativo e requisitam um número de seqüência nos níveis mais altos da árvore multiponto ($\#rets$). O procedimento das cápsulas de pedido de retransmissão é responsável por contabilizar o número de pedidos não atendidos e aumentar o tamanho do cache segundo a política determinada no algoritmo 3.9. A cada período de tempo igual a $Intervalo$, verifica-se a banda passante imposta pelas retransmissões não atendidas dentro deste período ($RetBW$). Se $RetBW$ for maior do que uma determinada fração (F, onde $0 < F \leq 1$) da diferença entre a banda passante disponível e utilizada ($DispBW$), então aumenta-se o tamanho do cache. Observa-se que existe um compromisso entre a eficiência do mecanismo e a economia de recursos de memória em um nó ativo na escolha do valor de F, pois quanto menor for, mais freqüente é o aumento do tamanho do cache e, portanto, maior é o consumo médio de memória.

O tamanho do cache diminui suavemente conforme o número de pedidos de retransmissão não atendidos diminui. A cada período de tempo equivalente a $Intervalo$, se não houverem pedidos de retransmissão não atendidos, então o tamanho do cache é decrementado de um valor igual a $E[Data_{size}]$ (média do tamanho de uma cápsula de dados). Esta média é calculada pelo procedimento das cápsulas de dados multipontos a cada passagem de uma cápsula. O objetivo de diminuir suavemente o tamanho do cache é atingir um tamanho que mantenha a taxa de retransmissões não atendidas abaixo do nível que causaria congestionamentos nos enlaces dos níveis da árvore multiponto acima de um determinado nó ativo, mas sem utilizar recursos de memória em excesso para o cache.

Capítulo 4

Resultados de Simulação

NESTE capítulo são discutidos os resultados obtidos por simulações. Primeiro, são apresentadas a ferramenta de simulação utilizada e as extensões implementadas para este simulador. Em seguida, o modelo de simulação é descrito, sendo apresentadas algumas características assumidas para o ambiente que foi simulado e alguns detalhes relacionados à movimentação das estações móveis e a utilização de cache de dados nos nós intermediários. Por último, são analisados os resultados obtidos, sendo divididos em duas categorias, de acordo com o tipo de topologia utilizada [33, 34].

4.1 O Simulador de Redes *ns*

O *ns* (*N*etwork Simulator) é um simulador de redes orientado a eventos discretos que vem sendo desenvolvido por uma cooperação, denominada projeto VINT (*Virtual InterNetwork Testbed*). O principal objetivo desta cooperação é o desenvolvimento de ferramentas para a simulação e a análise de serviços de rede voltados principalmente para a Internet [36, 37].

No *ns*, a execução de uma simulação depende de uma configuração de entrada definida por um cenário de simulação. Os cenários são constituídos por vários componentes: uma topologia de rede, incluindo a interconexão física entre os nós e as características estáticas dos nós e enlaces; os modelos de tráfego, que definem os padrões de uso da rede e a localização de transmissores e receptores; a geração de cenários de teste, que criam eventos tais como a distribuição de um grupo multiponto, isto é, a adesão e a saída de receptores; e a dinâmica da rede, que corresponde a ocorrência de falhas em nós e enlaces da rede [36].

As topologias utilizadas nas simulações podem ser criadas, ou escolhidas de um conjunto disponível, que oferece desde topologias simples, até topologias de redes reais em operação. A criação de topologias pode ser realizada através do uso de ferramentas destinadas a geração de topologias aleatórias, como o GT-ITM (*Georgia Tech - Internetwork Topology Model*). O GT-ITM é capaz de gerar aleatoriamente topologias que possuem estruturas de domínios semelhantes às encontrada na Internet, denominadas topologias do tipo *transit-stub* (T-S). Por definição, em um domínio *stub* trafegam somente pacotes destinados ou originados neste domínio, enquanto que em domínios *transits* não existe esta restrição. O propósito dos domínios *transits* é interligar os domínios *stubs* [38, 39]. A figura 4.1 ilustra uma topologia deste tipo. Para avaliar o serviço proposto nesta tese, topologias T-S foram utilizadas.

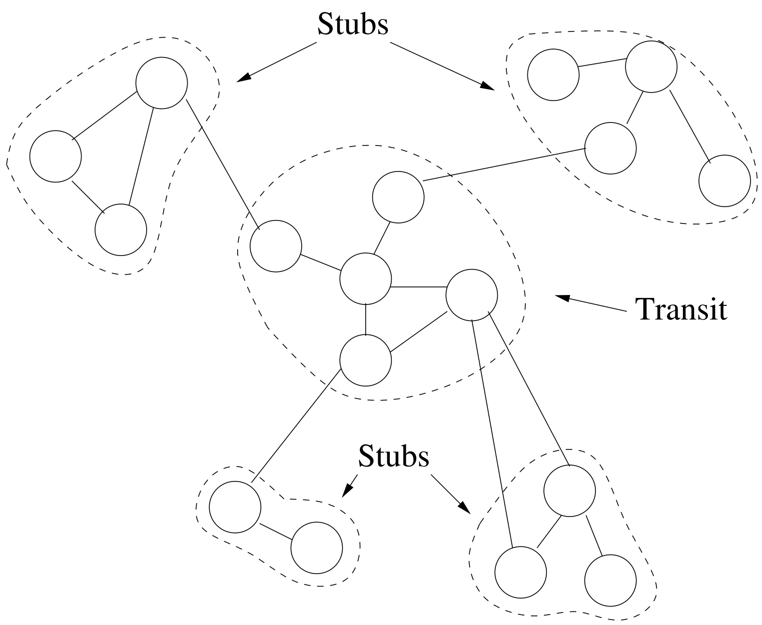


Figura 4.1: Exemplo de uma topologia T-S.

Nesta tese, os nós são ativos, ou seja, eles são capazes de executar os procedimentos de cápsulas inseridos pelas aplicações na rede. Como a versão original do *ns* não possui suporte a redes ativas, foi necessária a implementação de extensões a fim de possibilitar a simulação do serviço proposto no capítulo 3. Assim, nós ativos capazes de executar os procedimentos de cápsulas foram implementados e o conjunto de recursos necessários para o serviço, descrito na seção 3.1, foi disponibilizado para a programação dos procedimentos. Além dos nós ativos, as entidades de extremidade, denominadas nesta tese de agentes ativos, foram implementadas com o objetivo de gerar e consumir cápsulas, segundo o papel associado a cada uma, que pode ser de transmissor ou de receptor.

Mais detalhes a respeito da estrutura básica do *ns* e da implementação realizada podem ser encontrados no apêndice A. A descrição dos parâmetros necessários para a geração de topologias com o GT-ITM e os valores utilizados são apresentados no apêndice B.

4.2 O Modelo de Simulação

O objetivo das simulações é fornecer dados que permitam a avaliação das funcionalidades propostas para o serviço. Assim, para esta avaliação, foram realizadas simulações considerando o movimento de estações móveis em redes celulares, tendo sido necessária para isto a definição da estrutura da rede e a caracterização do movimento das estações. Algumas considerações foram feitas também para o mecanismo de recuperação de perdas, que incluem a determinação do tempo de estouro dos temporizadores e algumas simplificações para os cálculos do tamanho do cache de dados.

4.2.1 As Redes Celulares

A estrutura básica de rede utilizada foi apresentada na seção 3.1, onde existe o núcleo da rede formado por nós ativos e à periferia estão ligadas as redes celulares.

As estações móveis encontram-se nas redes celulares, movimentando-se dentro destas redes, e comunicando-se com o restante da rede através dos nós ativos celulares.

As estações, conforme se movimentam e cruzam as fronteiras entre as células, passam da célula atual para uma das células vizinhas. Nas simulações realizadas, não foi considerada a troca de rede celular, confinando o movimento de uma estação a rede celular onde iniciou o seu movimento. A figura 4.2 ilustra redes celulares com 3, 4 e 5 células e o respectivo movimento de uma estação móvel dentro das redes, como foi utilizado nas simulações. No início da simulação, a célula de onde parte uma estação móvel é escolhida aleatoriamente entre as células que compõem a rede.

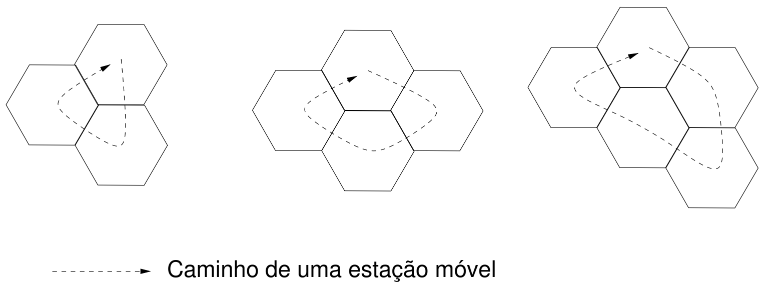


Figura 4.2: Disposição das células.

O deslocamento das estações móveis foi simulado através do movimento intercelular realizado por estas estações. Para isto, foram utilizados períodos entre dois *handoffs* de uma mesma estação obtidos aleatoriamente. A média deste período é fornecida no início da simulação, tendo sido utilizado o intervalo 4.1, quando a média fornecida é maior ou igual a $0.2s$ e o intervalo 4.2, quando a média é menor que $0.2s$, em ambos casos considerando uma distribuição de probabilidades uniforme. Assim, a velocidade máxima que poderia ser atingida por uma estação se utilizando deste serviço depende das dimensões das células. Estas dimensões são representadas nas simulações pelo atraso de propagação dos enlaces sem fio, pois o tempo entre o envio de dados por uma estação e a recepção no nó ativo celular depende da distância entre os dois. Para estas simulações, foi escolhido um atraso de propagação de $10\mu s$ para os enlaces sem fio, o que corresponde a células de $6km$ de diâmetro, considerando-se C $(3 \times 10^8 m/s)$ como sendo a velocidade de propagação e $10\mu s$ como o atraso máximo. Comparado ao atraso de transmissão para a banda passante escolhida para os enlaces sem fio, que foi de $2Mbits/s$, o atraso de propagação pode ser desprezado. Sendo assim, os resultados obtidos com este atraso de propagação são válidos também para células com o diâmetro entre $0.5km$ e $30km$, que são valores adotados atualmente pelas ferramentas de auxílio ao desenvolvimento de redes celulares [40]. Para valores maiores de banda passante, o atraso de propagação não seria desprezível se comparado ao atraso de transmissão e, portanto, os resultados estariam restritos a células com diâmetros em torno de $6km$.

$$[\text{Média} - 0.2s, \text{Média} + 0.2s] \tag{4.1}$$

$$[\text{Média} - 0.02s, \text{Média} + 0.02s] \tag{4.2}$$

A estratégia de movimento intercelular sem interrupção (*seamless handoff*) foi considerada para as simulações [41]. Nesta estratégia, uma estação ao decidir por um handoff estabelece uma conexão com o NAC responsável pela próxima célula, mantendo ativa a conexão com o NAC anterior. Ao completar o estabelecimento da conexão com o novo NAC, a conexão nova é ativada e a antiga é liberada. Desta forma, as estações móveis não permanecem sem comunicação física, mas perdem dados em trânsito.

4.2.2 O Mecanismo de Recuperação de Perdas

Os temporizadores de retransmissão disparados pelos receptores têm como função avisar quando um novo pedido de retransmissão deve ser enviado, caso o pedido anterior para um número de seqüência não tenha sido atendido. Para as simulações realizadas, o período de estouro inicial foi considerado fixo, tendo sido utilizados os seguintes valores: $0.08s$ para as topologias em árvore e $0.16s$ para as topologias *transit-stub*. Estes valores foram escolhidos a partir da observação dos valores máximos de RTT nas topologias e da multiplicação destes valores por quatro. Outro temporizador utilizado é o de silêncio, que se propõe a garantir que os receptores recebam a cápsula de fim de sessão. No cálculo do tempo de estouro deste temporizador, determinou-se que a média inicial é $0.5s$ e que o fator de ponderação da média (Γ) é cinco (5), pois, com este fator, obtém-se uma pequena variação da média na ocorrência de rajadas.

Com relação ao cache de dados existente nos nós ativos para atender aos pedidos de retransmissão, as simulações podem ser divididas em três categorias: cache fixo, cache adaptativo sem movimentação das estações e cache adaptativo com movimentação das estações. Para as simulações com cache fixo, o espaço de armazenamento destinado a esta função é configurado no início da simulação para cada nó e não é alterado durante o tempo de simulação, não se aproveitando da possibilidade de adaptação às condições da rede fornecida pela alocação adaptativa. As simulações utilizando cache adaptativo foram realizadas, primeiro, sem considerar o movimento das estações, visando a avaliação deste mecanismo, enquanto que as simulações considerando o movimento das estações tiveram como finalidade a avaliação do serviço como um todo.

O cálculo do tamanho do cache nos nós ativos foi descrito na seção 3.4. No entanto, como as simulações foram realizadas considerando cápsulas de dados com tamanho da ordem de *kbytes*, enlaces com banda passante da ordem de $Mbits$ e latência da ordem de ms, as duas parcelas finais no cálculo do mínimo, que consideram o tempo de processamento e o espaço para os dados de duas cápsulas, podem ser desprezadas, pois os dois primeiros foram considerados como sendo muito maiores que os dois últimos. Além disto, BW é a banda passante total disponível para a aplicação, o que garante um espaço extra de armazenamento quando é alocado um espaço com o tamanho mínimo, pois uma aplicação não necessariamente utiliza toda a banda passante disponível para a transmissão dos seus dados. Se o espaço de armazenamento não for suficiente, um aumento do tamanho do cache é negociado com o nó, conforme está descrito no algoritmo 3.9, tendo sido utilizados os valores 0.05 e 0.8 para as constantes $Const$ e F, respectivamente. Assim, para o cálculo do valor mínimo foi utilizada a equação 4.3.

$$Min = Lat \times BW \times (1 + \frac{E[Data_{size}]}{Subscribe_{size}}) \tag{4.3}$$

Para o cálculo do máximo, como foi desprezado o espaço para duas cápsulas de dados no cálculo do mínimo e como o valor utilizado para t_{ret} é muito maior que Lat, estes dois fatores não foram considerados. Desta forma, estas simplificações implicam em algumas alterações no cálculo do máximo, sendo que as versões alteradas das equações são mostradas em 4.4 e 4.5.

$$Max = K \times (Min + t_{ret} \times BW) \tag{4.4}$$

$$P_{perda}^{K} \leq Const \tag{4.5}$$

4.3 Resultados para Topologias em Árvore

Os resultados apresentados nesta seção são de simulações realizadas sobre árvores cheias, que são árvores em que todos os nós possuem o mesmo número de filhos, exceto as folhas que não possuem filhos, sendo que o número de filhos corresponde ao grau da árvore. Assim, mantendo o tamanho do cache fixo, estas topologias foram utilizadas para verificar a eficiência dos filtros presentes nos procedimentos de inscrição e pedidos de retransmissão conforme varia a dispersão dos usuários.

Outras simulações em uma topologia em forma de árvore foram realizadas com o objetivo de verificar a quantidade de retransmissões atendidas pelos nós ativos intermediários e a economia de recursos obtida quando utiliza-se o mecanismo de cache adaptativo. No entanto, estas simulações tiveram como intuito a averiguação do uso deste mecanismo e, portanto, não consideraram o movimento das estações integrantes do grupo.

4.3.1 Utilizando Cache Fixo

Foram utilizadas árvore cheias, com o grau variando entre um e quatro e o número de níveis fixo em três. Em cada folha destas árvores foram ligados quatro nós ativos celulares, cada um representando uma célula. Cada conjunto deste de

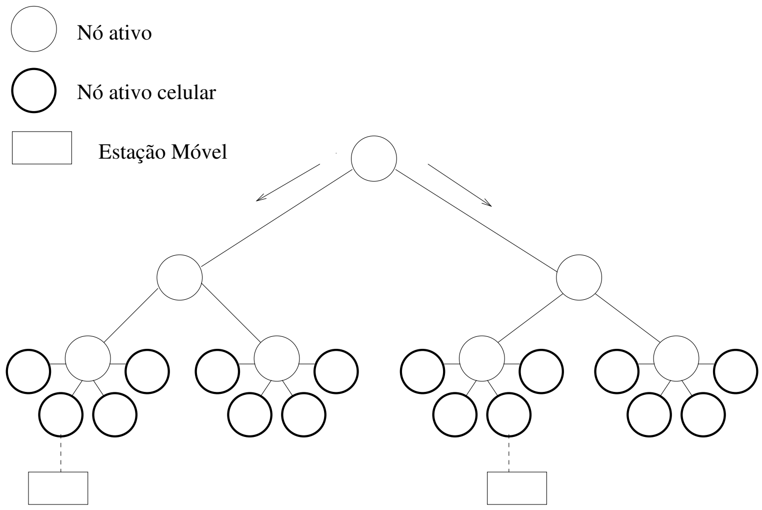
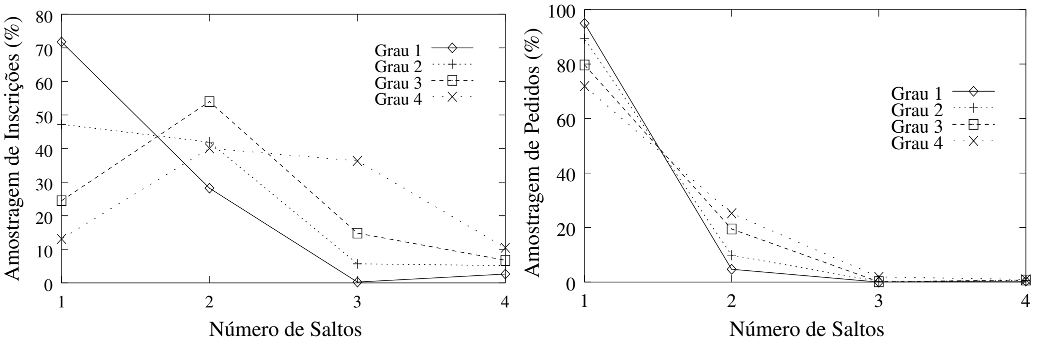


Figura 4.3: Exemplo da topologia em árvore utilizada com grau 2.

quatro nós celulares formam uma rede celular independente, sendo o movimento simulado de cada estação móvel restrito a uma destas redes com uma média de $0.6s$ entre handoffs consecutivos de uma mesma estação. Para cada árvore, o número de usuários foi fixado em 32, sendo este número distribuído igualmente entre as redes celulares existentes na árvore. Conforme o grau da árvore aumenta, o número de usuários do grupo multiponto por rede celular diminui. Um exemplo da árvore utilizada com grau 2 é apresentado na figura 4.3. A raiz da árvore é o transmissor do grupo multiponto, tendo sido utilizada uma fonte de tráfego CBR (*Constant Bit Rate*) com uma taxa de 50 cápsulas de 1000 bytes por segundo ($400kbits/s$). Para os enlaces sem fio, foi considerada uma banda passante de $2Mbits/s$, de forma a ser compatível com as propostas para as futuras redes sem fio e celulares [42, 40], e ainda uma taxa de erro de 10%. A banda passante dos enlaces sem fio foi considerada como sendo de $1Mbit/s$ para representar uma banda limitada pelo compartilhamento entre várias aplicações no núcleo da rede. Como a taxa de erro nos enlaces fixos é baixa, considerou-se uma taxa de 1%, sendo possível ainda perdas durante a ocorrência de congestionamentos.

A figura 4.4 mostra medidas feitas para a quantidade de cápsulas de inscrição e pedidos de retransmissão que atingem cada nível, dos receptores para o transmissor, medido em número de saltos. A figura 4.4 mostra as medidas obtidas para cápsulas



(a) Inscrição. (b) Pedido de Retransmissão.

Figura 4.4: Filtragem de cápsulas.

de inscrição, podendo-se observar o efeito da filtragem de acordo com a dispersão dos usuários. Quanto menos dispersos estiverem os usuários maior será o efeito da filtragem, ou seja, quanto menor for o grau da árvore, existirão mais usuários por rede celular e mais os nós ativos intermediários serão capazes de filtrar. Isto ocorre porque o nível de compartilhamento do caminho na árvore multiponto é maior para as árvore de menor grau.

A figura 4.4 mostra as medidas obtidas para cápsulas de pedidos de retransmissão. Como a maior parte das perdas ocorre devido às taxas de erro nos enlaces sem fio e à movimentação dos receptores, a maior parte da amostragem realizada percorre somente até o segundo nível da árvore, ou seja, realiza dois saltos. A diferença que aparece entre as curvas de árvores com grau diferente existe devido a dispersão dos usuários. Ao trocar de célula, uma estação móvel pode perder as cápsulas de dados em trânsito. Ao detectar a perda, um pedido de retransmissão é realizado e, assim, se outra estação já estiver presente na nova célula, o pedido será prontamente respondido pelo NAC responsável por esta célula. Portanto, para as árvores de menor grau, com menor dispersão dos usuários, a maior parte dos pedidos de retransmissão já é atendido no primeiro nível da árvore que corresponde aos nós ativos celulares.

4.3.2 Utilizando Cache Adaptativo com Usuários Fixos

Nestas simulações, considerou-se também a raiz da árvore como sendo o transmissor do grupo multiponto, com uma fonte de tráfego CBR de $533kbits/s$ (cápsulas de dados de 1000 bytes com intervalos de $0.015s$), e as folhas são os nós aos quais estão ligados os receptores, sendo que o grau da árvore escolhido foi três (3), pois fornece uma dispersão suficiente dos usuários do grupo multiponto para as medidas desejadas. A topologia utilizada está ilustrada na figura 4.5.

Para cada simulação, a posição dos membros do grupo foi escolhida aleatoriamente, tendo sido mantido constante o número de membros do grupo em 40. Para estes usuários, assume-se uma banda passante disponível de $1Mbit/s$, sendo este o valor atribuído a cada enlace. Perdas aleatórias foram consideradas para simular erros nos enlaces e congestionamentos, utilizando-se taxas de erro de 20%. No entanto, estas taxas de erro não foram associadas simultaneamente a todos os enlaces. Simulações foram realizadas considerando-se estas taxas de erro em apenas um nível por vez, ou seja, como está ilustrado na figura 4.5, apenas os enlaces de um dos quatro níveis marcados pelas linhas pontilhadas possuem taxas de erro de 20%, associando aos restantes taxas de erro de 0.25%. Os resultados para estas simulações estão apresentados nas figuras 4.6, 4.7 e 4.8.

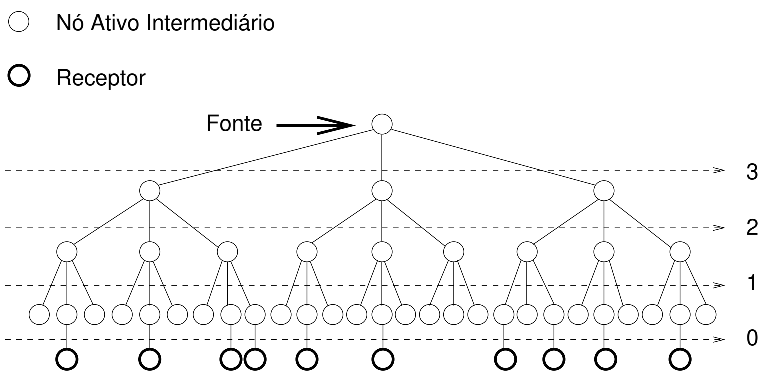
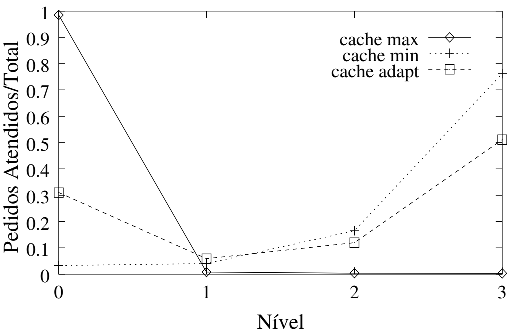
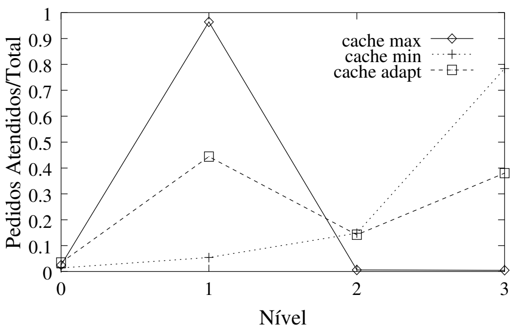


Figura 4.5: Árvore cheia com grau 3.

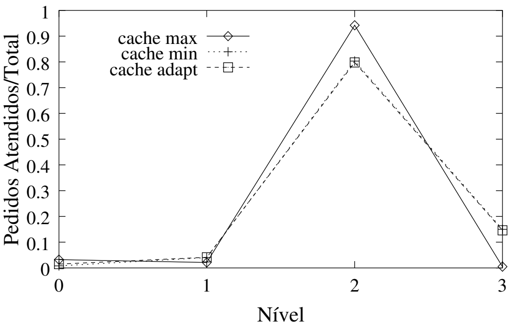
A figura 4.6 apresenta quatro gráficos que mostram a proporção de pedidos de retransmissão atendidos nos nós ativos de cada nível da árvore, sendo que cada gráfico considera taxas de erro de 20% em um nível diferente. Além das medidas



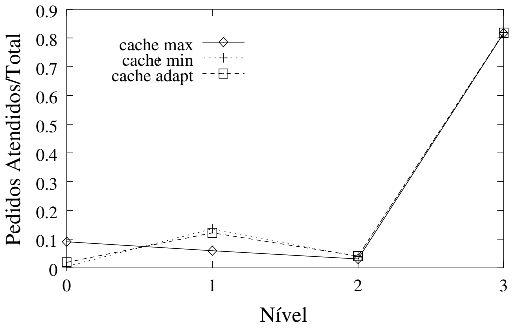
(a) Nível 0.



(b) Nível 1.



(c) Nível 2.



(d) Nível 3.

Figura 4.6: Retransmissões na topologia em árvore.

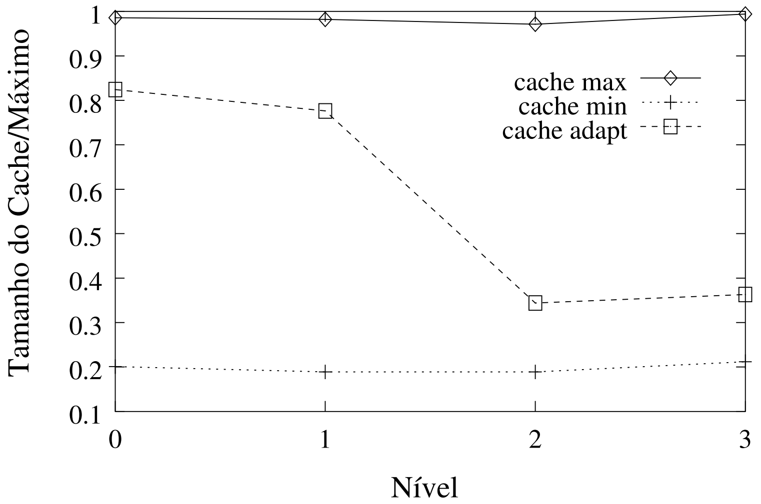


Figura 4.7: Utilização do cache com relação ao tamanho máximo negociado.

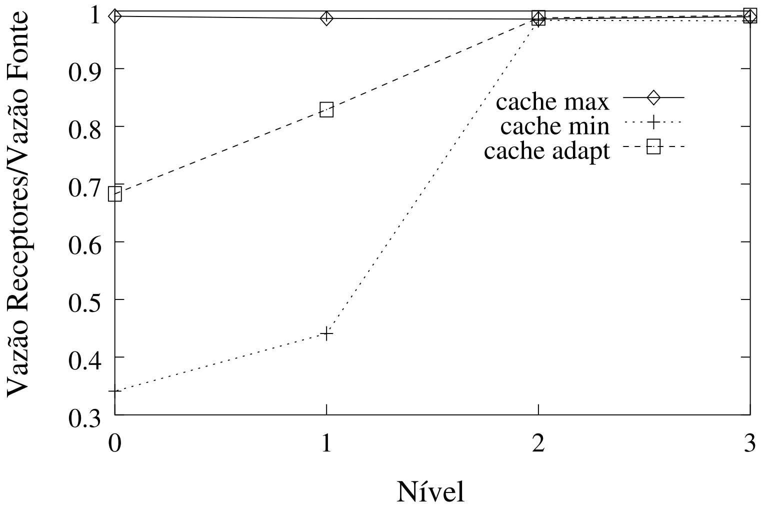


Figura 4.8: Vazão média obtida para os receptores.

utilizando cache adaptativo, foram realizadas medidas considerando alocações pelo mínimo e pelo máximo. Nestas medidas, o tamanho do cache não varia entre o mínimo e o máximo calculados, ou seja, o tamanho é sempre alocado pelo mínimo ou pelo máximo. Observa-se nestes gráficos que nos níveis mais baixos (0 e 1), como existem mais enlaces gerando perdas aleatoriamente, mais pedidos de retransmissão para números de seqüência diferentes serão enviados, ocasionando um aumento no uso de recursos de memória nos nós ativos dos níveis mais altos que recebem uma grande quantidade de pedidos de retransmissão, para o caso de cache adaptativo.

No entanto, para os nós ativos dos níveis mais baixos, como o número de pedidos de retransmissão não é elevado, a quantidade de recursos utilizada não é grande e, por conseqüência, o número de pedidos atendidos é menor, se for comparado com a alocação pelo máximo. Para os dados dos níveis mais altos (2 e 3), observa-se que as três curvas se aproximam, mostrando que a quantidade de recursos de memória necessária é menor devido ao menor número de pedidos de retransmissão para números de seqüência diferentes. Assim, o mecanismo de alocação adaptativo economiza recursos nesta situação.

O gráfico apresentado na figura 4.7 apresenta a proporção média de utilização do cache com relação ao máximo para os três casos e para taxas de erro em cada nível. Para o cache adaptativo, observa-se que para os níveis mais baixos a utilização de memória é maior, mas, ainda assim, não atinge o valor máximo. Por outro lado, nos níveis mais altos, a utilização de memória pelo cache se aproxima do mínimo. Desta forma, a economia média de recursos utilizando cache adaptativo para este caso varia entre 28-66 %.

O gráfico de vazão da figura 4.8 mostra que para os níveis mais baixos, a alocação pelo mínimo e a alocação adaptativa não foram suficientes para evitar uma queda no desempenho, apesar das medidas de vazão obtidas para a alocação adaptativa terem sido maiores que as medidas obtidas para a alocação pelo mínimo. Entretanto para os níveis mais altos, observa-se que a vazão obtida para os três casos é aproximadamente a mesma, mostrando que a alocação pelo máximo desperdiça recursos para estes casos.

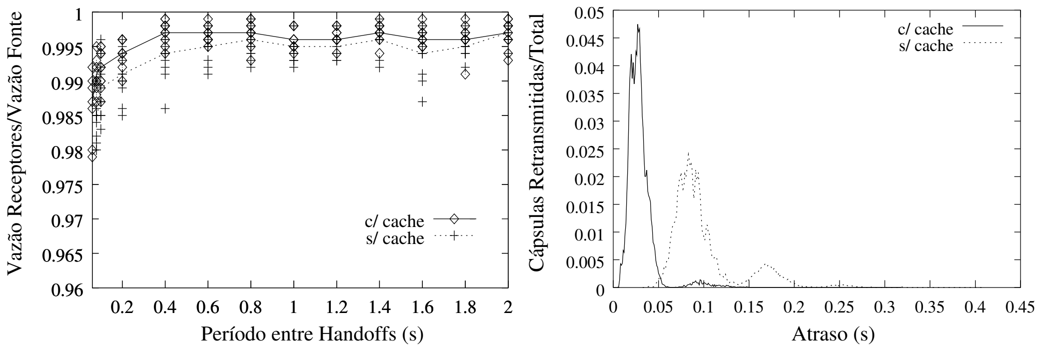
4.4 Resultados para Topologias *Transit-Stub*

As topologias do tipo *transit-stub* permitem simulações em estruturas semelhantes às encontradas na Internet e foram utilizadas para avaliar o desempenho do serviço proposto diante destas estruturas. Para estas simulações, considerou-se que os domínios *stubs* são as redes celulares, e as estações móveis são nós ligados a estas redes. Assim, quatro seqüências de simulações foram realizadas visando observar a influência do uso de cache nos nós intermediários, o desempenho com o aumento do número de usuários e a diferença no uso de cache fixo e adaptativo para o serviço.

4.4.1 Influência do Cache na Comunicação

Algumas simulações foram realizadas para determinar a influência do uso de cache em nós intermediários para retransmissões. Para estas simulações considerou-se enlaces com atrasos de propagação (proporcionais ao comprimento dos enlaces) no intervalo $[0.0001s, 0.01s]$. A taxa de transmissão escolhida para os enlaces sem fio foi de $2Mbits/s$, considerando-se para simular interferências nestes enlaces, uma taxa de perda de cápsulas de 10%. Nos enlaces fixos, a taxa de perda de cápsulas foi desconsiderada, pois assumiu-se que as perdas por interferência nestes enlaces é desprezível se comparada às perdas nos enlaces sem fio. No entanto, perdas devido a congestionamentos podem ocorrer se a banda passante requisitada ultrapassar a disponível e houver transbordo nas filas das interfaces dos nós. A taxa de transmissão escolhida para os enlaces fixos foi de $1Mbit/s$, pois deseja-se simular uma situação na qual a banda passante total destes enlaces não esteja completamente disponível para uma aplicação, devido ao compartilhamento do meio com outras aplicações. Portanto, $1Mbit/s$ é a taxa de transmissão disponível no interior da rede para a aplicação simulada.

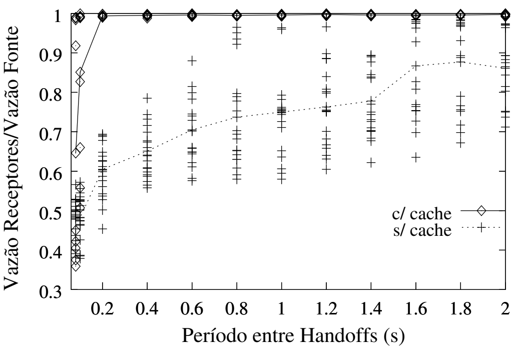
A vazão média obtida para um grupo multiponto foi medida, variando-se o período médio entre handoffs, isto é, a freqüência média com que os usuários trocam de célula. Para isto, foram consideradas sessões utilizando cache com tamanho fixo de $6kbytes$ por nó ativo (exceto as estações móveis) e outras sessões sem cache nos nós intermediários. Uma fonte de tráfego CBR foi utilizada com duas taxas de geração de dados diferentes abaixo de $1Mbit/s$: $400kbits/s$ (1 cápsula a cada $0.02s$, $1000bytes/cpsula$) e $533kbits/s$ (1 cápsula a cada $0.015s$, $1000bytes/cpsula$). A figura 4.9 apresenta os resultados para o tráfego de $400kbits/s$. Na figura 4.9.a, a vazão obtida com cache é ligeiramente superior se comparada à vazão obtida sem cache, devido às retransmissões mais rápidas. Para ambos os casos, a vazão média é aproximadamente constante no intervalo $[0.2s, 2s]$, diminuindo para valores abaixo de $0.2s$ conforme a geração de perdas se aproxima da capacidade de recuperação do serviço. Como pode ser observado na figura 4.9.b, que mostra a distribuição do atraso nas retransmissões considerando todos os períodos entre handoffs simulados, todos os atrasos na recuperação de perdas ocorrem em um período de tempo inferior a $0.35s$, sendo a maior parte inferior a $0.1s$. Desta forma, para períodos de tempo maiores do que $0.35s$, uma perda devido a um handoff é recuperada antes da ocorrência de um novo handoff. Além disto, os erros causados por interferências nos enlaces sem fio não foram suficientes para causar um acúmulo de dados não recebidos neste caso.



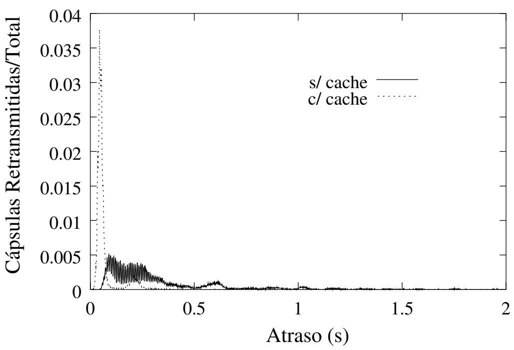
(a) Vazão. (b) Atraso nas retransmissões.

Figura 4.9: Fonte CBR de 400 kbits/s.

Utilizando a fonte de tráfego de $400kbits/s$, o serviço sem cache nos nós intermediários é capaz de recuperar cápsulas perdidas a tempo de evitar um acúmulo de dados não recebidos, apesar de obter uma vazão mais baixa. No entanto, para uma fonte de tráfego de $533kbits/s$, como é apresentado na figura 4.10.a, as estações móveis não são capazes de recuperar as perdas a tempo de evitar este acúmulo e, portanto, resulta em uma vazão média mais baixa que a vazão obtida para o serviço com cache. Na figura 4.10.b, a distribuição dos atrasos para a recuperação de perdas para o serviço sem cache mostra que existem atrasos de até $2s$ devido ao congestionamento causado pela grande número de retransmissões. Quando o serviço é utilizado com cache, no entanto, atrasos na retransmissão são ainda limitados por $0.35s$, permitindo que a vazão sature para períodos entre handoffs consecutivos superiores a 0.2 s.



(a) Vazão.



(b) Atraso na retransmissão.

Figura 4.10: Fonte CBR de 533 kbits/s.

Tabela 4.1: Proporção de atrasos de retransmissão abaixo do período médio entre handoffs consecutivos.

	CBR - 400 kbits/s		CBR - 533 kbits/s	
Handoff (s)	s/ cache (%)	c/ cache (%)	s/ cache (%)	c/ cache (%)
0.08	37.43	78.52	4.47	63.49
0.1	70.74	81.58	15.03	85.76
0.2	98.09	91.75	33.89	90.96
0.4	100.0	100.0	67.03	100.0
0.6	100.0	100.0	80.25	100.0
0.8	100.0	100.0	92.39	100.0
1.0	100.0	100.0	95.62	100.0
1.2	100.0	100.0	96.61	100.0
1.4	100.0	100.0	96.83	100.0
1.6	100.0	100.0	99.33	100.0
1.8	100.0	100.0	99.36	100.0
2.0	100.0	100.0	99.59	100.0

A tabela 4.1 fornece mais detalhes a respeito do atraso na recuperação de perdas. Cada valor nas colunas 2, 3 e 4 corresponde ao percentual de atrasos que está abaixo do período entre handoffs na coluna 1. Desta forma, pode-se observar que conforme o período entre handoffs aumenta e um número maior de retransmissões são atendidas neste período, a vazão média do grupo aumenta. Quando o percentual atinge 100%, todas as retransmissões causadas por perdas de dados em trânsito devido a um handoff são atendidas antes da ocorrência de outro handoff e, assim, a vazão se aproxima do máximo. Para o serviço sem cache e com a fonte de tráfego de 533 $kbits/s$, o percentual de atrasos obtido para o período entre handoffs de $2s$ não é de 100%, o que indica que ocorreu acúmulo de dados não recebidos. O mesmo não ocorre para a mesma fonte utilizando cache de dados nos nós intermediários, pois as retransmissões mais rápidas e localizadas evitam o congestionamento devido ao excesso de retransmissões.

4.4.2 Número de Usuários por Sessão

Para verificar o desempenho do serviço diante de um número grande de usuários em um grupo multiponto, simulações foram realizadas para grupos com até 140 usuários. Nestas simulações, o cache de um grupo em um nó intermediário foi considerado como tendo um tamanho fixo de $6kbytes$. Para os enlaces, as características assumidas foram as mesmas descritas na sub-seção anterior.

A figura 7 mostra um gráfico da variação da vazão com o aumento do número de usuários em uma sessão. Alguns parâmetros que foram mantidos fixos para todas as medidas foram: o período médio entre handoffs de $0.5s$ para cada estação móvel e o tráfego CBR gerado pelo transmissor com um intervalo entre envios de cápsula de $0.02s$, tendo sido utilizadas cápsulas de 1000 bytes, o que equivale a uma vazão de $400kbits/s$. Observa-se que a vazão se mantém aproximadamente constante devido às retransmissões localizadas e à ação dos filtros de cápsulas de inscrição e pedidos de retransmissão. O número de cápsulas de inscrição que é recebido pelo transmissor não aumenta com o número de usuários, pois somente chega ao transmissor a quantidade de cápsulas de inscrição necessária para manter as indicações nos *soft-state* dos nós ativos. Situação semelhante ocorre com as cápsulas de pedido de retransmissão, pois os pedidos de retransmissão duplicados são descartados, não aumentando com o número de usuários. Além disto, como a maior parte das retransmissões é feita somente para um sub-conjunto dos usuários, o aumento do tamanho do grupo não implica em um aumento do gasto de banda passante com retransmissões.

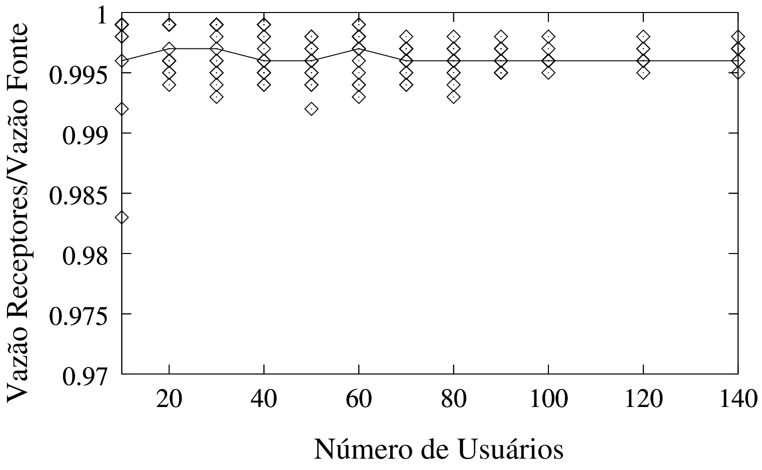


Figura 4.11: Número de usuários por sessão.

4.4.3 Utilizando Cache Adaptativo com Usuários Fixos

Quando o movimento das estações móveis não é considerado, não ocorrem perdas de dados em trânsito. Desta forma, as perdas ocorrem devido a erros nos enlaces e, possivelmente, devido a congestionamentos, sendo possível avaliar o desempenho do mecanismo de alocação adaptativo sem a influência das perdas causadas pelos handoffs.

Assim, para as simulações realizadas nesta sub-seção, a banda passante e a taxa média de perdas dos enlaces no interior da rede foram consideradas como sendo $1.5 Mbits/s$ e 10%, respectivamente, enquanto que os atrasos de propagação foram gerados aleatoriamente pelo GT-ITM, estando compreendidos no intervalo $[1.1ms, 4.6ms]$.

Os receptores do grupo multiponto são nós ligados às sub-redes (*stubs*). Um número fixo de cinco (5) usuários por sub-rede foi determinado, escolhendo-se aleatoriamente o nó da sub-rede ao qual cada usuário está ligado, totalizando um total de 40 usuários. Os enlaces que ligam os receptores aos nós da sub-rede possuem a mesma banda passante de $1.5 Mbits/s$, mas com atraso de propagação de $0.1ms$ e taxas médias de erro de 20%. Uma taxa média de erro mais elevada foi considerada para os enlaces que ligam os receptores para averiguar o comportamento do mecanismo proposto diante de enlaces sem fio, que possuem como característica elevadas taxas de perdas.

Para avaliar o funcionamento do cache adaptativo, medidas foram realizadas utilizando uma fonte de tráfego modelada por um processo de Poisson [43], isto é, a fonte utiliza uma variável aleatória com distribuição de probabilidades exponencial para gerar os intervalos entre envios de cápsulas de dados multipontos. A posição da fonte foi escolhida aleatoriamente entre os nós do núcleo para, tendo sido utilizadas diferentes taxas médias de geração de dados. Medidas para comparação foram igualmente realizadas considerando-se alocações pelo máximo e pelo mínimo.

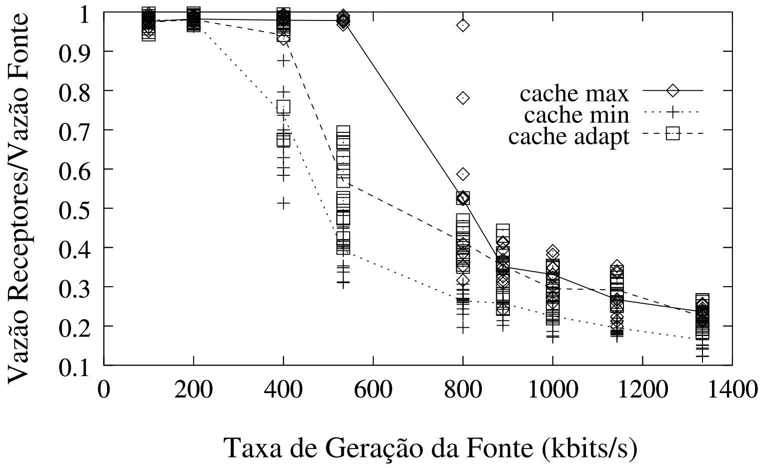


Figura 4.12: Vazão obtida variando a taxa média de geração da fonte.

O gráfico da figura 4.12 apresenta a razão entre a vazão média dos receptores e a vazão média da fonte de tráfego obtida para os três mecanismos de alocação de cache. Observa-se por este gráfico que até taxas médias de geração de dados de $200kbits/s$, os três mecanismos atingem a vazão máxima. Neste caso, como pode ser observado no gráfico da figura 4.13 que mostra a razão entre a média do tamanho do cache e o tamanho máximo calculado, a alocação pelo máximo desperdiça recursos, pois o cache adaptativo obtém o mesmo resultado utilizando uma quantidade próxima do mínimo. Entre $200kbits/s$ e $900kbits/s$, a vazão cai, sendo que a alocação pelo máximo mantém a maior vazão, pois aloca previamente a quantidade máxima. Para

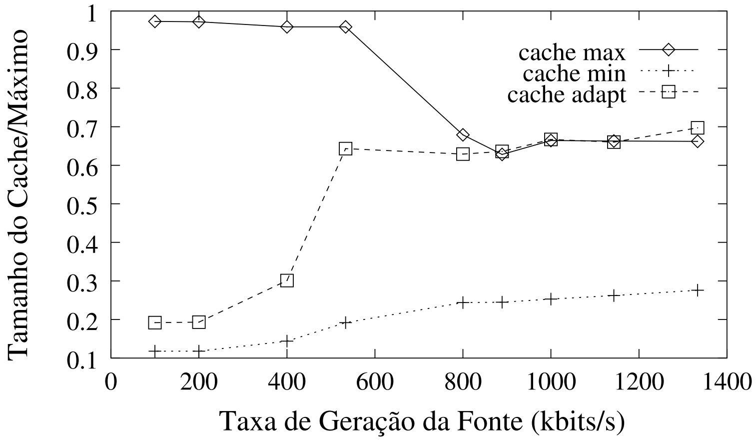
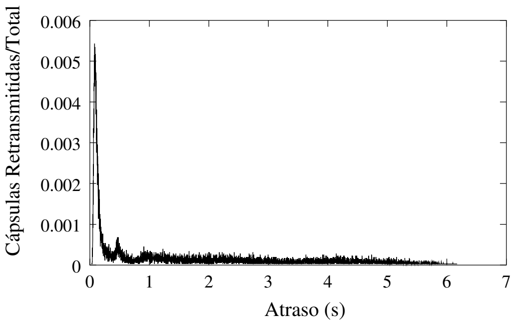
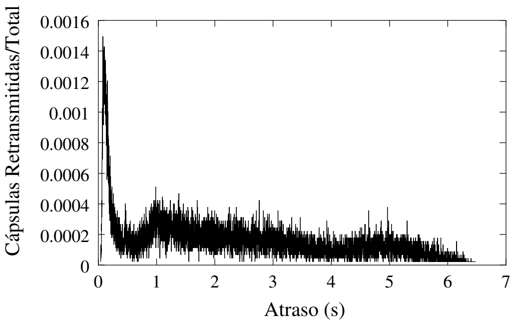


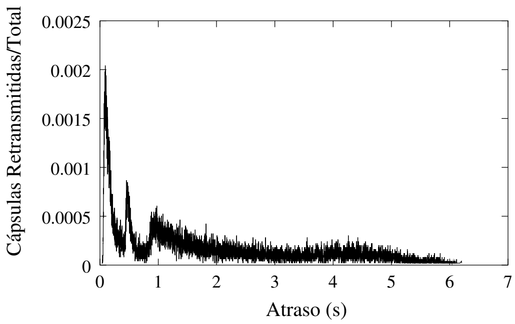
Figura 4.13: Utilização do cache com relação ao máximo negociado.



(a) Alocação pelo máximo.



(b) Alocação pelo mínimo.



(c) Cache Adaptativo.

Figura 4.14: Distribuição dos atrasos das retransmissões.

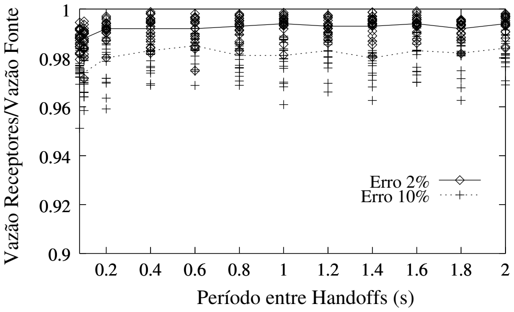
as taxas médias de geração neste intervalo, a quantidade de recursos utilizada pelo cache adaptativo cresce até saturar em $533 kbits/s$, que ocorre devido ao tamanho do cache que atinge o máximo. A partir deste mesmo valor, o uso de cache para a alocação pelo máximo decresce devido ao surgimento de congestionamentos no sentido da fonte causados pelo grande número de pedidos de retransmissão e o conseqüente aumento nos valores do cache mínimo e máximo calculados durante a simulação, que acarretam num valor médio menor do que o máximo valor observado. Para taxas médias de geração acima de $900kbits/s$, a vazão obtida para o cache adaptativo e a alocação pelo máximo são aproximadamente iguais, utilizando desta forma a mesma quantidade de recursos de memória para o cache.

A figura 4.14 mostra a distribuição do atraso dos pacotes retransmitidos durante as simulações realizadas. A alocação pelo máximo, por alocar uma quantidade maior de recursos mesmo quando não há necessidade, consegue realizar retransmissões mais rápidas (figura 4.14.a). Ao contrário, a alocação pelo mínimo, por alocar sempre o mínimo de recursos necessários, apresenta maiores atrasos nas retransmissões (figura 4.14.b). A utilização do mecanismo de cache adaptativo procura manter o cache com um tamanho capaz de realizar retransmissões o mais rápido possível, mas mantendo o compromisso de não desperdiçar recursos. Assim, a distribuição do atraso para as retransmissões apresentada na figura 4.14.c se encontra entre os casos de alocação pelo máximo e alocação pelo mínimo.

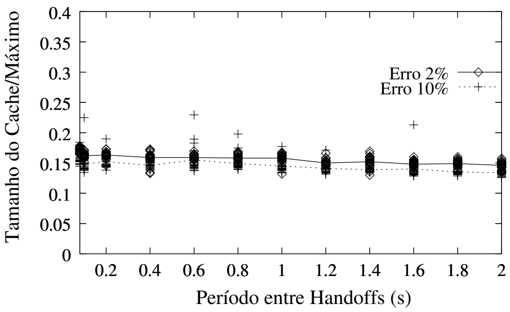
4.4.4 Utilizando Cache Adaptativo com Usuários Móveis

Após avaliar o serviço multiponto utilizando o mecanismo de alocação adaptativo para o cache dos nós intermediários sem considerar a mobilidade das estações, o passo seguinte foi verificar o desempenho do serviço considerando o cenário completo. Assim, duas seqüências de simulações foram realizadas com condições diferentes, supondo uma banda passante de $2Mbits/s$ e taxa de perda de 10% para os enlaces sem fio. Primeiro, foi considerado que a limitação de banda passante está no interior da rede devido ao compartilhamento com usuários de outras aplicações, supondo assim topologias com enlaces fixos $1.5Mbits/s$ (T1) [35]. Para a segunda seqüência, assumiu-se que a limitação de banda passante estaria nos enlaces sem fio, e por isso supôs-se enlaces fixos de $155Mbits/s$ (OC-3) [35].

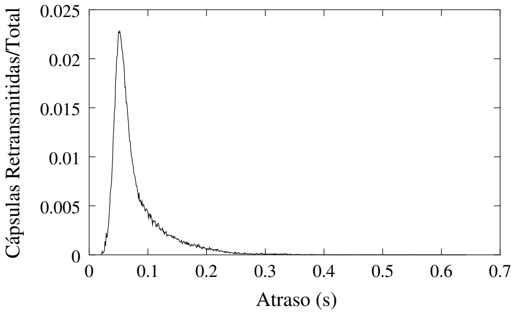
Os gráficos nas figuras 4.15, 4.16 e 4.17 apresentam os resultados obtidos para a primeira condição. Nestas simulações, as fontes de tráfego utilizaram variáveis aleatórias com distribuição de probabilidades exponencial para gerar os intervalos entre envios de cápsulas de dados multiponto, sendo que os intervalos médios empregados e as respectivas taxas médias de geração de dados foram: $0.02s$ ($400kbits/s$), $0.013s$ ($615kbits/s$) e $0.006s$ ($1.3Mbits/s$). Cada figura apresenta os resultados para uma taxa média de geração diferente, sendo que para cada taxa foram consideradas duas taxas de perda diferentes para os enlaces fixos: 2% e 10%.



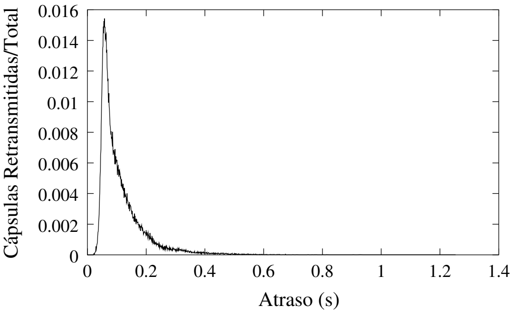
(a) Vazão.



(b) Utilização do cache.



(c) Atraso nas retransmissões, taxa de perda 2%.



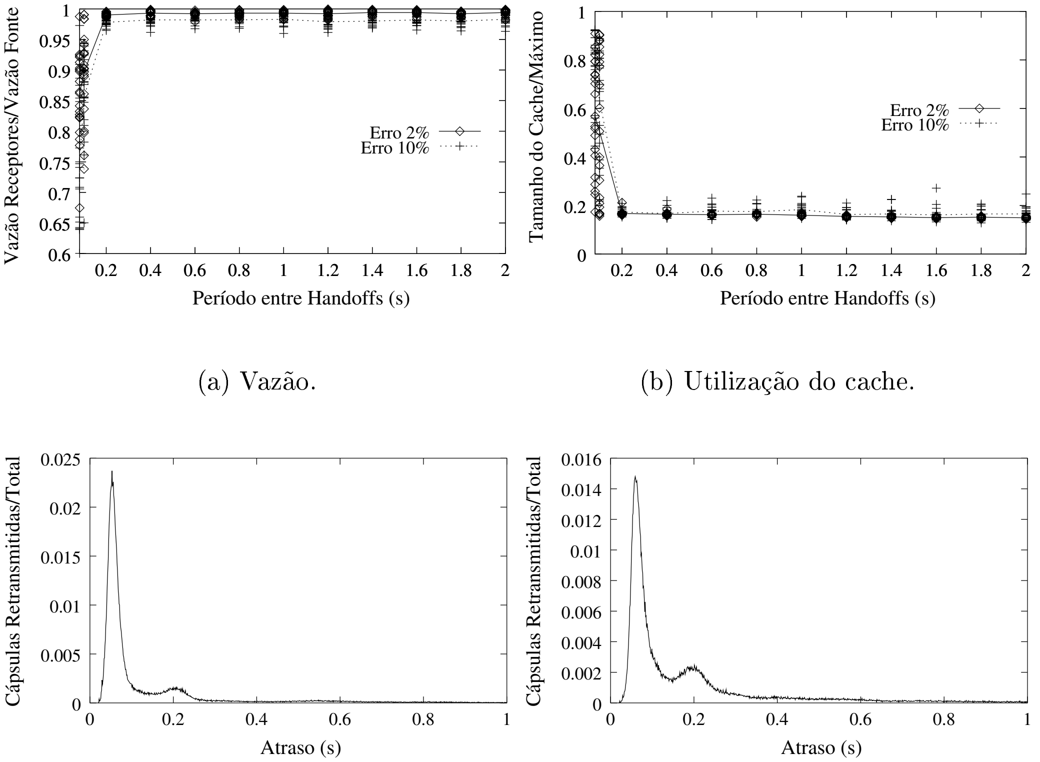
(d) Atraso nas retransmissões, taxa de perda 10%.

Figura 4.15: Taxa média da fonte 400 kbits/s. Enlaces fixos de 1.5 Mbits/s.

A figura 4.15.a mostra a vazão média obtida pelos usuários do grupo multiponto dividida pela vazão média da fonte ($400kbits/s$). Pode-se observar neste gráfico que o aumento da taxa de perda nos enlaces do interior da rede causou uma diferença entre as curvas de vazão de aproximadamente 1%, pois o aumento de 8% na taxa de erro não foi suficiente para causar um acúmulo de perdas nos receptores. Como já havia sido constatado na sub-seção 4.4.1, quando o período médio entre handoffs se aproxima do período de tempo necessário para a recuperação de perdas, a vazão diminui. Isto pode ser observado a partir dos gráficos 4.15.c e 4.15.d, que mostram as distribuições dos atrasos nas retransmissões. Nestes gráficos, a maior parte das perdas é recuperada dentro de um período de tempo inferior a $0.2s$, e assim para períodos médios entre handoffs inferiores a este valor causam um acúmulo de dados não recebidos e, portanto, acarretam em uma vazão menor. Para períodos acima de $0.2s$, a vazão se mantém aproximadamente constante.

Para avaliar o nível de utilização do cache, em cada nó foi medida a média do tamanho do cache e o tamanho máximo calculado durante todo o tempo de simulação. Assim, para cada nó foi calculado o percentual da média com relação ao máximo, sendo que os pontos apresentados no gráfico 4.15.c de utilização do cache correspondem a média deste percentual considerando todos os nós intermediários. As curvas na figura mostram que para o tráfego gerado, a taxa de utilização foi baixa, entre 14-17 %, pois como não houveram aumentos consideráveis no tamanho do cache, conclui-se que a banda passante exigida para comportar as retransmissões não ultrapassou ou ultrapassou pouco a banda disponível. Pode-se observar também que a taxa de utilização decai suavemente com o aumento do período entre handoffs devido ao decréscimo da banda passante imposta pelas retransmissões.

Os resultados para a fonte com taxa média de $615kbits/s$ apresentados na figura 4.16 mostram que o serviço mantém um bom desempenho para altas freqüências, mas apresenta queda de desempenho para as freqüências mais altas analisadas. Para períodos entre handoffs superiores a $0.2s$, observa-se na figura que a vazão se mantém estável, atingindo valores próximos do valor nominal da fonte, e, além disto, o nível de utilização do espaço de armazenamento para o cache se mantém baixo. No entanto, para períodos entre handoffs inferiores a $0.2s$, ocorre uma queda significativa na vazão média do grupo devido ao grande aumento na banda passante necessária para comportar as retransmissões. Isto pode ser verificado também através do gráfico de utilização do cache de dados e dos gráficos de distribuição dos atrasos. Quando ocorre um grande aumento no uso de banda passante, há uma tendência de crescimento



(a) Vazão.

(b) Utilização do cache.

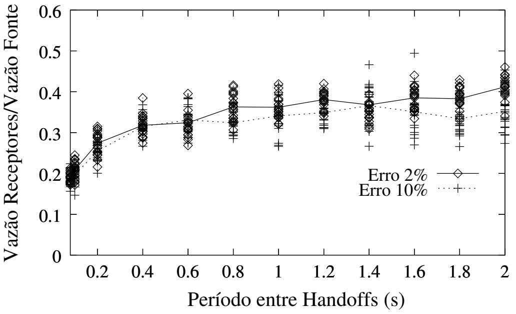
(c) Atraso nas retransmissões, taxa de perda 2%.

(d) Atraso nas retransmissões, taxa de perda 10%.

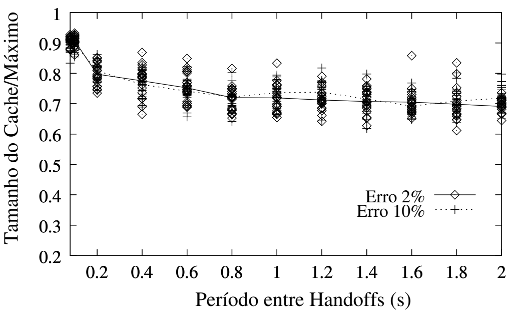
Figura 4.16: Taxa média da fonte 615 kbits/s. Enlaces fixos de 1.5 Mbits/s.

no tamanho do cache nos nós intermediários. Desta forma, pode-se observar que, para períodos entre handoffs inferiores a $0.2s$, houve um aumento significativo na quantidade de armazenamento destinada ao cache. Nos gráficos de distribuição dos atrasos de retransmissão para ambas as taxas de perda, nota-se que apesar da maior parte das retransmissões ocorrerem em períodos de tempo inferiores a $0.2s$, existem amostras que ultrapassam $0.8s$, caracterizando a ocorrência de congestionamentos.

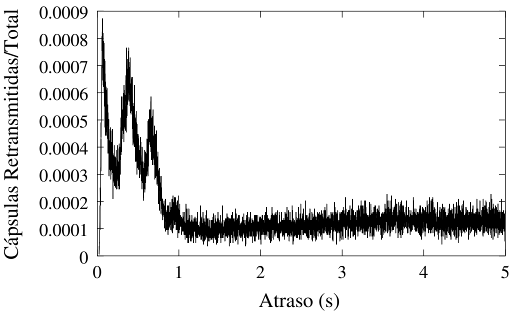
Para os resultados da fonte com taxa média de $1.3Mbits/s$, apresentados na figura 4.17, o desempenho cai para todos os períodos entre handoffs considerados, devido a pouca banda passante disponível para comportar as retransmissões. Assim, a vazão mantém-se abaixo de 40% do valor nominal da fonte para todos os períodos entre handoffs avaliados e o percentual de utilização do cache com relação ao máximo calculado mantém-se elevado, chegando a atingir 90% para os períodos mais curtos.



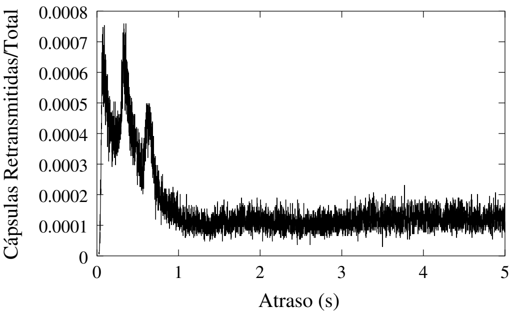
(a) Vazão.



(b) Utilização do cache.



(c) Atraso nas retransmissões, taxa de perda 2%.



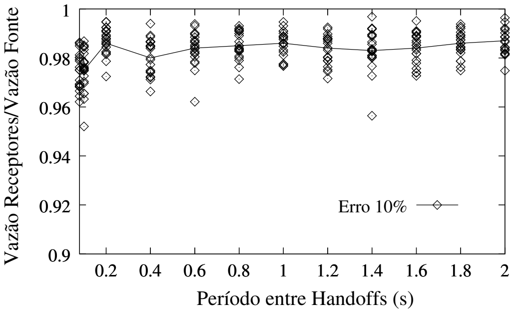
(d) Atraso nas retransmissões, taxa de perda 10%.

Figura 4.17: Taxa média da fonte 1.3 Mbits/s. Enlaces fixos de 1.5 Mbits/s.

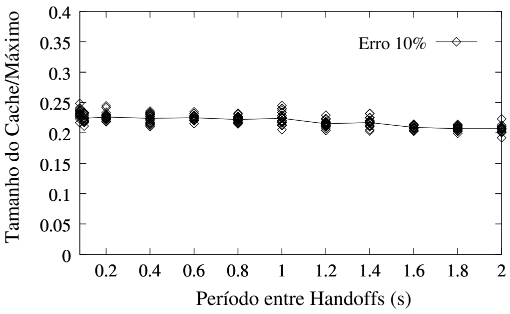
Observa-se ainda para o gráfico de vazão que conforme o período entre handoffs aumenta e a quantidade de perdas geradas diminui, a vazão cresce suavemente. Os gráficos de distribuição dos atrasos de retransmissão mostram que alguns números de seqüência não recebidos chegam a demorar $8s$ para serem recebidos pelos integrantes do grupo, caracterizando a ocorrência de congestionamentos acentuados.

Os gráficos apresentados nas figuras 4.18 e 4.19 são os resultados obtidos para a segunda condição. Nestas simulações foram avaliadas duas fontes de tráfego, também baseadas em variáveis aleatórias com distribuição de probabilidades exponencial, sendo os intervalos médios entre envios de cápsulas de $0.009s$ ($890kbits/s$) e $0.005s$ ($1.6Mbits/s$). Para estas medidas foi considerada uma taxa de perda de 10% nos enlaces fixos.

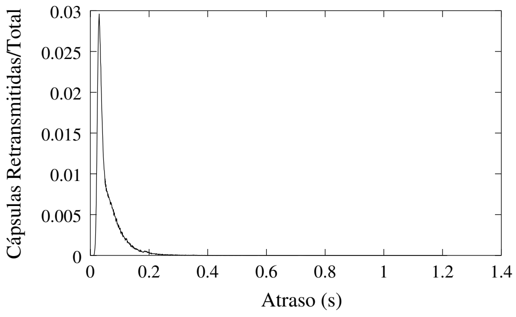
Os resultados para a fonte com taxa média de geração de dados de $890kbits/s$,



(a) Vazão.



(b) Utilização do cache.

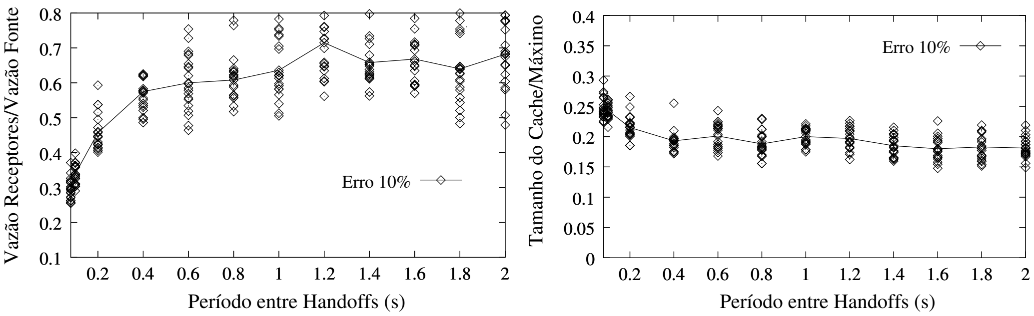


(c) Atraso nas retransmissões.

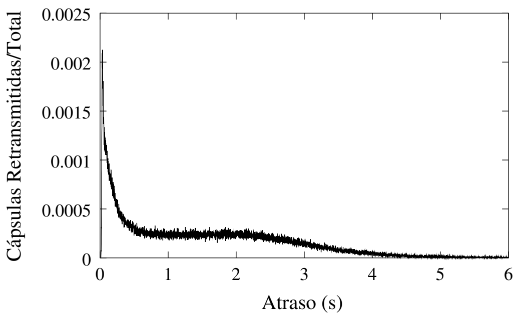
Figura 4.18: Taxa média da fonte 890 kbits/s. Enlaces fixos de 155 Mbits/s.

apresentados na figura 4.18, mostram que a banda passante disponível para o serviço foi suficiente para garantir um bom desempenho. No gráfico 4.18.a, a vazão se mantém aproximadamente constante e próxima do valor nominal da fonte para todas as freqüências, havendo uma pequena queda para períodos inferiores a $0.2s$ devido à proximidade do período de recuperação de perdas, como pode ser observado na figura 4.18.c. Para estas simulações, a maior parte das perdas foi recuperada em períodos de tempo inferiores a $0.1s$. Com relação ao cache, a média do espaço de armazenamento utilizado com relação ao máximo se apresentou baixa, sendo inferior a 25% para todos os períodos calculados e diminuindo suavemente com o aumento do período entre handoffs.

Para a fonte com taxa média de $1.6Mbits/s$, a limitação de banda passante dos enlaces sem fio causou congestionamentos, como demonstram os gráficos da figura 4.19. A vazão permaneceu sempre abaixo de 70%, chegando a atingir 30% para os



(a) Vazão. (b) Utilização do cache.



(c) Atraso nas retransmissões.

Figura 4.19: Taxa média da fonte 1.6 Mbits/s. Enlaces fixos de 155 Mbits/s.

períodos mais curtos entre handoffs. Como mostra a distribuição de atrasos de retransmissões, várias amostras tiveram atrasos de até $6s$, caracterizando a ocorrência de perdas múltiplas devida a congestionamentos. Para a quantidade de memória utilizada pelo cache, apesar da queda no desempenho causada pelos congestionamentos, a quantidade utilizada nos nós intermediários não aumenta muito devido a grande quantidade de banda passante disponível. O algoritmo 3.9, que decide pelo aumento do cache, não o faz porque considera que a banda passante imposta pelas retransmissões não ultrapassa a banda disponível. Somente os caches presentes nos NACs tem o seu tamanho aumentado regularmente, pois os congestionamentos ocorrem nestes nós na direção dos receptores. Assim, o percentual de utilização do cache se mantém baixo, em torno de 20%, tendo sido maior somente para os períodos mais curtos entre handoffs, chegando a 25% aproximadamente.

Capítulo 5

Conclusões

AS redes ativas propõem a disponibilização de uma interface para a programação de serviços, possibilitando aos usuários a alteração do comportamento funcional da rede. Com isso, as aplicações podem usufruir de serviços adequados às suas necessidades através da injeção de computações específicas de cada aplicação nos nós da rede. Assim, o paradigma de redes ativas se apresenta como uma solução bastante flexível à demanda crescente por novos serviços que exigem uma grande diversidade de requisitos de qualidade de serviço.

Esta tese propõe um serviço multiponto de transferência de dados confiável para estações móveis em redes celulares. Através de computações nos nós ativos, procura-se resolver os problemas relacionados a este tipo de comunicação, tais como: as altas taxas de erro nos enlaces sem fio devido à interferências, as perdas de dados em trânsito causadas por movimentos intercelulares, a localização das estações móveis para a entrega dos dados e a implosão de mensagens.

O serviço foi definido utilizando o conceito de cápsulas e o protocolo foi especificado através de quatro procedimentos a serem executados nos nós da rede. Para a formação de árvores multipontos, o mecanismo baseado em cápsulas de inscrição permitiu rápidas adequações a mudanças na árvore multiponto, a comunicação com o transmissor sem o intermédio de agentes e a filtragem de mensagens de inscrição nos nós intermediários, evitando o problema de implosão. A filtragem nos nós que também foi utilizada no mecanismo de recuperação de perdas de dados, tem como finalidade evitar o problema de implosão de cápsulas. Para agilizar as recuperações de dados perdidos, os nós ativos são providos de caches de dados, que evitam em muitos casos a ida dos pedidos de retransmissão até o transmissor e permitem retransmissões localizadas. Como o cache faz parte do *soft-state* de um fluxo em um nó ativo, o espaço de armazenamento destinado a esta funcionalidade deve ser negociado a cada nó. Para isto, foi proposto um mecanismo de alocação adaptativo que procura manter o espaço de armazenamento destinado ao cache adequado às necessidades da aplicação e às condições da rede.

A avaliação do serviço foi feita através do simulador de redes *ns*, que, apesar de prover o suporte para vários protocolos e serviços da Internet, não possui na sua versão original suporte para redes ativas. Desta forma, módulos de extensão ao *ns* foram implementados para viabilizar as simulações. Além disto, este simulador permite o uso de topologias *transit-stub* geradas aleatoriamente e que mantêm a estrutura básica existente na Internet, tendo sido usadas para a avaliação do serviço em estruturas similares às reais.

As simulações foram realizadas com o intuito de avaliar a eficiência dos mecanismos que compõem o serviço no que se refere à vazão e às freqüências máximas de handoff permitidas. Para isto, as simulações foram divididas basicamente em dois grupos, de acordo com o tipo de topologia: árvore e *transit-stub*.

Para as topologias em árvore, onde todos os usuários estão eqüidistantes do transmissor, duas seqüências de simulações foram realizadas considerando: cache de tamanho fixo com usuários móveis e cache adaptativo com usuários fixos. Para a primeira seqüência, pôde-se observar que a quantidade de cápsulas de inscrição que atinge o transmissor está relacionada com a dispersão dos usuários, ou seja, quanto maior for o compartilhamento do caminho entre os usuários, maior será o efeito da filtragem realizada nos nós ativos. A filtragem das cápsulas de pedido de retransmissão também exerce uma influência importante na escalabilidade do protocolo, pois como a maior parte das perdas ocorre devido à movimentação das estações e aos erros nos enlaces sem fio, mais de 70% dos pedidos de retransmissão são atendidos já no primeiro salto, havendo uma pequena variação conforme a dispersão.

A segunda seqüência de simulações para topologias em árvore teve como objetivo a verificação do mecanismo proposto de alocação adaptativa para o cache. Para isto, comparou-se com os mecanismos de alocação pelo mínimo e de alocação pelo máximo. Constatou-se que quando a banda passante requisitada está abaixo da banda passante disponível, a utilização de armazenamento para o cache de cápsulas se mantém próxima do mínimo calculado com a vazão aproximadamente igual a obtida para a alocação pelo máximo. Por outro lado, quando a banda passante requisitada ultrapassa a banda passante disponível, a utilização de armazenamento para o cache de cápsulas se mantém próxima ao máximo calculado com a vazão um pouco abaixo da obtida para a alocação pelo máximo, mas bastante acima da obtida para a alocação pelo mínimo.

As simulações para as topologias *transit-stub* visaram a observação de vários aspectos do serviço proposto quando os usuários estão a distâncias diferentes do transmissor, considerando-se muitos participantes simultâneos. A primeira série de medidas realizadas teve como objetivo a observação da influência do cache de dados na comunicação, utilizando um espaço de armazenamento fixo em cada nó para isto. Constatou-se que se a banda passante requisitada é inferior a banda passante disponível, a ausência de cache não provoca uma queda significativa no desempenho, havendo uma diferença de menos de 1% entre as medidas de vazão obtidas para este caso e para o caso em que é utilizado cache. Entretanto, se a banda passante requisitada se aproxima ou ultrapassa a banda passante disponível, uma queda significativa ocorre para o caso em que não é utilizado cache nos nós intermediários, tendo sido as medidas de vazão obtidas para este caso bastante inferiores às medidas obtidas para o caso em que os nós possuem caches. Outro aspecto avaliado foi o número de usuários comportados simultaneamente pelo serviço em uma sessão. Uma vazão aproximadamente constante e próxima da vazão nominal da fonte foi obtida para até 140 usuários movimentando-se entre as células com uma freqüência média de 2 handoffs por segundo.

As duas últimas seqüências de simulações realizadas para estas topologias utilizaram o mecanismo de alocação adaptativa para o cache. Na primeira seqüência, o movimento das estações não foi considerado, verificando-se com a variação da taxa média de geração de dados que o cache adaptativo é mais equilibrado em termos de vazão obtida e utilização de espaço de armazenamento, se for comparado com os mecanismos de alocação pelo mínimo e alocação pelo máximo. Para as simulações do serviço completo, constatou-se que para taxas médias mais baixas de geração de dados ($400kbits/s$ para a banda passante limitada em $1.5Mbits/s$ e $890kbits/s$ para a banda passante limitada em $2Mbits/s$), conseguiu-se uma vazão aproximadamente constante e próxima da vazão média da fonte para freqüências de até 12 handoffs por segundo. Conforme a taxa de geração da fonte aumenta, o desempenho cai, sendo mais acentuada a queda para as freqüências mais elevadas, pois a banda passante em uso aumenta não só pelo tráfego novo gerado mas também pelo maior número de retransmissões. Ainda assim, para as taxas de geração mais elevadas ($1.3Mbits/s$ para a banda passante limitada em $1.5Mbits/s$ e $1.6Mbits/s$ para a banda passante limitada em $2Mbits/s$), a vazão obtida em relação a vazão nominal da fonte atinge valores de até 70%, o que é um ótimo resultado, considerando-se os limites de banda passante e as altas taxas de erro.

A partir dos resultados obtidos, conclui-se que a injeção de computações específicas nos nós da rede em conjunto com os recursos de armazenamento disponibilizados nos nós ativos permitiram a definição de um serviço de transferência confiável de dados bastante eficiente. O serviço apresentado comportou altas taxas de movimento intercelular, mesmo para um grande número de participantes, economizando os recursos de memória dos nós ativos sempre que possível.

Como sugestões para trabalhos futuros, pode-se destacar:

- a realização de medidas de simulação para averiguar o comportamento com várias sessões simultâneas;
- a proposta e a implementação no simulador de uma extensão ao serviço para suportar comunicação multiponto-a-multiponto;
- a implementação de um protótipo utilizando o ANTS.

Referências Bibliográficas

[1] JUNQUEIRA, F., COSTA, L. H. K., AND DUARTE, O. C. M. B. Uma Arquitetura de Roteamento com Qualidade de Serviço baseada em Redes Ativas. In *II Seminário Franco-Brasileiro de Sistemas Informáticos Distribuídos* (Fortaleza, Ceará, novembro de 1997), pp. 257–268.

[2] CALDERON, M., SEDANO, M., AZCORRA, A., AND ALONSO, C. Active Network Support for Multicast Applications. *IEEE Network 12*, 3 (junho de 1998), 46–52.

[3] CALVERT, K., ZEGURA, E. W., BHATTACHARJEE, S., AND STERBENZ, J. Directions in Active Networks. *IEEE Communications Magazine 36*, 10 (outubro de 1998), 72–78.

[4] TENNENHOUSE, D., SMITH, J. M., SINCOSKIE, W. D., WETHERALL, D., AND MINDEN, G. A Survey of Active Network Research. *IEEE Communications Magazine* (janeiro de 1997), 80–86.

[5] TENNENHOUSE, D., AND WETHERALL, D. Towards an Active Network Architecture. *Computer Communication Review 26*, 2 (abril de 1996).

[6] BALAKRISHNAN, H., SESHAN, S., AND KATZ, R. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks 1*, 4 (dezembro de 1995).

[7] MANZONI, P., GHOSAL, D., AND SERAZZI, G. Impact of Mobility on TCP/IP: An Integrated Performance Study. *Journal on Selected Areas in Communications 13*, 5 (junho de 1995), 858–867.

[8] Valko, A., Campbell, A., and Gomez, J. Cellular IP. draft-valko-cellularip-00.txt. Internet Draft, novembro de 1998.

[9] Towsley, D., Kurose, J., and Pingali, S. A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols. *Journal on Selected Areas in Communications 15*, 3 (abril de 1997), 398–406.

[10] Lehman, L. H., Garland, S. J., and Tennenhouse, D. L. Active Reliable Multicast. In *IEEE INFOCOM'98* (São Francisco CA, EUA, abril de 1998), pp. 581–589.

[11] Levine, B. N., and Garcia-Luna-Aceves, J. J. A Comparison of Known Classes of Reliable Multicast Protocol. In *Proceedings of The International Conference on Network Protocols* (Columbus, Ohio, outubro de 1996).

[12] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *IEEE/ACM Transactions on Networking 5*, 6 (dezembro de 1997), 784–805.

[13] Bakre, A. V., and Bradrinath, B. R. Implementation and Performance Evaluation of Indirect TCP. *IEEE Transactions on Computers 46*, 3 (março de 1997), 260–278.

[14] Perkins, C. IP Mobility Support. Request For Comments 2002, outubro de 1996.

[15] Perkins, C. E. *Mobile IP - Design Principles and Practices.* Addison-Wesley, 1997.

[16] Xylomenos, G., and Polyzos, G. C. IP Multicast for Mobile Hosts. *IEEE Communications 35*, 1 (janeiro de 1997), 54–58.

[17] Deering, S. Host Extensions for IP Multicasting. Request For Comments 1112, agosto de 1989.

[18] Fenner, W. Internet Group Management Protocol, Version 2. Request For Comments 2236, novembro de 1997.

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

[20] Diot, C., Dabbous, W., and Crowcroft, J. Multipoint Communication: A Survey of Protocols, Functions and Mechanisms. *IEEE Journal on Selected Areas in Communications 15*, 3 (abril de 1997), 277–290.

[21] Smith, J., et al. Activating Networks: A Progress Report. *IEEE Computer 32*, 4 (abril de 1999), 32–41.

[22] Legedza, U., Wetherall, D. J., and Guttag, J. V. Improving the Performance of Distributed Applications Using Active Networks. In *IEEE INFOCOM'98* (São Francisco CA, EUA, abril de 1998), pp. 590–599.

[23] Wetherall, D., Legedza, U., and Guttag, J. Introducing New Internet Services: Why and How. *IEEE Network 12*, 3 (junho de 1998).

[24] Wetherall, D. J., Guttag, J., and Tennenhouse, D. ANTS: Network Services Without the Red Tape. *IEEE Computer 32*, 4 (abril de 1999), 42–48.

[25] 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, EUA, abril de 1998).

[26] Alexander, D. S., et al. A Secure Active Network Architecture: Realization in SwitchWare. *IEEE Network 12*, 3 (junho de 1998), 37–45.

[27] Alexander, D. S., et al. The SwitchWare Active Network Architecture. *IEEE Network 12*, 3 (junho de 1998), 29–36.

[28] Hicks, M., et al. PLAN: A Packet Language for Active Networks. In *International Conference on Functional Programming* (1998), pp. 86–93.

[29] Hartman, J. J., et al. Joust: A Platform for Liquid Software. *IEEE Computer 32*, 4 (abril de 1999), 50–56.

[30] 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).

[31] Ballardie, T., Francis, P., and Crowcroft, J. Core-based Trees (CBT) in a Architecture for Scalable Inter-Domain Multicast Routing. In *SIGCOMM'93* (São Francisco, CA, setembro de 1993), pp. 13–17.

[32] Sanjoy, P., Krishan, S., and Bhattacharyya, S. Reliable Multicast Transport Protocol (RMTP). *IEEE Journal on Selected Areas in Communications 15* (abril de 1997), 407–421.

[33] Junqueira, F. P., and Duarte, O. C. M. B. Comunicação Multiponto para Estações Móveis em Redes Ativas. In *XVII Simpósio Brasileiro de Redes de Computadores* (Salvador, BA, maio de 1999), pp. 389–404.

[34] Junqueira, F. P., and Duarte, O. C. M. B. Comunicação Multiponto Confiável em Redes Ativas Utilizando Cache Adaptativo. In *XXVI Seminário Integrado de Software e Hardware* (Rio de Janeiro, julho de 1999), pp. 73–87.

[35] Tanenbaum, A. S. *Computer Networks*. Prentice Hall, 1996.

[36] Bajaj, S., et al. Improving Simulation for Network Research. Tech. Rep. 99-702, University of Southern California, Department of Computer Science, março de 1999.

[37] Fall, K., and Varadhan, K. *NS Notes and Documentation*. The VINT Project, novembro de 1997.

[38] Zegura, E. W., Calvert, K., and Doar, M. B. Modeling Internet Topology. *IEEE Communications Magazine* (junho de 1997).

[39] Zegura, E. W., Calvert, K. L., 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.

[40] Dehgan, S., and Steele, R. Small Cell City. *IEEE Communications 35*, 8 (agosto de 1997), 52–59.

[41] Jabbari, B., Colombo, G., Nakajima, A., and Kulkarni, J. Network Issues for Wireless Communications. *IEEE Communications 33*, 1 (janeiro de 1995), 88–98.

[42] LaMaire, R. O., Krishna, A., Bhagwat, P., and Panian, J. Wireless LANS and Mobile Networking: Standards and Future Directions. *IEEE Communications* (agosto de 1996).

[43] Kleinrock, L. *Queueing Systems*, vol. 1. John Wiley & Sons, 1975.

[44] Costa, L. H. M. K. Uma Arquitetura Escalável de Roteamento com Requisitos de Qualidade de Serviço. Master's thesis, Programa de Engenharia Elétrica, COPPE/UFRJ, julho de 1998.

Apêndice A

O Simulador de Redes *n*s

O *NS* (*N*etwork Simulator) é um simulador de redes orientado a eventos discretos que vem sendo desenvolvido por uma cooperação, denominada projeto VINT (*Virtual InterNetwork Testbed*), 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*). O *ns* é uma ferramenta para a análise de serviços e protocolos, voltados principalmente para a Internet, oferecendo abstrações que permitem a simulação sobre variadas estruturas de rede [36, 37].

Como a versão original do simulador não inclui suporte à redes ativas, parte do trabalho consistiu no desenvolvimento de módulos para permitir a simulação do serviço com a infra-estrutura desejada. Este apêndice apresenta uma introdução aos conceitos de simulação do *ns* e também a implementação realizada com o objetivo de avaliar o serviço proposto.

A.1 A Estrutura Básica do *n*s

O *ns* foi implementado utilizando duas linguagens: C++ e OTcl (*Object Tool Command Language*). O núcleo foi implementado em C++ que é compilada e desta forma permite um melhor desempenho, comparando-se com uma linguagem interpretada. A linguagem OTcl é utilizada como interface para comandos e configurações, pois facilita pequenas alterações para as simulações. Para a integração entre as duas partes, o simulador suporta uma hierarquia de classes em C++ e uma hierarquia similar em OTcl. Do ponto de vista do usuário, existe uma relação um-para-um entre uma classe em OTcl e uma em C++. Assim, objetos são criados pelo usuário através do interpretador e são espelhados para objetos da hierarquia compilada, sendo qualquer alteração no estado de um objeto realizada em uma das hierarquias visível para o objeto equivalente na outra.

O primeiro passo para executar uma simulação é definir a topologia de rede a ser utilizada. As topologias de rede são criadas especificando-se a quantidade de nós e as ligações entre estes nós através da configuração dos enlaces. Entre os parâmetros de configuração dos enlaces estão a banda passante e o atraso. Com isso, as simulações podem ser realizadas em topologias com formatos variados, que vão desde topologias simples como em forma de árvore, de estrela ou linear, até topologias aleatórias, como as geradas pelo GT-ITM. Mais detalhes sobre a geração de topologias no GT-ITM são apresentadas no apêndice B.

A geração e o consumo de pacotes no nível de rede são feitos nos agentes, que são objetos ligados a nós determinados durante a especificação da topologia e representam entidades finais de comunicação. Alguns tipos de agente estão disponíveis no simulador implementando diferentes protocolos, como, por exemplo, vários tipos de TCP (Reno, Vegas, NewReno, Tahoe e outros), UDP e SRM. Estes agentes podem ser alimentados por diferentes modelos de tráfego, como CBR (*Constant Bit Rate*), rajada/silêncio e exponencial, entre outros. Entretanto, as simulações não estão restritas ao uso dos agentes disponibilizados na versão original do simulador, sendo possível a extensão da hierarquia de classes através da implementação de novos agentes para testes.

Os pacotes gerados por um agente são entregues inicialmente ao nó ao qual o agente está ligado, através do ponto de entrada que recebe não só os pacotes dos agentes, mas todos os pacotes destinados àquele nó, como está ilustrado na figura

A.1. O próximo passo dentro de um nó é a passagem pelo classificador, que decide se o pacote é destinado a um agente ligado ao próprio nó, ou se deve enviar para um dos enlaces de saída. Caso o pacote pertença a um agente conectado ao próprio nó, o classificador repassa este pacote para um demultiplexador, que por sua vez entrega ao agente correto, de acordo com o endereço de destino.

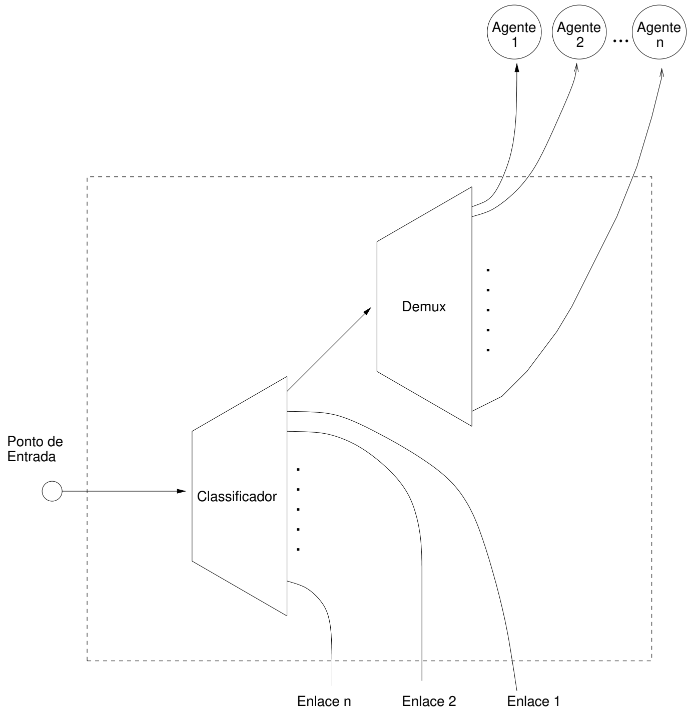


Figura A.1: Estrutura básica de um nó.

O endereçamento é feito através dos agentes, isto é, cada agente possui um endereço de 16 bits, no qual 8 bits correspondem ao identificador do nó e 8 bits identificam o agente no nó. A resolução dos endereços nos classificadores depende do tipo de estratégia de roteamento utilizada, que podem ser: estática, por sessão e dinâmica. As estratégias estática e por sessão utilizam o algoritmo SPF (*Shortest Path First*) de Dijkstra, sendo que a diferença está em quando este algoritmo é executado. Para o roteamento estático, este algoritmo é executado somente uma vez no início da simulação, enquanto que para a estratégia por sessão, além de ser executado no início, o algoritmo é executado toda a vez que houver uma alteração na topologia. O algoritmo utilizado pela estratégia dinâmica é o vetor-distância de Bellman-Ford. Cada nó envia periodicamente atualizações de rota para os nós vizinhos e, assim, atualiza dinamicamente as rotas de acordo com as alterações na topologia [19, 37, 44].

A.2 Extensões para Redes Ativas

Para as simulações realizadas, foram implementados agentes com o objetivo de gerar e consumir as cápsulas definidas para o serviço. Assim, a implementação consistiu em um agente ativo genérico com os atributos e métodos básicos e ainda duas classes de agente transmissor e duas de agente receptor, herdando atributos e métodos da classe genérica. Os agentes transmissores são responsáveis pelo envio de cápsulas multipontos, tendo sido implementados dois modelos de tráfego para este agentes: CBR (*default*) e exponencial. A diferença entre as duas classes encontra-se na capacidade de realizar retransmissões, pois somente em uma delas os pedidos de retransmissão são devidamente processados.

Os agentes receptores geram cápsulas de inscrição e de pedido de retransmissão. As cápsulas de inscrição são enviadas periodicamente e a cada handoff pelas estações interessadas no grupo multiponto, sendo o período de envio um parâmetro de configuração do agente. As cápsulas de pedido de retransmissão são enviadas somente por um dos agentes receptores, que são utilizados em conjunto com o agente transmissor capaz de manipular os pedidos de retransmissão. Ao detectar uma perda, uma cápsula de pedido de retransmissão é enviada e um temporizador é disparado com um tempo de estouro baseado no RTT para o transmissor. A cada estouro do temporizador, uma nova cápsula de pedido de retransmissão para o mesmo número de seqüência é enviada, e o temporizador é reiniciado com o período de estouro incrementado de um valor igual a metade do tempo de estouro inicial.

As cápsulas enviadas pelos agentes possuem um cabeçalho fixo com 6 parâmetros, como mostra a tabela A.1, sendo três campos estáticos e três dinâmicos. Os três campos estáticos são o identificador do tipo de cápsula, que é responsável pela resolução do procedimento a ser executado em cada nó ativo, o endereço da fonte, que identifica o agente originador da cápsula, e o identificador do fluxo, que estabelece

Tabela A.1: Campos do cabeçalho de uma cápsula

Campo	**Descrição**
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

os recursos disponíveis para a execução de uma cápsula. Os recursos implementados foram igualmente disponibilizados para todos os fluxos, havendo distinção apenas no *soft-state* de cada um. Entre os campos alterados dinamicamente estão o endereço do nó anterior, utilizado como parâmetro para alguns procedimentos de envio e também pelo mecanismo de carregamento de código do ANTS, os recursos, que informa a quantidade de recursos disponível para a execução de uma cápsula durante a sua existência na rede, e os parâmetros, que dependem do procedimento de envio e são utilizados na sua execução. Os campos de endereço do nó anterior e o de recursos são alterados em cada nó ativo, enquanto que os parâmetros podem ser alterados ou não, dependendo do procedimento de envio.

Para processar as cápsulas no interior da rede, foram implementados nós ativos, segundo a descrição dada anteriormente no capítulo 3. O classificador convencional foi substituído por um classificador ativo. O classificador ativo, ao receber uma cápsula, verifica o identificador de cápsula e executa o procedimento correspondente, passando a cápsula como parâmetro. O procedimento de cápsula retorna para o classificador uma lista de interfaces para as quais devem ser enviadas cópias da cápsula, retornando nulo se a cápsula for descartada. Em seguida, o classificador altera o campo *resources_* e *previous_* da cápsula e replica-a para quantas interfaces forem necessárias. Se a cápsula for destinada a um agente no próprio nó, então a cápsula é repassada para um demultiplexador baseado no identificador de fluxo, sendo assim necessária uma associação entre um agente e uma porta de saída do demultiplexador.

A instalação dos procedimentos foi feita de forma estática, isto é, no início da simulação os procedimentos a serem utilizados são instalados em cada nó. Para a implementação dos procedimentos de cápsulas, foi disponibilizada uma interface para consulta à tabela de rotas e também para a manipulação do *soft-state* dos fluxos. Para simular a eliminação do espaços de memória utilizados para o *soft-state* e que não são atualizados dentro de um determinado período de tempo, uma estampa de tempo é colocada no *soft-state* de cada fluxo e periodicamente um procedimento associado a um temporizador verifica os expirados e os remove, existindo chamadas na interface para a verificação da estampa de tempo e do período de expiração. Assim, a implementação de novos serviços baseados em redes ativas no *ns*, seguindo a mesma abordagem desta tese, consiste no desenvolvimento de procedimentos de cápsulas em C++ que são instalados nos nós ativos antes do início de uma simulação, sendo necessário ainda o desenvolvimento de agentes que geram as novas cápsulas segundo as funcionalidades definidas para o serviço.

Apêndice B

Parâmetros do GT-ITM

O GT-ITM (*Georgia Tech - Internetwork Topology Model*) é um gerador de topologias capaz de gerar aleatoriamente três tipos de topologias: planas, hierárquicas de nível n e *transit-stub*. Estes três tipos são variações do modelo padrão de geração de grafos aleatórios que distribui os vértices em posições aleatórias em um plano e adiciona uma aresta para cada par de vértices com uma probabilidade α. Este modelo é denominado puramente aleatório (*pure random model*). No entanto, o modelo não representa corretamente as estruturas das redes reais e, assim, outros modelos são propostos para resolver esta deficiência. Os outros modelos para a geração de topologias planas mantêm a distribuição aleatória de vértices, mas alteram a função para probabilidade de uma aresta numa tentativa de representar fielmente as topologias reais [38, 39].

As topologias hierárquicas de nível n procuram representar as estruturas em níveis, tais como as encontradas nas redes atuais. Assim, o modelo constrói recursivamente uma topologia, a partir da construção de um grafo conexo. Em seguida, cada nó na topologia corrente é substituído por um outro grafo conexo, e assim por diante até atingir o nível desejado. A decisão de quais vértice serão ligados às arestas criadas no nível anterior pode ser feita de várias formas, como por exemplo, os vértices podem ser escolhidos aleatoriamente.

As topologias do tipo *transit-stub* são grafos hierárquicos, mas que são construí-

Tabela B.1: Parâmetros para geração de topologias *transit-stub*

Parâmetro	*Significado*	*Exemplo*
T	# domínios transit	1
N_t	(média) nós/transit	8
K	(média) domínios stub/nó transit	1
N_s	(média) nós/domínio stub	4
	# total de nós	32

dos através da composição de domínios *transit* e *stub* interconectados. Primeiro, constrói-se um grafo aleatório conexo, sendo que cada vértice deste grafo corresponde a um domínio *transit* completo. Assim, cada vértice é substituído por um outro grafo aleatório conexo, que representa a topologia do *backbone* deste domínio *transit*. Em seguida, para cada nó de um domínio *transit*, são gerados novos grafos aleatórios representado os domínios *stub* ligados a este nó. Por último, são criadas ligações adicionais entre nós de um domínio *stub* e de um domínio *transit*, ou de dois domínios *stub* diferentes.

Os parâmetros apresentados na tabela B.1 controlam a distribuição de nós entre os domínios *transits* e *stubs*. Observa-se que alguns parâmetros são médias e, sendo assim, o número de nós é gerado aleatoriamente, mas mantém a média fornecida pelo usuário.

O *script* abaixo apresenta os dados contidos no arquivo utilizado durante as simulações para gerar as topologias. A primeira linha abaixo dos comentários indica o tipo de grafo a ser gerado, o número de grafos e a semente inicial para a geração de números aleatórios. Observa-se que foi solicitada a geração de um grafo do tipo *transit-stub* para cada execução do GT-ITM. Em seguida, os parâmetros indicam que só haverá um domínio *stub* para cada nó do domínio *transit*, sem arestas adicionais *transit-stub* ou *stub-stub*. A linha seguinte com os parâmetros "1 20 3 0.8" fornece os valores para a geração dos domínios *transit*. O primeiro parâmetro corresponde a T, e, portanto, indica a existência de somente um domínio *transit*. Os outro valores, que são comuns para as três últimas linhas, indicam: a escala, o modelo de probabilidades de arestas utilizado e um parâmetro dependente do modelo. A escala corresponde ao número de sub-divisões em cada nível da hierarquia, utilizado para calcular o tamanho das arestas. O modelo de probabilidades de arestas utilizado foi o puramente aleatório (3) e o parâmetro passado corresponde ao α, que é a probabilidade de haver um enlace entre dois nós de um mesmo domínio. Finalmente, as duas últimas linhas apresentam as características dos domínios *transit* e *stubs*, respectivamente.

```
## Comments :
## <#method keyword> <#number of graphs> [<#initial seed>]
## <#stubs/xit> <#t-s edges> <#s-s edges>
## <#n> <#scale> <#edgemethod> <#alpha> [<#beta>] [<#gamma>]
ts 1 47
1 0 0
1 20 3 1.0
8 20 3 0.6
4 10 3 0.6
```

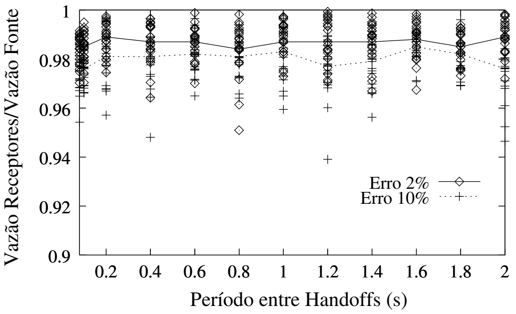
Apêndice C

Resultados Adicionais

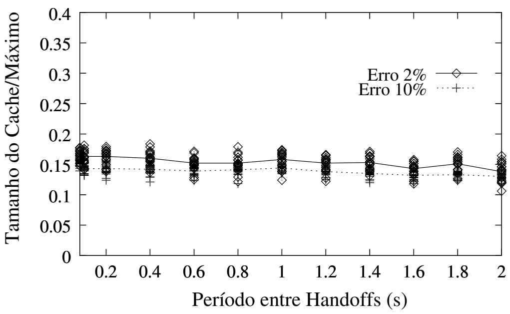
VÁRIAS fontes de tráfego foram utilizadas para avaliar o serviço proposto nesta tese. Entretanto, somente os resultados obtidos para alguma fontes foram apresentados na seção 4.4.4, pois foram suficientes para mostrar o que ocorre com o desempenho do serviço, conforme a vazão da fonte aumenta. O restante é apresentado neste apêndice.

Os gráficos apresentados nas figuras C.1, C.2, C.3 e C.4 foram obtidos utilizando uma topologia do tipo *transit-stub*, considerando enlaces fixos de $1.5Mbits/s$, nas mesmas condições que foram apresentadas na seção 4.4.4. Estas figuras apresentam resultados para fontes com intervalos médios entre envios de cápsulas de dados de: $0.08s$ ($100kbits/s$), $0.04s$ ($200kbits/s$), $0.01s$ ($800kbits/s$) e $0.009s$ ($890kbits/s$).

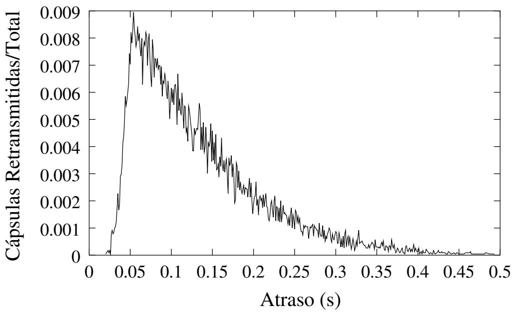
Como foi observado anteriormente, quando a banda passante requisitada está abaixo da banda disponível, considerando tanto a banda utilizada pelos dados novos gerados quanto a utilizada para as retransmissões, a vazão obtida para o grupo se mantém elevada e próxima da vazão nominal da fonte. Outra conseqüência deste fato é a baixa utilização de espaço de armazenamento para o cache, pois não há necessidade de aumentá-lo, dado que os limites de banda disponível não estão sendo ultrapassados. Isto pode ser observado pelas gráficos obtidos para as fontes de $100kbits/s$ e $200kbits/s$, em que a taxa de geração média de dados da fonte está bem abaixo do limite colocado pela rede.



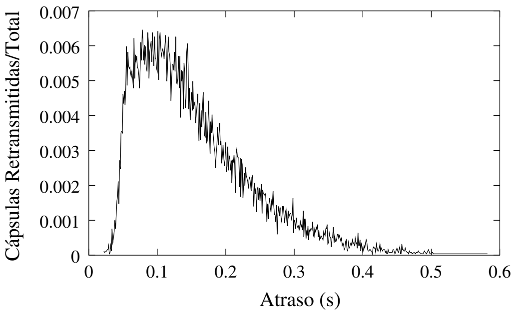
(a) Vazão.



(b) Utilização do cache.

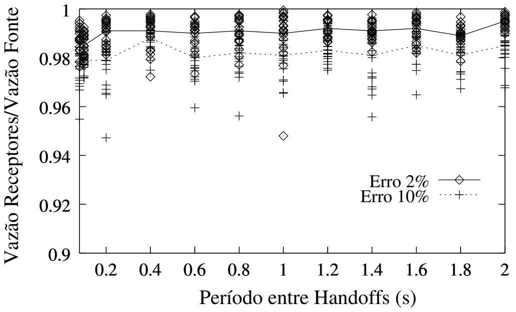


(c) Atraso nas retransmissões, taxa de perda 2%.

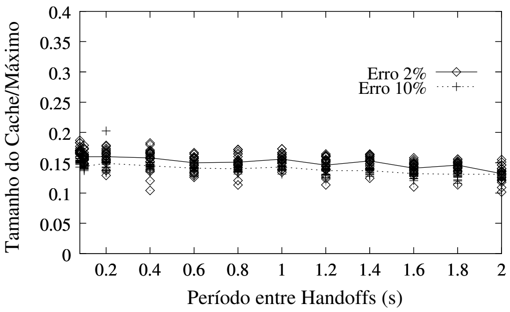


(d) Atraso nas retransmissões, taxa de perda 10%.

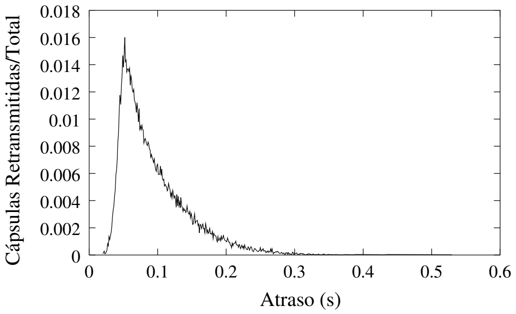
Figura C.1: Taxa média da fonte 100 kbits/s. Enlaces fixos de 1.5 Mbits/s.



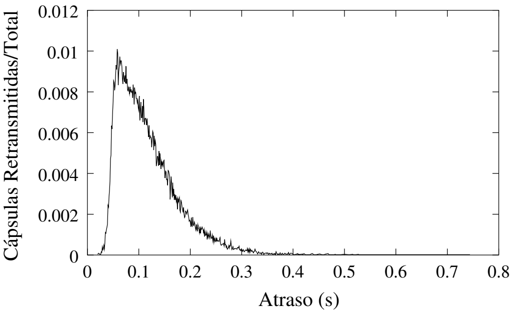
(a) Vazão.



(b) Utilização do cache.



(c) Atraso nas retransmissões, taxa de perda 2%.

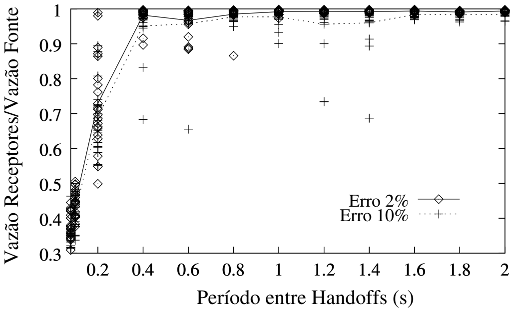


(d) Atraso nas retransmissões, taxa de perda 10%.

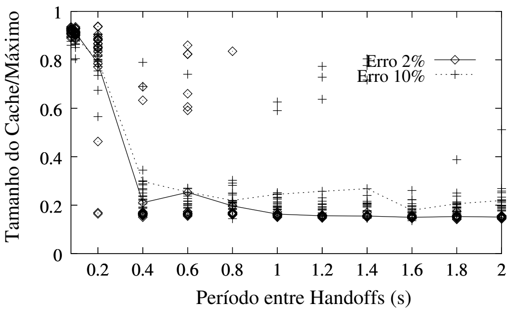
Figura C.2: Taxa média da fonte 200 kbits/s. Enlaces fixos de 1.5 Mbits/s.

Quando a taxa média de geração de dados da fonte aumenta e a taxa de geração de perdas é maior do que a capacidade de recuperação, a vazão diminui e o percentual de espaço de armazenamento utilizado para o cache aumenta. Os gráficos para as taxas médias de $800kbits/s$ e $890kbits/s$ mostram que, para os períodos entre handoffs mais curtos, a vazão média obtida para o grupo está bem abaixo da vazão nominal da fonte e o percentual de utilização do cache está elevado, pois para estes intervalo, a taxa de perdas é maior. Somente para períodos maiores que $0.4s$ para a fonte de $800kbits/s$ e $0.8s$ para a fonte de $890kbits/s$ são obtidas vazões próximas ao valor nominal da fonte.

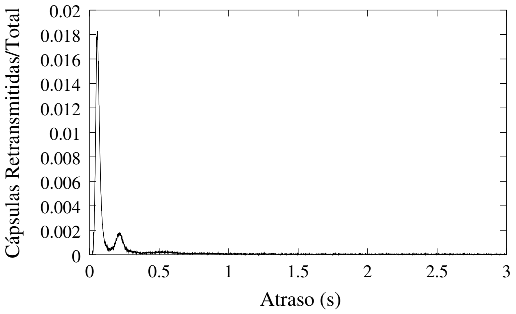
Se a banda passante disponível para a aplicação é maior, mais banda passante pode ser utilizada para as retransmissões, aumentando o desempenho da aplicação. A figura C.5 mostra os resultados obtidos para uma fonte com intervalos médios entre



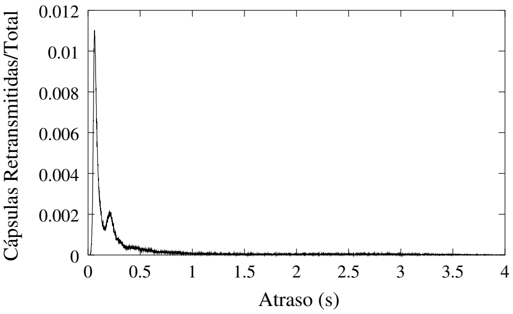
(a) Vazão.



(b) Utilização do cache.



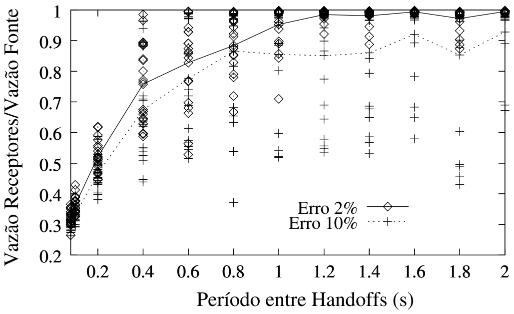
(c) Atraso nas retransmissões, taxa de perda 2%.



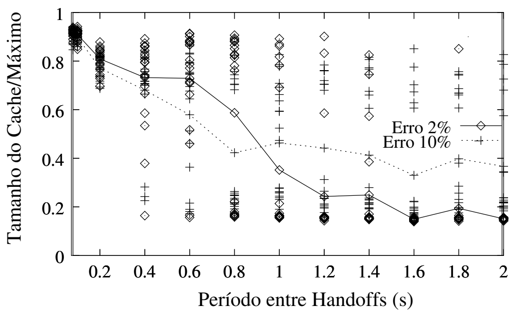
(d) Atraso nas retransmissões, taxa de perda 10%.

Figura C.3: Taxa média da fonte 800 kbits/s. Enlaces fixos de 1.5 Mbits/s.

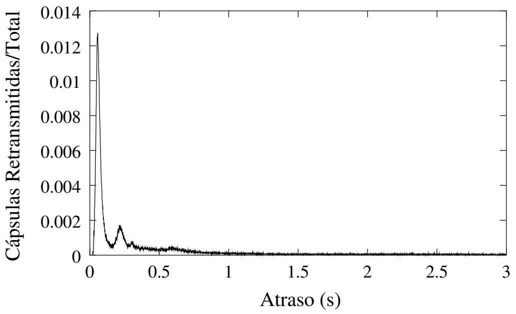
envios de cápsulas de $0.006s$ ($1.3Mbits/s$), em uma rede com enlaces no interior de $155Mbits/s$. Se for comparada a mesma fonte utilizada na rede com enlaces de $1.5Mbits/s$ (figura 4.17), pode-se notar o aumento da vazão, que agora atinge valores próximos do valor nominal da fonte para períodos entre handoffs acima de $0.2s$ e um menor percentual de utilização do cache. Além disto, os atrasos obtidos para as retransmissões, em média, são menores.



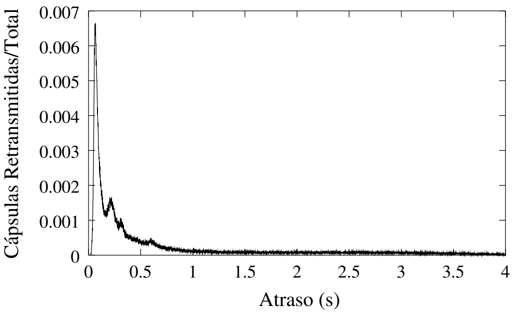
(a) Vazão.



(b) Utilização do cache.

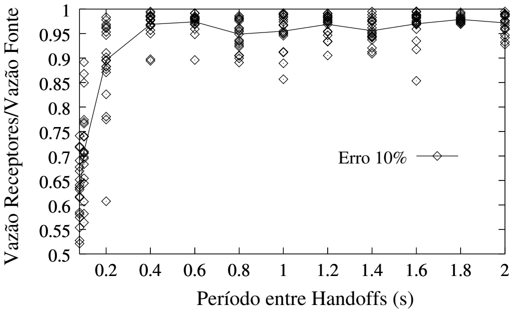


(c) Atraso nas retransmissões, taxa de perda 2%.

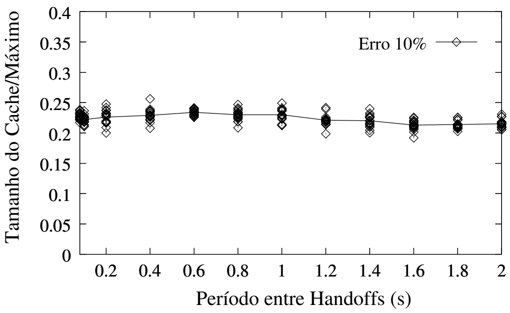


(d) Atraso nas retransmissões, taxa de perda 10%.

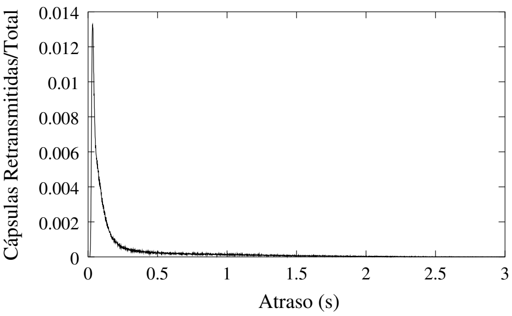
Figura C.4: Taxa média da fonte 890 kbits/s. Enlaces fixos de 1.5 Mbits/s.



(a) Vazão.



(b) Utilização do cache.



(c) Atraso nas retransmissões.

Figura C.5: Taxa média da fonte 1.3 Mbits/s. Enlaces fixos de 155 Mbits/s.