

Seamless Handover em Redes Móveis e sem Fio

MAC5701 - Tópicos em Ciência da Computação

Relatório Final

Vera Nagamuta

nagamuta@ime.usp.br

23 de junho de 2003

1 Introdução

Neste relatório apresentamos um estudo sobre *Seamless Handover* em redes móveis sem fio realizado durante o corrente semestre para a disciplina “Tópicos em Ciência da Computação” de acordo com o plano de estudos apresentado no início do semestre.

O principal objetivo deste trabalho é apresentar diferentes estratégias/protocolos para *seamless handover* (em particular, para micro-mobilidade) e comparar essas estratégias apresentando as vantagens e desvantagens de cada uma delas. Além disso, como resultado desse estudo, apresentamos um esboço de um *framework* que, a partir de determinados requisitos de QoS de um certo tipo de aplicação deve devolver um esquema de *seamless handover* específico. Este *framework* é baseado na composição de módulos canônicos - estratégias básicas de *handover*.

As próximas seções estão organizadas da seguinte forma: na seção 2, apresentamos o problema do *handover* em redes celulares, as dificuldades para se alcançar *seamless handover* e citamos alguns tipos de *handover*. Em seguida, na seção 3, apresentamos uma classificação dos possíveis significados de *seamless* em termos de propriedades quantitativas identificadas em serviços de rede/transporte. Na seção 4, apresentamos algumas classes de aplicações e seus parâmetros de QoS requeridos (valores esperados) que garantem o seu bom desempenho. A seguir, na seção 5, apresentamos as abordagens/protocolos existentes para *seamless/fast handover* para micro-mobilidade em redes móveis sem fio e uma comparação entre essas abordagens. Finalmente, na seção 6, apresentamos um *framework* para prover *seamless handover* em particular para micro-mobilidade em redes móveis sem fio.

2 O Problema do Handover

Handover é um procedimento requerido em redes celulares para manter a conexão com o dispositivo móvel (que chamaremos de *host móvel*) enquanto este se movimenta e ultrapassa os limites de uma célula. As tarefas desse procedimento são, basicamente, a atualização da localização do *host móvel*, redirecionamento de pacotes destinados ao mesmo e a reconfiguração do *path* de roteamento.

A fim de prover uma migração transparente ao usuário e às aplicações, esse procedimento deve ser rápido, com nenhuma ou uma mínima perda de dados, de modo que o fornecimento de serviço ao *host móvel* seja “ininterrupto” tendo em vista os requisitos da aplicação (este processo é chamado de *seamless handover* ou “handover suave”).

Seamless Handover é um tópico que tem merecido uma especial atenção por parte das pesquisas em Computação Móvel e ubíqua nos últimos anos. Devido à crescente proliferação de dispositivos móveis de computação e a demanda cada vez maior por aplicações e/ou serviços que envolvem não apenas voz, mas dados de imagem, vídeo, som, ou seja, aplicações de tempo real e interativas, é preciso oferecer uma infra-estrutura de software e hardware que, de fato, dê suporte a esses tipos de aplicações com uma qualidade próxima à que existe em redes fixas tradicionais.

As dificuldades em se prover *seamless handover* estão nas características inerentes à forma de comunicação através do meio sem fio (interface aérea) em que se baseiam as redes móveis sem fio:

- a comunicação através do meio sem fio é não confiável, sujeita a interferências, erros e desconexões repentinas;
- a largura de banda associada a cada conexão em uma célula é limitada e está sujeita a frequentes variações dependendo da frequência e número de usuários que entram e saem da célula;
- heterogeneidade: diferentes tecnologias de rede/arquiteturas/padrões de acesso/velocidade de transmissão/tamanho de célula/protocolos de gerenciamento de localização e de gerenciamento de *handover* de redes móveis sem fio.

Devido a essas restrições, durante o procedimento de *handover* podem ocorrer perda de dados e atrasos na transferência de dados entre a rede e o *host móvel* e vice-versa. Além disso, a característica de heterogeneidade impõe outras dificuldades para prover *seamless handover* entre redes e distintos domínios.

Na próxima subseção, apresentamos alguns tipos de *handovers*.

2.1 Tipos de handovers

A classificação de *handover* pode ser feita 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* [13]:
 - **micro-handover** (*in-LAN handover*): o *handover* é executado entre estações base na mesma subrede;
 - **macro-handover** (*cross-LAN handover*): o *handover* é executado entre estações base em subredes distintas.

Cáceres e Padmanabhan [3] usam o termo **local handover** para designar **micro-handover** e divide **macro-handover** em duas subclasses: **regional handover**, *handover* entre estações base relativamente próximas mas não necessariamente na mesma subrede (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:

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

3. De acordo com a forma de conexão do elemento móvel com as estações base [10]:

- **soft handover**: o *host móvel* tem conexão com a estação base da nova célula sem interromper a comunicação com a antiga estação base;
- **hard handover**: a conexão com a antiga estação base é interrompida e então uma conexão é estabelecida na nova estação base.

Os termos comumente utilizados são *handover horizontal* e *handover vertical*. Protocolos de *handover* geralmente estão embutidos em protocolos de suporte à mobilidade que, além do gerenciamento de *handover*, possuem outros mecanismos como gerenciamento de localização, roteamento de pacotes, autenticação/segurança, etc. No caso de mobilidade local/regional, em uma mesma subrede ou domínio administrativo, esses protocolos são chamados de **protocolos de micro-mobilidade** e, geralmente, incluem um protocolo de *handover horizontal* para tratar o *handover* quando há mudança de células. No caso de mobilidade global, os protocolos são chamados de **protocolos de macro-mobilidade**.

Na seção 5, apresentamos o Mobile IP [17] que é um protocolo que dá suporte à micro e macro-mobilidade de usuários. Conforme veremos, embora este protocolo tenha um bom desempenho para macro-mobilidade, no caso de micro-mobilidade, possui algumas deficiências que degradam seu desempenho. Apresentamos, também, protocolos propostos para melhorar o Mobile IP no caso de micro-mobilidade.

3 Significados de Seamless

O conceito de *seamless* (“suavidade”) pode ter diferentes significados. Esses significados podem depender, por exemplo, do tipo de aplicação, de seus requisitos (de QoS) e da capacidade de um protocolo/serviço em satisfazer essas propriedades. Embora, possivelmente, existam

outras formas para se classificar esse conceito, neste trabalho nos basearemos, especificamente, em determinadas propriedades de serviços de rede/transporte.

Serviços de rede/transporte podem ser divididos, basicamente, em dois grupos:

- **serviços orientados à conexão:** requerem que as aplicações estabeleçam uma conexão lógica entre seus pares (fonte e destino) antes da troca de mensagens. Provêm uma transferência de dados confiável, ordenada e com controle de fluxo, porém, com um alto *overhead* (devido à complexidade introduzida por essas características). Um exemplo de serviço orientado à conexão é o TCP, trata-se de um protocolo que garante entrega confiável entre sistemas similares ou não.
- **serviços não-orientados à conexão:** os dados são apresentados com um endereço destino e a rede os envia através da forma de envio *best-effort*. Não há nenhuma garantia em relação à perda de dados, atrasos, duplicações ou ordenação, as aplicações devem detectar e corrigir erros de transmissão. Apesar de não serem confiáveis, possuem menor *overhead* e, conseqüentemente, podem prover um melhor desempenho às aplicações. UDP é um exemplo de protocolo de entrega de datagramas (sem conexão) e com baixo *overhead*.

A partir das características desses dois tipos de serviços, identificamos algumas propriedades genéricas de uma comunicação que são úteis na classificação do conceito de *seamless* para diferentes tipos de aplicações. As propriedades são:

- **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;
- **Taxa de transmissão**, por exemplo, quantidade mínima de bits (bytes) por segundo.

Essas propriedades são expressas como valores máximos ou mínimos exigidos (ou esperados) pelas aplicações (requisitos de QoS). Um certo protocolo de *handover* é *seamless* (para uma determinada aplicação) se os valores esperados dos requisitos da aplicação são, de fato, satisfeitos pelo protocolo durante toda a execução da aplicação.

Em [7], apresentamos o conceito de *seamless handover* de uma maneira genérica e propomos uma classificação de algumas das possíveis abordagens para prover *seamless handover*.

Na próxima seção, apresentamos alguns tipos de aplicações e seus valores esperados para algumas das propriedades acima citadas.

4 Classes de Aplicações e Requisitos de QoS

Nesta seção, apresentamos três classes de aplicações que são diferenciadas, basicamente, pela forma de comunicação [9]: (1) conversação/tempo real, (2) serviços interativos e (3) serviços baseados em *streams*.

A seguir, apresentamos as características de cada uma dessas três classes e os valores esperados (na Tabela 1) para alguns parâmetros de QoS para algumas aplicações em cada uma dessas classes.

4.1 Serviços de Conversação/Tempo Real

Esta classe possui os mais fortes e estritos requisitos de QoS, é a única classe na qual esses requisitos são dados, estritamente, pelas percepções humanas. O ouvido humano é altamente intolerante à variação de atraso, porém, é tolerante em relação a alguma distorção no sinal devido à perdas. 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, jogos interativos, Telnet.

4.2 Serviços Interativos

Esta classe é caracterizada pelo esquema de comunicação de dados de tráfego interativo, 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 (o conteúdo dos pacotes devem ser transparentemente transferidos). Em relação 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: mensagens de voz, web-browsing (somente componentes HTML, não incluem componentes como imagens, áudio/vídeo clips, etc.), serviços transacionais (e-commerce) e correio eletrônico (E-mail).

4.3 Serviços baseados em streams

Essa classe se baseia em uma forma de transporte unidirecional (*one way*) de fluxo contínuo, com pequenos períodos de silêncio. 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. A variação do atraso aceitável neste caso é muito maior do que a variação de atraso dos limites da percepção humana. Há uma menor exigência em relação à taxa de perda de pacotes (taxa de pacotes perdidos ou corrompidos dentre todos os pacotes enviados). Exemplos de aplicações nesta classe: *audio streaming* (música), *one-way video* (vídeo clips, vídeo em tempo real), ftp.

Na Tabela 1, apresentamos os valores esperados (requisitos de QoS para aplicações e/ ou usuários finais) [9]. A tabela está dividida de acordo com as três classes referenciadas acima onde cada classe está separada por uma linha. Os parâmetros considerados são: **taxa de dados** (nas duas primeiras linhas da tabela há dois valores, o superior é o valor preferencial e o inferior é o valor limite), **atraso** (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 **perda de informações** (ou pacotes).

Conforme podemos observar, as aplicações da classe Conversação/Tempo Real possuem uma tolerância extremamente pequena em relação ao atraso (na ordem de milissegundos) e, portanto,

Aplicação	Taxa de dados	Atraso	Variação do atraso	Perda de dados
conversa�o/ voz	4-25 kb/s	< 150 ms < 400 ms	< 1 ms	< 3%
videofone	32-384 kb/s	< 150 ms < 400 ms	< 1 ms	< 1%
jogos interat.	< 1 KB	< 250 ms	N.A.	zero
Telnet	< 1 KB	< 250 ms	N.A.	zero
msg de voz	4-13 kb/s	< 1 s	< 1 ms	< 3%
web-browsing		< 4 s/p�g.	N.A.	< 3%
e-commerce		< 4 s	N.A.	zero
E-mail		< 4 s	N.A.	zero
audio stream (m�sica)	5-128 kb/s	< 10 s	< 2 s	< 1% taxa de perda pacote
video stream (v�deo clip)	20-384 kb/s	< 10 s	< 2 s	< 2% taxa de perda pacote
ftp	< 384 kb/s	< 10 s	N.A.	zero

Tabela 1: Valores esperados de aplica es/usu rio final

garantir a qualidade dessas aplica es   uma tarefa muito complexa devido, principalmente,   instabilidade do ambiente sem fio. J  no caso de aplica es interativas e aquelas baseadas em *streams* esse requisito   mais relaxado (na ordem de segundos).

Conforme veremos, dependendo do mecanismo de *handover* e de suas estrat gias empregadas para tratar o redirecionamento de pacotes e de atualiza es (de localiza o e *path*), o protocolo de *handover* pode ter maior ou menor lat ncia e perda de dados. Na subse o 5.7 fazemos uma compara o dos protocolos e suas estrat gias nesse sentido.

5 Estrat gias/Protocolos de Handover Horizontal

Nesta se o, apresentamos algumas estrat gias e protocolos de *handover horizontal*. Inicialmente, apresentamos o Mobile IP [17] e algumas extens es propostas. Em seguida, apresentamos alguns protocolos e suas estrat gias que visam melhorar o desempenho do Mobile IP no caso de micro-mobilidade. Esses protocolos empregam o pr prio Mobile IP no caso de macro-mobilidade.

5.1 Mobile IP

Mobile IP [17, 15, 18]   um protocolo desenvolvido pela IETF (Internet Engineering Task Force) para dar suporte   mobilidade de usu rios. Um *host m vel*   chamado de Mobile Node (MN) e a sua principal caracter stica   que o mesmo pode se mover (e mudar a sua localiza o) mantendo o mesmo endere o IP. Isso   poss vel pois o MN possui dois endere os IP: um deles indica a sua identifica o e o outro, a sua corrente localiza o na rede. O primeiro endere o (*home address*)   imut vel e est  associado ao seu Home Agent (HA) que representa o ponto de acesso (estac o base, roteador) na *home network* (rede que possui o mesmo prefixo de rede do

home address do MN). O segundo endereço, chamado de *care-of-address* (coa), é o endereço do Foreign Agent (FA) ou do ponto de acesso na rede onde o MN está visitando (*foreign network*). Um nó em comunicação com um MN é chamado Correspondent Node (CN).

O *handover* ocorre quando o MN detecta que está em uma *foreign network*, isso é feito através da comparação de mensagens enviadas por HAs e FAs (Agent Advertisements). O MN obtém o coa a partir da própria mensagem recebida do FA ou a partir de algum outro mecanismo (DHCP [6]) e registra esse novo endereço no HA.

Quando um pacote para o MN chega em sua *home network*, o HA o intercepta e envia o pacote para o coa através de encapsulamento (tunelamento). O FA desencapsula o pacote e o envia para o MN. Devido à essa forma de endereçamento, o Mobile IP possui as seguintes problemas:

- **triangle routing**: enquanto o MN envia pacotes através do FA por um caminho ótimo para um CN, todos os pacotes destinados ao MN devem passar pelo HA, através de um caminho que pode ser muito longo se o HA estiver muito distante do MN e o MN próximo do remetente;
- **perdas de pacotes**: pode ocorrer quando um MN deixa uma área antes de fazer o registro, fazendo com que os pacotes destinados ao mesmo sejam enviados ao antigo coa;
- **sobrecarga**: quando a frequência de *handovers* é grande, um grande número de mensagens de registro precisam percorrer a rede;
- **suporte a QoS**: a cada mudança de localização do Mh é preciso fazer uma nova negociação de QoS.

Para aliviar esses problemas foram propostas as seguintes extensões/melhorias:

- **Mobile IP Route Optimization** [16]. A idéia básica é informar o CN e o antigo FA sobre o novo coa do MN para que estes enviem pacotes diretamente ao MN. Devido ao fato de que a nova localização do MN deve ser constantemente atualizada no HA e em todos os nós que correntemente estão se comunicando com o MN, essa estratégia aumenta o tráfego na rede e latência. Portanto, essa abordagem não escala bem para um grande número de MNs movimentando-se freqüentemente entre as células de uma rede.
- **Gerenciamento hierárquico de FAs**. FAs em um mesmo domínio são organizados em uma hierarquia (árvore de FAs) para tratar movimentos locais dentro do domínio, evitando-se que atualizações sejam feitas no HA toda vez que o MN muda de endereço. As mensagens Agent Advertisement de um FA deve conter os coa de todos os FAs ancestrais mais o próprio endereço. O MN se registra no HA passando essa lista de coas e os pacotes direcionados ao MN são enviados através da hierarquia de FAs (por tunelamento, re-tunelamento, ...). Quando ocorre um *handover* o MN compara a nova lista de coas com a antiga lista e encontra o FA que aparece nas duas listas e que está no nível mais baixo possível nas duas hierarquias. Esse FA é notificado fazendo com que atualize a localização do correspondente MN e, o HA assim como os FAs, acima deste FA não precisam ser notificados. Um dos problemas com esse mecanismo é manter a eficiência na transferência de dados uma vez que

os mesmos possivelmente tenham que passar por vários túneis. Também possui problemas de falta de mecanismo de tolerância a falhas de FAs e dificuldade para prover QoS.

- **Buffering.** O FA armazena todos os pacotes enviados ao MN em um *buffer*. Quando o FA recebe uma notificação de mudança, envia os pacotes *bufferizados* para o novo FA. Essa estratégia reduz o problema de perda de pacotes mas, se o MN demorar muito tempo para encontrar um novo FA após ter perdido contato com o antigo FA, pode ocorrer *buffer overflow*. Além disso, a duplicação de pacotes degrada o desempenho e desperdiça recursos. Duplicação de pacotes é eliminada através do seguinte procedimento: o MN ao receber um pacote (datagrama), armazena o endereço e a identificação do datagrama em um buffer, quando o MN requisita um *handoff* para um FA, envia o conteúdo do buffer e esse conteúdo é passado para o antigo FA que filtra o buffer de pacotes enviados ao MN, eliminando aqueles que já foram recebidos.

Apesar dessas extensões, o Mobile IP não é apropriado para situações em que há frequentes migrações de *hosts móveis* por pequenas células (em uma área geográfica limitada) devido à latência causada pelos frequentes procedimentos de *handover* a um HA (possivelmente distante), causando uma degradação no desempenho da aplicação. Mesmo com o mecanismo de gerenciamento hierárquico de FAs, no pior caso, o FA nas duas listas de coas é o FA raiz, fazendo com que todos os FAs na árvore tenham que atualizar a localização do MN.

5.2 Estratégias de Handover Horizontal

Vários trabalhos têm sido propostos para melhorar o desempenho do *handover* no Mobile IP, em particular, para o caso de micro-mobilidade [2]. Diferentes estratégias são propostas: gerenciamento hierárquico de mobilidade, métodos para atualização de *paths* de roteamentos, *multicasting*, *bicasting*, *buffering*, *forwarding*, etc. Os protocolos de micro-mobilidade empregam uma ou mais dessas técnicas e objetivam, essencialmente, diminuir a latência e minimizar a perda de pacotes causados por frequentes *handovers* em uma área geográfica limitada.

Muitos protocolos de micro-mobilidade adotam a estratégia de gerenciamento hierárquico de mobilidade [1, 8, 3] observando o fato que, em grande parte, as migrações de usuários móveis se dão dentro de uma mesma subrede ou domínio (administrativo). Assim, como a idéia de gerenciamento hierárquico de FAs, as migrações dentro de um domínio são transparentes para os *hosts* correspondentes e para os HAs e, a grande vantagem dessa estratégia é a redução do número de atualizações de endereço dos *hosts móveis* em seus respectivos HAs. Isto é, enquanto um *host móvel* migra dentro de um domínio nenhuma atualização é enviada ao HA, o HA somente é notificado de atualização de endereço quando o *host móvel* migrar de um domínio para outro. Isso agiliza o processo de *handover* uma vez que as atualizações são apenas locais, evitando o seu percurso até um possivelmente distante HA. Porém, essa técnica não evita o problema de perda de dados.

A técnica de *buffering* é utilizada para reduzir a perda de dados, mantendo os pacotes no *buffer* durante o processo de *handover* para serem direcionados (*forwarding*) para a nova estação base. Algumas questões em relação à utilização de *buffers* é decidir o tamanho do *buffer* e quando remover pacotes que não são mais necessários para evitar que o *buffer* cresça indefinidamente e cause uma *overhead* no sistema.

Esquemas de atualização de *path* são usados para reduzir a latência no redirecionamento de pacotes [1, 8], porém, em alguns protocolos, é requerido que o *host móvel* tenha acesso (capacidade de comunicação) com mais de uma estação base ao mesmo tempo, o que nem sempre é possível garantir.

Multicast é uma forma de entrega de mensagens que permite o envio de dados a um grupo de destinatários independentemente da localização de cada um deles. Essa idéia tem sido empregada em diversos trabalhos para dar suporte à mobilidade [12, 11, 14]. As principais vantagens desse mecanismo são a redução da latência e a perda de pacotes durante o procedimento de *handover*. A principal idéia é antecipar o *handover* replicando pacotes para estações base vizinhas, permitindo, dessa forma, um rápido redirecionamento de dados (e reconfiguração do *path*) no momento do *handover*. O problema do *multicasting* é o *overhead* causado pelo (possivelmente) grande número de réplicas e do tráfego na rede.

Alguns protocolos usam *bicasting* em vez de *multicasting*, replicando pacotes apenas para a futura estação base, reduzindo o *overhead*, porém com uma maior susceptibilidade a perdas. Porém, em ambos os casos, existe o problema de sincronização entre as estações base que recebem os pacotes, pois cada uma delas pode estar recebendo pacotes em tempos diferentes dependendo do tráfego na rede em direção das estações base.

Nas próximas subseções apresentamos alguns abordagens/protocolos de micro-mobilidade baseados no Mobile IP que empregam as estratégias citadas acima.

5.3 Cellular IP

O Cellular IP [1] é um protocolo para micro-mobilidade desenvolvido pela Universidade de Columbia que se baseia na abordagem hierárquica e tem como principais objetivos oferecer um rápido *handover* com baixo atraso e com mínima ou zero perda de pacotes.

Redes Cellular IP são constituídas por: *gateways*, que fazem a conexão de domínios com a Internet e fazem (implicitamente) o papel de *foreign agent* e por *base stations* (BS), que são roteadores especiais para dar suporte à mobilidade e oferecer pontos de acesso sem fio aos *hosts móveis*.

Hosts móveis usam o endereço do *gateway* como sendo seu *coa*, que é registrado no HA. Dentro do domínio, *hosts móveis* são identificados pelo seu *home address*. Pacotes destinados ao *host móvel* são, como usualmente, enviados para o HA e tunelados para o *gateway*. O *gateway* desencapsula os pacotes e os direciona ao *host móvel* roteando-os através das BSs, porém, sem tunelamento. Pacotes do *host móvel* são inicialmente enviados ao *gateway* e em seguida, para a Internet.

As BSs possuem um *soft-state routing cache* que mantém informações que permitem o roteamento de pacotes. Essas informações incluem, tanto mapeamentos do tipo $\langle \textit{host móvel}, \textit{next hop} \rangle$ para rotear pacotes do *gateway* para o *host móvel* (*downward*) como também indicações para rotear pacotes do *host móvel* para o *gateway* (*upward*). Essas informações são frequentemente atualizadas pelo *gateway* através da difusão de *beacons* (caso *upward*) e pelos pacotes dos próprios *hosts móveis* transmitidos à Internet ou, na falta desses, por pacotes específicos (*route update packets*) que são transmitidos pelos *hosts móveis* em um certo intervalo de tempo. Devido ao fato de que os caches nas BSs são *soft-state*, as informações de roteamento de *hosts móveis* que migraram são automaticamente removidas depois de um certo *timeout*. Essa técnica

aumenta a robustez do protocolo em relação a falhas nos roteadores e nos *links* e, além disso, evita que o estado nos roteadores cresça indefinidamente.

O mecanismo de *handover* e de gerenciamento de localização estão baseados no próprio mecanismo de atualização do cache. O *host móvel* mede os sinais (*beacons*) transmitidos pelas BSs, ao perceber um sinal mais forte, ele envia o pacote *route update* (o mesmo para a atualização do cache) para a BS a fim de iniciar o processo de *handover*. A BS recebe o pacote e o repassa para a próxima BS (de acordo com a informação em seu *routing cache*). Esse processo é repetido até que o pacote chegue em uma BS cujo cache já continha uma entrada para o *host móvel* (*crossover base station*), no pior caso, é o próprio *gateway*. Desta forma, o *path* para roteamento de pacotes ao *host móvel* é construído/atualizado.

Existem dois tipos de *handover* suportados pelo Cellular IP:

- *hard handover*: o *host móvel* perde a conexão com a antiga BS ao iniciar o *handover* com a nova BS. Esse método reduz a latência do processo de *handover* mas não evita a perda de pacotes. Mesmo empregando um mecanismo para o redirecionamento de pacotes da antiga BS para a nova, enquanto a antiga BS não recebe de fato uma notificação de que o *host móvel* migrou, esta estará enviando pacotes para o *host móvel* e estes serão perdidos. Somente após o recebimento da notificação é que os pacotes que chegarem poderão ser encaminhados para a nova BS.
- *semi-soft handover*: o *host móvel* antes de iniciar o *handover* envia uma mensagem à nova BS para que esta antecipe o processo de *handover* estabelecendo o *path* de roteamento antes que o *host móvel* inicie o *handover*, fazendo com que os pacotes sejam entregues às duas BSs. Nesta abordagem, existe o problema da sincronização entre os pacotes enviados para as duas BSs, em casos em que há maior tráfego em um dos *paths*. Isto é solucionado com o emprego de *buffers* para manter os pacotes por um certo intervalo de tempo no *gateway* e BSs. Outra restrição dessa abordagem é que o *host móvel* deve ter acesso (capacidade de comunicação) às duas BSs simultaneamente (embora não precise estar simultaneamente conectado a ambas).

5.4 HAWAII

HAWAII (Handoff-Aware Wireless Access Internet Infrastructure) [8] é uma abordagem para dar suporte à micro-mobilidade intra-domínio.

De forma semelhante ao Cellular IP, o HAWAII também é baseado em gerenciamento hierárquico, possui um *gateway* (*domain root router*) que faz a conexão de um domínio com a Internet. Porém, em vez de roteadores específicos para gerenciamento de mobilidade como no Cellular IP, o HAWAII estende os roteadores IP tradicionais adicionando funcionalidades para dar suporte à mobilidade, porém, as informações para o roteamento de pacotes a *hosts móveis* são mantidas de forma semelhante ao esquema do Cellular IP. Uma diferença é que o HAWAII disponibiliza diferentes métodos para a atualização do *path* no momento do *handover*.

Cada *host móvel* possui um endereço IP e um *home domain*, que são associados no momento em que o *host móvel* se conecta pela primeira vez a um domínio HAWAII. Enquanto o *host móvel* se move dentro do domínio, ele mantém esse endereço IP. Pacotes destinados ao *host móvel* são direcionados ao *domain root router* baseados no endereço da subrede do domínio e são enviados

ao *host móvel* através de *path* de roteamento previamente estabelecidos. Está implícito, nesta abordagem, que o *domain root router* faz o papel do HA. Portanto, enquanto as migrações são intra-domínio, não há encapsulamento ou tunelamento para o roteamento de pacotes aos *hosts móveis* no domínio, somente no caso de migração intra-domínio.

Assim como no Cellular IP, as informações de roteamento (*forwarding entries*) contidas nos roteadores são *soft-state* e precisam ser constantemente atualizadas (*refreshing*). Porém, o processo de *handover* no HAWAII é independente dessas atualizações.

O *host móvel* inicia o *handover* enviando uma mensagem de registro à nova estação base informando qual era sua a antiga estação base. O redirecionamento (ou não) de pacotes e a reconfiguração do *path* depende do esquema de reconfiguração de *path* adotado. Basicamente, há dois esquemas básicos de reconfiguração de *path* que diferem, basicamente, na forma de atualização dos *forwarding entries* e na forma de executar o redirecionamento de pacotes (*crossover router* é o roteador mais próximo do *host móvel* que está na interseção entre o antigo e o novo *path*):

- *forwarding scheme*: neste esquema, os *forwarding entries* são atualizados a partir da antiga estação base “subindo” através dos roteadores intermediários até o *crossover router* e “descendo” até a nova estação base. Os pacotes que chegam na antiga estação base após a atualização de seu *forwarding entry* são direcionados para a nova estação base, porém, pode haver perda de dados durante o intervalo de tempo em que a mensagem de notificação de *handover* trafega até a antiga estação base (uma vez que o *handover* é iniciado na nova estação base através da mensagem de registro enviado pelo *host móvel*). Este esquema é voltado para situações em que é tolerável alguma perda de pacotes e quando não é possível manter a conexão com ambas estações base (*hard handover*).
- *non-forwarding scheme*: ao contrário do esquema anterior, neste esquema, os *forwarding entries* são atualizados a partir da nova estação base (assim que recebe a mensagem de registro do *host móvel*) “subindo” através dos roteadores intermediários até o *crossover router* e “descendo” através dos roteadores até a antiga estação base, que é a última a atualizar o seu *forwarding entry*. Neste caso, pacotes podem ser direcionados para o *host móvel* a partir do momento em que a atualização é feita no *crossover router* e é particularmente apropriado quando é possível a conexão com ambas estações (*soft handover*). As perdas de dados são reduzidas uma vez que o *host móvel* pode receber pacotes de ambas estações base. Porém há problemas de sincronização entre as estações base, duplicações no *host móvel* e capacidade do *host móvel* em “escutar” ambas estações base. .

5.5 Fast and Scalable Handoffs

Nesta seção, apresentamos um outro esquema baseado em hierarquias proposto por Cáceres e Padmanabhan [3]. A diferença com as outras abordagens é que esta possui um nível a mais na hierarquia:

1. local: a migração é entre estações base na mesma subrede (que cobre um edifício, por exemplo) conectadas por uma LAN (por ex. Ethernet). O protocolo proposto confina as mensagens na subrede local e opera de forma transparente as outros níveis da hierarquia.

2. intra-domínio: migração em um domínio, por exemplo, entre edifícios em um campus. A solução proposta é estender o Mobile IP, incluindo uma hierarquia de FAs, evitando atualizações no HA. Isso é feito da seguinte forma: considerando o caso de um campus, existem um ou mais FAs em cada subrede (*subnet* FAs) e, também existe um FA relativo ao domínio administrativo (*domain* FA). Os FAs da subrede incluem o endereço do *domain* FA em suas mensagens Agent Advertisement. Os MNs (habilitados a reconhecer esse tipo de mensagem) usam esse endereço como coa, que se mantém imutável enquanto o MN estiver dentro do domínio. Os FAs da subrede enviam qualquer mensagem Agent Solicitation que receberem de algum MN para o *domain* FA. O *domain* FA mantém uma entrada indicando o caminho (*host routing*) para cada MN em seu domínio e é atualizado quando ocorre uma migração.
3. global: quando a migração é entre subredes ou domínios administrativos. Este caso é tratado exatamente como no protocolo Mobile IP, sem alterações.

O protocolo de *handover* local é o seguinte: o *host móvel* inicia o *handover* ao perceber um sinal (*beacon*) mais forte e envia uma mensagem Greet para a nova estação base passando o seu endereço e o da antiga estação base. A nova estação notifica a antiga estação base fazendo com que esta direcione os pacotes recentemente enviados ao *host móvel* para a nova estação base. Esses pacotes são enviados para o *host móvel*. Tanto as estações base como o *host móvel* possuem mecanismos de *buffer* para armazenar as mensagens mais recentes a fim de possibilitar o seu re-envio em casos de *handover*.

Devido ao fato de que as estações base estão diretamente conectadas a uma rede local, a transmissão de mensagens e do conteúdo do buffer no momento do *handover* pela rede fixa é rápida, permitindo (segundo simulações apresentadas) o suporte a aplicações interativas como áudio.

5.6 Handover baseado em Multicast

Em [11] é apresentado um mecanismo baseado em multicast (M&M - Multicast-based Mechanism) para prover suporte à micro-mobilidade (intra-domínio). Aqui, as estações base são denotadas como Msss, a estação base na célula onde se encontra o *host móvel*, é chamada de Mss_resp (Mss responsável), o antigo Mss de Mss_old e o novo Mss de Mss_new. M&M usa multicast em dois níveis: um, para o roteamento de pacotes dentro do domínio (na rede fixa) até o atingir o Mss_resp e outro, para melhorar o desempenho do *handover*.

Assumindo que um *handover* é frequentemente executado entre Msss de células vizinhas, a principal idéia do mecanismo M&M é antecipar o procedimento de *handover* replicando pacotes aos possíveis candidatos a Mss_new (Msss cujas células são adjacentes à célula do atual Mss_resp). Isto permite uma rápida integração do novo Mss na árvore multicast e, possibilita a redução do tempo requerido para processar o *handover* assim como o número de pacotes perdidos durante o procedimento.

Quando um Mh entra em um novo domínio, é associado a ele um endereço unicast chamado de RCoA (Regional Care-of Address) pelo Mss responsável pela região de cobertura onde se encontra o Mh (Mss_resp). Através de um mapeamento algorítmico de endereço, o Mss_resp cria, a partir do RCoA um endereço multicast (Multicast Care-of Address - MCoA) e o associa ao Mh.

Esse endereço é único no domínio e é usado enquanto o Mh estiver dentro do domínio. Mss_resp dispara uma mensagem *Join(MCoA)* (implementada por um protocolo multicast) e estabelece a árvore de distribuição multicast. O próprio Mh registra no *home agent* (HA) o seu endereço unicast RCoA.

Pacotes destinados ao Mh são enviados ao seu *home address* e tunelados ao RCoA pelo HA. Quando chegam no domínio estrangeiro, esse pacotes são interceptados pelo BR e através do mapeamento de endereço RCoA para MCoA, os pacotes são difundidos pela árvore multicast. O Mss_resp modifica o endereço destino para RCoA e transfere os pacotes ao Mh.

5.6.1 Configuração inicial

Para melhorar o desempenho do procedimento de *handover* o protocolo assume a seguinte configuração:

- cada Mss mantém uma lista de Msss vizinhos (Msss cujas áreas de cobertura (células) são adjacentes à sua própria área de cobertura). Esse conjunto de Msss vizinhos é chamado de AM_set (Adjacent MSS set). Cada Mss possui um único AM_set. Na Figura 1, o AM_set de *Mss_1* (atual Mss_resp) corresponde ao conjunto de Msss 2 ao 7.

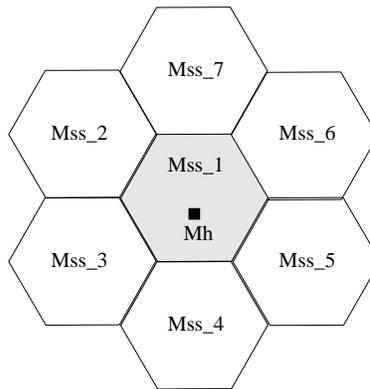


Figura 1: AM_set do MSS_1

- cada AM_set é associado a um endereço multicast específico e local chamado de AGA (AM_set Group Address), no qual os membros de um AM_set fazem parte desse grupo multicast. Isso possibilita que tais membros possam receber sinalizações multicast durante o *handover*.
- cada Mh pertence a um grupo multicast que é identificado pelo MCoA e é formado pelos roteadores em sua respectiva árvore de distribuição multicast e os Msss membros do AM_set relativo ao Mss_resp do Mh. Para manter a consistência do protocolo durante o *handover* cada Mss mantém uma tabela que indica seu *status* em relação à pertinência ou não desse Mss em cada grupo multicast existente (em relação aos Mhs na região do AM_set em que o Mss é um membro). Ou seja, existe uma linha na tabela de um Mss para cada Mh na região do AM_set desse Mss. Os possíveis estados para cada Mss são *joined* ou *left*.

A seguir, apresentamos um exemplo de tabela para um *Mss*. *MCoA* indica o grupo multicast, *MSS_resp* indica o atual *Mss* responsável pelo *Mh*, *CGA* indica o grupo multicast do *AM_set* a que pertence e, finalmente, o estado *Joined* indica que o *Mh* está na região coberta pelo seu *AM_set*.

MCoA	MSS_resp	CGA	State
Mh1	MSS_1	CGA_1	Joined
...

5.6.2 Mensagens do protocolo de *handover*

Durante o protocolo de *handover*, as seguintes mensagens são trocadas entre os *Mss*:

- $J(\text{MCoA})$: faz o receptor da mensagem entrar no grupo multicast do referido *MCoA*;
- $L(\text{MCoA})$: faz o receptor da mensagem sair do grupo multicast do referido *MCoA*;
- $\text{HO}(\text{RCoA}, \text{MCoA})$: mensagem trocada durante *handover* entre *Mss_old* e *Mss_new*.

5.6.3 Protocolo de *handover* intra-domínio

Vamos supor que um *Mh* está migrando de *Mss_1* para *Mss_5* (Figura 2). Quando a conexão é estabelecida entre o *Mh* e *Mss_5*, *Mss_5* faz um multicast da mensagem $J(\text{MCoA})$ para os membros de seu *AM_set* (*CGA_5*) requisitando que os mesmos entrem no grupo multicast *MCoA* do *Mh*. Em seguida, envia uma mensagem $\text{HO}(\text{RCoA}, \text{MCoA})$ para *Mss_1* para iniciar o processo *prune*. Ao receber a mensagem *HO*, o *Mss_1* faz um multicast da mensagem $L(\text{MCoA})$ para os membros de seu *AM_set* (*CGA_1*) requisitando que os mesmos saiam do grupo multicast *MCoA*.

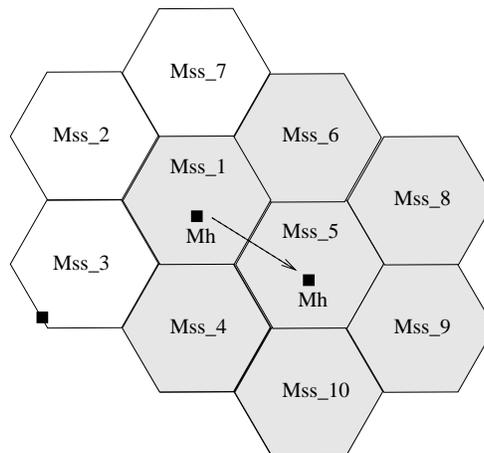


Figura 2: Handover entre *AM_sets*

A tabela (citada acima) em cada Mss é consultada toda vez que um Mss (que pertence a ambos AM_sets envolvidos no *handover*) recebe uma mensagem L(MCoA) para evitar que tal Mss se desligue do grupo em que, de fato, deveria pertencer. Isso poderia ocorrer quando tal Mss recebe uma mensagem L(MCoA) depois de uma mensagem J(MCoA), fazendo com que o mesmo deixe de receber réplicas dos pacotes destinados ao Mh. Para evitar esse problema, ao receber uma mensagem L(MCoA), o Mss verifica se o remetente da mensagem é o Mss_resp que corresponde ao MCoA na tabela (ou seja, o mesmo Mss que anteriormente requisitou a sua entrada em tal grupo multicast).

5.7 Comparação dos Protocolos de Micro-mobilidade

Nesta seção apresentamos uma comparação dos protocolos de micro-mobilidade descritos nas seções anteriores em relação ao desempenho do *handover* levando em consideração dois parâmetros: perda de dados e atraso.

O desempenho do *handover* depende do tempo que o mesmo consome desde que é iniciado pelo *host móvel* até o seu complemento. Este tempo corresponde à diferença entre o momento em que o *host móvel* recebeu o primeiro pacote na nova estação base e o momento em que o *host móvel* recebeu o último pacote da antiga estação base. Isso, por sua vez, depende do número de *hops* em que os pacotes precisam percorrer da antiga para a nova estação base (no caso de redirecionamento) ou, o número de *hops* entre o *crossover router* e a nova estação base.

Inicialmente, no Cellular IP, ambos os tipos de *handover* suportados (*hard handover* e *semi-soft handover*) não evitam completamente a perda de pacotes. No caso de *hard handover*, conforme vimos, mesmo que haja o redirecionamento de pacotes da antiga para a nova estação base, há um intervalo em que alguns pacotes são perdidos (enquanto a notificação da migração é encaminhada à antiga estação base). Para evitar esse problema, poderia ser empregado *buffer* na antiga estação base, porém, aumentaria a complexidade do protocolo para detectar a partir de quando a antiga estação base deve armazenar os pacotes. No caso de *semi-soft handover* é preciso garantir (de alguma forma) que o *host móvel* tem acesso (capacidade de comunicação) com ambas estações base, o que nem sempre é possível, somente em casos em que há interseção de células. Portanto, também nesse caso, não é garantida a redução da perda de pacotes.

Em relação ao desempenho, no caso de *hard handover*, o tempo requerido pelo *handover* depende do número de *hops* da antiga estação base (passando pelo *crossover router*) até a nova estação base e, no caso de *semi-soft handover*, depende do número de *hops* do *crossover router* até a nova estação base. No pior caso, o *crossover router* é o *gateway* e, dependendo da topologia da rede, o *gateway* pode estar muito distante (há muitos nós intermediários) até as estações base.

No HAWAII, o esquema *forwarding* perde dados enquanto a mensagem de atualização de *path* é encaminhada até a antiga estação base. No esquema *non-forwarding*, há uma menor perda, porém, assim como o protocolo *semi-soft* do Cellular IP, depende da estrutura da rede celular, se há ou não regiões de interseção.

Em relação ao desempenho, o esquema *non-forwarding* é mais rápido uma vez que os pacotes são direcionados a partir do *crossover router* durante o *handover*, enquanto que no esquema *forwarding*, os pacotes percorrem desde a antiga estação base, passando pelo *crossover router* até a nova estação base.

No caso do protocolo para *handover* local proposto por Cáceres, uma vez que todas as

estações base estão diretamente conectadas a uma rede LAN (de alta velocidade) é esperado que a latência do processo de *handover* seja bem menor do que nos casos anteriores, e, além disso, pacotes são armazenados e redirecionados permitindo que o número de perdas seja bem reduzido.

Handover baseado em *multicasting*, permite que a reconfiguração do *path* seja rápida e, conseqüentemente, impõe pouca latência no processo de atualização. Em relação às perdas, é possível que estas sejam minimizadas. Porém, nesta estratégia, assim como nas anteriores que envolvem réplicas, há uma preocupação com a sincronização entre os elementos recebendo as réplicas e com as duplicações. Dependendo da diferença de tempo em que as estações base recebem as réplicas (devido à diferença de tráfego nos *paths* entre as estações base), essa estratégia pode não ser tão vantajosa. Além do mais, existe a questão do alto *overhead*.

Concluindo, cada um desses protocolos de micro-mobilidade tem suas vantagens e desvantagens em relação às estratégias empregadas e o desempenho do protocolo de *handover* é influenciado por essas estratégias. Porém, conforme observamos, esses protocolos não são suficientes para prover *seamless handover* para diferentes tipos de aplicações com distintos (e possivelmente estritos) requisitos de QoS, conforme citamos na seção 4.

Na próxima seção apresentamos uma proposta de um *framework* para prover um mecanismo de *handover* a partir dos requisitos de QoS de uma aplicação.

6 Framework para a Geração de Mecanismos de Seamless Handover

Nesta seção, propomos um *framework* que permite a geração de mecanismos de *seamless handover* para alguns tipos de aplicações, de acordo com alguns requisitos de QoS dessas aplicações. Nas próximas seções, apresentamos a arquitetura e os principais componentes planejados do *framework*, os parâmetros de entrada e os resultados esperados. Uma visão geral da arquitetura deste *framework* pode ser vista na Figura 3.

6.1 Arquitetura e Componentes do Framework

O *framework* é composto por um conjunto de componentes, que chamamos de *módulos canônicos*, são técnicas que podem ser compostas para a formação de um mecanismo específico para prover *seamless handover* para um determinado tipo de aplicação. O tipo de aplicação é especificado previamente na instanciação do *framework* passando como parâmetros os valores esperados de alguns requisitos de QoS da aplicação (neste trabalho usaremos o termo *requisitos de QoS_h*, isto é os requisitos de QoS durante o processo de *handover*).

O gerenciador de módulos do *framework* recebe como entrada um conjunto de requisitos de *QoS_h* da aplicação e um conjunto de parâmetros que definem o perfil de mobilidade do usuário. De acordo com esses parâmetros, o gerenciador seleciona um conjunto de módulos (ou mecanismos). Esse conjunto de módulos implementa um é um possível esquema de *seamless handover* que supostamente “satisfaz” os requisitos fornecidos na entrada.

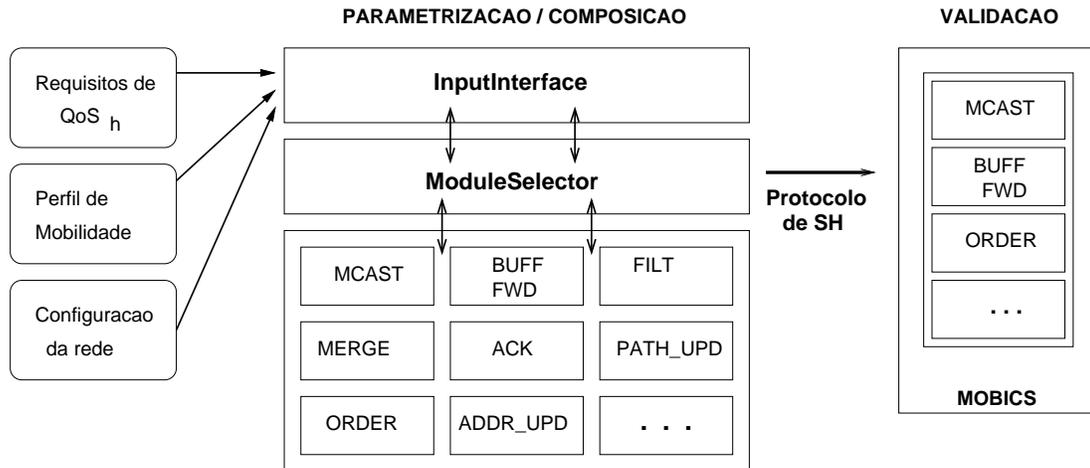


Figura 3: Framework para Seamless Handover

6.2 Parâmetros de Entrada

O *framework* recebe dois conjuntos de parâmetros:

- **Requisitos de QoS_h :** para cada tipo de aplicação, definiremos alguns possíveis requisitos de QoS_h (máximo atraso, máxima variação do atraso, percentagem aceitável de perdas de pacotes, se requer ou não ordenação, se permite ou não duplicações, etc.). De acordo com esses valores esperados e com os parâmetros definidos no perfil de mobilidade, os módulos são escolhidos e parametrizados. A agregação desses módulos representa um possível mecanismo de *seamless handover*.
- **Perfil de mobilidade:** especifica algumas características do padrão de mobilidade do usuário móvel (por exemplo, velocidade média de movimentação, probabilidades de migração para áreas cobertas por determinados Msss, etc.). O perfil de mobilidade pode ser útil, por exemplo, para facilitar a decisão da seleção por um determinado módulo e otimizar a utilização do mesmo, por exemplo, sabendo-se de antemão a provável direção do usuário, pode-se utilizar algum mecanismo para antecipar o *handover* na possível nova estação base (antecipação pode ser feita empregando *multicasting* ou *bicasting*, e neste caso em particular o uso de *bicasting* evita o *overhead* causado pela transmissão de réplicas).

6.3 Módulos Canônicos

Conforme vimos na seção 5.2, os protocolos de *handover horizontal* empregam diferentes estratégias que, de alguma forma estão embutidas nos protocolos de micro-mobilidade. Embora cada um desses protocolos tenha suas vantagens e desvantagens em relação às suas estratégias adotadas para prover suporte à mobilidade, esses protocolos não refletem diretamente os requisitos de QoS de diferentes tipos de aplicações. Diferentes tipos de aplicações possuem diferentes requisitos de QoS e, conseqüentemente, requerem diferentes tipos de soluções.

A principal idéia do *framework* é permitir uma combinação de estratégias básicas (módulos canônicos) de modo que a solução de *seamless handover* seja específica para cada tipo de aplicação e para os particulares parâmetros de entrada (requisitos de QoS e perfil de mobilidade).

A seguir, apresentamos alguns módulos inicialmente identificados:

- *Módulo Multicasting*: permite que o processo de *handover* seja antecipado, pacotes são replicados para estações base vizinhas. Reduz a latência para a atualização do *path* e redirecionamento de pacotes, pode ser utilizado quando há uma grande exigência em relação ao atraso máximo e perdas de dados. Um caso particular deste módulo é o *bicasting*, quando os pacotes são replicados apenas para a futura estação base. Possíveis parâmetros: tamanho máximo do grupo *multicast*.
- *Módulo Buffering&Forwarding*: *buffering* permite o armazenamento de pacotes em uma ou mais estações base para, posteriormente, serem redirecionados (*forwarding*). Essa técnica reduz a perda de pacotes porém, pode acarretar em algum atraso para o redirecionamento dos dados no *buffer*. Um dos parâmetros a serem fornecidos é o tamanho do *buffer*.
- *Módulo No_Duplication*: permite a detecção de duplicações, eliminando-as.
- *Módulo Ordering*: permite que o *host móvel* receba os pacotes em ordem.
- *Módulo ACK*: obtém Acks para cada pacote enviado, pode interagir com o módulo *Ordering* para a garantia de correto recebimento. Pode ser utilizado quando as perdas são intoleráveis pela aplicação.
- *Módulo Merging/Filtering*: permite a filtragem dos dados, por exemplo, em situações em que ocorre uma redução da largura de banda, alguns pacotes são eliminados (de acordo com um valor máximo, fornecido pelo usuário).
- *Módulo Agents*: pode ser empregado para monitorar a largura de banda disponível na atual célula e, possivelmente, nas células vizinhas. Pode ser utilizado para advertir outros módulos (por exemplo, *Merging/Filtering*) para modificar algum padrão de transmissão.

6.4 Validação do Framework

Pretendemos validar o *framework* através de simulações no ambiente de simulação de protocolos distribuídos MobiCS [5, 4]. O MobiCS possibilita a prototipagem, teste e avaliação de protocolos distribuídos e pode emular um ambiente de computação móvel no modo determinístico ou no modo estocástico. Para o *framework* proposto, nos interessa o modo de simulação estocástico, pois, neste modo o cenário de simulação é definido de acordo com o comportamento dos elementos de rede (estações base e *host móvel*) baseados na geração de eventos de acordo com parâmetros probabilísticos fornecidos pelo usuário para um modelo de simulação específico.

Isto é, para diferentes cenários, parâmetros de simulação (como, por exemplo, probabilidade de migração, frequência de envio de pacotes, carga nas células, etc.) e, possivelmente, topologias de rede, poderemos medir o número médio de pacotes perdidos, o tempo médio de atraso (tempo de duração do *handover*), e, possivelmente, a variação média do atraso. Esses resultados podem ser comparados aos requisitos de QoS_h iniciais (recebidos como parâmetros de entrada) a fim de comprovar a viabilidade do esquema gerado pelo *framework*.

6.4.1 Suposições e um Exemplo de Configuração da Rede

Para simular os esquemas gerados pelo *framework* consideraremos as seguintes suposições:

- uma única fonte gerando mensagens com um número de sequência (e, possivelmente, um *timestamp*, indicando o momento em que a mensagem foi gerada);
- um destino (*host móvel*) com um certo padrão de mobilidade.

Um exemplo de configuração de rede para ser utilizado em nossas simulações, poderia ser composto por:

- 1 *gateway* de domínio (GD), por onde passam todas as mensagens destinadas aos *hosts móveis* no domínio e dos *hosts móveis* à rede;
- n roteadores intermediários (R) (entre o GD e as estações base);
- m estações base (Mss).

Possivelmente, serão necessárias algumas adaptações no MobiCS a fim de permitir uma simulação mais realista dos esquemas de *seamless handover* gerados pelo *framework*. Por exemplo, para simular uma região de intersecção de células, seria necessário um artifício, criando uma estação base intermediária entre as duas células e fazendo com que esta receba as mesmas mensagens enviadas a cada uma das suas células vizinhas (que correspondem sua área de intersecção).

6.5 Questões

Há várias questões em relação ao *framework* proposto:

1. Quais outros módulos podem ser considerados? Quais os seus parâmetros?
2. Quais outros requisitos de QoS_h podem ser considerados?
3. Quais suposições podemos fazer? Em relação ao meio de transmissão?
4. A estratégia de gerenciamento hierárquico de mobilidade pode ser empregada como um módulo canônico?
5. A topologia da rede móvel influencia o desempenho do *handover*? Qual a topologia de rede a considerar? Deve ser um parâmetro de entrada?
6. Como fazer o roteamento de pacotes? Utilizar o Mobile IP?
7. A técnica de *buffer* requer o dimensionamento do tamanho do *buffer*. Qual é o tamanho apropriado?
8. O parâmetro perfil de mobilidade pode se tornar obsoleto? Como atualizar e recompor o esquema de *seamless handover*?
9. Qual a relevância prática deste *framework*? Como utilizar os resultados fornecidos?

7 Conclusão

Neste relatório, apresentamos os resultados de um estudo sobre *seamless handover* em redes móveis sem fio. O principal objetivo é identificar diferentes estratégias de *seamless handover* e compará-las, apresentando as suas vantagens e desvantagens em relação à viabilidade de cada um deles em oferecer uma migração transparente ao usuário móvel e às suas aplicações. Conforme observamos, todos os protocolos têm o objetivo comum de reduzir a latência e perda de pacotes (devido às restrições do Mobile IP no caso de micro-mobilidade), porém, nenhum deles se preocupa, particularmente, com os requisitos da aplicação.

Como resultado desse trabalho, apresentamos um *framework* para a geração de esquemas de *seamless handover* de acordo com os requisitos das aplicações. Diferentemente de todos os protocolos apresentados neste trabalho, este *framework* oferece uma forma de construir um esquema de *seamless handover* que é específico para um determinado tipo de aplicação.

Esse *framework* pode ser útil para (1) projeto de módulos canônicos e na definição de suas interações; (2) escolha sistematizada dos módulos para os requisitos de QoS_h de uma aplicação + perfil de mobilidade + características da rede e (3) para um estudo sistematizado da influência dos parâmetros dos módulos sobre a satisfação de QoS_h para diferentes tipos de serviços/aplicações.

Como planos futuros, pretendemos (1) explorar mais profundamente a arquitetura, estruturando os componentes e identificando as dependências entre eles, (2) instanciar o *framework* para alguns tipos de serviços e (3) implementar os módulos, simular no MobiCS e medir os resultados obtidos.

Referências

- [1] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, and Z. Turanyi. Design, implementation, and evaluation of cellular ip, 2000.
- [2] A. T. Campbell and J. Gomez. Ip micro-mobility protocols. In *ACM SIGMOBILE Mobile Computer and Communication Review (MC2R)*, volume 4, pages 45–54, October 2001.
- [3] Ramón Cáceres and Venkata N. Padmanabhan. Fast and Scalable Handoffs for Wireless Internetworks. In *Proceedings of ACM Mobicom'96*, pages 56–66, 1996.
- [4] R. C. A. da Rocha and M. Endler. Flexible Simulation of Distributed Protocols for Mobile Computing. In *Proc. 3rd Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, Boston, August 2000. (to appear).
- [5] R. C. A. da Rocha and M. Endler. Um Simulador de Protocolos Distribuídos para Computação Móvel. In *2nd Brazilian Workshop on Wireless Communication (2o. Workshop de Comunicação sem Fio - WCSF)*, pages 33–48, Belo Horizonte, 2000. DCC/ICE_x/UFMG.
- [6] Dynamic Host Configuration Protocol (DHCP). <http://www.dhcp.org/>.
- [7] Markus Endler and Vera Nagamuta. General Approaches for Implementing Seamless Handover. In *Workshop on Principles of Mobile Computing (POMC 2002)*, Toulouse, France, 2002.

- [8] 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.
- [9] Universal Mobile Telecommunications System (UMTS); Services & Services Capabilities. ETSI TS 122 105 (2002-06) v. 5.2.0, Release 5.
- [10] 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.
- [11] 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. (to appear).
- [12] Ahmed Helmy. A multicast-based protocol for IP mobility support. In *Networked Group Communication*, pages 49–58, 2000.
- [13] 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.
- [14] Jayanth Mysore and Vaduvur Bharghavan. A new multicasting-based architecture for internet host mobility. In *Mobile Computing and Networking*, pages 161–172, 1997.
- [15] C. Perkins. IP Mobility Support for IPv4. RFC 3220, January 2002.
- [16] C. E. Perkins and D. E. Johnson. Route Optimization in Mobile IP. Mobile IP Working Group, Internet Draft - work in progress, November 1997.
- [17] Charles E. Perkins. Mobile IP. *IEEE Communication Magazine*, May 1997.
- [18] Charles E. Perkins and David B. Johnson. Mobility support in ipv6. In *Mobile Computing and Networking*, pages 27–37, 1996.