Um Arcabouço para Composição, Teste e Simulação de Protocolos de Handover Suave

Vera Nagamuta

TESE APRESENTADA AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA DA
UNIVERSIDADE DE SÃO PAULO PARA
OBTENÇÃO DO TÍTULO DE DOUTOR EM
CIÊNCIAS

Área de Concentração: Ciência da Computação
Orientadores: Prof. Dr. Siang Wun Song
Prof. Dr. Markus Endler

— São Paulo, abril de 2006. —

Na elaboração deste trabalho, a autora obteve apoio financeiro da FAPESP.

Um Arcabouço para Composição, Teste e Simulação de Protocolos de Handover Suave

Este exemplar corresponde à redação final da tese, devidamente corrigida, defendida por Vera Nagamuta e aprovada pela comissão julgadora.

São Paulo, 20 de abril de 2006.

Banca examinadora:

Prof. Dr. Siang Wun Song (Orientador) — MAC-IME-USP

Prof. Dr. Markus Endler (Co-orientador) — DI-Puc-Rio

Prof. Dr. Antônio Alfredo Ferreira Loureiro — DCC-UFMG

Prof. Dr. Djamel Fawzi Hadj Sadok — DCC-UFPE

Prof. Dr. Francisco José da Silva e Silva — DCC-UFMA

À memória de meu pai,

um grande e eterno amigo,
e à minha maravilhosa mãe...

Agradecimentos

Aos meus queridos pais, por todo o apoio, confiança e dedicação, além da presença constante em todos os momentos da minha vida. Meu pai, Paulo Nagamuta, que lutou muito e venceu na vida, deixou as mais belas lembranças e muitas saudades, além de um exemplo único de honestidade, coragem, perseverança e humildade, agradeço muito por ter sempre me incentivado com o meu doutorado (até o seu último dia de vida) e por ter acreditado na minha capacidade, e, embora não tenha sido possível a sua presença física na conclusão deste trabalho, eu tenho a certeza de que está muito bem e feliz: OBRIGADA, MUITO OBRIGADA, MEU PAI, VOCÊ SEMPRE ESTARÁ EM MEU CORAÇÃO.

Aos meus orientadores, Prof. Siang Wun Song e Prof. Markus Endler, por todos os ensinamentos, colaboração e dedicação, além da prontidão para me auxiliar em todos os momentos durante a elaboração deste trabalho. Agradeço também pela amizade, compreensão e apoio nos momentos mais difíceis da minha vida. Prof. Markus Endler, agradeço-lhe pela sua dedicação e paciência apesar da grande distância e dificuldades para a comunicação, entre os loooonnnngos E-mails e telefonemas para a discussão da tese :-).

Ao Prof. Alfredo Goldman, pela sua presença na banca de qualificação, pelas suas sugestões e críticas com relação ao trabalho da tese e pela amizade e grande incentivo nos momentos difíceis na fase ("quase lá") final da tese. Agradeço-lhe pela sua preocupação e energia positiva que sempre me passou nas vezes em que nos encontramos pelos corredores do IME-USP :-).

Ao Prof. Djamel, pela sua participação na minha banca de defesa de tese, pelas ótimas sugestões e críticas construtivas que me fizeram refletir sobre diferentes aspectos e pontos de vista da tese.

Ao Francisco, pela sua amizade e pela sua participação na banca de defesa, pelas suas sugestões e questionamentos que me ajudaram a melhorar a minha tese.

Ao Prof. Loureiro, pela sua honrada presença na minha banca de defesa, pelas suas sugestões de melhoria e acima de tudo, pelo grande incentivo e confiança em meu trabalho e na minha pessoa, muito obrigada :-).

Aos professores do IME-USP: Cristina G. Fernandes, Carlos Eduardo Ferreira e Paulo Feofiloff, da banca do exame de AA; Ana Cristina C. Melo, Fabio Kon, Marcelo Finger, Francisco Reverbel e Yoshiko Wakabayashi, pelas disciplinas ministradas; Nami Kobayashi, Kunio Okuda e a todos os professores, pela amizade e apoio.

Ao Santos Alberto, pela sua grande amizade, incentivo, apoio e compreensão. Pela confiança em minha capacidade e pela grande força e mensagens de esperança nos momentos mais difíceis da minha vida.

Aos amigos Ricardo da Rocha ("pai do MobiCS" :-) e Renata, Uirá e Roberta, pela amizade e pela grande ajuda e atenção durante a minha estadia no Rio de Janeiro, o que possibilitou muita tranquilidade e concentração para trabalhar na tese. Ricardo, agradeço-lhe muito principalmente pela grande força que você me deu na fase final da minha tese, valeu mesmo!!!

Ao Nelson, pela amizade, incentivo e apoio. Pela sua preocupação e atenção com a finalização da tese.

Aos amigos Cintia, Olga e Rudimar, pela amizade, incentivo e bons momentos que passamos no IME.

Aos amigos: Alessandro, Alexandre, Arlindo, Celina, Clódis, Eduardo, Emmanuel, Fábio, Franklin, Gerard, Gordana, José Domingo, Leandro, Isabel, Maité, Maria do Carmo, Mateus, Ney, Said, Sandra, Sirley, Ulisses e a todos os amigos do IME-USP e aos amigos da Puc-Rio: Gustavo, Hana, Renato, Viterbo e Wagner.

Ao Pinho e a todo o pessoal da CPG, pela amizade, incentivo e pela enorme atenção e disposição que sempre tiveram para atender e auxiliar a mim e a todos os pós-graduandos do IME.

À D. Dalvina e D. Jovina pela amizade e gentileza, e pelos inúmeros cafés...

Ao Sr. Humberto (em memória) e à D. Maria, por toda a atenção e gentileza que sempre tiveram comigo e por terem me proporcionado esse lugarzinho tranquilo e agradável onde passei os longos anos do doutorado...

Aos meus vizinhos e amigos da Vila Indiana: José e Jonas; à Denise, pelas suas mensagens de luz, e ao Leno, pelo apoio, incentivo e pelas muitas e longas conversas e conselhos.

Ao Gilberto e ao pessoal do CCE, pela atenção e gentileza durante o período de estágio no CCE.

Aos meus sobrinhos, Fernando e Giselle, e aos seus pais, Sônia e Leandro, pelo apoio e incentivo.

A todos os meus familiares, parentes e amigos, pelo apoio e incentivo.

Ao Valentin, por estar sempre presente e por ser um grande amigo...

Resumo

Handover é uma das questões centrais a ser considerada no projeto de protocolos em rede móveis e sem fio. Idealmente, um protocolo de handover deve garantir que a migração de um computador móvel seja completamente suave (transparente), o que significa que qualquer efeito da mobilidade deve ser escondido das camadas superiores, das aplicações e usuários. Portanto, protocolos de handover devem ser rápidos, causar baixa carga de sinais e aplicar diversas técnicas para evitar (ou minimizar) atrasos e perdas dos dados transmitidos.

Porém, suavidade é difícil de se alcançar e depende não apenas do protocolo de handover, como também das características da rede sem fio, isto é, o tamanho das células, a existência ou não de áreas de intersecção, o tipo de rede fixa que interconecta as estações base, a freqüência de migração dos usuários/computadores móveis e o tipo específico de dados sendo transmitidos assim como o suporte a QoS, que é específico da aplicação.

Nesta tese estamos propondo um arcabouço para a composição de protocolos de handover suave para micro-mobilidade para redes móveis e sem fio. Este arcabouço (que chamamos de HOPF - HandOver Protocol Framework) permite a seleção, parametrização e combinação de técnicas básicas (chamadas de módulos canônicos) baseados nos requisitos de QoS das aplicações, no perfil de mobilidade do usuário e nas características da rede móvel. Para a validação desse conceito, nós usamos o nosso arcabouço para gerar alguns protocolos encontrados na literatura, simulamo-os e analisamos o seus comportamentos e desempenho em diferentes cenários e com distintos parâmetros de QoS.

A partir dos resultados de simulações para diferentes cenários, identificamos algumas influências que alguns módulos canônicos (ou combinações de módulos) têm sobre a qualidade do fluxo de dados transmitidos da rede para computadores móveis (com relação ao número de pacotes perdidos, atraso, variação do atraso, etc.) e, a partir disso, foi possível enunciar algumas heurísticas que podem ser utilizadas para direcionar a escolha e composição de módulos canônicos para a geração de protocolos de handover suave para diferentes requisitos de QoS das aplicações.

Acreditamos que o HOPF possa ser utilizado como uma ferramenta para a comparação qualitativa de protocolos de handover e para o estudo e a experimentação de protocolos de handover adaptados para micro-mobilidade.

Abstract

Handover is a central issue when designing network protocols for cellular and mobile network. Ideally, a handover protocol should ensure that host migration is completely seamless *(transparent), meaning that it should hide from the upper protocol layers, the applications and users any effect of mobility. Therefore, handover protocols have to be fast, generate little signaling overhead, and apply several techniques to prevent (or minimize) delay or loss of communicated data.*

However, seamlessness is difficult to achieve and depends not only on the handover protocol but also on the characteristics of the wireless network, i.e. the cell size, the amount of cell overlapping, the type of wired network interconnecting the Mobility Support Stations, the migration frequency of the mobile hosts, and the particular type of data traffic and its QoS, which is application-specific.

In this thesis we propose a framework for the composition of seamless handover protocols for micro-mobility for mobile wireless networks. This framework (which we called HOPF - HandOver Protocol Framework) supports the selection, parameterization and combination of basic techniques (called canonical modules), based on the QoS requirements of the application, on the mobility profile of the user, and on the characteristics of the mobile network. As a proof of concept, we used our framework to generate some protocols found in the literature, simulated them, and analyzed their behavior and performance in different scenarios and different QoS parameters.

From the simulation results, we had identified some influence caused by canonical modules (or combination of modules) over the quality of the data flow transmited from the network to mobile computers (related with the number of packet losses, delay, delay variation, etc.) and from that it was possible to generate some heuristics which can be used to guide the selection and composition of canonical modules for generating seamless handover protocols for different QoS requirements of the applications.

We believe that the HOPF can be used as a tool for the qualitative comparision of handover protocols and for the study and experimentation of customized handover protocols for micro-mobility.

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Computação Móvel é um novo paradigma de computação distribuída que permite a um dado usuário acessar informações da rede fixa a partir de qualquer lugar e instante através de um disposistivo móvel de computação (PDAs-*Personal Digital Assistants*, *palmtops*, *laptops*, etc.).

Com esse paradigma, surgem novos desafios e questões a serem consideradas no projeto de aplicações distribuídas, pois um ambiente computacional que envolve computadores móveis possui uma série de particularidades e restrições que o distingue de um ambiente distribuído convencional. Primeiro, a comunicação de um computador móvel com a rede fixa é feita através de um canal de comunicação sem fio, que possui baixa largura de banda, alta latência e está sujeito a freqüentes desconexões, quando comparado a um canal de comunicação baseado em fibra ótica. Em segundo lugar, a mobilidade permite ao dispositivo móvel se conectar à rede através de diferentes pontos de acesso, fazendo com que este seja forçado a se adaptar a diferentes condições do ambiente de rede e às variações na disponibilidade de recursos. Além disso, o dispositivo móvel tipicamente dispõe de menor quantidade de recursos e de uma quantidade de energia limitada pela sua bateria, quando comparado a um computador pessoal.

Devido a estes fatores, o desenvolvimento de software para computação móvel enfrenta muitos obstáculos que são inexistentes na computação distribuída convencional.

1.1 Gerenciamento de Mobilidade e Handover

Gerenciamento de mobilidade trata do problema de como oferecer suporte à mobilidade de usuários em uma rede sem fio. Um dos seus maiores desafios é prover migração transparente, isto é, permitir a um usuário transitar pelas áreas de cobertura dos diversos pontos de acesso (ou células de cada Estação Rádio Base, em uma rede celular), mantendo as suas conexões ativas de modo que não ocorram interrupções na execução de serviços de comunicação utilizados pelo usuário. O gerenciamento de mobilidade trata de duas questões-chave relacionadas à mo-

bilidade: Gerenciamento de Localização e Gerenciamento de *Handover*. A primeira tem como objetivo manter atualizada a informação de localização de um computador móvel, cada vez que este se movimenta e muda o seu ponto de acesso na rede, enquanto que a segunda trata da transferência da comunicação de uma estação base para outra, a fim de possibilitar a continuidade do fornecimento de serviços na nova estação base.

Nesta tese estamos particularmente interessados no procedimento de *handover*. O *handover* é uma das questões centrais a ser considerada no projeto de protocolos para redes móveis sem fio pois, dependendo das estratégias empregadas para tratá-lo, este pode afetar consideravelmente o desempenho dos serviços correspondentes. O principal desafio é garantir que a transição de uma célula para outra seja transparente, isto é, seja imperceptível para os protocolos das camadas superiores e às aplicações (neste caso, dizemos que o *handover* é "suave", i.e., *Seamless Handover*). Portanto, a parte de um protocolo para redes móveis responsável pelo *handover*, que chamaremos de *protocolo de handover*, deve ser eficiente no sentido garantir uma baixa latência da atualização da rota para encaminhamento de pacotes, gerar uma baixa sobrecarga na rede, bem como minimizar atrasos e perdas de pacotes para o computador móvel.

Garantir uma mobilidade transparente é, no entanto, uma tarefa complexa e não depende apenas do protocolo de *handover*, mas também das características da rede sem fio, como por exemplo, o tamanho das células, a existência ou não de áreas de intersecção, o tipo e a topologia da rede fixa que interconecta as estações base, a freqüência de migração dos usuários/computadores móveis, a natureza do fluxo de comunicação, assim como o suporte para Qualidade de Serviço (*Quality of Service* - QoS) existente na rede ou implementada na aplicação. Por causa disso, não existe um único protocolo de *handover* que melhor atenda a todos os requisitos de *handover* suave de uma aplicação para todas as possíveis situações de mobilidade de usuários e tipos de redes móveis.

Várias soluções existentes, para alcançar um *handover* suave, acabam sendo específicas para uma determinada tecnologia de rede sem fio e portanto possuem um escopo de aplicação limitado. Por exemplo, foram desenvolvidas soluções para redes GSM (*Global System for Mobile Communications*), GPRS (*General Packet Radio Service*) e UMTS (*Universal Mobile Telecommunication System*) [42, 49, 31]; extensões móveis de redes ATM [65, 14], assim como para redes locais sem fio (*Wireless LANs - WLANs*) [64, 49, 40]. Apesar da grande diversidade de tecnologias de acesso sem fio, são muitos os esforços a fim de possibilitar *suavidade* durante a movimentação de usuários móveis através de distintas redes e tecnologias sem fio. Em [17] é apresentado um serviço de informação para auxiliar a execução do *handover* entre tecnologias de rede sem fio heterogêneas provendo desde informações gerais da rede e de pontos de acesso nas proximidades, informações específicas da camada de enlace que são úteis para identificar as

características da rede sem fio e informações sobre os protocolos nas camadas superiores. Em particular, esse serviço trata da complexidade do controle e monitoramento do *handover* sobre distintas tecnologias de rede evitando-se que estes sejam tratados por protocolos na camada de rede e superiores.

Prover mobilidade transparente também é uma questão abordada na camada de transporte onde foram propostas algumas extensões para o TCP e melhorias para dar suporte à mobilidade de usuários [2, 3, 45].

Na camada de rede, e em particular para o protocolo IP, a solução mais conhecida para dar suporte à mobilidade na Internet é o Mobile IP [52, 25, 28, 63]. O Mobile IP é uma extensão elegante do protocolo IP que herda todas as suas características de flexibilidade, escalabilidade e robustez. Em sua versão básica, no entanto, o protocolo não oferece suporte para *handover* suave. Os principais problemas com o Mobile IP são a sua forma de manter e atualizar a informação sobre a localização corrente de computadores móveis, a falta de um mecanismo para atualizar a rota de encaminhamento de pacotes, e o problema do roteamento triangular. Tudo isto faz com que possa haver uma perda acentuada de pacotes IP quando há migrações entre células no Mobile IP.

Várias otimizações e extensões do Mobile IP foram então propostas, a fim de melhorar o seu desempenho. Em particular, muitas abordagens foram propostas para dar suporte a *handover* suave em regiões geográficas limitadas (por exemplo, em uma subrede ou domínio administrativo), que são coletivamente denominados *Protocolos IP de Micro-mobilidade*.

Protocolos de micro-mobilidade encontrados na literatura apresentam diferentes abordagens para tratar dos diversos problemas do apoio à mobilidade [25, 41, 20, 8, 66, 9, 26, 37]. Em particular, esses protocolos propõem e implementam diversas técnicas para o gerenciamento de *handover* como, por exemplo, replicação de rotas (e pacotes), armazenamento temporário (*buffering*) e redirecionamento (*forwading*), estratégias para o chaveamento eficiente entre as rotas antiga e nova para um computador em migração, entre outros.

Porém, muitos desses protocolos não levam em consideração as características específicas e os requisitos de QoS das aplicações. Com a proliferação de serviços de dados e aplicações voltados para redes móveis e com fortes requisitos sobre a transmissão de dados, como alta confiabilidade, uniformidade do fluxo, baixo atraso e/ou variação do atraso (*jitter*) - por exemplo para VoIP, transmissão de vídeo, aplicações de *e-commerce*, aplicações de tempo real, etc. - surgiu uma grande demanda por protocolos de *handover* configuráveis e adaptáveis a partir dos requisitos de QoS específicos de cada aplicação.

1.2 Objetivo da Tese

Motivados pelos problemas citados na seção anterior, o objetivo desta tese é propor e desenvolver um arcabouço[1] para a prototipação, simulação e avaliação de protocolos de *handover* suave para micro-mobilidade. Este *framework* oferece um conjunto de módulos, cada um implementando uma alternativa de um mecanismo básico que trata de um aspecto específico do *handover* suave. Cada um destes módulos define uma técnica que está presente em um ou mais protocolos de micro-mobilidade encontrados na literatura, e que podem ser combinados para produzir um protocolo de *handover* específico.

A principal característica deste *framework*, que chamamos de HOPF (*HandOver Protocol Framework*), é que o mesmo permite não apenas prototipar os principais protocolos de *handover* citados na literatura (a partir da combinação de técnicas), como também experimentar com diferentes combinações das mesmas, a fim de projetar protocolos adaptados e adequados para determinadas aplicações e redes móveis. Para a validação desse conceito, usamos o nosso *framework* para gerar alguns protocolos encontrados na literatura, simulamos-os e analisamos os seus comportamentos e desempenhos em diferentes cenários e com distintos parâmetros de QoS. Para a implementação e a simulação dos protocolos de *handover* utilizamos o Simulador de Protocolos Distribuídos MobiCS [16, 56].

1.3 Contribuições da Tese

As principais contribuições desta tese são:

- a identificação das técnicas e mecanismos fundamentais empregados em protocolos de *handover* para micro-mobilidade em redes móveis infra-estruturadas e a descrição modular dessas técnicas na forma de elementos independentes, chamados de módulos canônicos;
- o desenho de um *framework* genérico baseado em módulos canônicos para a prototipação e simulação de uma grande variedade de protocolos de micro-mobilidade existentes na literatura, assim como o projeto e experimentação com novos protocolos, a partir da combinação de diversos módulos canônicos;
- o projeto e implementação de um *framework* orientado a objetos flexível, que permite a fácil prototipação, simulação e análise de protocolos de *handover* para diferentes configurações, topologias e tamanhos de rede móvel, taxas de geração de pacotes da aplicação e taxas

[1] Por conveniência, nesta tese utilizaremos o termo *framework* no lugar de arcabouço por ser um termo já estabelecido na área.

de migração de computadores móveis, e que permite a comparação dos protocolos com relação a vários critérios de QoS;

- a implementação e simulação de protocolos de micro-mobilidade mais citados na literatura para diferentes cenários de mobilidade e configurações de rede móvel simulada, permitindo uma avaliação do comportamento dos protocolos, bem como a formulação de heurísticas para a seleção e combinação de módulos que melhor atendam aos requisito de uma aplicação em determinados cenários;

- do ponto de vista do usuário, esta é uma ferramenta voltada para o projetista de protocolos que possibilita a implementação, simulação, teste e comparação de diferentes técnicas para o desenvolvimento de protocolos de *handover* suave para micro-mobilidade para aplicações e cenários específicos.

1.4 Organização da Tese

Esta tese está organizada da seguinte forma: no Capítulo 2 descrevemos o problema do *handover*, definimos o conceito de *suavidade*, apresentamos algumas classes de aplicações com alguns valores concretos para os parâmetros de QoS, e descrevemos os modelos de rede móvel e de mobilidade adotados na tese. A seguir, no Capítulo 3, discutimos alguns *frameworks* orientados a objetos para a composição de protocolos a partir de módulos de composição. No Capítulo 4, apresentamos algumas soluções para dar suporte à macro e micro-mobilidade, enfatizando em particular o procedimento de *handover*. No Capítulo 5, apresentamos a arquitetura do HOPF, a estrutura e funcionamento de seus componentes, algumas heurísticas para a escolha de módulos obtidas a partir da experiência com as simulações e uma especificação dos módulos canônicos implementados. No Capítulo 6, discutimos alguns aspectos sobre a implementação dos componentes do *framework*. No Capítulo 7, apresentamos os resultados de simulações para alguns protocolos de *handover* para micro-mobilidade propostos na literatura em termos de requisitos de QoS e enunciamos um conjunto de regras empíricas para a seleção de módulos canônicos. As conclusões e trabalhos futuros são apresentados no Capítulo 8. No apêndice A, apresentamos os valores das médias dos resultados apresentados nos gráficos do Capítulo 7 e os seus respectivos intervalos de confiança.

Capítulo 2

Handover

Handover ou *handoff* é o procedimento empregado em redes sem fio infra-estruturadas (por exemplo, redes celulares) para tratar a transição entre células por um computador móvel durante uma migração. O *handover* consiste em transferir a responsabilidade da comunicação de dados de uma estação base para outra, isto é, iniciar uma comunicação em uma nova estação base e proceder uma atualização na rede de modo que o computador móvel mantenha as suas comunicações em andamento.

O *handover* é um procedimento custoso pois envolve diversas tarefas, conforme discutimos a seguir, e pode causar interrupções no fornecimento de serviços aos computadores móveis assim como uma degradação no desempenho das aplicações. Esse fato se agrava quanto maior for a freqüência de migração e transição entre células, pois em conseqüência, maior é o número de ocorrências de *handover*. Um dos grandes desafios é minimizar os efeitos do *handover* e possibilitar uma migração transparente aos usuários e aplicações, ou seja, prover *handover suave*.

Basicamente, o procedimento de *handover* pode ser dividido em duas fases principais:

- Fase 1: Detecção, Atribuição e Transferência. Nesta fase, a detecção de mobilidade (i.e., a identificação da necessidade de se iniciar um *handover*), a alocação e atribuição de um novo canal de comunicação, assim como a transferência do sinal de rádio da antiga para a nova estação base são executados.

- Fase 2: Atualização. Durante essa fase, elementos de rede que mantém a informação de localização do computador móvel são notificados e atualizados de modo que o tráfego de pacotes possa ser direcionado para a nova localização. Diversas técnicas de otimização podem ser empregadas nesta fase de modo a reduzir a latência e perda de pacotes durante esse procedimento de atualização.

A Fase 1 é executada no nível de enlace e depende da tecnologia sem fio adotada, enquanto qua a Fase 2 é o principal foco dos protocolos de mobilidade que atuam na camada de rede

(protocolos de mobilidade baseados em IP). Desde que essas fases são independentes e ocorrem em diferentes níveis, não há necessariamente uma sincronização quanto à seqüência de execução das tarefas nessas duas fases. Por um lado, a Fase 2 pode ocorrer em conseqüência da Fase 1, que é o caso em que o computador móvel perde subitamente a conexão com a estação base e inicia o *handover* (na camada de rede) quando já está conectado com a nova estação base (na camada de enlace). Por outro lado, a Fase 2 pode iniciar antes mesmo da Fase 1, quando o computador móvel ou a estação base (ou ambos) possuem alguma forma para prever uma candidata à nova estação base e podem, dessa forma, preparar antecipadamente a rede e agilizar o procedimento da Fase 2.

Nas próximas seções, apresentamos uma breve classificação dos tipos de *handover*, uma descrição das tarefas envolvidas nesse procedimento e uma discussão sobre o significado de *suavidade*. Além disso, apresentamos algumas classes de aplicações com alguns valores concretos para os parâmetros de QoS, e descrevemos os modelos de rede móvel e de mobilidade adotados na tese.

2.1 Tipos de Handover

Um *handover* pode ser classificado de acordo com vários fatores, conforme citamos abaixo:

1. De acordo com a distância (do ponto de vista da rede) entre estações bases, segundo Liu *et al.* [43]:

   - **micro-handover** (*in-LAN handover*): quando o *handover* ocorre entre estações base em uma rede em um mesmo domínio administrativo ou subrede;
   - **macro-handover** (*cross-LAN handover*): quando o *handover* é executado entre estações base em redes de domínios administrativos distintos.

   Cáceres e Padmanabhan [15] usam o termo **handover local** para designar **micro-handover** e dividem **macro-handover** em duas subclasses: **handover regional**, quando o *handover* ocorre entre estações base relativamente próximas mas não necessariamente na mesma sub-rede (podem pertencer a um mesmo domínio administrativo, por exemplo, um campus) e **handover global**, que é o *handover* entre estações base muito distantes uma da outra.

2. De acordo com o tipo de célula/tecnologia de rede sem fio [46]:

- **handover horizontal**: quando o *handover* ocorre entre células/pontos de acesso do mesmo tipo (em termos de cobertura, velocidade de transmissão, mobilidade). Exemplo: UMTS para UMTS, WLAN para WLAN.
- **handover vertical** quando o *handover* ocorre entre células/pontos de acesso de tipos diferentes. Exemplo: UMTS para WLAN. De acordo com o tamanho da célula, pode ser classificado em:
  - **upward handover**: quando a migração ocorre de uma célula pequena para uma célula grande;
  - **downward handover**: quando a migração ocorre de uma célula grande para uma célula pequena.

3. De acordo com o escopo, a camada em que mobilidade é tratada:

- Na camada de enlace:

  (1) **hard handover**: o computador móvel perde a conexão repentinamente com a antiga estação base e inicia o *handover* na nova estação base. Exemplo: redes baseadas em TDMA (*Time Division Multiple Access*) [42].

  (2) **soft handover**: quando o computador móvel tem a capacidade de se conectar a mais de uma estação base. Exemplo: redes baseadas em CDMA (*Code Division Multiple Access*) [42].
- Na camada de rede:

  (3) **handover reativo**: este tipo de *handover* ocorre quando o computador móvel pode se comunicar com apenas uma estação base de cada vez, ou quando há áreas de sombra/interferência na cobertura dos sinais de rádio. Não há conhecimento a priori da nova estação base. Este tipo de *handover* sem mecanismos de otimização pode causar perdas de pacotes. Por exemplo, este tipo de *handover* é empregado no Mobile IP [52, 28, 63] onde um computador móvel detecta uma migração através de anúncios de FAs (*Agent Advertisements*) quando já está na nova localização e não tem mais comunicação com o antigo FA.

  (4) **handover pró-ativo**: é conhecido a priori a estação base ou um conjunto de potenciais estações base para onde o computador móvel vai se conectar. Isto pode ser usado para iniciar mecanismos de otimização de *handover* (configuração de caminhos de roteamento de pacotes para a nova estação base, redirecionamento de pacotes da antiga para a nova estação, etc.). Este tipo de *handover* é empregado em protocolos como o M&M (Multicast-based Micro-Mobility) [37], um protocolo baseado em *multicast* e no Cellular IP (no caso de *semi-soft handover*) [8, 66, 9].

Nesta tese, tratamos basicamente dos seguintes tipos de *handover*: micro (em um mesmo domínio administrativo), horizontal, *hard*, *soft*, reativo e pró-ativo.

2.2 Tarefas do Handover

Conforme mencionamos acima, o procedimento de *handover* pode ser dividido em duas fases e, cada uma delas envolve algumas tarefas, conforme descrevemos a seguir:

Detecção do handover e início: Para iniciar um *handover*, duas questões devem ser consideradas: (1) como identificar a necessidade de um *handover* e (2) quem inicia o *handover*. Para tratar a primeira questão, em sistemas de comunicação sem fio (e.g. redes celulares) em geral, é feita uma freqüente medição das potências de sinais pelo computador móvel e pelas estações base. Essas medidas são utilizadas para determinar a qualidade do sinal em um canal de comunicação sem fio como, por exemplo, *Word Error Indicator* (WEI), *Received Signal Strentgh Indication* (RSSI), *Quality Indicator* (QI) [42, 64]. Através dessas medições é possível determinar o momento para o início do *handover* e a estação base candidata. Devido a diversos problemas de interferência no sinal como obstáculos físicos (edifícios, torres, montanhas) que reduzem a potência do sinal, ou causam fenômenos de reflexão ou difração, além da própria redução de potência do sinal devido ao distanciamento da estação base, a tomada de decisão para o *handover* requer uma medição constante por um período de tempo suficiente a fim de evitar uma tomada de decisão imprecisa e causar a execução de *handovers* desnecessários.

Para a segunda questão, quem inicia o *handover*, existem três abordagens propostas: *Mobile-Controlled Handover* (MCHO), em que o computador móvel monitora a qualidade do sinal da estação base atual e das estações base candidatas ao *handover* e decide o início do *handover* de acordo com algum critério; *Network-Controlled Handover* (NCHO), na qual a rede monitora a qualidade do sinal emitido por um computador móvel através da cooperação entre as estações base e toma a decisão para o início do *handover*, e *Mobile-Assisted Handover* (MAHO), que é uma variante do caso anterior em que o computador móvel faz o monitoramento do sinal e notifica os resultados à estação base onde é verificado a necessidade de um *handover* e para qual estação base [42].

Na camada de rede, em protocolos de mobilidade baseados em IP (por exemplo, o Mobile IP [63]), a detecção do *handover* ou, detecção de mobilidade, é feita através de mensagens *Agent Advertisements* emitidas pelas estações base. Um computador móvel ao receber essa mensagem é capaz de identificar a ocorrência de uma migração e, a partir de então, iniciar o *handover* (Seção 4.1).

Autenticação e permissão de acesso: envolve os processos de autenticação/autorização para verificar se o computador móvel tem permissões para acessar a nova estação base (funções AAA - *Authentication*, *Authorization*, *Accouting*).

Reserva de recursos e atribuição de canais: inclui estratégias para a reserva de recursos em uma ou mais estações base candidatas incluindo a reserva/alocação de canais de comunicação e, por exemplo, estruturas de *buffer* para o armazenamento temporário de pacotes de dados nas estações base. Para permitir suavidade, é preciso executar uma pré-alocação de recursos no início do *handover*.

Existem algumas abordagens para tratar a atribuição de canais aos computadores móveis, visando, principalmente uma melhor utilização do espectro de freqüências e ao mesmo tempo, a redução de falhas de *handover* e a conseqüente perda de comunicação devido à indisponibilidade de canais na nova estação base [42]. Dentre esses esquemas podemos citar: (1) Esquema não prioritário com ou sem reserva de canais (*Nonprioritized scheme*), no qual um *handover* é bloqueado caso não haja canais disponíveis, e quando há reserva de canais, um número de canais são mantidos e utilizados para tratar somente as requisições de *handover*; (2) Esquema de Fila Prioritária (*Queuing Priority Scheme*), onde as requisições de *handover* não atendidas por indisponibilidade de canais são mantidas em uma fila e atendidas de acordo com alguma política de escalonamento; (3) Esquema de divisão (*Subrating Scheme*), no qual quando não há canais disponíveis, um novo canal temporário é obtido a partir da divisão de um canal em dois canais com a metade da capacidade de transmissão para cada um deles.

Atualização da rede: trata basicamente da atualização da informação de localização do computador móvel em um ou vários nós na rede fixa (e.g. Home Agent, roteadores, etc.), para garantir que os pacotes sejam encaminhados corretamente para o novo destino do computador móvel (nova estação base). A fim de agilizar esse procedimento e prover *handover* suave, essa tarefa de atualização pode ser executada antecipadamente, quando há uma identificação prévia de uma ou mais estações base candidatas. Alguns protocolos utilizam informações da camada de enlace para possibilitar a identificação dessas estações base candidatas (Seção 4.2).

Controle/otimização do fluxo de pacotes: a fim de reduzir atrasos, variação do atraso (*jitter*) e minimizar a perda de pacotes e duplicações, diversos mecanismos têm sido empregados como, por exemplo, *buffering*, para o armazenamento de pacotes nas estações base, *forwarding points*, para o redirecionamento de pacotes para a nova estação base, replicação

do fluxo de pacotes (por exemplo, *multicast* para várias estações base simultaneamente), intercalação dos fluxos de pacotes, entre outros.

A execução dessas tarefas depende das características da rede, dos protocolos de mobilidade empregados, assim como dos requisitos das aplicações. Conforme descrevemos na seção 4, protocolos de *handover* empregam diferentes técnicas/estratégias para executar essas tarefas e o objetivo comum desses protocolos é minimizar a latência e perdas durante o procedimento de *handover*.

2.3 Handover Suave

Uma das dificuldades para se prover *handover* suave está relacionada às limitações da forma de comunicação através do meio sem fio (interface aérea): (1) a comunicação através do meio sem fio não é confiável, sujeita a interferências, erros e desconexões repentinas; (2) a largura de banda associada a cada conexão é limitada e está sujeita a freqüentes variações dependendo do número de usuários que compartilham a mesma célula, e (3) a rede pode ser heterogênea, apresentando diferentes tecnologias e/ou padrões de transmissão/arquiteturas/velocidades de transmissão/tamanho de célula/protocolos de gerenciamento de localização e de gerenciamento de *handover*.

Esses fatores, associados à topologia e às propriedades da rede cabeada bem como as técnicas empregadas para tratar as tarefas de atualização de localização/redirecionamento de pacotes, podem causar perda de dados ou atrasos na transferência de pacotes entre a rede e o computador móvel e vice-versa. Como conseqüência, isto pode gerar uma degradação no desempenho da aplicação. Além disso, a característica de heterogeneidade impõe outras dificuldades para prover *handover* suave entre redes e domínios distintos.

O conceito de *suavidade* pode ter diferentes significados para diferentes tipos de aplicações. Esses significados podem depender, por exemplo, do tipo de aplicação e de seus requisitos de QoS (*Quality of Service*). Em particular, neste trabalho, o termo *Qualidade de Serviço* é empregado para identificar o nível de desempenho exigido por um determinado tipo de aplicação durante o procedimento de *handover* (utilizaremos o termo QoS_h).

Para um determinado tipo de aplicação, um protocolo de *handover* é considerado *suave* se o mesmo satisfaz os requisitos de QoS_h da aplicação durante a sua execução. Como exemplos de requisitos de QoS_h podemos citar: percentagem aceitável de pacotes perdidos, número máximo de pacotes fora de ordem, valores máximos de atraso e variação do atraso, etc., que foram divididos de acordo com algumas propriedades, conforme apresentamos a seguir:

- **Confiabilidade**, por exemplo, número (percentagem) de dados (pacotes) perdidos;

- **Ordenação**, por exemplo, número de pacotes fora de ordem;
- **Temporização**, por exemplo, atraso, variação do atraso;
- **Duplicação**, por exemplo, se pode ou não haver duplicações;
- **Overhead**, por exemplo, o número de mensagens de controle do protocolo geradas durante a execução, número de pacotes replicados ou retransmitidos;
- **Taxa de transmissão**, por exemplo, quantidade de bits transmitidos por segundo.

Chamamos essas propriedades de *critérios de suavidade* e essas são expressas como valores máximos ou mínimos exigidos pelas aplicações.

Em [24], apresentamos o conceito de *handover* suave de uma maneira genérica e propomos uma classificação de algumas das possíveis abordagens para prover *handover* suave. Em um outro trabalho [48], apresentamos uma proposta de um serviço de notificação para clientes móveis que se baseia no uso de elementos (*proxies*) na rede fixa, especificamente, nas estações base, e cujo principal objetivo é manter a continuidade de um serviço requisitado pelo cliente móvel. Isto permite uma migração transparente para o cliente móvel e para a aplicação e, portanto, é uma forma de se prover *handover* suave.

2.4 Classes de Aplicações e Requisitos de QoS

Nesta seção, apresentamos três classes de serviços/aplicações que são divididas de acordo com os tipos de requisitos de QoS e a forma de comunicação [29]:

1. **Serviços de Conversação/Tempo Real**: Esta classe possui os mais fortes requisitos de QoS pois estes são determinados, estritamente, pelas percepções humanas. Por exemplo, o ouvido humano é altamente sensível à variação de atraso no sinal de voz, porém, é tolerante em relação a alguma distorção no sinal devido à perda de dados. Os principais requisitos para aplicações/serviços nesta classe são: preservação da variação do atraso e um estrito e baixo atraso. Alguns exemplos de aplicações/serviços que estão compreendidos nesta classe são: serviços de conversação/voz, videofone, telnet.
2. **Serviços Interativos**: Essa classe é caracterizada por uma forma interativa de comunicação de dados, onde há um padrão requisição/resposta do usuário final. Os principais requisitos considerados nesta classe são: baixo tempo de atraso *round-trip* e baixa taxa de erros. Comparada à classe anterior, esta classe é mais tolerante em relação ao atraso e, em algumas aplicações, o parâmetro de variação do atraso não se aplica.

Exemplos de aplicações nesta classe são: acesso à WWW (somente componentes HTML, não incluem componentes como imagens, áudio/vídeo clips, etc.), serviços transacionais (comércio eletrônico) e correio eletrônico (E-mail), chat.

3. **Serviços baseados em** *streams*: Essa classe se baseia em uma forma de transporte unidirecional (*one-way*) de fluxo contínuo. A variação do atraso do fluxo deve ser limitada, existe um valor limite para o recebimento de dados no equipamento do usuário (*start-up delay*), os dados que ultrapassarem esse valor limite são descartados. Exemplos de aplicações nesta classe são: *audio streaming* (música), *one-way video* (vídeo clip), ftp.

Na Tabela 2.1, apresentamos os valores esperados (requisitos de QoS) para algumas aplicações[1] [29, 13]. A tabela está dividida de acordo com as três classes referenciadas acima. Os parâmetros considerados são: **taxa de dados** ou largura de banda requerida, **atraso** ou tempo de resposta esperado por um usuário, (na primeira classe é o atraso fim-a-fim, na segunda é o atraso *one-way* (unidirecional) e na terceira classe, é o atraso *start-up*), **variação do atraso**, e taxa de **perda de dados** tolerável.

Aplicação	Taxa de dados	Atraso	Variação do atraso	Perda de dados
conversação/ voz	4-25 kbps	< 150 ms (preferencial) < 400 ms (limite)	< 1 ms	< 3%
videofone	32-384 kbps	< 150 ms (preferencial) < 400 ms (limite)	< 1 ms	< 1%
chat	< 1 kbps	< 1 s	N.A.	zero
telnet	< 1 kbps	< 2 s	N.A.	zero
msg voz	4-13 kbps	< 1 s	< 1 ms	< 3%
web-browsing	< 30.5 kbps	< 2-5 s/page	N.A.	< 3%
e-commerce	< 24 kbps	< 2-4 s/page	N.A.	zero
e-mail	< 10 kbps	< 2-5 s	N.A.	zero
audio stream (música)	5-128 kbps	< 10 s	< 2 s	< 1%
video stream (video clip)	20-384 kbps	< 10 s	< 2 s	< 2%
ftp	< 384 kbps	< 10 s	N.A.	zero

Tabela 2.1: Valores esperados de QoS para diferentes classes de aplicações.

[1] Utilizamos as seguintes abreviações: pref.: valor preferencial, lim.: valor limite, N.A.: não se aplica.

A primeira classe de serviços se baseia fortemente no critério de suavidade de temporização, ou seja, os serviços/aplicações possuem uma forte exigência com relação ao intervalo de tempo de chegada de pacotes sucessivos (variação do atraso) e o tempo máximo para a chegada de cada pacote (atraso). As outras duas classes estão mais fortemente relacionadas com o critério de confiabilidade.

2.5 Modelo de Rede

Para este trabalho, consideramos uma estrutura de rede que possui os seguintes elementos de rede:

- **Domínio**: corresponde a um conjunto de nós em um mesmo domínio admistrativo de rede;
- **Gateway** (Gw): este elemento corresponde ao ponto de conexão entre o domínio e o restante da rede;
- **Estação Base** (EB): oferece um ou mais pontos de acesso para a comunicação com computadores móveis através do meio sem fio;
- **Roteador**: elemento na rede fixa que têm o papel de encaminhar pacotes;
- **Computador Móvel**: elemento capaz de comunicar-se através do meio sem fio quando conectado a um ponto de acesso em uma estação base;
- **Fonte**: elemento na rede fixa conectado diretamente ao *gateway* que faz o envio de pacotes ao computador móvel.

Na Figura 2.1 temos uma ilustração da estrutura da rede que estamos considerando e o relacionamento entre os elementos citados acima.

Para o propósito deste trabalho, esse conjunto de elementos é suficiente uma vez que os mesmos possuem papéis específicos em protocolos de mobilidade e podem contemplar a arquitetura de rede para micro-mobilidade.

Além disso, empregamos apenas um nó fonte e a principal razão disso é que o nosso objetivo é testar/analisar a influência que cada módulo canônico do *framework* proposto pode ter no desempenho do *handover*. O uso de vários nós fontes contribuiria apenas com um aumento da carga na rede e, possivelmente, mostraria como o compartilhamento de recursos nos nós e enlaces influenciariam no desempenho, porém, esse não é foco de nosso trabalho.

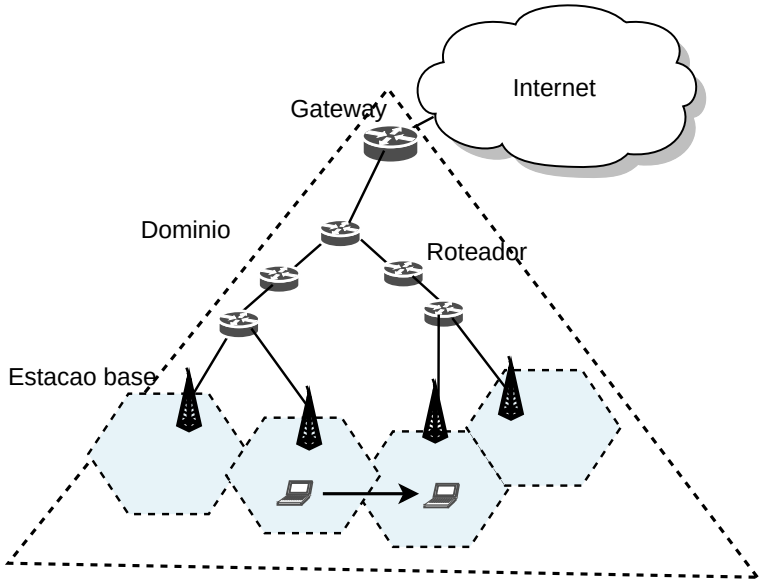


Figura 2.1: Modelo de rede

2.6 Modelo de Mobilidade

Na literatura científica encontra-se uma grande quantidade e variedade de modelos de mobilidade [55], cada um apropriado para o estudo e simulação de um determinado tipo de rede móvel, e seguindo uma abordagem para modelagem bem específica que dá ênfase em certas propriedades de uma população de usuários/dispositivos móveis.

Sendo o principal objetivo deste trabalho o estudo de protocolos de *handover* suave, acreditamos que seria suficiente utilizar um modelo de mobilidade simples que descrevesse somente os aspectos relevantes e necessários para caracterizar diferentes cenários de *handover* no contexto de micro-mobilidade. Em particular, os principais aspectos que podem influenciar o desempenho de um protocolo de *handover* são: a existência ou não de uma região de intersecção entre duas células adjacentes, a probabilidade de um computador móvel entrar em uma região dessas, e o tempo médio que um computador móvel permaneça (ou a probabilidade que ele saia) desta região. A existência ou não de regiões de intersecção nos permite simular e comparar casos quando *hard* ou *soft handover* são empregados. Além disso, a fim de retratar técnicas para a antecipação de *handover* (que chamamos de *pré-handover*, na qual é necessário um prévio conhecimento da futura estação base), utilizamos um modelo de mobilidade retilíneo no qual o computador móvel se move em uma direção fixa (da esquerda para a direita e da direita para a esquerda) atravessando as células e regiões de intersecção entre células da rede.

Portanto, acreditamos que para o nosso propósito é suficiente focar na migração entre duas

células vizinhas e definir as seguintes variáveis probabilísticas para o modelo de mobilidade: P_{mig} e $P_{migInter}$, que indicam, respectivamente, a probabilidade de um computador móvel mover-se para uma célula vizinha e a probabilidade de mover-se para dentro de uma região de intersecção entre duas células vizinhas. Essas probabilidades de migração indicam, basicamente, o intervalo de tempo (simulado) de permanência do computador móvel em uma determinada célula ou região de intersecção.

Por fim, deve-se dizer que a escolha desse modelo de mobilidade também foi fortemente influenciada pela capacidade de modelagem fornecida pelo MobiCS, que foi o ambiente de simulação utilizado no projeto.

Capítulo 3

Frameworks OO para a Composição de Protocolos

Muitos sistemas e *frameworks* orientados a objetos [39, 22, 4, 5, 34, 35, 58] têm sido propostos com a finalidade de construir protocolos distribuídos adaptados baseados na composição de componentes ou unidades básicas de composição (ou módulos canônicos, conforme definimos neste trabalho).

Dependendo do sistema e da abordagem empregada, essas unidades básicas podem implementar uma funcionalidade, ou seja, cada unidade de composição executa uma função específica de um protocolo. Ou, cada unidade de composição pode implementar uma propriedade de um serviço, por exemplo, diferentes políticas de ordenação de mensagens. Além disso, o tamanho dessas unidades de composição (granularidade), as formas de composição e a interação entre as mesmas também podem variar consideravelmente de um sistema para outro.

A modularização permite uma composição flexível de protocolos com funcionalidades ou propriedades específicas que melhor atendam a determinados requisitos da aplicação ou usuário, sistema operacional ou recursos disponíveis. Além disso, uso de padrões de projeto e *frameworks* orientados a objetos facilitam o desenvolvimento de software reduzindo a complexidade e o custo para a recriação de abstrações e soluções conhecidas oferecendo um maior grau de reutilização de componentes [30]. Um *framework* é um conjunto integrado de componentes de software que colaboram para produzir uma arquitetura reutilizável para um família de aplicações [32].

Neste capítulo, apresentamos os seguintes sistemas para composição de protocolos: x-kernel [39, 22], Coyote [4, 5], Bast [34, 35] e ACE [58] e destacamos algumas de suas características com relação às unidades básicas de composição, sua forma de interação e composição e as comparamos com o nosso *framework* proposto. Em particular, a escolha desses sistemas se deve, basicamente, às diferentes abordagens que estes empregam para tratar o problema da composição de protocolos e por serem alguns dos mais conhecidos *frameworks* orientados a objetos para esse

propósito.

3.1 *x*-kernel

O *x*-kernel [39, 22] é um dos trabalhos pioneiros no desenvolvimento de protocolos a partir da composição de módulos independentes. Foi originalmente projetado por Norm Hutchinson e Larry Peterson, na Universidade do Arizona e provê uma arquitetura modular e extensível para dar suporte à prototipação e validação de protocolos de rede. Tem sido utilizado em diversos projetos de pesquisa [7, 47, 38, 23] e também em cursos de Redes de Computadores para ilustrar os conceitos de redes [1].

A unidade básica de composição no *x*-kernel, chamada de *protocol*, encapsula as funcionalidades de um protocolo típico de comunicação (por exemplo, IP, TCP, UDP, etc). Em uma extensão do *x*-kernel [54], são propostas técnicas para a composição de unidades menores que implementam apenas uma função específica de um protocolo. A principal vantagem disto é uma maior flexibilidade de composição e capacidade de reuso.

O *x*-kernel se baseia em uma composição hierárquica de protocolos: cada nível da hierarquia corresponde a um protocolo e este pode se comunicar apenas com os protocolos que estão nos níveis adjacentes superior e inferior na hierarquia. Essa hierarquia é representada por um grafo direcionado acíclico (*grafo de protocolos*). Os nós do grafo correspondem a protocolos e as arestas representam a relação *depende de*, isto é, se um protocolo A envia mensagens usando um protocolo B, então existe uma aresta do nó A para o nó B.

A fim de prover flexibilidade na composição de protocolos, o *x*-kernel define uma interface uniforme para os protocolos de modo que a inserção/remoção de um protocolo na hierarquia seja possível sem a necessidade de efetuar alterações nos outros protocolos. Essa interface uniforme especifica um conjunto de primitivas de comunicação através das quais os protocolos interagem entre si, através da troca de mensagens.

No *x*-kernel a composição de protocolos ocorre, particularmente, em duas fases: a primeira, durante a criação e inicialização dos protocolos a partir de um grafo de protocolos, e a segunda, durante a execução, quando são definidas as conexões (ou *sessões*). A primeira fase da composição corresponde a uma composição estática, pois o grafo de protocolos não pode ser modificado em tempo de execução. Na segunda fase, podemos dizer que a composição é dinâmica pois diferentes "caminhos" no grafo de protocolos podem ser gerados na instanciação de sessões através da da técnica de *protocolos virtuais* [54]. Esta técnica permite uma maior flexibilidade na composição de protocolos em tempo de execução: um protocolo virtual liga um protocolo no nível superior a não apenas um, mas a vários protocolos no nível inferior. Em tempo de

execução, o protocolo virtual determina para qual protocolo no nível inferior uma mensagem recebida do nível superior deve ser repassada.

Uma sessão representa a comunicação entre duas entidades (protocolos), estabelece uma associação entre as mesmas e mantém o estado dessa comunicação. Por exemplo, uma sessão para o protocolo TCP pode conter informações como os números das portas origem e destino, controle da janela deslizante, janela de congestionamento, etc., e para o protocolo IP, uma sessão contém os endereços IP das duas máquinas envolvidas na comunicação. Na Figura3.1-(a) temos um grafo de protocolos que pode ser configurado em uma dada instância do *x*-kernel e em 3.1-(b) temos uma visão esquemática dos objetos do *x*-kernel: protocolos (retângulos), sessões (círculos) e uma mensagem ilustrada como uma *thread* que visita uma seqüência de protocolos e sessões.

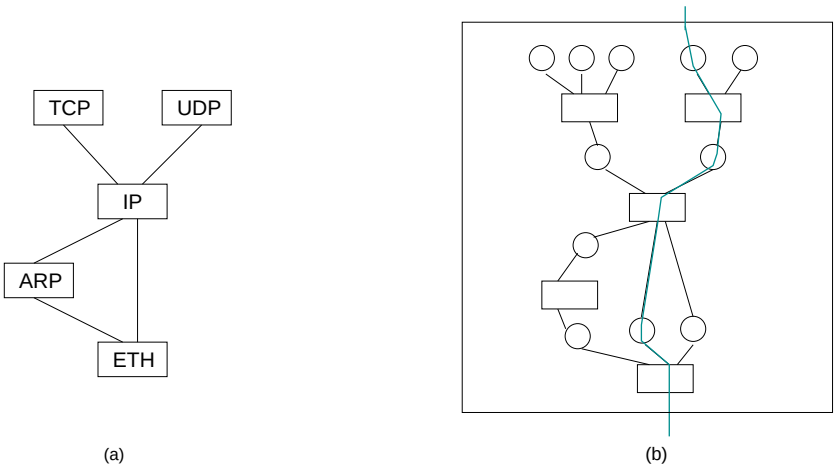


Figura 3.1: Exemplo de configuração no *x*-kernel

3.2 Coyote

O Coyote [4, 5] é um sistema que oferece suporte à construção modular de serviços de comunicação e protocolos de alto nível como, por exemplo, protocolo de *multicast* atômico ordenado, RPC em grupo, transações distribuídas, etc. É uma extensão do modelo do *x*-kernel e utiliza unidades de composição de granularidade mais fina (*micro-protocolos*).

Um micro-protocolo implementa apenas uma propriedade de um determinado serviço, por exemplo, para o *multicast* atômico, um micro-protocolo poderia implementar entrega confiável enquanto que outro poderia implementar uma determinada propriedade de ordenação (por exemplo, total, causal, FIFO).

No Coyote, a composição de protocolos ocorre em dois níveis. Um protocolo é construído a partir da combinação de micro-protocolos que é chamado de *composite protocol*. Esse *composite*

protocol é então adicionado em um nível na hierarquia de protocolos do *x*-kernel para combiná-lo com outros protocolos e formar um subsistema de rede.

A diferença de um *composite protocol* e um protocolo na hierarquia do *x*-kernel é que o primeiro, além dos micro-protocolos, possui uma estrutura interna para a comunicação e controle de execução de micro-protocolos.

A interação entre micro-protocolos é baseada em eventos. O Coyote provê um sistema de execução que faz o gerenciamento de eventos. O sistema de execução mantém uma estrutura onde cada tipo de evento está associado a uma lista de micro-protocolos (que se registraram anteriormente) e que devem ser invocados na ocorrência do evento correspondente (Figura 3.2). O sistema de execução também provê mecanismos que possibilitam a interação do *composite protocol* com protocolos nos níveis superior e inferior da hierarquia, de acordo com a especificação do *x*-kernel.

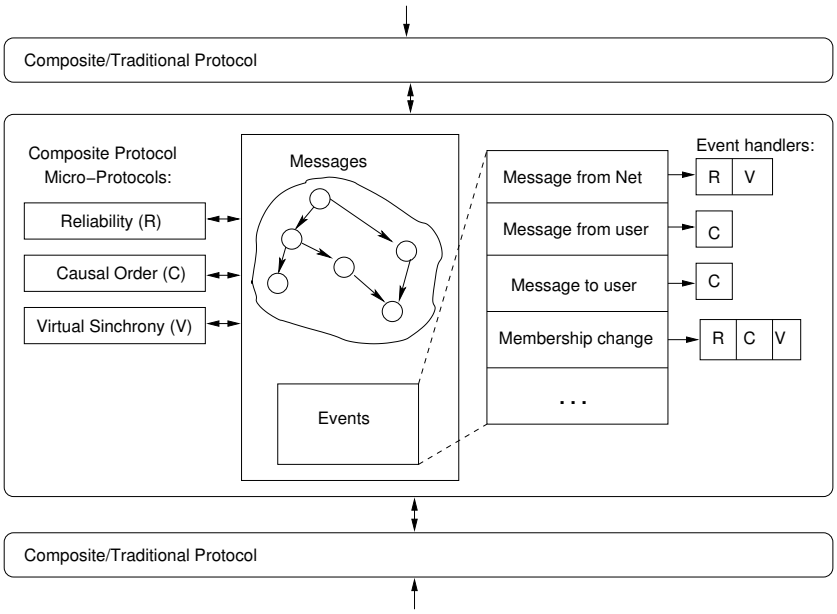


Figura 3.2: Arquitetura do Coyote

Em [6] é apresentado um exemplo de como o Coyote pode ser utilizado para o projeto de micro-protocolos para sistemas de computação móvel, enfatizando as funções de QoS e *handover*. Embora o conjunto de micro-protocolos apresentado demonstre a diversidade de composição de protocolos, o mesmo é restrito a um determinado contexto (*handover* na camada de enlace).

3.3 Bast

O Bast [34, 35] é um *framework* extensível orientado a objetos para facilitar a programação

de sistemas distribuídos confiáveis. Várias questões complexas precisam ser tratadas no desenvolvimento desses sistemas, como a detecção de falhas, a comunicação confiável, a ordenação de mensagens, o gerenciamento de réplicas, transações atômicas, etc. O Bast oferece uma abordagem hierárquica para estruturar essa complexidade e permitir a composição de protocolos distribuídos de maneira flexível.

As unidades básicas de composição no Bast correspondem a protocolos ou algoritmos distribuídos que foram propostos na literatura para tratar essas questões mencionadas acima. Esses protocolos/algoritmos distribuídos são encapsulados em *objetos protocolos* e estruturados em uma única hierarquia de classes de protocolos, de acordo com a relação de dependência existente entre os mesmos. Na Figura 3.3 temos uma ilustração da organização da arquitetura do Bast que corresponde a uma hierarquia de classes de protocolos. Cada uma dessas classes de protocolos oferece um conjunto de operações específicas relativas às funcionalidade que provêem e são definidas em suas respectivas interfaces.

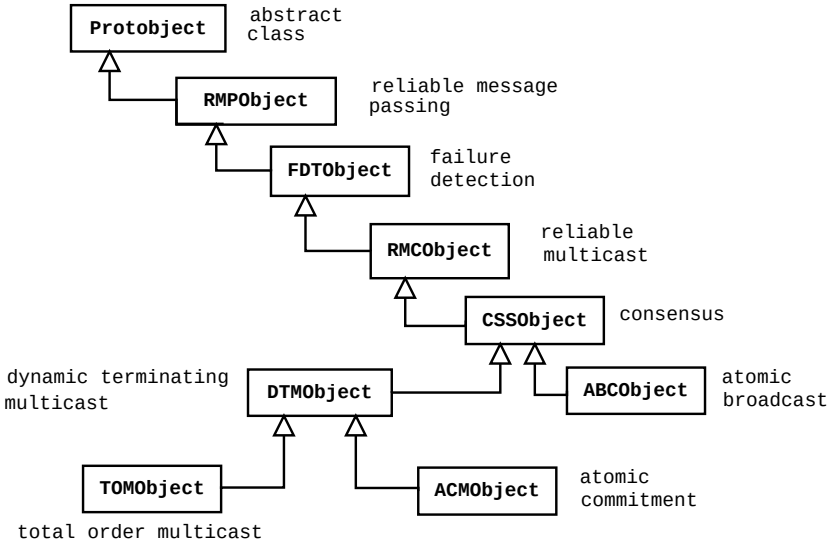


Figura 3.3: Arquitetura do Bast

Através da herança de classes, um protocolo em um nível da hierarquia é capaz de prover, além das operações definidas em sua interface, todas as operações definidas nas interfaces das classes superiores na hierarquia (superclasses). Essa herança é levada em consideração na composição de protocolos, por exemplo, um protocolo que trata o problema de consenso distribuído deverá requerer a invocação de primitivas de protocolos de comunicação confiável, *multicast* confiável e detecção de falhas.

O Bast emprega recursivamente o padrão *Strategy* [33, 32] para oferecer flexibilidade na composição de protocolos. O padrão *Strategy* permite desacoplar os algoritmos dos protocolos

que os utilizam. Algoritmos são encapsulados em objetos (estratégias) e são utilizadas por protocolos (contextos). A principal vantagem deste padrão para a composição de protocolos é que permite que uma estratégia seja escolhida (de acordo com algum critério) dinamicamente, em tempo de execução. Isso, além de aumentar a flexibilidade na composição de protocolos, também permite a geração de novos algoritmos ou otimizações para uma mesma funcionalidade sem a modificação dos algoritmos existentes e protocolos que os utilizam.

3.4 ACE

O ACE (Adaptive Communication Environment) [58] é uma ferramenta orientada a objetos que provê um conjunto de mecanismos de comunicação para simplificar o desenvolvimento de serviços e aplicações distribuídas, em particular, aplicações de tempo real para redes de alto desempenho e tem sido utilizado em vários ambientes de pesquisa e comerciais.

O ACE implementa diversos padrões de comunicação concorrente [60]. Esses padrões oferecem flexibilidade para a configuração de protocolos e favorecem o reuso e a portabilidade para múltiplas plataformas de hardware e software.

O ACE provê um *framework* de execução (ASX - ADAPTIVE Service eXecutive [59, 61]) que é responsável pela configuração de protocolos/serviços e o controle de execução. O ASX incorpora conceitos de vários outros *frameworks* de comunicação modular existentes como o System V STREAMS [], o *x*-kernel e o Conduit []. Estes sistemas permitem a configuração flexível de subsistemas de comunicação através do composição de blocos ou unidades de composição de protocolos e componentes de serviço.

Na Figura 3.4 temos as categorias de classes do ASX (os retângulos representam as categorias de classes e as linhas representam relações de dependência). Uma aplicação pode ser configurada no ASX através da especialização e composição dos seguintes componentes [61]:

- Stream: componentes nesta categoria de classes são responsáveis por coordenar a configuração e execução de um *Stream* (i.e., um objeto que contém um conjunto de serviços relacionados definidos por uma aplicação).

- Reactor: nesta categoria, os componentes são responsáveis pela distribuição de eventos aos tratadores de eventos apropriados (registrados previamente) para processar os eventos.

- Service Configurator: componentes nesta categoria fazem a configuração dinâmica de serviços ligando ou removendo dinamicamente serviços de um espaço de endereços de uma aplicação durante a sua execução.

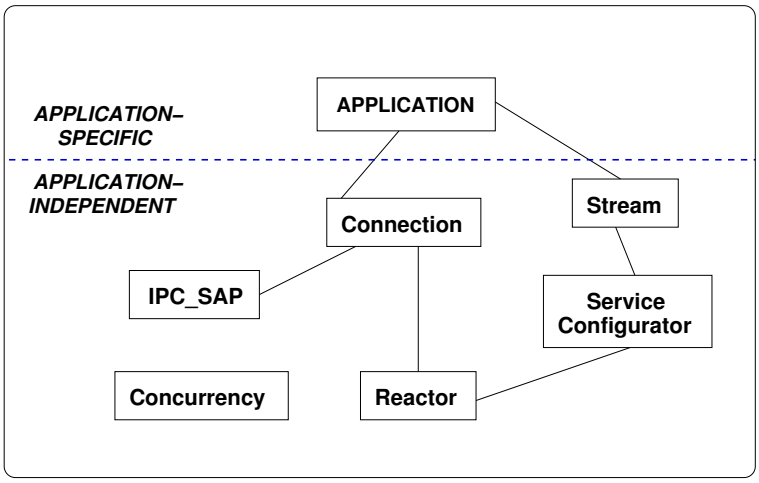


Figura 3.4: Categoria de classes no ASX

- Concurrency: os componentes nesta categoria são responsáveis pela criação, execução e sincronização de serviços durante a execução através de uma ou mais *threads* de controle dentro de um ou mais processos.
- IPC SAP: nesta categoria, os componentes encapsulam mecanismos de comunicação entre processos (IPC - *Interprocess Communication*) em uma interface orientada a objetos portável.

O ASX separa as funções/tarefas específicas da aplicação daquelas que são independentes da aplicação. Essas tarefas/funções correspondem às unidades básicas de composição no ACE. As funções específicas da aplicação (*tasks*) são configuradas em módulos e esses são organizados hierarquicamente em um objeto *Stream*. Um *Stream* é um objeto que contém um conjunto de serviços relacionados de maneira hierárquica (como camadas em uma pilha de protocolos de comunicação) definidos por uma aplicação. Módulos interagem através do envio de mensagens e podem ser configurados dinamicamente em um *Stream*.

As funções independentes da aplicação como, por exemplo, mecanismos para comunicação entre processos, demultiplexação e distribuição de eventos, (re)configuração dinâmica de serviços distribuídos, controle de concorrência, etc., são agrupadas em distintos componentes e interagem entre si e com a aplicação através da chamada de métodos especificados em interfaces bem definidas.

3.5 Comparação de Frameworks OO

Nas seções anteriores, apresentamos alguns *frameworks* OO para a composição de protocolos

e pudemos observar alguns aspectos em comum. Em primeiro lugar, o uso de unidades básicas de composição: embora cada uma das abordagens tenha um conceito particular quanto à granularidade e tipo de unidades básicas (se implementam uma função ou uma propriedade), há um concenso com relação às vantagens que esta abordagem oferece no desenvolvimento de protocolos. Em segundo lugar, o uso de hierarquias: em todas os *frameworks* propostos podemos observar a utilização de hierarquias de alguma forma para a composição de protocolos.

Dividir protocolos em módulos independentes e permitir a sua composição além de reduzir a complexidade no desenvolvimento, oferece as seguintes vantagens:

- configurabilidade: uma vez que permite a construção de protocolos adaptados e direcionados para os requerimentos de cada aplicação;
- eficiência: pois apenas as funções necessárias são configuradas, evitando-se uma utilização desnecessária de recursos e sobrecarga na execução com funções que não são utilizadas;
- reusabilidade: pois diferentes protocolos podem utilizar uma mesma unidade de composição em vez de implementá-la novamente desde o início;
- extensibilidade: novas funções podem ser facilmente acrescentadas, simplesmente adicionando-se novas unidades de composição.

Com relação à granularidade, no Coyote, a unidade de composição é de granularidade mais fina do que aquelas empregadas em outros *frameworks* e expressa propriedades em vez de funções. Unidades de composição de granularidade mais fina permitem maior configurabilidade e reusabilidade do que aquelas que possuem granularidade mais grossa [4].

O *x*-kernel, através de seu modelo hierárquico, permite gerenciar a complexidade envolvida no desenvolvimento de software de rede de modo que cada nível na hierarquia trate um determinado aspecto da comunicação. Esse modelo tem servido como base para outros *frameworks* de composição de protocolos. No Coyote, porém, um serviço/protocolo adaptado em si é implementado de maneira não hierárquica, e sim dentro de uma mesma camada, o que facilita a interação entre diferentes unidades de composição. A hierarquia é utilizada para compor um protocolo adaptado com outros usando o *x*-kernel. O ACE também estrutura uma composição de maneira hierárquica, porém, oferece facilidades para a inserção/remoção de unidades de composição dinamicamente em tempo de execução.

Através da utilização do padrão *Strategy*, o Bast permite a reconfiguração dinâmica de protocolos em tempo de execução, oferecendo maior flexilibilidade de composição. O ACE emprega vários padrões de projeto para implementar funções independentes da aplicação. Já o *x*-kernel e Coyote não implementam padrões orientados a objetos.

De uma maneira geral, algumas dessas caracteríticas contribuíram para o projeto do nosso *framework*. Em primeiro lugar, o uso de pequenos módulos independentes (unidades de composição), permite a implementação de funções e técnicas de otimização de *handover* de maneira independente, de modo que estas possam ser facilmente combinadas, testadas e analisadas de distintas formas. Em particular, as nossas unidades de composição são semelhantes aos micro-protocolos quanto à granularidade.

Em segundo lugar, assim como no Coyote, adotamos o modelo não-hierárquico para a composição de módulos. Isso porque, a princípio, nos protocolos de *handover* para micro-mobilidade que estudamos, não identificamos uma estrutura de dependência entre as tarefas que justificasse o uso de uma organização em camadas. Além disso, o modelo não-hierárquico permite uma maior flexibilidade para a composição/interação entre unidades de composição do que o modelo hierárquico, uma vez que estas não estão restritas à comunicação apenas com as unidades de composição adjacentes na hierarquia.

O *x*-kernel com suas unidades de composição de granularidade mais grossa, onde cada unidade de composição corresponde a um protocolo, poderia ser empregado para a composição de protocolos de *handover* separando-se as funcionalidades da camada de enlace das funcionalidades da camada de rede como distintos protocolos. Porém, aspectos mais específicos, como a composição de diferentes estratégias para atualização da rede ou para o tratamento do fluxo de pacotes durante o intervalo de transição do *handover*, teriam que ser implementados como combinações particulares em diferentes protocolos. Dessa forma, para testar diferentes técnicas, seria necessário a geração de vários protocolos implementando cada uma (ou uma combinação desejada) dessas técnicas.

Com relação a isso, conforme descrevemos anteriormente, o Coyote oferece uma maior flexibilidade de composição uma vez que suas unidades de composição são de granularidade mais fina e, dessa forma, diferentes técnicas podem ser compostas para formar um protocolo de *handover* adaptado com maior praticidade.

O Bast com a sua organização hierárquica de composição de protocolos facilita a composição de protocolos uma vez que estabelece uma dependência entre os mesmos e define, previamente, as relações de interação entre os protocolos. Porém, essa restrição quanto à dependência hierárquica dificulta a composição de protocolos de *handover* uma vez que nem sempre as tarefas envolvidas e as possíveis técnicas para tratá-las possuem uma clara relação desse tipo de dependência.

Capítulo 4

Protocolos baseados em IP para Redes Móveis Estruturadas

Na literatura vários trabalhos têm sido propostos para prover suporte à mobilidade suave de computadores [52, 11]. Particularmente, em redes baseadas em IP, a principal dificuldade é devido à forma estática de endereçamento e ao roteamento específico adotados pelo Protocolo IP (Internet Protocol) [18]. O protocolo IP foi projetado sem qualquer consideração com respeito à mobilidade, cada computador em uma rede IP é associado a um endereço de rede (endereço IP), que representa a sua identificação e localização ou ponto de acesso na rede. Uma mudança de ponto de acesso devido a uma migração, pode causar alguns problemas em conseqüência da mudança do endereço IP, como a perda das sessões/conexões em andamento. Isto pode ocorrer, por exemplo, em protocolos fim-a-fim, como o TCP [19], que empregam endereços IP origem e destino para identificar uma conexão.

O Mobile IP [52, 25, 28, 63], é uma extensão do protocolo IP para dar suporte à mobilidade de computadores. O Mobile IP permite a um computador se movimentar e mudar o ponto de acesso na rede de forma transparente às camadas superiores, sem que haja a necessidade de reiniciar aplicações ou bloquear conexões em andamento. Embora represente uma solução para tratar a *macro-mobilidade*, isto é, a mobilidade entre domínios administrativos, o Mobile IP possui alguns problemas, conforme descreveremos na Seção 4.1.1, que o torna inadequado para tratar o caso particular de mobilidade em pequenas regiões, quando há freqüentes *handovers*.

Devido a este fato, diversas abordagens têm sido propostas a fim de melhorar o desempenho do Mobile IP, tratando separadamente o caso de mobilidade em pequenas regiões geográficas, isto é, em um mesmo domínio administrativo de rede (*micro-mobilidade*) [11, 10]. Na Figura 4.1 ilustramos os casos de macro e micro-mobilidade.

Neste capítulo apresentamos o Mobile IP, ilustrando uma solução para o problema da macro-mobilidade, e os protocolos Mobile IP Hierárquico [36, 12], Fast Handovers [41], IDMP [20], Cel-

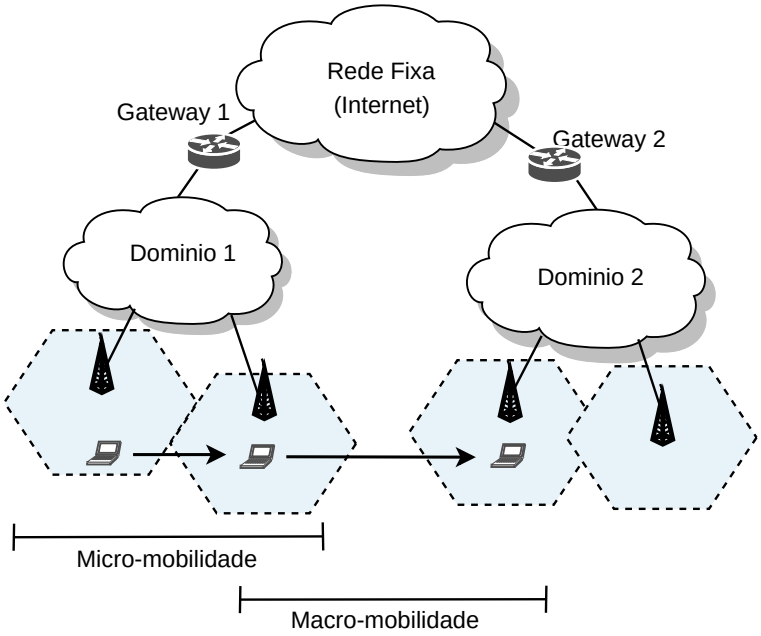


Figura 4.1: Micro e macro-mobilidade

lular IP [8, 66, 9], HAWAII [26] e M&M [37] que foram propostos para tratar micro-mobilidade. Na Seção 4.2.7 apresentamos um quadro comparativo desses protocolos de micro-mobilidade enfatizando, em particular, as técnicas de *handover* e otimizações empregadas para *handover* suave.

4.1 Macro-mobilidade: Mobile IPv4

O Mobile IP [52, 63] foi desenvolvido pela IEFT (Internet Engeneering Task Force) Mobile IP Working Group e se tornou um padrão para tratar o gerenciamento de mobilidade na Internet. O Mobile IP oferece uma solução simples e escalável com o objetivo de permitir que a mobilidade de computadores seja transparente às camadas superiores. Aqui discutimos os princípios básicos do Mobile IPv4 [63] e na Seção 4.1.2 apresentamos algumas diferenças entre o Mobile IPv4 e Mobile IPv6 [28].

O Mobile IP permite que um computador móvel mantenha inalterado o seu endereço IP durante as suas migrações e, desta forma, possibilita manter a continuidade de suas conexões/comunicações em andamento, evitando-se a necessidade de reinicializações de aplicações ou serviços a cada mudança de localização.

Para isso, cada computador móvel recebe dois endereços IP: um deles representa a sua identificação e o outro, reflete a sua localização corrente na rede. O primeiro endereço, chamado

de *home address*, é um endereço permanente, está associado à sua rede de origem (*home network*) e é o endereço conhecido pelos nós correspondentes. O segundo endereço, chamado de Care-of Address (CoA), é um endereço temporário e representa a atual localização (ponto de acesso na rede) do computador móvel quando o mesmo não está em sua rede de origem (ou seja, está em uma rede estrangeira - *foreign network*).

Dois elementos (chamados de agentes de mobilidade) gerenciam as mudanças de localização de um computador móvel: o Home Agent (HA) e o Foreign Agent (FA). O HA está localizado na rede de origem do computador móvel e possui, basicamente, duas funções: manter a atual localização (CoA) do computador móvel e interceptar pacotes destinados ao mesmo enviando-os através de tunelamento para o CoA. O FA age como um representante do computador móvel na rede estrangeira e, dentre as suas funções, podemos citar: desencapsulamento e entrega de pacotes provenientes do HA ao computador móvel (conforme descrevemos abaixo), auxiliar no procedimento de *handover* (atualização da localização no HA) e prover um CoA quando um computador móvel é proveniente de uma outra rede.

Pacotes destinados a um computador móvel são interceptados pelo HA e são encapsulados e enviados ao FA por tunelamento. O encapsulamento consiste em acrescentar um novo cabeçalho (*header*) aos pacotes originais destinados a um computador móvel, configurando como endereço destino o endereço o CoA. Esses pacotes seguem por um "túnel" que é o caminho entre o HA e o FA. Ao receber esses pacotes, o FA desencapsula os pacotes originais e os envia ao computador móvel. Na Figura 4.2 temos uma ilustração dos elementos descritos acima e o encaminhamento de pacotes de um nó na rede (nó correspondente) ao computador móvel através do HA. Na primeira parte do percurso, isto é, do nó correspondente até o HA o encaminhamento se baseia no protocolo IP tradicional e, na segunda parte do percurso, do HA ao FA, os pacotes são enviados por tunelamento.

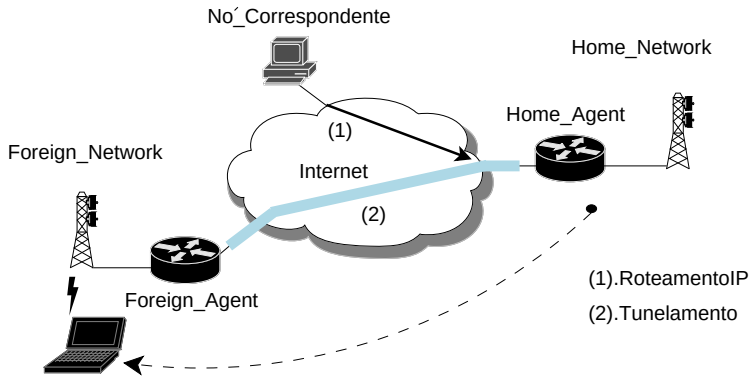


Figura 4.2: Elementos do Mobile IP e o encaminhamento de pacotes ao computador móvel

No Mobile IP, o procedimento de *handover* é constituído por duas fases: (1) *Agent Discovery*, em que um computador móvel detecta uma migração e identifica a presença de agentes de mobilidade; e (2) *Registration*, que corresponde ao processo de atualização da localização no HA. O *Agent Discovery* se baseia em mensagens *Agent Advertisement* que são periodicamente difundidas pelos HAs e FAs. Um computador móvel em uma região próxima ao HA ou a um FA "ouve" essas mensagens e detecta uma migração de uma das formas: quando o *lifetime* de uma mensagem *Agent Advertisement* recebida anteriormente se expira (*lazy detection*), ou através da comparação dos prefixos de rede dos endereços contidos no anterior e atual *Agent Advertisements* recebidos (*eager detection*). Na ausência de *Agent Advertisements* um computador móvel pode solictar o seu envio através da mensagem *Agent Solicitation*. O novo CoA pode ser obtido da própria mensagem *Agent Advertisement* de um FA ou através de um servidor DHCP [27], na ausência de FAs (neste caso o endereço é chamado de *Collocated Care-of Address*), ou ainda, por configuração manual.

A segunda fase, *Registration*, é executada quando o computador móvel verifica que houve uma mudança de ponto de acesso na rede. A atualização de seu atual CoA é feita pelo próprio computador móvel através de uma notificação para o HA (*Registration Request*) passando o novo CoA. O HA atualiza o registro do computador móvel e confirma enviando uma mensagem *Registration Reply*. Geralmente, essas mensagens de registro são intermediadas pelo FA. Este mecanismo de registro também é utilizado quando o computador móvel retorna à sua rede de origem.

4.1.1 Problemas do Mobile IP e Algumas Extensões Propostas

Conforme descrevemos acima, no Mobile IP, cada vez que um computador móvel migra e muda o seu ponto de acesso na rede, este deve atualizar a sua localização no HA (durante a fase *Registration*). Este forma de manter atualizada a localização corrente dos computadores móveis implica nos problemas:

- **triangle routing**: ou roteamento triangular, se refere ao roteamento ineficiente causado pela forma em que os pacotes são tunelados ao FA. Enquanto o computador móvel envia pacotes através do FA por um caminho ótimo para um nó correspondente, todos os pacotes destinados ao mesmo devem passar antes pelo HA para serem tunelados ao FA, possivelmente por uma rota maior. Isso pode causar atrasos na entrega de pacotes ao computador móvel e uma degradação no desempenho das aplicações.

- **perda de pacotes durante o handover**: enquanto o procedimento para a atualização de localização é executado (durante o envio da mensagem *Registration Request*, antes de

alcançar o HA), pacotes destinados ao computador móvel são direcionados para a sua antiga localização e são perdidos. Quanto maior o tempo para completar o *handover*, maior é a perda.

- **latência**: existem duas causas para a latência durante o *handover*: a primeira ocorre devido à forma como uma migração é descoberta e depende do *Agent Advertisements*. A segunda, durante a atualização do CoA no HA, e depende da distância em que a mensagem de registro deve percorrer e carga na rede;

- **sobrecarga na rede**: quando a freqüência de migrações é alta, assim também será a freqüência de *handovers*. Em conseqüência, um grande número de mensagens de registro precisam percorrer a rede.

- **suporte a QoS**: A cada *handover*, o caminho entre o HA e computador móvel é modificado e, portanto, é preciso fazer uma nova reserva de recursos. Além disso, o uso de tunelamento no Mobile IP complica a tarefa de reserva de recursos uma vez que as mensagens que contêm a descrição do fluxo para o qual os recursos devem ser reservados (por exemplo, as mensagens RSVP Path e Resv) são encapsuladas em novos pacotes.

Para aliviar os problemas causados pelo roteamento triangular, foi proposto o Mobile IP Route Optimization [51]. A idéia é informar os nós correpondentes sobre o novo CoA do computador móvel de modo que os mesmos possam tunelar pacotes diretamente ao computador móvel. Desta forma, em boa parte das situações, os pacotes não precisam mais passar pelo HA, fazendo com que sigam por um caminho mais curto.

Além do Route Optmization, uma outra proposta para o Mobile IP é o Mobile IP Smooth Handover [50]. Nesta proposta, o computador móvel, através do novo FA, envia uma notificação de seu novo CoA para o antigo FA. O novo CoA recebido pelo antigo FA é então inserido em seu cache de registros (*bindings*) como um ponteiro de redirecionamento para a nova localização. Qualquer datagrama tunelado que chegue ao antigo FA depois que este ponteiro tenha sido criado pode então ser re-tunelado para o novo CoA.

Conforme veremos na Seção 4.2, algumas extensões foram propostas para melhorar o desempenho do Mobile IP, dentre elas, o uso de hierarquias para tratar mobilidade localmente, assim como diversas estratégias empregadas em protocolos de micro-mobilidade a fim de prover *handover* suave.

4.1.2 Mobile IPv6

O IPv6 [28] é a nova versão do IP e que já oferece suporte à mobilidade de computadores.

Assim como no Mobile IPv4, o Mobile IPv6 permite a um computador móvel manter seu endereço IP inalterado durante as suas movimentações, associando ao mesmo um endereço CoA que indica a sua atual localização. Porém, no Mobile IPv6 não temos mais os FAs, todos os endereços CoA são do tipo *collocated* e podem ser obtidos, por exemplo, através de um servidor DHCP ou por um mecanismo de autoconfiguração [57]. Outra diferença com relação ao Mobile IPv4 é que o Mobile IPv6 incorpora a otimização de rota em seu protocolo, de modo que os nós correspondentes enviam pacotes diretamente ao CoA de um computador móvel.

Na ocorrência de um *handover*, mecanismos semelhantes ao *Agent Discovery* e *Registration* do Mobile IPv4 são empregados no Mobile IPv6: *Router Advertisements* são difundidos periodicamente pelos roteadores e HAs e *Router Solicitations* são enviados por computadores móveis quando estes desejam receber *Router Advertisements*. Ao detectar uma migração (de maneira semelhante ao Mobile IPv4), um computador móvel obtém um CoA e envia a mensagem *Binding Update* ao HA e a todos os nós correspondentes. Essa mensagem é confirmada através de *Binding Acknowledgment*.

4.2 Protocolos IP para tratamento de Micro-mobilidade

Muitos protocolos de micro-mobilidade têm sido propostos com o objetivo de estender ou complementar o Mobile IP para tratar migrações em um mesmo domínio admistrativo [36, 12, 41, 20, 8, 66, 9, 26, 37, 11, 10]. O principal problema do Mobile IP (v4 e v6) e que afeta o desempenho do *handover* é devido ao custo da atualização do CoA no HA e possivelmente nos nós correspondentes. Em casos em que há freqüentes migrações, este processo de atualização em um HA possivelmente distante pode causar perdas e latência na entrega de pacotes e, como conseqüência, uma degradação no desempenho da aplicação, o que pode ser problemático, principalmente, em aplicações que precisam de entrega confiável ou entrega com um limite de tempo pré-estabelecido.

De uma forma geral, protocolos para micro-mobilidade assumem uma estrutura de rede composta por domínios (uma ou mais redes em um mesmo domínio administrativo - Figura 4.1), onde cada domínio está conectado à Internet através de um roteador especial chamado de *gateway*. O *gateway* possui duas principais funcionalidades: (1) servir como ponto de referência para os computadores móveis presentes no domínio, sendo que o seu endereço (*gateway*) é registrado nos respectivos HAs como se fosse um CoA "fixo" (isto é, um CoA que é válido enquanto o computador móvel estiver no mesmo domínio); e (2) interceptar pacotes tunelados por HAs e nós correspondentes e direcioná-los aos respectivos computadores móveis destinatários dentro do domínio. Portanto, o *gateway* pode ser visto como um segundo HA, o HA para o gerenciamento

de localização intra-domínio.

A principal vantagem de protocolos de micro-mobilidade é o processamento local do *handover*, ou seja, toda migração de um computador móvel dentro de um domínio não precisa ser notificada ao seu HA, mas apenas ao *gateway*. Isto possibilita um rápido processamento de *handover*, limitando-se à propagação de mensagens de atualização dentro do domínio e reduzindo-se desta forma a perda de pacotes e a latência. Porém, no caso de migração inter-domínio, os protocolos de micro-mobilidade adotam o próprio Mobile IP original para tratar o *handover*.

Nas diversas propostas que apresentamos a seguir, será dada ênfase especial aos seguintes aspectos relacionados ao gerenciamento de *handover*:

- detecção de mobilidade: quem inicia o *handover* e quais mecanismos são empregados;
- atualização na rede: estratégias para tratar a reconfiguração de caminho, incluindo a geração de um novo caminho e a remoção do caminho antigo e quais/quantos elementos participam dessa atualização assim como técnicas para atualizar a localização;
- otimizações: algoritmos e estratégias para prover *handover* suave, a fim de reduzir a perda de pacotes e latência, transferência de contexto, suporte à QoS, etc.
- transmissão de pacotes: mecanismos para o envio de pacotes na rede, por exemplo, tunelamento, *multicasting*, etc.

Por se tratar de aspectos importantes e que podem afetar consideravelmente o desempenho de protocolos de *handover* para micro-mobilidade, estes foram considerados em nosso *framework* e serviram como uma base para a estruturação dos componentes do mesmo.

4.2.1 Mobile IP Hierárquico

A utilização de hierarquias para "regionalizar" o gerenciamento de mobilidade pode ser observada em vários trabalhos anteriores, em redes ATM e redes celulares [65, 14]. A princípio, o objetivo é reduzir o número de mensagens de atualização na rede e também o tempo de processamento dessas atualizações tratando-se a mobilidade apenas localmente.

Mobile IP Regional Registration

O Mobile IP Regional Registration [36] (ou Mobile IP Hierárquico - HMIPv4) estende o Mobile IP empregando uma hierarquia de FAs para tratar localmente o processamento de *handover*.

No topo da hierarquia está o *gateway* e está associado a um FA (chamado de GFA). Abaixo deste, podemos ter um ou mais Regional Foreign Agents (RFA). Inicialmente, ao entrar em

domínio, um computador móvel se registra em sua rede de origem (HA) usando como CoA o endereço IP associado ao GFA. O GFA mantém uma lista de computadores móveis visitantes no domínio, associando a cada um deles a sua atual localização dentro do domínio (endereço de CoA do RFA).

Pacotes destinados ao computador móvel são tunelados do HA para o GFA. Este desencapsula os pacotes e faz um re-tunelamento para o RFA correspondente de acordo com o atual CoA e, finalmente, o RFA desencapsula os pacotes e direciona ao computador móvel.

Da mesma forma como no Mobile IP, os RFAs anunciam a sua presença através de mensagens *Router Advertisement*, porém, com algumas modificações. Por exemplo, neste caso, um *Router Advertisement* contém dois endereços de CoA: um corresponde ao RFA e o outro corresponde ao GFA. Através da comparação destes endereços com seu atual CoA, um computador móvel é capaz de detectar a necessidade de executar um *handover*.

Enquanto o computador móvel migra dentro de um mesmo domínio, isto é, o endereço do GFA em *Router Advertisement* se mantém constante e o endereço RFA varia, o computador móvel faz apenas atualizações locais ao domínio. Isso é feito através do envio da mensagem *Regional Registration Request* passando o novo CoA ao GFA. O GFA então faz a atualização e responde com *Regional Registration Reply*.

A hierarquia pode ter vários níveis, como uma árvore de RFA's. Essa árvore pode ser construída de forma arbitrária, mas apropriada de acordo com a decisão do administrador de rede. Neste caso, a mensagem *Agent Advertisement* contém uma seqüência de CoAs referentes aos RFA's que compõem a hierarquia, desde o GFA até o RFA no nível mais baixo.

Quando ocorre um *handover*, o computador móvel compara a seqüência de CoAs contida na atual mensagem *Agent Advertisement* com a seqüência obtida da mensagem recebida anteriormente, a fim de encontrar o RFA comum (que aparece nas duas seqüências) e que está no mais baixo possível nível hierárquico. A mensagem de atualização é enviada a partir do RFA no nível mais baixo da hierarquia e é passada ao RFA no nível superior até chegar a este RFA comum. Com este mecanismo é possível reduzir o número de elementos envolvidos no procedimento de *handover* e minimizar a distância (em número de saltos) a ser percorrida pela mensagem de atualização. No pior caso, porém, o RFA comum é o próprio GFA.

Um dos principais problemas desse mecanismo é manter a eficiência no encaminhamento de pacotes uma vez que os mesmos precisam ser tunelados seguidas vezes.

Hierarchical Mobile IPv6

O Mobile IPv6 também possui uma extensão para prover gerenciamento hierárquico de mobilidade, denominada Hierarquical Mobile IPv6 [12] (HMIPv6). Assim como o HMIPv4, o prin-

cipal objetivo é reduzir as mensagens de atualização, a fim de diminuir a latência e a chance de perda de pacotes em *handovers*. Porém, o HIPv6 introduz alguns novos mecanismos, conforme descrevemos a seguir.

O HMIPv6 introduz um novo elemento, o *Mobile Anchor Point* (MAP). De maneira semelhante ao GFA do HMIPv4, o MAP serve como uma referência para um computador móvel em um domínio estrangeiro, sendo o intermediário na comunicação entre o HA e o computador móvel. Porém, no HMIPv6, o MAP pode estar localizado em qualquer nó da rede, inclusive em roteadores de acesso (estações base). Além disso, podemos ter mais de um MAP em um mesmo domínio e um computador móvel pode estar registrado em mais de um MAP simultaneamente. Isso permite um balanceamento de carga sobre a utilização da largura de banda na rede fixa uma vez que o computador móvel pode associar cada MAP a um grupo de nós correspondentes.

Assim como no HIPv4, ao registrar-se em um domínio, um computador móvel é associado a dois endereços: o RCoA (*Regional Care-of Address*), que corresponde ao endereço do MAP (que é o endereço registrado no HA e nós correspondentes) e o LCoA (*On-link Care-off Address*), que corresponde ao atual ponto de acesso.

Com relação aos *Router Advertisements*, a principal diferença é que estes podem conter um ou mais endereços MAP, estando cada endereço MAP associado a alguns valores como preferência e distância, entre outros. Preferência indica o grau de disponibilidade de um MAP e distância corresponde à "distância" (em número de *hops*) do mesmo até um roteador de acesso. Esses valores são configurados inicialmente por um MAP em mensagens *Router Advertisements* e estas são difundidos na rede. A preferência é especificada como um valor inteiro entre 0 e 9, quanto maior o valor, maior a indicação do grau de disponibilidade. O valor inicial de distância é igual a 1 e é incrementado cada vez que a mensagem (*Router Advertisement*) passa por um roteador. Dessa forma, um *Router Advertisement* indica não apenas a presença de um MAP em um domínio como também a sua distância ao roteador de acesso. Esta informação pode ser utilizada pelo computador móvel para a seleção de um MAP para o encaminhamento de pacotes.

A escolha de um MAP pode depender de diversos fatores. Porém, a princípio deve ser escolhido um MAP cujo valor de preferência seja o mais alto possível (o MAP que tem a maior disponibilidade). Além disso, a escolha de um MAP mais distante reduz a probabilidade de mudança de MAPs quando a freqüência de migrações é alta.

Quando um computador móvel migra dentro de um domínio, apenas o MAP é informado sobre o seu novo LCoA. Como uma melhoria, o computador móvel pode enviar uma mensagem para o antigo MAP para proceder o redirecionamento de pacotes para o novo MAP.

4.2.2 Fast Handover

O *Fast Handover*, com versões para o Mobile IPv4 [44] e Mobile IPv6, se baseia no Mobile IP Hierárquico e tem como principal objetivo agilizar o procedimento de *handover* através da detecção antecipada de uma migração pelo computador móvel. Ambas as versões possuem princípios e funcionalidades semelhantes. Portanto, nesta seção daremos um enfoque maior à versão do Fast Handover para o Mobile IPv6.

O *Fast Handover* supõe a possibilidade de interação com a camada de enlace a fim de "descobrir" pontos de acesso que são potenciais candidatos a se tornarem o novo ponto de acesso ao qual o computador móvel irá se conectar após o *handover*. Isto permite que a nova estação base seja notificada (através da antiga estação base) antes que o *handover* ocorra de fato.

A descoberta de novos pontos de acesso depende do mecanismo específico na camada de enlace (tecnologia sem fio). Na maioria dos casos, é feita uma constante medição dos sinais emitidos pelas estações base e, com base nisso, é possível analisar a qualidade do sinal e identificar a iminência de um *handover*. Quando isso ocorre, é feita uma sinalização da camada de enlace para a camada de rede através de um evento (*L2 trigger*).

Essas informações sobre os pontos de acesso disponíveis são embutidas em mensagens *Proxy Router Advertisement* (PrRtAdv). Um *handover* pode ser iniciado pelo computador móvel ou pela rede (estação base), dependendo de quem recebe o evento da camada de enlace (*L2 trigger*) indicando a iminência de um *handover*. No caso de *handover* iniciado pelo computador móvel, este envia uma mensagem *Router Solicitation for Proxy Advertisement* (RtSolPr) para a atual estação base a fim de pedir informações sobre pontos de acesso disponíveis na nova estação. No caso de *handover* iniciado pela rede, a estação base difunde a mensagem.

Podemos ter dois tipos de *handover*: pró-ativo e reativo. O *handover* pró-ativo ocorre quando o computador móvel faz o registro na nova estação base através da atual estação base. Isso garante a disponibilidade de um ponto de acesso na nova estação base e o tunelamento de pacotes da atual estação base para a nova estação base. Quando o computador móvel se conecta no novo ponto de acesso, este já começa a receber os pacotes re-encaminhados pela antiga estação base.

No segundo caso, *handover* reativo, o computador móvel se registra diretamente junto à nova estação base. Isto ocorre, por exemplo, quando o computador móvel perde a conexão repentinamente com a atual estação base. Neste segundo caso não há garantia de que o ponto de acesso candidato está de fato disponível. No *handover* pró-ativo, a atual estação base checa essa disponibilidade com a nova estação base. No caso em que o ponto de acesso não está disponível, é preciso selecionar um novo ponto de acesso e isso pode causar alguma latência no *handover*. Após isso, a nova estação base notifica a antiga estação base e a partir de então os pacotes

são re-direcionados para a nova estação base, isso também causa um atraso no recebimento de pacotes.

4.2.3 IDMP - Intra-Domain Mobility Management Protocol

O IDMP (*Intra-Domain Mobility Management Protocol*) [20] é uma extensão do protocolo de mobilidade intra-domínio proposto pelo TeleMIP (*Telecommunications-Enhaced Mobile IP Architecture*) [21]. A arquitetura proposta pelo TeleMIP emprega alguns conceitos do Mobile IP Hierárquico, porém introduz novos elementos funcionais para gerenciar mobilidade em um domínio.

A rede (domínio) é dividida em várias subredes de acordo com a localização geográfica. O MA (*Mobility Agent*), semelhante ao GFA do Mobile IP Hierárquico, é o responsável por redirecionar pacotes dentro do domínio. Um SA (*Subnet Agent*) possui a função similar a um FA ou um servidor DHCP, para prover CoA ou *collocated* CoA, respectivamente.

Um computador móvel possui dois endereços de CoA: *Local Care-of Address* (LCoA), que é um endereço associado a uma subrede, indica a localização de um computador móvel dentro do domínio, e *Global Care-of Address* (GCoA) que indica a localização do computador móvel em nível de domínios. Diferentemente de outros protocolos de micro-mobilidade, o IDMP não supõe que o gerenciamento de macro-mobilidade seja feito pelo Mobile IP. Em vez disso, após o primeiro registro de um computador móvel em um domínio (causando uma atribuição de LCoA e GCoA), o próprio computador móvel deve notificar o GCoA sobre seu atual endereço aos nós correspondentes ou ao HA (no caso de empregar o Mobile IP) usando os protocolos correspondentes.

Pacotes para um computador móvel em um domínio são direcionados ao GCoA e interceptados pelo MA. O MA, por sua vez, direciona os pacotes ao corrente LCoA.

Um *handover* intra-domínio ocorre quando o computador móvel muda de subrede em um mesmo domínio. A detecção de mobilidade é feita através de mensagens similares aos *Agent Advertisements* do Mobile IP. Ao receber essa mensagem, o computador móvel requisita um novo LCoA e o SA envia uma mensagem de confirmação passando o LCoA. Finalmente, o computador móvel informa o MA de seu novo LCoA através de uma mensagem de atualização e o MA envia uma confirmação.

Além deste procedimento de *handover* comum que também existe no TeleMIP, o IDMP oferece um segundo procedimento de *handover* (*fast handover*) que depende da interação com a camada de enlace. De maneira semelhante ao protocolo apresentado na Seção 4.2.2, supõe-se que uma indicação de uma iminente mudança na conexão é informada pela camada de enlace. O computador móvel, ou o SA, ao receber essa mensagem, gera uma mensagem *MovementImminent*

para o MA. A partir disso, o MA começa a replicar os pacotes destinados ao computador móvel para todas os SA's vizinhos do atual SA. No recebimento desses pacotes, cada SA os armazena em um *buffer*. Quando o computador móvel se registra com o novo SA, este logo em seguida começa a enviar os pacotes que estão no *buffer*. Em seguida, o computador móvel atualiza o LCoA no MA.

4.2.4 Cellular IP

O Cellular IP [8, 66, 9], proposto pelo consórcio da Universidade de Columbia com a Ericson, é um protocolo de micro-mobilidade IP que incorpora alguns mecanismos tipicamente encontrados em redes celulares como, por exemplo, suporte a usuários inativos (*idle users*) e *Paging*, além de técnicas para acelerar o procedimento de *handover* e melhorar o desempenho quando há freqüentes migrações de computadores móveis.

A forma de encaminhamento de pacotes em uma rede Cellular IP é semelhante ao mecanismo de roteamento *hop-by-hop* empregado pelo protocolo IP. Porém, uma vez que os computadores na rede mudam seus endereços físicos dinamicamente, estruturas específicas de *cache* de roteamento são implementadas nos nós da rede, a fim de manter esses endereços e gerenciar a localização dos computadores móveis no domínio.

Dentro de uma rede Cellular IP, um computador móvel é identificado pelo seu endereço IP da rede de origem (*home address*) e a informação sobre a sua atual localização no domínio é mantida de forma distribuída nos *caches* de roteamento nos roteadores. Os *caches* de roteamento armazenam informações parciais sobre a localização de um computador móvel: de maneira simplificada, pode-se dizer que um registro no *cache* associa o seu identificador (*home address*) ao "próximo nó" no caminho em direção à corrente estação base em que se encontra o mesmo. A seqüência de nós do *gateway* até a estação base forma o caminho de roteamento de pacotes até o computador móvel.

Os registros nos *caches* de roteamento permanecem válidos por um determinado período de tempo (*soft-state*) e antes que o prazo se expire, estes registros são atualizados. O Cellular IP aproveita os pacotes de dados provenientes do próprio computador móvel para atualizar esses registros. Isto reduz o tráfego de mensagens de atualização de localização na rede. Porém, no caso em que um computador móvel não faz o envio de pacotes, este deve periodicamente enviar mensagens de controle (*Route-update packets*) a fim de manter esses registros ativos. A principal vantagem da utilização desse tipo de *cache* é que as entradas nos *caches* no antigo caminho não precisam ser removidas a cada migração, pois estas se expiram e são eliminadas automaticamente.

Um computador móvel detecta sua mobilidade através de freqüentes medições da potência

dos sinais emitidos pelas estações base e inicia um *handover* de acordo com isso. Há dois tipos de *handover* no Cellular IP: *hard* e *semi-soft handover*.

Hard handover inicia com uma mensagem *Route Update* que é enviada pelo computador móvel para a nova estação base. A nova estação base acrescenta um registro no *cache* de roteamento e passa a mensagem ao próximo nó. Isso é feito por cada um dos nós no caminho até o *gateway*. A simplicidade desse tipo de *handover* causa uma baixa carga de sinalização (mensagens de atualização) na rede mas, em contrapartida, pode acarretar um certo volume de perda de pacotes. Esse número está associado ao tempo necessário para que *Route Update* alcance um nó que já contenha um registro do computador móvel no *cache*. Esse é o primeiro nó na intersecção entre o antigo e novo caminho no sentido estação base-*gateway*, e é chamado de *roteador do cruzamento* (RC) (ou *crossover router*). Quando essa mensagem chega no RC os pacotes destinados ao computador móvel são desviados para a nova estação base. No pior caso, porém, o RC é o próprio *gateway*.

No *semi-soft handover* o procedimento de *handover* é antecipado, configurando-se o caminho de roteamento da nova estação base ao *gateway* antes que o computador móvel se conecte efetivamente à nova estação base. Isto é feito da seguinte forma: quando é detectada uma nova estação base, o computador móvel notifica a mesma através de *Route Update* e permanece "ouvindo" a corrente estação base. Essa mensagem de atualização é passada de nó em nó no caminho em direção ao *gateway*, formando o novo caminho de roteamento. Quando essa mensagem atinge o RC, os pacotes de dados destinados ao computador móvel são replicados para ambos os caminhos, isto é, para a antiga e nova estação base. Após um intervalo de tempo pré-estabelecido (*semi-soft interval*), o computador móvel se conecta à nova estação base e recomeça a receber os pacotes de dados a partir da mesma.

Em comparação com o *hard handover*, o *semi-soft handover* reduz significantemente a perda de pacotes uma vez que durante a configuração do novo caminho de roteamento o computador móvel continua recebendo pacotes da antiga estação base. Um problema com o *semi-soft handover* é a necessidade da existência de algum mecanismo para tratar a diferença entre os possíveis atrasos na entrega de pacotes na antiga e nova estação base. Estes atrasos podem ser influenciados pela "distância" (em número de saltos do *gateway* à uma estação base), assim como a carga de mensagens na rede. Se a antiga estação base recebe pacotes com maiores atrasos do que a nova estação base, haverá perda de pacotes (a nova estação base está "adiantada" com relação à antiga estação base). Caso contrário, se a antiga estação base recebe pacotes com menores atrasos, então haverá duplicação de pacotes (a nova estação está "atrasada" com relação à antiga estação base). Para evitar esse problema, o Cellular IP propõe um mecanismo de atraso (*delay device*) que é uma espécie de *buffer* e que permite manter os pacotes por um intervalo

de tempo. Uma outra restrição com relação ao *semi-soft handover* é que o mesmo requer que um computador móvel tenha capacidade de se conectar a mais de uma estação base ao mesmo tempo, mas isso depende da tecnologia comunicação sem fio e nem sempre é possível.

4.2.5 HAWAII

HAWAII (*Handoff-Aware Wireless Access Internet Infrastructure*) [26] foi proposto pela Lucent Technologies para oferecer um suporte eficiente à micro-mobilidade cujo maior enfoque está na otimização de rotas para o encaminhamento de pacotes.

Nas redes HAWAII, o *gateway* (chamdo de *domain root router*) possui dois principais papéis: serve de HA (*home agent*) para os computadores móveis que inicialmente foram registrados em seu domínio (e que o tem como *home domain*), e serve como FA (*foreign agent*) para os computadores móveis registrados em qualquer outro domínio, e neste caso é chamado de *foreign domain*.

Em uma rede HAWAII um computador móvel é identificado pelo seu endereço IP enquanto se movimenta dentro de um domínio, porém, quando migra para um novo domínio é associado a um *co-located care-of address* (ccoa) pelo *domain root router* (DDR) deste novo domínio (FA). Este ccoa é registrado no HA em seu *home domain* de modo que os pacotes destinados ao mesmo possam ser tunelados pelo HA para o FA.

Dentro de um domínio pacotes destinados a um computador móvel são encaminhados usando roteamento IP. Os nós em uma rede HAWAII são roteadores IP com algumas extensões para tratar as mensagens de controle do protocolo HAWAII. Esses roteadores mantêm entradas da forma <*endereçoIP*, *interface*>, onde *endereçoIP* corresponde ao endereço IP do computador móvel (*home address*) e *interface* corresponde à interface relativa ao próximo nó em direção à estação base em que o computador móvel está correntemente conectado. Assim como no Cellular IP, essas entradas são *soft-state*, ou seja, são válidas por um certo período de tempo e são removidas quando este tempo se expira. Porém, uma diferença com relação ao Cellular IP, para manter válidas essas entradas nos roteadores e estações base, os computadores móveis enviam freqüentemente mensagens de atualização específicas, as entradas não podem ser atualizadas com dados transmitidos pelo computador móvel como no Cellular IP.

Ao detectar a necessidade de *handover*, o computador móvel envia uma mensagem *Path Setup* para a nova estação base a fim de iniciar o *handover*. Essa mensagem é direcionada à antiga estação base, passando pelo RC. A atualização da rede e a entrega de pacotes durante o *handover* depende do tipo de esquema de atualização de caminho empregado. Essencialmente, o HAWAII possui dois esquemas de atualização de caminhos e algumas possíveis variações destes:

- (1) *forwarding path setup scheme*: pacotes da antiga estação base são redirecionados para

a nova estação base. Há duas variações: (a) redirecionamento em um único fluxo de pacotes (*Single Stream Forwarding* - SSF), ou seja, os pacotes são redirecionados antes que novos pacotes sejam enviados ao computador móvel, novos pacotes são mantidos no RC; (b) redirecionamento em múltiplos fluxos de pacotes (*Multiple Stream Forwarding* - MSF), ocorre quando novos pacotes são enviados à nova estação base assim que a mensagem de atualização alcance o RC e estes se intercalam com os pacotes sendo redirecionados.

- (2) (*non-forwarding path setup scheme*): não há o redirecionamento de pacotes da antiga para a nova estação base. Aqui também há duas variações: (a) sem replicação de pacotes (*Unicast Non-Forwarding* - UNF), onde os novos pacotes que chegam em RC começam a ser desviados para a nova estação base assim que a mensagem de atualização é recebida; (b) com replicação de pacotes (*Multicast Non-Forwarding* - MNF), onde os novos pacotes são enviados para a antiga e nova estações base pelo RC por um período de tempo.

Esquemas do tipo (1), com redirecionamento de pacotes, reduzem a perda de pacotes e podem manter a ordenação de pacotes (como no caso (a)), porém com alguma latência na entrega. Já no caso de esquemas do tipo (2), ou seja, sem redirecionamento, essa latência na entrega pode ser menor, porém com alguma perda de pacotes.

4.2.6 Multicast-based Mobility (M&M)

Multicast-based Mobility (M&M) [37] é uma arquitetura de mobilidade intra-domínio baseada na difusão de mensagens e foi proposta pela Universidade de Southern California. O IP *Multicast* permite a entrega eficiente de pacotes independentemente da localização. Este conceito é empregado pelo M&M para reduzir a latência e perda de pacotes durante o *handover*.

A idéia básica desta abordagem é replicar pacotes para todas as estações base cujas áreas de cobertura sejam adjacentes à atual área de cobertura onde o computador móvel se encontra. Dessa forma, seja qual for a célula vizinha para a qual o computador móvel migre, a nova estação base poderá lhe enviar pacotes logo após a conexão, reduzindo a latência na entrega de pacotes. As estações base adjacentes fazem parte do grupo *multicast* e cada vez que ocorre um *handover*, o grupo *multicast* é atualizado através da entrada/saída de estações base no grupo (através de operações join/leave).

Assim como os outros protocolos de micro-mobilidade apresentados, este protocolo também se baseia no Mobile IP para tratar *handover* inter-domínio. Cada computador móvel em um domínio é associado a um endereço *multicast* (MCoA - *Multicast Care-of Address*), que é utilizado para o encaminhamento de pacotes e um endereço *unicast* (RCoA - *Regional Care-of Address*),

que é o endereço registrado no HA. Ambos endereços se mantêm fixos enquanto o computador móvel está no mesmo domínio.

O *gateway* faz a alocação e manuntenção de endereços *multicast* dos computadores móveis no domínio. Dessa forma, pacotes destinados ao computador móvel são tunelados ao *gateway* e então são enviados ao endereço *multicast*. Também foi proposta uma abordagem algorítmica na qual endereços *multicast* IPv6 (MCoA) são inferidos automaticamente a partir de endereços *unicast* IPv6 (RCoA), de modo que o gerenciamento centralizado no *gateway* não seja mais necessário. Maiores detalhes podem ser encontrados em [37].

Um *handover* é iniciado pelo computador móvel ao detectar uma migração através da medição da potência de sinais das estações base. Assim que um computador móvel se conecta com a nova estação base, esta envia uma mensagem *Join* para todas as estações base vizinhas passando o MCoA. Ao receber esta mensagem, cada uma dessas estações base entra para o grupo *multicast* e começa a receber réplicas dos pacotes destinados ao computador móvel. Em seguida, a nova estação base envia uma mensagem *Handover* para a antiga estação base. Através dessa mensagem, a antiga estação base notifica as suas estações base vizinhas para saírem do grupo *multicast* através da mensagem *Leave*, passando o MCoA.

Esta abordagem para o tratamento de *handover* permite reduzir a latência e conseqüentemente a perda de pacotes durante a transição. Porém, há um considerável uso de recursos e além disso, um computador móvel pode receber múltiplas cópias de dados. Outras abordagens propostas como o IDMP (Seção 4.2.3), por exemplo, fazem a replicação de pacotes somente durante o *handover* e não durante todo o tempo, como é feito neste caso. Em [62], são propostas algumas políticas para determinar quais/quantas estações base devem receber réplicas, por exemplo, de acordo com as características de mobilidade (rápido, médio, devagar) de forma a evitar que a replicação seja feita para toda a vizinhança de uma estação base.

4.2.7 Comparação dos Protocolos de Micro-Mobilidade

Nesta seção apresentamos uma comparação entre os protocolos de *handover* para micro-mobilidade apresentados nas seções acima. Em particular, enfatizamos as técnicas de *handover* suave empregadas para reduzir a perda de pacotes e latência.

A latência do procedimento de *handover* em protocolos de micro-mobilidade IP (isto é, *handover* na camada de rede) é causada, basicamente, por dois fatores principais: a latência para a detecção de mobilidade, isto é, o tempo em que um computador móvel demora para identificar uma nova estação base após uma migração e a latência para a atualização da rede, ou seja, o tempo para que a informação de localização seja atualizada e o computador móvel receba o primeiro pacote a partir da nova estação base. Essas latências dependem da forma particular como

são tratadas essas tarefas (detecção de mobilidade e atualização da rede) em cada protocolo de *handover*.

No Mobile IP, a detecção de mobilidade é feita através do mecanismo de *Agent Advertisements* difundidos pelas estações base. Um dos principais problemas com este mecanismo é que um computador móvel detecta uma migração para uma nova localização através desta mensagem, quando possivelmente já não tem mais conexão com a antiga estação base (*handover* reativo). Como vimos, muitos protocolos de micro-mobilidade mantêm esse mecanismo para a detecção de migração, mas alguns utilizam informações da camada de enlace a fim de prever a nova estação base e antecipar o *handover*.

A atualização na rede depende da "distância" (em número de saltos) que uma mensagem de atualização precisa percorrer na rede até atingir o elemento que mantém informações sobre a localização de um computador móvel (por exemplo, HA no Mobile IP). Quanto maior a distância, maior é a latência. Para reduzir essa latência, alguns protocolos propõem o uso de hierarquias a fim de "aproximar" o ponto de atualização do computador móvel. Porém, quanto mais próximo esse ponto, maior é a ocorrência de *handovers* no caso em que há freqüentes migrações. O uso de *Mobility Anchor Points* que é empregado no HMIPv6 (Seção 4.2.1), permite que o ponto de atualização (MAP) seja alocado em qualquer nó no domínio de modo que possa ser selecionado de acordo com as preferências de um computador móvel, por exemplo, o seu grau de mobilidade.

Em protocolos baseados em roteamento específico (Cellular IP, HAWAII), o ponto de atualização corresponde ao roteador no cruzamento entre o antigo e novo caminho (RC), que no pior caso é o *gateway*. Nos protocolos que usam hierarquias em apenas dois níveis, o ponto de atualização é sempre o *gateway*.

A forma de transmissão de pacotes no domínio também tem influência no desempenho de um *handover*. Tunelamento permite que pacotes sejam direcionados na rede sem a modificação dos roteadores IP. Porém, esta técnica dificulta a reserva de recursos para a provisão de QoS, uma vez que o encapsulamento esconde as informações atrás de um novo cabeçalho. Além disso, a utilização de hierarquias de FAs em vários níveis é ineficiente devido ao tunelamento e re-tunelamento sucessivos. O *multicast* permite reduzir a latência na entrega de pacotes através da difusão de pacotes, porém, pode acarretar em uma maior utilização dos recursos e carga na rede, além de causar um considerável número de duplicação de pacotes.

Na Tabela 4.1 apresentamos esses aspectos mencionados acima para cada um dos protocolos de micro-mobilidade. A coluna "Custo" indica a latência causada pelo procedimento de atualização na rede que é proporcional à distância (em número de saltos). A coluna "Nós envolvidos" indica quantos elementos na rede participam de um procedimento de atualização.

Protocolo	Detecção de mobilidade	Atualização na rede	Custo	Nós envolvidos	Transmissão
HMIPv4	AA	msg ao GFA	$d(EB_n, RC)$	FAs, HA	tunelamento
HMIPv6	AA	msg ao MAP	$d(EB_n, MAP$	EB_n	tunelamento
Fast Handover MIPv6	L2 + AA	msg ao GFA msg ao EB_o	$d(EB_n, GFA)$+ $d(EB_o, EB_n)$	EB_o, EB_n	tunelamento
IDMP Básico	AA	msg ao MA	$d(EB_n, MA)$	EB_n, MA	tunelamento
IDMP Fast HO	L2	msg ao MA	$d(EB_n, MA)$	EB_n, MA	tunelamento
CIP Hard HO	L2	msg ao GW	$d(EB_n, RC)$	EB_n, $n(EB_n, RC)$	unicast
CIP Soft HO	L2	msg ao GW	$d(EB_n, RC)$	EB_n, $n(EB_n, RC)$	unicast
HW SSF e MSF	AA	msg ao EB_o	$d(EB_n, EB_o)$	$n(EB_o, EB_n)$	unicast
HW UNF e MNF	AA	msg ao EB_o	$d(EB_n, EB_o)$	$n(EB_o, EB_n)$	unicast
M&M	L2	join/leave	join/leave oper.+ $d(EB_n, EB_o)$	$EB_o - adj$, $EB_n - adj$	multicast

Tabela 4.1: Comparação dos protocolos de micro-mobilidade

Legenda:

HMIPv4	:	Hierarchical Mobile IPv4
Fast MIP	:	Mobile IP Fast Handover
IDMP	:	Intra-Domain Mobility Protocol
IDMP Fast HO	:	Intra-Domain Mobility Protocol Fast Handover
CIP Hard HO	:	Cellular IP Hard Handover
CIP Soft HO	:	Cellular IP Soft Handover
HW SSF	:	Hawaii Single Stream Forwarding
HW MSF	:	Hawaii Multiple Stream Forwarding
HW UNF	:	Hawaii Unicast Non-Forwarding
HW MNF	:	Hawaii Multicast Non-Forwarding
M&M	:	Multicast-based Mobility
AA	:	*Agent Advertisement*
L2	:	*handover* auxiliado pela camada de enlace
HO	:	*handover*
GFA	:	Gateway Foreign Agent
MAP	:	Mobile Anchor Point
EB_n e EB_n	:	antiga e nova estação base, respectivamente
EB_{adj}	:	grupo de estações base adjacentes a EB
$d(A, B)$	:	distância de um nó A a um nó B
$n_{A,B}$	:	número de nós no caminho de A a B

Outras técnicas têm sido empregadas a fim de melhorar o desempenho do *handover*, conforme apresentamos na Tabela 4.2. Protocolos que redirecionam pacotes de uma estação base para outra usam a técnica de Buffer+Forward, como o Fast MIP, HW SSF e MSF. PréHandover depende da detecção antecipada de mobilidade (*handover trigger*) que é gerado pela camada de enlace. Alguns protocolos usam essa técnica para gerar um caminho de roteamento de pacotes para a nova estação base e a partir disso, usam a técnica para o redirecionamento de pacotes da antiga para a nova estação base através desse novo caminho (Fast MIP). Outros usam a técnica de replicação de pacotes para ambas as estações base por um período de tempo (Bicast, no CIP Soft HO) ou fazem a replicação de pacotes para todas as estações base vizinhas da atual célula (Multicast, no IDMP).

Em comparação ao IDMP, o M&M faz a replicação de pacotes para as estações base vizinhas durante todo o tempo de execução do protocolo, enquanto que o IDMP o faz somente durante um *handover*.

Em comparação com o CIP Soft HO, o HW MNF também faz a replicação de pacotes (Bicast) para a nova estação base porém, ao contrário do CIP Soft HO, o HW MNF não usa PreHandover, o *handover* é iniciado a partir da nova estação base e o CR ao receber a mensagem de atualização de caminho, começa a replicar pacotes de dados para ambas estações base por um período de tempo.

Protocolo	Pré-handover	*Bicast*	*Multicast*	*Buffer*	*Forward*
HMIPv4	não	não	não	não	não
Fast MIP	sim	não	não	sim	sim
IDMP Básico	não	não	não	não	não
IDMP Fast HO	sim	não	sim	sim	não
CIP Hard HO	não	não	não	não	não
CIP Soft HO	sim	sim	não	não	não
HW SSF e MSF	não	não	não	sim	sim
HW UNF	não	não	não	não	não
HW MNF	não	sim	não	não	não
M&M	sim	não	sim	não	não

Tabela 4.2: Técnicas de handover suave

Conforme apresentamos nesse capítulo existe uma diversidade muito grande de abordagens e técnicas para *handover* suave para micro-mobilidade. Porém, cada uma dessas soluções é mais adequada para certos tipos de requisitos de aplicações e tipos de rede.

Em vista disso, é desejável ter um ambiente em que fosse possível combinar, prototipar e

avaliar estas e outras técnicas em diferentes cenários de simulação. Na próxima seção, apresentamos o HOPF: um arcabouço para prototipação, simulação e teste de protocolos de *handover* suave a partir da composição de módulos (técnicas) independentes.

Capítulo 5

HOPF: HandOver Protocol Framework

Neste capítulo apresentamos o HOPF: *HandOver Protocol Framework*, um *framework* para prototipação, teste e simulação de protocolos de *handover* suave para micro-mobilidade. A principal característica do HOPF é a utilização de unidades de composição independentes, que chamamos de *módulos canônicos*, que permite uma composição flexível de protocolos de *handover* suave.

Conforme vimos na Seção 3, a abordagem de se combinar pequenos módulos (unidades de composição) tem sido empregada para permitir a composição de protocolos de modo que melhor atendam os requisitos da aplicação/usuário, sistema operacional ou recursos disponíveis. Em particular, o uso dessas unidades de composição oferece algumas vantagens para o desenvolvimento de protocolos/sistemas, como: reusabilidade, configurabilidade, extensibilidade e eficiência, conforme mencionamos na Seção 3.5.

Protocolos baseados em IP para micro-mobilidade (Seção 4.2) empregam algumas técnicas e otimizações a fim de reduzir a latência e a perda de pacotes causados pelo Mobile IP durante o *handover* em casos de mobilidade freqüente em um mesmo domínio administrativo. O principal problema com essas abordagens é que um conjunto fixo de técnicas e otimizações não é suficiente para tratar os diferentes requisitos das aplicações, com diferentes padrões de mobilidade de usuário e características da rede.

Por exemplo, o emprego de hierarquias de FA's isoladamente (como no caso do Mobile IP Hierárquico) não é adequado quando a aplicação é sensível a perdas de pacotes e ao atraso. Isto porque, em particular, quando a hierarquia possui apenas dois níveis (i.e., FA's localizados no *gateway* de domínio e nas estações base), a mensagem de atualização (*Binding Update*) deve ser encaminhada até o *gateway*, o que é um processo custoso principalmento quando a freqüência de migração do usuário móvel é alta e a distância média (número de *hops*) entre *gateway* e estações

base é grande. Uma simples melhoria para este caso poderia ser, por exemplo, a utilização de uma estrutura para o armazenamento temporário de pacotes nas estações base (*buffer*) para possibilitar o redirecionamento posterior para a nova estação base e, desta forma, reduzir o número de pacotes perdidos durante o processamento do *handover*. Isto, porém, não alivia o problema do atraso na entrega de pacotes. Um exemplo de estratégia para tratar ou, reduzir o atraso na entrega de pacotes poderia ser, por exemplo, o uso da técnica de replicação de pacotes para uma ou mais estações base cujas áreas de cobertura são adjacentes.

Dessa forma, conforme observamos no exemplo anterior, combinações específicas de técnicas ou otimizações podem produzir protocolos de *handover* adaptados que possam oferecer *suavidade* às aplicações durante as migrações de um usuário móvel. Porém, projetar um protocolo de *handover* suave específico para um determinado contexto é uma tarefa complexa devido à dificuldade para selecionar, combinar e testar diferentes técnicas e otimizações que possam satisfazer de maneira eficiente os requisitos iniciais.

Isso nos motivou a desenvolver este *framework*, que é baseado em elementos estruturais (*módulos canônicos*) onde cada um deles implementa uma técnica ou otimização comumente empregados em diferentes protocolos de *handover* para micro-mobilidade propostos na literatura. Esses módulos canônicos podem ser compostos em componentes específicos para tratar uma determinada tarefa do procedimento de *handover* e gerar protocolos de *handover* adaptados a partir de distintas combinações de módulos. Utilizando este *framework*, esses protocolos podem então ser testados e simulados empregando-se diferentes parâmetros de simulação e de rede. A partir dos resultados de simulação, os protocolos podem ser avaliados e comparados a fim de detectar as técnicas ou otimizações que melhor se adequam aos requisitos específicos da aplicação com relação à suavidade do *handover*.

5.1 Decomposição de Protocolos de Handover

Com a finalidade de identificar módulos canônicos e suas categorias assim como os componentes funcionais do *framework*, o primeiro passo foi a decomposição de protocolos de *handover* em um conjunto de tarefas envolvidas nesse procedimento.

Na Seção 2.2, apresentamos essas tarefas que são executadas durante o procedimento de *handover*. Conforme mencionamos nessa seção, o *handover* ocorre na verdade em dois níveis: no nível de enlace, onde são executadas a detecção de mobilidade, alocação e liberação de canais e a transferência da conexão de uma estação base para outra e, no nível de rede, onde o *handover* geralmente ocorre após o *handover* na camada de enlace e de forma independente. E, devido a isso, nesse caso também há um procedimento para a detecção de mobilidade (porém, com

diferentes técnicas daquelas adotadas na camada de enlace) e a atualização na rede para manter a continuidade do fluxo de pacotes sendo encaminhados ao computador móvel. Em ambos os casos, a detecção de mobilidade identifica o momento em que é necessário a execução de um *handover*.

Em particular, protocolos de micro-mobilidade se concentram em estratégias para tratar o *handover* na camada de rede. Dessa forma, as tarefas associadas ao *handover* na camada de enlace não são consideradas por esses protocolos. Porém, a título de ilustração, acrescentamos em nosso *framework* um componente que agrupa essas tarefas relativas ao *handover* na camada de enlace a fim de que haja a possibilidade de comtemplar, em um trabalho futuro, distintos tipos de protocolos de *handover*, não limitado para o caso de micro-mobilidade, como, por exemplo, protocolos de *handover* em redes celulares, nos quais a ênfase está, essencialmente, na camada de enlace.

Um dos principais objetivos dos protocolos de micro-mobilidade é a execução eficiente da atualização da rede após uma migração. Diferentes técnicas têm sido empregadas para tratar, em particular, as tarefas de atualização da localização de um computador móvel em um ou mais elementos de rede que mantém essa informação e a atualização de caminho de roteamento de pacotes para a nova localização. Para possibilitar a composição de diferentes estratégias para tratar essas tarefas, estas foram agrupadas em um outro componente.

Alguns protocolos de micro-mobilidade empregam otimizações que visam antecipar o *handover* através de uma interação com a camada de enlace (Seção 4.2.2). Neste caso, algumas ações como a pré-configuração do caminho de roteamento para uma ou mais estações base, a replicação de dados para estas estações, entre outros, são executadas para reduzir a latência causada pelo *handover* tradicional. Essas tarefas são independentes daquelas associadas à atualização da rede, descritas no parágrafo anterior, pois os registros de localização anteriores (referentes à antiga localização do computador móvel) são mantidos até o final do *handover* em paralelo com os novos registros. Além disso, o tipo de evento que dispara essas tarefas também é distinto.

Além das tarefas específicas do *handover*, algumas técnicas e otimizações foram propostas para tratar o fluxo de pacotes destinados a um computador móvel durante o *handover*, a fim de se evitar a perda de pacotes e latência na entrega.

De acordo com isso, identificamos a necessidade de quatro componentes principais que abrangem as seguintes tarefas de um *handover* suave: detecção de mobilidade e início do *handover*, atualização na rede, otimização do fluxo de pacotes e antecipação de *handover*.

Identificamos e extraímos as técnicas/estratégias empregadas em diferentes protocolos de *handover* para micro-mobilidade separando-as em categorias de acordo com o tipo de tarefa ou funcionalidade relacionadas. Essas técnicas/estratégias foram mapeadas em módulos canônicos.

Na Seção 5.2.1 apresentamos uma lista de módulos canônicos e suas categorias.

5.2 Arquitetura e Componentes

O principal foco do HOPF está na execução e simulação de protocolos de *handover* adaptados e gerados a partir da composição de módulos canônicos. Para este fim, o HOPF se baseia nos seguintes elementos: um conjunto extensível de componentes para tratar as tarefas do *handover*, um elemento que faz o controle de execução (Controller) e uma biblioteca extensível de módulos canônicos. Os componentes executam as tarefas do *handover* de acordo com os módulos selecionados para um protocolo, uma vez que estes módulos implementam técnicas ou estratégias para tratar essas tarefas. Dessa forma, o comportamento de um componente para diferentes protocolos (diferentes composições de módulos) pode ser distinto dependendo dos módulos canônicos selecionados.

A seleção de módulos canônicos é feita em uma etapa inicial, a partir da biblioteca de módulos canônicos e de acordo com algumas informações específicas como, por exemplo, os requisitos de QoS da aplicação, o perfil de mobilidade do usuário e as características da rede. A escolha e combinação de módulos a partir desses parâmetros é uma tarefa complexa e que dificilmente pode ser automatizada, devido ao grande número de combinações possíveis. No entanto, a partir da experiência de prototipação e simulação de alguns protocolos para micro-mobilidade, identificamos algumas regras empíricas (heurísticas) que podem auxiliar no desenvolvimento de protocolos de *handover* suave utilizando o HOPF (Seção 5.2.3).

Após a seleção dos módulos canônicos, é gerado um fluxo de execução para o protocolo de *handover* sendo descrito, de acordo com os módulos selecionados e dos tipos de eventos que estes devem tratar (Seção 5.2.2). O conjunto de módulos canônicos e o fluxo de execução são utilizados para instanciar o protocolo de *handover* suave no HOPF em cada elemento de rede (estações base, *gateway*, roteadores e computador móvel). Na Figura 5.1 apresentamos uma visão geral do HOPF, ilustrando, de maneira simplificada, as etapas de seleção de módulos canônicos e de execução/simulação.

Nas próximas seções apresentamos os módulos canônicos e suas categorias (além de uma especificação de cada um deles no final deste capítulo), a estrutura e os componentes do HOPF e uma breve descrição sobre a seleção de módulos e regras empíricas de seleção que foram geradas a partir de nossas experiências de simulação com o HOPF.

5.2.1 Módulos Canônicos

Um módulo canônico pode implementar uma única funcionalidade de um protocolo de *han-*

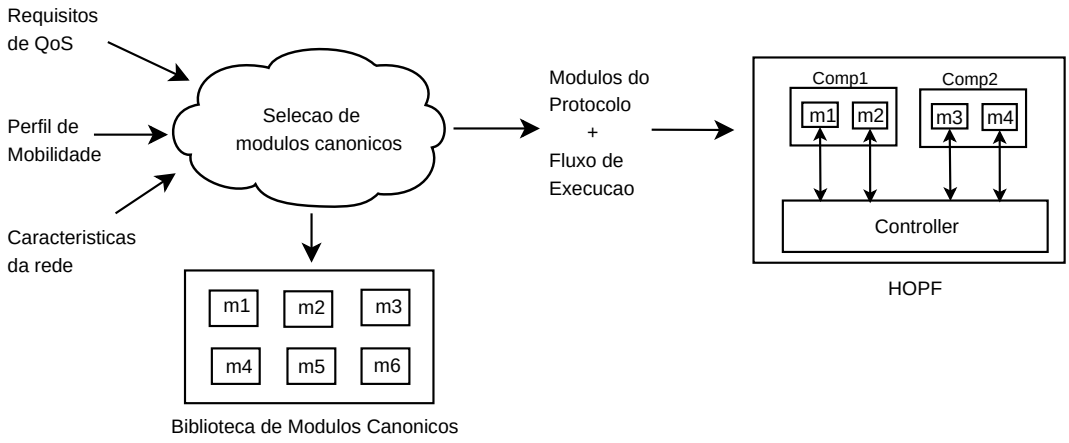


Figura 5.1: Visão geral do HOPF

dover, uma técnica, estratégia ou algoritmo para otimizar o desempenho do protocolo ou mesmo estruturas de dados e elementos (por exemplo, agentes de mobilidade) que fazem parte da infra-estrutura de um protocolo de mobilidade.

Na Figura 5.2, apresentamos um conjunto de módulos canônicos identificados até o momento e que foram separados em distintas categorias de acordo com o tipo de tarefa ou funcionalidade que provêem. Tarefas do *handover*, no centro da figura, podem estar associadas a categorias de módulos canônicos, indicando que um ou mais módulos em uma categoria podem ser empregados para tratar uma tarefa de *handover* (linhas cheias). Módulos canônicos podem ter relações de dependência entre si: um módulo A depende de um módulo B quando A requer a funcionalidade do módulo B para a sua execução (e é representado por uma linha tracejada).

A seguir apresentamos uma descrição das categorias de módulos canônicos e dos módulos canônicos associados.

1. Detecção de mobilidade (*MobDetectionTec*): módulos nesta categoria implementam estratégias para tratar a detecção de mobilidade e início de um *handover*. Através da detecção de mobilidade é possível identificar uma mudança de localização e o momento em que é necessário transferir a conexão de uma estação base para outra. Dentre algumas abordagens para esta tarefa podemos citar: na camada de enlace, através do monitoramento das potências de sinais emitidos pelas estações base (SignalMeasurement); e na camada de rede, por exemplo, no Mobile IP, podemos ter a detecção baseada no *lifetime* das mensagens *Agent Advertisement* (LazyDetection), e a detecção que se baseia na comparação de prefixos de rede contidos nessas mensagens (EagerDetection). As principais vantagens da detecção de mobilidade através do monitoramento de sinais são: a seleção de uma estação base de acordo com a qualidade do sinal e a tomada de decisão antecipada para iniciar

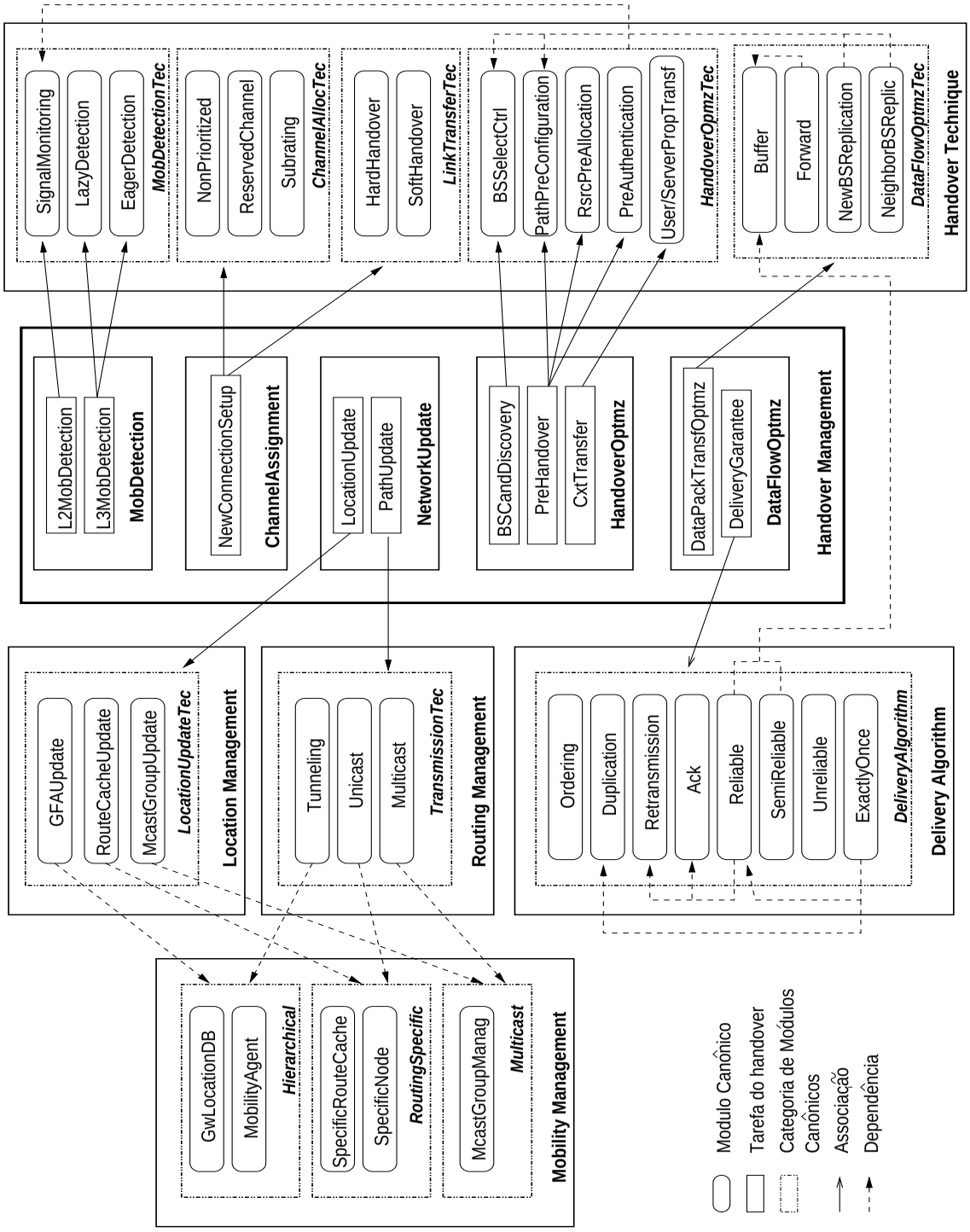


Figura 5.2: Tarefas do *handover* e categorias de módulos canônicos

um *handover*, o que possibilita uma otimização durante a transferência da comunicação de uma estação base para outra. No Mobile IP, uma vez que a detecção de mobilidade se baseia em mensagens (*Agent Advertisement*) difundidas pelos FAs, na maioria das vezes, a tomada de decisão para o início do *handover* se dá quando o computador móvel não possui mais conexão com a antiga estação base e o *handover* ocorre de maneira abrupta, ou seja, não suave.

2. Atribuição de canais (*ChannelAllocTec*): nesta categoria, os módulos implementam algoritmos para tratar alocação de canal na nova estação base a fim de gerar a nova conexão entre o computador móvel e a estação base. Alguns exemplos de esquemas de alocação de canais: NonPrioritized, no qual não há um tratamento especial para os *handovers*, caso não exista um canal disponível na nova estação base a requisição é bloqueada; Reserved-Channel, no qual alguns canais são reservados especialmente para requisições de *handovers*, reduzindo-se a probabilidade de bloqueios pela ausência de canais disponíveis; e Subrating, onde um canal ocupado é dividido em dois canais de capacidades iguais à metade do canal original e a requisição de *handover* é servida por uma delas. Este último esquema, por um lado, pode ser útil no caso em que não há canais disponíveis na nova estação base e, em particular, o bloqueio de uma requisição de *handover* é crítico para o desempenho da aplicação/serviço. Porém, por outro lado, a metade da taxa de transmissão pode não ser suficiente para alguns tipos de aplicações e pode acabar degradando o desempenho.

3. Transferência da conexão (*LinkTransferTec*): módulos que implementam diferentes formas para tratar a transferência da conexão (enlace de rádio), isto é, a ruptura da conexão na antiga estação base e a geração de uma nova conexão na nova estação. Alguns exemplos são: esquema no qual um computador móvel se conecta com apenas uma estação base de cada vez (HardHandover), e esquema onde os computadores móveis recebem/transmitem de/para mútiplas estações base simultaneamente (SoftHandover). No primeiro caso, no intervalo de tempo durante desconexão/conexão com a antiga e nova estações base, respectivamente, o computador móvel fica sem qualquer capacidade de comunicação com a rede, e portanto, todos os pacotes de dados enviados ao mesmo neste intervalo de tempo são perdidos. Já no segundo caso, o computador móvel mantém a conexão com a antiga estação base durante toda a execução do procedimento de *handover*, evitando-se essa interrupção causada pelo primeiro caso. O ideal seria se houvesse uma coordenação entre as duas estações base envolvidas durante um *handover* a fim de minimizar interrupções.

4. Atualização da localização (*LocationUpdateTec*): módulos que tratam da atualização da localização do computador móvel na rede fixa. A atualização da localização depende de como

esta informação é mantida na rede, em um ou mais elementos de rede e isto, por sua vez, depende do protocolo de mobilidade empregado. No Mobile IP Hierárquico, por exemplo, a atualização de localização é feita através do envio de uma notificação ao GFA (GFAUpdate), que é o elemento que mantém a informação de localização do computador móvel (GFA). Em procolos como o Cellular IP e HAWAII, em que a informação de localização é mantida de forma distribuída nos nós da rede, a mensagem de atualização deve passar por todos os nós que mantêm a informação de localização do computador móvel (*RouteCacheUpdate*) no caminho da nova estação base em direção ao *gateway* a fim de gerar a nova rota para o encaminhamento de pacotes. A principal diferença entre essas duas estratégias está no fato de que quando a localização do computador móvel é mantida de forma centralizada, como no primeiro caso, a mensagem de atualização inevitavelmente deve chegar até o *gateway* do domínio para que então os pacotes de dados possam ser encaminhados para a nova estação base. Já no segundo caso, em que a informação é mantida nos nós da rede, a mensagem de atualização deve chegar apenas até o nó comum entre o antigo e o novo caminho (e que está mais próximo das estações base), a partir do qual os pacotes começam a ser desviados para a nova estação base. No caso de um protocolo de mobilidade baseado em *multicast* (MulticastGroupUpdate), a atualização deve ser feita no grupo *multicast* cujos membros são estações base vizinhas à estação base corrente). Essa atualização se baseia no envio de mensagens *Join*/*Leave* às estações base vizinhas. Uma atualização eficiente da rede permite aos pacotes de dados serem direcionados mais rapidamente à nova localização do computador móvel, evitando-se perdas e diminuindo a latência.

5. Transmissão de pacotes (*TransmissionTec*): módulos cuja função é controlar a forma de transmissão de pacotes e roteamento de pacotes para um computador móvel em uma rede. Protocolos de mobilidade empregam técnicas como encapsulamento+tunelamento (Tunneling), Multicast e Unicast (neste caso, o encaminhamento *hop-by-hop* baseado em nós IP modificados ou em nós específicos é empregado para tratar mobilidade). Cada forma de transmissão possui vantagens e desvantagens, por exemplo: tunelamento, não requer modificação nos roteadores IP porém dificulta a provisão de QoS e requer um elemento centralizado que mantém a localização de computadores móveis; Multicast, que possibilita a redução de perda de pacotes e latência mas pode acarretar em considerável carga na rede e utilização de recursos e Unicast, que não requer um elemento central para o gerenciamento de localização mas é preciso implementar nós específicos.

6. Otimização do *handover* (*HandoverOptmzTec*): esta categoria contém módulos que implementam estratégias para a antecipação do procedimento de *handover*, como por exemplo,

a configuração prévia do caminho de roteamento para a nova estação base (PathPreConfig), a seleção e notificação de uma ou mais estações base candidatas para algum elemento que mantém as informações sobre a localização do computador móvel a fim de proceder uma replicação de pacotes para as mesmas (PacketReplication). Além disso, com a informação sobre a nova estação base também é possível realizar transferência de contexto (Context-Transfer) e pré-alocação de recursos (PreRscrAllocation). Essas técnicas dependem de um evento da camada de enlace (*handover trigger*) que possui meios para prever as estações base candidatas através do monitoramento constante das potências de sinais emitidas pelas mesmas. O uso dessas técnicas permite reduzir a latência do processo de detecção de mobilidade.

7. Otimização do fluxo de dados (*DataFlowOptmzTec*): módulos nesta categoria empregam técnicas para tratar os pacotes de dados sendo enviados ao computador móvel durante o período de transição de uma estação base a outra. Por exemplo, Buffer, para o armazenamento temporário de pacotes nas estações base, que pode ser combinado com Forward para o redirecionamento desses pacotes da antiga para a nova estação base durante o *handover*. A fim de reduzir a perda de pacotes e a latência na entrega, a técnica de replicação de pacotes para uma ou mais estações base (NewBSReplication, NeighborBSReplication) pode ser empregada. Em particular, essas técnicas dependem da pré-seleção da nova estação base e da pré-configuração do caminho.

8. Algoritmos de garantia de entrega (*DeliveryAlgorithm*): nesta categoria os módulos implementam técnicas ou algoritmos para tratar o fluxo de pacotes durante a transmissão e prover QoS. Como exemplos, podemos citar: políticas de ordenação de pacotes (Ordering) e detecção de duplicações (Duplication), retransmissão de pacotes (Retransmission), confirmação de recebimento pelo computador móvel (Ack), algoritmos de garantia de entrega (Reliable, SemiReliable, ExactlyOnce, etc).

9. Infra-estrutura de mobilidade (*MobilityManagement*): módulos nesta categoria implementam alguma estrutura para dar suporte à mobilidade de acordo com o tipo de protocolo, como por exemplo, no caso hierárquico (Mobile IP Hierárquico), emprega um banco de dados de localizações GWLocationDB e agentes de mobilidade (MobilityAgent) para manter atualizada a localização do computador móvel. No caso de protocolos baseados em roteamento específico (Cellular IP, HAWAII), são utilizados nós e *caches* específicos para tratar mobilidade (SpecRouterCache). E, no caso de protocolos baseados em *multicast*, são necessários mecanismos para o gerenciamento do grupo *multicast*.

As categorias 1, 2 e 3 de módulos canônicos citadas acima consistem de técnicas empregadas

para tratar o *handover* na camada de enlace, por exemplo, em redes celulares. Porém, alguns protocolos de micro-mobilidade supõem a existência de determinadas técnicas, por exemplo, evento *handover trigger* da camada de enlace para a antecipação do *handover*, ou a possibilidade de conexão com mais de uma estação base simultaneamente para o caso de protocolos do tipo *soft handover*.

As categorias Transmissão de Pacotes, Atualização da Localização e Infra-estrutura de Mobilidade contêm mecanismos independentes de protocolos de *handover* mas que são utilizados para dar suporte à simulação, a fim de refletir as características de protocolos de micro-mobilidade.

5.2.2 Componentes do HOPF e Controller

O HOPF gerencia a execução de um protocolo de *handover* durante a sua simulação. Os componentes do HOPF para tratar as tarefas básicas do *handover* e oferecer suporte a *handover* suave são: Componente de Detecção de Mobilidade e Início do *Handover* (MobDetectionInit), Componente de Atualização da Rede (NetworkUpdate), Componente de Otimização do Fluxo de Pacotes de Dados (DataFlowOptmz) e Componente de *Pré-handover* (PreHandover). Além desses componentes, o HOPF também possui um elemento, que chamamos de Controller, que é o responsável pelo gerenciamento de eventos de envio e recebimento de mensagens entre os elementos de rede (e.g., *gateway*, estações base, roteadores e computador móvel), possibilitando a sua comunicação. O Controller faz a distribuição dos eventos de recebimento de mensagens aos respectivos módulos canônicos tratadores desses eventos.

Na Figura 5.3 temos uma ilustração da estrutura do HOPF e seus componentes em um elemento de rede. Eventos envolvidos na comunicação interna entre módulos canônicos, isto é, em um mesmo elemento de rede são chamados de *eventos internos* enquanto que os eventos na comunicação entre módulos canônicos em distintos elementos de rede, são chamados de *eventos externos*.

Cada componente do HOPF pode conter um ou mais módulos canônicos, de acordo com as tarefas associadas ao mesmo e esses módulos dependem do tipo de protocolo de *handover* suave e das técnicas selecionadas. Exemplos de módulos canônicos para as diferentes tarefas de *handover* para esses componentes foram apresentados na Seção 5.2.1.

O conjunto de componentes a ser instanciado para uma execução depende do protocolo de *handover* suave (e dos módulos canônicos selecionados) assim como de cada elemento de rede específico. Ou seja, para um mesmo protocolo de *handover* suave, a combinação de componentes em um elemento de rede é distinta, dependendo das funcionalidades providas por este elemento. Por exemplo, em um dado protocolo, uma estação base pode ter que instanciar todos os componentes na Figura 5.3, enquanto que um roteador pode precisar utilizar apenas o componente

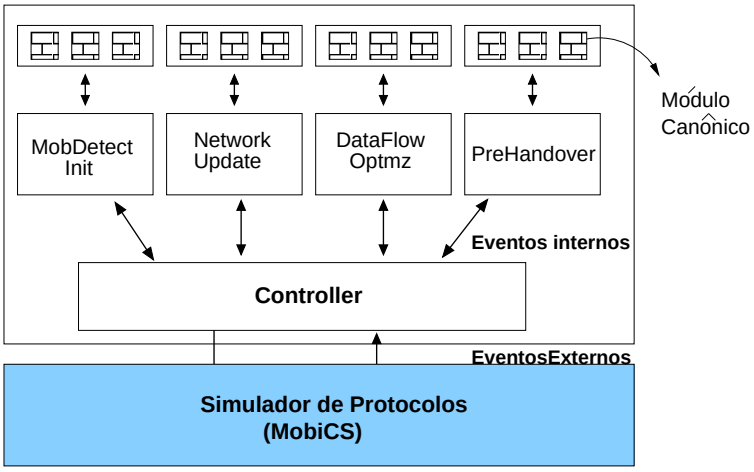


Figura 5.3: Estrutura do HOPF

NetworkUpdate.

A seguir, apresentamos os componentes do HOPF e suas funcionalidades em maiores detalhes.

Componente de Detecção de Mobilidade e Início do Handover (MobDetectionInit)

Este componente é usado para tratar as seguintes tarefas: detecção de mobilidade, que consiste em determinar o momento em que se verifica a necessidade de um *handover* devido a uma mudança de localização; início do *handover*, através de uma sinalização (evento) para algum elemento de rede; alocação e liberação de recursos (por exemplo, alocação e liberação de canais, *buffer*, etc.) e a transferência da comunicação da antiga para a nova estação base. Esse componente é instanciado particularmente no computador móvel (quando este é o responsável pela detecção de mobilidade) e estações base que interagem entre si para tratar a transferência da comunicação.

Na Figura 5.4 apresentamos um diagrama de classes simplificado para este componente que pode ser instanciado no computador móvel (MhMobDetectionInit) e nas estações base (MssMobDetectionInit). No computador móvel este componente faz a detecção de mobilidade por exemplo, na camada de enlace, através de freqüentes medições dos sinais emitidos pelas estações base (*beacons*) ou, na camade de rede, por meio de mensagens *Agent Advertisements* difundidas pelos FA's, como é realizado no Mobile IP.

Ao identificar a necessidade de iniciar um *handover*, é disparado um evento HOTriggerEvent, que pode ser, por exemplo, o envio de uma mensagem Greet à nova estação base. O componente MssMobDetectionInit trata esse evento e usa algumas técnicas para a alocação de canais

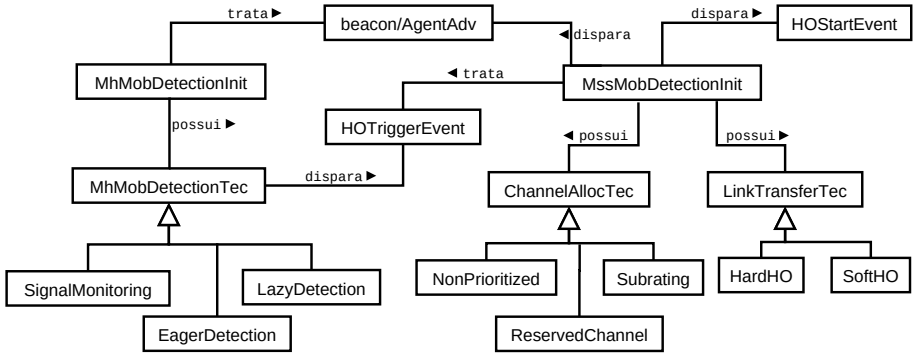


Figura 5.4: Componente MobDetectionInit

(ChannelAllocTec) e a transferência da conexão (LinkTransferTec). Também dispara o evento HOStartEvent para a notificação aos outros elementos de rede e é tratado pelo componente NetworkUpdate, conforme apresentamos a seguir.

Componente de Atualização da Rede (NetworkUpdate)

Na ocorrência de um *handover*, os elementos de rede devem ser atualizados de modo que o computador móvel continue recebendo pacotes em sua nova localização. Para isso, o caminho de roteamento de pacotes deve ser modificado para alcançar a nova estação base. Além disso, a nova localização do computador móvel deve ser notificada aos elementos de rede que mantêm a informação de localização do mesmo, por exemplo, o *gateway*.

Esse componente deve ser instanciado nas estações base, roteadores e *gateway*, conforme apresentamos na Figura 5.5. Quando em uma estação base, esse componente trata o evento de início de *handover* (HOStartEvent) emitido pelo componente MobDetectionInit. De acordo com o protocolo de *handover* (i.e., os módulos canônicos selecionados), esse componente dispara um evento para a atualização da localização (LocUpdateEvent) e/ou para a atualização do caminho de roteamento de pacotes (PathUpdateEvent). Por exemplo, em protocolos como o Mobile IP, é enviada uma mensagem *Registration Request* ou *Binding Update* em direção ao *gateway* para o HA e não é necessário o evento de atualização de caminho uma vez que os pacotes são enviados por tunelamento dentro do domínio. No caso do Cellular IP, por outro lado, a atualização de localização é feita juntamente com a atualização de caminho, uma vez que a informação de localização é mantida nos *caches* de roteamento e, portanto, é necessário apenas o envio de uma mensagem *PathUpdate* em direção ao *gateway*.

Esses eventos (LocUpdateEvent e PathUpdateEvent) são tratados nos roteadores e no *gateway*

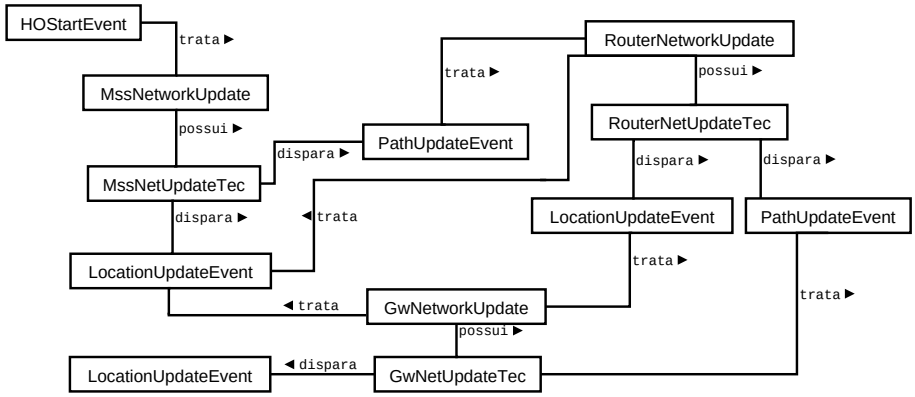


Figura 5.5: Componente NetworkUpdate

pelos componentes RouterNetworkUpdate e GwNetworkUpdate, respectivamente. Nos roteadores, dependendo do protocolo, por exemplo, no Mobile IP, a mensagem de atualização é passada diretamente para o próximo elemento de rede, enquanto que no caso do Cellular IP, é feita a atualização do *cache* de roteamento e então a mensagem é repassada ao próximo elemento de rede. No *gateway*, no caso do Mobile IP, é feita a atualização do registro de localização do computador móvel e a mensagem é repassada ao HA, que pode estar localizado no próprio *gateway* e, no caso do Cellular IP, o *cache* de roteamento é atualizado.

Componente de Otimização do Fluxo de Dados (DataFlowOptmz)

Este componente emprega módulos que implementam técnicas para controlar o fluxo de pacotes durante o *handover*, com o objetivo de minimizar a perda de pacotes e atrasos na entrega dos mesmos. Diversas técnicas podem ser empregadas, conforme mencionamos na Seção 5.2.1.

Na Figura 5.6 apresentamos uma ilustração desse componente que pode ser instanciado em todos os elementos de rede, dependendo das técnicas de otimização selecionadas. Essas técnicas de otimização se concentram em duas categorias: DataFlowOptmzTec e DeliveryAlgorithm. Em uma estação base (MssDataFlowOptmz), diversos algoritmos de garantia de entrega e de tratamento de duplicações, ordenação, podem ser implementados como módulos canônicos e estes podem interagir com os módulos respectivos em componentes de outros elementos de rede (e.g. GwDataFlowOptmz). Em particular, para reduzir a perda de pacotes, a técnica de Buffer pode ser empregada nas estações base, cuja principal funcionalidade é armazenar pacotes provenientes da rede e redirecioná-los para a nova localização quando há a ocorrência de um evento que indica a migração do computador móvel.

Nos roteadores, em particular, no *crossover router*, no caso dos protocolos que mantêm a

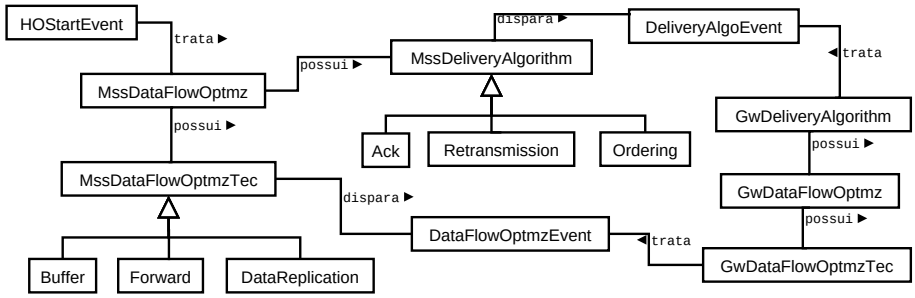


Figura 5.6: Componente DataFlowOptmz

informação de localização do computador móvel nesses elementos (e.g., Cellular IP, Hawaii), a principal técnica de otimização que pode ser empregada é a de replicação de pacotes para a antiga e nova estações base durante o conforme apresentamos na próxima seção.

Componente de Pré-handover (PreHandover)

O componente de *Pré-handover* implementa algumas tarefas para a otimização do desempenho do *handover*, executando antecipadamente algumas ações como, por exemplo: pré-notificação da nova localização do computador móvel, pré-alocação de recursos em uma ou mais estações base, pré-configuração do caminho de roteamento de pacotes até uma ou mais estações base, etc.

Esse componente pode ser implementado no computador móvel, estações base e *gateway* (MhPreHO, MssPreHO e GwPreHO, respectivamente), Figura 5.7. No computador móvel, a tarefa desse componente é notificar a futura estação base para o início do *Pré-handover* através de um evento PreHOStart. Nas estações base, pode ser feita uma pré-alocação de recursos para a futura conexão e o evento é repassado para a rede em direção ao *gateway*. Em protocolos em tunelamento, como o Mobile IP, esse evento é tratado no *gateway*, que faz a atualização da localização e começa a replicar pacotes para a antiga e nova estações base. Em protocolos baseados em roteamento específico, como Cellular IP, esse evento é tratado pelo componente DataFlowOptmz no *crossover router* que faz a replicação de pacotes para as estações base.

Controller

O Controller faz o gerenciamento da execução de um protocolo de *handover* em cada elemento de rede, tratando eventos de envio e recebimento de mensagens. Na Figura 5.8 temos uma visão geral do Controller em cada elemento de rede. Cada Controller está associado a um conjunto de componentes específico que contém os módulos canônicos que efetivamente tratam os eventos.

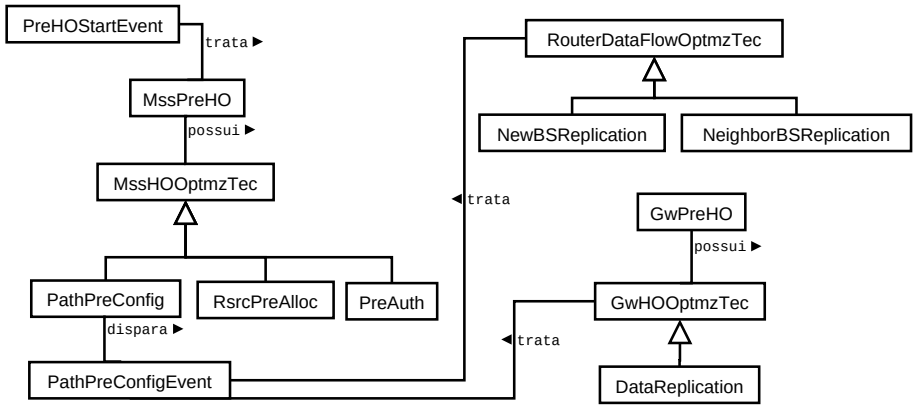


Figura 5.7: Componente PreHandover

Na ocorrência de eventos de recebimento de mensagem, o Controller faz a distribuição desses eventos aos respectivos módulos canônicos tratadores em algum componente. Para isso, o Controller mantém uma lista de associações <*tipo_de_ mensagem*, *seqüência_de_módulos_canônicos*> e, para cada mensagem recebida, o Controller repassa essa mensagem para cada módulo canônico na seqüência. Essa lista de associações define o fluxo de execução de um protocolo de *handover*.

O fluxo de execução de um protocolo de *handover* depende dos módulos canônicos selecionados para tratar as tarefas de *handover* e para as otimizações a serem empregadas. Dessa forma, o fluxo de execução é definido após a seleção de módulos canônicos e é especificado como um arquivo de configuração, o qual é passado como parâmetro ao HOPF e se mantém fixo durante a simulação (Seção 6.4).

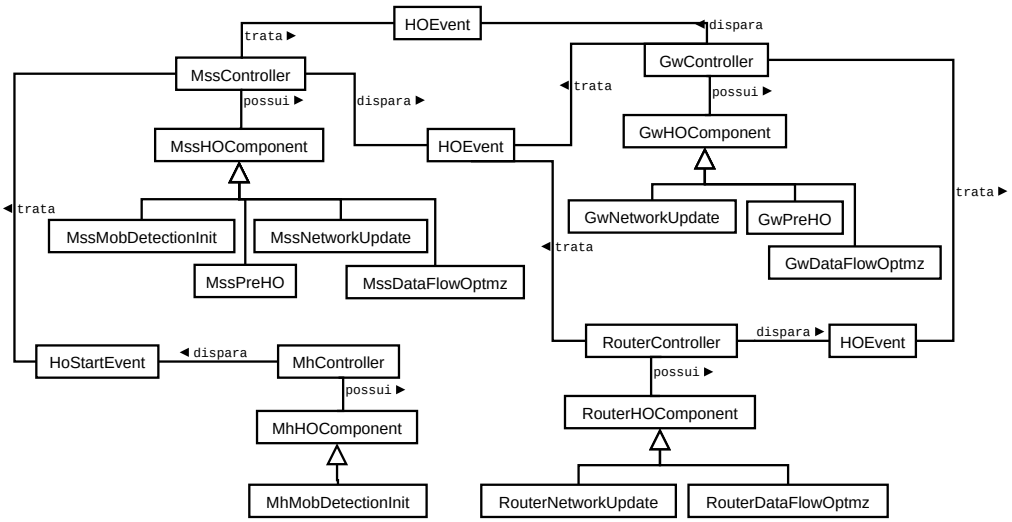


Figura 5.8: Controller

Uma vez que podemos ter mais de um módulo canônico para tratar uma mesma tarefa, não é desejável que a troca de um módulo cause mudanças no Controller e nos componentes, após a uma nova seleção de módulos e instanciação do protocolo. Para tal, optamos por um projeto que permita um total desacoplamento entre o Controller e os componentes dos módulos canônicos. A fim de se evitar mudanças no Controller e componentes de protocolo devido à uma troca de um módulo canônico, a nossa abordagem foi empregar o padrão *Chain of Responsability* [32] que possibilita a invocação uniforme de objetos na ocorrência de um evento.

O padrão *Chain of Responsability* oferece flexibilidade na distribuição de responsabilidades entre um conjunto de objetos. A idéia deste padrão é desacoplar os remetentes de uma requisição de seus receptores, permitindo, assim, que múltiplos objetos tratem uma requisição. A requisição é passada através de uma corrente de objetos permitindo que a mesma seja tratada por mais de um objeto.

A principal vantagem de se aplicar esse padrão em nosso *framework* é que o mesmo permite que uma mensagem possa ser tratada por um ou mais módulos canônicos, sendo necessário apenas que a mensagem seja passada pela seqüência de módulos canônicos associada (que foi definida no fluxo de execução). Dessa forma, as mensagens são tratadas de maneira uniforme no Controller não requerendo modificações quando novos tipos de mensagens são utilizadas com a substituição ou inclusão de novos módulos canônicos (em uma nova instanciação do protocolo de *handover*). Porém, apenas o fluxo de execução precisa ser alterado de acordo quando há a inclusão/substituição de módulos canônicos.

Na Figura 5.9-(1) apresentamos um exemplo de implementação do padrão *Chain of Responsability* (EventHandlerInterface) e alguns módulos do componente NetworkUpdate que devem implementar essa interface. Na Figura 5.9-(2) apresentamos uma cadeia de objetos gerada pela ocorrência de um evento a partir do Controller e sendo passado ao componente NetworkUpdate e seus respectivos módulos canônicos que devem tratar o evento (LocationUpdate e PathUpdate).

5.2.3 Processo de Seleção de Módulos Canônicos

Para a seleção de módulos canônicos, poderiam ser considerados três conjuntos de informações: requisitos de QoS da aplicação, perfil de mobilidade do usuário e características da rede. A seguir, apresentamos exemplos de dados que julgamos relevantes para cada tipo de informação.

- Requisitos de *QoS*: valores numéricos ou boleanos de alguns requisitos de QoS da aplicação, por exemplo, máximo atraso aceitável, máxima variação do atraso aceitável, percentual aceitável de perda de pacotes, percentual aceitável de duplicações, se requer ou não ordenação, etc.

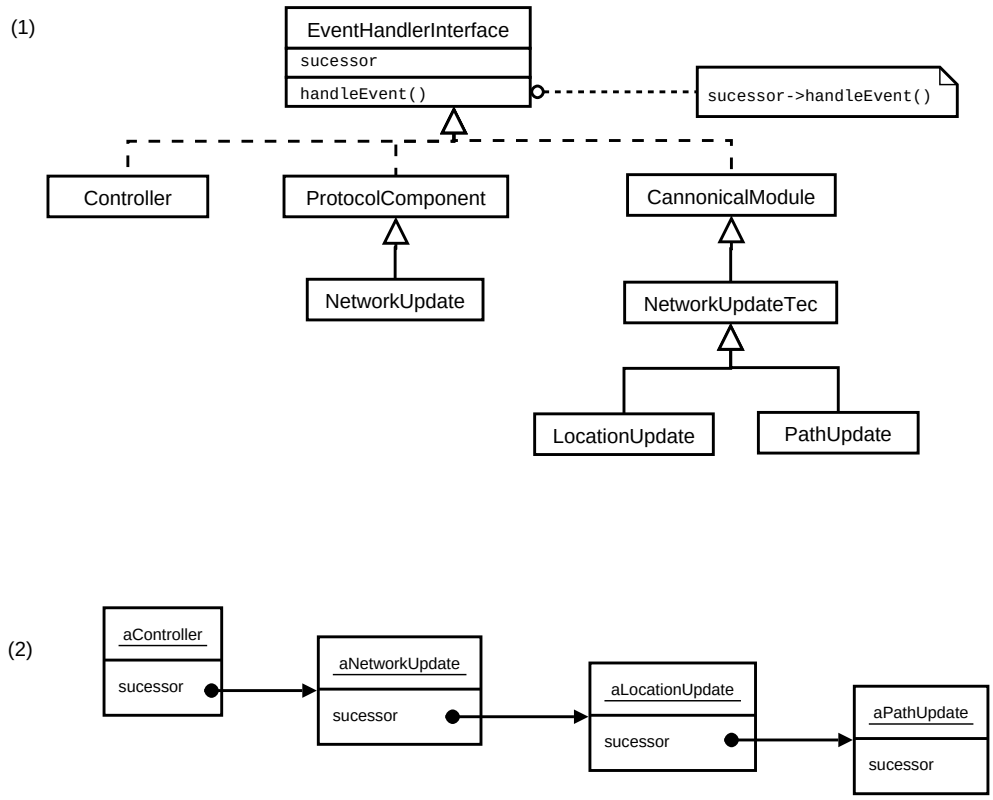


Figura 5.9: (1) Interface EventHandler e (2) exemplo de uma cadeia de objetos

- Perfil de mobilidade do usuário: especifica algumas características do padrão de mobilidade de um usuário móvel, tal como velocidade média de movimentação, probabilidade de migração para regiões cobertas por determinadas estações base, etc.
- Parâmetros de configuração da rede: informações sobre a estrutura da rede fixa/móvel com relação aos elementos de rede (nós, pontos de acessos, quantidade de células adjacentes/vizinhas, regiões de intersecção de células).

Algumas Regras para a Seleção de Módulos Canônicos

Muitos fatores podem influenciar no desempenho de protocolos de *handover*, desde as técnicas empregadas para tratar as tarefas básicas do *handover* (por exemplo, atualização de localização) até a inclusão de determinadas técnicas de otimização, o grau de mobilidade do usuário móvel e a topologia da rede. Aqui apresentaremos algumas regras empíricas para a seleção de módulos canônicos obtidas a partir dos resultados de simulação de diferentes combinações de módulos canônicos e com diferentes parâmetros de simulação (maiores detalhes sobre as simulações e uma lista mais completa dessas regras podem ser encontradas no Capítulo 7).

- Para aplicações sensíveis à perda de pacotes a técnica de *multicast* pode oferecer uma menor taxa de perdas, independentemente da taxa de mobilidade. Porém, essa técnica acarreta em um elevado número de duplicação de pacotes, assim como uma sobrecarga muito alta em termos de mensagens de controle, que são proporcionais à taxa de mobilidade.

- O uso de *buffer* também permite a redução da perda de pacotes, porém, não é aconselhável a sua utilização quando a aplicação possui requisitos muito fortes com relação ao atraso.

- A técnica de antecipação do *handover* (*Pré-handover* ou Preho) permite reduzir a perda de pacotes e atraso com relação à técnica de *bufferização*. Quando os requisitos de perda de pacotes e atraso são muito fortes, uma possibilidade é empregar uma combinação de *buffer* + *Pre-handover* que se mostrou bastante eficiente com relação às outras otimizações.

- Se o principal requisito da aplicação é ter a menor sobrecarga de mensagens de controle, então protocolos que empregam mecanismos para atualização na rede como o Hawaii ou Cellular IP se mostraram mais adequados.

5.2.4 Especificação dos Módulos Canônicos Implementados

Nesta seção, apresentamos uma especificação de alguns módulos canônicos implementados, de acordo com os seguintes aspectos: funcionalidade básica do módulo, elementos de rede aonde devem ser instanciados, os parâmetros de entrada, as formas de interação (dependência e conflitos) com outros módulos e alguns exemplos de protocolos de micro-mobilidade que poderiam ser implementados empregando o módulo.

Tendo como objetivo o teste do HOPF para a implementação, composição e simulação de protocolos de *handover* suave para micro-mobilidade, implementamos as seguintes categorias de módulos canônicos:

- Infra-estrutura de suporte a mobilidade + roteamento de pacotes: esta categoria de módulos permite implementar distintas formas de suporte à mobilidade e transmissão de pacotes, sobre o qual é possível implementar um modo de gerenciamento de localização e esquemas de atualização de caminhos e localização. Dependendo de como e onde a informação de localização de um computador móvel é mantida, o processo de atualização possui um custo associado e este custo reflete diretamente no desempenho do procedimento de *handover*. Módulos implementados nesta categoria: *unicast* centralizado (UcastC), *unicast* distribuído (UcastD) e *multicast* (Mcast);

- Atualização: a forma de atualização de localização de um computador móvel na rede após uma migração assim como a reconfiguração do caminho de roteamento de pacotes estão

diretamente relacionadas com o tipo de infra-estrutura de mobilidade+rotemento selecionado. Porém, em uma mesma infra-estrutura de mobilidade é possível implementar distintos esquemas de atualização, dependendo do protocolo de *handover*. Os seguintes módulos foram implentados: UpdateUcastC, com especializações baseadas no Mobile IP e Mobile IP com otimização de rotas; UpdateUcastD, com especializações para os esquemas de *handover*: Cellular IP *hard handover*, Hawaii MSF (*Multiple Stream Forwaring*) e Hawaii MNF (*Multicast Non-Forwarding*); UpdateMcast, com especialização para o M&M;

- Otimização de handover: a fim de melhorar o desempenho de um protocolo de *handover*, algumas otimizações podem ser empregadas de modo a reduzir perdas de pacotes, atrasos e duplicações. Implementamos os seguintes módulos canônicos: Buffer, que permite o armazenamento de pacotes nas estações base; Forward, que permite o re-direcionamento de pacotes da antiga para a nova estação base; Retransmission, para que pacotes possam ser retransmitidos ao computador móvel; Ack, que faz com que o computador móvel confirme todo pacote de dado recebido a fim, evitando-se duplicações; PreHO, que permite uma antecipação do procedimento de *handover*, pré-configurando o caminho de rotemento de pacotes e, opcionalmente, replicando pacotes para a antiga e nova estações base por um período de tempo (Replication).

A seguir, apresentamos cada um desses módulos canônicos em maiores detalhes.

1. **Módulo UcastC**

   Categoria: Infra-estrutura de mobilidade

   Funcionalidade: este módulo implementa um modo centralizado para a manuntenção da informação de localização de computadores móveis em um elemento de rede específico (por exemplo, o *gateway*). Para o encaminhamento de pacotes é empregada uma abordagem semelhante ao tunelamento do Mobile IP: pacotes para um computador móvel são encapsulados em um novo pacote cujo destinatário é a atual estação base responsável pelo mesmo.

   Elementos de rede: esse módulo deve ser implementado no *gateway*, estações base e roteadores.

   Parâmetros: tabela (*cache*) de roteamento, uma estrutura que associa a cada elemento de rede (estações base, *gateway*, roteadores) um elemento que é o próximo nó em direção a esse elemento. Este *cache* é configurado no momento da instanciação e configuração do protocolo e da rede simulada.

Dependências: módulo da categoria Atualização da rede UpdateUcastC.

Conflitos: módulos na mesma categoria e módulos de atualização diferentes de UcastC.

Exemplos de protocolos: Mobile IP

2. **Módulo UcastD**

Categoria: Infra-estrutura de mobilidade

Funcionalidade: implementa um modo distribuído para manter a informação de localização de computadores móveis em elementos de rede como o *gateway*, roteadores e estações base. O encaminhamento de pacotes se baseia nessa informação e é feito de maneira *hop-by-hop*.

Elementos de rede: esse módulo deve ser implementado no *gateway*, estações base e roteadores.

Parâmetros: tabela de roteamento específico, que além das funções do cache comum, permite a inserção/remoção de registros de computadores móveis.

Dependências: módulo da categoria de Atualização da rede UpdateUcastD.

Conflitos: módulos na mesma categoria e módulo de atualização UcastC.

Exemplos de protocolos: Cellular IP e Hawaii.

3. **Módulo Mcast**

Categoria: Infra-estrutura de mobilidade

Funcionalidade: implementa um modo distribuído para manter a informação de localização e a transmissão de pacotes é baseada em *multicast*. Um grupo *multicast* é associado a cada computador móvel e é formado pela atual estação base responsável além de todos as suas estações base vizinhas. Durante toda a execução, pacotes de dados são replicados para todos os elementos no grupo *multicast*.

Elementos de rede: esse módulo deve ser implementado no *gateway*, estações base e roteadores.

Parâmetros: tabela de roteamento; para cada estação base, o seu respectivo grupo de estações base vizinhas.

Dependências: módulo da categoria Atualização de rede UpdateMcast.

Conflitos: módulos na mesma categoria e módulos da categoria Atualização da rede diferentes de UpdateMcast.

Exemplos de protocolos: M&M.

4. **Módulo UpdateUcastC**

   Categoria: Atualização da rede

   Funcionalidade: permite a atualização da localização de um computador móvel quando a informação de localização é mantida em um elemento de rede específico (por exemplo, no *gateway*). Essa atualização consiste no envio de uma notificação pela nova estação base ao *gateway*, indicando a nova localização do computador móvel, de modo que o registro de localização seja atualizado.

   Elementos de rede: esse módulo deve ser implementado no *gateway* e estações base.

   Parâmetros: endereço da nova estação base, identificador do computador móvel.

   Dependências: módulo de infra-estrutura de mobilidade UcastC.

   Conflitos: módulos na mesma categoria.

   Exemplos de protocolos: Mobile IP e Mobile IP Route Optimisation.

5. **Módulo UpdateUcastD**

   Categoria: Atualização da rede

   Funcionalidade: permite a atualização da localização de um computador móvel quando a informação de localização sobre o mesmo é mantida de forma distribuída em vários elementos de rede (por exemplo, nos roteadores). O esquema de atualização em si se baseia no protocolo de *handover* selecionado, porém, uma operação básica é a atualização dos caches de roteamento em cada elemento de rede que contém uma entrada para um computador móvel.

   Elementos de rede: esse módulo deve ser implementado no *gateway*, roteadores e estações base.

   Parâmetros: endereço da nova estação base e identificador do computador móvel.

   Dependências: módulo de infra-estrutura de mobilidade UcastD.

   Conflitos: módulos na mesma categoria.

   Exemplos de protocolos: Cellular IP (*hard* e *soft handover*), Hawaii (nos quatro esquemas de atualização propostos: MSF, MNF, USF, UNF).

6. **Módulo UpdateMcast**

Categoria: Atualização na rede

Funcionalidade: permite a atualização da localização de um computador móvel quando o esquema de transmissão de pacotes e gerenciamento de mobilidade se baseia em *multicast*. A atualização do grupo *multicast* se baseia em operações join/leave no grupo *multicast*.

Elementos de rede: esse módulo deve ser implementado nas estações base.

Parâmetros: identificador do computador móvel.

Dependências: módulo de infra-estrutura de mobilidade Mcast.

Conflitos: módulos na mesma categoria.

Exemplos de protocolos: M&M.

7. **Módulo Buffer/Forward**

Categoria: Otimização do fluxo de dados

Funcionalidade: permite o armazenamento de pacotes em estações base em um *buffer* e, opcionalmente, o re-direcionamento desses pacotes da antiga para a nova estação base.

Elementos de rede: esse módulo deve ser implementado nas estações base.

Parâmetros: tamanho do *buffer*; um valor booleano que indica se requer ou não o re-direcionamento de pacotes; um algoritmo/política para tratar *buffer overflow*.

Dependências: o uso opcional do módulo Ack permite remover do *buffer* os pacotes que foram confirmados pelo computador móvel.

Conflitos: não há.

Exemplos de protocolos: qualquer protocolo de *handover* pode empregar este módulo.

8. **Módulo Ack**

Categoria: Algoritmos de garantia de entrega

Funcionalidade: permite que um computador móvel confirme o recebimento de pacotes.

Elementos de rede: esse módulo deve ser implementado no computador móvel e nas estações base.

Parâmetros: não há.

Dependências: Opcionalmente, pode ser combinado com o módulo Buffer ou Mcast, permite reduzir duplicações.

Conflitos: não há.

Exemplos de protocolos: qualquer protocolo pode empregar este módulo.

9. **Módulo Retransmission**

Categoria: Algoritmos de garantia de entrega

Funcionalidade: permite que o computador móvel requisite retransmissões de pacotes não recebidos de acordo com o *timestamp* da mensagem em intervalos de tempos. Este módulo é empregado para possibilitar a redução de perdas de pacotes.

Elementos de rede: esse módulo deve ser implementado no computador móvel, estações base, *gateway* e nó fonte.

Parâmetros: intervalo de retransmissão de pacotes.

Dependências: opcionalmente, combinado com o módulo Buffer, possivelmente grande parte das retransmissões podem ser tratadas pelas estações base, somente quando o pacote não está no *buffer*, a requisição é enviada ao nó fonte.

Conflitos: não há.

Exemplos de protocolos: qualquer protocolo pode empregar este módulo.

10. **Módulo PreHO/Replication**

Categoria: Otimização do Handover

Funcionalidade: este módulo permite uma pré-configuração de caminho de roteamento de pacotes do *gateway* até a nova estação base e a replicação de pacotes para esta nova estação durante o *handover*. Essas otimizações permitem reduzir a latência e perdas de pacotes causadas pelo *handover*.

Elementos de rede: esse módulo deve ser implementado no computador móvel, estações base, *gateway* e, no caso de gerenciamento de mobilidade distribuído, nos roteadores também.

Parâmetros: o endereço da nova estação base.

Dependências: módulos que tratam as tarefas de atualização de caminho e replicação de pacotes.

Conflitos: módulos relacionados com transmissão *multicast*, pois estes implicitamente empregam uma forma de antecipação de *handover*, através da replicação de pacotes para todas as estações base vizinhas.

Exemplos de protocolos: qualquer protocolo como Mobile IP, Cellular IP, HAWAII, etc. pode empregar este módulo, com exceção de protocolos baseados em *multicast*.

Capítulo 6

Implementação

O HOPF foi implementado em Java e utiliza o Simulador de Protocolos Distribuídos MobiCS [16, 56] para simular a topologia de rede, características dos enlaces com e sem fio e mobilidade de computadores. A principal razão para a escolha deste simulador se deve, basicamente, à sua flexibilidade e facilidade de uso que permitem uma rápida prototipação e implementação de protocolos para ambientes móveis.

6.1 MobiCS

O MobiCS (Mobile Computing Simulator) [16, 56] é um ambiente para prototipação, teste e simulação de protocolos para computação móvel. O MobiCS possui dois modos de simulação: o determinístico e estocástico. O modo determinístico é utilizado para avaliar protocolos em situações específicas, a simulação se baseia em um *script* que expressa o comportamento dinâmico do ambiente de computação móvel, desde a movimentação dos computadores móveis, o instante de envio/recebimento de mensagens até a ordem global de ocorrência dos eventos. No modo estocástico, o MobiCS executa uma simulação exaustiva nos protocolos distribuídos, com o objetivo de avaliar o desempenho do protocolo em um cenário aleatório e mais realístico. Com esse modo de simulação é possível também observar o comportamento do protocolo em cenários maiores e exaustivos, cuja descrição é impraticável através de *scripts* determinísticos. Em nossas simulações de protocolos de *handover*, utilizamos o modo estocástico de simulação do MobiCS.

O MobiCS se baseia em três tipos de abstrações para a especificação de modelos de simulação: gerador de eventos, atraso de comunicação e mobilidade. Um gerador de eventos indica como os eventos são gerados durante uma simulação. O atraso de comunicação especifica o comportamento dos canais de comunicação como uma função do tempo de envio de uma mensagem pelo canal. A mobilidade define abstrações sobre a localização e movimentação de computadores móveis.

O modelo de programação de protocolos adotado pelo MobiCS é baseado no conceito de micro-protocolos que são módulos que implementam uma funcionalidade bem definida. Um protocolo é uma classe que implementa um conjunto de micro-protocolos e deve ser uma extensão da classe Protocol.

Para o nosso *framework*, os módulos canônicos correspondem aos micro-protocolos do MobiCS no sentido em que implementam uma técnica ou funcionalidade específica de um protocolo de *handover*. Porém, no nosso caso, os módulos canônicos podem ser combinados flexivelmente, permitindo a geração de protocolos de *handover* que implementam diferentes estratégias ou técnicas para tratar as tarefas do *handover*.

6.2 Componentes do HOPF

Nesta seção apresentamos alguns detalhes de implementação dos elementos do HOPF: mensagens, módulos canônicos e Controller.

6.2.1 Mensagens

As mensagens são abstrações para a comunicação entre módulos canônicos em um mesmo ou entre distintos elementos de rede. Uma mensagem é uma extensão da classe mobics.ppi.message.Message do MobiCS. No HOPF, o Controller faz o envio e recebimento de mensagens possibilitando a comunicação entre os elementos de rede e módulos canônicos. Uma vez que cada protocolo de *handover* ou otimização define um conjunto específico de mensagens de controle, e para evitar a implementação de tratadores específicos para cada tipo de mensagem no Controller para cada tipo de protocolo a ser simulado, o HOPF especifica um conjunto de mensagens padrões (ou genéricas) onde são definidos atributos que identificam cada uma dessas mensagens de controle específicas.

O atributo type indica o tipo da mensagem, que pode ser applData quando se trata de uma mensagem da aplicação ou hoCtrl quando se refere a uma mensagem de controle do protocolo de *handover*. No caso de mensagens de controle, há também o atributo ctrlTypeName que indica o tipo específico da mensagem de controle (Figura 6.1). Definimos dois tipos de mensagens padrões: DataPacket, que corresponde aos pacotes de dados da aplicação e que são gerados pelo nó fonte e CtrlPacket, que são as mensagens de controle de um protocolo de *handover* ou de uma técnica de otimização.

O uso de mensagens padrão facilita a implementação de protocolos uma vez que requer que as interfaces de cada elemento de rede especifiquem apenas esses dois tipos de mensagens (whenDataPacket e whenCtrlPacket). E, como conseqüência, o Controller em cada elemento de

rede precisa implementar apenas os tratadores dessas duas mensagens. Esses tratadores apenas fazem a invocação de módulos canônicos que devem tratar efetivamente as mensagens.

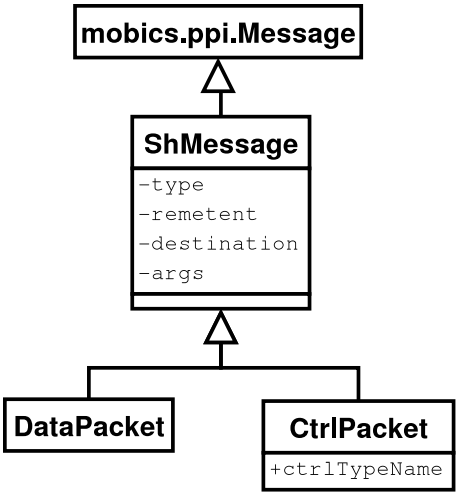


Figura 6.1: Mensagens padrão

6.2.2 Módulos Canônicos

Um módulo canônico implementa uma funcionalidade/técnica de otimização ou algoritmo para tratar uma determinada tarefa do *handover*. Corresponde a uma classe Java que estende a classe abstrata CanonicalModule e implementa o método handleEvent. O método handleEvent é a parte central de um módulo canônico, pois neste método é especificado o comportamento do mesmo através dos tratadores de mensagens que este módulo implementa.

6.2.3 Controller

O Controller é o responsável pela comunicação entre os módulos canônicos e pelo controle de execução, permitindo que seja alcançado o comportamento desejado de um protocolo de *handover*. O Controller trata eventos de recebimento de mensagens e invoca os tratadores de eventos de cada módulo canônico que deve tratar o evento.

O Controller foi implementado como uma extensão da classe mobics.ppi.protocol.Protocol, que é a classe onde são implementadas as funcionalidades de um protocolo de rede a ser simulado no MobiCS. Para o HOPF, as funcionalidades ou tarefas dos protocolos de *handover* estão implementadas nos módulos canônicos que compõem o protocolo. Em cada elemento de rede deve ser instanciado um elemento Controller correspondente.

Na Figura 6.2 temos um diagrama de classes para o Controller. Na ocorrência de um evento

de recebimento de mensagem, um dos dois tratadores de eventos é invocado: whenDataPacket ou whenCtrlPacket, de acordo com o tipo da mensagem.

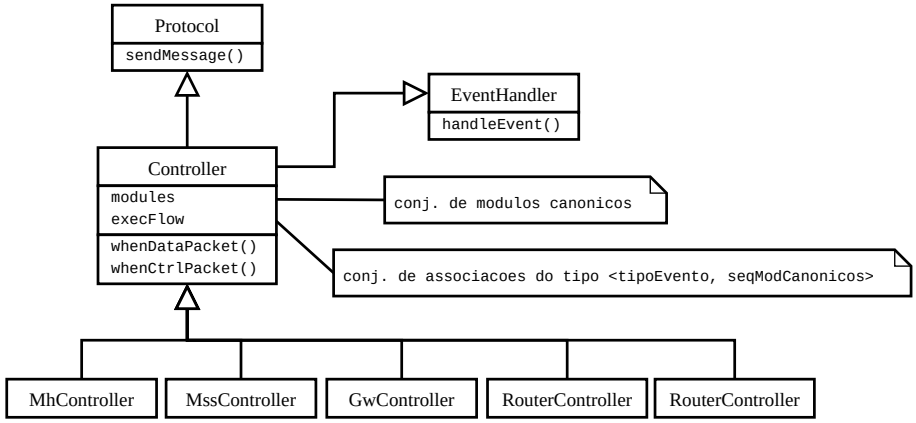


Figura 6.2: Diagrama de classes do Controller

O Controller tem como atributos um conjunto de módulos canônicos modules e uma estrutura que representa o fluxo de execução execFlow. O conjunto de módulos canônicos (modules) contém referências para os módulos que fazem parte do protocolo de *handover* sendo simulado, e que foram instanciados previamente.

O fluxo de execução é formado por seqüências de módulos canônicos (cadeias de nomes de módulos) associadas a um tipo de mensagem. A seqüência de módulos canônicos para cada mensagem corresponde aos módulos que devem tratar a mensagem quando esta é recebida pelo Controller. O execFlow deve conter associações $< tipoMensagem, seqModCanonicos >$ para todos os possíveis tipos de mensagens de controle requeridas pelo protocolo de *handover* e possíveis otimizações adicionais.

Na ocorrência de um evento, através do atributo tipo da mensagem de controle (ctrlTypeName) obtido da própria mensagem, é possível obter a seqüência (cadeia) de módulos canônicos tratadores da mensagem a partir de execFlow. Em seguida, de maneira seqüencial, para cada módulo canônico nesta cadeia, é invocado o tratador de eventos do módulo e é passada a mensagem como parâmetro.

6.3 Interface de Simulação e Testes do HOPF

A fim de facilitar os testes, instanciação e comparações dos vários esquemas de *handover* propostos na literatura, implementamos uma interface de simulação. Através dessa interface, é possível selecionar um esquema de *handover* e combiná-lo com uma ou mais técnicas de otimização. Até o momento, implementamos os esquemas de *handover* dos seguintes protocolos de

mobilidade: Mobile IP, Mobile IP Smooth Handover, Cellular IP, Hawaii MSF, Hawaii MNF e Multicast. Temos também as seguintes técnicas de otimização para tratar o fluxo de pacotes durante o *handover*: Buffer+Forward, com ou sem redirecionamento de pacotes, PreHO+Replication (para antecipar o *handover*), Ack (para o envio de confirmação de recebimento de pacotes) e Retrans (para a retransmissão de pacotes) (Figura 6.3).

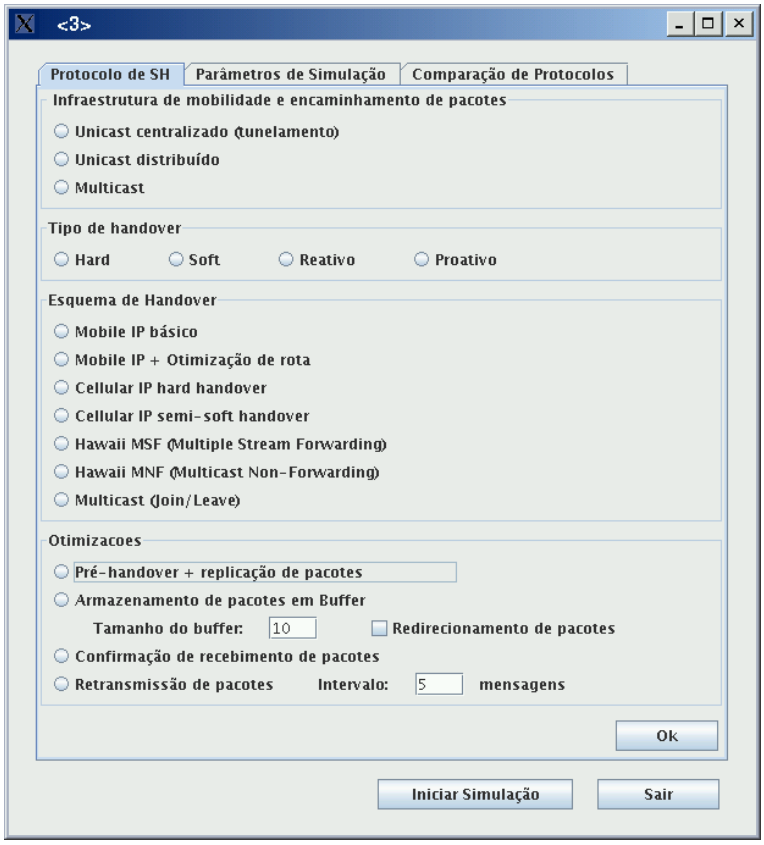


Figura 6.3: Interface para configuração/teste de um protocolo de *handover*

Através dessa interface também é possível configurar alguns parâmetros de simulação como valores mínimo e máximo para o atraso nos enlaces com e sem fio, taxas de mobilidade (baixa, média e alta), presença/ausência de regiões de intersecção e o número de simulações (execuções) do protocolo. (Figura 6.4).

Essa interface de simulação também permite a configuração e simulação de vários esquemas de *handover* simultaneamente combinados a uma ou mais técnicas de otimização distintas. Também oferece uma saída padronizada, com os resultados de diversos parâmetros como, por exemplo, pacotes perdidos, duplicados, fora de ordem, replicados, redirecionados, carga de mensagens de controle, atraso e variação do atraso, para cada um dos protocolos. Esses resultados incluem a média, desvio padrão, variância e valores máximo e mínimo obtidos nas simulações permitindo,

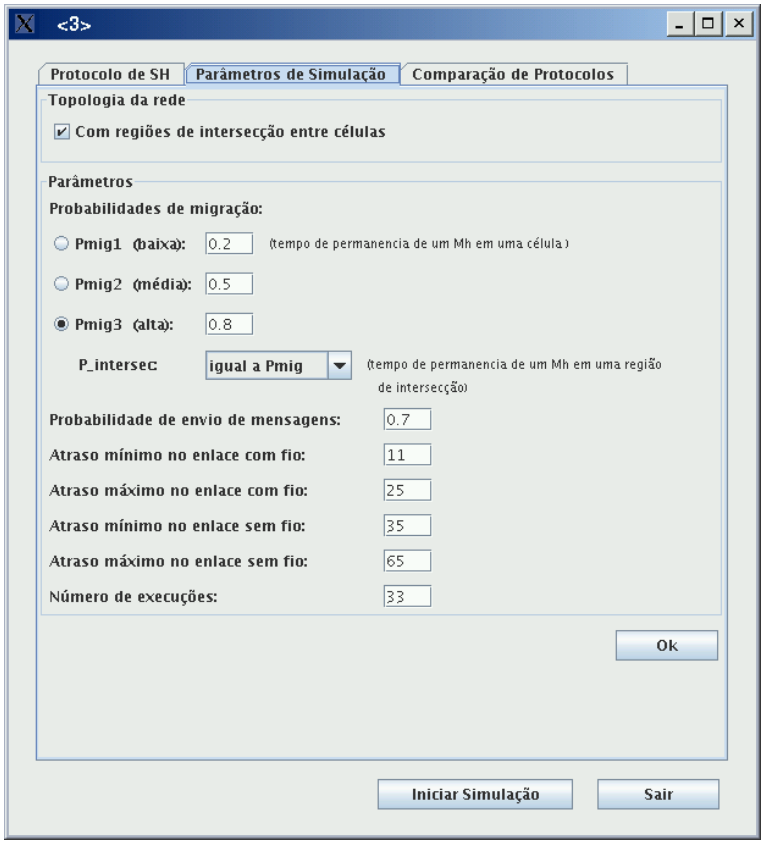


Figura 6.4: Configuração de parâmetros de simulação

assim, a comparação dos protocolos de *handover*.

6.4 Arquivo de Configuração

Cada tipo possível de composição/combinação de esquemas de *handover* com técnicas de otimização gera conjuntos distintos de tipos de mensagens e fluxos de execução. Com a finalidade de facilitar a configuração de protocolos de *handover* suave gerados a partir dessas distintas composições, estabelecemos um arquivo de configuração de protocolos. Esse arquivo contém associações $tipoMensagem \rightarrow modCanSeq$, isto é, tipos de mensagens associadas a uma seqüência de tipos de módulos canônicos que devem tratá-lo, para toda mensagem em cada tipo possível de composição de protocolo. Essas associações possuem a seguinte sintaxe:

$nomeEsquemaHandover\_otmz_1\_...\_otmz_n.elementoRede\_tipoMensagem = mod_1...mod_k$, onde:

- $nomeEsquemaHandover$: nome do esquema de *handover* (que deve ser configurado previamente para todos os tipos de esquemas disponíveis);

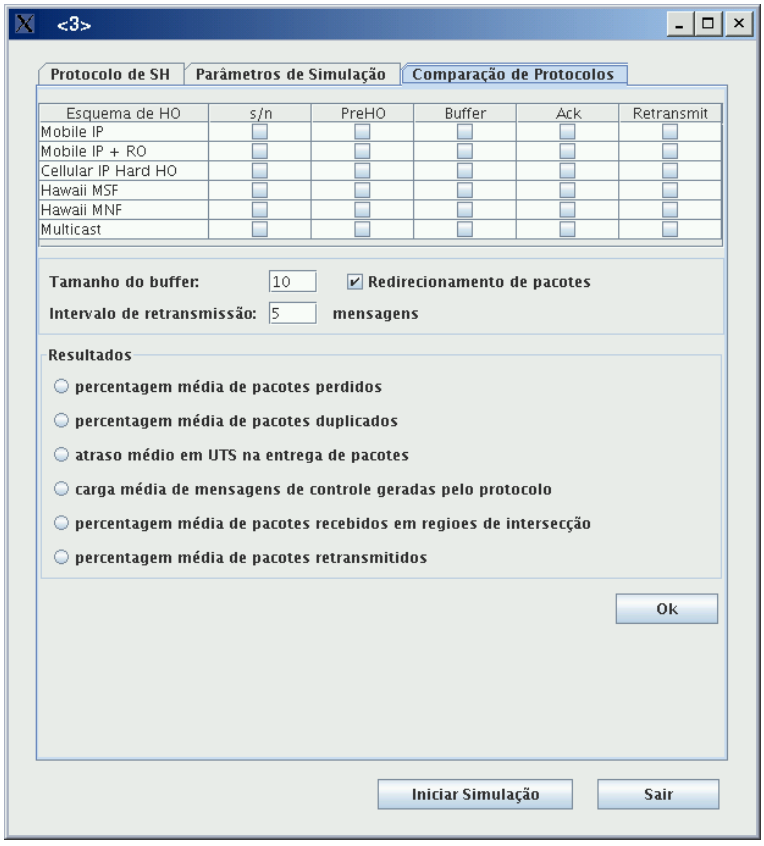


Figura 6.5: Configuração de comparação de vários protocolos de *handover* e otimizações

- $otmz_1\_...\_otmz_n$: nome dos tipos de técnicas de otimizações selecionadas separadas por "_";

- *elementoRede*: elemento de rede que deve tratar o evento;

- $tipoMensagem$: nome da mensagem a ser tratada;

- $mod_1...mod_k$: seqüência de nomes de módulos canônicos tratadores da mensagem, separados por um espaço em branco.

De maneira simplificada, essa associação indica qual é a seqüência de módulos canônicos para um dado tipo de mensagem, em um determinado elemento de rede, quando um esquema de

handover é utilizado e combinado com alguma otimização. Por exemplo, para obter a seqüência de módulos canônicos para a mensagem BindingUpdate no *gateway* com o esquema de *handover* do protocolo Mobile IP com a otimização *buffer*, a seguinte associação deve ser obtida no arquivo de configurações:

$MobileIP\_Buffer.Gateway\_BindingUpdate$.

Esse arquivo de configuração é usado no momento da instanciação dos elementos de rede e módulos canônicos, possibilitando a configuração das estruturas que contêm os fluxos de execução nos elementos Controller, conforme descrevemos acima, no início deste capítulo.

6.5 Estendendo o HOPF

Para acrescentar novos módulos canônicos no HOPF é preciso seguir os seguintes passos:

- A partir da técnica/algoritmo a ser implementado, identificar as tarefas, os tipos de mensagens que são gerados/tratados pelo mesmo e as seqüências de execução relacionadas às mensagens;
- Identificar os elementos de rede que participam da execução das tarefas e os tipos de mensagens que devem gerar/tratar;
- Estender a classe CanonicalModule para cada elemento de rede participante, implementando no corpo do método handleEvent todos os tratadores de mensagens em que esse elemento está envolvido;
- Identificar a possibilidade de composição com outros módulos e determinar as seqüências de execução para cada evento em cada possível composição;
- Acrescentar no arquivo de configuração os eventos associados às seqüências de execução de acordo com a sintaxe apresentada acima.

6.6 Um Exemplo de Geração de Módulos Canônicos para o Cellular IP

A seguir, apresentamos uma seqüência de passos para a geração de módulos canônicos para implementar um protocolo como o Cellular IP.

1. Identificar os elementos de rede e suas funções básicas. A infraestrutura do Cellular IP para mobilidade intra-domínio emprega um tipo especializado de roteadores chamado de

Cellular IP Nodes. Esses roteadores possuem duas principais funções: redirecionamento de pacotes e servir como ponto de acesso para comunicação sem fio com computadores móveis. A conexão entre uma rede Cellular IP e a Internet é feita através de um roteador especial (*gateway*).

2. Identificar a forma de transmissão de pacotes na rede fixa. No caso do *hard handover*, o Cellular IP emprega um único fluxo de pacotes (Unicast).

3. Identificar como o suporte à mobilidade é feito, isto é, como é mantida a localização do computador móvel e como é atualizada. A informação de localização é mantida de uma forma distribuída empregando caches de roteamento específicos nos roteadores da rede Cellular IP. Cada entrada em um cache de roteamento corresponde a um mapeamento do tipo <MhID, NextHop>. As entradas são *soft-state* e são atualizadas por pacotes de dados provenientes do respectivo computador móvel. Quando o computador móvel não estiver enviando dados, uma mensagem especial de controle é enviada periodicamente pelo mesmo.

4. Identificar quem é o responsável para tratar o início do handover (detecção e decisão para procedê-lo). No Cellular IP, o handover é iniciado pelo computador móvel, que faz detecção e decisão baseado em *beacons* enviados periodicamente pelas estações base.

5. Identificar como é feita a transferência do sinal, isto é, se um computador móvel pode "ouvir" a mais de uma estação base ou não. Desde que estamos tratando de *hard handover*, o computador móvel não está habilitado a se conectar a mais de uma estação base simultaneamente, ele pode ouvir a apenas uma estação base de cada vez.

6. Identificar como é feita a reconfiguração do caminho de roteamento. Quando um computador móvel identifica a necessidade de um handover, o mesmo envia uma mensagem RouteUpdate para a nova estação base e o handover se inicia. Ao receber esta mensagem, a nova estação base adiciona uma nova entrada para o computador móvel em seu cache de roteamento e envia a mensagem para seu vizinho no caminho em direção ao *gateway* (*uplink neighbor*). Esse vizinho faz as mesma tarefas e esse procedimento é repetido sucessivamente até qua a mensagem chegue no *gateway*. Porém, os pacotes de dados destinados ao computador móvel são direcionados à nova estação base quando esta mensagem de atualização chega no roteador na intersecção entre os dois caminhos e que está mais próximo da nova estação base (*crossover router*). A mensagem RouteUpdate possui duas funções: atualizar a localização de um computador móvel e, ao mesmo tempo, reconfigurar o caminho de roteamento na rede fixa.

7. Identificar se existe alguma técnica ou otimização é empregada para tratar o fluxo de pacotes durante a transição, como, por exemplo, um buffer. O Cellular IP não emprega nenhum mecanismo particular para a entrega de pacotes durante a transição.

A partir dos passos acima, podemos identificar alguns módulos canônicos conforme ilustramos na Tabela 6.1. Para cada tarefa, temos um número de módulos canônicos e os elementos de rede onde os mesmos devem ser implementados.

Tarefa	Módulo Canônico	Implementado em
Suporte à mobilidade	CIPBaseStationModule	estações base
	CIPRouterModule	roteadores
	CIPRoutingCache	roteadores
Transmissão de pacotes	CIPUnicastModule	roteadores
Detecção de handover	MhHODetectionModule	computador móvel
Transferência do sinal	CIPHardHOModule	estações base
Atualização da localização	CIPLocUpdateModule	estações base
		gateway
Reconfig. caminho de roteamento	CIPRoutingUpdateModule	roteadores

Tabela 6.1: Exemplos de módulos canônicos para o protocolo hard handover do Cellular IP

Capítulo 7

Simulação de Protocolos de Handover

Neste capítulo, apresentamos os resultados das simulações realizadas utilizando-se o HOPF. Foram implementados diferentes esquemas de *handover* baseados em protocolos de mobilidade existentes e propostos na literatura como o Mobile IP, Cellular IP, Hawaii e M&M. Também implementamos algumas técnicas de otimização como *buffer*, *forward*, *pré-handover*, *bicast*, dentre outros, para a combinação com os esquemas de *handover* a fim de comprovarmos a viabilidade da composição de módulos canônicos para melhorar o desempenho dos protocolos durante o *handover*. Para a comparação do desempenho dos protocolos simulados, levamos em consideração alguns requisitos de QoS como, perda de pacotes, atraso/variação do atraso na entrega de pacotes, carga de mensagens de controle de *handover*, pacotes duplicados, replicados e pacotes fora de ordem.

Procedemos as simulações em diferentes cenários de simulação, com distintas topologias de rede, ausência/presença de regiões de intersecção, diferentes freqüências de migração do usuário móvel, dentre outros, conforme apresentamos na próxima seção. Um dos principais resultados que obtivemos com essas simulações foi, além do teste e comparação de diferentes composições de protocolos de *handover* com o HOPF, a geração de um conjunto de regras empíricas para a seleção de módulos canônicos (heurísticas) a partir da verificação dos efeitos causados pelas técnicas de otimização no comportamento e desempenho dos protocolos durante as simulações. Em particular, essas regras empíricas podem ser úteis para fornecer uma diretriz ao usuário do HOPF durante o projeto e desenvolvimento de protocolos de *handover* de acordo com os requisitos particulares de suas aplicações.

Nas próximas seções, apresentamos os esquemas de *handover* e técnicas de otimização implementados, os parâmetros de simulação considerados e os resultados obtidos, assim como o conjunto de regras empíricas para a seleção de módulos canônicos.

7.1 Protocolos Simulados e Otimizações

Implementamos e simulamos seis esquemas de *handover* conforme listamos a seguir, a partir de algumas adaptações de protocolos existentes na literatura. Maiores detalhes sobre o fucionamento desses protocolos e sobre a implementação/composição podem ser encontrados nos Capítulos 4 e 6, respectivamente.

- *MobileIP-like Handover* (MobileIP), esse protocolo consiste em uma simplificação do mecanismo de *handover* do Mobile IP para o caso de micro-mobilidade: consideramos o HA localizado no *gateway* de domínio e FA's localizados em estações base. Toda migração dentro do domínio é notificada ao HA pelo FA da nova estação base e portanto, a mensagem de notificação deve percorrer a rede até alcançar o *gateway* do domínio.

- *MobileIP-Smooth Handover* (MobileIPSH), como uma extensão do protocolo anterior, acrescentamos o mecanismo de *smooth handover* (Seção 4.1.1), que consiste em enviar notificações de mudança de localização não apenas ao HA, mas também ao FA na antiga estação base. Os FA's mantém uma estrutura chamada de *forwarding point* que mantém a nova localização (endereço da nova estação base). Isso permite que pacotes recebidos pela antiga estação base após o *handover* possam ser redirecionados para a nova estação base.

- *CellularIP Handover* (CellularIP), esse protocolo possui dois tipos de *handover*: *hard handover*, que é empregado quando o computador móvel pode se comunicar com apenas uma estação base e *semi-soft handover*, que é apropriado para o caso em que o computador móvel pode "ouvir" a mais de uma estação base simultaneamente. Neste segundo caso, o protocolo implementa uma replicação de pacotes durante o *handover* a fim de otimizar o desempenho. A principal característica do Cellular IP é que a informação de localização de um computador móvel é mantida de maneira distribuída nos roteadores de modo que não é necessária a atualização em um ponto específico da rede (por exemplo, no *gateway*). O caminho de roteamento de pacotes é reparado a partir da nova estação base através de uma notificação enviada à rede em direção ao *gateway*.

- *HAWAII-MSF Handover* (HawaiiMSF), implementa o esquema MSF (*Multiple Stream Forwarding*) para a atualização de caminhos e de localização do computador móvel. Nesse esquema, pacotes na antiga estação base são re-direcionados para a nova estação base e utiliza implicitamente um mecanismo de *buffer* nas estações base para o armazenamento de pacotes.

- *HAWAII-MNF Handover* (HawaiiMNF), implementa o esquema MNF (*Multicast Non-Forwarding*) no qual durante um pequeno intervalo de tempo (durante o *handover*), pacotes

são replicados para a antiga e nova estações base (*bicast*). No HAWAII, em ambos os casos, a informação de localização do computador móvel é mantida de maneira distribuída nos roteadores, assim como no Cellular IP, porém, a principal diferença é que no Cellular IP são empregados roteadores específicos, enquanto que os nós em uma rede HAWAII são roteadores IP com algumas extensões para dar suporte à mobilidade e tratar as mensagens de controle do protocolo HAWAII. Uma outra diferença é que as mensagens de atualização de localização no HAWAII são enviadas à antiga estação base, enquanto que no Cellular IP essas são enviadas em direção ao *gateway*.

- *Multicast-based Handover* (Multicast), esse protocolo emprega *multicasting* para a transmissão de pacotes na rede de modo que todas as estações base vizinhas à estação base corrente recebam réplicas dos pacotes destinados ao computador móvel. Isso permite que o fluxo de pacotes seja desviado rapidamente para o computador móvel a partir da nova estação base após um *handover*.

Implementamos as seguintes técnicas de otimização:

- Buffer: este mecanismo permite o armazenamento temporário de pacotes em estações base. Pacotes recebidos pela estação base são armazenados no *buffer* e, quando o mesmo fica cheio (*buffer overflow*), o pacote com o menor *timestamp* é removido. Utilizamos em nossas simulações *buffers* de tamanho fixo igual a 10.

- Forward: complementa o módulo Buffer e permite o redirecionamento dos pacotes armazenados no *buffer* para a nova estação base. Nem todos os protocolos requerem a combinação com este módulo quando empregam o Buffer, por exemplo, protocolos como o Multicast podem ser combinados com o módulo Buffer porém não necessitam o redirecionamento de pacotes uma vez que os pacotes são replicados para todas as estações base vizinhas.

- PreHandover (PreHO): técnica para antecipar o *handover*, na qual assume-se a existência de um evento da camada de enlace que indica a iminência de um *handover*. Esse evento tem como parâmetro a estação base candidata à nova estação e através dessa informação, um novo caminho de roteamento de pacotes é configurado na rede e elementos que mantém a informação de localização do computador móvel na rede são notificados antecipadamente. O procedimento de *handover* em si inicia-se após essa pré-configuração.

- Bicast: esta técnica pode ser combinada com o PreHandover para permitir uma replicação de pacotes para a antiga e nova estações base a partir do momento em que o novo caminho de roteamento de pacotes é gerado e, essa replicação ocorre durante todo o procedimento de *handover*.

- Retransmission (Retrans): este mecanismo é empregado para possibilitar a retransmissão de pacotes em intervalos de tempo (simulado) ao computador móvel, a fim de reduzir a perda de pacotes e aumentar a confiabilidade.

- Ack: essa técnica permite a confirmação do recebimento de pacotes pelo computador móvel e pode ser combinada com Retransmission para evitar retransmissões sucessivas de um mesmo pacote.

Essas técnicas foram implementadas em módulos canônicos e foram utilizadas para testar novas combinações (ou composições) com os esquemas de *handover* citados acima. Um dos objetivos disso foi verificar como que estas técnicas podem influenciar no desempenho do *handover* para cada um desses esquemas e analisar os possíveis benefícios para a provisão de QoS.

7.2 Aspectos Gerais das Simulações

Dentre os objetivos das simulações, podemos citar: implementar, testar e comparar protocolos de *handover* para micro-mobilidade existentes na literatura na filosofia modular do HOPF; testar e analisar os efeitos da composição de diferentes módulos canônicos no desempenho de protocolos; comparar qualitativamente o desempenho dos protocolos com relação a alguns requisitos de QoS como: número médio de pacotes perdidos, duplicados ou fora de ordem, valor médio do atraso/variação do atraso, carga média de mensagens de controle de *handover* e de pacotes replicados/redirecionados na rede; e a geração de um conjunto de regras empíricas para a seleção de módulos canônicos.

Realizamos simulações baseadas em *hard* e *soft handover*, e *handover* reativo e pró-ativo. A fim de possibilitar o teste e comparação entre *hard* e *soft handover*, utilizamos topologias de rede sem e com áreas de intersecção de células, respectivamente. Para simular *hard handover*, que é o caso em que o computador móvel pode se comunicar com apenas uma estação base, utilizamos a topologia de rede sem áreas de intersecção entre células, de modo que o computador móvel possa receber pacotes somente da estação base com a qual está conectado em um dado momento. A topologia com áreas de intersecção de células permite simular *soft handover* uma vez que para cada migração entre células, o computador móvel deve passar por uma área de intersecção e, durante o momento em que o computador móvel se encontra nessa área, o mesmo pode "ouvir" a antiga e nova estações base e receber pacotes de ambas. *Handover* reativo é o tipo de *handover* abrupto, no qual o computador móvel perde a conexão com a estação base de maneira repentina, enquanto que no caso de *handover* pró-ativo o computador móvel é notificado antecipadamente sobre a necessidade de iniciar um *handover* e, através disso, são executadas algumas tarefas para

um pré-processamento do *handover*. Em nossas simulações, a principal diferença entre *handover* reativo e pró-ativo é através da combinação ou não da técnica Pré-Handover em um esquema de *handover*.

Na próxima seção apresentamos os parâmetros utilizados nas simulações e em seguida, os resultados obtidos.

7.2.1 Parâmetros de Simulação

Consideramos as topologias de rede ilustradas na Figura 7.1, que são semelhantes, a menos das áreas de interseção entre células (Figura 7.1(b)). Consideramos os seguintes elementos de rede: um *gateway* de domínio, um computador móvel, um nó fonte que faz o envio de pacotes de dados em uma certa freqüência ao computador móvel, nós intermediários (roteadores), estações base e as suas respectivas áreas de cobertura (células). A escolha dessa topologia nos permite ter diferentes "distâncias" (em número de *hops*) entre *gateway* e estações base assim como entre as próprias estações base, e isso nos possibilita a comparação do desempenho dos protocolos com diferentes esquemas de atualização de localização do computador móvel na rede durante o *handover*.

Um *crossover router* (CR) é o roteador que está na intersecção de caminhos a partir de estações base em direção ao *gateway* e que está no nível mais baixo possível (mais "próximo") das estações base. Por exemplo, na Figura 7.1, r_3 é o *crossover router* de Mss_1 e Mss_2, r_2 de Mss_2 e Mss_3 e, o *gateway* é o *crossover router* de Mss_3 e Mss_4. Conforme veremos nas próximas seções, a distância do *crossover router* pode influenciar no desempenho de alguns protocolos de *handover*, e, em particular, na Seção 7.3.5 apresentamos uma comparação dos protocolos de acordo com diferentes distâncias do *crossover router*.

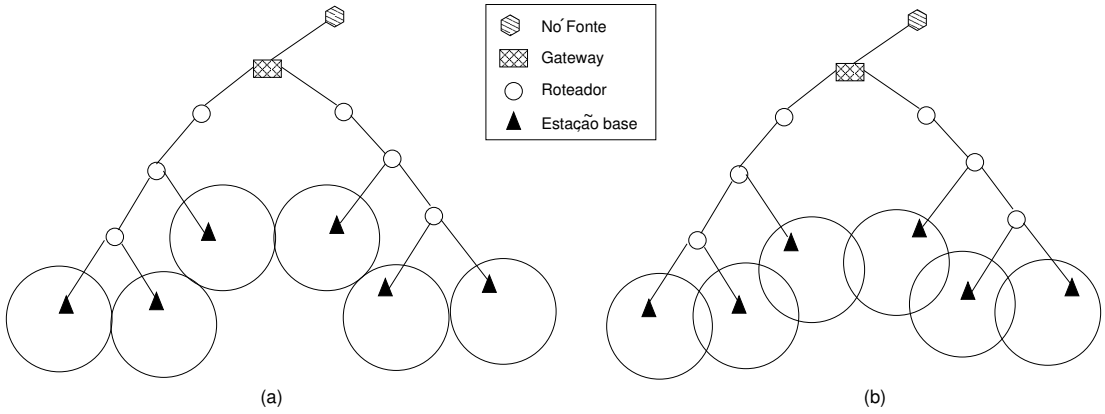


Figura 7.1: Topologias de rede utilizadas nas simulações: (a) sem regiões de intersecção e (b) com regiões de intersecção entre células

O computador móvel segue um padrão de mobilidade retilíneo, com um percurso de migração pré-definido, entre as células da esquerda para a direita e vice-versa, passando pelas regiões de interseção, no caso em que é utilizada a topologia (b). Utilizando um itinerário fixo dessa maneira, podemos comparar os resultados das simulações dos protocolos sem a preocupação se as migrações se concentraram em apenas uma ou outra parte específica da rede simulada.

Consideramos três taxas de mobilidade para o computador móvel: baixa, média e alta, de acordo com as probabilidades de geração de eventos de migração (P_{mig}) iguais a 0.2, 0.3 e 0.8, respectivamente. A probabilidade de migração para uma região de intersecção de células (P_{mig_inter}) é igual ao dobro de P_{mig}, a fim de retratar um tempo de permanência menor nessas regiões.

Os eventos de migração foram gerados em intervalos de tempo de 350 UTS (Unidades de Tempo Simulado)[1]. Em cada simulação, consideramos um número fixo de eventos de envio de mensagens pelo nó fonte igual a 250 mensagens, gerados em intervalos de tempo de 40 UTS.

Nos enlaces com fio, atribuímos um valor de atraso fixo de 20 UTS para cada conexão e, nos enlaces sem fio, o valor do atraso é igual a 90 UTS. Esses valores foram escolhidos arbitrariamente, embora de forma apropriada a fim de evitar um congestionamento de mensagens na rede fixa com relação à taxa de envio de mensagens pela fonte.

7.3 Resultados

Nesta seção, apresentamos os resultados obtidos a partir das simulações. Dividimos os resultados de acordo com parâmetros de QoS, onde para cada parâmetro (i.e., perda de pacotes, atraso, carga de mensagens de controle, etc.) apresentamos uma comparação dos resultados obtidos com os esquemas de *handover* apresentados na Seção 7.1 e os mesmos combinados com uma ou mais técnicas de otimização (módulos Buffer, associado ao módulo Forward para o armazenamento e redirecionamento de pacotes, Ack+Retransmission para a confirmação de recebimento e retransmissão de pacotes e PreHandover combinado com Bicast para o pré-processamento de *handover* e replicação de pacotes. Essa forma de organização dos resultados permite a comparação do desempenho dos esquemas de *handover* e o impacto causado pela combinação de uma ou mais técnicas aos mesmos.

Também comparamos os protocolos e combinações de técnicas de acordo com diferentes topologias de rede (Seção 7.3.5), em particular com relação a diferentes distâncias (em número de *hops*) entre o *gateway* e *crossover router* e entre este e estações base. Através dessas comparações foi possível constatar a influência de uma topologia de rede no desempenho dos protocolos e

[1] No MobiCS, o tempo simulado não possui qualquer relação com tempo real.

identificar a viabilidade de se empregar uma determinada técnica de acordo com o tipo de topologia de rede.

Nos gráficos que apresentamos a seguir, cada um dos pontos corresponde à média dos resultados obtidos em 33 simulações realizadas. No eixo x, temos o número médio de migrações com ocorrências de *handover* e no eixo y, o valor médio do parâmetro de QoS em questão. Cada uma das barras representa um determinado protocolo de *handover* combinado ou não a uma técnica de otimização.

7.3.1 Pacotes Perdidos

Nesta seção, comparamos os resultados obtidos das simulações com relação ao número médio de pacotes perdidos. Esse número foi determinado a partir da média das diferenças entre o número de pacotes enviados pelo nó fonte e o número de pacotes recebidos pelo computador móvel em cada simulação.

Nos gráficos da Figura 7.2 podemos observar a variação do número de pacotes perdidos de acordo com o tipo de otimização empregada para *hard handover* (onde utilizamos a topologia de rede da Figura 7.1-(a)). No gráfico 7.2-(a) temos os resultados das simulações dos protocolos de *handover* puros, isto é, sem o emprego de nenhuma técnica de otimização. Conforme podemos observar, quando a taxa de migração é alta, também é alta a quantidade de pacotes perdidos, principalmente, para o MobileIP, com uma média de 80% de perdas. O MobileIPSH apresenta uma melhora de 20% com relação ao MobileIP e isso se deve ao emprego da técnica de *forwarding pointer*, no qual a antiga estação base mantém um ponteiro associado ao computador móvel e na ocorrência de um *handover*, a nova estação base notifica a antiga estação sobre a nova localização. Dessa forma, qualquer pacote recebido durante a fase de transição na antiga estação base é redirecionado para a nova a estação. Os protocolos HawaiiMSF, que emprega Buffer e redirecionamento de pacotes e Multicast, que faz replicações de pacotes para todas as estações base vizinhas, apresentaram o melhor desempenho com aproximadamente 20% de perdas quando a taxa de migração é alta. No caso do Multicast, apesar da replicação de pacotes, as perdas se justificam devido ao fato de que as estações base recebem os pacotes em tempos (simulados) diferentes, devido à diferença do número de *hops* entre o *gateway* e estações base e os atrasos (totais) nos canais de comunicação com fio entre o *gateway* e estações base. Quando a atual estação base recebe pacotes com maior atraso em comparação aos pacotes recebidos na nova estação base (a atual estação base está "atrasada" com relação à nova estação base), alguns pacotes podem ser descartados na nova estação base até que o computador móvel se conecte à mesma durante o *handover*.

Observando-se os resultados dos protocolos combinados com o módulo Buffer (gráfico 7.2-

(b)), verificamos que houve uma melhoria aproximada de 30% para os protocolos baseados no MobileIP quando comparados aos mesmos sem otimizações (gráfico 7.2-(a)). No caso do protocolo Multicast, a composição com Buffer possibilitou zerar o número de pacotes perdidos (gráficos 7.2-(b), (d), (f)). Com exceção do protocolo Multicast, para todos os protocolos essas melhorias se devem ao armazenamento de pacotes na antiga estação base durante a fase de transição (transferência da conexão), evitando-se a sua perda, pois são redirecionadas para a nova estação base após a retomada da comunicação com a mesma. No caso Multicast, uma vez que os pacotes são replicados para todas as estações base vizinhas, não existe a necessidade do redirecionamento de pacotes, porém, o uso de Buffer em cada estação base permite reduzir as diferenças dos tempos de recebimento de pacotes pelas estações base, conforme o problema mencionado no parágrafo anterior e, com isso, reduzir as perdas.

No caso da combinação com o módulo PreHO (gráfico 7.2-(c)), também podemos observar uma considerável melhora com relação aos resultados dos protocolos sem otimizações (gráfico 7.2-(a)), principalmente para o Mobile IP, em que a redução de perdas foi maior do que aquela com o uso de Buffer (gráfico 7.2-(b)). O PreHO antecipa o procedimento de *handover*, notificando a futura estação base e os elementos que mantém a informação de localização do computador móvel na rede (por exemplo, *gateway* ou roteadores), possibilitando uma prévia configuração do novo caminho de roteamento de pacotes até a futura estação base (*handover* pró-ativo). Além disso, essa técnica está associada à técnica de replicação de pacotes, na qual após a geração do novo caminho, pacotes são replicados para a atual e futura estações base (*bicast*) durante o procedimento de *handover*. Conforme veremos, esta técnica tem um maior efeito quando a rede possui regiões de intersecção entre células (Figura 7.3). A técnica PreHO combinada com o módulo Buffer permitiu uma maior redução no número de perdas, principalmente para o protocolo Mobile IP, com uma redução de aproximadamente 50% com relação ao caso em que nenhuma otimização foi empregada. Em paticular, o protocolo Multicast é o único caso em que o módulo PreHO não se aplica uma vez que o mesmo emprega implicitamente um mecanismo de antecipação de *handover*, replicando pacotes não apenas para uma, mas para um conjunto de estações base que são consideradas como candidatas à futura estação base.

A combinação dos módulos Ack+Retrans permitiu o melhor desempenho com relação ao número de perdas para todos os protocolos e isso se deve, em particular, ao re-envio de pacotes em intervalos de tempo pela estação base até a confirmação de recebimento dos pacotes pelo computador móvel (gráficos 7.2-(e), (f)). Em particular, a combinação com a técnica Buffer, o desempenho se demonstrou ainda melhor (gráfico 7.2-(f)), uma vez que uma requisição para a retransmissão de um pacote é enviada ao nó fonte somente quando o mesmo não se encontra no *buffer*. Porém, conforme veremos nas próximas seções, essas técnicas apesar de oferecerem uma

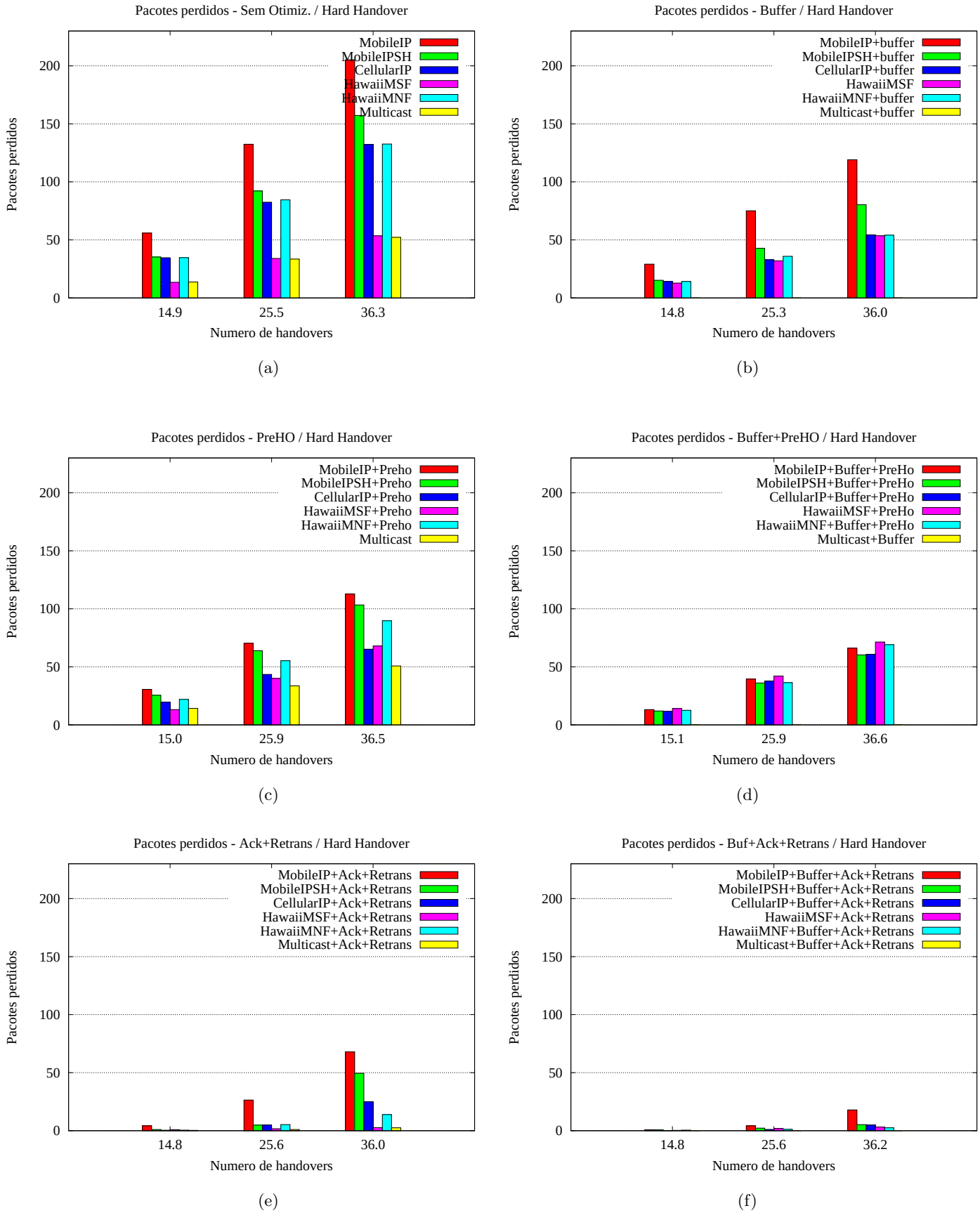


Figura 7.2: Perda de pacotes (*hard handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f) Buffer+Ack+Retrans

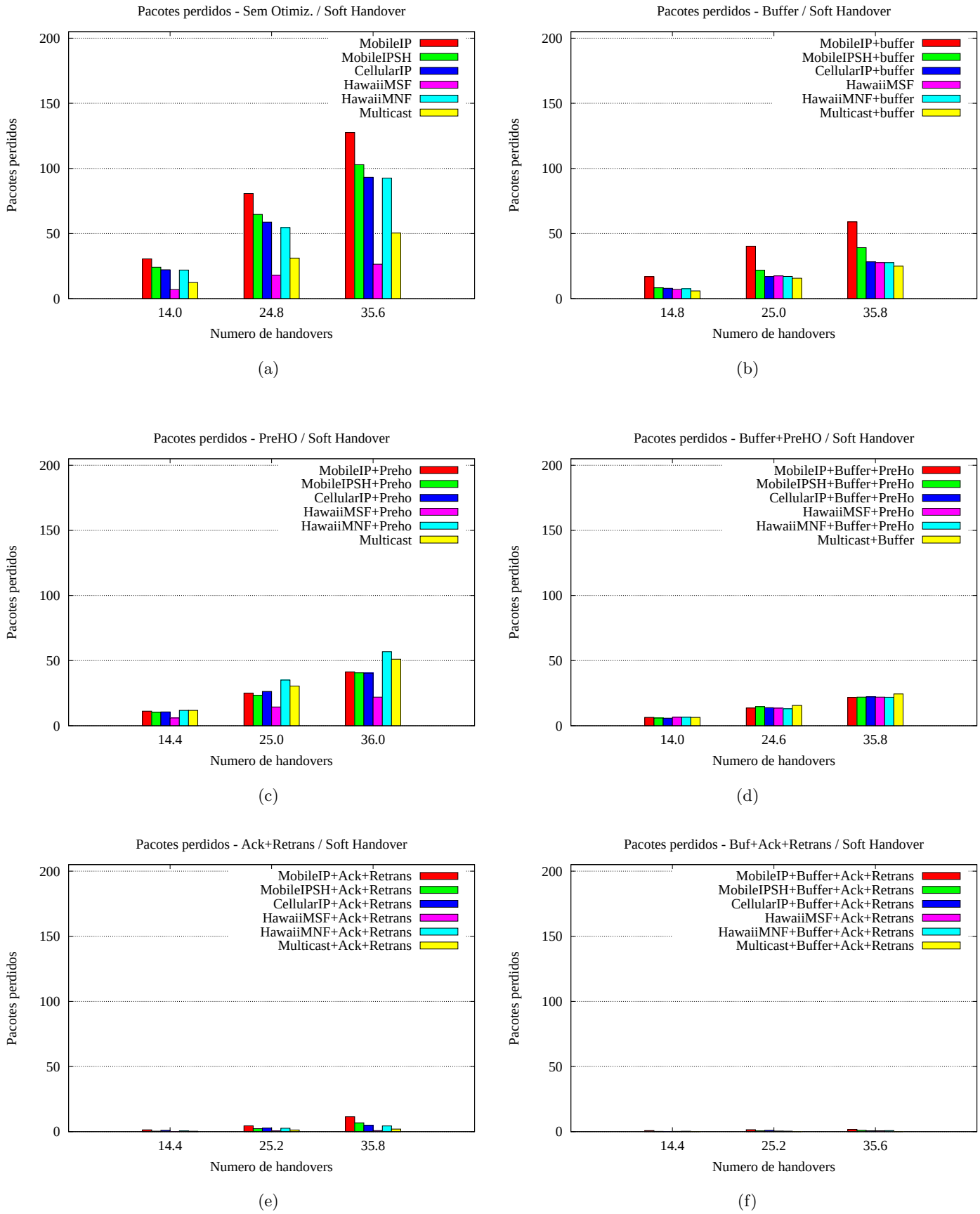


Figura 7.3: Perda de pacotes (*soft handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f) Buffer+Ack+Retrans

maior *suavidade* com relação ao requisito perda de pacotes, podem causar um grande atraso na entrega de pacotes além de uma considerável sobrecarga de mensagens de controle.

Na Figura 7.3 temos os resultados das simulações para as mesmas configurações, parâmetros de simulação e combinações de módulos para o caso de *soft handover*, no qual a diferença está na topologia de rede (Figura 7.1-(b)), com regiões de intersecção entre as células onde o computador móvel pode receber pacotes das duas estações base correspondentes.

Conforme podemos observar, somente pela existência dessas regiões de intersecção, há uma grande redução no número de pacotes perdidos até mesmo quando nenhuma otimização é empregada (gráfico 7.3-(a)). Isso porque enquanto o computador móvel está em uma região de intersecção, o mesmo é capaz de "ouvir" às duas estações base e, portanto, todos os pacotes enviados pela antiga estação base continuam sendo recebidos pelo mesmo possibilitando uma continuidade na entrega de pacotes por um período de tempo maior, e, em conseqüência, uma perda menor. Em paticular, o módulo PreHO se beneficia quando há regiões de intersecção, uma vez que a pré-configuração de caminho pode ser executada enquanto o computador móvel está em uma região de intersecção e, a decisão para tratar a transferência da comunicação de uma estação base para outra pode ser feita em um momento específico, de acordo com algum critério, por exemplo, no momento em que a nova estação base receber o primeiro pacote replicado.

7.3.2 Atraso e Variação do Atraso

Atraso e variação do atraso são requisitos de QoS importantes para muitas aplicações, por exemplo, aplicações multimídia de tempo real. No MobiCS, a noção de tempo não possui qualquer relação com o tempo real, é utilizado o conceito de Unidade de Tempo Simulado (UTS). Eventos são agendados em uma determinada freqüência em UTS no início da simulação, e são disparados durante a simulação no "tempo" (simulado) determinado. Nesse sentido, os resultados obtidos com as simulações não são comparáveis quantitativamente a qualquer resultado baseado em tempo real. Porém, esses resultados nos fornece uma noção da tendência do comportamento dos protocolos e técnicas combinadas e nos permite uma comparação entre os mesmos.

O atraso corresponde ao "tempo" médio desde o momento em que um pacote foi enviado pelo nó fonte até o momento em que o mesmo é recebido pelo computador móvel. Uma das dificuldades encontradas foi identificar exatamente o intervalo de tempo de cada procedimento de *handover* e medir o atraso para cada pacote durante este intervalo de tempo. Em vez disso, utilizamos a seguinte estratégia: uma estação base ao receber um pacote para um computador móvel, acrescenta ao mesmo o atual *timestamp* de simulação ao pacote[2]. Quando um computador

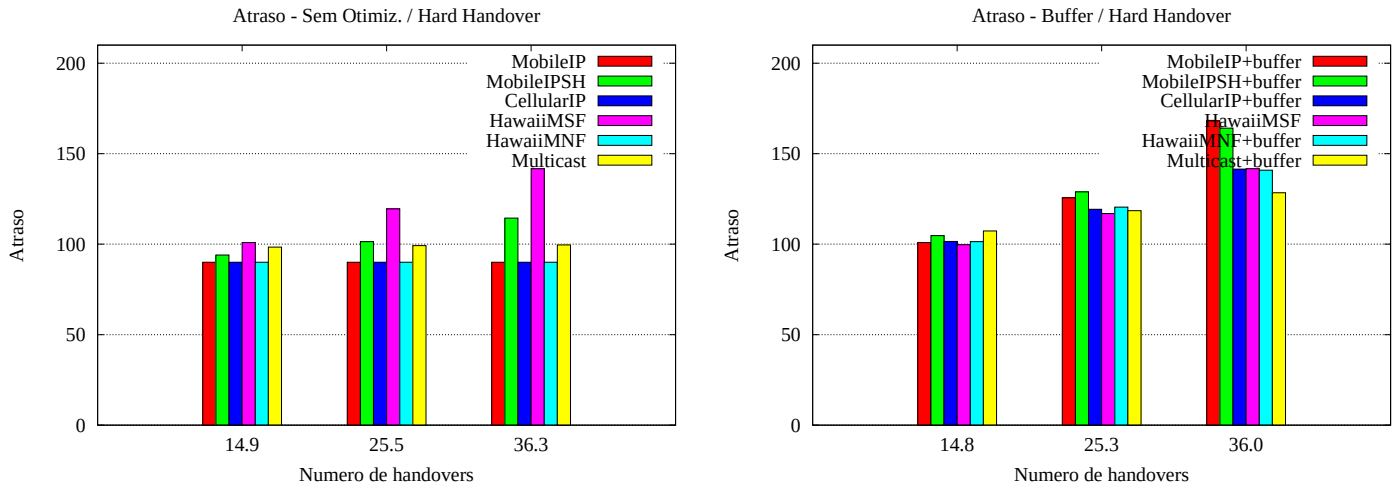
[2]No MobiCS a simulação de protocolos se baseia em *timestamps*, que correspondem valores inteiros e que indicam um momento específico da simulação.

móvel recebe o pacote, este obtém o atual *timestamp* e calcula o atraso subtraindo-se o *timestamp* contido no pacote. Dessa forma, o atraso é calculado apenas a partir do momento em que o pacote é recebido por uma estação base até o momento em que o mesmo é recebido pelo computador móvel, considerando-se os intervalos de tempo devido aos procedimentos de *handover*. Pacotes que são recebidos enquanto o computador móvel não está em migração, ou seja, pacotes que são recebidos fora de um intervalo de execução do *handover* são entregues com um atraso igual ao valor do atraso no canal de comunicação sem fio (90 UTS).

Nas Figuras 7.4 e 7.5 apresentamos os resultados para os valores médios do atraso para os casos de *hard* e *soft handover*. Conforme podemos observar no gráfico 7.4-(a), os protocolos MobileIP, CellularIP, e HawaiiMNF, apresentam um atraso igual ao próprio atraso no canal de comunicação sem fio (90 UTS) uma vez que estes não empregam qualquer técnica para o redirecionamento de pacotes na rede. No caso do MobileIPSH (que usa *forwarding points*), HawaiiMSF (que usa *buffer*) e Multicast (replicação de pacotes) podemos verificar um acréscimo no valor do atraso.

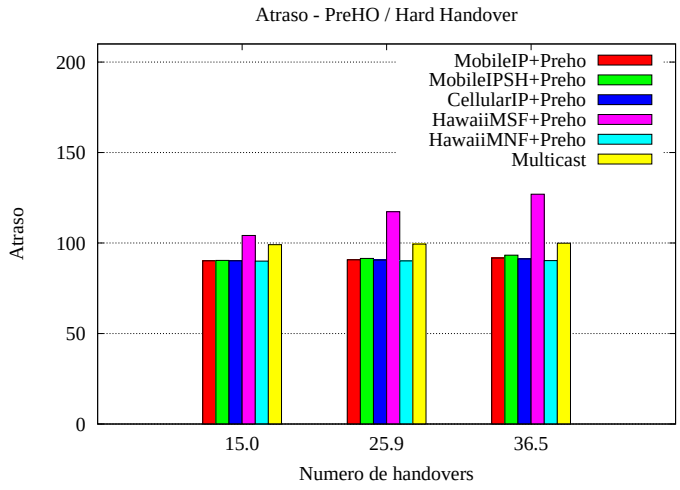
Com o emprego do Buffer, podemos observar um maior atraso em todos os protocolos, principalmente para o MobileIP e MobileIPSH, uma vez que estes fazem o redirecionamento de pacotes da antiga para a nova estação base através do *gateway*, ou seja, todo pacote redirecionado deve passar pelo *gateway* devido ao fato de que a forma de encaminhamento de pacotes na rede para esses protocolos se baseia em encapsulamento+tunelamento. Já os protocolos baseados em roteamento específico, no qual os roteadores mantém informações do computador móvel, os pacotes redirecionados são desviados para a nova estação base em algum ponto mais próximo, isto é, no ponto de intersecção entre os dois caminhos (*crossover router*).

No caso da utilização da técnica PreHO (gráfico 7.4-(c)) conforme podemos observar, apenas o protocolo HawaiiMSF apresentou um pequeno atraso, devido ao uso implícito de Buffer e esse acréscimo podemos verificar no gráfico seguinte (7.4-(d)), onde foi empregada a combinação PreHO+Buffer. Um atraso excessivamente alto foi observado na combinação Ack+Retrans, principalmente para o protocolo MobileIP. Essa técnica permite a retransmissão de pacotes em intervalos de tempo de acordo com a requisição do computador móvel. Ao receber um pedido de retransmissão, o nó fonte gera uma réplica do pacote enviado anteriormente, associando o mesmo *timestamp* atribuído anteriormente. O principal problema com o MobileIP é que os pacotes retransmitidos durante o intervalo de tempo desde o início do *handover* e até o recebimento de uma notificação de atualização de localização são perdidos, pois são enviados à antiga estação base. Porém, no caso do CellularIP e Hawaii (MSF e MNF), a probabilidade de perda é menor uma vez que o caminho de roteamento de pacotes é atualizado após a migração (a partir da nova estação base em direção ao *gateway*), permitindo que os pacotes sendo retransmitidos durante o

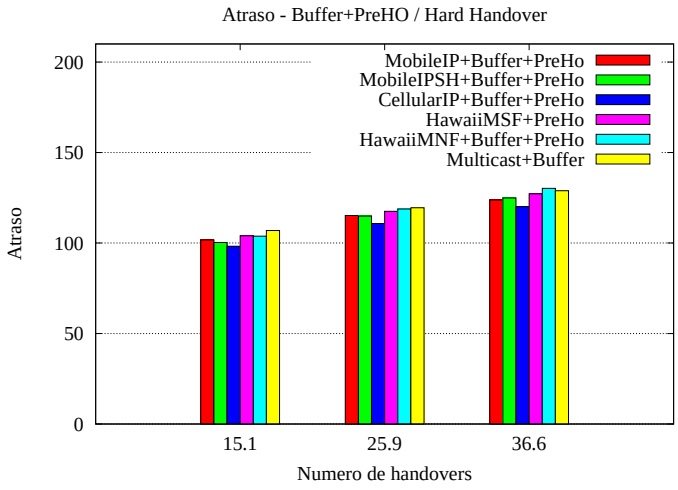


(a)

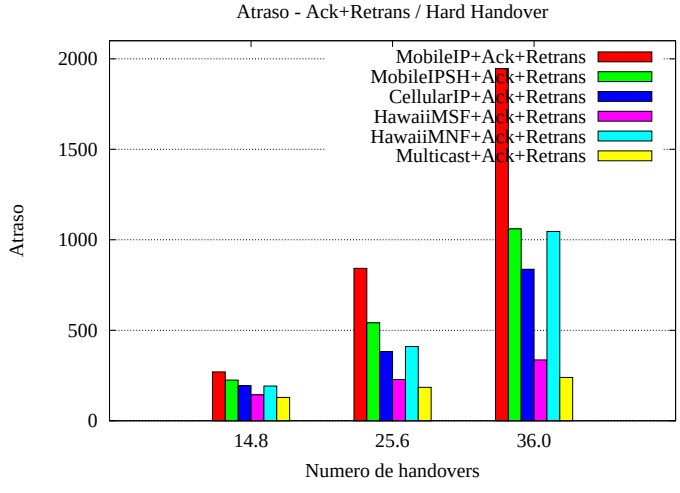
(b)



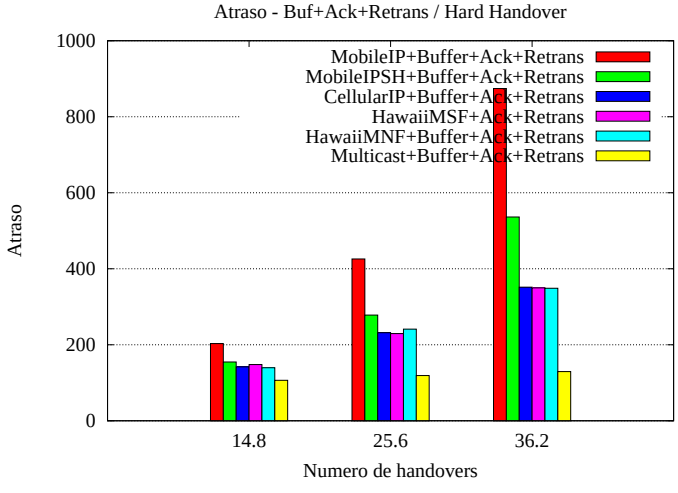
(c)



(d)

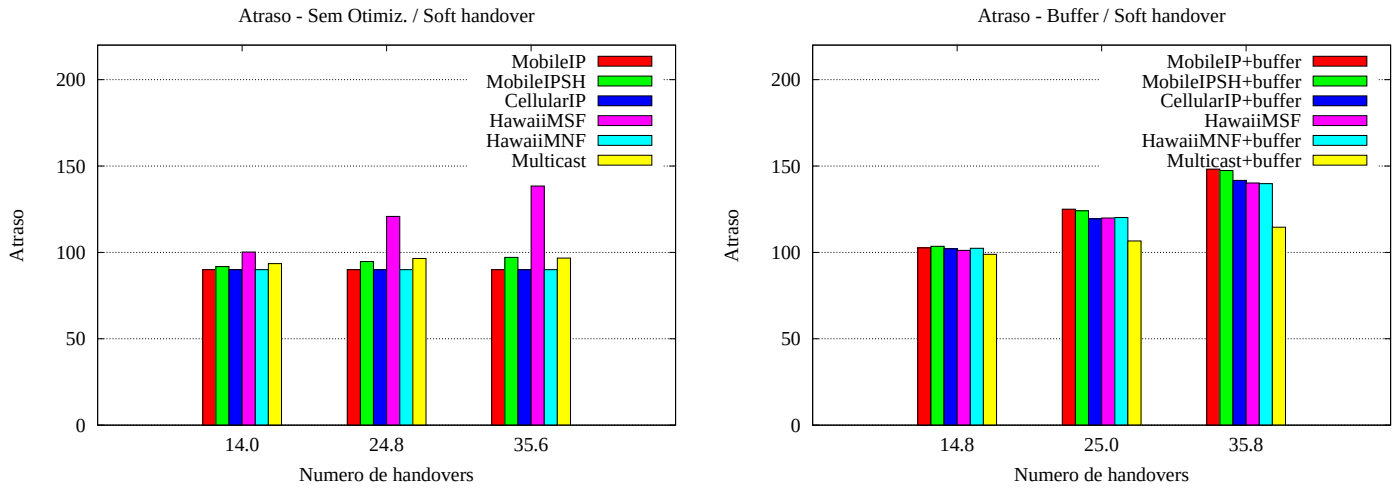


(e)



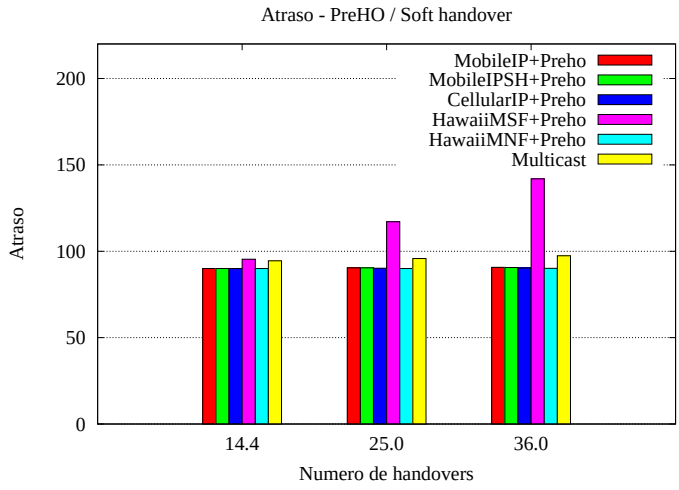
(f)

Figura 7.4: Atraso médio (em UTS - *hard handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f)Buffer+Ack+Retrans

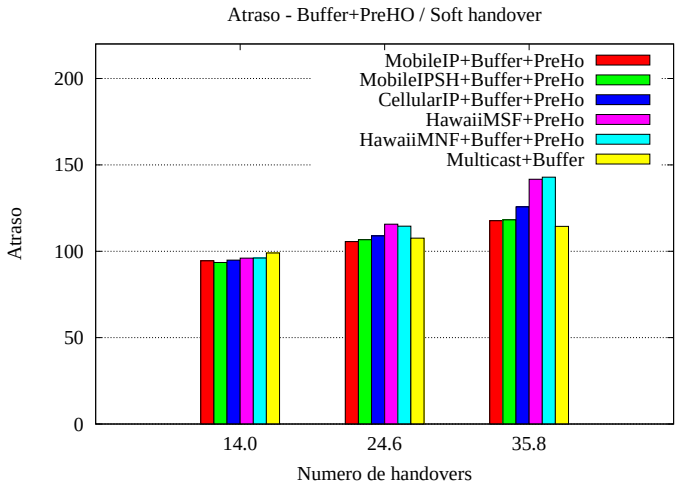


(a)

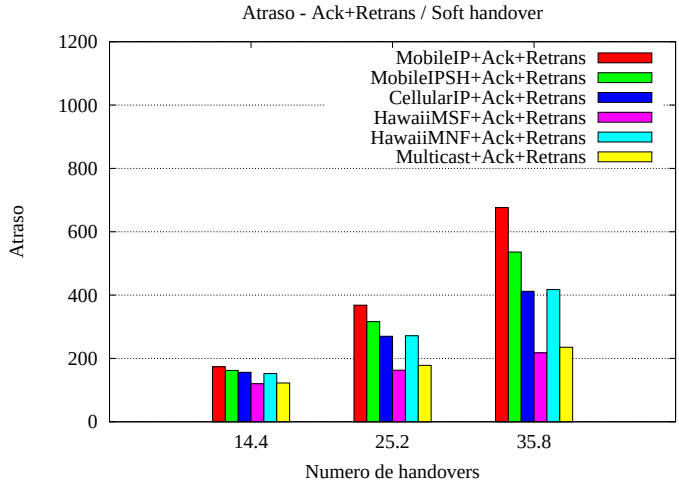
(b)



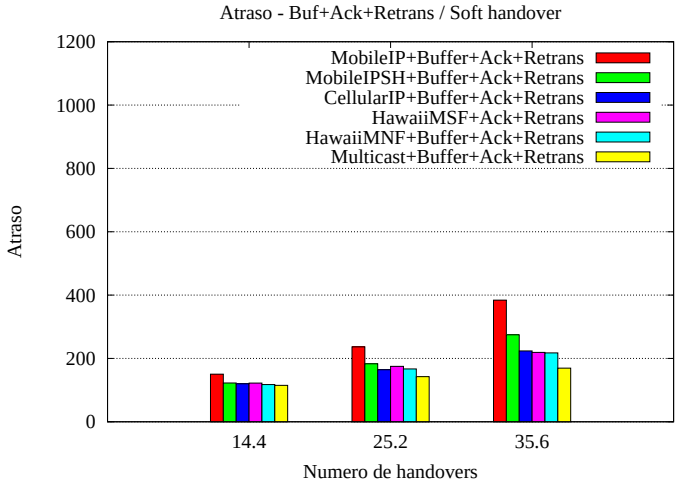
(c)



(d)



(e)



(f)

Figura 7.5: Atraso médio (em UTS - *soft handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f)Buffer+Ack+Retrans

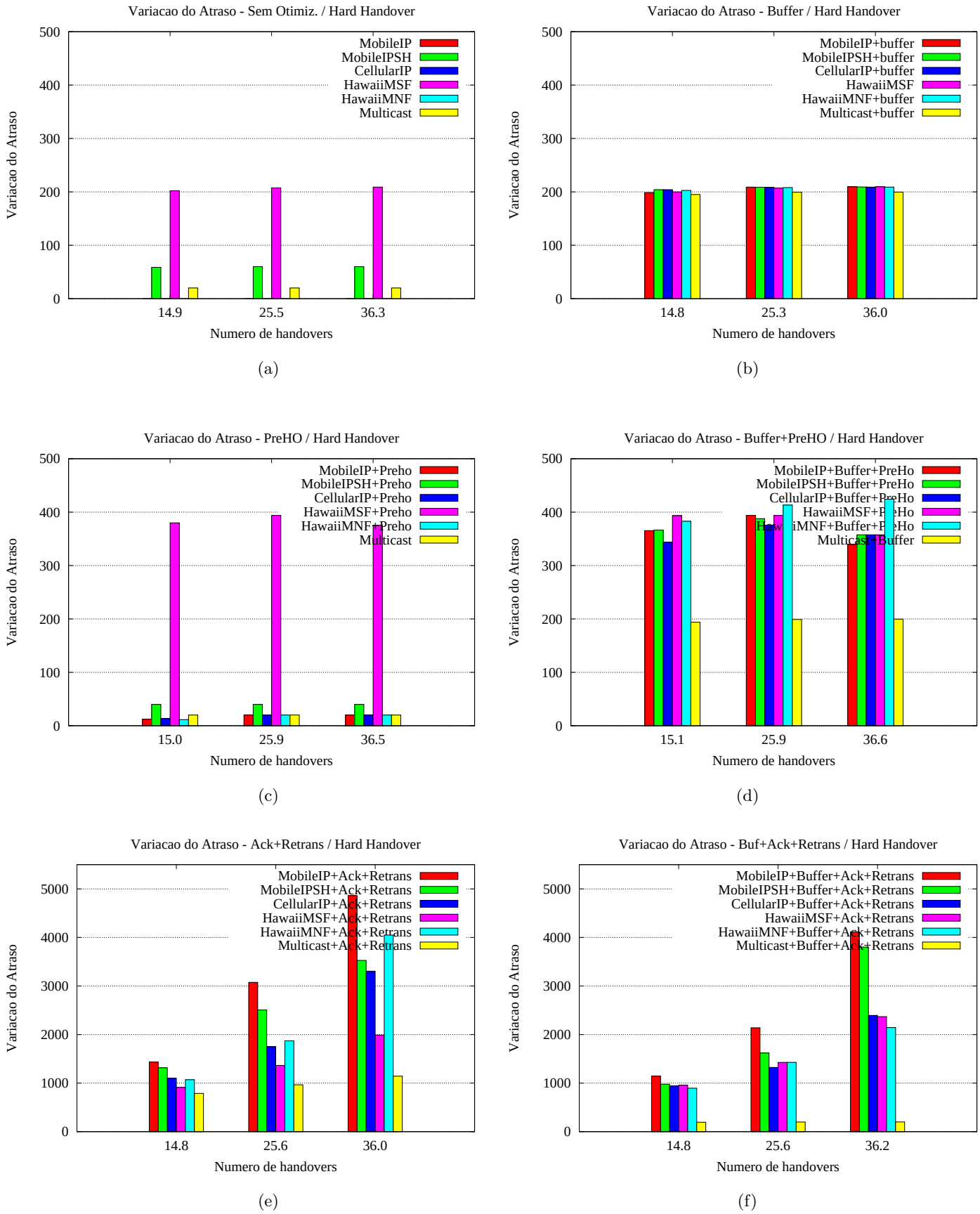


Figura 7.6: Variação média do atraso (*hard handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f)Buffer+Ack+Retrans

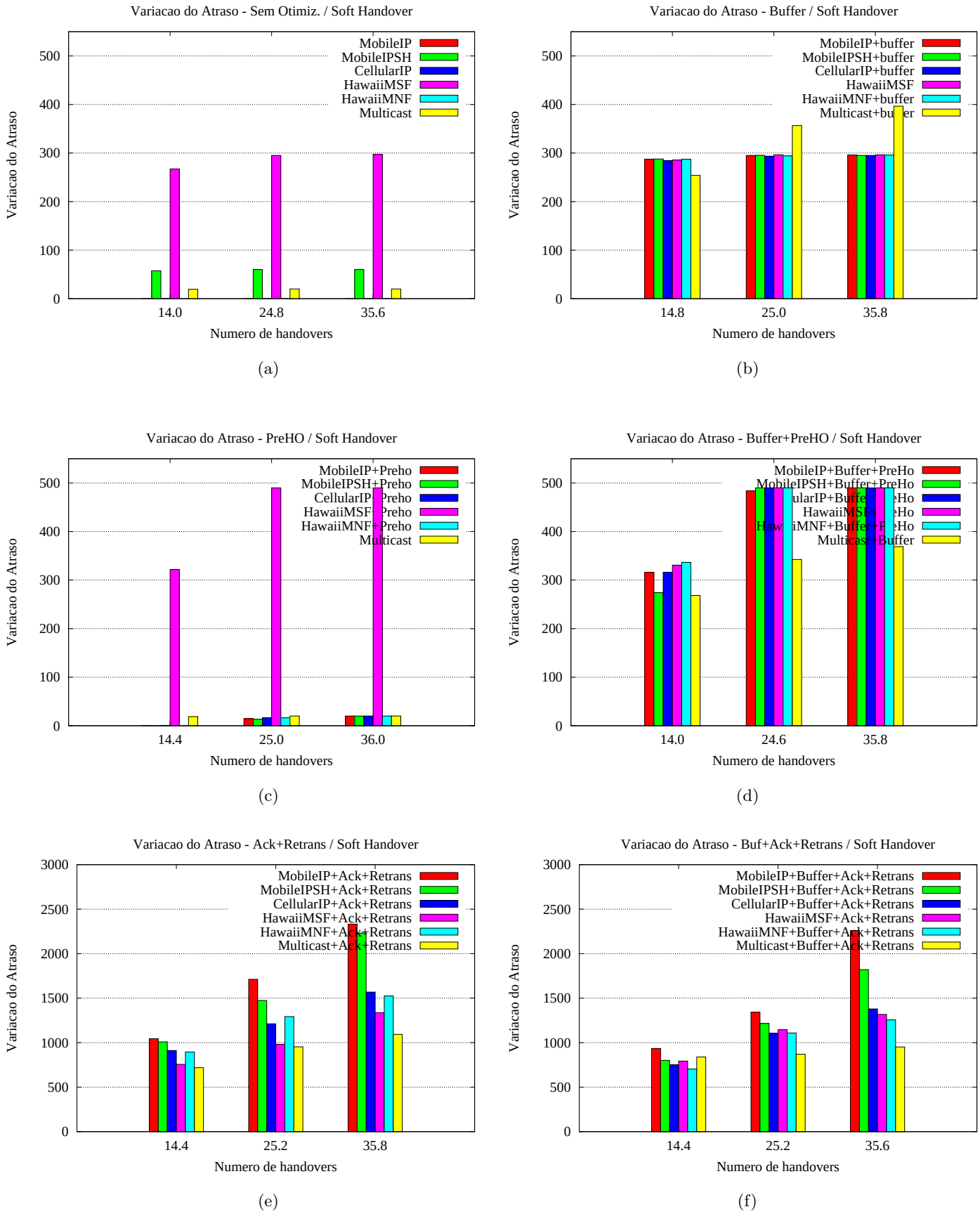


Figura 7.7: Variação média do atraso (*soft handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f)Buffer+Ack+Retrans

handover sejam desviados em algum ponto (*crossover router*) para a nova estação base.

Porém, com a combinação com o módulo Buffer, o atraso reduziu para menos da metade com relação ao gráfico anterior no caso de alta taxa de migração para a maioria dos protocolos (gráfico 7.4-(f)). Isso se deve, basicamente, ao menor número de requisições de pacotes ao (possivelmente distante) nó fonte, e isso ocorre apenas quando um pacote não está mais no *buffer*.

Na Figura 7.5 apresentamos os resultados no caso da rede com regiões de intersecção. A principal diferença está na redução do atraso em aproximadamente 70% com relação à composição Ack+Retrans e 50% no caso de Buffer+Ack+Retrans (gráficos 7.5-(e) e 7.5-(f)).

Para o cálculo da variação do atraso, consideramos a diferença entre os valores mínimo e máximo do atraso em cada execução e obtivemos a média de todas as execuções. Nas Figuras 7.6 e 7.7 apresentamos os resultados para a variação do atraso para as topologias de rede sem e com regiões de intersecção, respectivamente.

Podemos observar que o uso de Buffer pode causar uma considerável variação do atraso (gráficos 7.6-(a) (no caso do protocolo HawaiiMSF), (b) e (d)), porém, no caso da combinação Ack + Retrans essa variação é extremamente alta, principalmente para o protocolo MobileIP e quando a freqüência de migração é alta (gráficos 7.6-(e) e (f)). O protocolo Multicast foi o que apresentou uma menor variação do atraso, mesmo para altas taxas de mobilidade, o que indica que o mesmo é um protocolo mais estável com relação à variação do atraso.

Quando comparamos com os resultados obtidos na rede com regiões de intersecção, observamos que embora para os casos de combinações com Buffer e Buffer + PreHO houve um pequeno aumento na variação do atraso (gráficos 7.7-(b) e (d)), para os casos de Ack + Retrans e Buffer + Ack + Retrans, houve uma redução de aproximadamente 50%, conforme podemos observar nos gráficos 7.7-(e) e (f). A principal razão é que com as regiões de intersecção, a probabilidade de recebimento de pacotes retransmitidos é maior e desta forma, menor é o tempo em que estes pacotes percorrem a rede para alcançar o computador móvel.

7.3.3 Sobrecarga de Mensagens

Nesta seção apresentamos a carga média de mensagens geradas pelos protocolos de *handover* e otimizações. Dependendo da abordagem empregada para tratar o *handover*, cada protocolo e otimização gera um número de mensagens de controle para serem executados. Em particular, essas mensagens têm a finalidade de tratar as atualizações da localização do computador móvel e do caminho de roteamento de pacotes durante o *handover*. Além dessas mensagens de controle, comparamos também o número médio de pacotes redirecionados (de uma estação base a outra), o número médio de pacotes replicados (para um número de estações base) e o número médio

de pacotes retransmitidos quando as técnicas Buffer, Multicast e Retrans, respectivamente, são empregadas.

Na Figura 7.8 apresentamos o número médio de mensagens de controle geradas pelos protocolos de *handover* e otimizações para o caso de *hard handover*. Uma vez que os resultados para *soft handover* são relativamente semelhantes pois a existência de regiões de intersecção não influi nos mesmos, omitiremos esses resultados. Conforme podemos observar, o protocolo de *handover* que gera o menor número de mensagens de controle é o CellularIP, uma vez que após a mensagem Greet enviada pelo computador móvel à estação base, apenas uma mensagem (PathUpdate) é enviada pela estação base e esta é repassada pelos roteadores em direção ao *gateway*, fazendo com que cada um deles faça a atualização necessária.

Os protocolos MobileIPSH e Multicast são os que geraram um maior número de mensagens de controle. No caso do MobileIPSH, as mensagens para criar e atualizar os *forwarding points* a cada migração acabam causando uma grande carga de mensagens na rede. Já no caso do Multicast, são necessárias as mensagens Join e Leave para manter o grupo *multicast* atualizado (essas mensagens são enviadas para cada membro (estação base) do grupo) e também a mensagem Handover para notificar a antiga estação base sobre a migração do computador móvel. Principalmente com a combinação das técnicas Ack+Retrans podemos observar uma alta carga de mensagens, devido às requisições e retransmissões de pacotes na rede.

Na Figura 7.9 apresentamos uma comparação do número médio de pacotes redirecionados, replicados ou retransmitidos de acordo com o tipo de protocolo ou otimização empregada. No gráfico 7.9-(a) podemos observar que o protocolo MobileIPSH faz um pequeno número de redirecionamento de pacotes e esse número é proporcional ao tempo em que a mensagem Update demora para alcançar o *gateway*, a partir do qual pacotes não são mais enviados para a antiga estação base. Nesse sentido, essa estratégia de *forwarding points* é apropriada, em particular, quando a distância (em número de *hops*) entre a estação base e *gateway* é maior (ou, bem maior) do que a distância entre as duas estações base envolvidas, permitindo a criação e atualização de *forwarding points* e o redirecionamento de pacotes enquanto a mensagem de atualização é enviada ao *gateway*.

Nos gráficos 7.9-(b), (c) e (d) temos o número médio de pacotes redirecionados quando é empregado um Buffer e nestes casos, somente para o Multicast, o resultado corresponde ao número de pacotes replicados, uma vez que este protocolo, mesmo quando combinado com Buffer, não requer o redirecionamento de pacotes. Podemos observar nesses gráficos o alto número de pacotes replicados pelo protocolo Multicast uma vez que cada pacote é enviado para todas as estações base vizinhas. Portanto, apesar da baixa perda de pacotes e do baixo atraso, este protocolo produz uma alta carga de mensagens de controle, replicações de pacotes e, em

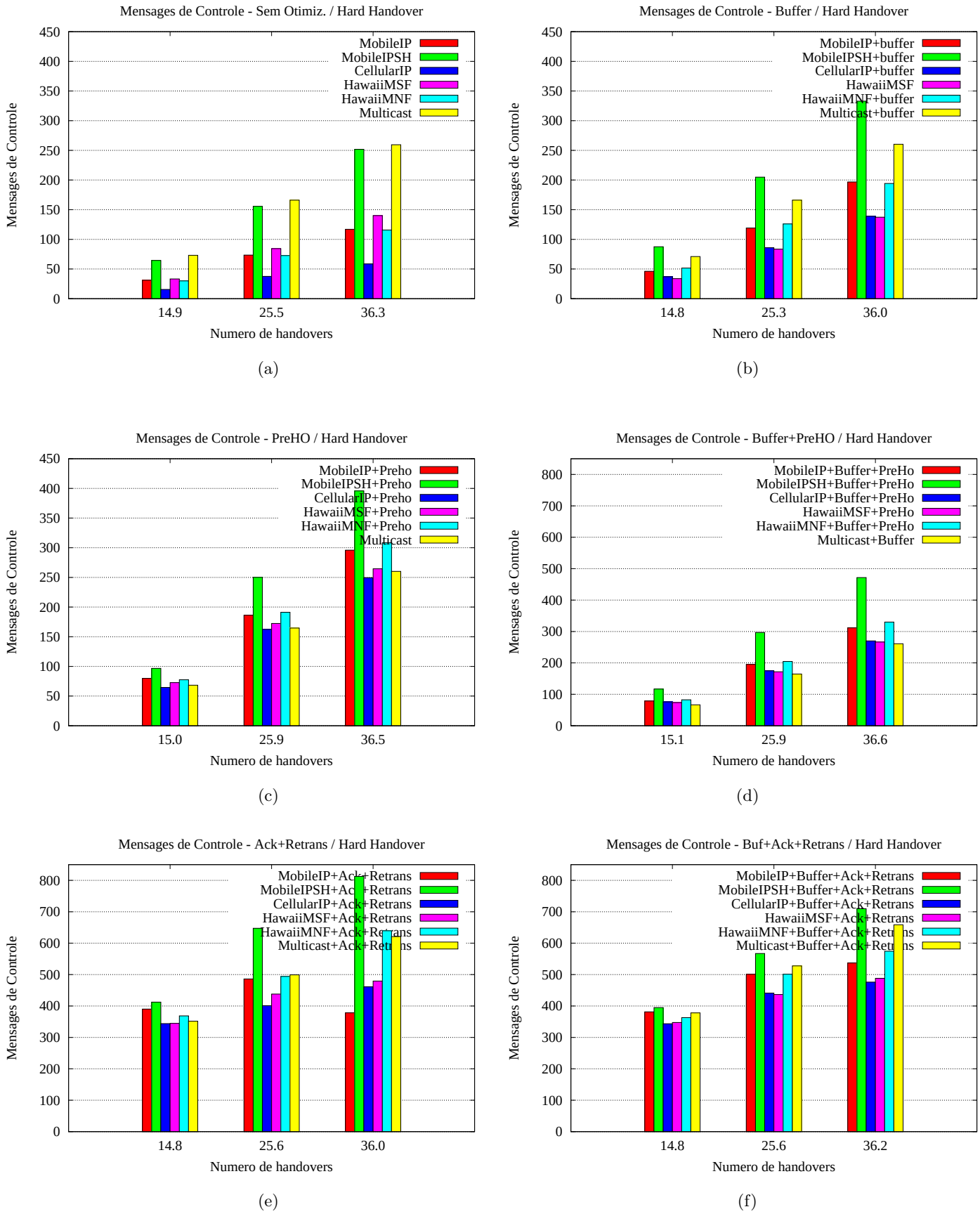


Figura 7.8: Carga média de mensagens de controle (*hard handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f) Buffer+Ack+Retrans

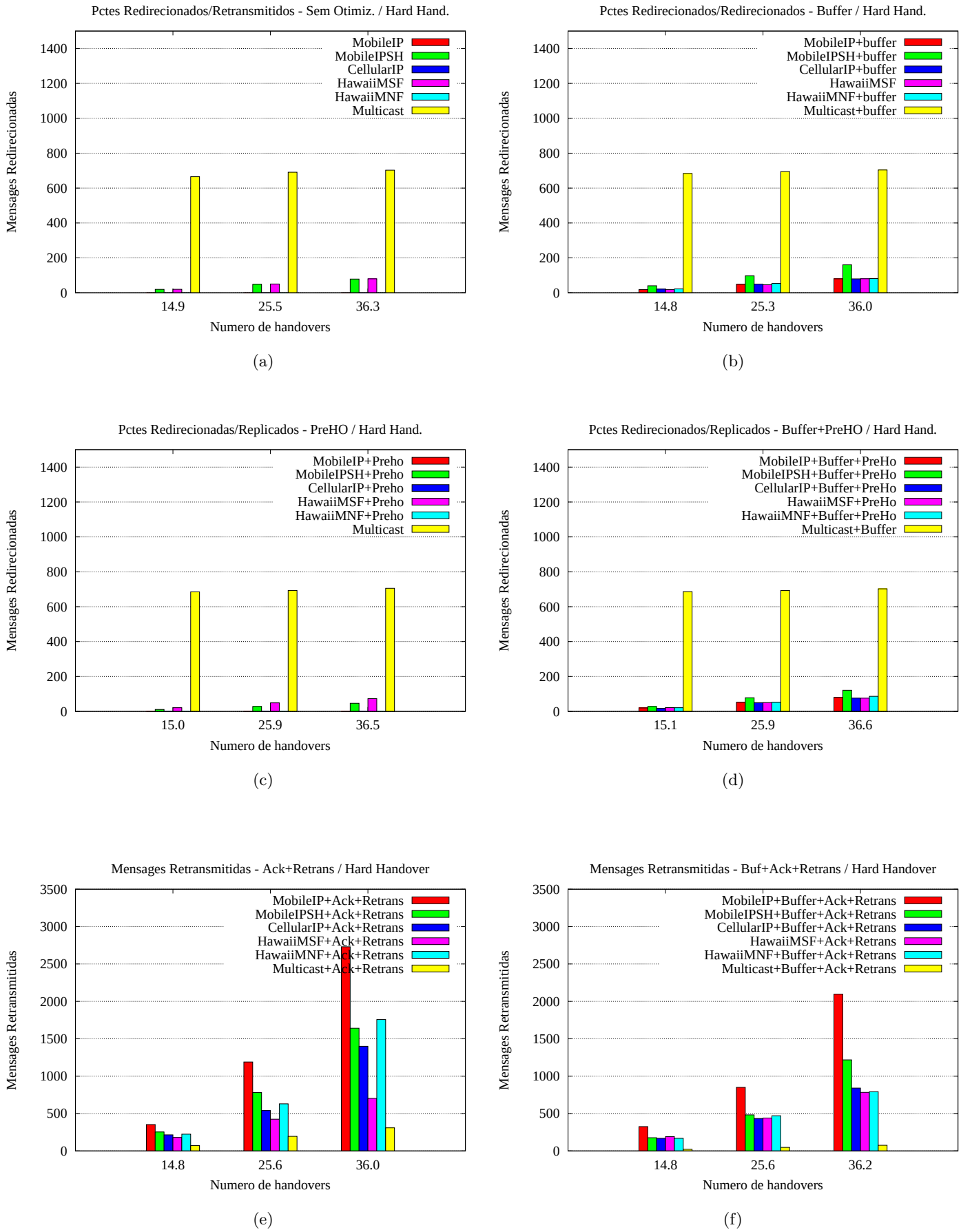


Figura 7.9: Número médio de pacotes redirecionados, replicados e retransmitidos (*hard handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f) Buffer+Ack+Retrans

conseqüência, uma alta utilização de recursos na rede.

A retransmissão de pacotes ocorre apenas quando as técnicas Ack + Retrans são empregadas (gráficos 7.9-(e) e (f)). Basicamente, o número de retransmissões de pacotes depende do número de perdas, quanto mais suceptível a perdas for o protocolo, maior será a necessidade de retransmissão de pacotes.

Dessa forma, além da carga de mensagens de controle gerada pelos protocolos de *handover* deve também ser levado em consideração o número de pacotes redirecionados, replicados ou retransmitidos devido ao emprego de uma ou mais técnicas que também podem causar um grande consumo de recursos.

7.3.4 Pacotes Duplicados e Pacotes Fora de Ordem

Na Figura 7.10 apresentamos o número médio de pacotes duplicados recebidos pelo computador móvel no caso de *soft handover*. Podemos observar que, na maioria dos casos, com exceção dos gráficos 7.10-(c) e (d), o protocolo Multicast ocasionou o maior número de pacotes duplicados, principalmente quando as técnicas Ack+Retrans são combinadas. O uso de Buffer causou um grande número de duplicações devido ao armazenamento de réplicas em cada estação base vizinha (gráfico 7.10-(b)). Cada vez que o computador móvel migra e entra em uma nova célula, a estação base envia todo o conteúdo do *buffer* para o mesmo e possivelmente, muitos pacotes recebidos na antiga célula são recebidos novamente. Em outros experimentos, observamos que reduzindo-se o tamanho do *buffer*, o número de duplicações também se reduz proporcionalmente. Porém, por outro lado, ocasionou um número de perdas maior.

Quando a técnica PreHO é empregada (gráfico 7.10-(c)), o número de pacotes duplicados é alto para a maioria dos protocolos, e isso se deve ao fato de que esta técnica está combinada com a técnica Bicast, ou seja, durante o *handover*, pacotes são replicados para a atual e futura estações base. Quando essa técnica é combinada com um Buffer (gráfico 7.10-(d)), o número de duplicações aumenta ainda mais, uma vez que as réplicas são armazenadas e enviadas somente quando o computador móvel se conecta efetivamente com a nova estação base.

No caso de Ack+Retrans (gráficos 7.10-(e) e (f)), a principal razão para a alta taxa de duplicações, em particular, para o protocolo Multicast, é devido fato de que todo pacote retransmido pelo nó fonte também é replicado para todas as estações base vizinhas, aumentando as chances de receber um mesmo pacote repetidas vezes principalmente quando a taxa de mobilidade é alta.

Na Figura 7.11 apresentamos os resultados para ordenação de pacotes, isto é, o número médio de pacotes fora de ordem recebidos pelo computador móvel durante as simulações. Quando não há o emprego de otimizações (gráfico 7.11-(a)), exceto pelo caso do protocolo HawaiiMSF que

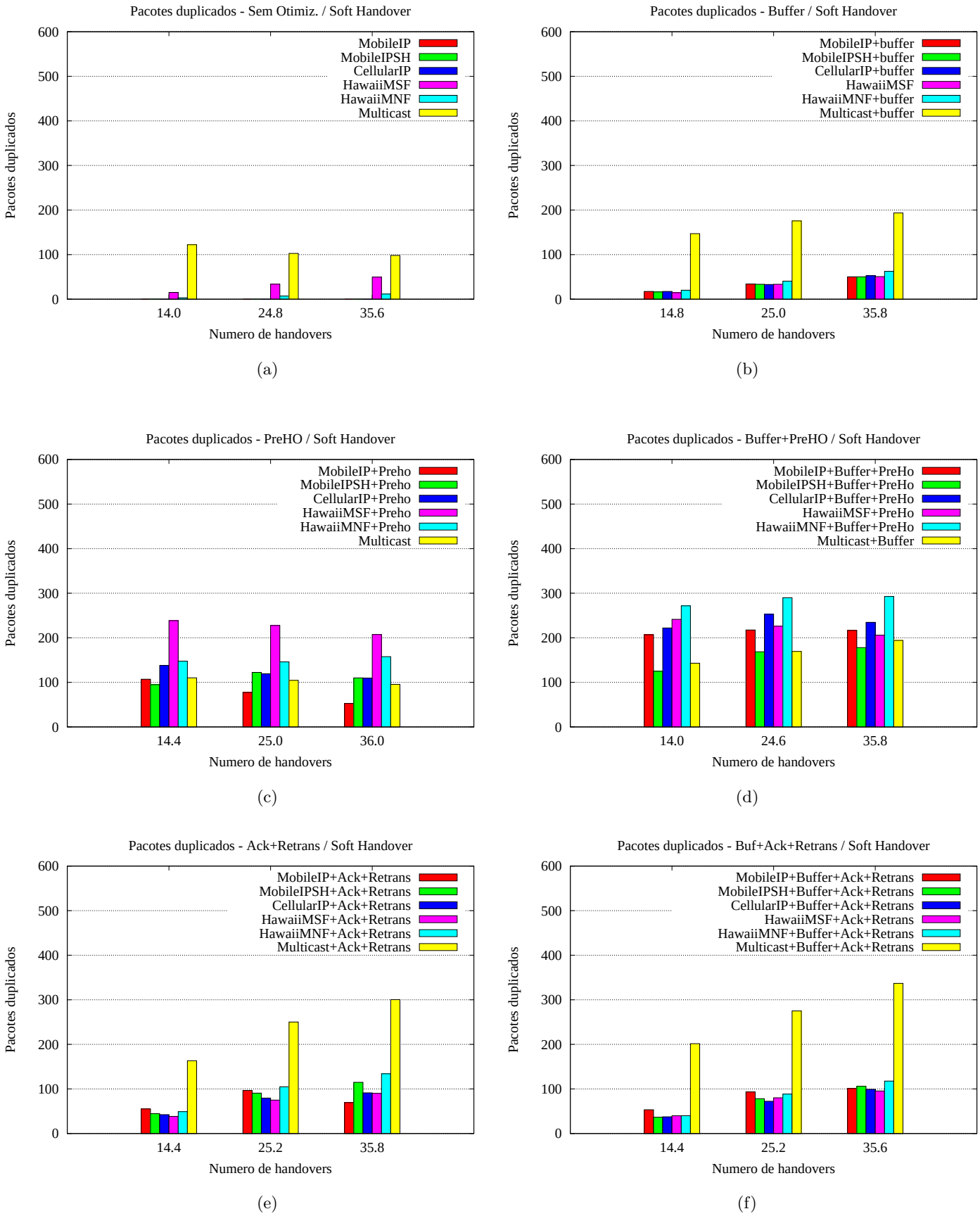


Figura 7.10: Número médio de pacotes duplicados (*soft handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f) Buffer+Ack+Retrans

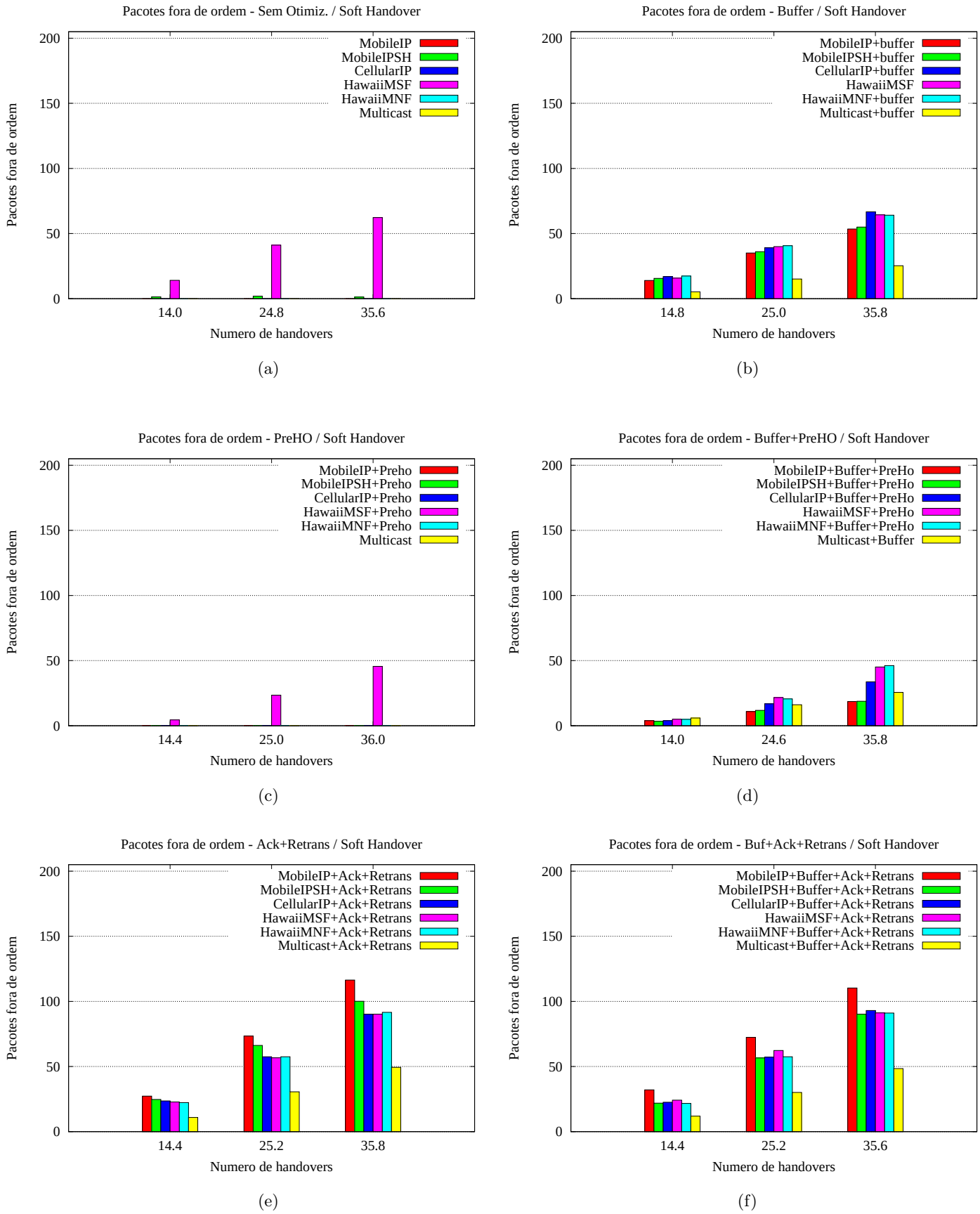


Figura 7.11: Número médio de pacotes fora de ordem (*soft handover*): (a) sem otimização, (b) Buffer, (c) PreHO, (d) Buffer+PreHO, (e) Ack+Retrans e (f) Buffer+Ack+Retrans

utiliza *buffer*, os protocolos não causam desordenação de pacotes uma vez que os pacotes são enviados na mesma ordem em que são recebidos pelas estações base. Também é o caso quando apenas a técnica PreHO é empregada (gráfico 7.11-(c)).

Com o emprego do Buffer (gráfico 7.11-(b)), pacotes armazenados na antiga estação base são re-enviados para a nova estação e, durante esse re-envio, novos pacotes são direcionados à nova estação base e transmitidos ao computador móvel independentemente dos pacotes sendo redirecionados. No caso do protocolo Multicast, em que não há o redirecionamento de pacotes, o número de pacotes fora de ordem se deve, basicamente, às diferenças de tempo (simulado) em que os pacotes são recebidos pelas estações base (que depende das distâncias até o *gateway*).

O principal causador de pacotes fora de ordem é quando há retransmissões de pacotes (gráficos 7.11-(e) e (f)), principalmente para o Mobile IP, mas, particularmente, esse resultado é proporcional ao número de pacotes retransmitidos durante as simulações.

7.3.5 Comparação de Topologias

Em um protocolo de *handover*, uma das tarefas mais importantes é a atualização na rede a fim de permitir que os pacotes destinados ao computador móvel sejam direcionados corretamente para a sua nova localização. O desempenho de um protocolo de *handover* pode ser afetado consideravelmente dependendo da forma em que esta operação é executada e da topologia da rede fixa por onde atravessam as mensagens de atualização. A fim de comparar o desempenho em diferentes topologias organizamos os protocolos de acordo com o "Ponto de Atualização" (PA) na rede. O ponto de atualização é o elemento de rede que mantém a informação de localização do computador móvel no domínio e é aquele que deve ser notificado a cada migração para a atualização da localização. De acordo com o PA, identificamos três categorias de protocolos de *handover* dentre as quais os protocolos simulados neste capítulo se enquadram:

- Categoria 1 (GwPA), quando a informação de localização de um computador móvel é mantida de forma centralizada no *gateway* e a mensagem de atualização deve ser enviada até o mesmo. Quanto maior a distância (em número de *hops*) entre uma estação base e o *gateway*, maior é o tempo para a mensagem de atualização alcançar o mesmo. Dentre os protocolos neste categoria podemos citar: MobileIP e MobileIPSH.

- Categoria 2 (RouterPA), nessa categoria a informação de localização é mantida de maneira distribuída nos roteadores no caminho da estação base ao *gateway*. O PA, neste caso, corresponde ao roteador na intersecção entre o antigo e novo caminho de roteamento de pacotes (*crossover router*). Exemplos de protocolos nessa categoria são: CellularIP, HawaiiMSF e HawaiiMNF.

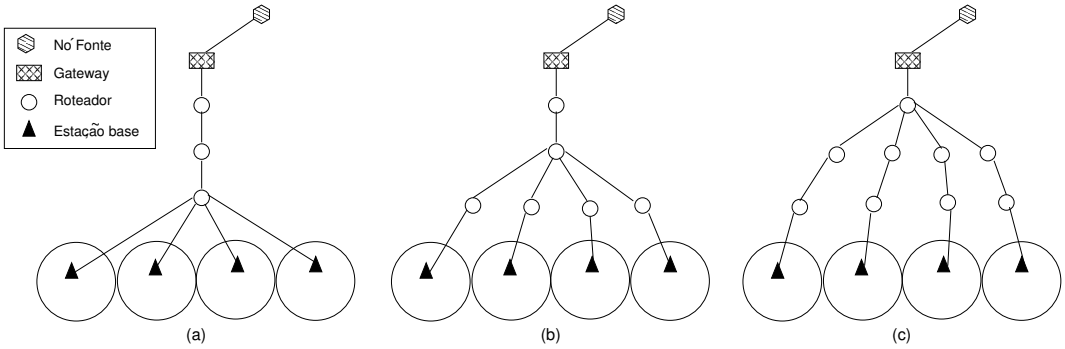


Figura 7.12: Topologias de rede utilizadas nas simulações com distâncias EB-CR iguais a: (a) 1, (b) 2 e (c) 3

- Categoria 3 (McastPA), neste caso, a informação de localização do computador móvel está nas estações base. Exemplo de protocolo: Multicast, no qual a forma de transmissão é baseada em *multicast* e o PA corresponde a um conjunto de estações base que fazem parte do grupo *multicast*.

Como representantes dessas categorias de protocolos utilizamos para a Categoria 1 o MobileIP, para a Categoria 2, o CellularIP e para a Categoria 3, o protocolo Multicast. A escolha do MobileIP e CellularIP se deve ao fato de que estes são os protocolos mais simples dentre os protocolos dessas categorias.

Na Figura 7.12 temos as três topologias de rede empregadas para as simulações, a principal diferença entre estas está nas distâncias entre as estações base e os *crossover routers* (CR). Utilizamos distâncias iguais a 1, 2 e 3 para as topologias (a), (b) e (c), respectivamente.

Nos gráficos a seguir, comparamos os resultados para os seguintes parâmetros: número médio de pacotes perdidos e atraso médio na entrega de pacotes para taxas de mobilidade baixa e alta. Estas taxas de mobilidade correspondem às mesmas freqüências de geração de eventos de migração do computador móvel empregadas nas seções anteriores. No eixo x temos as três topologias representadas pelas distâncias entre estação base e *crossover router* (EB-CR). Na Figura 7.13 e 7.14 temos os resultados para o número de pacotes perdidos. Conforme podemos observar, quando não há o emprego de nenhuma otimização (gráficos 7.13-(a) e (b)), o número de perdas no caso da Categoria 1 (Mobile IP) se mantiveram estáveis, uma vez que a atualização da localização do computador móvel independe da distância do *crossover router*, o PA é o *gateway*. Porém, no caso da Categoria 2 (CellularIP) podemos observar um gradativo aumento do número de perdas de acordo com o aumento da distância EB-CR, principalmente no caso em que a taxa de migração é alta. Na Categoria 2, o PA é o *crossover router*, que é o elemento que ao identificar uma migração, começa a desviar os pacotes para a nova estação base. Quanto mais próximo da

estação base estiver o *crossover router*, mais rápida será a atualização da localização (uma vez que a mensagem de atualização é enviada a partir da nova estação base em direção ao *gateway*) e, com isso, menor será o número de pacotes perdidos. A Categoria 3 (Multicast) apresentou o menor número de perdas e também se manteve estável com relação às diferentes topologias, uma vez que o PA está nas estações base.

Com o uso de Buffer (gráficos 7.13-(c) e (d)), podemos observar uma diferença maior com relação ao número de perdas para o MobileIP e CellularIP quando a taxa de mobilidade é alta. Isso porque a distância EB-CR também influencia no redirecionamento de pacotes, uma vez que estes devem ser encaminhados pela rede fixa, passando pelo *crossover router*. Quando a distância EB-CR é grande, o redirecionamento se torna inviável caso a freqüência de migração do computador móvel seja alta pois os pacotes são sucessivamente redirecionados a fim de alcançar o mesmo. Quando combinamos com o módulo PreHO (gráficos 7.13-(e), (f) e 7.14-(a), (b)), podemos observar um comportamento semelhante aos anteriores, porém, com menores perdas. Os gráficos 7.14-(c), (d), (e) e (f) mostram os resultados quando os módulos Ack+Retrans são empregados. Em particular para alta taxa de mobilidade, podemos observar uma considerável diferença no número de perdas ao comparar as três categorias de protocolos.

Nas Figuras 7.15 e 7.16 comparamos os resultados para o atraso médio para as três topologias de rede. Conforme podemos observar, algumas variações no atraso ocorrem quando é empregado o módulo Buffer (gráficos 7.15-(c) e (d)), porém, a influência da topologia é de fato notória quando são empregadas as técnicas Ack+Retrans e, principalmente, quando a taxa de mobilidade é alta (gráficos 7.16-(c), (d), (e) e (f)). Em particular, os resultados para a Categoria 3 (Multicast) se mantiveram praticamente estáveis com relação ao atraso.

7.4 Algumas Regras Empíricas para a Seleção de Módulos Canônicos

A partir das experiências obtidas com as simulações, derivamos algumas regras empíricas para a seleção de módulos canônicos e as organizamos de acordo com os requisitos de QoS que consideramos nas simulações, conforme listamos a seguir:

Pacotes Perdidos - baixa taxa de mobilidade

- Os esquemas de *handover* baseados no HawaiiMSF e Multicast são apropriados quando a taxa de mobilidade é baixa pois causam uma baixa perda de pacotes (em torno de 5%) mesmo sem o emprego de nenhuma técnica de otimização [Figura 7.2-(a)].

- O uso de Buffer permite reduzir pela metade o número de pacotes perdidos para todos os protocolos quando comparados com os resultados obtidos dos protocolos sem otimizações

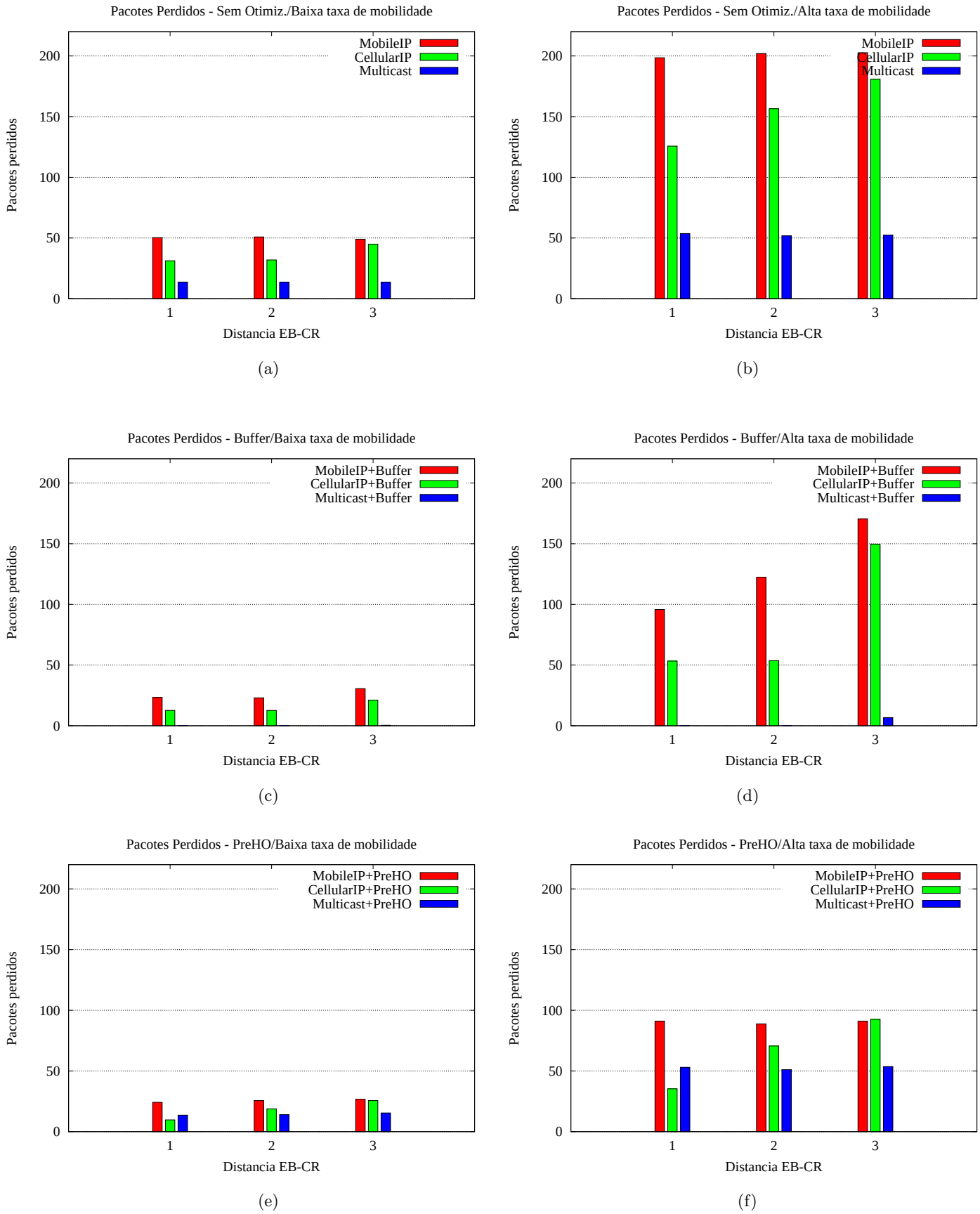


Figura 7.13: Comparação de topologias (perda de pacotes, *hard handover*): (a) e (b) sem otimização, (c) e (d) Buffer, (e) e (f) PreHO

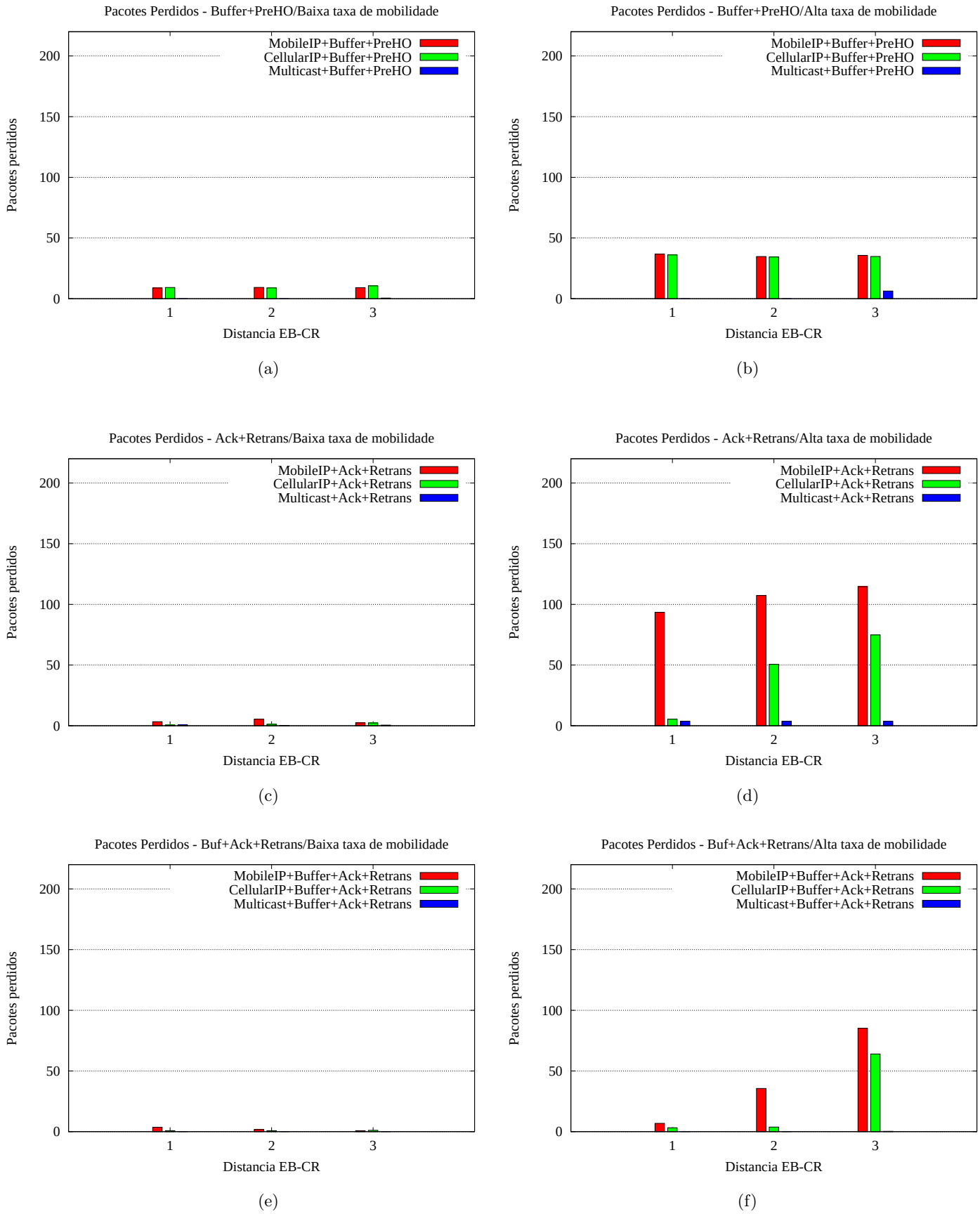


Figura 7.14: Comparação de topologias (perda de pacotes, *hard handover*): (a) e (b) Buffer+PreHO, (c) e (d) Ack+Retrans, (e) e (f) Buffer+Ack+Retrans

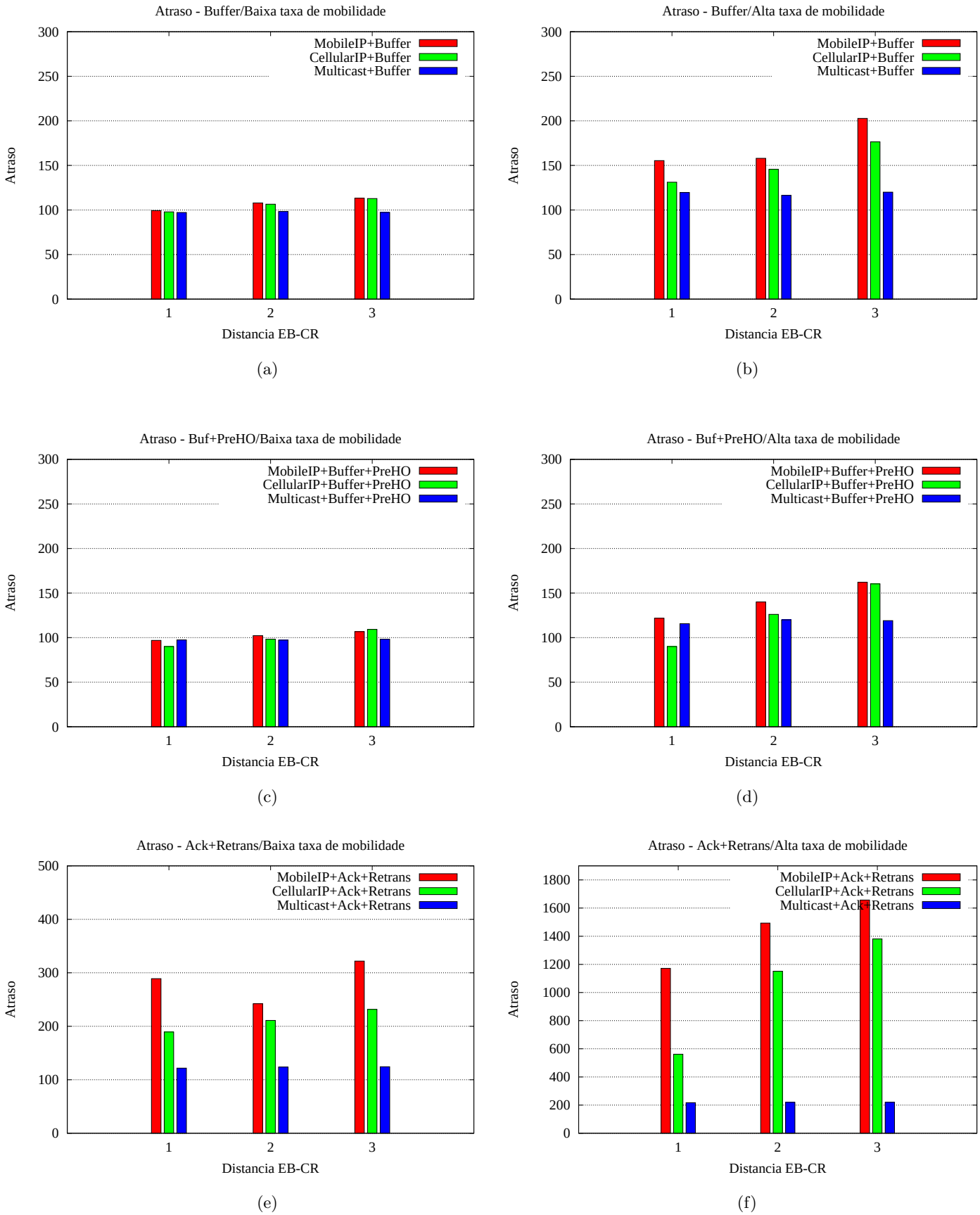


Figura 7.15: Comparação de topologias (atraso médio (em UTS), *hard handover*): (a) e (b) Buffer, (c) e (d) Buffer+PreHO, (e) e (f) Ack+Retrans

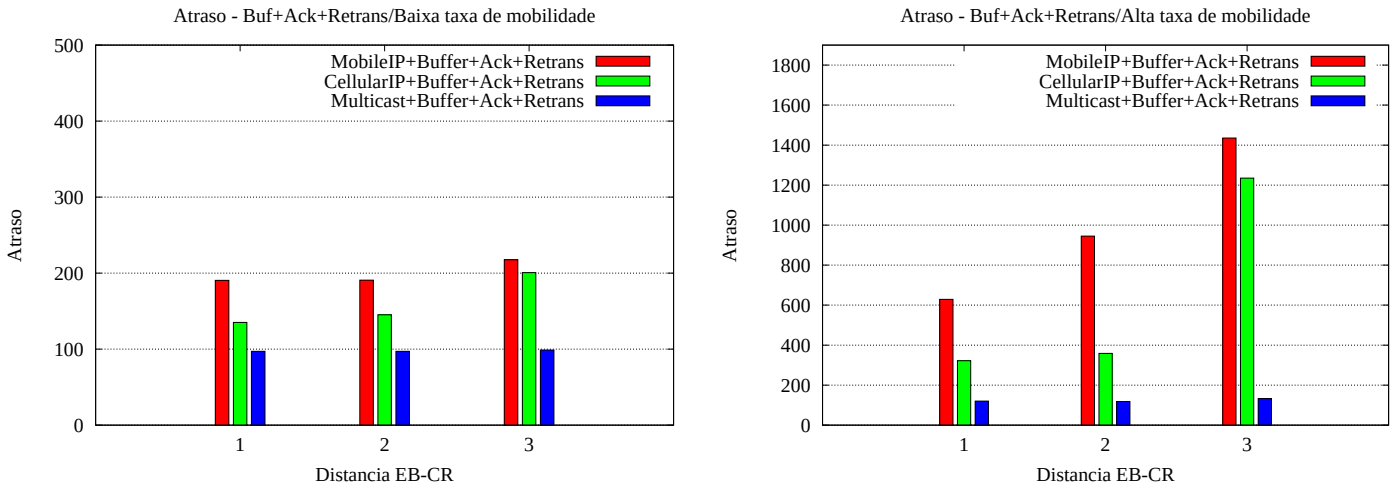


Figura 7.16: Comparação de topologias (atraso médio (em UTS), *hard handover*): (a) e (b) Buffer+Ack+Retrans

(em torno de 10% de perdas para o MobileIP, e 5% para os outros protocolos, com exceção do Multicast, que obteve perda zero) [Figura 7.2-(b)].

- A combinação Buffer+PreHO se mostrou benéfica para todos os protocolos, causando uma taxa média de perdas em torno de 5% (inclusive para o MobileIP), com exceção do Multicast, que obteve perda zero [Figura 7.2-(d)].

- Quando existe a possibilidade de se empregar a técnica de *soft handover*, por exemplo, em redes baseadas em CDMA, a quantidade de pacotes perdidos é reduzida significantemente para todas as combinações de técnicas, em particular, com a combinação Buffer+PreHO que permite uma taxa de perdas menor do que 3% para todos os protocolos, sendo, portanto, apropriada para aplicações que requerem uma baixa taxa de perdas [Figura 7.3-(d)].

- Quando a tolerância a perdas é extremamente baixa, e se não há exigências com relação a outros requisitos de QoS, como atraso ou sobrecarga de mensagens, uma possibilidade seria a combinação Ack+Retrans ou Buffer+Ack+Retrans, que possibilitou perdas praticamente nulas [Figuras 7.2-(e), (f) e 7.3-(e), (f)].

Pacotes Perdidos - alta taxa de mobilidade

- A combinação Buffer+Multicast apresentou um melhor desempenho quando a freqüência de migração é alta, com perdas igual a zero, para *hard handover* [Figuras 7.2-(b) e (d)].

- Para *soft handover*, a combinação Buffer+PreHO apresentou uma taxa de perdas de aproximadamente 10% para todos os protocolos [Figura 7.3-(d)]. E, para qualquer outra combinação de técnicas empregando-se *soft handover*, houve uma redução de mais de 50% de perdas com relação aos resultados sem o emprego de técnicas de otimização [Figuras 7.3-(b), (c), (d), (e) e (f) com relação a 7.3-(a)].

- As combinações Ack+Retrans ou Buffer+Ack+Retrans (com exceção dos protocolos baseados no MobileIP), apresentaram a menor taxa de perdas com relação às outras técnicas, porém devem ser empregadas com cautela, principalmente quando a freqüência de mobilidade é alta, devido ao grande aumento do atraso na entrega e a excessiva carga de mensagens [Figuras 7.2-(e), (f) e 7.3-(e), (f)].

- Quando a distância média (em número de *hops*) entre estações base e *gateway* (EB-GW) é maior ou, bem maior do que a distância média entre estações base e *crossover router* (EB-CR), protocolos em que a atualização de localização do computador móvel é feita nos CRs (por exemplo, CellularIP e Hawaii), são mais eficientes e causam uma menor taxa de perdas com relação aos protocolos que fazem a atualização no *gateway* (por exemplo, MobileIP) [Figuras 7.13 e 7.14].

Atraso

- Quando não são empregadas técnicas de otimização, os protocolos MobileIP, CellularIP e HawaiiMNF não causam nenhum atraso adicional na entrega de pacotes (o atraso mínimo que estamos considerando é o atraso relativo ao canal de comunicação sem fio (igual a 90 UTS, Seção 7.3.2) [Figuras 7.4-(a) e 7.5-(a)].

- A combinação com a técnica Buffer causa um aumento no atraso em cerca de 10% quando a taxa de mobilidade é baixa e de até 50% quando esta é alta, com relação ao atraso mínimo [Figuras 7.4-(b) e 7.5-(b)].

- A melhor opção quando há fortes exigências com relação ao atraso é a combinação da técnica PreHO, que não causa praticamente nenhum atraso adicional independentemente da taxa de mobilidade, para todos os protocolos, com exceção do HawaiiMSF e Multicast [Figuras 7.4-(c) e 7.5-(c)].

- As combinações Ack+Retrans e Buffer+Ack+Retrans devem ser evitadas quando o atraso é um requisito importante, pois causam um atraso excessivamente alto, principalmente para o protocolo MobileIP. Para *soft handover*, esse atraso foi um pouco menor, porém, ainda assim possivelmente inviável para muitas aplicações [Figuras 7.4-(e), (f) e 7.5-(e), (f)].

- O aumento da distância média (em número de *hops*) entre estações base e *crossover routers* pode causar um aumento no atraso principalmente quando a taxa de mobilidade é alta e para algumas técnicas como Buffer e Ack+Retrans, e em particular para protocolos baseados no MobileIP [Figuras 7.15 e 7.16].

Variação do Atraso

- Quando nenhuma técnica de otimização é empregada, os protocolos CellularIP e HawaiiMNF não causam nenhuma variação do atraso [Figuras 7.6-(a) e 7.7-(a)].

- O uso de Buffer não é aconselhado quando há uma forte exigência com relação à variação do atraso, para todos os tipos de protocolos e para qualquer taxa de mobilidade [Figuras 7.6-(b) e 7.7-(b)]. A combinação Buffer+PreHO também causa uma grande variação do atraso e portanto, da mesma forma, não é aconselhável o seu emprego [Figuras 7.6-(d) e 7.7-(d)].

- Com exceção do protocolo HawaiiMSF, a técnica PreHO (sem a combinação com Buffer) apresenta-se como a melhor opção para uma baixa variação do atraso, para todos os tipos de protocolos, independentemente da freqüência de migração e, em particular quando esta freqüência é baixa e é empregado *soft handover* [Figuras 7.6-(c) e 7.7-(c)].

- A combinação de técnicas Ack+Retrans apresentou um efeito extremamente maléfico para a variação do atraso, com valores altíssimos, principalmente para o MobileIP. Com *soft handover*, essa variação reduziu-se pela metade, porém, ainda assim essa combinação é completamente inviável [Figuras 7.6-(e) e 7.7-(e)].

Sobrecarga de mensagens

- O CellularIP é o tipo de protocolo que gera a menor carga de mensagens de controle para tratar um *handover*. Em seguida, são os protocolos do tipo Hawaii [Figura 7.8-(a)].

- O protocolo do tipo Multicast gera uma excessiva quantidade de mensagens de controle de *handover* e também um número extremamente alto de pacotes da aplicação replicados, causando uma grande utilização de recursos e sobrecarga na rede [Figuras 7.8 e 7.9-(a), (b), (c) e (d)].

- O uso da técnica Buffer causa um aumento da carga de mensagens de controle principalmente quando a taxa de mobilidade é alta e, em particular, para os protocolos MobileIPSH e Multicast [Figuras 7.8-(b) e (d)]. Porém, o número de pacotes redirecionados é bem menor do que o número de pacotes replicados gerados pelo protocolo Multicast (em torno de 10%) [Figuras 7.8 e 7.9-(a), (b), (c) e (d)].

- A técnica de retranmissão de pacotes causa uma altíssima carga de mensagens de controle, principalmente quando a taxa de mobilidade é alta e, o mesmo ocorre quando é combinado com Buffer [Figuras 7.8-(e) e (f)]. Além disso, dependendo do protocolo, a carga de pacotes retransmitidos na rede também é extremamente alta, em particular, para o MobileIP [Figuras 7.9-(e) e (f)].

Duplicação de pacotes e Ordenação

- Quando um dos requisitos da aplicação é a baixo número de pacotes duplicados, o protocolo Multicast não é viável pois gera um grande número de duplicações, principalmente quando combinado com outra técnica, como Buffer e ou Ack+Retrans, com uma taxa de duplicações em torno de 40% (sem otimização) e 80% (no caso da combinação Buffer+Ack+Retrans) [Figuras 7.10-(a), (b), (e) e (f)].

- Os esquemas de *handover* que não geram duplicações são aqueles baseados no MobileIP e CellularIP, quando empregados sem a combinação qualquer de técnica [Figura 7.10-(a)].

- O uso de Buffer e PreHO também geram duplicações, em particular quando essas técnicas são empregadas em conjunto e, principalmente, para o protocolo HawaiiMNF. Quando a taxa de mobilidade é baixa, o uso de Buffer pode ser aconselhado (para tratar outro requisito, por exemplo, perda de pacotes) pois gera um pequeno número de duplicações [Figuras 7.10-(b), (c) e (d)].

- O menor número de pacotes fora de ordem ocorre quando os protocolos não são combinados a técnicas de otimização, com exceção da técnica PreHO, que não causou a desordenação de pacotes, exceto apenas para o protocolo HawaiiMSF [Figuras 7.11-(a) e (c)].

- A técnica de Buffer acarreta em um número de pacotes fora de ordem, porém, esse número é baixo quando a taxa de mobilidade é baixa (abaixo de 3%) e entre 10 e 20% quando a taxa de mobilidade é alta [Figura 7.11-(b)].

- Além de outros problemas, as técnicas Ack+Retrans causam também um alto percentual de pacotes fora de ordem (acima de 40% para o MobileIP), e com a menor taxa para o protocolo Multicast, em torno de 20%, quando a taxa de mobilidade é alta [Figura 7.11-(e) e (f)].

7.5 Conclusões Finais

Embora as nossas simulações estejam limitadas a um simulador de protocolos (MobiCS)

e a determinadas configurações como topologia de rede e parâmetros de simulação, pudemos verificar certas tendências no comportamento dos protocolos simulados.

Na literatura existem alguns trabalhos que apresentam resultados de simulações e comparações entre protocolos de micro-mobilidade. Por exemplo, em [37] temos uma comparação entre os protocolos Cellular IP (*semi-soft handover*), Hawaii MSF e M&M (as simulações foram realizadas no ns-2). Conforme também constatamos em nossos experimentos, o M&M (baseado em *multicast*) foi o que obteve o menor número de perdas seguido pelo Hawaii MSF e, com relação ao atraso, o M&M e Cellular IP tiveram o melhor desempenho, o Hawaii MSF por empregar um *buffer*, causou um certo atraso e, principalmente, quando a distância percorrida para redirecionar os pacotes no *buffer* foi maior. Esses resultados também mostraram o número de pacotes fora de ordem, que também foi maior para o Hawaii MSF devido ao uso do *buffer*.

Em [53] são comparados o número de pacotes perdidos para o Mobile IP Hierárquico, Cellular IP (*hard handover*) e Hawaii MSF, para algumas topologias de rede com relação às distâncias entre *crossover router* e estações base e entre *gateway* e estações base. Também de acordo com os nossos resultados (embora tenhamos tratado do Mobile IP básico), foi mostrado que o Mobile IP Hierárquico teve o pior desempenho, com o maior número de perdas. Além disso, quanto mais próximo o *crossover router* estiver da estação base, melhor é o desempenho para os protocolos Cellular IP e Hawaii, o que também foi constatado em nossos experimentos.

O principal diferencial de nossos experimentos com relação aos outros está, basicamente, na capacidade de simular e comparar não apenas o desempenho dos protocolos puros, conforme são encontrados na literatura, mas também diferentes combinações desses protocolos com técnicas de otimização.

Para de fato alcançar *handover* suave talvez seria necessário considerar muitos outros fatores além daqueles que tratamos nesta tese, porém, dentro do escopo que estamos considerando, para aplicações com requisitos particulares, com características específicas de rede e usuário, acreditamos que as heurísticas a que chegamos podem nos dar alguns indicativos para se ter *handover* suave.

Capítulo 8

Conclusão

Nesta tese apresentamos uma proposta de um *framework* para a prototipação e a avaliação de protocolos de *handover* para micro-mobilidade em redes móveis infra-estruturadas. Um dos principais problemas dos protocolos baseados em IP para estes tipos de rede é o gerenciamento eficiente de *handover*, de forma a minimizar a latência da comunicação e a perda de dados durante a migração de um computador móvel de uma célula para outra.

Diversas abordagens e mecanismos têm sido propostos para lidar com determinados aspectos relacionados à provisão *handover* suave e foram implementados em protocolos para micro-mobilidade descritos na literatura. Porém, visando atender determinados requisitos das aplicações, e geralmente assumindo características ou topologias específicas da rede móvel, cada um destes protocolos implementa algumas dessas técnicas, porém, de uma forma particular e com alto grau de acoplamento. Além disso, o bom desempenho destes protocolos depende muito do tipo de fluxo de dados da aplicação e do perfil de mobilidade dos usuários móveis, tornando-os bastante específicos. Devido a todos estes fatores, geralmente torna-se muito difícil analisar e comparar estes protocolos.

Assim, verificamos que seria útil tentar identificar e implementar de forma modular cada uma dessas técnicas a fim de alcançar uma melhor compreensão de seu papel e de sua influência no desempenho de protocolos de micro-mobilidade. Para isso, desenvolvemos um *framework* que permite não apenas a prototipação dos principais protocolos de *handover* citados na literatura, mas também experimentar diferentes combinações das mesmas, a fim de projetar protocolos adaptados e adequados para determinadas aplicações e tipos de redes móveis. O *framework*, denominado HOPF, possibilita a prototipação, simulação e comparação de protocolos de *handover* para micro-mobilidade a partir da combinação de módulos canônicos independentes. Para a implementação do *framework* e a simulação dos protocolos de *handover* gerados utilizamos o Simulador de Protocolos Distribuídos MobiCS [16, 56].

A partir dos resultados de simulações para diferentes cenários, identificamos algumas in-

fluências que alguns módulos (ou combinações de módulos) têm sobre a qualidade do fluxo de dados transmitidos da rede para computadores móveis (com relação ao número de pacotes perdidos, atraso, variação do atraso, etc.) e a partir disso foi possível enunciar algumas heurísticas que podem ser utilizadas para direcionar a escolha e composição de módulos canônicos para a geração de protocolos de *handover* para diferentes requisitos de QoS das aplicações.

Implementamos e testamos os seguintes esquemas de *handover*: (1) Mobile IP-like, que é uma adaptação/simplificação do protocolo Mobile IP para um domínio; (2) Mobile IP Smooth, que estende o Mobile IP acrescentando a técnica de *forwarding points*; (3) Cellular IP, (4) Hawaii MSF, (5) Hawaii MNF que possuem uma forma semelhante para a manutenção da informação de localização de forma distribuída (nos roteadores), porém, possuem distintas técnicas para tratar a atualização na rede após um *handover*, e (6) Multicast, um protocolo baseado na difusão de mensagens. Todos esses protocolos foram testados em sua forma original e também combinados com outros módulos canônicos que implementam algumas técnicas para a otimização do desempenho.

A partir dos resultados obtidos das simulações, identificamos algumas regras empíricas (heurísticas) para a seleção de módulos canônicos e, resumidamente, podemos concluir que: (a) dentre os esquemas de *handover* simulados e sem o emprego de qualquer otimização, o Multicast apresentou um melhor desempenho com relação à perda de pacotes, porém, acarretou em uma grande sobrecarga de pacotes replicados na rede e de pacotes duplicados no computador móvel; (b) a técnica de *bufferização* reduziu consideravelmente o número de perdas em todos os esquemas de *handover*, em particular, para o Multicast em que a perda foi nula, mesmo para altas taxas de migração (*hard handover*). O principal problema com essa técnica é o aumento considerável no atraso e variação do atraso, principalmente para os esquemas Mobile IP-like e Mobile IP Smooth; (c) a técnica de *Pré-handover* também permite reduzir as perdas, mas principalmente, com um baixo atraso e variação do atraso; (d) a técnica *Pré-handover* combinada com *buferização* permite uma maior redução e perdas se comparada com as outras técnicas de otimização para todos os protocolos simulados; (e) esquemas de *handover* como o Cellular IP e Hawaii causam a menor sobrecarga de mensagens de controle e também um pequeno número de pacotes fora de ordem quando não combinados a outras técnicas.

Ressaltamos que as heurísticas são oriundas de nossa experiência particular e que foram formuladas a partir de um conjunto limitado de simulações. A fim de se realmente confirmar essas heurísticas, seria necessário um conjunto muito mais amplo e detalhado de simulações.

Como uma das principais contribuições, acreditamos que o HOPF possa ser utilizado por um projetista/desenvolvedor de protocolos como uma ferramenta para a comparação qualitativa de protocolos de *handover* e para o estudo e a experimentação de protocolos de *handover*

customizados para micro-mobilidade.

Como trabalhos futuros podemos vislumbrar os seguintos desafios: (a) estender e aprimorar os modelos de rede, de fluxo de comunicação (considerando também os pacotes transmitidos pelo computador móvel), e de mobilidade dos usuários; (b) transformar as heurísticas em regras formais de seleção e composição de módulos canônicos a partir dos parâmetros de QoS, tipo de rede e perfil de mobilidade; (c) automatizar a composição de protocolos de *handover*; e (d) criar mecanismos e estratégias para a composição e configuração dinâmica de protocolos a depender do estado de congestionamento da rede, da taxa de transmissão de pacotes, e da freqüência média de migrações de cada computador móvel.

Médias e Intervalos de Confiança

A seguir são apresentadas tabelas com os valores das médias obtidas a partir das simulações e que foram apresentadas nos gráficos da Seção 7. Para cada tabela com as médias, há uma tabela correspondente contendo os respectivos intervalos de confiança.

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
55.9	35.3	34.6	13.5	34.7	13.8
132.4	92.2	82.5	34.0	84.5	33.5
205.2	157.2	132.3	53.5	132.6	52.2

Tabela A.1: Médias referentes ao gráfico da Figura 7.2-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[49.11, 62.68]	[31.13, 39.46]	[30.13, 39.06]	[11.48, 15.51]	[30.63, 38.76]	[12.33, 15.26]
[124.21, 140.58]	[86.22, 98.17]	[76.87, 88.12]	[31.50, 36.49]	[78.90, 90.0]	[31.21, 35.78]
[200.86, 209.53]	[151.91, 162.48]	[127.62, 136.97]	[52.0, 54.96]	[128.53, 136.66]	[50.45, 53.94]

Tabela A.2: Intervalos de confiança para as médias da Tabela A.1

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
29.1	15.2	14.2	12.8	14.2	0.0
75.1	42.8	32.9	32.0	35.9	0.0
119.1	80.3	54.4	53.6	54.1	0.0

Tabela A.3: Médias referentes ao gráfico da Figura 7.2-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[24.83, 33.36]	[12.74, 17.65]	[12.28, 16.11]	[11.49, 14.13]	[12.35, 16.04]	[0.0, 0.0]
[69.84, 80.35]	[39.59, 46.00]	[31.02, 34.77]	[29.78, 34.21]	[33.78, 38.01]	[0.0, 0.0]
[114.90, 123.29]	[77.57, 83.02]	[52.38, 56.41]	[51.58, 55.61]	[52.08, 56.11]	[0.0, 0.0]

Tabela A.4: Intervalos de confiança para as médias da Tabela A.3

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
30.6	25.5	19.6	13.2	22.0	14.2
70.3	63.9	43.5	40.0	55.3	33.6
112.8	103.3	65.2	68.0	89.7	50.8

Tabela A.5: Médias referentes ao gráfico da Figura 7.2-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[27.15, 34.04]	[22.90, 28.09]	[17.14, 22.05]	[11.11, 15.28]	[19.13, 24.86]	[12.32, 16.07]
[66.00, 74.59]	[59.83, 67.96]	[40.49, 46.50]	[36.17, 43.82]	[51.20, 59.39]	[31.75, 35.44]
[108.63, 116.96]	[98.83, 107.76]	[62.53, 67.86]	[63.35, 72.64]	[86.97, 92.42]	[48.99, 52.60]

Tabela A.6: Intervalos de confiança para as médias da Tabela A.5

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
13.2	12.0	11.6	14.1	12.6	0.0
39.5	36.1	37.8	42.1	36.4	0.0
66.1	60.3	60.8	71.4	69.1	0.0

Tabela A.7: Médias referentes ao gráfico da Figura 7.2-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[11.56, 14.83]	[10.29, 13.70]	[9.68, 13.51]	[11.81, 16.38]	[10.75, 14.44]	[0.0, 0.0]
[35.88, 43.11]	33.60, 38.59]	[34.83, 40.76]	[38.21, 45.98]	[32.85, 39.94]	[0.0, 0.0]
[62.65, 69.54]	[57.39, 63.20]	[58.71, 62.88]	[67.44, 75.35]	[66.06, 72.13]	[0.0, 0.0]

Tabela A.8: Intervalos de confiança para as médias da Tabela A.7

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
4.2	1.0	0.4	0.9	0.5	0.3
26.4	4.8	4.9	1.7	5.3	1.0
67.9	49.5	25.0	2.7	13.9	2.5

Tabela A.9: Médias referentes ao gráfico da Figura 7.2-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[2.18, 6.21]	[0.07, 1.92]	[0.05, 0.74]	[0.52, 1.27]	[0.12, 0.87]	[0.02, 0.57]
[17.49, 35.30]	[2.47, 7.12]	[2.88, 6.91]	[1.15, 2.24]	[2.50, 8.09]	[0.31, 1.68]
[41.55, 94.24]	[31.21, 67.78]	[13.91, 36.08]	[1.77, 3.62]	[8.71, 19.08]	[1.95, 3.04]

Tabela A.10: Intervalos de confiança para as médias da Tabela A.9

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.8	0.7	0.1	0.3	0.5	0.0
4.2	2.2	1.1	1.9	1.2	0.0
17.9	5.2	4.9	3.2	2.5	0.0

Tabela A.11: Médias referentes ao gráfico da Figura 7.2-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.35, 1.24]	[0.32, 1.07]	[-0.07, 0.27]	[0.02, 0.57]	[0.19, 0.80]	[0.0, 0.0]
[2.35, 6.04]	[0.76, 3.63]	[0.69, 1.50]	[1.31, 2.48]	[0.85, 1.54]	[0.0, 0.0]
[11.38, 24.41]	[2.29, 8.10]	[3.02, 6.77]	[2.38, 4.01]	[1.88, 3.11]	[0.0, 0.0]

Tabela A.12: Intervalos de confiança para as médias da Tabela A.11

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
30.6	24.2	22.2	7.0	22.0	12.4
80.7	64.7	58.8	18.0	54.6	31.2
127.6	102.9	93.2	26.5	92.6	50.4

Tabela A.13: Médias referentes ao gráfico da Figura 7.3-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[27.35, 33.84]	[21.40, 26.99]	[19.98, 24.41]	[6.04, 7.95]	[19.03, 24.96]	[10.89, 13.90]
[74.21, 87.18]	[60.12, 69.27]	[55.38, 62.21]	[16.94, 19.05]	[50.98, 58.21]	[28.87, 33.52]
[123.16, 132.03]	[98.77, 107.02]	[90.84, 95.55]	[25.71, 27.28]	[89.49, 95.70]	[48.76, 52.03]

Tabela A.14: Intervalos de confiança para as médias da Tabela A.13

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
16.9	8.4	8.0	7.1	7.7	6.0
40.3	21.9	17.0	17.5	17.1	15.7
59.0	39.2	28.3	27.7	27.7	25.0

Tabela A.15: Médias referentes ao gráfico da Figura 7.3-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[15.15, 18.64]	[7.54, 9.25]	[7.11, 8.88]	[6.38, 7.81]	[6.81, 8.58]	[5.35, 6.64]
[37.91, 42.68]	[20.09, 23.70]	[15.73, 18.26]	[16.51, 18.48]	[16.11, 18.08]	[14.64, 16.75]
[57.22, 60.77]	[37.32, 41.07]	[27.13, 29.46]	[26.60, 28.79]	[26.60, 28.79]	[24.04, 25.95]

Tabela A.16: Intervalos de confiança para as médias da Tabela A.15

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
11.2	10.3	10.6	6.0	11.8	11.8
25.1	23.4	26.3	14.4	35.1	30.5
41.4	40.6	40.4	22.0	56.8	51.1

Tabela A.17: Médias referentes ao gráfico da Figura 7.3-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[10.00, 12.39]	[9.13, 11.46]	[9.09, 12.10]	[5.18, 6.81]	[9.82, 13.77]	[10.43, 13.16]
[22.71, 27.48]	[21.48, 25.31]	[23.94, 28.65]	[13.30, 15.49]	[32.64, 37.55]	[28.69, 32.30]
[40.27, 42.52]	[39.33, 41.86]	[39.41, 41.38]	[21.14 , 22.85]	[53.83, 59.76]	[49.56, 52.63]

Tabela A.18: Intervalos de confiança para as médias da Tabela A.17

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
6.4	5.9	5.8	6.5	6.5	6.4
13.6	14.7	13.9	13.5	13.0	15.6
21.8	22.0	22.2	22.0	21.9	24.4

Tabela A.19: Médias referentes ao gráfico da Figura 7.3-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[5.64, 7.15]	[5.14, 6.65]	[4.87, 6.72]	[5.68, 7.31]	[5.64, 7.35]	[5.64, 7.15]
[12.37, 14.82]	[13.36, 16.03]	[12.56, 15.23]	[12.54, 14.45]	[11.97, 14.02]	[14.50, 16.69]
[21.39, 22.20]	[21.31, 22.68]	[21.55, 22.84]	[21.38, 22.61]	[21.11, 22.68]	[23.58, 25.21]

Tabela A.20: Intervalos de confiança para as médias da Tabela A.19

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
1.4	0.4	1.0	0.0	0.6	0.3
4.5	2.3	2.8	0.6	2.6	1.3
11.4	6.8	4.8	0.8	4.4	2.0

Tabela A.21: Médias referentes ao gráfico da Figura 7.3-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.47, 2.32]	[0.05, 0.74]	[0.35, 1.64]	[-0.10, 0.10]	[0.19, 1.00]	[0.06, 0.53]
[2.38, 6.61]	[1.10, 3.49]	[1.40, 4.19]	[0.25, 0.94]	[1.37, 3.82]	[0.82, 1.77]
[8.56, 14.23]	[4.51, 9.08]	[2.99, 6.60]	[0.42, 1.17]	[2.59, 6.20]	[1.59, 2.40]

Tabela A.22: Intervalos de confiança para as médias da Tabela A.21

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.8	0.2	0.0	0.2	0.3	0.0
1.5	0.6	1.0	0.5	0.4	0.0
1.7	1.0	0.7	0.7	0.8	0.0

Tabela A.23: Médias referentes ao gráfico da Figura 7.3-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.32, 1.27]	[-0.03, 0.43]	[-0.10, 0.10]	[-0.03, 0.43]	[0.02, 0.57]	[0.0, 0.0]
[0.88, 2.11]	[0.22, 0.97]	[0.59, 1.40]	[0.19, 0.80]	[0.12, 0.67]	[0.0, 0.0]
[1.05, 2.34]	[0.52, 1.47]	[0.39, 1.00]	[0.39, 1.00]	[0.42, 1.17]	[0.0, 0.0]

Tabela A.24: Intervalos de confiança para as médias da Tabela A.23

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
90.0	94.0	90.0	100.9	90.0	98.4
90.0	101.4	90.0	119.5	90.0	99.2
90.0	114.4	90.0	141.7	90.0	99.6

Tabela A.25: Médias referentes ao gráfico da Figura 7.4-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[90.0, 90.0]	[93.45, 94.54]	[90.0, 90.0]	[99.19, 102.60]	[90.0, 90.0]	[97.20, 99.59]
[90.0, 90.0]	[100.37, 102.42]	[90.0, 90.0]	[117.31, 121.68]	[90.0, 90.0]	[98.55, 99.84]
[90.0, 90.0]	[112.72, 116.07]	[90.0, 90.0]	[139.95, 143.44]	[90.0, 90.0]	[99.22, 99.97]

Tabela A.26: Intervalos de confiança para as médias da Tabela A.25

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
100.9	104.7	101.5	99.7	101.4	107.3
125.6	128.9	119.3	116.9	120.5	118.5
168.1	164.1	141.5	141.7	140.9	128.4

Tabela A.27: Médias referentes ao gráfico da Figura 7.4-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[99.09, 102.70]	[102.51, 106.88]	[99.89, 103.10]	[98.53, 100.86]	[99.83, 102.96]	[105.86, 108.73]
[121.91, 129.28]	[125.89, 131.90]	[117.35, 121.24]	[114.95, 118.84]	[118.17, 122.82]	[117.23, 119.76]
[163.22, 172.97]	[161.26, 166.93]	[139.41, 143.58]	[139.61, 143.78]	[138.95, 142.84]	[127.64, 129.15]

Tabela A.28: Intervalos de confiança para as médias da Tabela A.27

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
90.2	90.4	90.2	104.2	90.0	99.1
90.7	91.5	90.6	117.3	90.1	99.4
91.7	93.3	91.2	127.0	90.3	99.9

Tabela A.29: Médias referentes ao gráfico da Figura 7.4-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[90.16, 90.23]	[90.33, 90.46]	[90.16, 90.23]	[101.40, 106.99]	[90.0, 90.0]	[98.11, 100.08]
[90.63, 90.76]	[91.32, 91.67]	[90.53, 90.66]	[115.01, 119.58]	[90.1, 90.1]	[98.81, 99.98]
[91.59, 91.80]	[93.09, 93.50]	[91.13, 91.26]	[124.47, 129.52]	[90.3, 90.3]	[99.59, 100.20]

Tabela A.30: Intervalos de confiança para as médias da Tabela A.29

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
101.7	100.3	98.2	104.0	103.8	106.9
115.2	114.9	110.8	117.5	118.8	119.5
123.8	124.9	120.1	127.2	130.2	128.9

Tabela A.31: Médias referentes ao gráfico da Figura 7.4-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[100.06, 103.33]	[98.90, 101.69]	[96.76, 99.63]	[101.67, 106.32]	[101.30, 106.29]	[105.50, 108.29]
[113.52, 116.87]	[113.09, 116.70]	[108.85, 112.74]	[115.07, 119.92]	[115.52, 122.07]	[118.13, 120.86]
[122.05, 125.54]	[123.26, 126.53]	[118.59, 121.60]	[124.36, 130.03]	[128.05, 132.34]	[127.70, 130.09]

Tabela A.32: Intervalos de confiança para as médias da Tabela A.31

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
270.1	224.8	194.3	143.4	192.1	129.1
842.1	541.3	382.8	227.8	409.9	185.0
1946.0	1060.6	837.4	336.3	1046.4	239.4

Tabela A.33: Médias referentes ao gráfico da Figura 7.4-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[233.55, 306.64]	[197.40, 252.19]	[175.09, 213.50]	[135.38, 151.41]	[176.67, 207.52]	[123.77, 134.42]
[733.73, 950.46]	[476.67, 605.92]	[351.85, 413.74]	[217.80, 237.79]	[363.70, 456.09]	[179.13, 190.86]
[1496.5, 2395.4]	[854.89, 1266.30]	[718.90, 955.89]	[323.06, 349.53]	[870.72, 1222.0]	[232.74, 246.05]

Tabela A.34: Intervalos de confiança para as médias da Tabela A.33

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
203.0	154.6	141.8	148.0	139.6	106.4
425.7	278.0	231.7	229.5	241.1	119.0
874.2	536.2	351.5	349.8	348.8	129.5

Tabela A.35: Médias referentes ao gráfico da Figura 7.4-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[187.16, 218.83]	[147.16, 162.03]	[135.28, 148.31]	[141.00, 154.99]	[132.98, 146.21]	[104.93, 107.86]
[389.09, 462.30]	[262.20, 293.79]	[222.07, 241.32]	[215.88, 243.11]	[229.94, 252.25]	[117.66, 120.33]
[769.93, 978.46]	[499.07, 573.32]	[329.66, 373.33]	[324.85, 374.74]	[334.09, 363.50]	[128.64, 130.35]

Tabela A.36: Intervalos de confiança para as médias da Tabela A.35

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
90.0	91.8	90.0	100.2	90.0	93.5
90.0	94.7	90.0	120.8	90.0	96.5
90.0	97.1	90.0	138.4	90.0	96.7

Tabela A.37: Médias referentes ao gráfico da Figura 7.5-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[90.0, 90.0]	[91.52, 92.07]	[90.0, 90.0]	[98.42, 101.97]	[90.0, 90.0]	[92.51, 94.48]
[90.0, 90.0]	[94.32, 95.07]	[90.0, 90.0]	[118.82, 122.77]	[90.0, 90.0]	[95.74, 97.25]
[90.0, 90.0]	[96.79, 97.40]	[90.0, 90.0]	[136.72, 140.07]	[90.0, 90.0]	[96.32, 97.07]

Tabela A.38: Intervalos de confiança para as médias da Tabela A.37

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
102.6	103.5	102.1	101.2	102.4	98.9
125.0	124.1	119.4	119.9	120.2	106.6
148.2	147.4	141.7	140.2	139.8	114.6

Tabela A.39: Médias referentes ao gráfico da Figura 7.5-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[101.20, 103.99]	[102.23, 104.76]	[100.70, 103.49]	[99.86, 102.53]	[100.76, 104.03]	[97.09, 100.70]
[122.40, 127.59]	[121.64, 126.55]	[117.14, 121.65]	[118.12, 121.67]	[118.08, 122.31]	[104.75, 108.44]
[145.94, 150.45]	[145.31, 149.48]	[139.85, 143.54]	[138.59, 141.80]	[138.09, 141.50]	[113.64, 115.55]

Tabela A.40: Intervalos de confiança para as médias da Tabela A.39

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
90.0	90.0	90.0	95.4	90.0	94.5
90.4	90.4	90.2	117.1	90.0	95.8
90.7	90.6	90.3	141.9	90.1	97.4

Tabela A.41: Médias referentes ao gráfico da Figura 7.5-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[90.0, 90.0]	[90.0, 90.0]	[90.0, 90.0]	[93.52, 97.27]	[90.0, 90.0]	[93.33, 95.66]
[90.33, 90.46]	[90.33, 90.46]	[90.16, 90.23]	[115.35, 118.84]	[90.0, 90.0]	[95.04, 96.55]
[90.66, 90.73]	[90.56, 90.63]	[90.3, 90.3]	[138.04, 145.75]	[90.1, 90.1]	[97.02, 97.77]

Tabela A.42: Intervalos de confiança para as médias da Tabela A.41

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
94.5	93.5	94.9	96.0	96.1	99.0
105.6	106.7	108.9	115.7	114.5	107.6
117.7	118.2	125.7	141.7	142.9	114.4

Tabela A.43: Médias referentes ao gráfico da Figura 7.5-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[93.51, 95.48]	[92.74, 94.25]	[93.53, 96.26]	[94.12, 97.87]	[94.12, 98.07]	[96.91, 101.08]
[104.26, 106.93]	[105.36, 108.03]	[107.56, 110.23]	[114.02, 117.37]	[112.48, 116.51]	[105.82, 109.37]
[116.60, 118.79]	[117.03, 119.36]	[124.26, 127.13]	[137.94, 145.45	139.52, 146.27]	[113.17, 115.62]

Tabela A.44: Intervalos de confiança para as médias da Tabela A.43

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
173.6	162.3	156.4	120.1	152.1	122.5
368.2	316.1	270.1	162.9	271.6	178.2
676.4	536.0	411.9	217.9	417.6	235.2

Tabela A.45: Médias referentes ao gráfico da Figura 7.5-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[162.37, 184.82]	[150.73, 173.86]	[147.08, 165.71]	[115.45, 124.74]	[143.19, 161.00]	[118.23, 126.76]
[340.05, 396.34]	[293.58, 338.61]	[254.95, 285.24]	[157.57, 168.22]	[254.74, 288.45]	[172.57, 183.82]
[623.75, 729.04]	[501.36, 570.63]	[391.63, 432.16]	[210.12, 225.67]	[396.99, 438.20]	[227.42, 242.97]

Tabela A.46: Intervalos de confiança para as médias da Tabela A.45

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
150.4	122.4	119.9	122.2	117.4	115.0
236.7	183.1	164.4	174.9	166.8	142.5
384.2	274.8	223.8	219.2	217.4	169.3

Tabela A.47: Médias referentes ao gráfico da Figura 7.5-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[143.09, 157.70]	[118.10, 126.69]	[115.80, 123.99]	[117.08, 127.31]	[114.22, 120.57]	[112.03, 117.96]
[224.48, 248.91]	[177.98, 188.21]	[159.24, 169.55]	[170.90, 178.89]	[161.03, 172.56]	[139.56, 145.43]
[371.13, 397.26]	[264.49, 285.10]	[217.41, 230.18]	[211.69, 226.70]	[209.10, 225.69]	[165.41, 173.18]

Tabela A.48: Intervalos de confiança para as médias da Tabela A.47

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
31.0	64.4	15.6	33.3	30.0	73.1
73.5	155.6	37.6	84.4	72.6	166.0
116.9	251.7	58.6	140.0	115.8	259.4

Tabela A.49: Médias referentes ao gráfico da Figura 7.8-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[28.81, 33.18]	[59.72, 69.07]	[14.26, 16.93]	[30.26, 36.33]	[27.67, 32.32]	[67.91, 78.28]
[70.70, 76.29]	[148.84, 162.35]	[35.89, 39.30]	[80.47, 88.32]	[70.34, 74.85]	[159.03, 172.96]
[115.26, 118.53]	[246.54, 256.85]	[57.33, 59.86]	[137.13, 142.86]	[113.58, 118.01]	[253.83, 264.96]

Tabela A.50: Intervalos de confiança para as médias da Tabela A.49

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
46.2	87.5	37.0	33.8	51.7	71.0
119.2	204.8	85.9	83.5	126.0	166.1
196.7	332.9	138.9	137.4	194.2	260.3

Tabela A.51: Médias referentes ao gráfico da Figura 7.8-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[42.82, 49.57]	[80.16, 94.83]	[33.92, 40.07]	[31.51, 36.08]	[46.65, 56.74]	[65.33, 76.66]
[114.49, 123.90]	[195.89, 213.70]	[82.41, 89.38]	[79.64, 87.35]	[120.74, 131.25]	[159.89, 172.30]
[192.91, 200.48]	[327.16, 338.63]	[135.86, 141.93]	[134.09, 140.70]	[190.68, 197.71]	[254.32, 266.27]

Tabela A.52: Intervalos de confiança para as médias da Tabela A.51

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
79.8	96.7	64.6	72.7	77.5	68.3
186.2	250.1	162.4	172.4	191.2	164.6
295.9	395.7	249.5	264.5	308.3	260.1

Tabela A.53: Médias referentes ao gráfico da Figura 7.8-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[73.59, 86.0]	[88.17, 105.22]	[59.31, 69.88]	[66.79, 78.60]	[71.52, 83.47]	[62.56, 74.03]
[178.11, 194.28]	[240.64, 259.55]	[154.96, 169.83]	[165.88, 178.91]	[182.15, 200.24]	[157.94, 171.25]
[290.2, 301.59]	[386.28, 405.11]	[244.75, 254.24]	[258.08, 270.91]	[302.70, 313.89]	[254.16, 266.03]

Tabela A.54: Intervalos de confiança para as médias da Tabela A.53

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
78.9	116.8	76.0	73.7	82.5	66.6
195.3	296.9	174.8	171.8	204.6	164.7
311.6	471.8	270.0	266.6	330.1	261.0

Tabela A.55: Médias referentes ao gráfico da Figura 7.8-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[72.96, 84.83]	[108.30, 125.29]	[71.35, 80.64]	[68.17, 79.22]	[75.64, 89.35]	[60.08, 73.11]
[190.07, 200.52]	[284.17, 309.62]	[166.03, 183.56]	[166.37, 177.22]	[196.10, 213.09]	[156.75, 172.64]
[305.76, 317.43]	[463.03, 480.56]	[265.66, 274.33]	[261.48, 271.71]	[323.82, 336.37]	[256.42, 265.57]

Tabela A.56: Intervalos de confiança para as médias da Tabela A.55

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
390.2	412.3	343.9	344.6	368.6	351.8
486.0	647.4	401.2	438.0	494.0	499.2
378.2	812.7	461.1	479.5	639.6	620.9

Tabela A.57: Médias referentes ao gráfico da Figura 7.8-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[371.50, 408.89]	[387.73, 436.86]	[333.01, 354.78]	[334.33, 354.86]	[355.46, 381.73]	[342.92, 360.67]
[454.50, 517.49]	[594.75, 700.04]	[390.07, 412.32]	[429.81, 446.18]	[472.53, 515.46]	[488.75, 509.64]
[328.45, 427.94]	[671.82, 953.57]	[413.19, 509.00]	[473.87, 485.12]	[517.48, 761.71]	[612.84, 628.95]

Tabela A.58: Intervalos de confiança para as médias da Tabela A.57

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
381.3	394.8	343.5	347.6	363.1	378.0
501.0	567.2	441.1	436.4	501.1	527.6
537.0	709.6	476.1	488.1	574.4	658.5

Tabela A.59: Médias referentes ao gráfico da Figura 7.8-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[368.67, 393.92]	[383.19, 406.40]	[335.27, 351.72]	[338.96, 356.23]	[353.00, 373.19]	[368.65, 387.34]
[488.92, 513.07]	[554.37, 580.02]	[433.93, 448.26]	[428.72, 444.07]	[492.12, 510.07]	[516.85, 538.34]
[503.15, 570.84]	[694.62, 724.57]	[470.19, 482.00]	[481.07, 495.12]	[568.05, 580.74]	[651.94, 665.05]

Tabela A.60: Intervalos de confiança para as médias da Tabela A.59

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.0	19.7	0.0	20.1	0.0	665.2
0.0	48.4	0.0	50.3	0.0	690.9
0.0	78.7	0.0	80.2	0.0	702.6

Tabela A.61: Médias referentes ao gráfico da Figura 7.9-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.0, 0.0]	[17.41, 21.98]	[0.0, 0.0]	[17.06, 23.13]	[0.0, 0.0]	[649.50, 680.89]
[0.0, 0.0]	[45.26, 51.53]	[0.0, 0.0]	[46.99, 53.60]	[0.0, 0.0]	[682.33, 699.43]
[0.0, 0.0]	[76.27, 81.12]	[0.0, 0.0]	[77.91, 82.48]	[0.0, 0.0]	[698.43, 706.76]

Tabela A.62: Intervalos de confiança para as médias da Tabela A.61

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
18.8	40.2	21.1	18.1	22.3	683.7
48.6	96.9	50.0	46.2	53.4	694.7
80.1	160.2	79.5	80.3	81.6	703.8

Tabela A.63: Médias referentes ao gráfico da Figura 7.9-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[15.96, 21.63]	[34.33, 46.06]	[18.30, 23.89]	[15.98, 20.21]	[19.29, 25.30]	[671.41, 695.98]
[45.15, 52.04]	[90.58, 103.21]	[47.03, 52.96]	[43.26, 49.13]	[49.71, 57.08]	[686.64, 702.75]
[77.23, 82.96]	[156.00, 164.39]	[76.83, 82.16]	[77.63, 82.96]	[79.10, 84.09]	[700.42, 707.17]

Tabela A.64: Intervalos de confiança para as médias da Tabela A.63

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.0	10.9	0.0	21.0	0.0	685.0
0.0	28.1	0.0	48.8	0.0	693.1
0.0	46.4	0.0	73.2	0.0	705.6

Tabela A.65: Médias referentes ao gráfico da Figura 7.9-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.0, 0.0]	[9.46, 12.33]	[0.0, 0.0]	[17.82, 24.17]	[0.0, 0.0]	[671.79, 698.20]
[0.0, 0.0]	[26.01, 30.18]	[0.0, 0.0]	[45.89, 51.70]	[0.0, 0.0]	[686.27, 699.92]
[0.0, 0.0]	[44.42, 48.37]	[0.0, 0.0]	[69.75, 76.64]	[0.0, 0.0]	[702.80, 708.39]

Tabela A.66: Intervalos de confiança para as médias da Tabela A.65

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
19.9	27.6	17.8	22.0	21.0	686.3
52.0	78.1	50.1	50.4	51.5	692.8
80.1	121.0	76.2	75.8	86.5	702.2

Tabela A.67: Médias referentes ao gráfico da Figura 7.9-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[]17.34, 22.45]	[23.98, 31.21]	[15.30, 20.29]	[19.03, 24.96]	[18.03, 23.96]	[674.05, 698.54]
[48.24, 55.75]	[73.39, 82.80]	[46.61, 53.58]	[47.15, 53.64]	[47.61, 55.38]	[686.11, 699.48]
[77.02, 83.17]	[116.70, 125.29]	[74.28, 78.11]	[73.27, 78.32]	[83.66, 89.33]	[698.37, 706.02]

Tabela A.68: Intervalos de confiança para as médias da Tabela A.67

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
352.8	253.2	214.7	180.8	224.3	71.3
1189.6	780.7	539.2	424.6	629.5	195.0
2729.5	1640.8	1399.2	702.7	1756.5	308.9

Tabela A.69: Médias referentes ao gráfico da Figura 7.9-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[291.35,414.24]	[203.62,302.77]	[178.02,251.37]	[153.64,207.95]	[192.63,255.96]	[60.24,82.35]
[1044.69,1334.50]	[667.73,893.66]	[488.39,590.01]	[392.93,456.26]	[539.62,719.37]	[183.02,206.97]
[2197.4,3261.5]	[1415.0,1866.5]	[1198.0,1600.3]	[659.47,745.92]	[1481.3,2031.6]	[295.62,322.17]

Tabela A.70: Intervalos de confiança para as médias da Tabela A.69

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
325.3	176.5	168.3	192.0	168.8	21.5
850.0	482.1	433.7	440.3	470.0	47.5
2096.9	1217.9	840.8	782.8	792.3	75.5

Tabela A.71: Médias referentes ao gráfico da Figura 7.9-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[280.87,369.72]	[156.81,196.18]	[148.68,187.91]	[169.07,214.92]	[146.48,191.11]	[18.05,24.94]
[763.67,936.32]	[441.05,523.14]	[402.82,464.57]	[393.14,487.45]	[435.53,504.46]	[43.26,51.73]
[1827.76,2366.03]	[1097.56,1338.23]	[748.95,932.64]	[693.47,872.12]	[732.52,852.07]	[71.13,79.86]

Tabela A.72: Intervalos de confiança para as médias da Tabela A.71

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.0	0.0	0.0	15.1	3.1	122.2
0.0	0.0	0.0	33.7	7.0	102.7
0.0	0.0	0.0	49.6	11.6	98.1

Tabela A.73: Médias referentes ao gráfico da Figura 7.10-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[13.12, 17.07]	[2.69, 3.50]	[107.69, 136.70
[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[31.75, 35.64]	[6.55, 7.44]	[94.44, 110.95]
[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[47.85, 51.34]	[11.29, 11.90]	[94.82, 101.37]

Tabela A.74: Intervalos de confiança para as médias da Tabela A.73

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
17.4	16.2	17.4	14.5	20.2	147.0
34.1	33.5	33.0	33.4	40.4	175.6
50.1	50.2	52.7	50.6	62.5	193.5

Tabela A.75: Médias referentes ao gráfico da Figura 7.10-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[15.21, 19.58]	[14.73, 17.66]	[15.35, 19.44]	[12.48, 16.51]	[17.91, 22.48]	[130.48, 163.51]
[31.84, 36.35]	[30.83, 36.16]	[30.67, 35.32]	[31.52, 35.27]	[37.32, 43.47]	[166.72, 184.47]
[48.05, 52.14]	[48.42, 51.97]	[50.95, 54.44]	[48.96, 52.23]	[60.72, 64.27]	[189.06, 197.93]

Tabela A.76: Intervalos de confiança para as médias da Tabela A.75

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
107.1	94.8	137.9	238.4	147.6	110.2
77.3	122.3	119.3	227.7	146.0	104.5
52.6	110.0	109.6	207.3	157.7	95.4

Tabela A.77: Médias referentes ao gráfico da Figura 7.10-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[87.00, 127.19]	[76.13, 113.46]	[109.37, 166.42]	[211.71, 265.08]	[127.33, 167.86]	[95.42, 124.97]
[67.13, 87.46]	[109.53, 135.06]	[101.35, 137.24]	[218.93, 236.46]	[135.90, 156.09]	[97.40, 111.59]
[48.57, 56.62]	[100.13, 119.86]	[104.61, 114.58]	[201.94, 212.65]	[151.08, 164.31]	[91.30, 99.49]

Tabela A.78: Intervalos de confiança para as médias da Tabela A.77

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
206.8	124.8	222.0	241.4	271.7	142.9
217.5	168.6	252.5	226.0	289.9	169.3
216.6	178.0	234.7	205.9	292.7	194.1

Tabela A.79: Médias referentes ao gráfico da Figura 7.10-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[193.59, 220.00]	[103.74, 145.85]	[194.60, 249.39]	[223.72, 259.07]	[245.97, 297.42]	[125.66, 160.13]
[210.30, 224.69]	[157.37, 179.82]	[242.12, 262.87]	[216.85, 235.14]	[281.88, 297.91]	[161.17, 177.42]
[215.06, 218.13]	[170.52, 185.47]	[232.00, 237.39]	[200.98, 210.81]	[289.15, 296.24]	[189.35, 198.84]

Tabela A.80: Intervalos de confiança para as médias da Tabela A.79

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
55.1	44.6	42.3	38.5	49.0	163.3
96.6	90.7	79.5	75.0	104.8	250.1
69.3	114.9	91.3	90.0	134.0	300.3

Tabela A.81: Médias referentes ao gráfico da Figura 7.10-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[48.75, 61.44]	[38.15, 51.04]	[36.97, 47.62]	[32.93, 44.06]	[41.69, 56.30]	[144.70, 181.89]
[88.99, 104.20]	[84.45, 96.94]	[74.45, 84.54]	[69.54, 80.45]	[96.98, 112.61]	[234.33, 265.86]
[57.66, 80.93]	[107.22, 122.57]	[83.21, 99.38]	[85.93, 94.06]	[126.56, 141.43]	[286.78, 313.81]

Tabela A.82: Intervalos de confiança para as médias da Tabela A.81

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
53.3	36.7	37.5	39.9	40.0	201.5
93.3	78.1	72.3	80.2	88.4	275.0
101.0	106.0	99.2	95.5	117.6	336.8

Tabela A.83: Médias referentes ao gráfico da Figura 7.10-(f)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[48.38, 58.21]	[32.46, 40.93]	[33.20, 41.79]	[35.02, 44.77]	[35.25, 44.74]	[184.57, 218.43]
[87.15, 99.44]	[73.32, 82.87]	[66.73, 77.86]	[75.55, 84.84]	[82.59, 94.20]	[264.55, 285.44]
[93.15, 108.84]	[100.71, 111.28]	[95.00, 103.39]	[90.82, 100.17]	[112.92, 122.27]	[325.23, 348.36]

Tabela A.84: Intervalos de confiança para as médias da Tabela A.83

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.0	1.4	0.0	14.1	0.0	0.0
0.0	2.0	0.0	41.1	0.0	0.0
0.0	1.3	0.0	62.3	0.0	0.0

Tabela A.85: Médias referentes ao gráfico da Figura 7.11-(a)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.0, 0.0]	[1.12, 1.67]	[0.0, 0.0]	[11.71, 16.48]	[0.0, 0.0]	[0.0, 0.0]
[0.0, 0.0]	[1.62, 2.37]	[0.0, 0.0]	[38.54, 43.65]	[0.0, 0.0]	[0.0, 0.0]
[0.0, 0.0]	[0.99, 1.60]	[0.0, 0.0]	[60.38, 64.21]	[0.0, 0.0]	[0.0, 0.0]

Tabela A.86: Intervalos de confiança para as médias da Tabela A.85

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
14.0	15.6	17.0	15.8	17.4	5.3
35.1	36.1	39.1	40.0	40.7	15.1
53.5	55.0	66.7	64.4	64.1	25.3

Tabela A.87: Médias referentes ao gráfico da Figura 7.11-(b)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[12.49, 15.50]	[14.09, 17.10]	[15.05, 18.94]	[13.88, 17.71]	[15.11, 19.68]	[4.48, 6.11]
[32.84, 37.35]	[33.74, 38.45]	[36.30, 41.89]	[37.64, 42.35]	[37.93, 43.46]	[13.83, 16.36]
[51.96, 55.03]	[53.29, 56.70]	[64.48, 68.91]	[62.52, 66.27]	[62.15, 66.04]	[24.41, 26.18]

Tabela A.88: Intervalos de confiança para as médias da Tabela A.87

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
0.0	0.0	0.0	4.5	0.0	0.0
0.0	0.0	0.0	23.5	0.0	0.0
0.0	0.0	0.0	45.6	0.0	0.0

Tabela A.89: Médias referentes ao gráfico da Figura 7.11-(c)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[3.27, 5.72]	[0.0, 0.0]	[0.0, 0.0]
[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[21.48, 25.51]	[0.0, 0.0]	[0.0, 0.0]
[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.0]	[42.73, 48.46]	[0.0, 0.0]	[0.0, 0.0]

Tabela A.90: Intervalos de confiança para as médias da Tabela A.89

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
3.9	3.4	3.9	5.0	4.9	5.9
10.9	11.8	16.8	21.6	20.6	16.1
18.6	18.8	33.7	45.1	46.2	25.6

Tabela A.91: Médias referentes ao gráfico da Figura 7.11-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[3.31, 4.48]	[2.85, 3.94]	[3.04, 4.75]	[3.77, 6.22]	[3.56, 6.23]	[5.14, 6.65]
[9.97, 11.82]	[10.81, 12.78]	[14.99, 18.60]	[19.48, 23.71]	[18.41, 22.78]	[14.90, 17.29]
[18.05, 19.14]	[18.21, 19.38]	[32.60, 34.79]	[42.37, 47.82]	[43.94, 48.45]	[24.57, 26.62]

Tabela A.92: Intervalos de confiança para as médias da Tabela A.91

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
27.3	24.7	23.6	22.8	22.3	10.9
73.4	66.2	57.5	56.7	57.5	30.6
116.4	100.1	90.2	90.2	91.6	49.4

Tabela A.93: Médias referentes ao gráfico da Figura 7.11-(d)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[23.88, 30.71]	[21.15, 28.24]	[20.56, 26.63]	[19.11, 26.48]	[19.22, 25.37]	[9.53, 12.26]
[67.97, 78.82]	[61.86, 70.53]	[53.37, 61.62]	[53.18, 60.21]	[53.20, 61.79]	[28.65, 32.54]
[111.31, 121.48]	[96.00, 104.19]	[87.09, 93.30]	[86.75, 93.64]	[88.01, 95.18]	[47.45, 51.34]

Tabela A.94: Intervalos de confiança para as médias da Tabela A.93

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
32.1	21.9	22.6	24.2	21.7	11.9
72.4	56.6	57.2	62.3	57.5	30.1
110.3	90.2	93.0	91.2	91.1	48.4

Tabela A.95: Médias referentes ao gráfico da Figura 7.11-(e)

MobileIP	MobileIPSH	CellularIP	HawaiiMSF	HawaiiMNF	Multicast
[28.48, 35.71]	[19.30, 24.49]	[19.49, 25.70]	[20.34, 28.05]	[19.10, 24.29]	[10.50, 13.29]
[68.54, 76.25]	[54.24, 58.95]	[53.20, 61.19]	[59.70, 64.89]	[53.64, 61.35]	[28.42, 31.77]
[107.84, 112.75]	[86.89, 93.50]	[90.30, 95.69]	[88.57, 93.82]	[87.75, 94.44]	[46.25, 50.54]

Tabela A.96: Intervalos de confiança para as médias da Tabela A.95

MobileIP	CellularIP	Multicast
50.3	31.2	13.7
50.9	31.9	13.5
49.0	44.9	13.5

Tabela A.97: Médias referentes ao gráfico da Figura 7.13-(a)

MobileIP	CellularIP	Multicast
[44.43, 56.16]	[27.71, 34.68]	[12.16, 15.23]
[44.00, 57.79]	[27.36, 36.43]	[11.89, 15.10]
[43.37, 54.62]	[39.54, 50.25]	[11.82, 15.17]

Tabela A.98: Intervalos de confiança para as médias da Tabela A.97

MobileIP	CellularIP	Multicast
198.5	125.7	53.7
202.0	156.5	51.8
202.7	180.8	52.4

Tabela A.99: Médias referentes ao gráfico da Figura 7.13-(b)

MobileIP	CellularIP	Multicast
[191.60, 205.39]	[121.94, 129.45]	[51.92, 55.47]
[197.25, 206.74]	[151.14, 161.85]	[50.12, 53.47]
[196.49, 208.90]	[176.05, 185.54]	[50.65, 54.14]

Tabela A.100: Intervalos de confiança para as médias da Tabela A.99

MobileIP	CellularIP	Multicast
23.3	12.6	0.0
23.0	12.6	0.0
30.6	21.1	0.4

Tabela A.101: Médias referentes ao gráfico da Figura 7.13-(c)

MobileIP	CellularIP	Multicast
[20.29, 26.30]	[10.96, 14.23]	[0.0, 0.0]
[19.89, 26.10]	[11.37, 13.82]	[0.0, 0.0]
[27.18, 34.01]	[16.49, 25.70]	[0.19, 0.60]

Tabela A.102: Intervalos de confiança para as médias da Tabela A.101

MobileIP	CellularIP	Multicast
95.8	53.3	0.0
122.3	53.6	0.0
170.4	149.5	6.7

Tabela A.103: Médias referentes ao gráfico da Figura 7.13-(d)

MobileIP	CellularIP	Multicast]
[91.33, 100.26]	[51.49, 55.10]	[0.0, 0.0]
[119.70, 124.89]	[52.16, 55.03]	[0.0, 0.0]
[162.38, 178.41]	[142.53, 156.46]	[6.05, 7.34]

Tabela A.104: Intervalos de confiança para as médias da Tabela A.103

MobileIP	CellularIP	Multicast
24.2	9.7	13.6
25.6	18.8	14.0
26.6	25.6	15.4

Tabela A.105: Médias referentes ao gráfico da Figura 7.13-(e)

MobileIP	CellularIP	Multicast
[20.71, 27.68]	[8.16, 11.23]	[11.48, 15.71]
[21.81, 29.38]	[16.61, 20.98]	[12.25, 15.74]
[23.15, 30.04]	[21.71, 29.48]	[13.86, 16.93]

Tabela A.106: Intervalos de confiança para as médias da Tabela A.105

MobileIP	CellularIP	Multicast
91.1	35.3	53.0
88.8	70.6	51.1
91.1	92.6	53.7

Tabela A.107: Médias referentes ao gráfico da Figura 7.13-(f)

MobileIP	CellularIP	Multicast
[87.99, 94.20]	[33.83, 36.76]	[51.32, 54.67]
[85.76, 91.83]	[68.07, 73.12]	[49.32, 52.87]
[87.96, 94.23]	[89.76, 95.43]	[52.09, 55.30]

Tabela A.108: Intervalos de confiança para as médias da Tabela A.107

MobileIP	CellularIP	Multicast
8.8	9.2	0.0
9.3	8.9	0.0
9.1	10.6	0.5

Tabela A.109: Médias referentes ao gráfico da Figura 7.14-(a)

MobileIP	CellularIP	Multicast
[7.36, 10.23]	[7.86, 10.53]	[0.0, 0.0]
[7.76, 10.83]	[7.84, 9.95]	[0.0, 0.0]
[7.73, 10.46]	[9.13, 12.06]	[0.26, 0.73]

Tabela A.110: Intervalos de confiança para as médias da Tabela A.109

MobileIP	CellularIP	Multicast
36.7	36.1	0.0
34.7	34.4	0.0
35.7	34.8	6.3

Tabela A.111: Médias referentes ao gráfico da Figura 7.14-(b)

MobileIP	CellularIP	Multicast
[35.23, 38.16]	[34.90, 37.29]	[0.0, 0.0]
[33.19, 36.20]	[32.76, 36.03]	[0.0, 0.0]
[34.53, 36.86]	[33.77, 35.82]	[5.68, 6.91]

Tabela A.112: Intervalos de confiança para as médias da Tabela A.111

MobileIP	CellularIP	Multicast
3.3	0.6	0.6
5.4	1.2	0.3
2.4	2.3	0.5

Tabela A.113: Médias referentes ao gráfico da Figura 7.14-(c)

MobileIP	CellularIP	Multicast
[1.08, 5.51]	[0.19, 1.00]	[0.22, 0.97]
[2.19, 8.60]	[0.58, 1.81]	[-0.07, 0.67]
[1.06, 3.73]	[0.86, 3.73]	[0.12, 0.87]

Tabela A.114: Intervalos de confiança para as médias da Tabela A.113

MobileIP	CellularIP	Multicast
93.5	5.4	3.7
107.3	50.6	3.6
114.6	74.8	3.6

Tabela A.115: Médias referentes ao gráfico da Figura 7.14-(d)

MobileIP	CellularIP	Multicast
[67.26, 119.73]	[4.51, 6.28]	[3.15, 4.24]
[79.35, 135.24]	[31.80, 69.39]	[3.08, 4.11]
[88.32, 140.87]	[52.86, 96.73]	[3.01, 4.18]

Tabela A.116: Intervalos de confiança para as médias da Tabela A.115

MobileIP	CellularIP	Multicast
3.6	0.8	0.0
1.8	0.8	0.0
0.8	1.1	0.0

Tabela A.117: Médias referentes ao gráfico da Figura 7.14-(e)

MobileIP	CellularIP	Multicast
[2.13, 5.06]	[0.35, 1.24]	[0.0, 0.0]
[0.98, 2.61]	[0.39, 1.20]	[0.0, 0.0]
[0.08, 1.51]	[0.11, 2.08]	[0.0, 0.0]

Tabela A.118: Intervalos de confiança para as médias da Tabela A.117

MobileIP	CellularIP	Multicast
6.8	3.2	0.0
35.6	3.8	0.0
85.2	64.0	0.3

Tabela A.119: Médias referentes ao gráfico da Figura 7.14-(e)

MobileIP	CellularIP	Multicast
[-3.64, 17.24]	[2.72, 3.67]	[0.0, 0.0]
[34.64, 36.55]	[3.18, 4.41]	[0.0, 0.0]
[63.19, 107.20]	[43.93, 84.06]	[0.16, 0.43]

Tabela A.120: Intervalos de confiança para as médias da Tabela A.119

MobileIP	CellularIP	Multicast
99.3	97.9	97.2
107.9	106.5	98.3
113.2	112.8	97.3

Tabela A.121: Médias referentes ao gráfico da Figura 7.15-(a)

MobileIP	CellularIP	Multicast
[98.31, 100.28]	[96.87, 98.92]	[96.21, 98.18]
[105.27, 110.52]	[104.75, 108.24]	[97.31, 99.28]
[110.36, 116.03]	[109.69, 115.90]	[96.34, 98.25]

Tabela A.122: Intervalos de confiança para as médias da Tabela A.121

MobileIP	CellularIP	Multicast
155.4	131.2	119.6
158.0	145.6	116.4
202.7	176.4	120.0

Tabela A.123: Médias referentes ao gráfico da Figura 7.15-(b)

MobileIP	CellularIP	Multicast
[151.20, 159.59]	[129.49, 132.90]	[118.57, 120.62]
[152.88, 163.11]	[142.69, 148.50]	[115.37, 117.42]
[199.35, 206.04]	[173.70, 179.09]	[119.07, 120.92]

Tabela A.124: Intervalos de confiança para as médias da Tabela A.123

MobileIP	CellularIP	Multicast
96.9	90.0	97.4
102.2	98.3	97.2
106.7	109.2	98.2

Tabela A.125: Médias referentes ao gráfico da Figura 7.15-(c)

MobileIP	CellularIP	Multicast
[95.77, 98.02]	[90.0, 90.0]	[96.51, 98.28]
[100.15, 104.24]	[97.31, 99.28]	[96.24, 98.15]
[104.31, 109.08]	[106.53, 111.86]	[97.34, 99.05]

Tabela A.126: Intervalos de confiança para as médias da Tabela A.125

MobileIP	CellularIP	Multicast
121.8	90.0	115.6
140.0	126.1	119.9
162.1	160.3	119.1

Tabela A.127: Médias referentes ao gráfico da Figura 7.15-(d)

MobileIP	CellularIP	Multicast
[120.33, 123.26]	[90.0, 90.0]	[114.67, 116.52]
[137.57, 142.42]	[124.15, 128.04]	[118.87, 120.92]
[159.47, 164.72]	[157.97, 162.62]	[118.34, 119.85]

Tabela A.128: Intervalos de confiança para as médias da Tabela A.127

MobileIP	CellularIP	Multicast
289.0	189.3	121.7
242.2	211.0	123.8
321.9	231.7	124.4

Tabela A.129: Médias referentes ao gráfico da Figura 7.15-(e)

MobileIP	CellularIP	Multicast
[254.30, 323.69]	[177.42, 201.17]	[116.99, 126.40]
[217.90, 266.49]	[188.17, 233.82]	[119.63, 127.96]
[284.53, 359.26]	[201.50, 261.89]	[120.10, 128.69]

Tabela A.130: Intervalos de confiança para as médias da Tabela A.129

MobileIP	CellularIP	Multicast
1171.7	561.6	217.4
1493.1	1151.4	219.7
1657.3	1380.6	220.0

Tabela A.131: Médias referentes ao gráfico da Figura 7.15-(f)

MobileIP	CellularIP	Multicast
[757.28, 1586.11]	[539.08, 584.11]	[213.51, 221.28]
[1035.73, 1950.46]	[856.57, 1446.22]	[215.46, 223.93]
[1331.01, 1983.58]	[1075.91, 1685.28]	[215.73, 224.26]

Tabela A.132: Intervalos de confiança para as médias da Tabela A.131

MobileIP	CellularIP	Multicast
190.5	135.2	97.2
190.8	145.2	97.2
217.6	200.7	98.8

Tabela A.133: Médias referentes ao gráfico da Figura 7.16-(a)

MobileIP	CellularIP	Multicast
[177.02, 203.97]	[129.46, 140.93]	[96.14, 98.25]
[174.96, 206.63]	[136.80, 153.59]	[96.31, 98.08]
[183.82, 251.37]	[173.88, 227.51]	[97.469, 100.13]

Tabela A.134: Intervalos de confiança para as médias da Tabela A.133

MobileIP	CellularIP	Multicast
628.8	321.9	120.0
945.1	358.7	118.5
1435.5	1235.2	132.4

Tabela A.135: Médias referentes ao gráfico da Figura 7.16-(b)

MobileIP	CellularIP	Multicast
[465.71, 791.88]	[309.20, 334.59]	[119.21, 120.78]
[914.32, 975.87]	[348.05, 369.34]	[117.44, 119.55]
[1098.53, 1772.46]	[961.05, 1509.34]	[130.35, 134.44]

Tabela A.136: Intervalos de confiança para as médias da Tabela A.135

Referências Bibliográficas

[1] *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers, 1999.

[2] A. Bakre and B. Badrinath. Inplementatin and Performance Evaluation of Indirect TCP. *IEEE Transactions on Computers*, 46(3):279–289, 1997.

[3] H. Balakrishnan, S. Seshan, and R. H. Katz. Improving Reliable Transport and Handoff Performance in Celluar Wireless Networks. *ACM Wireless Networks*, 1(4):469–481, 1995.

[4] N. Bhatti, M. Hiltunen, R. Schlichting, and W. Chiu. Coyote: a system for constructing fine-grain configurable communication services. *ACM Transactions on Computer Systems*, 16(4):321–366, November 1998.

[5] Nina T. Bhatti. *A System for Constructing Configurable High-level Protocols,*. PhD thesis, University of Arizona, December 1996.

[6] Nina T. Bhatti and Richard D. Schlichting. Configurable Communication Protocols for Mobile Computing. In *Proceedings of the 4th International Symposium on Autonomous Decentralized Systems*, pages 220–227, Tokyo, March 1999.

[7] Lawrence S. Brakmo, Sean W. O'Malley, and Larry L. Peterson. TCP vegas: New techniques for congestion detection and avoidance. In *SIGCOMM*, pages 24–35, 1994.

[8] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, and Z. Turanyi. Design, implementation, and evaluation of Cellular IP. In *IEEE Personal Commun. Mag.*, volume 7, August 2000.

[9] A. T. Campbell, J. Gomez, C-Y. Wan, Z. Turanyi, and A. Valko. Cellular IP. Internet Draft, draft-ietf-mobileip-cellularip-00.txt, January 2000. work in progress.

[10] Andrew T. Campbell, Javier Gomez, Sanghyo Kim, Zoltán Turányi, András Gergely Valkó, and Chieh-Yih Wan. Internet micromobility. *J. High Speed Netw.*, 11(3-4):177–198, 2002.

[11] Andrew T. Campbell and Javier Gomez-Castellanos. IP micro-mobility protocols. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(4):45–53, 2000.

[12] Claude Castelluccia. HMIPv6: A hierarchical mobile IPv6 proposal. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(1):48–59, 2000.

[13] Y. Chen, T. Farley, and N. Ye. QoS Requirements of Network Applications on the Internet. *Information-Knowledge-Systems Management*, 4:55–76, 2004. IOS Press.

[14] C. Chrysostomou, A. Pitsillides, and F.-N. Pavlidou. Survey of Wireless ATM Handover Issues. In *Proceedings of the International Symposium of 3G Infrastructure and Services, 3GIS*, pages 34–39, Athens, Greece, July 2001.

[15] Ramón Cáceres and Venkata N. Padmanabhan. Fast and Scalable Handoffs for Wireless Internetwoks. In *Proceedings of ACM Mobicom'96*, pages 56–66, 1996.

[16] Ricardo C.A. da Rocha and Markus Endler. MobiCS: An Environment for Prototyping and Simulating Distributed Protocols for Mobile Networks. In *Proc. 3rd IEEE Intern. Conference on Mobile and Wireless Communications Networks (MWCN2001), Recife - Brazil*, pages 44–51, August 2001.

[17] Greg Daley and Stefano Faccin. Some Requirements for a Media Independent Handover Information Service. Internet Draft, draft-faccin-mih-infoserv-00.txt, June 2005. Internet Draft - Work in Progress.

[18] DARPA. Internet Protocol. RFC 791, September 1981.

[19] DARPA. Transmission Control Protocol. RFC 793, September 1981.

[20] S. Das, A. Dutta, A. McAuley, A. Misra, and S. Das. IDMP: An IntraDomain Mobility Management Protocol using Mobility Agents. Internet Dratf, draft-elmalki-mobileip-fast-handoffs-03.txt, January 2000. Internet Draft - Work in Progress.

[21] S. Das, A. Misra, and P. Agrawal. TeleMIP: Telecommunication Enhanced Mobile IP Architecture for Fast Intra-Domain Mobility. IEEE PERSONAL Communications, TBD, Aug. 2000.

[22] P. Druschel, M. B. Abbott, M. Pagels, and L. L. Peterson. Network subsystem design. *IEEE Network (Special Issue on End-System Support for High Speed Networks)*, 7(4):8–17, July 1993.

[23] Peter Druschel, Mark B. Abbott, Michael A. Pagals, and Larry L. Peterson. Network subsystems design. *IEEE Network*, 7(4):8–17, 1993.

[24] Markus Endler and Vera Nagamuta. General Approaches for Implementing Seamless Handover. In *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 17–24, New York, NY, USA, 2002. ACM Press.

[25] H. Soliman et. al. IP Mobility Support for IPv4. RFC 4140, August 2005.

[26] R. Ramjee et al. HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks. In *Proc. International Conf. Network Protocols*, November 1999.

[27] R. Woundy et. al. Dynamic Host Configuration Protocol (DHCP). Internet Dratf, draft-elmalki-mobileip-fast-handoffs-03.txt, March 1997. Internet Draft - Work in Progress.

[28] S. Deering et. al. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998.

[29] Universal Mobile Telecommunications System (UMTS); Services & Services Capabilities. ETSI TS 122 105 (2002-06) v. 5.2.0, Release 5.

[30] Mohamed E. Fayad, Douglas C. Shimidt, and Ralph E. Johnson. *Building Application Frameworks*. Wiley Computer Publishing, 1999.

[31] G. Fleming, A. El-Hoiydi, J. DeVriendt, G. Nikolaidis, F. Piolini, and M. Maraki. A flexible Architecture for UMTS. *IEEE Personal Communications Magazine*, April 1998.

[32] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley Pub Co, 1995.

[33] B. Garbinato and R. Guerraoui. Using the Strategy Design Pattern to Compose Reliable Distributed Protocols. In *Proceedings of the 3rd Conference on Object-Oriented Technologies and Systems (COOTS-3)*, pages 221–232, Portland, Oregon, USA, June 1997.

[34] B. Garbinato and R. Guerraoui. Flexible Protocol Composition in Bast. In *Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS-18)*, pages 22–29, Amsterdam, The Netherlands, May 1998. IEEE Computer Society Press.

[35] B. Garbinato and R. Guerraoui. An open framework for reliable distributed computing. *ACM Computing Surveys*, 32(1), March 2000.

[36] E. Gustafsson, A. Jonsson, and C. Perkins. Mobile IP Regional Registration, March 2000. Internet Draft, draft-ietf-mobileip-reg-tunnel-02.

[37] A. Helmy, M. Jaseemuddin, and G. Bhaskera. Efficient Micro-Mobility using Intra-domain Multicast-based Mechanism (M&M). In *ACM SIGCOMM Computer Communications Review*, 2002.

[38] O. Hilarie, E. Sean, and O. Larry. A Fast and General Implementation of Mach IPC in a Network. In Usenix Association, editor, *Proceedings of the USENIX Mach III Symposium*, pages 75–88, 1993.

[39] N. C. Hutchinson and L. L. Peterson. The x-Kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering*, 17, 1991.

[40] K. Kim, C-K. Kim, and T. Kim. A Seamless Handover Mechanism for IEEE 802.16e Broadband Wireless Access. In Springer-Verlag, editor, *Lecture Notes in Computer Science*, pages 527–534. 2005.

[41] R. Koodli. Fast Handovers for Mobile IPv6. RFC 4068, July 2005.

[42] Yi-Bing Lin and Imrich Chlamtac. *Wireless and Mobile Network Architectures*. John Wiley & Sons, Inc, 2001.

[43] Juntong Liu, Gerald Q. Maguire Jr., and George Liu. Enhancing the Efficiency and Reliability of Handover and Routing Performance in Wireless Mobile Internetworking Environments. In *Proceedings of the 2nd Workshop on Personal Wireless Communication (Wireless Local Access)*, Frankfurt, December 1996.

[44] K. El Malki and H. Soliman. Fast Handoffs in Mobile IPv4. Internet Dratf, draft-elmalki-mobileip-fast-handoffs-03.txt, September 2000.

[45] D. A. Maltz and P. Bhagwat. MSOKCS: An Arquitecture for Transport Layer Mobility. In *Proc. of IEEE INFOCOM'98*, pages 1037–1045, San Francisco, CA, USA, April 1998.

[46] J. Manner and M. Kojo. Mobility Related Terminology. RFC 3753, June 2004.

[47] S. Mishra, L. L. Peterson, and R. D. Schlichting. Consul: A communication substrate for fault-tolerant distributed programs. *Distributed Systems Engineering Journal*, 1993.

[48] V. Nagamuta and M. Endler. Simulando um Protocolo Confiável para Clientes Móveis baseado em Proxies Móveis. In *III Workshop de Comunicação Sem Fio e Computação Móvel (WCSF2001)*, Igarassú, PE, August 2001.

[49] Basavaraj Patil, Yousuf Saifullah, Stefano Faccin, Srinivas Sreemanthula, Lachu Aravamudhan, Sarvesh Sharma, and Risto Mononen. *IP in Wireless Networks*. Prentice Hall, 2003.

[50] C. Perkins and K. Y. Wang. Optimized Smooth Handoffs in Mobile IP. In *ISCC '99: Proceedings of the The Fourth IEEE Symposium on Computers and Communications*, page 340, Washington, DC, USA, 1999. IEEE Computer Society.

[51] C. E. Perkins and D. E. Johnson. Route Optimization in MobileIP. MobileIP Working Group, Internet Draft - work in progress, November 1997.

[52] Charles E. Perkins. Mobile IP. *IEEE Communication Magazine*, May 1997.

[53] Liesbeth Peter, Ingrid Moerman, Bart Dhoedt, and Piet Demeester. Influence of topology on the performance of micromobility protocols. In *Proceedings of WiOpt'03*, pages 287–292, Sophia Antipolis, France, March 2003.

[54] L. Peterson, N. Hutchinson, S. O'Malley, and M. Abbott. RPC in the x-Kernel: evaluating new design techniques. In *SOSP '89: Proceedings of the twelfth ACM symposium on Operating systems principles*, pages 91–101, New York, NY, USA, 1989. ACM Press.

[55] Mauro Nacif Rocha. *Simulação e Gerenciamento de Unidades Móveis em Ambientes de Comunicação sem Fio*. PhD thesis, Dept. of Computer Science, Universidade Federal de Minas Gerais, Abril 2001.

[56] Ricardo C.A. Rocha. MobiCS Home Page. http://www.lcpd.ime.usp.br/~mobics/. (Last visited on July 2005).

[57] T. Narte S. Thomson. IPv6 Stateless Address Autoconfiguration. RFC 2462, December 1998.

[58] D. Schmidt. The ADAPTIVE Communication Environment: an Object-Oriented Network Programming toolkit for developing Communication Software. In *Proc. of 11th Sun Users Group Conference*, December 1993.

[59] Douglas C. Schmidt. ASX: An object-oriented framework for developing distributed applications. In *Proceedings of the 6 th USENIX C++ Technical Conference*, pages 207–226, 1994.

[60] Douglas C. Schmidt, Michael Stal, Hans Rohnert, and Frank Buschmann. *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. Wiley & Sons, 2000.

[61] Douglas C. Schmidt and Tatsuya Suda. An Object-Oriented Framework for Dinamically Configuring Extensible Distributed Systems. *BCS/IEE Distributed Systems Engineering Journal (Special Issue on Configurable Distributed Systems*, 1995.

[62] S. Seshan, H. Balakrishnan, and R. H. Katz. Handoffs in cellular wireless networks: The daedalus implementation and experience. *Kluwer International Journal on Wireless Personal Communications*, January 1997.

[63] James D. Solomon. *Mobile IP: The Internet Unplugged*. Prentice Hall, 1998.

[64] William Stallings. *Wireless Communications and Networking*. Prentice Hall, 2001.

[65] Chai-Keong Toh. A hybrid handover protocol for local area wireless ATM networks. *Mob. Netw. Appl.*, 1(3):313–334, 1996.

[66] A. G. Valko. Cellular IP: A New Approach to Internet Host Mobility. In *Comp. Commun. Review*, volume 29, pages 50–65. January 1999.