THIAGO MEREGE PEREIRA

GERENCIAMENTO DE POLÍTICAS DE QUALIDADE DE SERVIÇO COM SUPORTE À MOBILIDADE

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de mestre em Informática.

Curitiba
2008

THIAGO MEREGE PEREIRA

GERENCIAMENTO DE POLÍTICAS DE QUALIDADE DE SERVIÇO COM SUPORTE À MOBILIDADE

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de mestre em Informática.

Área de Concentração: Redes de Computadores

Orientador: Prof. Dr. Edgard Jamhour

Curitiba
2008

PEREIRA, Thiago Merege.
GERENCIAMENTO DE POLÍTICAS DE QUALIDADE DE SERVIÇO COM SUPORTE À MOBILIDADE. Curitiba, 2008.

Dissertação - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Qualidade de Serviço 2. Mobilidade 3. *Diffserv* 4. PBNM
I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática.

Agradecimentos

Agradeço inicialmente ao meu orientador, Edgard Jamhour, pelas conversas e pela troca de idéias, principalmente no início do mestrado, e pela confiança depositada no final. Ao colega André Beller pela grande contribuição a este trabalho. Ao professor Mauro Fonseca, pelo auxílio nas implementações e publicações. Aos meus colegas e amigos que de alguma forma auxiliaram-me nesta jornada.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACK	*Acknowledgment*
AF	*Assured Forwarding*
AR	*Access Router*
ARP	*Address Resolution Protocol*
BA	*Behavior Aggregate*
BA	*Binding Acknowledgement*
BB	*Bandwidth Broker*
BU	*Binding Update*
CIM	*Common Information Model*
CLPM	*Configuration Level Policy Model*
CN	*Correspondent Node*
CoA	*Care-of Address*
COPS	*Common Open Policy Service*
COPS-PR	*Common Open Policy Service for Policy Provisioning*
CSMA/CA	*Carrier Sense Multiple Access with Collision Avoidance*
DCF	*Distributed Coordination Function*
DCoA	*Domain Care-of Address*
DHCP	*Dynamic Host Configuration Protocol*
Diffserv	*Differentiated Services*
DIFS	*Distributed Inter Frame Space*
DS	*Differentiated Services*
DSCP	*Differentiated Services Code Point*
ECN	*Explicit Congestion Notification*
EF	*Expedited Forwarding*
EGW	*Edge Gateway*
ER	*Edge Router*
FA	*Foreign Agent*
FEC	*Forwarding Equivalence Class*
GFA	*Gateway Foreign Agent*

GPL	*General Public License*
HA	*Home Agent*
HFA	*Hierarchical Foreign Agent*
HLPM	*High Level Policy Model*
HMIPv6	*Hierarchical Mobile Internet Protocol version* 6
HMRSVP	*Hierarchical Mobile Resource reSerVation Protocol*
HTTP	*HyperText Transfer Protocol*
IETF	*Internet Engineering Task Force*
Intserv	*Integrated Services*
IP	*Internet Protocol*
IPSec	*Internet Protocol Security*
LCoA	*Local Care-of Address*
LCoA	*On-link Care-of Address*
LDP	*Label Distribution Protocol*
LER	*Label Edge Routers*
LSP	*Label Switch Path*
LSR	*Label Switch Routers*
MAP	*Mobility Anchor Point*
MF	*Multi Field*
MMPLS	*Mobile Multi Protocol Label Switching*
MN	*Mobile Node*
MPLS	*Multi Protocol Label Switching*
MR	*Mobile Router*
MRSVP	Mobile *Resource reSerVation Protocol*
NEMO	*Network Mobility*
OID	*Object Identifier*
PBNM	*Policy Based Network Management*
PCIM	*Policy Core Information Model*
PCIMe	*Policy Core Information Model Extensions*
PDA	*Personal Digital Assistant*
PDP	*Policy Decision Point*
PEP	*Policy Enforcement Point*
PHB	*Per-Hop Behavior*
PIB	*Policy Information Base*

PRC	*Provisioning Class*
PRI	*Provisioning Instances*
QoS	*Quality of Service*
QPIM	*Policy Quality of Service Information Model*
RAN	*Radio Access Network*
RAS	*Radio Access Router*
RCoA	*Regional Care-of Address*
RPT	*Report State*
RSVP	*Resource reSerVation Protocol*
RSVP-MP	*Resource reSerVation Protocol Mobility Proxy*
SIFS	*Short Inter Frame Space*
SLA	*Service Level Agreement*
SLS	*Service Level Specification*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Procotol*
SOAP	*Simple Object Access Protocol*
SPPI	*Structure Policy Provisioning Information*
TCB	*Traffic Conditioning Block*
TCP	*Transmission Control Protocol*
TE	*Traffic Engineering*
TOS	*Type of Service*
TTL	*Time to Live*
UDP	*User Datagram Protocol*
VPN	*Virtual Private Network*
WSDL	*Services Description Language*
XML	*Extensible Markup Language*
XPointer	*Extensible Markup Pointer Language*

Resumo

Pretende-se com este trabalho prover um meio para garantir a qualidade do serviço oferecida ao usuário de maneira independente de sua localização. Para gerenciar a mobilidade do usuário, é utilizada a arquitetura de gerenciamento de mobilidade IPv4 Móvel, enquanto a escolha da metodologia utilizada para a garantia do QoS recai sobre a arquitetura *diffserv*. A arquitetura *diffserv* envolve a utilização de vários dispositivos que precisam ser configurados adequadamente, e uma das técnicas muito utilizadas para isso é o gerenciamento de rede baseado em políticas (PBNM). Para o uso das políticas, são definidos o PDP, entidade responsável pela tomada de decisões com o uso de políticas, e o PEP, que é responsável por executar as decisões providas pelo PDP. As políticas utilizadas pelo PDP são armazenadas em um repositório e são representadas através de uma PIB (*Policy Information Base*), a qual é formada por classes e instâncias dessas classes. Através de uma PIB, é possível modelar de maneira independente de dispositivo os mecanismos *diffserv* nele presentes, na forma do tratamento dos pacotes e dos parâmetros destes tratamentos. Há três abordagens possíveis para a provisão de uma PIB: uma completa encarnação é enviada ao PEP toda vez que a configuração do dispositivo é alterada; múltiplas encarnações são enviadas, mas somente uma é ativada por vez; somente as atualizações da PIB (classes e instâncias da PIB) são enviadas pelo PDP ao PEP. Em um ambiente móvel, a melhor solução consiste enviar somente as atualizações da PIB em resposta a um evento de mobilidade e, para isso, os elementos da PIB são classificados em informações estáticas e informações dinâmicas. A informação estática é provisionada no momento da inicialização do PEP, enquanto a configuração dinâmica deve ser atualizada conforme a mobilidade do usuário. Testes foram realizados para averiguar o impacto da introdução de qualidade de serviço aos usuários móveis com o uso da abordagem de atualização da PIB dos dispositivos. Os resultados demonstraram que a arquitetura mostrou-se viável quando comparada ao atraso já introduzido pelo processo de *handover* da arquitetura de mobilidade.

Palavras Chave: 1. Qualidade de Serviço 2. Mobilidade 3. *Diffserv* 4. PBNM

Abstract

This work provides a means to ensure the quality of service offered to users independently of their location. To manage the user mobility, it is used the Mobile IPv4 (MIP) architecture, and the QoS methodology used is Differentiated Services (diffserv**).** The diffserv architecture requires the use o multiple devices that need to be configured properly, and one of the techniques that is widely used for this is the Policy-Based Network Management (PBNM) framework. The framework defines the Policy Decision Point (PDP), a logical entity where policy decisions are made, and the Policy Enforcement Point (PEP), a local entity that enforces policy decisions. The PDP access a policy repository where the information is represented in terms of a PIB (Policy Information Base), which is composed by classes and instances of these classes. The diffserv PIB is device-independent and is structured based on the need to configure the sequential diffserv treatments being applied to a packet and the parameterization of these treatments. There are three possible approaches to the PIB provisioning: a complete incarnation is sent to the PEP whenever the configuration of the device changes; multiple incarnations are sent, but only one is active at a time; only the PIB updates (classes and instances) are sent by the PDP. In a mobile environment, the best solution consists in sending only the PIB updates in response to a mobility event. According to this approach, the PIB components are classified into static and dynamic information. The static information is provisioned at the initalization of the PEP**,** while the dynamic configuration is updated in response to a mobility event. Tests were made to evaluate the impact of the introduction of quality of service on a mobile environment with the use of the PIB update approach. The results showed that the architecture proved to be viable when compared to the latency already introduced by the MIP handover process.

Keywords: 1. Quality of Service 2. Mobility 3. Diffserv 4. PBNM

Capítulo 1

Introdução

O protocolo IP (*Internet Protocol*) é altamente difundido para a comunicação de dispositivos através de redes. A utilização do protocolo inclui não somente computadores, mas também celulares e *Personal Digital Assistants* (PDAs). Devido à crescente necessidade de acesso onipresente à Internet, a utilização de redes sem fio baseadas no protocolo IP é cada vez maior, tanto através de pontos de acesso públicos quanto de ponto de acessos privados (pagos).

Para que a área de abrangência de uma rede sem fio seja maximizada, são necessários vários pontos de acesso, cujas áreas de cobertura muitas vezes sobrepõem-se. A mobilidade dos usuários e a conseqüente troca entre esses pontos de acesso, conhecida como *handover*, faz com que, na forma como foi concebido o protocolo IP, haja a interrupção das sessões (conexões) ativas, pois não é inerente ao protocolo a mobilidade dos dispositivos. Uma vez realizada a conexão ao novo ponto de acesso, a conexão deve ser restabelecida.

Diversas propostas existem para contornar o problema da interrupção das conexões no momento do *handover*. Diferentes pesquisas realizadas por grupos de trabalho resultaram no protocolo IPv4 Móvel (PERKINS, 2002), para a versão 4 do protocolo IP, IPv6 Móvel (JONHSON, 2004), para a versão 6 do protocolo IP, e NEMO (*Network Mobility*) (DEVARAPALLI, 2005), para o suporte à mobilidade não de dispositivos, mas de redes. Outras pesquisas propuseram extensões a estes protocolos, com a adição do suporte à hierarquia de agentes móveis (agentes responsáveis por controlar a mobilidade do usuário), como, por exemplo, os protocolos IPv4 Móvel com Registro Regional (FOGELSTROEM, 2007) e IPv6 Móvel Hierárquico (SOLIMAN, 2005).

Além de garantir uma cobertura ampla dos pontos de acesso para a conexão à Internet, este acesso é acompanhado pela necessidade cada vez maior de largura de banda por parte dos usuários, tanto para a transmissão de voz e vídeo quanto para a troca de dados. As arquiteturas propostas para o gerenciamento de mobilidade herdam a característica do protocolo IP em não fornecer nativamente garantia à qualidade de serviço (*Quality of Service* – QoS). Tal garantia se faz cada vez mais presente com a mudança do perfil de tráfego dos usuários, o qual é sensível à perda de pacotes e/ou ao atraso, como é o caso da transmissão de voz e vídeo.

Deste modo, independente do tipo de acesso à rede ou da mobilidade do dispositivo, é necessário garantir a qualidade do serviço oferecido. Duas arquiteturas de qualidade de serviço foram propostas pelo *Internet Engineering Task Force* (IETF): serviços integrados (*Integrated Services – intserv*) (BRADEN, 1994) e serviços diferenciados (*Differentiated Services – diffserv*) (NICHOLS, 1998). Nos serviços integrados, os pacotes ingressos são classificados pelos nós da rede, e os recursos da rede são explicitamente identificados e reservados. Nos serviços diferenciados, ao invés de uma reserva explícita, o tráfego, por questões de escalabilidade, é diferenciado em um conjunto de classes, e os nós da rede fornecem diferentes prioridades de tratamento de acordo com essas classes. Além dessas duas arquiteturas, a arquitetura *Multi Protocol Label Switching* (MPLS) (ROSEN, 2001) quando, além de abordar roteamento, também engloba aspectos de QoS (opcional), pode ser considerada uma abordagem distinta do serviços integrados e dos serviços diferenciados para garantir a qualidade do serviço oferecido.

Um acordo de nível de serviço (*Service Level Agreement* - SLA) é um documento resultante da negociação entre o consumidor e o provedor de serviço, o qual pode especificar, dentre outras coisas, quais os níveis de disponibilidade e desempenho do serviço oferecido. Para atender aos requisitos de um SLA, no que se refere ao desempenho de rede, os provedores de serviço podem utilizar diferentes arquiteturas de qualidade de serviço (*diffserv*, por exemplo) para que o acordo estabelecido não seja violado, ou que seja violado somente dentro de limites pré-estabelecidos. Nesse acordo, é estabelecida a qualidade do serviço entregue pelo provedor, as garantias dos usuários e as punições pelo não cumprimento do acordado.

Para que o provedor de serviço esteja em conformidade com o SLA, os dispositivos de rede devem ser configurados de forma adequada para alcançar as metas definidas no contrato firmado com o consumidor. O modelo de gerenciamento de redes

baseado em políticas (*Policy Based Network Management* – PBNM) permite reduzir a complexidade do gerenciamento de redes de larga escala bem como também facilitar o gerenciamento de dispositivos heterogêneos. Através do PBNM, o administrador atribui a um recurso um ou mais papéis, e então especifica as políticas para cada um destes papéis, ao contrário de configurar de maneira separada cada recurso presente na rede.

As políticas podem ser representadas através de uma PIB (*Policy Information Base*), a qual é formada por classes e instâncias dessas classes. Através de uma PIB, é possível modelar de maneira independente de dispositivo os mecanismos *diffserv* nele presentes, na forma do tratamento dos pacotes e dos parâmetros destes tratamentos.

1.1 Motivação

O uso do protocolo IP permite que aplicações e serviços sejam projetados independentemente do ambiente em que operam, seja em uma rede sem fio ou em uma rede guiada. Contudo, a natureza das redes sem fio traz novos desafios para as metodologias e protocolos inicialmente pensados para as redes fixas.

As arquiteturas de qualidade de serviço propostas pelo IETF não levam em consideração a mobilidade dos dispositivos, sendo necessário buscar soluções para prover qualidade de serviço em ambientes móveis. Estas arquiteturas não foram pensadas inicialmente na natureza de mobilidade das redes sem fio, o que torna necessário adaptá-las a esse ambiente.

Os mecanismos originalmente propostos de provisão e sinalização devem ser adaptados à mobilidade do usuário, o qual pode mudar constantemente seu ponto de acesso à Internet. Tal trabalho prova-se não trivial e custoso (TAHA, 2005).

De forma separada, as metodologias para a garantia da qualidade de serviço e as propostas para o suporte à mobilidade dos usuários funcionam adequadamente, pois justamente foram pensadas de maneira independente. Para realizar a integração da mobilidade do usuário ao mesmo tempo em que lhe é garantida a qualidade de serviço, adaptações e alterações devem ser realizadas nos protocolos originalmente concebidos.

Os problemas de garantia de qualidade de serviço em redes sem fio nas quais há a mobilidade dos usuários estão relacionados em fornecer o serviço contratado em um SLA mesmo que haja mudança do ponto de acesso à rede. *Handovers* entre diferentes pontos de acesso e mudanças no endereço IP podem ocasionar violações do SLA (isto é, do serviço contratado). Essas violações podem ocorrer na demora durante o *handover*,

na perda de pacotes nesse ínterim e/ou na falta de garantia de qualidade de serviço no novo ponto de acesso ao qual o usuário se conectou.

Dessa forma, é necessário integrar adequadamente a mobilidade do usuário com a contínua garantia de qualidade de serviço a ele oferecida, para que o contrato estabelecido entre o provedor de serviço e o usuário móvel seja obedecido independente de sua localização.

Do ponto de vista do provedor de serviço que faz parte de um SLA, a garantia de QoS deve, além de manter o estabelecido no contrato, ocorrer de forma que os recursos sejam utilizados de maneira eficiente. A simples solução de superdimensionamento da rede nem sempre é possível e pode onerar em excesso o provedor do serviço em termos financeiros.

1.2 Objetivos

Pretende-se com este trabalho prover um meio para garantir a qualidade do serviço oferecida ao usuário de maneira independente de sua localização. Para gerenciar a mobilidade do usuário, é utilizada a arquitetura de gerenciamento de mobilidade IPv4 Móvel. Escolha da metodologia utilizada para a garantia do QoS recai sobre a arquitetura *diffserv*.

A associação de mobilidade com qualidade de serviço levanta a questão de gerenciamento dos mecanismos de qualidade de serviço dos dispositivos pelos quais os pacotes trafegam. Desta forma, a arquitetura IPv4 Móvel e a metodologia *diffserv* devem ser conciliadas com o gerenciamento dos dispositivos de rede, tarefa realizada com o uso da arquitetura PBNM.

Os eventos de mobilidade do usuário podem ser considerados eventos externos ao ambiente gerenciamento PBNM. O suporte a eventos externos à arquitetura PBNM já é previsto em Chan (2001), mas não é definido nenhum padrão para a comunicação desses eventos nem como esses eventos devem ser tratados. Desta forma, faz parte dos objetivos deste trabalho definir como serão tratados esses eventos externos à arquitetura PBNM, definidos nesta proposta como *eventos de reavaliação*. Em um ambiente de mobilidade, este evento externo à arquitetura PBNM toma o nome de *evento de mobilidade*.

Dentro da arquitetura de PBNM, as políticas devem ser modeladas para atender a um novo requisito, que é a mobilidade do usuário. Após a verificação da alteração de

localização de um nó móvel, notificada através de um *evento de mobilidade*, novas políticas devem ser aplicadas nos dispositivos de rede para reconfigurar os mecanismos *diffserv* neles presentes. Contudo, a aplicação dessas políticas deve ser realizada de modo a minimizar o tráfego de sinalização. Cabe neste ponto definir quais são as informações presentes nas políticas que são dependentes da mobilidade do usuário e necessitam ser aplicadas conforme sua localização e quais informações são independentes da mobilidade do usuário, as quais permitem a aplicação independente da localidade do usuário.

Definida a integração entre a metodologia *diffserv* e o protocolo IPv4 Móvel, a arquitetura proposta deve ser implementada e validada através de testes para verificar se um possível SLA acordado entre o provedor do serviço e o consumidor é respeitado. Além do tempo de resposta para a aplicação das configurações nos dispositivos, é necessário verificar a escalabilidade do sistema conforme cresce a quantidade de usuários móveis na rede e haja sua locomoção através dos diferentes pontos de acesso.

1.3 Estruturação

Este trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta as formas de gerenciamento de qualidade de serviço, quais sejam, a metodologia *intserv*, a metodologia *diffserv* e a abordagem utilizada pelo MPLS; o Capítulo 3 expõe os mecanismos de gerenciamento de mobilidade mais difundidos tanto para a solução dos problemas de macro-mobilidade quanto para os problemas de micro-mobilidade; o Capítulo 4 contém os conceitos e padrões concernentes ao gerenciamento de redes baseado em políticas utilizados ao longo do processo de decisão e aplicação das políticas nos dispositivos de rede; o Capítulo 5 aponta alguns trabalhos correlatos, nos quais há a integração de diferentes modos de gerenciamento de mobilidade com diferentes metodologias de qualidade de serviço; o Capítulo 6 introduz a arquitetura proposta, a qual integra vários dos conceitos apresentados nos capítulos anteriores; o Capítulo 7 apresenta a implementação da arquitetura proposta e as decisões de projeto tomadas; o Capítulo 8 expõe cenários de testes para a validação da arquitetura proposta, bem como os resultados numéricos auferidos; finalmente, o Capítulo 9 apresenta as conclusões do presente trabalho e possibilidades futuras de continuidade.

Capítulo 2

Metodologias de Qualidade de Serviço

Determinadas aplicações necessitam de requisitos de qualidade de serviço mais rigorosos do que o modelo de melhor esforço (*best-effort*) é capaz de oferecer. Neste modelo, a largura de banda é alocada aos usuários da melhor maneira possível, visando atender a todos, sem o comprometimento com taxa de transmissão ou outra qualidade de serviço (CLARK, 1998).

Aplicações de tempo real, por exemplo, não funcionam adequadamente no modelo originariamente concebido para a transmissão de dados utilizando a pilha de protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*), em virtude dos atrasos variáveis nas filas dos dispositivos responsáveis para a transmissão de pacotes e devido às perdas de pacotes decorrentes do congestionamento.

Através da diferenciação de tráfego, é possível estabelecer diferentes níveis de serviço para diferentes aplicações e usuários. Em busca desta solução, vários grupos de trabalho do IETF direcionaram suas pesquisas para os temas relacionados à provisão de qualidade de serviço em redes IP. Como resultado das pesquisas dos grupos do IETF, surgiram duas metodologias: os serviços integrados (*IntServ*) (BRADEN, 1994) e os serviços diferenciados (*DiffServ*) (NICHOLS, 1998). Paralelamente à pesquisa de QoS, o trabalho em torno de métodos de encaminhamento melhores com o uso de rótulo no cabeçalho dos pacotes culminou no surgimento do *Multi Protocol Label Switching* (MPLS), cujos cabeçalhos e rótulos podem ser utilizados também para a priorização do tráfego.

Na metodologia de serviços integrados, os nós de rede classificam os pacotes de entrada e os recursos de rede são explicitamente identificados e reservados. A reserva de recursos é realizada individualmente por fluxo de transmissão, e o protocolo RSVP

(*Resource reSerVation Protocol*) (BRADEN, 1997) é o protocolo comumente responsável por sinalizar os requisitos de qualidade de serviço de um determinado fluxo.

Na metodologia de serviços diferenciados, ao invés de uma reserva explícita de recursos, o tráfego é diferenciado em classes de serviço e os nós de rede priorizam o tratamento dos pacotes de acordo com essas classes. A reserva de recursos é realizada por meio de classes de tráfego agregado ao invés de ser realizada por fluxos individuais, como ocorre com a metodologia de serviços integrados.

O MPLS consiste no encaminhamento dos pacotes com a utilização de um rótulo atribuído a cada pacote, e não através da análise do endereço IP presente no cabeçalho do pacote. Com o uso da informação presente no rótulo, obtém-se o próximo roteador ao qual o pacote deve ser encaminhado. Uma vez que os rótulos possuem tamanho reduzido e fixo, o MPLS pode ser mais eficiente quando comparado ao roteamento baseado no endereço IP. Além disso, o MPLS facilita o provisionamento de QoS.

2.1 Serviços Integrados

O termo serviços integrados é utilizado para um modelo de serviço na Internet que inclui o serviço *best-effort*, o serviço de tempo real (serviço garantido) (SHENKER, 1997a) e o compartilhamento controlado do enlace (serviço de carga controlada) (WROCLAWSKI, 1997a). Os serviços garantidos fornecem aos fluxos dos usuários uma quantidade garantida de largura de banda, limites rígidos de atraso fim-a-fim e a ausência de perda de pacotes nas filas dos dispositivos. O serviço de compartilhamento controlado do enlace assegura aos usuários um serviço tão próximo possível ao recebido pelo serviço *best-effort* em uma rede com pouco tráfego. Por fim, o serviço *best-effort* é caracterizado pela falta de garantia de qualidade de serviço e pela entrega dos pacotes de um fluxo da melhor maneira possível.

Para que ocorra a reserva de recursos aos fluxos de dados e haja a garantia de qualidade de serviço, é necessário que os roteadores tenham o conhecimento do estado desses fluxos. Além disso, é necessário um mecanismo explícito de configuração dos dispositivos para prover o serviço requerido.

Em Braden (1994), é proposta uma arquitetura de referência para implementar o modelo de serviços integrados, a qual possui quatro componentes: o escalonador de

pacotes, a rotina de controle de admissão, o classificador e o protocolo de configuração de reserva (*resource reservation protocol*).

Para que ocorra a diferenciação do tráfego e para que seja possível garantir a qualidade de serviço para determinados fluxos, o roteador deve implementar a qualidade de serviço apropriada a cada fluxo, de acordo com o modelo de serviço (*best-effort*, serviço garantido ou serviço de carga controlada). A função do roteador que determina as diferentes qualidades de serviço é chamada *controle de tráfego*, o qual é implementado por três componentes: o escalonador de pacotes, o classificador e o controle de admissão.

O escalonador de pacotes gerencia o encaminhamento de diferentes fluxos de pacotes com o uso de um conjunto de filas e outros mecanismos, como temporizadores. O escalonador de pacotes atua onde os pacotes são enfileirados.

O classificador é responsável por mapear os pacotes ingressos para alguma classe. Todos os pacotes pertencentes a uma determinada classe recebem o mesmo tratamento pelo escalonador de pacotes. O mapeamento pode ser realizado de acordo com o cabeçalho do pacote e com algum número de classificação adicionado a cada pacote.

O controle de admissão é responsável por determinar se a aceitação de um novo fluxo e a garantia de qualidade de serviço para esse fluxo pode ou não impactar nas garantias de qualidade de serviço anteriormente definidas.

O protocolo de configuração de reserva é responsável por criar e manter o estado dos fluxos nos dispositivos finais e nos roteadores que estão no caminho percorrido pelo fluxo. Para criar e manter o estado dos fluxos nos dispositivos, é definido em Braden (1997) o protocolo *Resource Reservation Protocol* (RSVP). O protocolo RSVP pode ser utilizado por *hosts* ou por roteadores. No primeiro caso, ele é utilizado para requisitar um determinado QoS da rede para um fluxo ou uma aplicação em particular. No segundo caso, os roteadores utilizam o protocolo para transmitir os pedidos de QoS a todos os nós ao longo do caminho dos fluxos e para estabelecer e manter o estado do serviço requisitado. De maneira geral, as requisições do protocolo RSVP resultam na reserva de recursos nos dispositivos ao longo do caminho de dados.

A Figura 2.1 apresenta o modelo de funcionamento da metodologia *intserv* com o protocolo RSPV quando utilizado por *hosts* e roteadores.

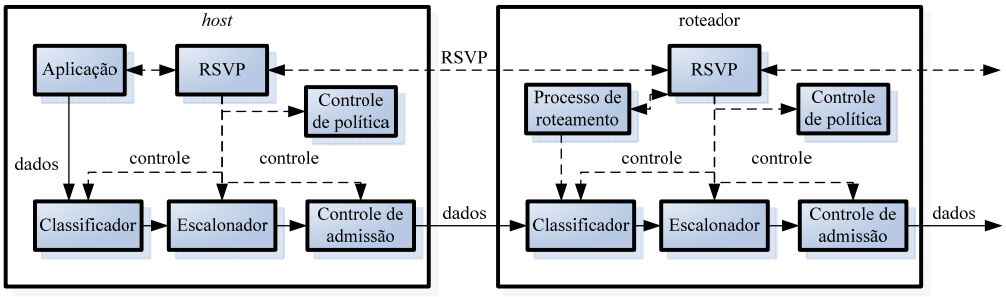

Figura 2.1: Modelo da metodologia *intserv*

Durante a configuração da reserva de recursos, um pedido de QoS através do protocolo RSVP é verificado pelo controle de política e pelo controle de admissão, para determinar se o usuário possui permissão administrativa para realizar a reserva e se há recursos suficientes para prover a requisição de QoS, respectivamente. Se ambas as condições são satisfeitas, os parâmetros do classificador de pacotes são configurados para obter o QoS desejado. Se alguma das condições não é satisfeita, é notificado um erro pelo protocolo RSVP à aplicação que realizou a solicitação.

A lista de parâmetros de qualidade de serviço requeridos por uma aplicação é chamada de *flowspec* e é definida em Wroclawski (1997b), a qual é transportada pelo protocolo de RSVP. Esta lista é submetida ao controle de admissão e, se aceita, é utilizada para a configuração do mecanismo escalonador de pacotes.

O *flowspec* é composto por dois elementos, quais sejam: a especificação de reserva (RSpec), que indica a classe de serviço desejada; e a especificação de tráfego (TSpec), que indica as características do que tráfego transmitido. A definição dos parâmetros *flowspec* geralmente é utiliza o modelo *Token Bucket*.

A partir do modelo *Token Bucket*, a especificação do tráfego (TSpec) (SHENKER, 1997b) é composta pelos seguintes parâmetros:

- *Token rate* (r): taxa média em bytes/s;
- *Bucket depth* (b): tamanho da rajada (*burst*) sustentável, em bytes
- *Peak rate* (p): taxa de pico;
- *Minimum policed size* (m): pacotes menores que *m* são considerados com o tamanho de *m* bytes;
- *Maximum packet size* (M): tamanho máximo de pacote (em bytes) que pode ser enviado pelo fluxo.

A especificação de reserva (RSpec) também utiliza o modelo *Token Bucket* e é composta pelos seguintes parâmetros:

- *Rate* (R): taxa média solicitada;
- *Slack term* (S): saldo de retardo, que consiste no valor de atraso que pode ser utilizado pelos nós intermediários. Corresponde à diferença entre o atraso garantido se a banda R for reservada e o atraso realmente necessário, especificado pela aplicação.

O procedimento de reserva de recursos através do protocolo RSVP é sempre unidirecional. Assim, para uma mesma aplicação transmitir e receber dados com tratamento de QoS são necessárias duas reservas, uma em direção ao transmissor e uma em direção ao receptor. A reserva de recursos nos dispositivos que fazem parte da transmissão de um fluxo se dá através da troca de mensagens PATH e RESV.

2.2 Serviços Diferenciados

Os serviços diferenciados (*diffserv*) tencionam a discriminação dos serviços no protocolo IP sem a necessidade de identificar o estado de cada fluxo nem sinalizar cada nó no caminho dos dados. Um serviço define certas características na transmissão de um pacote em uma direção através de um conjunto de um ou mais caminhos dentro da rede. Essas características podem ser especificadas em termos quantitativos ou estatísticos de *throughput*, atraso, *jitter* e/ou perda de pacotes, ou podem ser especificadas em termos de prioridade de acesso aos recursos da rede. Justamente, a diferenciação de serviços visa acomodar requisitos de determinadas aplicações e expectativas dos usuários heterogêneos (NICHOLS, 1998).

A arquitetura de serviços diferenciados é baseada em um modelo no qual o tráfego que ingressa na rede é classificado, possivelmente condicionado e atribuído a um determinado comportamento agregado (*behavior aggregate*) nas bordas da rede. Cada comportamento agregado é identificado por um *codepoint* DS (*differentiated services*). No interior da rede, os pacotes são encaminhados de acordo com o PHB (*per-hop behavior*) associado com o *codepoint* DS.

Uma rede *diffserv* seleciona os pacotes através do valor no campo DS, juntamente com gerenciamento de fila e mecanismos de escalonamento de pacotes

capazes de entregar o tratamento definido pelo valor presente no campo DS. A definição do valor do campo DS e o condicionamento dos pacotes marcados precisa somente ser realizada nas bordas da rede.

A arquitetura *diffserv* possui dois componentes principais (NICHOLS, 1998): o primeiro é o comportamento no encaminhamento dos pacotes e o segundo são as políticas e os componentes para configurar os parâmetros usados no encaminhamento. O comportamento no encaminhamento, conhecido como *per-hop behavior* (PHB), inclui o tratamento individual que um pacote recebe, implementado por disciplinas e gerenciamento de filas. Os serviços diferenciados são obtidos através do mapeamento do código presente no campo DS para um PHB em cada nó ao longo do caminho dos dados.

O campo DS visa substituir o octeto *Type of Service* (TOS) do protocolo IPv4 e o octeto *Traffic Class* do protocolo IPv6. Seis *bits* do campo DS são usados como *codepoint* (DSCP – *Differentiated Services Code Point*) para selecionar o PHB atribuído a um pacote em cada nó. Os dois *bits* restantes do octeto (ECN – *Explicit Congestion Notification*) são utilizados para o controle de congestionamento (RAMAKRISHNAN, 2001). A estrutura do campo DS é definida na Figura 2.2.

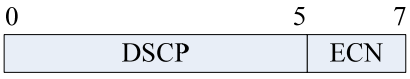

Figura 2.2: Campo DS

A tradução do campo DS nas bordas de um domínio *diffserv* é objeto de um acordo de nível de serviço (SLA) entre usuários e provedores de serviço.

Um PHB padrão deve ser definido em um nó *diffserv*. O comportamento *default*, definido em Baker (1995), é o encaminhamento por melhor esforço (*best-effort*). Quando não há um acordo pré-definido, assume-se que os pacotes pertençam ao PHB padrão. O *codepoint* recomendado para o PHB *default* são os *bits "000000"*.

A marcação é realizada pelos condicionadores de tráfego (*traffic conditioners*) nos limites da rede, incluindo as bordas da rede (primeiro roteador ou dispositivo de origem), e nos limites administrativos. Os condicionadores de tráfego podem incluir primitivas de marcação, medição, policiamento e *shaping* (suavização/descarte). Os serviços são realizados com uma classificação individual dos pacotes e mecanismos de

condicionamento de tráfego nas bordas juntamente com a concatenação de PHBs ao longo do caminho do tráfego.

O condicionador de tráfego pode ser formado pelos elementos presentes na Figura 2.3 (BLAKE, 1998).

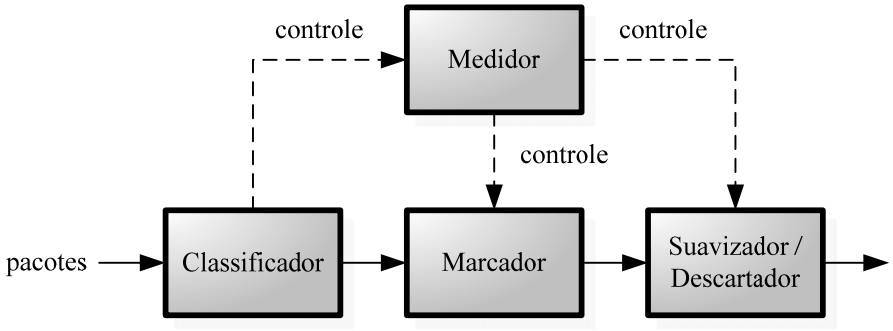

Figura 2.3: Condicionador de Tráfego

O classificador seleciona os pacotes através dos cabeçalhos e encaminha aqueles que correspondem às regras de classificação para processamento posterior. O modelo *diffserv* especifica dois tipos de classificadores:

- Multicampos (*Multi Field* – MF): além do byte DS pode utilizar outras informações do cabeçalho IP, como endereço, porta etc.;
- Comportamento Agregado (*Behavior Aggregate* – BA): a classificação é baseada somente no byte DS.

O medidor verifica se o fluxo de tráfego selecionado pelo classificador está de acordo com o perfil do tráfego especificado no contrato (SLA) estabelecido entre o cliente e o provedor de serviços. O medidor passa informações de estado para as outras funções de condicionamento, para que se possa tomar uma ação particular para cada pacote que atende ou não os requisitos QoS.

O marcador é responsável pela marcação ou remarcação do byte DS. A marcação é realizada nos pacotes emitidos sem marcação pelo cliente. A remarcação é executada quando o nó seguinte no encaminhamento possuir uma interpretação diferente para o byte DS.

O suavizador/descartador é responsável pela adaptação e descarte dos pacotes de acordo com as propriedades estabelecidas pelo PHB. Um suavizador geralmente tem um buffer de tamanho definido e os pacotes podem ser descartados se não houver espaço suficiente no buffer para armazenar os pacotes atrasados. Os descartadores realizam o descarte de alguns ou de todos os pacotes baseados em regras específicas. Por exemplo, podem-se aceitar todos os pacotes até a taxa máxima permitida e descartar todos eles que excedem a taxa configurada.

O condicionador de tráfego em um dispositivo de borda garante que os pacotes que vão passar pelo domínio sejam marcados de acordo com um PHB de um dos grupos de PHBs suportados pelo domínio. Isso é necessário porque domínios *diffserv* diferentes podem ter grupos diferentes de PHBs, o que significa que a mesma entrada no byte DS pode ser interpretada de formas distintas em domínios diferentes.

Se um pacote passar através de vários domínios, o byte DS pode ser remarcado em cada dispositivo de borda, para garantir que a qualidade de serviço contratada no SLA seja cumprida.

2.2.1 Domínio *Diffserv*

Um domínio *diffserv* é um conjunto contíguo de nós *diffserv* que opera sob as mesmas políticas de provisão (*provisioning*) e implementa os mesmos PHBs em cada nó. Um domínio *diffserv* tem limites bem definidos, consistindo em *nós de borda* e *nós de núcleo*.

Os *nós de borda* conectam o domínio *diffserv* a outros domínios *diffserv* ou não-*diffserv*. Os *nós de borda* classificam e podem condicionar o tráfego que ingressa no domínio, garantindo que os pacotes pertencentes a esse tráfego estejam devidamente marcados com um PHB suportado pelo domínio.

Os *nós de núcleo*, que estão dentro do domínio *diffserv*, selecionam o comportamento no encaminhamento dos pacotes através do *codepoint* neles presentes, mapeando esse valor para um dos PHBs suportados. Os *nós de núcleo* podem realizar condicionamento de tráfego limitado, como, por exemplo, a remarcação dos *codepoints* DS, ou podem realizar também classificação e condicionamento de tráfego de forma mais complexa, no que se assemelham aos *nós de borda*.

Um domínio *diffserv* regularmente consiste em uma ou mais redes sobre a mesma administração. A administração do domínio *diffserv* é responsável pela garantia

de que os recursos adequados serão disponibilizados e reservados para suportar e cumprir com os SLAs oferecidos pelo domínio aos seus clientes.

A Figura 2.4 apresenta um exemplo de um domínio *diffserv* com suas entidades.

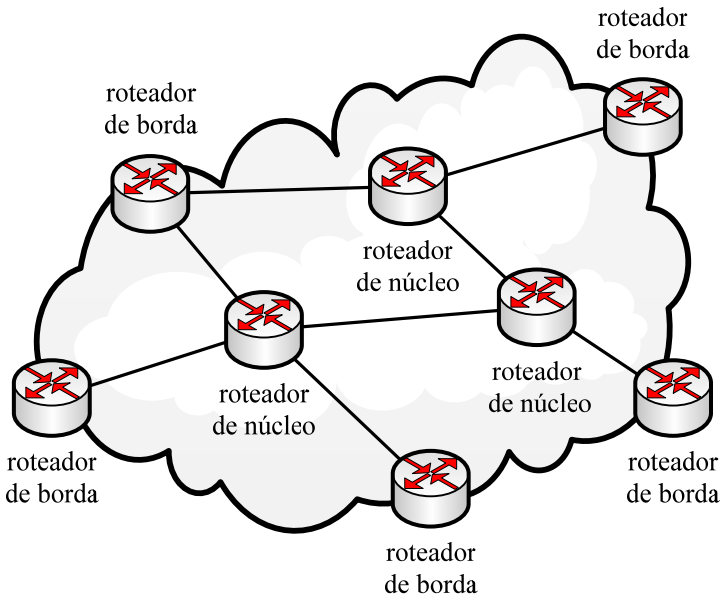

Figura 2.4: Domínio *diffserv*

2.2.2 Assured Forwarding Per-Hop Behavior

Em Heinanen (1999) é definido um grupo PHB chamado *Assured Forwarding* (AF), no qual são definidas quatro classes independentes para a transmissão de pacotes IP. Dentro de cada classe, há três níveis de prioridade de descarte que podem ser atribuídos a um pacote. Os recursos para a transmissão dos pacotes, dentro de cada classe AF, são expressos através da largura de banda e através do tamanho do *buffer* da fila.

Dentro de cada classe AF, os pacotes IP são marcados com um dos três níveis possíveis de prioridade de descarte. No caso de congestionamento, o nível de prioridade de descarte de um pacote determina sua importância dentro de uma classe AF em relação aos outros pacotes pertencentes à mesma classe; pacotes com prioridade de descarte menor são mantidos enquanto pacotes com prioridade de descarte maior são descartados.

Três aspectos são levados em consideração para a transmissão de um pacote IP (HEINANEN, 1999): *i)* a quantidade de recursos de transmissão alocada para a classe AF à qual o pacote pertence; *ii)* qual é a situação atual da classe AF, no que se refere à

carga; e, caso haja congestionamento na classe, *iii)* qual é a prioridade de descarte do pacote.

Para implementar cada classe AF, um nó *diffserv* deve alocar uma quantidade configurável mínima de recursos (largura de banda e tamanho do *buffer* da fila). Cada classe deve possuir recursos de maneira a alcançar a taxa de serviço configurada (largura de banda) tanto em períodos curtos de tempo quanto em períodos mais longos. Não há requisitos quantificáveis de tempo (*delay* ou *jitter*) associados à transmissão de pacotes de uma classe AF.

Um domínio *diffserv* pode, nas suas bordas, controlar a quantidade de tráfego AF que ingressa ou egressa do domínio através de vários níveis de prioridade de descarte. O condicionamento do tráfego pode incluir suavização (*traffic shaping*), descarte de pacotes, aumento ou redução da prioridade de descarte dos pacotes e atribuição de pacotes de uma classe AF a outra. Entretanto, as ações para o condicionamento do tráfego não devem reordenar os pacotes que pertençam a um mesmo microfluxo. De acordo com Nichols (1998), microfluxo é uma única instância de um fluxo de pacotes com origem em uma aplicação e destino a outra aplicação. O fluxo é identificado pelo endereço de origem, endereço de destino, identificação do protocolo, porta de origem e porta de destino.

Em Heinanen (1999) são definidos os *codepoints* recomendados para as quatro classes AF, cada uma possuindo três níveis de prioridade de descarte. Os *codepoints* são apresentados na Tabela 2.1.

Tabela 2.1: *Codepoints* para as classes AF

	Classe 1	**Classe 2**	**Classe 3**	**Classe 4**
Prioridade baixa de descarte	001010 (AF11)	010010 (AF21)	011010 (AF31)	100010 (AF41)
Prioridade média de descarte	001100 (AF12)	010100 (AF22)	011100 (AF32)	100100 (AF42)
Prioridade alta de descarte	001110 (AF13)	010110 (AF23)	011110 (AF33)	100110 (AF43)

2.2.3 Expedited Forwarding Per-Hop Behavior

Em Davie (2002), juntamente com Charny (2002), é definido um PHB chamado *Expedited Forwarding* (EF), com o objetivo de fornecer serviços com baixo atraso, baixo *jitter* e baixa perda de pacotes.

As principais causas de atraso de um pacote na rede são o atraso de propagação no meio utilizado (provenientes do atraso da velocidade da luz) e os atrasos das filas em *switches* e roteadores. Uma vez que o atraso de propagação é definido de acordo com a topologia da rede, o atraso e o *jitter* são minimizados quando os atrasos das filas em *switches* e roteadores são minimizados.

O objetivo do PHB EF é fornecer um PHB no qual os pacotes devidamente marcados como tal encontrem filas vazias ou pequenas nos nós da rede. Uma vez que as filas são pequenas comparadas ao tamanho do *buffer* disponível a elas, a perda de pacotes também tende a um mínimo possível.

Para garantir que os pacotes marcados com o PHB EF encontrem filas pequenas, é necessário garantir que a taxa de egresso seja maior que a taxa de ingresso de uma interface durante intervalos longos e curtos de tempo, independente da quantidade de outro tipo de tráfego que não seja EF.

Outros PHBs podem estar presentes juntamente com o PHB EF, mas o tráfego desses outros PHBs não deve ser levado em consideração pelo tráfego EF. O tráfego que não possui a marcação EF sofre preempção, através de filas com diferentes níveis de prioridade, para priorizar o tráfego EF. Contudo, deve haver um limitador para que o tráfego não-EF seja prejudicado com uma preempção por tempo indefinido. Uma vez definido o limite de tráfego EF, o que exceder esse limite é descartado.

O *codepoint* recomendado para este PHB é *"101110"*.

2.3 Multi Protocol Label Switching (MPLS)

A arquitetura *Multi Protocol Label Switching* (MPLS) presente em Rosen (2001) propõe que o encaminhamento dos pacotes na rede seja realizado através de rótulos (*labels*) atribuídos na entrada de um domínio MPLS, o qual é definido como um conjunto contíguo de nós, os quais operam o roteamento e o encaminhamento dos pacotes com o uso do MPLS, e também estão em um mesmo domínio de roteamento ou domínio administrativo.

No roteamento IP convencional, cada pacote é analisado em cada roteador para verificar seu endereço de destino e então decidir por qual enlace ele será encaminhado. No MPLS, uma vez atribuída uma classe de equivalência de encaminhamento (*Forwarding Equivalence Class* – FEC) a um pacote, os roteadores não mais precisam

analisar o cabeçalho da camada de rede do pacote (IP, na maioria dos casos) para verificar qual é o caminho a ser utilizado.

A atribuição de uma FEC é realizada somente uma vez na entrada do domínio MPLS. Uma vez atribuída, ela é codificada no rótulo (*label*) presente no cabeçalho MPLS (Figura 2.5). Quando o pacote é encaminhado ao próximo roteador, o rótulo também é enviado. Dessa forma, nos roteadores subseqüentes, somente o rótulo é utilizado como índice para verificar qual é o próximo roteador para o qual será encaminhado o pacote. Neste índice, há também um novo rótulo a ser atribuído ao pacote antes de ele ser encaminhado ao próximo roteador, uma vez que o escopo dos rótulos é somente ponto-a-ponto, e não fim-a-fim.

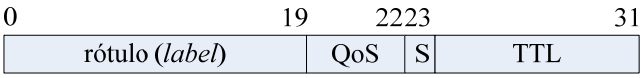

Figura 2.5: Cabeçalho MPLS

Além do índice presente no campo *label*, o cabeçalho MPLS contém o campo QoS, o qual indica a classe de serviço, o campo S, que indica o empilhamento de rótulos, e o campo TTL (*Time to Live*), o qual é utilizado para prevenir *loopings* e para limitar o escopo de um pacote (quantidade de roteadores pelos quais o pacote passa).

As FECs resumem informações essenciais sobre os pacotes tais como o destino, a precedência, se é parte de um grupo VPN, informações relacionadas ao QoS, e a rota do pacote a ser seguida, estabelecida pela engenharia de tráfego (*Traffic Engineering* – TE). Pacotes pertencentes à mesma FEC recebem tratamento similar em todos os roteadores da rede.

Os roteadores de entrada e saída de uma rede MPLS são chamados de *label edge routers* (LER), os quais são responsáveis, respectivamente, por atribuir um rótulo ao pacote ingresso (*push*) e retirar o rótulo no pacote egresso da rede (*pop*). Os roteadores que realizam o roteamento somente com o uso do rótulo MPLS são chamados de *label switch routers* (LSRs). O caminho percorrido pelos pacotes através de um ou mais LSRs é chamado de *Label Switch Path* (LSP).

Para a distribuição de rótulos e o estabelecimento de LSPs entre os roteadores LERs e LSRs é utilizado o protocolo *Label Distribution Protocol* (LDP) (ANDERSSON, 2001). Os LSRs em uma rede MPLS trocam informações acerca dos rótulos e rotas para levantar a topologia da rede pela qual eles encaminham os pacotes.

Em um domínio MPLS, a qualidade de serviço pode ser obtida através da análise do campo QoS do cabeçalho MPLS para priorizar o tráfego em cada nó ou, através de TE, selecionar determinados caminhos no domínio para determinar rotas e reservar recursos para determinados fluxos de dados.

Além disso, com a utilização de rótulos, o MPLS pode ser visto como uma tecnologia de tunelamento que supera as técnicas normais de tunelamento IP-IP (utilizado, por exemplo, no protocolo IPv4 Móvel).

Capítulo 3

Gerenciamento de Mobilidade

A atribuição de um endereço IP a um dispositivo é dependente da localidade onde este dispositivo se encontra. Os protocolos para gerenciamento de mobilidade buscam justamente superar essa dependência de localidade através de mecanismos para a tradução de endereços e mecanismos para a distribuição de pacotes de qualquer e para qualquer dispositivo, seja ele fixo ou móvel.

De acordo com o escopo de atuação dos protocolos de gerenciamento de mobilidade, eles podem ser classificados em duas categorias: protocolos de global ou macro-mobilidade e protocolos de regional ou micro-mobilidade.

A macro-mobilidade consiste na mobilidade de um nó através de diferentes domínios IP mantendo as sessões do protocolo de transporte. A macro-mobilidade é transparente para as aplicações e para os protocolos de transporte, os quais funcionam de igual maneira para nós fixos ou móveis. Propostas para solucionar o problema da macro-mobilidade são os protocolos IPv4 Móvel (PERKINS, 2002), IPv6 Móvel (JOHNSON, 2004) e NEMO (DEVARAPALLI, 2005).

A micro-mobilidade consiste na movimentação dentro de um domínio. Os protocolos de micro-mobilidade visam diminuir a quantidade de tráfego de sinalização e diminuir o tempo de difusão da nova localização de um nó (MANNER, 2002). As propostas para protocolos de micro-mobilidade utilizam hierarquias de *Foreign Agents* (FAs) e/ou *Home Agents* (HAs), nas quais um pedido de registro ou atualização da localização não precisa ser enviado até o HA, mas permanece regionalizado, como nos trabalhos em Fogelstroem (2007), McCann (1999) e Malki (2007) para o protocolo IPv4 e nos trabalhos em Soliman (2005) e Malinen (2001) para o protocolo IPv6. Outras propostas de micro-mobilidade utilizam esquemas para melhoramento localizado do

roteamento, como em Ramjee (1999) e Shelby (2000), nas quais um protocolo específico configura rotas dentro do domínio para a transmissão de pacotes aos nós móveis. A micro-mobilidade evita a tradução de endereços, o tunelamento de tráfego e possibilita *handovers* mais rápidos.

Enquanto os protocolos de macro-mobilidade podem ser aplicados quando a movimentação de um nó envolve mais de um domínio (mobilidade intra-domínio), os protocolos de micro-mobilidade visam a mobilidade inter-domínio, isto é, a mobilidade que envolve somente um domínio de gerenciamento (Figura 3.1).

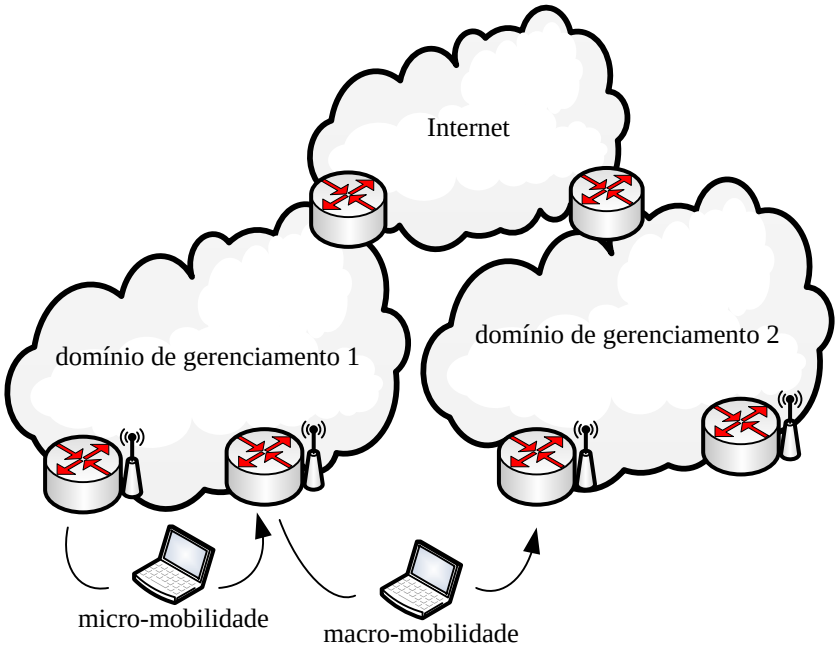

Figura 3.1: Micro- e macro-mobilidade

3.1 IPv4 Móvel

A versão 4 do protocolo IP assume que um endereço IP de um nó identifica de forma única seu ponto de acesso à Internet. Para receber um pacote destinado a seu endereço IP, o nó deve estar localizado na rede à qual pertence seu endereço, o que, de outra forma, acarretaria na não entrega de pacotes a esse nó. Para que o nó possa alterar seu ponto de conexão com a Internet sem que isso ocasione a perda das comunicações ativas do nó, é proposto um mecanismo em Perkins (2002) chamado IPv4 Móvel, o qual permite a alteração do ponto de conexão de um nó a Internet sem que seja necessária a alteração de seu endereço IP, isto é, o nó é sempre identificado pelo seu endereço IP de origem (*home address*).

Para os nós que estão se comunicando com o assim chamado nó móvel, essa mobilidade deve ser transparente, bem como para os dispositivos intermediários dessa comunicação e que não fazem parte da arquitetura proposta em Perkins (2002), como roteadores ou outros dispositivos intermediários.

O protocolo IPv4 Móvel permite a movimentação de um nó tanto através de redes homogêneas quanto de redes heterogêneas. Isto significa que o protocolo permite a movimentação de um nó entre segmentos com o padrão Ethernet ou a movimentação de um nó entre um segmento Ethernet e uma rede celular, por exemplo, desde que na rede celular seja também possível identificar o nó através de seu endereço IP.

O protocolo, como concebido em Perkins (2002), é voltado ao gerenciamento de macro-mobilidade. Um dos pressupostos do protocolo consiste que um nó móvel não irá alterar seu ponto de conexão à Internet mais do que uma vez por segundo. Para o gerenciamento de micro-mobilidade, o protocolo IPv4 Móvel não é muito adequado. Por exemplo, o *handover* realizado entre transmissores que cobrem uma pequena área geográfica e que pertencem à mesma sub-rede IP apresenta melhores resultados (convergência mais rápida e menor *overhead*) se for realizado com mecanismos de mobilidade na camada de enlace.

A Figura 3.2 apresenta um cenário onde estão presentes as entidades (agentes móveis) que são introduzidas com o protocolo IPv4 Móvel.

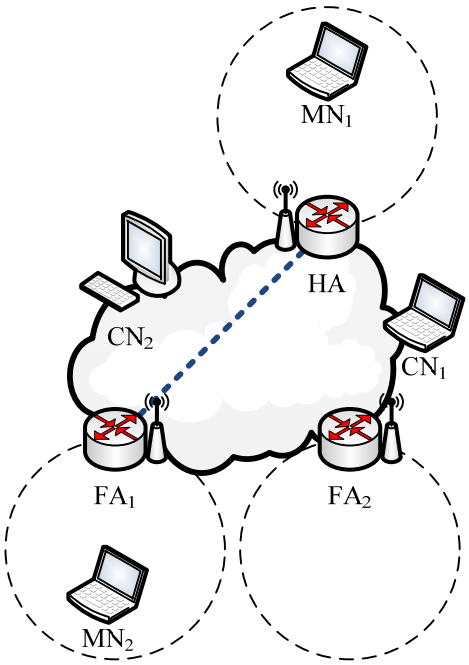

Figura 3.2: Entidades do protocolo IPv4 Móvel

O nó móvel (*Mobile Node* – MN) é o dispositivo que muda seu ponto de conexão entre uma rede e outra sem mudar seu endereço IP de origem. Uma vez possuindo conectividade em um ponto de conexão, ele pode continuar comunicando-se com outros nós na Internet independente de sua localização.

O *Home Agent* (HA) é um roteador na rede de origem do nó móvel responsável por realizar o tunelamento dos pacotes destinados ao nó móvel quando este está localizado fora de sua rede de origem, isto é, localizado em uma rede estrangeira. O HA é responsável também por manter informações sobre a localização do nó móvel e por criar um túnel entre ele e o FA (ou ele e o nó móvel, conforme o esquema de endereçamento utilizado).

O *Foreign Agent* (FA) é um roteador presente na rede estrangeira responsável por prover serviços de roteamento para o nó móvel enquanto este estiver registrado fora de sua rede de origem. O FA pode ser responsável por retirar os pacotes do túnel e entregá-los ao nó móvel. O FA pode também servir de roteador *default* para o nó móvel que está conectado a ele e deseje transmitir algum pacote.

O nó correspondente (*Correspondent Node* – CN) é o dispositivo com o qual um nó móvel se comunica. O CN pode ser tanto fixo quanto móvel. Para o CN, a mobilidade do usuário é transparente, pois ele sempre referencia o nó móvel através de seu endereço de origem, independente da localização do nó.

Na sua rede de origem, o nó móvel possui um endereço IP de longa duração conhecido como endereço de origem (*home address*), atribuído da mesma forma que o seria para um nó não-móvel. Quando o nó móvel está em uma rede estrangeira, é atribuído a ele um *Care-of Address* (CoA), que reflete o atual ponto de conexão do nó.

O protocolo IPv4 Móvel prevê dois tipos de CoA, que refletem no ponto final do túnel criado a partir do HA. O *foreign agent* CoA é o endereço do FA no qual o nó móvel está conectado e o *co-located* CoA é um endereço local obtido externamente (através do protocolo *Dynamic Host Configuration Protocol* – DHCP, por exemplo) e atribuído a uma das interfaces de rede do nó móvel. No primeiro caso, o túnel é criado entre o HA e o FA, sendo este o responsável por retirar os pacotes do túnel e entregá-los ao nó móvel; no segundo caso, o túnel é criado entre o HA e o nó móvel, método no qual o próprio no móvel retira os pacotes do túnel.

Com o uso do *foreign agent* CoA, o nó móvel utiliza seu endereço de origem (*home address*) como endereço de origem em todos os pacotes IP enviados. Este modo de operação é preferível porque permite a vários nós móveis compartilhar o mesmo

CoA (endereço do FA), permitindo que outros dispositivos utilizem os endereços IPv4 disponíveis na rede.

Com o uso do *co-located* CoA, é possível o funcionamento de um nó móvel sem que seja necessário a presença de um FA. Contudo, os endereços IPv4 disponíveis na rede são utilizados por esses nós, e não por outros possíveis dispositivos, o que pode gerar problemas na distribuição de endereços na sub-rede devido à sua escassez.

A Figura 3.3 demonstra o funcionamento do protocolo IPv4 Móvel.

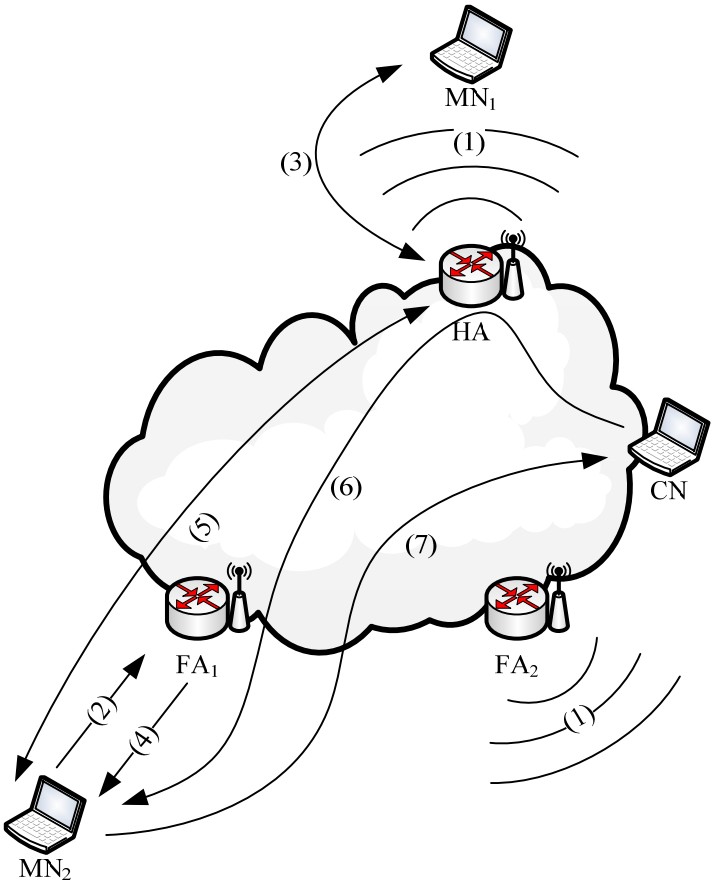

Figura 3.3: Funcionamento do protocolo IPv4 Móvel

(1) Os agentes móveis (FAs e HAs) enviam mensagens *agent advertisement* informando sua presença.

(2) Opcionalmente, um nó móvel pode solicitar uma mensagem *agent advertisement* de um agente móvel através de uma mensagem *agent solicitation.* É através de uma mensagem *agent advertisement* que o nó móvel determina se está em sua rede de origem (no caso do MN_1) ou em uma rede estrangeira (no caso do MN_2).

(3) Se o nó móvel retorna à sua rede de origem (MN_1), estando anteriormente registrado em outra rede, ele efetua o cancelamento do registro com seu HA

através de uma troca de mensagens *registration request* e *registration reply*. O nó móvel, quando está em sua rede de origem, opera sem os mecanismos de mobilidade.

(4) Quando um nó móvel detecta que está em uma rede estrangeira (MN_2), ele obtém um CoA ou através de uma mensagem *agent advertisement* (1) ou através de uma entidade de uma atribuição externa, como, por exemplo, o protocolo DHCP (*co-located* CoA).

(5) O nó móvel operando em uma rede estrangeira registra seu novo CoA com seu HA através de uma troca de mensagens *registration request* e *registration reply*, possivelmente através do FA ao qual está conectado. O registro no HA é realizado por intermédio do FA, no caso de ser utilizado o *foreing* CoA, ou diretamente no HA, no caso de utilização de um *co-located* CoA.

(6) Pacotes enviados ao endereço de origem do nó móvel são interceptados pelo HA, encapsulados, enviados pelo túnel ao CoA do nó, desencapsulados e entregues. O HA deve ser capaz de interceptar todos os pacotes destinados aos nós móveis nele registrados.

(7) No sentido inverso, pacotes enviados pelo nó móvel são entregues utilizando os mecanismos normais de roteamento IP, não necessariamente passando pelo HA. O protocolo IPv4 Móvel também prevê a utilização de túnel reverso (MONTENEGRO, 2001), com o qual os pacotes enviados pelo nó móvel também são encapsulados e enviados primeiramente ao HA, para depois serem encaminhados para o CN.

Dois procedimentos de registro são possíveis no protocolo IPv4 Móvel: o primeiro através do FA, o qual retransmite o registro enviado pelo nó móvel ao seu HA; o segundo diretamente com o HA do nó móvel. Se o nó móvel está registrando seu CoA, deve fazê-lo através do FA; o registro através do FA também deve ser realizado quando o nó móvel, mesmo utilizando um *co-located* CoA, receber uma mensagem *agent advertisement* com o bit R habilitado, o que significa que o registro com o FA é necessário. O registro direito com o HA pode ocorrer quando o nó móvel estiver utilizando um *co-located* CoA e se o nó móvel estiver em sua rede de origem.

Através do pedido de registro (*registration request*), o nó móvel informa ao HA que deseja criar ou atualizar uma associação entre seu endereço de origem (*home address*) com o endereço de sua atual localização (CoA). As mensagens utilizam o

protocolo UDP (*User Datagram Protocol*) para serem enviadas. Juntamente com o cabeçalho do protocolo UDP, seguem as informações relacionadas ao protocolo IPv4 Móvel (Figura 3.4)

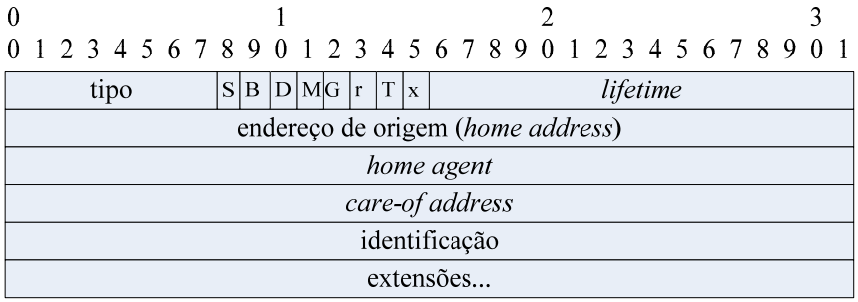

Figura 3.4: Campos do protocolo IPv4 Móvel

Quando com o valor 1, o campo *tipo* identifica uma mensagem *registration request*. O campo *S* identifica se o nó móvel deseja realizar uma associação simultânea (*simultaneous binding*). O campo *B* indica que o nó móvel deseja receber os *broadcasts* da rede de origem. O campo *D* é utilizado para informar que o túnel possui como ponto final o nó móvel (*co-located* CoA). O *T* indica o uso de tunelamento reverso.

O campo de tempo de vida (*lifetime*) indica o número de segundos pelo qual o registro é válido. O valor 0 (zero) informa um *desregistro* e valor 0xffff informa que o registro possui validade indeterminada. Quem determina na realidade qual é o período de tempo no qual um registro é considerado válido é o HA na mensagem *registration reply*. O tempo de vida informado pelo HA é menor ou igual ao presente na mensagem de *registration request*.

O campo de *identificação* é utilizado para ser verificado com a mensagem *registration reply*, a qual também contém esse campo, e para a proteção contra ataques através dessas mensagens. Por fim, o campo *extensões* contém uma ou mais extensões suportadas pelo protocolo IPv4 Móvel, como extensões utilizadas para autenticação das entidades.

Maiores informações sobre os campos e seus possíveis valores estão presentes em Perkins (2002).

3.2 IPv6 Móvel

Enquanto o protocolo IPv4 Móvel permite que os nós sejam acessíveis em uma rede IPv4, o protocolo IPv6 Móvel, descrito em Johnson (2004), permite que os nós

permaneçam acessíveis enquanto se movimentam em uma rede IPv6. Cada nó móvel é identificado de maneira permanente por seu endereço de origem (*home address*), independentemente do seu ponto de conexão. Da mesma forma que o protocolo IPv4 Móvel, quando situados em outro ponto de conexão que não o seu de origem, os nós possuem também um *care-of address* (CoA), o qual informa sua atual posição.

O nó móvel sempre poderá ser endereçado através de seu endereço de origem. Enquanto o nó estiver em sua rede de origem, os pacotes endereçados ao endereço de origem são roteados para a rede de origem do nó utilizando os mecanismos tradicionais de roteamento.

Se o nó estiver conectado a uma rede estrangeira, ele é endereçável por um ou mais *care-of address*, adquirido através de mecanismos convencionais do protocolo IPv6, como a configuração automática de endereços sem a presença de um servidor (*stateless*) ou através de um servidor DHCPv6 (*statefull*) (THOMSON, 1998). Enquanto permanecer em uma rede estrangeira, os pacotes endereçados ao *care-of address* serão roteados ao nó móvel. Ademais, o nó móvel poderá aceitar pacotes endereçados a diversos *care-of address*, como no caso de ser acessível por mais de um *link* quando estiver em movimentação.

Diferentemente da forma em que foi concebido o protocolo IPv4 Móvel (PERKINS, 2002), é possível no protocolo IPv6 Móvel a comunicação direta entre um nó correspondente (nó par com o qual o nó móvel comunica-se, podendo ser tanto móvel quanto fixo) e o nó móvel. Este protocolo permite aos nós IPv6 armazenar a associação entre o endereço de origem e o *care-of address* do nó móvel, possibilitando o envio de pacotes ao nó móvel diretamente através de seu *care-of address*. Para realizar essa operação, é definido um novo protocolo IPv6 e uma nova opção de destino no cabeçalho do pacote IP (DEERING, 1998).

Na otimização de rota, quando há a comunicação direta entre o nó correspondente e o nó móvel, é necessário que o nó móvel registre sua localização, através do *care-of address*, no nó correspondente. Para enviar um pacote a um nó IPv6, o nó correspondente verifica suas associações armazenadas em busca do endereço de destino do pacote. Se é encontrada uma associação, o nó utiliza um novo cabeçalho de roteamento definido em Deering (1998) para enviar o pacote ao nó móvel com o uso do *care-of address* armazenado. Com o uso da otimização de rota, é utilizado o caminho mais curto para a comunicação dos nós, eliminando a dependência e o congestionamento no *home agent* e na rede de origem do nó móvel.

Quando os pacotes são roteados diretamente ao nó móvel, o nó correspondente define o *care-of address* do nó móvel como endereço de destino no cabeçalho do pacote IPv6. De maneira correspondente, o nó móvel define o endereço de origem do pacote IPv6 como seu atual *care-of address*. Em ambos os casos, é incluído o endereço de origem na extensão do cabeçalho IPv6, para que essa comunicação seja transparente para as camadas acima da camada de rede na pilha de protocolos. Desta forma, as aplicações continuam utilizando o endereço de origem do nó para enviar e receber dados.

O protocolo IPv6 Móvel também permite que a comunicação entre o nó correspondente e o nó móvel não ocorra de maneira direta, mas sim através de um túnel bidirecional, à semelhança do protocolo IPv4 Móvel. Neste modo de comunicação, não é necessário que o nó correspondente ofereça suporte ao IPv6 Móvel, tampouco que haja o registro de localização no nó correspondente por parte do nó móvel. Os pacotes endereçados ao nó móvel são roteados ao *home agent* e então enviados através de um túnel ao nó móvel; os pacotes endereçados ao nó correspondente percorrem o caminho inverso até o *home agent* (através de um túnel) e então são encaminhados ao nó correspondente. O tunelamento é realizado com o encapsulamento IPv6 (CONTA, 1998).

Outras diferenças do protocolo IPv6 Móvel em relação ao protocolo IPv4 Móvel são apontadas em Johnson (2004):

- Não há necessidade da existência de *foreign agents*;
- Há o suporte nativo à otimização de rota;
- A otimização de rotas pode operar de maneira segura mesmo sem pré-associações de segurança;
- Há o suporte para a coexistência da otimização de rotas com roteadores que realizam filtro de ingresso;
- A maioria dos pacotes enviados ao nó móvel fora de sua rede de origem utiliza o cabeçalho de roteamento IPv6 ao invés de encapsulamento IP, reduzindo o *overhead* quando comparado ou IPv4 Móvel;
- O protocolo IPv6 Móvel é dissociado de qualquer camada de enlace, uma vez que usa a descoberta de vizinhos do protocolo IPv6 ao invés do protocolo *Address Resolution Protocol* (ARP) utilizado no protocolo IPv4 Móvel.

- O uso do encapsulamento IPv6, juntamente com o cabeçalho de roteamento, faz com que o protocolo IPv6 Móvel não necessite gerenciar o estado de túneis.
- O mecanismo de descoberta dinâmica do endereço do HA presente no protocolo IPv6 Móvel retorna uma única resposta ao nó móvel, enquanto a abordagem de *broadcast* utilizada no protocolo IPv4 retorna endereços separados de cada HA.

O protocolo IPv6 Móvel oferece suporte a múltiplos *home agents*. Um mecanismo conhecido *descoberta dinâmica de endereço do home agent* permite ao nó móvel dinamicamente descobrir o endereço IP do *home agent*, estando na sua rede de origem ou mesmo em uma rede estrangeira.

3.3 Network Mobility (NEMO)

O protocolo para suporte à mobilidade de rede (*Network Mobility* – NEMO) (DEVARAPALLI, 2005) é uma extensão do protocolo IPv6 Móvel. O protocolo permite que redes móveis conectem-se a diferentes pontos na Internet mantendo a continuidade das sessões de comunicação ativas. Com este protocolo, embora haja a mobilidade da rede, os nós conectados à rede móvel sempre são acessíveis, sendo que esta mobilidade se dá de forma transparente aos nós que estão dentro da rede móvel.

O roteador móvel (*Mobile Router* – MR), que é responsável por conectar a rede móvel à Internet e é o *default gateway* para os nós que estão nele conectados, roda o protocolo NEMO juntamente com seu HA. Diferentemente da definição de um nó móvel do Mobile IPv6, o MR possui a capacidade de roteamento entre seu ponto de conexão à Internet (CoA) e a rede que está conectada a ele.

Quando o MR está fora de sua rede de origem e conecta-se a um novo roteador de acesso, ele adquire um CoA e, em seguida, envia uma mensagem *Binding Update* (BU) para seu HA informando o atual ponto de conexão. Ao receber a mensagem BU, o HA cria uma entrada em seu *cache* ligando o endereço de origem (*home address*) do MR com o CoA enviado na mensagem BU. Então, o HA envia uma mensagem *Binding Acknowledgement* (BA) ao MR, informando que os pacotes destinados ao MR serão interceptados e transmitidos através do HA. Finalmente, é criado um túnel bidirecional entre o MR e o HA, por onde todo o tráfego entre os nós conectados ao MR e seus nós

correspondentes passa pelo HA. Através do túnel criado, são trocadas também mensagens do protocolo de roteamento, conforme esteja ou não configurado um protocolo de roteamento intra-domínio entre o MR e o HA.

Um nó móvel pode agir de duas maneiras: como um *host* móvel, onde o HA somente mantém o controle da associação do endereço de origem do nó com seu CoA adquirido no seu novo ponto de conexão, ou como um MR, e, nesse caso, o HA, além de manter a associação entre o endereço de origem com o CoA do MR, mantém informações relacionadas aos prefixos atribuídos à rede móvel (redes que estão conectadas ao MR). A distinção entre os dois modos de operação é representada por um *flag*, o *mobile router flag* (R), enviado nas mensagens de BU e BA e em outras mensagens de controle. Com a *flag* habilitada, o HA pode repassar ao MR pacotes destinados aos nós da rede móvel. Em suma, um MR é uma extensão da definição de um nó móvel no protocolo IPv6 Móvel, no qual há o acréscimo de capacidade de roteamento entre o ponto de conexão (CoA) e a rede móvel conectada ao MR.

A Figura 3.5 apresenta de maneira simplificada o funcionamento do protocolo NEMO. Maiores informações sobre o protocolo podem ser obtidas em Devarapalli (2005).

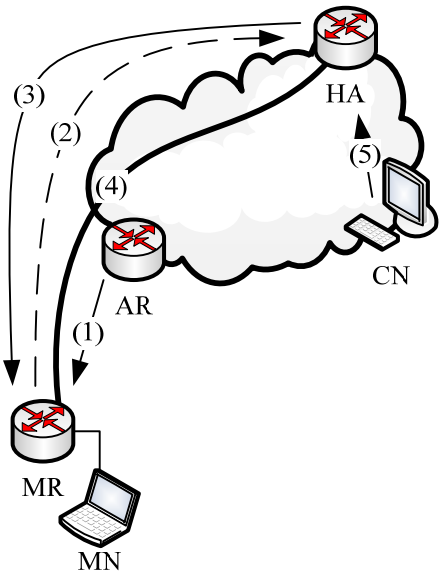

Figura 3.5: Funcionamento do protocolo NEMO

(1) Quando o MR sai de sua rede de origem e conecta-se a um novo roteador de acesso, ele obtém um CoA.

(2) Agindo como um MR ele envia uma mensagem BU para seu HA com a *flag* (R) habilitada informando seu CoA.
(3) Após receber o BU e atualizar suas informações internas, o HA envia uma mensagem BA para o MR. Uma validação com a *flag* (R) habilitada significa que o HA repassará os pacotes destinados à rede móvel.
(4) Uma vez que o processo de atualização (BU) e confirmação (BA) estiver finalizado, um túnel bidirecional é criado entre o HA e o MR. As extremidades desse túnel são o CoA do MR e o endereço IP do HA.
(5) Quando um nó correspondente (CN) envia um pacote a um nó da rede móvel, o pacote é roteado ao HA, o qual possui a localização do MR. Então, o pacote é encapsulado e enviado pelo túnel criado em (4), desencapsulado pelo MR e repassado à rede móvel para o nó de destino (MN). Os pacotes com origem em um nó da rede móvel percorrem o caminho inverso, isto é, são encapsulados, enviados pelo túnel ao HA, desencapsulados e repassados ao CN.

3.4 IPv4 Móvel com Registro Regional

Em Fogelstroem (2007) é descrita uma alteração opcional no protocolo IPv4 Móvel chamada IPv4 Móvel com Registro Regional. Com esta alteração, os registros solicitados pelo nó móvel ficam restritos ao domínio visitado e são realizados através de uma entidade chamada *Gateway Foreign Agent* (GFA), a qual introduz uma hierarquia no domínio visitado. O registro regional reduz a quantidade de mensagens de sinalização à rede de origem do nó móvel, e reduz o atraso na sinalização quando o nó móvel muda de um FA para outro dentro de um mesmo domínio. Desta forma, a quantidade de tráfego na rede de origem é reduzida e o tempo de *handover* no domínio visitado pelo nó móvel é menor.

O endereço do GFA faz parte das mensagens *agent advertisement* enviadas pelos FAs no domínio visitado. Quando um nó móvel encontra-se pela primeira vez em um domínio, ele realiza um registro em sua rede de origem (*home registration*), isto é, um registro com seu HA. Nesse registro, o nó móvel utiliza como seu CoA o endereço do GFA. Quando o nó móvel movimenta-se para um diferente FA dentro do mesmo domínio visitado, não é mais necessário registrar-se com o HA, mas somente se registrar regionalmente com o GFA. Além do primeiro registro em um domínio, o registro com a rede de origem também é realizado quando o nó móvel requisita um

novo HA, quando há alteração do GFA ou para renovar o registro com o HA antes que ele expire.

Além do registro com sua rede de origem, como definido em Perkins (2002), e com a intermediação do GFA no caso do protocolo IPv4 Móvel com registro regional, é definido em Fogelstroem (2007) um registro regional que envolve o nós móveis, os FAs e os GFAs quando a movimentação ocorre dentro de um domínio. Este registro é utilizado pelo nó móvel para comunicar ao GFA sua movimentação dentro de um mesmo domínio.

A alteração no protocolo IPv4 Móvel pode funcionar de dois modos. Na maneira mais simples, os registros regionais (registros no GFA) são realizados de forma transparente ao HA (o endereço do GFA é obtido através das mensagens *agent advertisement* enviadas pelo FA). O outro modo de operação consiste na atribuição dinâmica de um GFA ao nó móvel, mas requer o suporte de todas as entidades da arquitetura de mobilidade (nó móvel, FA, GFA e HA).

O protocolo de registro regional pode suportar um ou mais níveis de hierarquia de FAs abaixo do GFA. No topo da hierarquia, há pelo menos um GFA, que consiste em um FA com características adicionais. Abaixo do GFA, há um ou mais FAs conectados a ele de forma hierárquica.

A Figura 3.6 mostra o funcionamento da arquitetura IPv4 Móvel com Registro Regional.

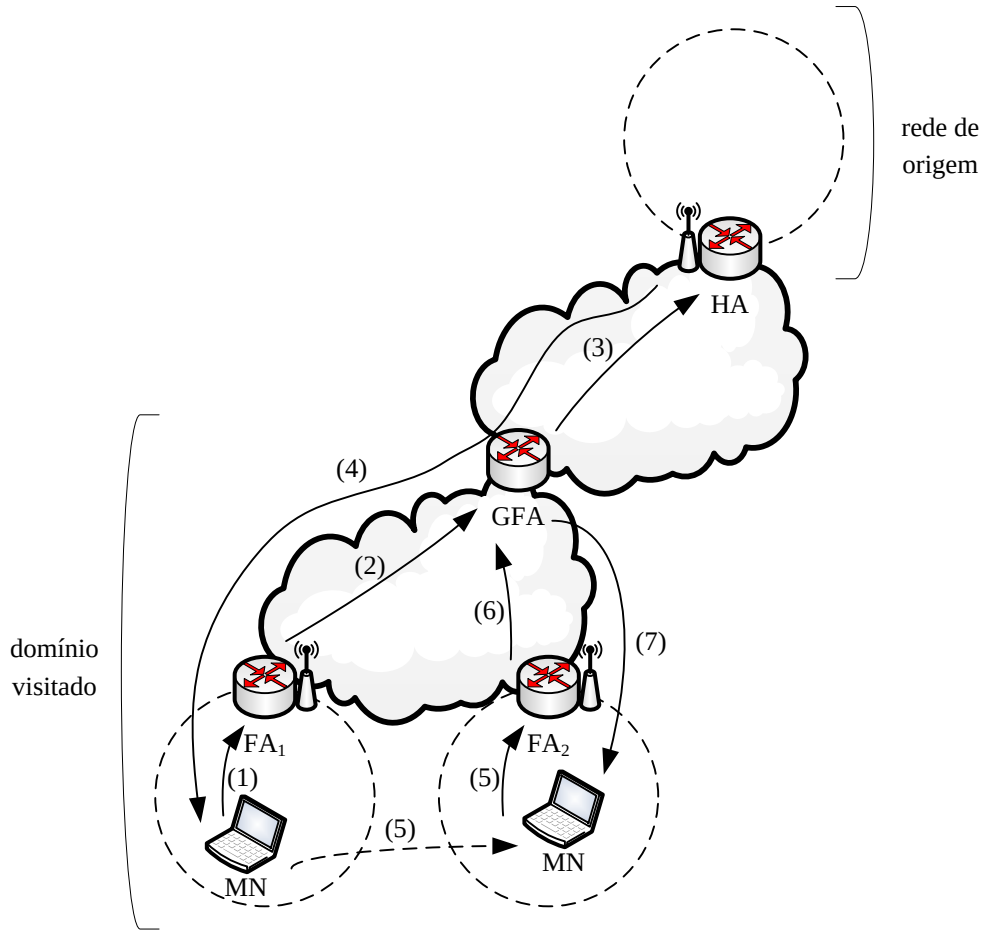

Figura 3.6: Funcionamento do IPv4 Móvel com Registro Regional

(1) Quando o nó móvel está pela primeira vez em um domínio, ele deve realizar um registro com sua rede de origem. Inicialmente, ele requisita um registro regional com o envio de um pedido de registro normal ao FA, mas definindo o CoA como o endereço do GFA.

(2) O FA adiciona ao pedido do nó móvel uma extensão chamada *Hierarchical FA* (HFA), a qual contém o endereço IP do FA, e o retransmite ao GFA apropriado.

(3) O GFA recebe o pedido de registro retransmitido pelo FA, adiciona ao pedido uma extensão contendo seu endereço IP e transmite a mensagem ao HA.

(4) O HA processa o pedido de registro, e envia uma mensagem *registration reply* ao nó móvel contendo a extensão adicionada pelo GFA com seu endereço IP.

(5) Após a movimentação de um nó dentro de um mesmo domínio, ele deve realizar novamente o registro, mas dessa vez através de um registro regional.

(6) Como o CoA local do nó móvel sofreu modificação, o novo FA encaminha o registro regional ao GFA, mas o HA continua usando o endereço do GFA como o CoA do nó móvel.

(7) O GFA responde ao nó móvel com uma mensagem *regional registration reply*.

3.5 IPv6 Móvel Hierárquico

O protocolo IPv6 Móvel Hierárquico (*Hierarchical Mobile* IPv6 – HMIPv6) é uma extensão introduzida em Soliman (2005) para permitir o gerenciamento local da mobilidade com a utilização do protocolo IPv6 Móvel. No protocolo IPv6 Móvel (JOHNSON, 2004), a troca de mensagens de registro não é realizada somente com o HA, mas também com os nós com os quais o nó está se comunicando. Dessa forma, com a introdução de hierarquia no protocolo HMIPv6, a quantidade de tráfego de sinalização entre o nó móvel, seu HA e seus CNs é reduzida.

O protocolo HMIPv6 prevê a existência de uma nova entidade chamada *Mobility Anchor Point* (MAP), a qual pode estar localizada em qualquer local na hierarquia dos roteadores de rede, incluindo os roteadores de acesso. Com o MAP, a quantidade de sinalização do protocolo IPv6 Móvel fora do domínio local é limitada, pois o nó móvel não precisa enviar as mensagens *binding update* para o HA e para os nós correspondentes, mas somente para o MAP local. Além disso, somente uma mensagem *binding update* precisa ser enviada ao MAP para que todo o tráfego seja redirecionado para a nova localização do nó móvel, e não mais para cada nó com o qual o nó móvel esteja se comunicando, como ocorre no protocolo IPv6 Móvel.

Com o protocolo HMIPv6, o nó móvel possui dois endereços: o *Regional Care-of Address* (RCoA), o qual pertence à sub-rede do MAP, e o *On-link Care-of Address* (LCoA), o qual pertence à sub-rede do roteador de acesso ao qual o nó móvel está conectado. O MAP exerce a função de um HA local responsável por associar os endereços RCoA e LCoA do nó móvel.

Conforme a Figura 3.7, o MAP pode prover mobilidade transparente ao nó móvel (em relação ao HA e aos CNs) quando este muda localização do roteador de acesso (*Access Router* – AR) 1 (AR_1) para o roteador de acesso 2 (AR_2). A movimentação ocorre sem que haja a perda da comunicação com os nós correspondentes (CNs).

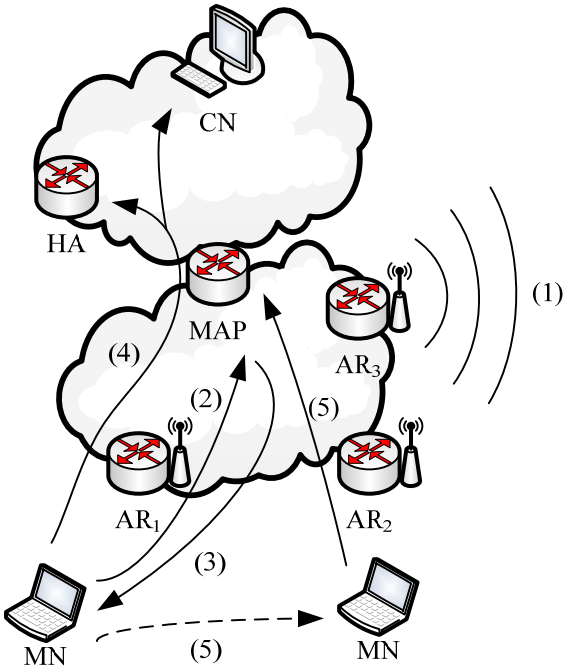

Figura 3.7: Funcionamento do IPv6 Móvel Hierárquico

(1) Um nó móvel, ao entrar em um domínio de um MAP, recebe mensagens *router advertisement* (RA) contendo informações de um ou mais MAPs locais. Também é informado ao nó móvel a distância em relação ao MAP.

(2) O nó móvel inicialmente registra-se com o MAP através de uma mensagem *binding update* (BU) contendo seu RCoA, formado com a informação recebida na mensagem RA, e seu LCoA.

(3) Após associar o RCoA com o LCoA recebidos na mensagem BU, o MAP envia uma mensagem *binding acknowledgement* (BA) ao nó móvel confirmando o registro. Com isso, um túnel bidirecional é criado entre o MAP e o nó móvel, e o MAP pode encaminhar os pacotes destinados ao nó móvel provenientes do HA e dos CNs. Todos os pacotes enviados pelo nó móvel são encaminhados pelo túnel ao MAP. No túnel entre o MAP e o nó móvel, o cabeçalho externo contem o LCoA do nó móvel como endereço de origem e o endereço do MAP como destino. O cabeçalho interno do pacote contém o RCoA do nó móvel como endereço de origem e o nó com o qual o nó móvel está se comunicando como endereço de destino.

(4) Uma vez registrado com o MAP, o nó móvel deve efetuar o registro do seu novo RCoA com seu HA e com seus CNs através de mensagens BU. Estas mensagens associam o endereço de origem do nó móvel com seu RCoA.

(5) Se o nó móvel alterar sua localização dentro do domínio do MAP local, somente é necessário um novo registro com o MAP informando seu novo LCoA. O endereço RCoA permanece o mesmo enquanto o nó móvel estiver conectado ao mesmo domínio do MAP. A verificação do domínio pelo nó móvel é realizada com a informação recebida nas mensagens RAs enviadas pelos ARs, as quais contêm o(s) MAP(s) do domínio.

Capítulo 4

Gerenciamento de Redes Baseado em Políticas

De acordo com Westerinen (2001), o termo política pode ser definido em duas perspectivas. Na primeira delas, política é um objetivo definitivo, caminho ou método de ação para guiar e determinar decisões atuais e futuras; as políticas são implementadas ou executadas dentro de um contexto particular. Dentro da segunda perspectiva, política é um conjunto de regras para administrar, gerenciar e controlar o acesso a recursos de rede. Uma política pode ser representada em diferentes níveis de abstração, desde objetivos do negócio até parâmetros de configuração de um dispositivo.

A arquitetura de controle de admissão baseado em política (YAVATKAR, 2000) define duas entidades para o controle de políticas: o servidor de políticas (*Policy Decision Point* – PDP) e o cliente de políticas (*Policy Enforcement Point* – PEP).

O PDP é a entidade responsável pela tomada de decisões baseadas em políticas, as quais ficam armazenadas em um repositório e são consultadas pelo PDP durante um processo de decisão. Essas decisões podem ser vistas do ponto de vista do processo ou do ponto de vista do resultado: na perspectiva do processo são avaliadas as condições de uma política; na perspectiva do resultado são tomadas ações quando as condições de uma política são consideradas verdadeiras.

O PEP é a entidade responsável por executar decisões baseadas em políticas, decisões estas recebidas pelo PDP. O PEP é responsável por iniciar a conexão com o PDP.

A arquitetura proposta no Capítulo 6 é baseada no modelo de gerenciamento de redes baseado em políticas (PBNM) e utiliza em diferentes pontos os conceitos descritos nas seções deste capítulo. Com o gerenciamento de redes baseado em políticas, é possível reduzir a complexidade do gerenciamento de redes de larga escala e

também facilitar o gerenciamento de dispositivos heterogêneos. Nas seções a seguir são descritos alguns conceitos relacionados a esse modelo de gerenciamento.

4.1 Policy Core Information Model (PCIM)

O modelo *Policy Core Information Model* (PCIM) é um modelo para representação de informações de política. Esse modelo define duas hierarquias de classes: classes estruturais, que representam informações de política e controle de políticas, e classes de associação, que indicam como as instâncias das classes estruturais relacionam-se entre si. As classes de informação e associação definidas no modelo PCIM são suficientemente genéricas para permitir representar políticas relacionadas a temas diversos. Entretanto, Moore (2001) expõe que as aplicações iniciais do IETF utilizando o modelo seriam políticas relacionadas à qualidade de serviço (serviços integrados e serviços diferenciados) e relacionadas ao IPSec (*Internet Protocol Security*).

A maneira preferencial para estender o modelo é através do uso direto das classes *PolicyGroup, PolicyRule* e *PolicyTimePeriodCondition* como base para representar e comunicar informações de política (MOORE, 2001). Subclasses de *PolicyCondition* e *PolicyAction* podem representar definições de condições e ações específicas de alguma aplicação.

As classes que compreendem o modelo PCIM servem, através de especialização, como um modelo extensível de hierarquia de classes para definir políticas. Essas políticas permitem desenvolvedores de aplicações, administradores de rede e administradores de política representar políticas de diferentes tipos. A Figura 4.1 apresenta as principais classes do modelo PCIM.

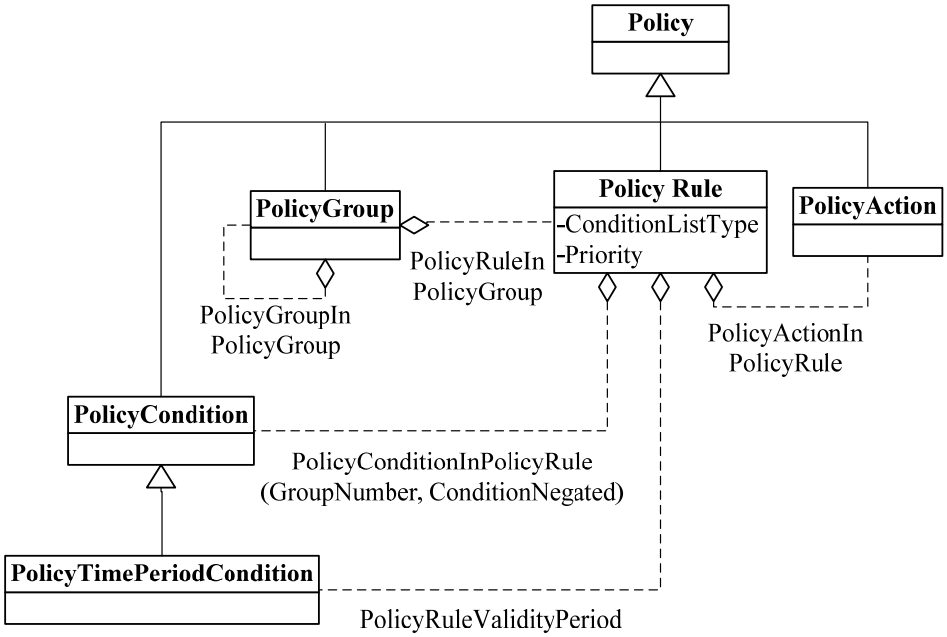

Figura 4.1: Principais classes do modelo PCIM

Através das classes modeladas no modelo PCIM, uma política (classe *Policy*) é definida por um conjunto de regras (classe *PolicyRule*), as quais são agrupadas de forma coerente utilizando-se a classe *PolicyGroup*. Cada regra é composta por um conjunto de condições (classe *PolicyCondition*) e um conjunto de ações (classe *PolicyAction*), que estabelecem a semântica *"se condição então ação"*.

O conjunto de condições associadas a uma regra especifica quando esta é aplicável. Este conjunto de condições pode ser expresso na forma disjuntiva (*disjunctive normal form* – DNF) ou conjuntiva (*conjunctive normal form* – CNF). No primeiro caso, as condições de um mesmo grupo são agrupadas com a operação lógica "E" e os diferentes grupos são unidos com a expressão lógica "OU". No segundo caso, as condições que pertencem a um mesmo grupo são unidas pela operação lógica "OU" e a união de diferentes grupos se dá com a operação lógica "E". A forma como são associadas as condições e os grupos de condições são definidas pelo atributo *ConditionListType* em *PolicyRule*.

Se o conjunto de condições de uma regra for satisfeito e avaliado como verdadeiro, as ações relacionadas a esta regra serão executadas. Para o conjunto de ações associadas a uma regra *PolicyRule*, é possível especificar a ordem de execução e se esta ordem é obrigatória ou recomendada. Além disso, é possível definir que a ordem de execução não é relevante.

As próprias regras *PolicyRule* podem ser priorizadas para, por exemplo, expressar que uma política genérica que contém algumas exceções. A prioridade é determinada pelo atributo *Priority* em *PolicyRule*.

As regras podem também conter condições de período de tempo, as quais são representadas pela classe *PolicyTimePeriodCondition*. A estrutura das classes permite o aninhamento de grupos (*PolicyGroupInPolicyGroup*) para formar uma hierarquia de políticas.

4.2 Policy Core Information Model Extensions (PCIMe)

O modelo *Policy Core Information Model Extensions* (PCIMe) apresenta modificações ao modelo PCIM definido em Moore (2001). Com o modelo PCIMe, novos elementos são introduzidos e elementos previamente definidos são substituídos.

Em Moore (2003) há uma breve lista de mudanças do PCIMe em relação ao PCIM:

- Depreciação e substituição do *PolicyRepository* e suas associações;
- Expansão das maneiras de como *PolicyRules* e *PolicyGroups* são agregados;
- Mudança na representação das prioridades para *PolicyRules*;
- Expansão dos papéis dos *PolicyRoles* e inserção de meios de associação de um *PolicyRole* com um recurso;
- Introdução de combinações de condições de política e combinações de ações de política;
- Introdução de variáveis e valores no modelo;
- Introdução de subclasses para filtragem de cabeçalhos de pacotes;
- Introdução de classes para o filtro de pacotes a nível de dispositivo.

A Figura 4.2 mostra a estrutura geral do PCIMe. As novas classes possuem sombreamento diferenciado.

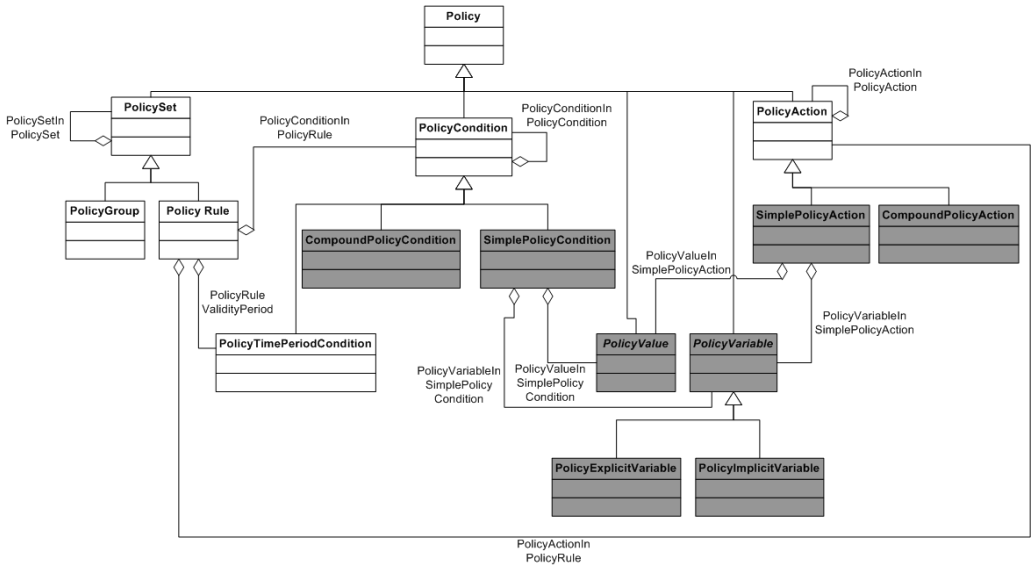

Figura 4.2: Estrutura geral do modelo PCIMe

4.3 Policy Quality of Service Information Model (QPIM)

O modelo *Policy Quality of Service Information Model* (QPIM) especializa o modelo genérico PCIM no que diz respeito à representação de políticas para administração, gerenciamento e controle dos recursos de qualidade de serviço utilizados nas metodologias *intserv* e *diffserv*.

O modelo QPIM não consiste em um modelo completo de representação de políticas, mas somente novas classes que descrevem ações de qualidade de serviço. Em Snir (2003) é sugerida a combinação de elementos QPIM com PCIM para modelar políticas referentes à qualidade de serviço.

Para a definição dessas políticas, três tipos de informação são necessárias (SNIR, 2003):

- Regras de negócio;
- Topologia da rede gerenciada;
- Mecanismo de Qualidade de Serviço a ser utilizado.

As classes QPIM, mostradas na Figura 4.3, modelam diretamente apenas as informações relacionadas ao controle dos mecanismos de qualidade de serviço (*intserv* e *diffserv*), estabelecendo uma forma padronizada para representar ações de QoS e perfis de tráfego. A implementação do modelo QPIM auxilia o mapeamento das regras de

negócio em um modo que define os requisitos para condicionamento dos diferentes tipos de tráfego na rede, seguindo a semântica *"se condição então ação"*.

O modelo QPIM permite a representação de políticas de configuração independentes de dispositivo. A partir deste conceito, é possível o reuso de configurações de QoS, isto é, políticas de configuração destinadas a vários dispositivos que desempenham função similar na rede podem ser definidas uma única vez no sistema. Fica sob responsabilidade do PDP ou PEPs tratar dos detalhes específicos de configuração de cada dispositivo ou fabricante.

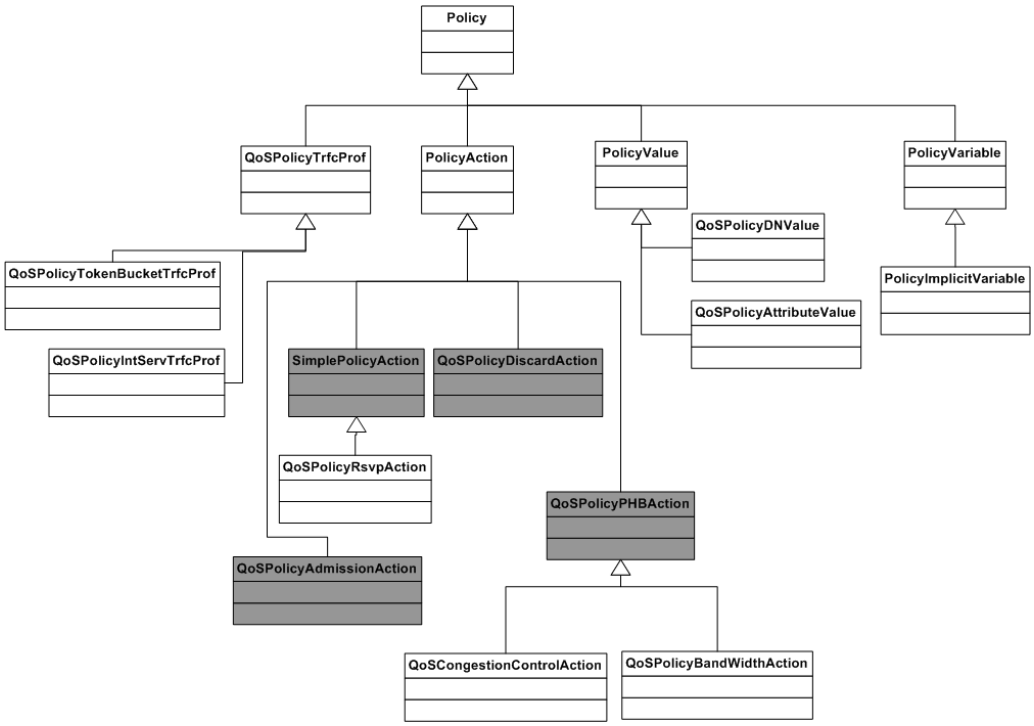

Figura 4.3: Classes do modelo QPIM

4.4 Common Open Policy Service (COPS)

O protocolo *Common Open Policy Service* (COPS) é um protocolo de pergunta/resposta que pode ser utilizado para trocar informações de políticas entre um servidor de políticas (PDP) e seus clientes (PEPs). As principais características do protocolo são (DURHAM, 2000):

- O protocolo emprega um modelo cliente/servidor, onde o PEP envia pedidos, atualizações e remoções de políticas para o PDP, e o PDP retorna decisões ao PEP;

- O protocolo usa o TCP como protocolo de transporte. O protocolo TCP permite a troca confiável de mensagens entre os clientes de política e o servidor;
- Os objetos transportados pelo protocolo são autônomos e auto-identificáveis, possibilitando a transmissão de diferentes objetos sem que haja alteração do protocolo;
- O protocolo fornece segurança através de autenticação e controle de integridade das mensagens. O COPS pode reutilizar protocolos de segurança já existentes, como o IPSec;
- O protocolo é *stateful*, isto é, os pedidos do PEP são instalados ou lembrados pelo PDP até que sejam explicitamente apagados pelo PEP e os pedidos e suas respectivas decisões anteriores instaladas influenciam em novas respostas dadas pelo PDP a um pedido de requisição do PEP.
- O protocolo é *stateful* pois permite ao PDP aplicar e remover políticas do PDP quando for necessário.

A Figura 4.4 mostra como o protocolo COPS é utilizado na arquitetura de controle de admissão baseado em política (YAVATKAR, 2000). As políticas informadas pelo PDP ao PEP dependem do(s) tipo(s) de cliente(s) implementado(s) pelo PEP.

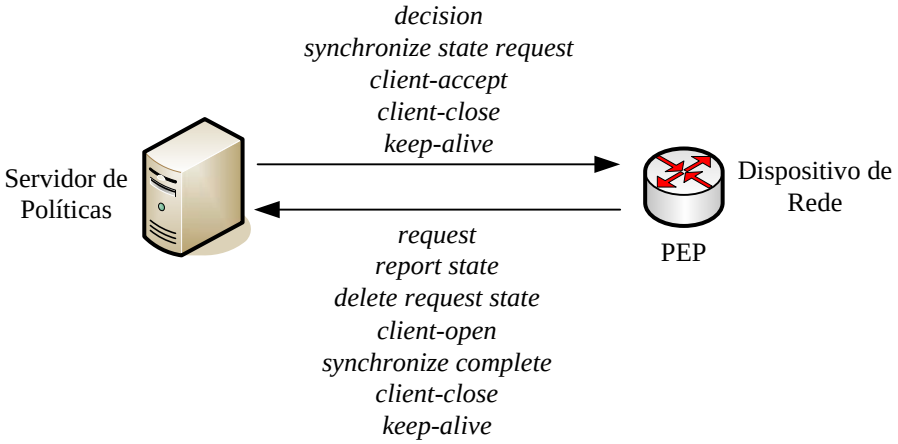

Figura 4.4: Protocolo COPS

Cada mensagem COPS consiste em um cabeçalho definido pelo protocolo seguido de objetos COPS (Figura 4.5).

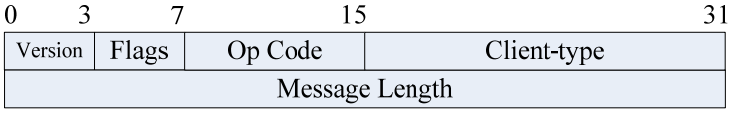

Figura 4.5: Cabeçalho do protocolo COPS

O campo *version* contém a versão do protocolo COPS; como definido em Durham (2000), a versão atual do protocolo é 1 (um). O campo *flags* é utilizado para sinalização; o valor 0x1 para este campo identifica uma mensagem solicitada por outra mensagem COPS. O campo *op code* indica os tipos de operações suportadas pelo protocolo COPS (Tabela 4.1).

Tabela 4.1: Tipo de operações do protocolo COPS

Valor	**Tipo de operação**
1	*Request* (REQ)
2	*Decision* (DEC)
3	*Report State* (RPT)
4	*Delete Request State* (DRQ)
5	*Synchronize State Req* (SSQ)
6	*Client-Open* (OPN).
7	*Client-Accept* (CAT).
8	*Client-Close* (CC).
9	*Keep-Alive* (KA)
10	*Synchronize Complete* (SSC).

O campo *client-type* identifica o tipo de cliente; os objetos encapsulados pelo cabeçalho COPS referem-se ao tipo de cliente definido neste campo. O campo *message length* indica o tamanho (em octetos) da mensagem, englobando o cabeçalho e os objetos encapsulados.

Todos os objetos COPS seguem o mesmo formato apresentado na Figura 4.6.

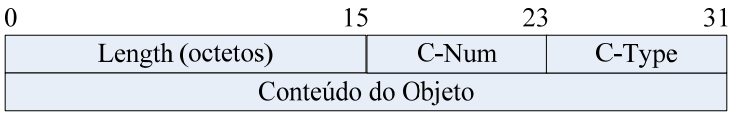

Figura 4.6: Formato do objeto COPS

O campo *length* contém o tamanho do objeto, incluindo o cabeçalho. O campo *c-num* possui identifica a classe da informação contida no objeto e o campo *c-type* identifica o subtipo da informação. Os valores definidos para o campo *c-num* e seus significados estão presentes na Tabela 4.2**.**

Tabela 4.2: Classes do objeto COPS

Valor	Classe
1	*Handle*
2	*Context*
3	*In Interface*
4	*Out Interface*
5	*Reason Code*
6	*Decision*
7	*LPDP Decision*
8	*Error*
9	*Client Specific Info*
10	*Keep-Alive Timer*
11	*PEP Identification*
12	*Report Type*
13	*PDP Redirect Address*
14	*Last PDP Address*
15	*Accounting Timer*
16	*Message Integity*

Os valores de *C-Type* são dependentes dos valores de *C-Num* e estão descritos em Durham (2000).

4.5 Common Open Policy Service for Policy Provisioning (COPS-PR)

O protocolo COPS suporta dois modelos de controle de política: *outsourcing* e *provisioning*.

No modelo *outsourcing*, os eventos quem ocorrem no PEP demandam uma decisão imediata. Neste modelo, o PEP solicita a um servidor de políticas (PDP) qual decisão deve ser tomada. O modelo *outsourcing* é principalmente implementado em redes com suporte a serviços integrados: quando o protocolo de sinalização RSVP solicita a reserva de recursos em um nó da rede (PEP), este consulta o PDP, que por sua vez decide se os parâmetros da reserva podem ser atendidos totalmente, parcialmente ou não podem ser atendidos.

No modelo *provisioning*, as requisições e as decisões ocorrem de forma assíncrona. Neste modelo o PEP, após estabelecer conexão com o PDP, envia uma requisição de configuração contendo suas políticas e parâmetros de configuração. Como resposta para solicitação de configuração, o PDP envia todas as políticas relevantes para o dispositivo no momento atual. A provisão pode ser realizada em partes (atualização de filtros de marcação *diffserv*, por exemplo) ou em totalidade (configuração completa de um dispositivo de rede).

Para o transporte de informações no modelo *provisioning*, é utilizado o protocolo *Common Open Policy Service for Policy Provisioning* (COPS-PR). Neste protocolo, as requisições de política descrevem o PEP e seus parâmetros de configuração; se há uma mudança nesses parâmetros, uma solicitação de atualização é enviada pelo PEP. Como resposta a essas requisições, o PDP mensagens de decisão ao PEP.

O protocolo COPS-PR independe do tipo de política transportada, que pode ser tanto de qualidade de serviço, de segurança etc.

4.6 Policy Information Base (PIB)

A *Policy Information Base* (PIB) é uma estrutura de dados com formato conceitual de árvore para armazenar informações sobre políticas. Sua estrutura consiste em classes de provisão (*Provisioning Classes – PRCs)* e instâncias dessas classes (*Provisioning Instances – PRIs*).

Para uma determinada classe, pode haver uma ou mais instâncias (Figura 4.7). Por exemplo, caso haja necessidade de múltiplos filtros para o controle de acesso, a PRC pode representar um filtro genérico, enquanto cada PRI pode representar um filtro de controle de acesso específico (CHAN, 2001).

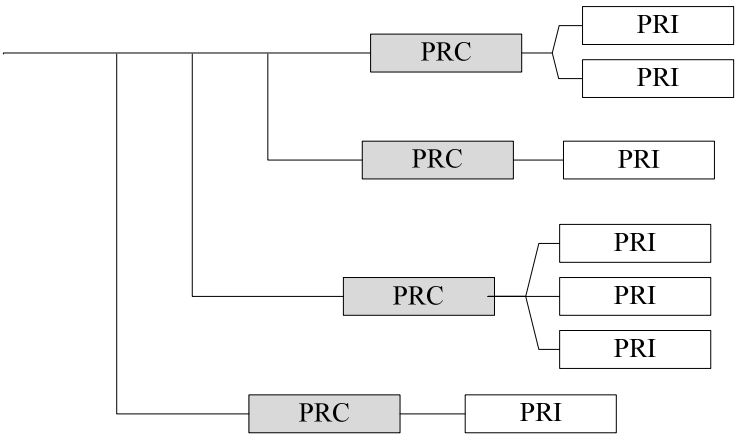

Figura 4.7: Estrutura da PIB

Enquanto cada PRC é unicamente identificada por um *Object Identifier* (OID), cada instância de provisão é identificada por um *Provisioning Instance Identifier* (PRID).

Os módulos PIB são descritos conforme a estrutura especificada na *Structure of Policy Provisioning Information* (SPPI) (MCCLOGHRIE, 2001). Esses módulos contêm informações sobre políticas que podem ser transmitidas para um dispositivo de rede para configuração desse dispositivo.

As informações estruturadas em uma PIB são transportadas pelo protocolo COPS-PR. Essas informações podem identificar o tipo e o propósito das informações em três situações: *i)* políticas não solicitadas que são enviadas do PDP ao PEP; *ii)* políticas solicitadas pelo PEP e provisionadas pelo PDP; e *iii)* políticas notificadas pelo PEP ao PDP. O espaço de nomes (*namespace*) da PIB é comum ao PEP e ao PDP e as instâncias dos dados dentro desse espaço são únicas dentro do escopo de um dado tipo de cliente (*Client-Type*) e um estado da requisição (*Request-State*) por conexão TCP entre um PEP e um PDP.

Um dispositivo de rede pode implementar múltiplos tipos de cliente COPS, mas as instâncias devem ser fornecidas para cada tipo de cliente. Não há compartilhamento de instâncias entre os tipos de cliente implementados pelo PEP, mesmo se as classes instanciadas são do mesmo tipo e compartilham o mesmo identificador de instância.

De acordo com suas funções, as PRCs possuem diferentes formas de acesso. Esse atributo é definido da seguinte forma em McCloghrie (2001): *install*, para indicar uma PRC que pode ser instalada no PEP pelo PDP como informação de provisão; *notify*, para indicar uma PRC que o PEP deve notificar, com todas as suas instâncias e valores de atributos, ao PDP; *install-notify*, para indicar PRCs que possuem ambas as características de *install* e *notify*; e *report-only*, para indicar uma PRC que não possuem nem a característica *install* nem a característica *notify*.

Em Sahita (2003) é definido um conjunto de PRCs e convenções textuais (PIB *Framework*) que são comuns a todos os clientes que recebem provisão de políticas através do protocolo COPS-PR. Em Chan (2003), é descrita uma PIB para dispositivos que implementam a arquitetura de serviços diferenciados (PIB *Diffserv*), onde as PRCs fornecem políticas para controlar recursos que implementem a arquitetura *diffserv*. Finalmente, em Rawlins (2003), são descritas PRCs para controlar o monitoramento e a emissão de relatórios do uso de políticas presentes no dispositivo (PIB *Framework Feedback*).

4.6.1 PIB Framework

O protocolo COPS-PR suporta múltiplos tipos de clientes, os quais podem provisionar políticas de diferentes domínios, como qualidade de serviço, VPNs (*Virtual Private Networks*) ou segurança. Cada cliente possui um conjunto independente de módulos PIB que podem conter informações em comum. Essas informações em comum, embora presentes em cada módulo PIB de um domínio específico, são descritas de maneira genérica em Sahita (2003) através da PIB *Framework.*

A PIB *Framework* define quatro grupos de PRCs: *Base PIB, Device Capabilities, Classifier* e *Marker*. A Figura 4.8 apresenta as classes da PIB *Framework*, na qual as classes sem preenchimento são do tipo *notify*, as classes com preenchimento claro são do tipo *install* e as classes com preenchimento escuro são do tipo *install/notify*.

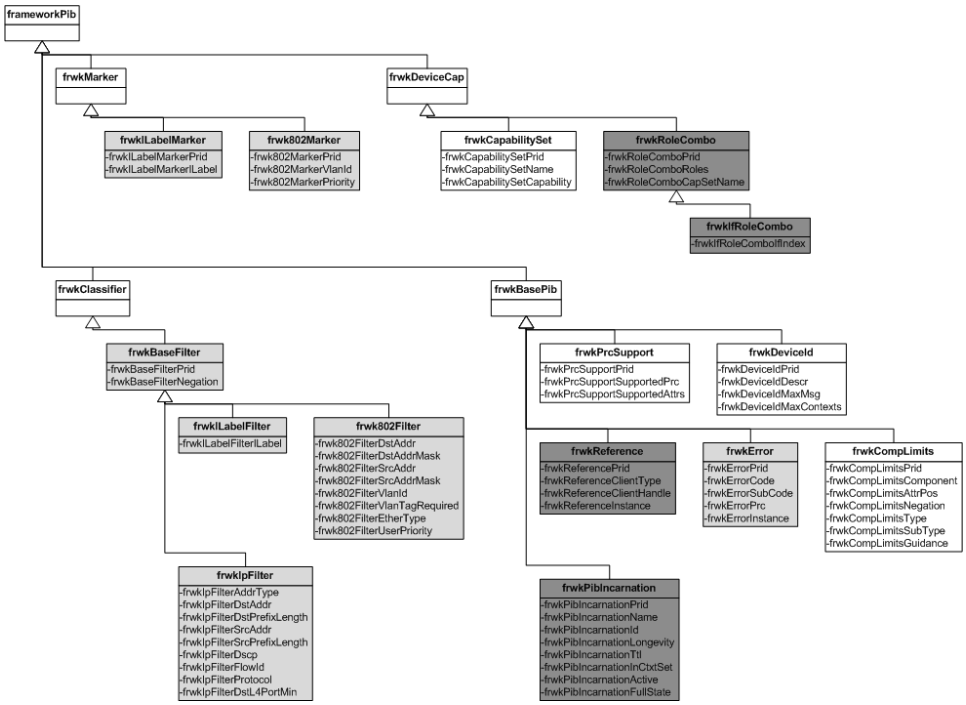

Figura 4.8: Classes da PIB Framework

O grupo *Base PIB* descreve as PRCs suportadas pelo PEP, as limitações das classes e/ou atributos e a configuração atual dessas PRCs.

O grupo *Device Capabilities* contém as PRCs que descrevem as características das interfaces do dispositivo em questão e a combinação dos papéis a elas associadas.

O grupo *Classifier* contém filtros IP, filtros IEEE 802 e o classificador *Internal Lable*.

O grupo *Marker* contém as PRCs que representam o marcador 802 e o marcador *Internal Lable*. O marcador 802 pode ser aplicado em pacotes 802 que necessitam de um identificador de VLAN (*Virtual Local Area Network*) e/ou um valor de prioridade. O marcador *Internal Label* é aplicado ao tráfego para identificá-lo com um rótulo de um dispositivo específico. Esse rótulo é usado na diferenciação de um fluxo ingresso após a agregação com outros fluxos.

4.6.2 PIB Diffserv

Para configurar as políticas relacionadas ao domínio de qualidade de serviço, mais especificamente ao domínio *diffserv*, é definida em Chan (2003) a PIB *Diffserv*, a qual consiste em um conjunto de PRCs a serem utilizadas por um cliente do protocolo COPS-PR do tipo *diffserv*.

A PIB *Diffserv* é projetada de acordo com o modelo presente em Bernet (2002), o qual descreve como são modeladas as interfaces de ingresso e egresso de um roteador de “n” portas. A configuração e gerenciamento de uma interface *diffserv* seguem a estrutura de um *Traffic Conditioning Block* (TCB), que é composto por zero ou mais classificadores, medidores, ações, algoritmos de descarte, filas e escalonadores.

Para representar os elementos de um TCB e sua seqüência, o atributo “*next*” de cada elemento indica o próximo passo no processamento *diffserv*, seja ele um classificador, um medidor, uma ação, um algoritmo de descarte, uma fila, um escalonador ou uma decisão para enviar um pacote. Dentro da PIB, a indicação do processamento *diffserv* que um pacote possa receber é expressa através de um *data path*.

Compõe a PIB *Diffserv* PRCs que representam elementos funcionais no *data path* (classificadores, medidores, ações etc.) e classes que especificam parâmetros a serem aplicados a um determinado tipo de elemento funcional (por exemplo, um medidor *Token Bucket* ou um marcador DSCP). A representação em classes separadas dos parâmetros dos elementos funcionais permite o reuso dessas classes por múltiplas políticas.

As classes da PIB *Diffserv* são divididas em dois grupos: *dsCapability* e *dsPolicy*. A Figura 4.9 apresenta as classes da PIB *Diffserv*, na qual as classes sem preenchimento são do tipo *notify* e as classes com preenchimento são do tipo *install*.

As classes do grupo *dsCapability* são do tipo *notify* e descrevem as capacidades e limitações do dispositivo, utilizando a estrutura genérica e extensível PIB *Framework*.

Através deste grupo o PEP informa ao PDP quais classes (PRCs) são implementas em sua PIB, indicando os elementos funcionais que podem ser configurados no dispositivo.

O grupo *dsPolicy* contém classes do tipo *install* que são utilizadas para definir a seqüência de tratamento a ser aplicada aos pacotes e os parâmetros destes tratamentos.

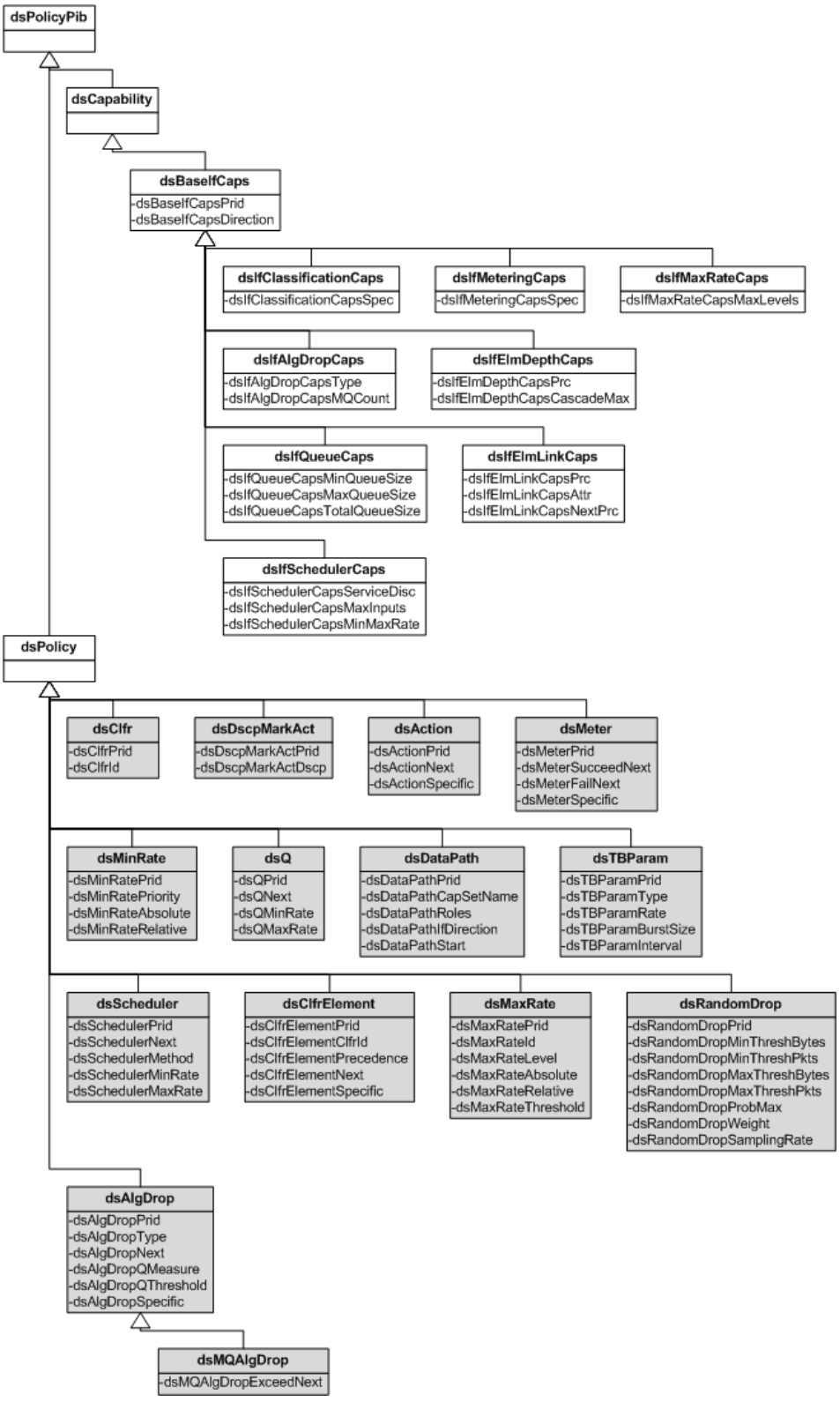

Figura 4.9: Classes da PIB *Diffserv*

4.6.3 PIB Framework Feedback

Em Rawlins (2003) é proposta uma PIB para monitorar e fornecer relatórios de políticas nos dispositivos de rede. Além das classes de monitoramento e de relatórios de políticas, são definidas classes para controlar o intervalo, suspensão, recomeço e solicitação dos relatórios.

Para definir quais as informações o PEP deve monitorar, registrar e relatar posteriormente, há três tipos básicos de políticas: a política de critérios de seleção, política de uso e a política de associação (*feedback report linkage*).

A política de critério de seleção é fornecida pelo PDP e define as condições usadas pelo PEP para monitorar e registrar uma política de uso.

A política de uso define quais atributos são registrados pelo PEP. Usualmente, esta política especifica contadores relacionados a uma ação específica, como o número de pacotes descartados. Duas classes de política de uso são definidas em Rawlins (2003): *frwkFeedbackTraffic* e *frwkFeedbackIfTraffic*.

Para associar a política de uso com a política de critérios de seleção é definida uma terceira política de associação, através da classe *frwkFeedbackLinkTable*, que especifica também quando os relatórios devem ser emitidos. A classe *frwkFeedbackLinkTable* possui como atributos o identificador da instância da política de critério de seleção bem como o identificador da PRC de uso. Detalhes de todas as classes da *PIB Framework Feedback*, bem como seus atributos, estão detalhados em Rawlins (2003).

4.7 Arquitetura para Gerenciamento de QoS Baseado em Políticas

O trabalho presente em Beller (2005) propõe uma arquitetura de gerenciamento de redes baseado em políticas (PBNM) para automatizar os processos de geração e distribuição de configuração para os dispositivos em um ambiente *diffserv*. A arquitetura é baseada nos padrões do IETF e introduz um novo modelo de política de alto nível para simplificar o processo de descrição das políticas de QoS.

A arquitetura é definida em três camadas: modelo de política de alto nível (que estende o modelo PCIM/PCIMe), modelo de política de configuração independente de dispositivo (que estende o modelo QPIM) e um modelo de política dependente de dispositivo (baseado na estrutura da PIB *Diffserv*). As políticas de alto nível inseridas no sistema pelo administrador são convertidas para políticas de configuração através da

execução de um processo de tradução. As políticas de configuração são armazenadas no repositório que é acessado pelo servidor de políticas (PDP) durante o processo de decisão, o qual é executado para responder a solicitação de configuração de um dispositivo de rede (PEP). O processo de decisão leva em consideração a função desempenhada pelas interfaces do dispositivo e suas capacidades para selecionar as políticas de configuração e convertê-las em instâncias das tabelas da DiffServ PIB. Para comunicação entre PEP e PDP é implementado o protocolo COPS e sua extensão COPS-PR.

Os elementos da arquitetura são descritos por documentos XML (*Extensible Markup Language*). O conceito de reuso de informações é aplicado em todas as camadas da arquitetura, considerando dois contextos diferentes: reaproveitamento dos objetos CIM (*Common Information Model*) normalmente cadastrados nos sistemas e utilização de contêineres que armazenam informações reutilizáveis de políticas. Para implementação da proposta, os modelos são mapeados em esquemas XML e o reuso de informação é possível a partir de referências XPointer (*Extensible Markup Pointer Language*).

A Figura 4.10 apresenta a arquitetura proposta em Beller (2005).

No modelo de política de alto nível (HLPM – *High Level Policy Model*), o administrador expressa os objetivos de negócio através de regras que associam usuários, aplicações e servidores a níveis de serviço, durante um determinado período de tempo.

O modelo de política de nível de configuração (CLPM – *Configuration Level Policy Model*) possibilita a representação tanto da identificação de tráfego (condições) como do nível de serviço correspondente (ações). As condições definem a filtragem de pacotes IP e as ações são especificadas através de parâmetros de QoS, tais como controle de congestionamento e alocação de largura de banda. No modelo de configuração, um nível de serviço é representado por um conjunto ordenado de ações QPIM.

A Diffserv PIB representa o nível de configuração dependente de dispositivo, ou seja, ela é selecionada conforme o papel das interfaces gerenciadas e a capacidades dessas interfaces.

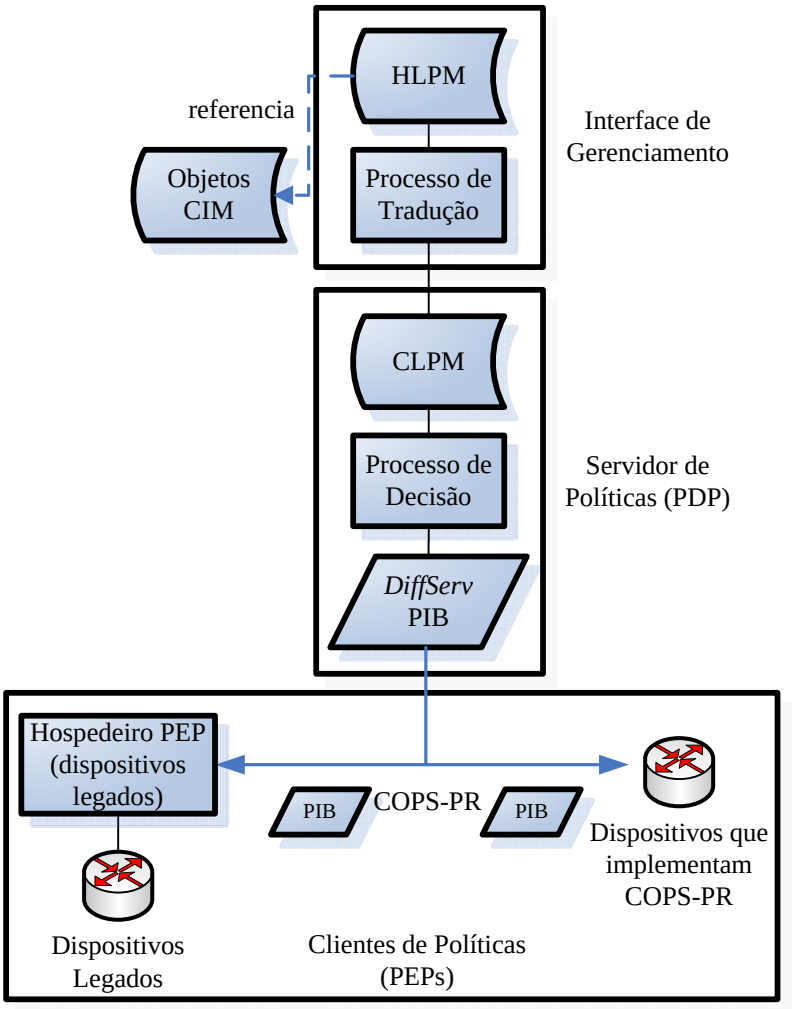

Figura 4.10: Visão geral da arquitetura PBNM

Capítulo 5

Trabalhos correlatos

Outros trabalhos também abordam a integração de qualidade de serviço com mobilidade. Para facilitar o entendimento das propostas presentes na literatura e posteriormente compará-las à arquitetura proposta neste trabalho, elas são classificadas e diferenciadas da seguinte forma: *i)* qual é a arquitetura de qualidade de serviço (serviços diferenciados ou serviços integrados); *ii)* qual é o cenário de mobilidade analisado (macro- ou micro-mobilidade); *iii)* qual arquitetura de gerenciamento de mobilidade é utilizada (IP Móvel, NEMO); *iv)* qual é a versão do protocolo IP (IPv4 ou IPv6), o que também afeta a arquitetura de gerenciamento de mobilidade escolhida; *v)* se há ou não há protocolo de sinalização e como essa sinalização é feita, dividindo as propostas onde o SLA é estático ou pode haver negociação dinâmica do SLA por parte do nó móvel.

5.1 QoS Provisioning for Mobile IP Users

Em Stattenberger (2001a), é proposto um protocolo de sinalização para que os usuários móveis contatem o *Bandwidth Broker* (BB) para negociar o QoS para um determinado fluxo. O protocolo de sinalização é também utilizado para a transmissão de SLSs (*Service Level Specifications*) entre BBs de diferentes domínios.

Um usuário móvel pode conectar-se a diferentes redes de acesso pertencentes a diferentes domínios administrativos. Uma vez que o usuário somente negocia um SLS com seu domínio de origem, mas quer manter o nível de serviço independentemente do domínio onde ele esteja conectado, o SLS negociado deve ser transmitido aos domínios visitados pelo usuário móvel. A Figura 5.1 mostra o cenário utilizado na proposta.

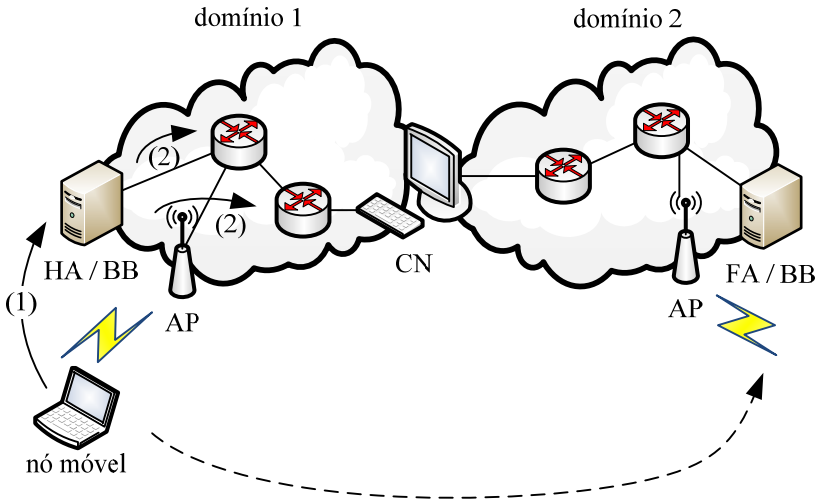

Figura 5.1: Cenário da proposta em Stattenberger (2001a)

Nesta proposta, a negociação do SLS começa quando um usuário móvel envia um pacote contendo a largura de banda desejada e algumas informações de alto nível, como a sensibilidade ao atraso é à perda de pacotes, ao BB da sua rede de origem. Essa sinalização é realizada após o registro com seu HA (*Home Agent*) utilizando a arquitetura IP Móvel (1). Posteriormente, o BB tenta configurar os roteadores de acordo com os requisitos do usuário e com a topologia da rede (2). A troca de mensagens entre o usuário móvel e o BB é realizada com o protocolo TCP.

Para a comunicação entre o nó móvel e o BB e também para a comunicação entre BBs, é definido um descritor de fluxos abstrato (SLS) que pode ser mapeado para diferentes metodologias de qualidade de serviço pelo BB, dependendo da configuração da rede. No caso do modelo proposto, é utilizada a metodologia *diffserv*. A Tabela 5.1 apresenta o formato do pacote para a sinalização do SLS. Maiores informações estão em Stattenberger (2001a).

A informação fornecida pelo SLS é uma especificação de alto nível do nível de serviço. Esta informação pode ser traduzida para os parâmetros de configuração de um roteador como tamanho de fila, largura de banda etc. de diversas maneiras. Na proposta, é utilizada a abordagem de Stattenberger (2001b) para realizar essa tradução do SLS para a configuração dos roteadores. De acordo com o tipo de fluxo do usuário móvel e com os requisitos de QoS, os pacotes do usuário móvel são marcados com uma classe AF ou mesmo com a classe EF da metodologia *diffserv*.

Tabela 5.1: Formato do pacote para a sinalização do SLS

unsigned long	Endereço de origem
unsigned short	Porta de origem
unsigned long	Endereço de destino
unsigned short	Porta de destino
unsigned chart	ID do protocolo (TCP ou UDP)
Double	Largura de banda (média)
Double	Largura de banda (pico)
Boleano	Tempo-real (sensibilidade ao atraso e ao *jitter*)
Boleano	Perda (sensibilidade à perda de pacotes)
unsigned short	ID do fluxo
unsigned long	*Status* do fluxo
unsigned long	Tempo de início do fluxo (absoluto)
unsigned long	Tempo de término do fluxo (absoluto)
unsigned long	Tempo de início do fluxo (relativo)
unsigned long	Tempo de término do fluxo (relativo)

Quando o usuário móvel desloca-se para um domínio estrangeiro, após obter o seu CoA (*Care-of Address*), ele sinaliza o BB deste domínio, que entra em contato com o BB da rede de origem do nó móvel, conhecido através do endereço IP de origem do usuário, para obter o SLS. De posse do CoA e do SLS do usuário, o BB da rede estrangeira pode estabelecer o serviço na rede estrangeira. A Figura 5.2 traz a seqüência de mensagens trocadas entra as entidades durante a mudança de rede de um usuário móvel.

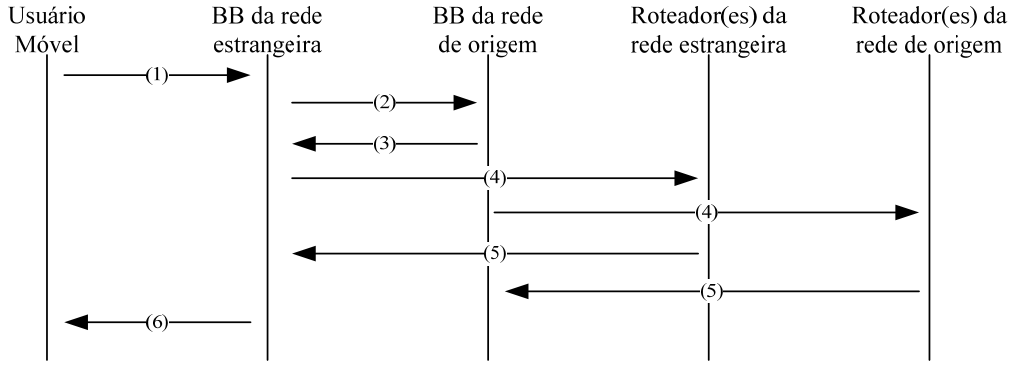

Figura 5.2: Seqüência de mensagens para a transferência do SLS do usuário móvel

(1) O usuário móvel requisita ao BB da rede estrangeira a transferência do seu SLS.

(2) O BB da rede estrangeira solicita o SLS para o BB da rede de origem do usuário móvel. O BB de origem é conhecido através do endereço IP de origem do usuário móvel, enviado para o BB da rede estrangeira em (1). Esse mesmo endereço IP é utilizado na solicitação do SLS.

(3) O BB da rede de origem transmite o SLS ao BB da rede estrangeira com o pacote da Tabela 5.1.

(4) O BB da rede estrangeira, com o CoA do usuário móvel, configura os roteadores que estão na sua rede. O BB da rede de origem reconfigura os roteadores de sua rede para liberar os recursos que antes estavam alocados para o usuário móvel.
(5) Os roteadores informam aos BBs se as configurações foram ou não aplicadas com sucesso.
(6) O BB da rede estrangeira informa ao usuário móvel se houve sucesso ou falha na transferência do SLS.

Alternativamente, o usuário móvel pode estabelecer um SLS totalmente novo com o BB da rede estrangeira e não utilizar seu SLS da rede de origem. Este processo é idêntico ao processo de negociação do QoS na rede de origem do usuário móvel, mas, neste caso, a comunicação é realizada com o BB da rede estrangeira.

A Tabela 5.2 apresenta um resumo da proposta, conforme os itens citados no início deste capítulo.

Tabela 5.2: Resumo da proposta em Stattenberger (2001a)

Arquitetura de QoS	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
Serviços diferenciados	Macro-mobilidade	IP Móvel	IPv4	Sim. Comunicação via TCP

5.2 A Dynamic QoS Provisioning Model for Network Mobility

Em Noor (2006), é proposto um modelo dinâmico de provisão de QoS para a mobilidade de rede. A mobilidade de rede consiste em manter a conectividade da rede ao mesmo tempo em que ela se move. Diferentemente da mobilidade de usuário, onde a movimentação é verificada nos nós móveis, a mobilidade de rede é verificada nos dispositivos que fornecem conectividade aos nós (por exemplo, roteadores). No caso da mobilidade de rede, a mobilidade é transparente aos nós que estão conectados na rede móvel.

Para prover conectividade, uma questão importante é garantir que a rede visitada possua recursos suficientes para atender à rede móvel. Idealmente, o processo de reserva de recursos deve ser feito anteriormente ao processo de *handover* (NOOR, 2006). É importante, deste modo, prever a localização da rede móvel para que seja possível identificar qual será a que rede visitada e então alocar os recursos necessários

previamente. Caso a rede visitada não possa atender aos requisitos da rede móvel, é preciso que esta procure outra rede mais próxima a qual possa se conectar e que então os requisitos sejam atendidos.

Para realizar a reserva de recursos de forma antecipada, deve-se prever a que(ais) rede(s) uma rede móvel pode se conectar. Para simplificar esse aspecto de previsão de mobilidade de uma rede móvel, adota-se em Noor (2006) um comportamento previsível de mobilidade, que consiste na movimentação da rede móvel dentro de um trem, onde é possível saber de antemão qual é o caminho percorrido pela rede móvel e a que redes esta rede móvel poderá se conectar.

Além de ser necessário saber a futura localização da rede móvel para realizar a reserva de recursos, é preciso também definir a quantidade de recursos a ser reservada. A proposta pressupõe padrões de tráfego conhecidos (durante e fora de horários de pico). Conhecidos a movimentação da rede móvel e os padrões de tráfego, é possível reservar recursos de forma antecipada, evitando assim o descarte de pacotes devido ao *handover* e à ausência de QoS.

Neste trabalho, os dispositivos de comunicação são divididos em dois domínios: o domínio de rede e o domínio do nó (Figura 5.3). O domínio de rede consiste nos pontos de acesso (AP), roteadores de acesso (AR), *home agent* e nó correspondente (CN). O domínio do nó consiste nos roteadores móveis (MRs) configurados com o protocolo NEMO e os nós da rede móvel, estes que formam uma rede móvel (roteador e nós). Os APs que estão nos domínios de rede do trem provêm acesso aos roteadores móveis. Estes domínios estão sob a mesma administração.

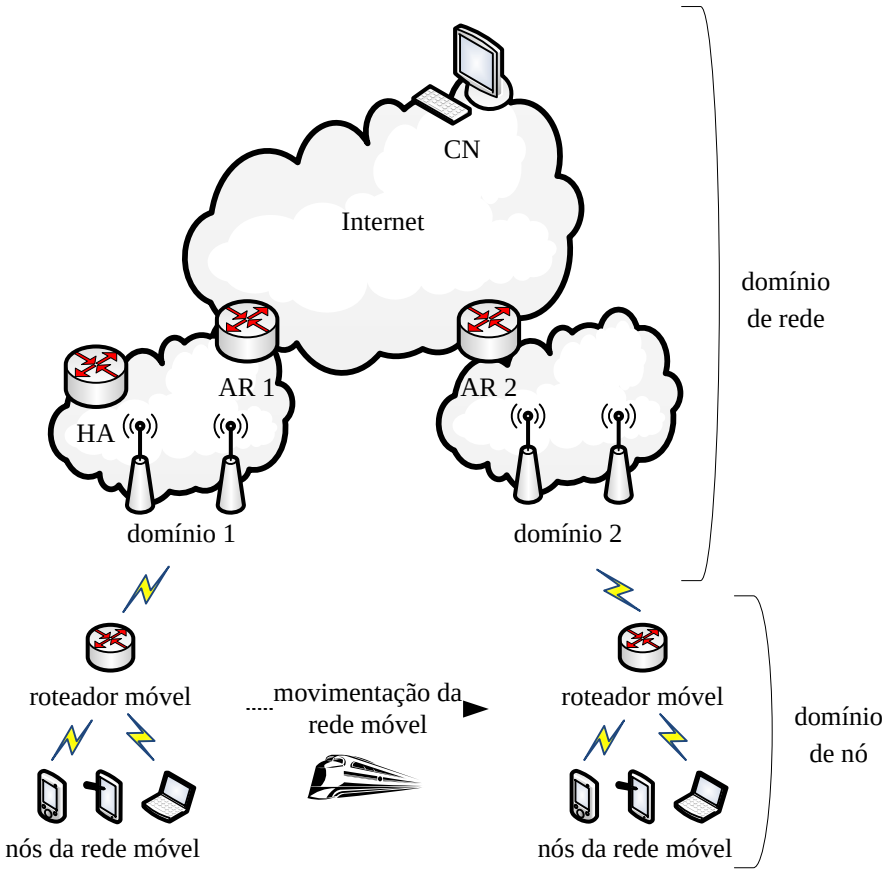

Figura 5.3: Cenário do modelo em Noor (2006)

O processo de provisão de recursos é divido em duas etapas: antes do processo de *handover* e depois do processo de *handover*. O primeiro é realizado no domínio de rede, onde o roteador móvel move-se de um domínio de rede do trem a outro. O segundo é realizado no domínio de nó, onde o roteador móvel aloca recursos para cada classe de tráfego.

No cenário da proposta, no domínio de nó, a rede móvel é o trem e o nó móvel pode ser um passageiro com um computador, PDA ou celular. A conectividade à Internet é fornecida aos nós móveis pelo roteador móvel configurado com o protocolo NEMO. Os serviços fornecidos de acesso à Internet variam de acordo com o usuário, se ele é um assinante ou não. De acordo com o tipo de assinatura e com o tipo de tráfego, os pacotes são marcados com uma classe *diffserv*: EF para aplicações de tempo real, como VoIP, AF para voz ou vídeo unidirecional e BE para os demais casos.

Para suportar QoS no protocolo IPv6 Móvel, em Chaskar (2001) é proposto um campo opcional *QoS object*. A idéia básica é incluir o campo opcional *QoS object* na extensão *hop-by-hop* do cabeçalho IPv6 nos pacotes que são enviados na mesma direção que os fluxos sensíveis ao QoS enviados pelo MN. Uma vez que a mensagem *binding*

update (BU) é enviada assim que o MN adquire seu novo CoA após o handover, o campo opcional *QoS object* é enviado juntamente com a mensagem, portando informações como os requisitos de QoS para as classes de tráfego agregado, volume de tráfego, classificação dos pacotes e marcação específica para os nós das rede móvel. O mesmo ocorre com a mensagem *binding acknowlegment* (BA), enviada do CN ao MN. Como no IPv6 Móvel essas mensagens são fim-a-fim, os requisitos de QoS também ficam conhecidos por todo o caminho percorrido pelas mensagens originadas no MN e destinadas ao MN partindo do CN. Dependendo da metodologia de QoS utilizada nos domínios por onde os pacotes passam, o campo *QoS Object* é analisado e os roteadores configuram o tratamento para esses pacotes.

O campo opcional *QoS Object* pode conter zero ou mais objetos. A composição de um objeto é mostrado na Figura 5.4. Nesta proposta, é adicionado ao objeto de Chaskar (2001) o campo para marcação de pacotes. Os pacotes pertencentes à mesma classe de tráfego são agregados em um mesmo *QoS Object*. Maiores explicações sobre cada campo podem ser encontradas em Chaskar (2001).

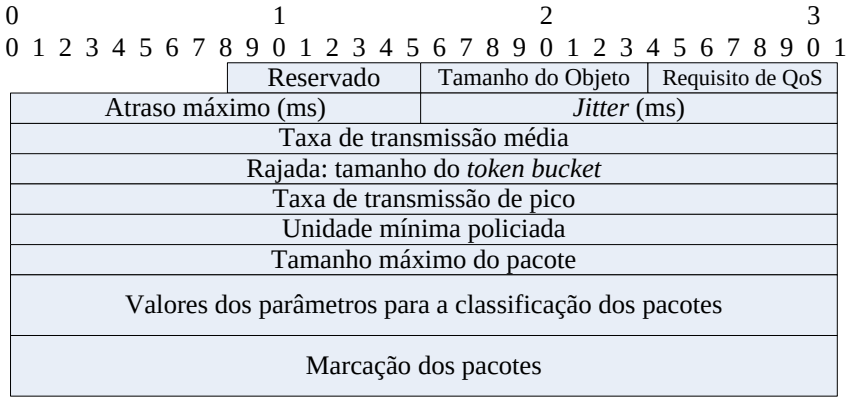

Figura 5.4: *QoS Object* modificado

A Figura 5.5 traz a seqüência de troca de mensagens entre as entidades.

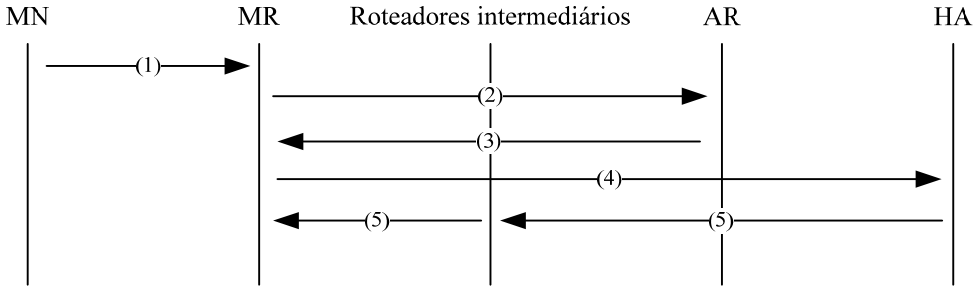

Figura 5.5: Troca de mensagens em Noor (2006)

(1) Os MNs informam o tipo de serviço ao MR para que os pacotes sejam devidamente marcados. O MR realiza a marcação dos pacotes, atribuindo-os a uma classe *diffserv*, de acordo com o serviço requerido.

(2) De acordo com a entrada de novos MNs, o MR solicita ao AR uma nova reserva de recursos. Durante os horários de pico, o uso dos recursos é maior do que os outros horários, e, com o padrão de tráfego previamente estimado, é possível estimar a quantidade de recursos necessária quando houver uma nova conexão após o *handover*.

(3) No momento do *handover* da rede móvel, o AR informa um novo CoA ao MR.

(4) De posse do novo CoA, o MR envia uma mensagem BU juntamente com o *QoS Object* para o HA. O *QoS Object* é enviado com a opção de destino, onde somente o destinatário da mensagem analisa-o. As mesmas classes de tráfego são agregadas em um único *QoS Object*.

(5) Após atualizar suas entradas nas tabelas de endereçamento, o *QoS Object* é enviado juntamente com a mensagem BA ao MR, mas com a opção *hop-by-hop*, onde os roteadores intermediários no fluxo dos pacotes são informados sobre os requisitos de QoS e realizam a reserva de recursos necessária. Desse modo, todos os roteadores por onde passam os fluxos dos MNs possuem a informação de como tratar os pacotes marcados pelo MR. Enquanto os requisitos de QoS não são repassados aos dispositivos, os pacotes recebem o tratamento *default*, isto é, *best effort*.

A Tabela 5.3 apresenta um resumo da proposta.

Tabela 5.3: Resumo da proposta em Noor (2006)

Arquitetura de QoS	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
Serviços diferenciados	Macro-mobilidade	NEMO	IPv6	Sim. Juntamente com o *binding update* da arquitetura de gerenciamento de mobilidade

5.3 Quality of Service and Mobility for the Wireless Internet

Em García-Macías (2001) é proposta uma arquitetura hierárquica de QoS para a metodologia *diffserv* em redes sem fio. Neste trabalho, o modelo *diffserv* é estendido para redes de acesso sem fio de tal modo a prover qualidade de serviço consistente aos nós móveis.

Em redes sem fio, o canal de rádio é um ponto crítico para a garantia de qualidade de serviço e pode afetar em grande parte o desempenho da arquitetura. Isso se deve, em parte, ao método de acesso DCF (*Distributed Coordination Function*), que distribui de igual maneira o canal de comunicação a todos os usuários, sem contar também que o desempenho do canal de comunicação possui oscilações em seu desempenho dependendo das condições do ambiente e dos usuários.

No método de acesso *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA), o nó deve esperar um período de tempo (DIFS – *Distributed Inter Frame Space*) antes de transmitir, se o meio estiver livre. O receptor, ao receber o pacote corretamente, deve esperar outro período de tempo (SIFS – *Short Inter Frame Space*) antes de transmitir um ACK (*Acknowledgment*) do pacote recebido. Se o ACK não for recebido, presume-se que houve uma colisão e, após outro período aleatório de tempo, o nó tenta retransmitir o pacote. Conforme o número de usuários aumenta, a concorrência ao acesso ao meio também aumenta, ocorrendo maiores colisões e reduzindo a utilização do meio. Desse modo, o trabalho limita, em primeiro lugar, o número de nós que concorrem no acesso ao meio para garantir QoS em um meio sem fio.

Em segundo lugar, como a probabilidade de acesso ao meio é igual a todos os nós, um nó que possui uma velocidade de transmissão mais baixa penaliza os outros nós que estão concorrendo pelo mesmo meio. Para evitar isso, o trabalho limita a extensão geográfica para garantir a mesma taxa de acesso para todos os nós, pois, conforme a distância em relação ao ponto de acesso, menor é a velocidade de transmissão.

Em terceiro lugar, como o método de acesso CSMA/CA distribui de igual maneira o acesso ao meio aos nós, o trabalho, para diferenciar o desempenho das fontes de tráfego nos nós móveis e garantir QoS, limita as fontes para que as que possuem baixa prioridade sejam diferenciadas das fontes de alta prioridade.

A metodologia para a garantia de QoS *diffserv* foi escolhida na proposta por apresentar algumas vantagens em relação à metodologia *intserv* (GARCÍA-MACÍAS, 2001): em uma rede sem fio, não é possível definir limites rígidos – por exemplo, não é

possível garantir o atraso em um meio sem fio, mesmo que recursos suficientes sejam reservados através do RSVP. Além disso, há o *overhead* quando é usado um protocolo de sinalização e atrasos de configuração no momento em que um nó faz um *handover*.

A Figura 5.6 mostra os elementos da arquitetura. Cada célula sem fio (área de cobertura de um ponto de acesso) é gerenciada por um roteador de acesso (*Access Router – AR*) que encaminha pacotes entre os nós móveis em uma célula e a conecta a um roteador de borda (*Edge Router* – ER) por uma rede guiada. Todos os nós móveis e os ARs possuem mecanismos *diffserv* para que as fontes de tráfego sejam controladas de acordo com as condições variáveis da célula.

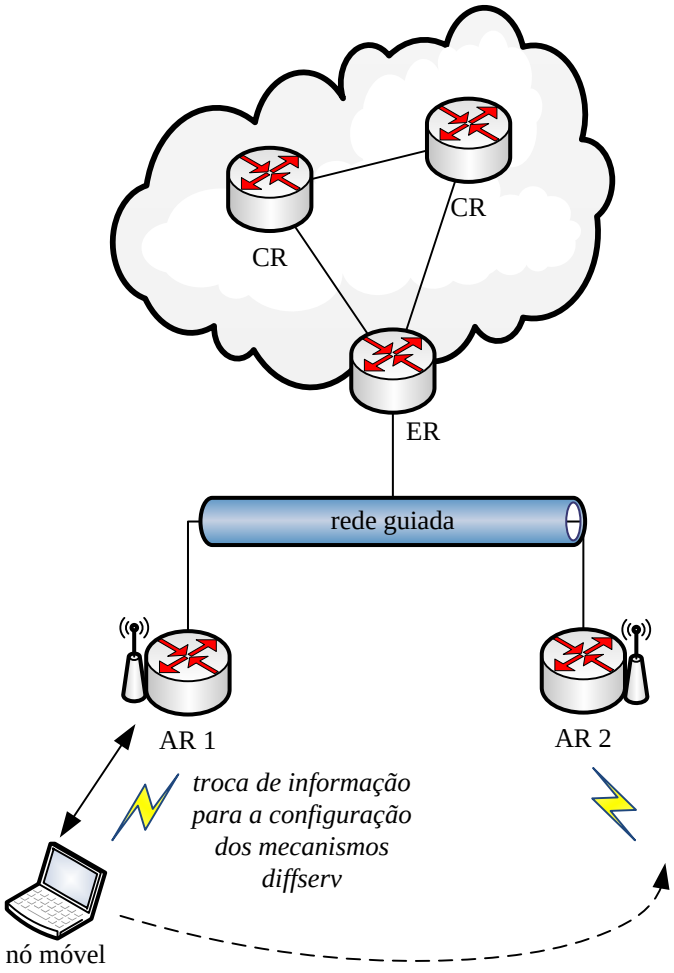

Figura 5.6: Cenário para mensagens de sinalização

A arquitetura é hierárquica porque há dois níveis de gerenciamento: intra-célula e inter-célula. O primeiro nível de gerenciamento é local para uma célula e é realizado pelo AR, que gerencia mudanças rápidas ocorridas na célula. Os nós móveis informam ao AR a largura de banda requerida e o AR configura seus mecanismos de QoS. Para obter o comportamento desejado, os ARs devem ser informados, além da largura de

banda, o número de nós na célula, pois a largura de banda disponível depende da quantidade de nós ativos na célula e do tráfego agregado de cada classe.

O nível de gerenciamento inter-célula refere-se a um conjunto de células sem fio conectadas a um ER através de uma rede guiada. Neste nível, as condições mudam de maneira mais lenta que no meio sem fio. Este gerenciamento global é realizado pelo ER, que determinada políticas de longa duração aos ARs. O ER atua com um gerenciador global de QoS para os ARs. Ele define políticas a serem seguidas pelos ARs, como controle de admissão e reserva de recursos (QOS_POLICY).

Para a alocação de banda, o nó móvel envia uma mensagem ao AR (QOS_REQUEST) e o AR interpreta essa mensagem e a satisfaz com a configuração apropriada dos mecanismos *diffserv* (QOS_CONFIG). O módulo de QoS no nó móvel configura a taxa de saída das classes EF e AF/BF e fixa a proporção entre as classes AF e BE.

A Figura 5.7 apresenta a troca de mensagens proposta no trabalho para a configuração dos mecanismos *diffserv* nos ARs e nos nós móveis.

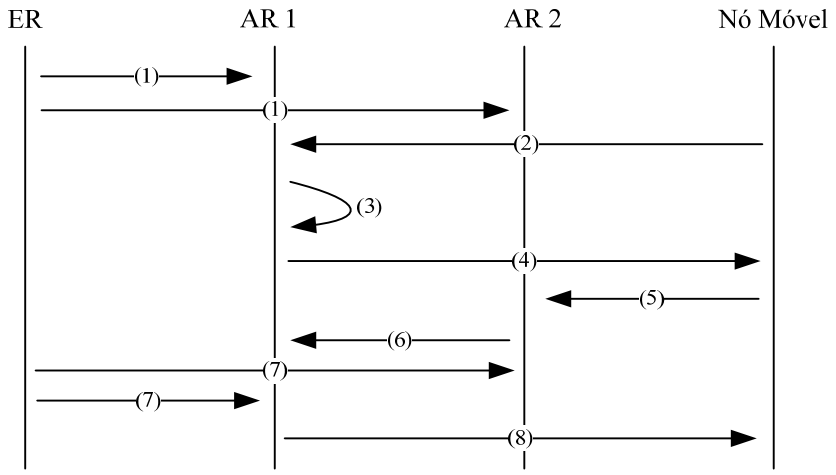

Figura 5.7: Troca de mensagens em García-Macías (2001)

(1) O ER define as políticas a serem seguidas pelos ARs que estão na mesma rede através da mensagem QOS_POLICY.

(2) Através da mensagem QOS_REQUEST, o nó móvel solicita a reserva de recursos para o AR ao qual está conectado.

(3) Para atender aos requisitos de QoS solicitados pelo nó móvel, o AR configura seus mecanismos *diffserv*.

(4) Juntamente com sua auto-configuração, o AR informa como devem ser configurados os mecanismos *diffserv* do nó móvel, para que as fontes de dados sejam controladas conforme a prioridade de cada.
(5) A decisão do nó para mover-se a outra célula pode ser baseada em parâmetros como a relação sinal/ruído ou pode levar em consideração parâmetros de QoS. Uma vez tomada a decisão, o nó móvel envia uma mensagem HO_REQ para o AR de destino (AR2) através do AR de origem (AR1), contendo o endereço do AR de destino e a solicitação para a reserva de recursos.
(6) Se é possível realizar a reserva e recursos e o handover é aceito, o AR de destino atualiza sua tabela de roteamento com a rota para o nó móvel e envia uma mensagem HO_ACK através do ER para o AR de origem.
(7) O ER informa os ARs que estão na rede guiada a nova localização do nó móvel.
(8) Quando receber a mensagem HO_ACK, o nó móvel define o AR de destino (AR2) como seu novo roteador *default* e muda para o canal utilizado na nova célula.

Há dois modos para sinalizar os requisitos de QoS entre as entidades: usar os próprios pacotes de dados para a comunicação (sinalização *in-band*), inserindo informações em um campo do cabeçalho do protocolo IP ou utilizando extensões do cabeçalho, ou gerar pacotes de controle ICMP (sinalização *out-band*). Para transmitir as mensagens de sinalização de QoS no trabalho, é proposto um novo tipo de ICMPv6, pois podem ocorrer situações onde não há dados a serem transmitidos e sinalizações sejam necessárias.

O modelo de gerenciamento de mobilidade utilizado na proposta é similar ao apresentado em no projeto HAWAII (RAMJEE, 1999), onde há uma comunicação prévia com o AR de destino para verificar se há recursos suficientes para os requisitos de QoS do nó móvel sejam atendidos, e somente em caso positivo há efetivamente o handover para a nova célula.

A Tabela 5.4 apresenta um resumo da proposta.

Tabela 5.4: Resumo da proposta (García-Macías (2001)

Arquitetura de QoS	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
Serviços diferenciados	Micro-mobilidade	Similar ao do projeto HAWAII	IPv6 (na prática IPv4)	Sim. Através de mensagens ICMPv6

5.4 HMRSVP: A Hierarchical Mobile RSVP Protocol

Em Tseng (2003), é proposta uma integração do protocolo RSVP com o protocolo IPv4 Móvel com registro regional (FOGELSTROEM, 2007). As reservas de recursos são realizadas antecipadamente somente quando há a iminência de uma movimentação inter-domínio do nó.

O protocolo RSVP, utilizado na metodologia *intserv* para a garantia de QoS, não pode ser usado, na forma que foi concebido, diretamente em ambientes de computação móvel por dois motivos: (*i*) as mensagens RSVP são invisíveis as roteadores intermediários de um túnel criado pelo protocolo IP Móvel porque um túnel IP é implementado através de um esquema de encapsulamento IP-IP (PERKINS, 1996a). As mensagens RSVP *PATH* e *RESV* são encapsuladas pelo protocolo IP, que possui um número de identificação diferente do protocolo RSVP (4 para o protocolo IP e 46 para o protocolo RSVP), impossibilitando que os roteadores que estão no caminho do túnel IP reconheçam adequadamente as mensagens RSVP; (*ii*) depois que um nó móvel muda de localização, os recursos previamente alocados não estão mais disponíveis, e a qualidade de serviço oferecida ao nó pode sofrer uma degradação significativa devido à falta de reserva de recursos na nova localização do nó.

Para resolver o problema de tunelamento, é aproveitado o trabalho em Terzis (2000), onde a idéia básica é aplicar recursivamente o protocolo RSVP no túnel presente no caminho das mensagens *PATH* e *RESV*. Nesta nova sessão, o ponto de entrada do túnel (onde há o encapsulamento) envia uma mensagem PATH e o ponto de saída do túnel (onde há o desencapsulamento) envia uma mensagem RESV, desse modo ocorrendo a reserva de recursos também no túnel para as comunicações que passem por ele através da troca extra de mensagens *PATH* e *RESV* (Figura 5.8). Para realizar a reserva de recursos de maneira antecipada, é utilizado o protocolo Mobile RSPV (TALUKDAR, 1999). Originalmente, este protocolo realiza antecipadamente a reserva de recursos em múltiplos locais onde o nó móvel possa estar durante o período de serviço.

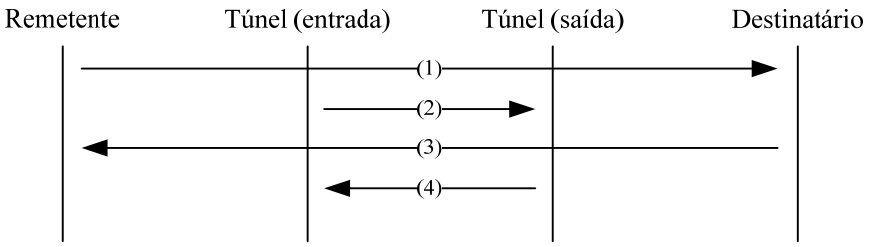

Figura 5.8: Reserva de recursos com o protocolo RSVP em túneis IP

(1) Inicialmente, o remetente envia uma mensagem *PATH* para o destinatário do fluxo, a qual contém os endereços do remetente e do destinatário como origem e destino no cabeçalho do pacote IP e o também o número 46 referente ao protocolo RSVP. A mensagem, ao passar pelo túnel, é encapsulada e desencapsulada normalmente com o encapsulamento IP-IP.

(2) A entrada do túnel, após enviar a mensagem *PATH* encapsulada, envia uma nova mensagem *PATH*, mas dessa vez com os endereços da entrada e da saída do túnel como origem e destino no cabeçalho IP.

(3) Em resposta à mensagem *PATH*, o destinatário envia uma mensagem *RESV* ao remetente. Da mesma maneira, quando a mensagem chega ao túnel, ela é encapsulada na entrada e desencapsulada na saída do túnel.

(4) O ponto de saída do túnel envia uma mensagem *RESV* para a entrada do túnel, ocorrendo dessa forma também a reserva de recursos no caminho onde há o tunelamento de pacotes.

Utilizando-se das soluções anteriores, juntamente com o conceito de hierarquia do IPv4 Móvel com Registro Regional, o trabalho propõe um novo protocolo baseado no protocolo Mobile RSVP (MRSVP) chamado Hierarchical Mobile RSVP (HMRSVP). Diferentemente de Talukdar (1999), a reserva antecipada de recursos para um nó móvel somente é realizada quando o nó está em uma área de cobertura onde atua mais de uma célula e onde o atraso devido ao *handover* tende a ser longo.

No protocolo IPv4 Móvel, toda vez que um nó móvel conecta-se a uma rede diferente, ele deve registrar-se com seu HA. Nos casos em que o HA está localizado longe do ponto de conexão do nó móvel, o processo de registro pode ser excessivamente longo. Com o protocolo IPv4 Móvel com Registro Regional, o processo de registro fica restrito dentro de uma região quando a movimentação do nó é intra-domínio (micro-

mobilidade). Por região entende-se um conjunto de roteadores e redes dispostos hierarquicamente, a qual pode ser compreendida por uma rede empresarial ou de campus. Devido à natureza hierárquica e às propriedades de roteamento da Internet, os FAs podem realizar o processo de registro com certo grau de independência em relação ao HA, e os registros intra-domínios podem ficar restritos a uma região somente.

A Figura 5.9 ilustra o funcionamento do HMRSVP quando o nó móvel não está localizado em uma fronteira de região. A linha contínua representa o caminho onde a reserva de recursos está ativa, ou seja, o caminho pelo qual os pacotes percorrem e há a garantia de QoS obtida através da troca de mensagens RSVP. Como a figura demonstra, o nó móvel está em uma célula que não faz fronteira com uma célula de outra região, e presume-se que o nó móvel somente realizará *handovers* intra-domínio. Desse modo, o HMRSVP somente mantém a reserva ativa, sem realizar reserva antecipada de recursos.

O *gateway* é responsável por realizar o tunelamento do protocolo IP Móvel com Registro Regional para todos os nós conectados na área de cobertura de um *proxy*. O *gateway*, juntamente com os *proxies* que estão a ele conectados, formam uma topologia hierárquica.

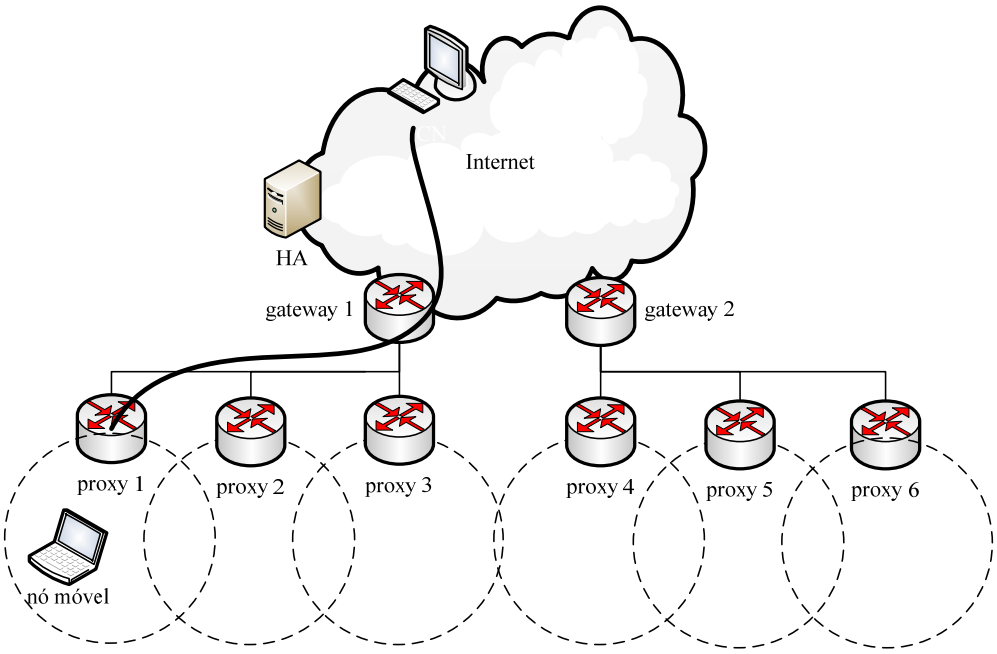

Figura 5.9: Cenário do HMRSVP sem reserva antecipada de recursos

Na Figura 5.10, o nó móvel situa-se em uma área onde há a presença de duas regiões distintas. Neste cenário, o HMRSVP estabelece uma reserva de recursos passiva (linha pontilhada), pois o nó móvel, nesse caso, pode realizar um *handover* inter-

domínio. Ao contrário do MRSVP, o HMRSVP somente realiza a reserva passiva antecipada de recursos na iminência de um *handover* inter-domínio, e não em todas as células adjacentes.

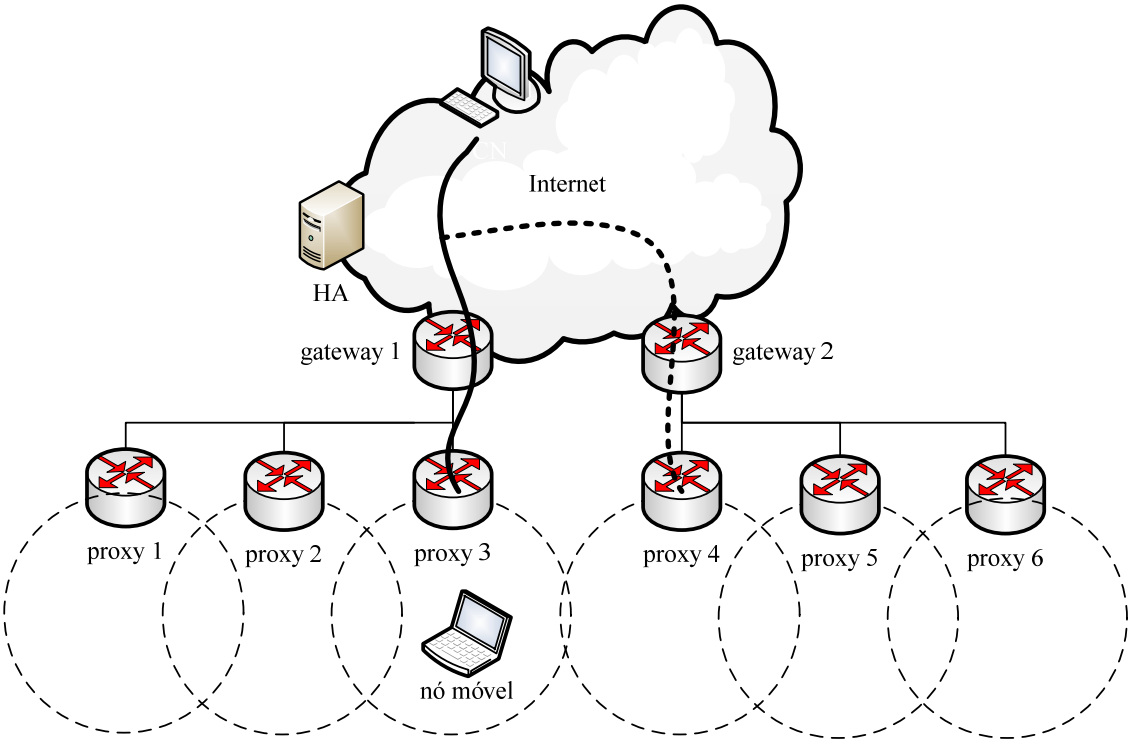

Figura 5.10: Cenário do HMRSVP com reserva antecipada de recursos

A Figura 5.11 traz a seqüência de troca de mensagens entre as entidades para uma movimentação intra-domínio. O cenário para a troca de mensagens é o mesmo da Figura 5.9.

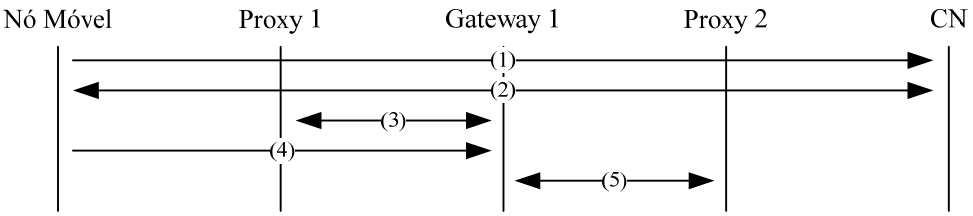

Figura 5.11: Troca de mensagens para movimentação intra-domínio em Tseng (2003)

(1) Inicialmente, o nó móvel envia uma mensagem *receiver_mpsec{gateway 1}* para informar ao CN que ele está em um rede estrangeira conectada ao *gateway 1*.

(2) O CN e o nó móvel trocam um par de mensagens *PATH* e *RESV* para estabelecer a reserva RSVP fim-a-fim entre o CN e o nó móvel.

(3) Adicionalmente às mensagens de reserva de recursos fim-a-fim, um túnel RSVP é estabelecido entre o *gateway 1* e o *proxy 1*. Este túnel, juntamente com o túnel criado em (2), constituem uma reserva RSVP ativa.

(4) Quando o nó móvel desloca-se para a área de cobertura do *proxy 2*, uma mensagem de registro é enviada somente para o *gateway 1*, e não até o HA, o que ocorre no protocolo IPv4 Móvel sem o registro regional.

(5) Após receber a mensagem de registro, um novo túnel é criado entre o *gateway 1* e o *proxy 2* com a troca de mensagens *PATH* e *RESV*. Essa reserva torna-se ativa e a reserva anterior é liberada. Essa nova reserva pode ser feita rapidamente porque os *proxies 1 e 2* estão conectados ao mesmo *gateway*.

A troca de mensagens para uma movimentação inter-domínio do nó móvel é apresentada na Figura 5.12. O cenário é o presente na Figura 5.10.

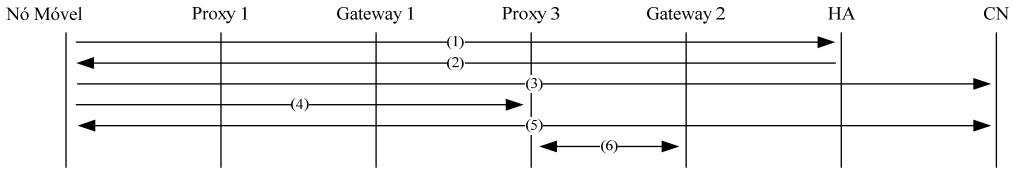

Figura 5.12: Troca de mensagens para movimentação inter-domínio em Tseng (2003)

(1) Quando o nó móvel movimenta-se para uma área onde há a atuação de *proxies* conectados a *gateways* distintos, ele realizará um registro múltiplo simultâneo para obter um novo CoA do *proxy* 3. Essa mensagem de registro será enviada ao *gateway 2*, que repassará a mensagem para o HA do nó móvel.

(2) O HA confirma o registro para o nó móvel com uma mensagem *registration reply* e o nó obtém um novo CoA do *gateway 2*.

(3) O nó móvel envia uma mensagem *receiver_mspec{gateway 1, gateway 2}* para o CN informando que está em uma área coberta pelos dois *gateways*.

(4) Juntamente com a mensagem (3), o nó móvel comunica ao *proxy 3*, através de uma mensagem *receiver_spec*, os parâmetros originais de QoS. Com essa mensagem, o CN pode inicializar a reserva passiva de recursos no caminho entre ele e o *proxy 3*, passando pelo *gateway 2*.

(5) O nó móvel e o CN trocam mensagens *PATH* e *RESV* para estabelecer a reserva passiva de recursos fim-a-fim.

(6) Uma reserva passiva também é estabelecida entre o *gateway 2* e o *proxy 3* através da troca de mensagens *PATH* e *RESV*. Se o nó móvel sair da cobertura

do *proxy 2*. A reserva de recursos passiva ficará ativa, enquanto a reserva de recursos ativa anterior será liberada.

A Tabela 5.5 apresenta um resumo da proposta.

Tabela 5.5: Resumo da proposta em Tseng (2003)

Arquitetura de QoS	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
Serviços integrados	Micro-mobilidade	IP Móvel com Registro Regional	IPv4	Sim. Protocolo RSVP

5.5 An Efficient RSVP-Mobile IP Interworking Scheme

Em Paskalis (2003) é proposta uma abordagem para integrar mobilidade com QoS de modo a melhorar a eficiência no uso dos recursos de rede e a reduzir a deterioração de QoS que um nó sofre durante sua movimentação. Para isso, modificações de rede relacionadas à mobilidade e ao QoS são limitadas dentro de um domínio (no caso da proposta, o domínio consiste na rede de acesso).

A integração do protocolo Mobile IP como o protocolo RSVP, sem que haja um melhor refinamento nos protocolos, é lento, ineficiente e consome muita largura de banda, de acordo com Paskalis (2003). Alguns dos maiores problemas são: *i)* a demora em restabelecer a reserva de recursos, pois as mensagens RSVP devem percorrer o caminho fim-a-fim duas vezes para restabelecer uma sessão, resultando em uma degradação de QoS nos fluxos ativos; *ii)* reserva duplicada de recursos durante um período de tempo relevante, pois, após o *handover*, os recursos reservados quando o nó móvel estava na antiga localização não são liberados automaticamente, e a reserva duplicada dura enquanto não houver uma liberação explícita ou enquanto a reserva de recursos não expirar; *iii)* probabilidade maior que as novas solicitações de recursos não sejam atingidas, pois as reservas de recursos duplicadas, em ambientes de alta mobilidade ou com grande quantidade de nós móveis, podem afetar a eficiência da rede; *iv)* custo maior para fornecer serviços com QoS, pois a reserva duplicada de recursos tende a um nível de utilização efetivo médio menor pelo mesmo custo se esses recursos estivessem sendo usados de maneira mais eficiente.

Para minimizar os restabelecimentos de fluxos RSVP e assim minimizar as modificações dentro da rede de acesso enquanto um nó está em movimento, é proposto que o nó móvel possua dois diferentes CoAs: um *Local Care-of Address* (LCoA), para a movimentação dentro do domínio de acesso, e um CoA global, chamado de *Domain*

Care-of Address (DCoA), através do qual o nó é sempre acessível, seja por tunelamento, tradução de endereços, ou qualquer tipo de roteamento dentro de uma abordagem de gerenciamento de mobilidade hierárquico, como Soliman (2005) e Fogelstroem (2007).

O DCoA é utilizado na proposta como identificador único e permanente para todos os fluxos RSVP, assim como as propostas em Thomas (2002) e Shen (2001), para que a integração entre o RSVP e o Mobile IP se dê de maneira eficiente. A manutenção do mesmo DCoA deve ser realizada somente quando há conexões ativas utilizando esse identificador, podendo ser liberado para a utilização por outro nó assim que não hajam mais conexões.

Para controlar a tradução dinâmica entre LCoAs e DCoAs, é introduzido o *RSVP Mobility Proxy* (RSVP-MP), um roteador na borda da rede de acesso para gerenciar as mensagens RSVP, notificar as mudanças dos LCoAs e notificar os dispositivos que gerenciam a mobilidade de rede (*foreign agents*, por exemplo). A Figura 5.13 apresenta a topologia de rede com o RSVP-MP.

As reservas RSVP são realizadas com o uso do DCoA, que é único para cada nó móvel. A tradução de endereços somente é realizada no cabeçalho do pacote na borda da rede através do encapsulamento e desencapsulamento. As mensagens RSVP, as quais possuem os endereços de origem e destino dos nós, são alteradas para conter os LCoAs ou os DCoAs, dependendo de qual é a direção das mensagens (em direção à rede de acesso ou no sentido inverso, respectivamente). Nas reservas que são realizadas fora da rede de acesso, é utilizado o DCoA, enquanto nas reservas realizadas dentro da rede de acesso é utilizado o LCoA. Isso permite que os eventos relacionados à mobilidade fiquem restritos à rede de acesso e não seja necessário propagar a mudança de topologia a outras redes. A Figura 5.14 apresenta a troca de mensagens para a reserva de recursos na proposta.

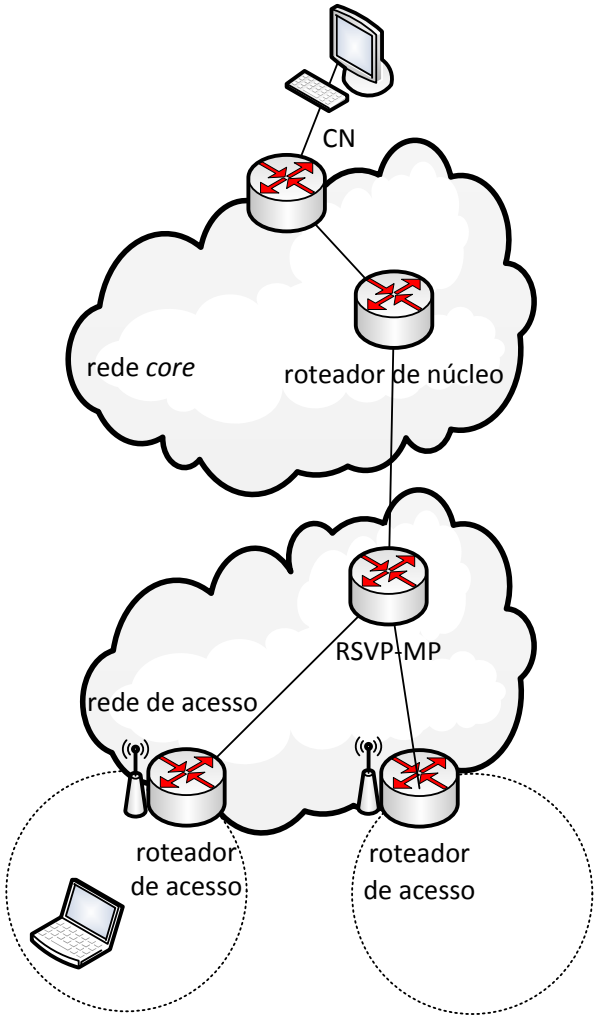

Figura 5.13: Cenário de Paskalis (2003)

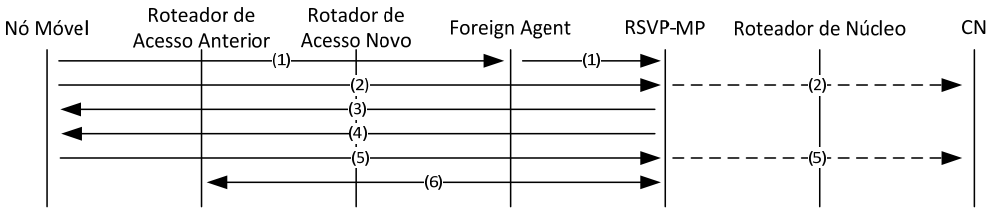

Figura 5.14: Troca de mensagens em Paskalis (2003)

(1) No caso de um *handover*, o nó móvel obtém um novo LCoA. Para restabelecer a reserva de recursos para a comunicação entre ele e o CN, os recursos devem ser reservados em ambas as direções. Ao enviar uma mensagem *binding update*, o nó móvel informa seu novo LCoA, e o dispositivo que gerencia a mobilidade dos nós comunica o evento ao RSVP-MP.

(2) Com a introdução do RSVP-MP, as mensagens *PATH* e *RESV* devem ser intermediadas por ele para que haja a conversão do LCoA para o DCoA, ou vice-versa, conforme o caso. Primeiramente, com seu LCoA, o nó móvel envia

uma mensagem *PATH* para o CN para restabelecer a reserva de recursos nesse sentido. Essa mensagem é interceptada pelo RSVP-MP, o qual realiza a tradução do LCoA para o DCoA (tanto no cabeçalho quanto no conteúdo do pacote) e a envia para o CN.

(3) O RSVP-MP responde à mensagem PATH do nó móvel sem aguardar uma resposta do CN.

(4) Simultaneamente a (2), o RSVP-MP verifica a reserva de recursos no sentido do nó móvel com o uso do DCoA. Se houver uma comunicação ativa, o RSVP-MP envia uma mensagem *PATH* contendo o endereço IP do CN, como se ele tivesse sido enviada por ele.

(5) O nó móvel responde a mensagem (4) com uma mensagem *RESV* para o CN, a qual é interceptada pelo RSVP-MP e o LCoA é traduzido para o DCoA. A sinalização RSVP é restrita ao domínio da rede de acesso, enquanto que as mensagens *PATH* e *RESV* transmitidas para fora do domínio servem apenas como mensagens de atualização do estado.

(6) Paralelamente à nova reserva de recursos, os recursos referentes ao antigo LCoA devem ser liberados. O RSVP-MP envia uma mensagem *PathTear/ResvTear* para o antigo roteador de acesso, que também envia uma mensagem *PathTear/ResvTear*. Desse modo, os recursos em ambos os sentidos são liberados.

O atraso para o restabelecimento da reserva de recursos depende do caminho a ser percorrido pelas mensagens RSVP. O atraso na reserva é diretamente proporcional ao atraso fim-a-fim e à distância entre os nós que estão em comunicação. No caso do RSVP-MP, este atraso depende somente do atraso na rede de acesso.

Assim como a melhora no atraso, o RSVP-MP elimina as reservas de recursos duplicadas na rede de acesso, permitindo maior eficiência na utilização dos recursos de rede e um maior número de atendimento de novas solicitações, sem que haja falta de recursos disponíveis.

Através de simulações, foi verificado que a probabilidade de bloqueio é menor no caso do RSVP-MP quando comparado ao protocolo RSVP original, variando-se a taxa de chegada das requisições de reservas de recursos e a velocidade de locomoção dos nós. Da mesma forma, a largura de banda não é utilizada de maneira desnecessária

pelo RSVP-MP, enquanto no protocolo RSVP há uma ocupação média de 20% da largura de banda com sessões antigas mantidas ativas.

Observa-se também uma redução na quantidade de mensagens de sinalização fora da rede de acesso, tanto na rede *core* quanto no dispositivo final, pois estas são mantidas, no restabelecimento da reserva de recursos, somente dentro da rede de acesso.

Contudo, com a introdução do RSVP-MP, há um aumento de complexidade do roteador de borda (roteador de acesso), único dispositivo afetado pela proposta. Os outros componentes de rede, como roteadores de núcleo, outros roteadores na rede de acesso e os nós móveis não são afetados pela proposta, devendo somente ser compatíveis com o protocolo RSVP.

A Tabela 5.6 apresenta um resumo da proposta.

Tabela 5.6: Resumo da proposta em Paskalis (2003)

Arquitetura de QoS	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
Serviços integrados	Micro-mobilidade	IP Móvel com Registro Regional ou IPv6 Móvel hierárquico	IPv4 ou IPv6	Sim. Protocolo RSVP

5.6 Supporting Mobility Events within a Hierarchical Mobile IP-over-MPLS Network

Para realizar a integração da arquitetura MPLS com o gerenciamento de mobilidade, mais especificamente micro-mobilidade, é proposta em Vassiliou (2007) uma arquitetura chamada *Overlay* MMPLS (Mobile MPLS), a qual cria uma rede MPLS com o suporte à mobilidade utilizando o protocolo HMIPv6, sem que haja alterações nos protocolos já existentes.

A arquitetura proposta prevê dois tipos de utilização: *i)* prover opções de mobilidade a um domínio MPLS já estabelecido; ou *ii)* como uma rede de acesso para a extensão de uma infra-estrutura maior, na qual o MPLS é usado para o encaminhamento dos pacotes. A topologia básica considerada no trabalho é uma rede de acesso via rádio (*Radio Access Network* – RAN), ilustrada na Figura 5.15.

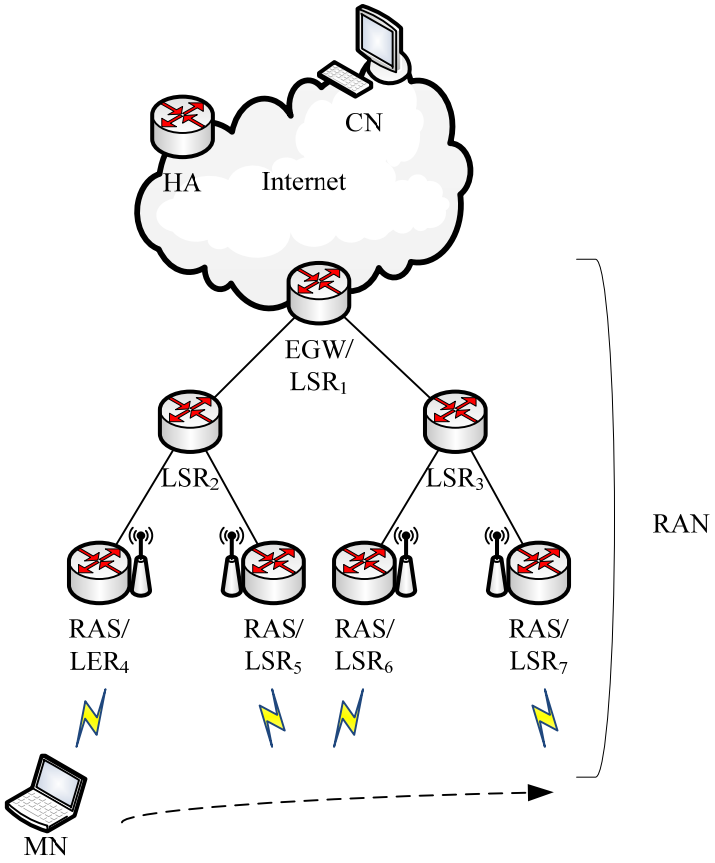

Figura 5.15: Cenário de Vassiliou (2007)

No cenário da proposta, a RAN consiste em dois ou mais níveis hierárquicos de LSRs. Somente através de uma hierarquia de roteadores é que o HMIPv6 pode trazer vantagens na sinalização da mobilidade do usuário. Os primeiros dispositivos MPLS vistos pelos nós móveis são os roteadores de acesso via rádio (*Radio Access Routers* – RASs). Os RASs são conectados a um ou mais *Edge Gateways* (EGW), os quais fornecem acesso a outras redes. Os roteadores entre os EGWs e os RASs possuem as funcionalidades MPLS.

Os agentes responsáveis pelo controle da mobilidade dos nós operam com os LSRs. A integração dos dois ocorre somente no uso das tabelas de roteamento pelos LSRs atualizadas através do HMIPv6. O registro do HMIPv6 e a configuração dos LSPs são independentes e não há troca de informações entre as bases de dados.

A proposta assume que os LSPs somente são criados a partir do momento em que algum dado necessita ser transmitido entre os nós. Desse modo, há duas formas de operação: a primeira, na qual o nó móvel inicia a comunicação, e a segunda, na qual o CN é responsável por iniciar a comunicação. A Figura 5.16 apresenta a troca de mensagens para a reserva de recursos na proposta no cenário em que o CN inicia a

transmissão. O modo de operação no qual o nó móvel inicia a comunicação exige um menor número de troca de mensagens e está descrito em Vassiliou (2007).

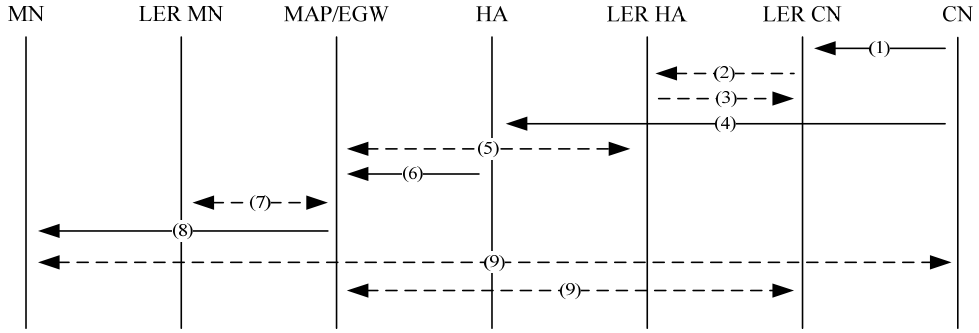

Figura 5.16: Troca de mensagens em Vassiliou (2007)

(1) A criação de um LSP somente ocorre quando o CN necessita enviar algum dado ao MN.

(2) Quando o CN inicia a comunicação com um nó móvel, inicialmente ele analisa seu registro para verificar se já não possui o CoA do nó móvel. Em caso negativo, uma mensagem *label request* é enviada para o endereço de origem do MN.

(3) O LER do HA responde com o mapeamento do rótulo e o LSP criado termina nele. Neste ponto, o LER do CN possui um rótulo de saída para marcar os pacotes destinados ao nó móvel e o LER do HA possui um rótulo de entrada para receber os pacotes destinados ao MN. A configuração de um LSP é realizada toda vez que é há uma nova requisição de comunicação.

(4) Quando um pacote chega ao HA verifica se o MN está em sua rede de origem. Caso o nó esteja em uma rede estrangeira, no HA consta a informação do RCoA do nó móvel, o qual nada mais é que o endereço do MAP.

(5) Após a troca de mensagens, um LSP é criado entre o LER do HA e o MAP ao qual o nó móvel está conectado.

(6) A primeira vez que um pacote rotulado chega ao MAP, o rótulo é retirado e é verificado na tabela interna do MAP qual é o LCoA associado ao RCoA recebido no pacote.

(7) Um novo LSP é criado entre o MAP e o LER que corresponde ao LCoA verificado no passo anterior, o qual indica a posição do MN.

(8) Após ser criado o LSP, o pacote é finalmente entregue ao nó móvel.

(9) Faz parte do protocolo HMIPv6 notificar diretamente ao CN qual a localização do nó móvel. Deste modo, uma mensagem *binding update* é enviado ao CN, o qual responde com uma mensagem *binding acknowlegment*.

(10) Um novo LSP é criado entre o LER do CN e o MAP, para que a comunicação entre o nó móvel e o CN não precise primeiramente ser intermediada pelo HA. Uma vez que o trabalho proposto assume que cada pedido possui seu próprio LSP, o MAP e o LER do MN possuem entradas tanto para a comunicação entre o CN e o MN quanto para a comunicação entre o HA e o MN.

A abordagem utilizada para os *handovers* na proposta utiliza a modificação juntamente com a extensão dos LSPs. No primeiro caso, um novo LSP é criado refletindo a nova localização do nó móvel. No segundo caso, o LSP é estendido do LER anterior para o novo LER. Com a combinação dessas duas abordagens, não há perda de pacotes. A troca de mensagens para a mobilidade de um nó intra-RAN é mostrada na Figura 5.17.

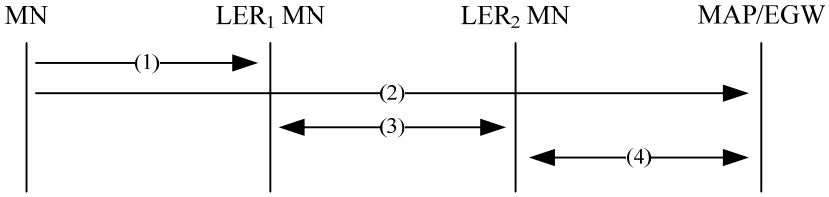

Figura 5.17: Troca de mensagens em um *handover* intra-RAN

(1) Quando o nó móvel realizar a mudança de rede do LER_1 para o LER_2, ele obtém um novo LCoA. De posse desse endereço, ele envia uma mensagem *binding update* ao LER_1 com seu novo LCoA. Dessa forma, os pacotes em trânsito em direção ao LER_1 podem ser redirecionados ao LER_2.

(2) Simultaneamente a (1), o nó móvel envia uma mensagem *binding update* local para seu MAP e para os CNs que porventura estejam na mesma RAN.

(3) Após o *handover*, o LER_1 inicia a configuração de um LSP entre ele e o novo LER do nó móvel.

(4) Um LSP deve ser criado entre o MAP e o novo LER do nó móvel, para que os novos pacotes recebidos pelo MAP sejam encaminhados para posição atualizada do MN.

A mobilidade inter-RAN, com a alteração do MAP ao qual o nó móvel está conectado, inclui todos os passos da mobilidade intra-RAN, com o acréscimo que o HA e os CNs que estão fora da RAN terão que estabelecer LSPs com o novo MAP do MN. Este procedimento ocorre quando os CNs e o HA recebem uma mensagem *binding update* do nó móvel informando seu novo RCoA.

Por fim, através de engenharia de tráfego, é possível com o MPLS prover qualidade de serviço através da reserva de recursos para determinados LSPs. Uma vez que a proposta considera a criação de um novo LSP a cada comunicação entre dispositivos diferentes, cada fluxo de dados criado entre eles pode receber uma prioridade diferente, embora a proposta não aponte isso por não possuir o foco em qualidade de serviço.

A Tabela 5.7 apresenta um resumo da proposta.

Tabela 5.7: Resumo da proposta em Vassiliou (2007)

Arquitetura de QoS	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
MPLS	Micro-mobilidade	IPv6 Móvel Hierárquico	IPv6	Sim. RSVP-TE, por exemplo.

5.7 Análise comparativa

Cabe neste ponto analisar as semelhanças e diferenças dos trabalhos apresentados neste capítulo com a proposta apresentada no capítulo seguinte. Dessa forma, é possível verificar quais são as contribuições da arquitetura proposta e de que forma as soluções apresentadas se diferenciam dos trabalhos atualmente publicados. A Tabela 5.8 apresenta o resumo das propostas apresentadas anteriormente.

A utilização de uma ou outra arquitetura de qualidade de serviço possui suas vantagens e desvantagens. De um lado, é possível com a metodologia de serviços integrados reservar os recursos de rede de forma mais precisa, individualmente para cada fluxo. Por outro lado, tal abordagem não é tão escalável quanto a metodologia de serviços diferenciados, na qual a diferenciação de tráfego é realizada por classes de serviço, e não por fluxos individuais.

Tabela 5.8: Análise comparativa dos trabalhos correlatos

Proposta	**Arquitetura de QoS**	**Cenário de mobilidade**	**Arquitetura de gerenciamento de mobilidade**	**Versão do protocolo IP**	**Protocolo de sinalização**
1	*Diffserv*	Macro-mobilidade	IP Móvel	IPv4	Sim. Comunicação via TCP
2	*Diffserv*	Macro-mobilidade	NEMO	IPv6	Sim. Juntamente com o binding update da arquitetura de gerenciamento de mobilidade
3	*Diffserv*	Micro-mobilidade	Similar ao do projeto HAWAII	IPv6 (na prática IPv4)	Sim. Através de mensagens ICMPv6
4	*Intserv*	Micro-mobilidade	IP Móvel com Registro Regional	IPv4	Sim. Protocolo RSVP
5	*Intserv*	Micro-mobilidade	IP Móvel com Registro Regional ou IPv6 Móvel Hierárquico	IPv4 ou IPv6	Sim. Protocolo RSVP
6	MPLS	Micro-mobilidade	IPv6 Móvel Hierárquico	IPv6	Sim. RSVP-TE, por exemplo.

Em um ambiente móvel, a metodologia *intserv* requer que a reserva individual por fluxo deve ser restabelecida toda vez que um nó altera sua localização. Novas reservas de recursos ao longo do novo caminho percorrido pelo fluxo de dados devem ser realizadas através do protocolo de sinalização, qual seja, o protocolo RSVP. Além disso, os recursos de rede reservados no antigo caminho percorrido pelo fluxo de dados devem ser liberados. A solução encontrada pela proposta (4) é realizar uma reserva antecipada de recursos através do protocolo de sinalização quando o nó estiver em uma área de cobertura onde atua mais de uma célula, de modo a evitar que, após um *handover* intra-domínio, haja a degradação do serviço oferecido. A solução encontrada por (5), ao estabelecer os CoAs para o nó móvel, necessita de uma nova entidade à arquitetura de mobilidade para controlar e realizar a conversão entre LCoA e DCoA e vice-versa, permitindo delimitar quais são os dispositivos de rede afetados e que necessitam de uma nova configuração disparada pelo protocolo RSVP.

Na metodologia *diffserv*, a reserva de recursos e a garantia de qualidade de serviço são realizadas através de classes de serviço, as quais independem, de certo modo, da rede de acesso ao qual o nó ligado. Uma vez definido o valor do campo DS e o condicionamento dos pacotes nas bordas de um domínio *diffserv*, cabe aos nós de núcleo selecionar o comportamento no encaminhamento dos pacotes através do

codepoint neles presentes, mapeando esse valor para um dos PHBs suportados pelo domínio.

Ademais, como apontado em García-Macías (2001), não se pode esperar uma previsibilidade no atraso quando se leva em consideração uma rede sem fio, a qual está intimamente relacionada com a mobilidade do usuário, mesmo que recursos suficientes sejam reservados através do protocolo RSVP. A natureza da metodologia *diffserv*, embora não seja tão precisa em termos de reserva de recursos, não define limites rígidos de qualidade de serviço, permitindo trabalhar com um máximo e um mínimo estipulados dentro de um contrato de nível de serviço (SLA).

No que se refere à reserva de recursos, a arquitetura MPLS assemelha-se à arquitetura *intserv* pois, através de engenharia de tráfego e através da configuração dos LSPs, é possível estabelecer caminhos entre os roteadores e estabelecer prioridades paras os fluxos que utilizam o LSP priorizado. Contudo, da mesma forma que na metodologia *intserv*, ocorre na arquitetura MPLS a necessidade de reconfiguração e nova reserva de recursos caso haja mudança no fluxo de dados, como se observa, não de maneira explícita, na proposta (6).

Diante destes pontos levantados, dada a escabilidade da metodologia *diffserv*, esta é a utilizada na arquitetura proposta. Os trabalhos propostos em (1), (2) e (3), os quais se baseiam na metodologia *diffserv*, utilizam alguma forma de sinalização por parte do nó móvel para a comunicação dos requisitos de QoS exigidos, seja através de mensagens ICMP ou outro modo de sinalização. Um problema que surge na sinalização a partir do nó móvel é a questão de segurança envolvida, pois essa sinalização deve ser autenticada e verificada com as permissões do nó móvel. Por outro lado, a arquitetura proposta neste trabalho utiliza outra forma para a comunicação dos requisitos de QoS exigidos pelo nó móvel. Com a utilização da arquitetura PBNM, os roteadores de borda são dinamicamente configurados quando um nó móvel altera seu ponto de acesso à rede, sem que haja um *overhead* do protocolo de sinalização na rede de acesso. Como as políticas são previamente acordadas entre o usuário e o administrador do domínio *diffserv*, sobrepõe-se o problema de autenticação do usuário. Detalhes de como esta sinalização é realizada estão presentes no Capítulo 6. Neste ponto, a arquitetura proposta neste trabalho assemelha-se à proposta (6).

Embora a sinalização dos requisitos de QoS como presente nas propostas (1) e (5) possa trazer vantagens em relação ao tempo necessário para a reserva de recursos após o *handover* de um nó móvel, tal sinalização importa em alteração dos protocolos já

existentes, tanto quando realizada junto com os pacotes de dados (*in-band*), quanto realizada à parte do tráfego (*out-band*).

A arquitetura proposta neste trabalho usa os mecanismos já existentes no que se refere ao controle da mobilidade e ao gerenciamento de qualidade de serviço. Além disso, não são introduzidas novas entidades diferentes daquelas já presentes nas arquiteturas utilizadas, mas somente uma interação entre as já existentes, o que significaria, caso contrário, em um aumento de mensagens de sinalização entre novas entidades e um aumento do *overhead* para prover QoS em redes móveis.

A escolha da versão do protocolo IP (IPv4 ou IPv6) e do cenário de mobilidade, no qual se prevê somente a macro- ou também a micro-mobilidade dependem de qual é arquitetura de gerenciamento de mobilidade utilizada. Embora o protocolo IPv6 traga melhoramentos em relação ao espaço de endereçamento, configuração automática de endereços de maneira mais simples que o protocolo IPv4 e melhoramento na autenticação e segurança (DEERING, 1998), ele não possui ainda a ubiqüidade do protocolo IPv4. Desse modo, atualmente, as arquiteturas de gerenciamento de mobilidade que levam em consideração o protocolo IPv4 possuem uma aplicação mais imediata na prática.

Conforme o tipo de cenário apresentado, pode se optar pela utilização dos protocolos de macro- e micro-mobilidade. Em determinados ambientes, a macro-mobilidade gera uma quantidade significativa de tráfego de sinalização no núcleo da rede, mesmo que a movimentação do nó seja local. Em segundo lugar, há um atraso considerável na difusão da atualização da localização do nó móvel e, em terceiro lugar, os *handovers* são longos, ocasionando interrupções e perdas de pacotes (MANNER, 2002).

Atualizações de localização são sempre geradas quando o nó móvel muda de localização. Usuários com alta mobilidade podem notificar com freqüência o HA, resultando em um alto *overhead* de controle. Em situações onde há alta quantidade de nós móveis, a quantidade de sinalização pode tornar-se parte significativa do tráfego.

Para contornar o problema da sinalização e da demora nos *handovers*, as mudanças de rede do nó móvel e a sua sinalização são regionalizadas através de protocolos que suportam a micro-mobilidade. O uso de protocolos de micro-mobilidade minimiza o atraso entre o *handover* e a instalação de informações de roteamento e de QoS. O tempo necessário para restabelecer rotas influencia no tempo necessário para

reconfigurar os recursos necessários para prover qualidade de serviço na nova localização do nó móvel.

Dessa forma, as propostas (3), (4), (5) e (6), que prevêem o uso de protocolos de micro-mobilidade para a movimentação dentro de um domínio, apresentam uma vantagem em relação às propostas (1) e (2), pois reduzem o atraso e a perda de pacotes durante o *handover*, eliminando o registro entre o nó móvel e seu HA que pode estar em uma rede distante, regionalizando a troca de mensagens de controle.

Contudo, somente é possível observar as vantagens dos protocolos de micro-mobilidade se há uma hierarquia de agentes móveis (FAs e HAs, por exemplo) na rede. É através da hierarquia que a sinalização dos protocolos de mobilidade fica restrita a um número menor de dispositivos de rede e não necessita trafegar um longo caminho até que haja a autenticação do nó móvel. Em ambientes onde não há a hierarquia de agentes móveis, os resultados esperados dos protocolos de micro- e de macro-mobilidade, no que se refere à sinalização desses protocolos, são os mesmos.

Em vista do cenário proposto na arquitetura proposta do Capítulo 6, não se faz necessário o uso de um protocolo de micro-mobilidade. Em cenários com um maior número de agentes móveis dispostos hierarquicamente, contudo, o uso dos protocolos de micro-mobilidade seria mais vantajoso.

Por fim, cabe ressaltar que, com a utilização da arquitetura MPLS na proposta (6), não é mais necessário a utilização das técnicas de tunelamento apresentadas conjuntamente com os protocolos de gerenciamento de mobilidade. A troca de rótulos nos dispositivos intermediários permite que o problema de roteamento quando da presença de usuários móveis seja contornado assim como a utilização do tunelamento IP-IP nas demais propostas apresentadas neste capítulo.

Capítulo 6

Arquitetura Proposta

Assim como os trabalhos correlatos apresentados no capítulo anterior, pretende-se com este trabalho realizar a integração do gerenciamento de mobilidade com o gerenciamento de qualidade de serviço. Na arquitetura proposta, os roteadores e seus mecanismos para garantir a qualidade de serviço são configurados através do gerenciamento de redes baseado em políticas (PBNM). É através do PBNM que são configurados os mecanismos *diffserv* dos roteadores presentes do domínio administrativo (domínio gerenciado).

As arquiteturas de mobilidade e de qualidade de serviço funcionam de forma adequada quando tratadas em separado, mas, para que haja sua integração, algumas questões são levantadas e necessitam de solução, o que faz com que essa integração não seja trivial.

Para uma melhor explanação da arquitetura proposta, os pontos de integração são primeiramente tratados separadamente, para que depois convirjam em uma arquitetura única e coesa. Partindo-se do pressuposto da atual existência de uma arquitetura de integração de gerenciamento de redes baseado em políticas com gerenciamento de qualidade de serviço (BELLER, 2005), há dois os pontos base a serem resolvidos: (a) integração de gerenciamento de mobilidade com gerenciamento de rede; (b) integração de mobilidade com qualidade de serviço. Ambos os pontos importam em alterações na arquitetura PBNM presente em Beller (2005).

6.1 Integração de Mobilidade com Gerenciamento de Rede

O primeiro ponto a ser tratado neste trabalho é a integração do protocolo de gerenciamento de mobilidade com o protocolo de gerenciamento de rede; no caso deste trabalho, a integração do protocolo IPv4 Móvel com a arquitetura PBNM.

Os protocolos de gerenciamento de dispositivos de rede, como, por exemplo, o protocolo SNMP (*Simple Network Management Procotol*) e o protocolo COPS-PR (*Common Open Policy Service for Policy Provisioning*), possuem, em suas definições, o suporte a eventos gerados pelos dispositivos gerenciados através destes protocolos. Estes eventos são gerados quando uma determinada configuração ou estado do dispositivo é alterado, e esta alteração necessita de algum modo ser informada à entidade de gerenciamento. No protocolo SNMP, os eventos são notificados através de uma mensagem *SNMP Trap*, enquanto no protocolo COPS-PR os eventos são notificados através de uma mensagem *Report State* (RPT).

Determinados eventos, por outro lado, podem ser gerados por entidades externas àquelas presentes em um ambiente de gerenciamento (Figura 6.1). Entende-se por entidades presentes em um ambiente de gerenciamento, neste trabalho, as entidades de gerenciamento e os objetos por ela gerenciados.

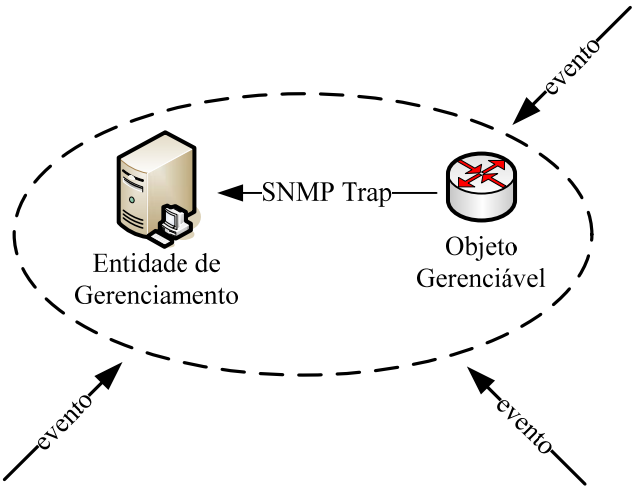

Figura 6.1: Eventos externos ao ambiente de gerenciamento

Neste trabalho, os eventos gerados por entidades externas ao ambiente de gerenciamento são chamados de *eventos de reavaliação*. O *evento de reavaliação* pode ser utilizado, por exemplo, para a comunicação do deslocamento do usuário em um

ambiente IP Móvel ou para obtenção de informações de outras entidades presentes na rede que não o objeto gerenciável. Em um ambiente de mobilidade, este evento externo toma o nome, neste trabalho, de *evento de mobilidade*.

No caso do *evento de mobilidade* ser utilizado para a notificação da mobilidade de um dispositivo, o seguinte cenário se apresenta: no momento em que o HA responde à mensagem *registration request* do nó móvel com uma mensagem *registration reply*, o HA notifica essa mudança a uma entidade responsável pelo gerenciamento de rede (neste caso, o PDP) através de um *evento de mobilidade*. Tal evento importa na reavaliação das políticas de gerenciamento (realizada pelo PDP) e a aplicação de novas políticas nos dispositivos gerenciados (PEPs). A Figura 6.2 sintetiza o mecanismo de funcionamento de um *evento de mobilidade* na integração da arquitetura de mobilidade com a arquitetura PBNM.

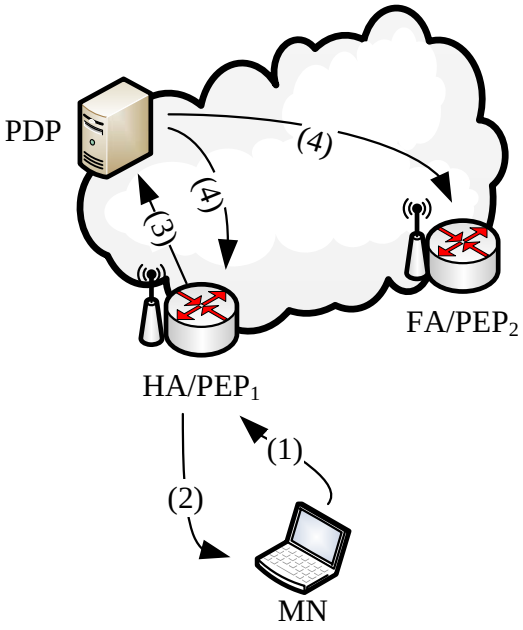

Figura 6.2: Integração IP Móvel e PBNM

(1) Através de uma mensagem *registration request*, o nó móvel informa a seu HA sua nova localização, a qual pode ser uma rede estrangeira ou a sua própria rede de origem.

(2) O HA informa ao nó móvel que o registro foi recebido e as informações pertinentes ao nó móvel (associação entre o endereço *home* do nó móvel e o CoA) foram alteradas. Desse modo, os pacotes enviados ao nó móvel são corretamente encaminhados à sua atual localização.

(3) Concomitantemente a (2), o HA informa ao servidor de políticas (PDP) a alteração da localização do nó móvel através de um *evento de mobilidade*.
(4) Dada a nova posição do nó móvel, o PDP aplica as novas políticas nos PEPs, seja acrescentando informações relativas ao nó móvel, seja removendo informações relativas ao nó.

Conforme a Figura 6.2, deve haver uma forma de comunicação entre a entidade externa ao ambiente de gerenciamento (neste caso, o HA), com o servidor de políticas (PDP), ou seja, de alguma forma o *evento de mobilidade* deve ser notificado à entidade responsável por reaplicar as políticas nos objetos gerenciáveis (neste caso, os PEPs).

Advinda a necessidade de comunicação de eventos externos ao ambiente de gerenciamento, a arquitetura proposta neste trabalho utiliza a comunicação via *web service* para integrar o gerenciamento de mobilidade com o gerenciamento de rede. Este modelo permite a disponibilização de serviços distribuídos que utilizem interfaces de acesso simples e bem definidas. Como a comunicação se dá de forma padronizada, é possível a independência de plataforma e de linguagem de programação. Os detalhes de como é realizada a comunicação via *web services* estão presentes na seção 7.3.

6.2 Integração de Mobilidade com Qualidade de Serviço

A arquitetura PBNM permite automatizar os processos de geração e distribuição de configuração para os dispositivos em um ambiente *diffserv*. Para integrar o gerenciamento de mobilidade com qualidade de serviço, o trabalho de Beller (2005) precisa sofrer suportar os *eventos de mobilidade* e assim prover a possibilidade de mudança de localização de um nó móvel, para que este continue recebendo a qualidade de serviço contratada e especificada em um SLA. Além disso, a as antigas configurações referentes ao nó móvel precisam ser removidas da sua anterior localização, isto é, removidas do roteador de borda ao qual o nó móvel estava conectado.

A Figura 6.3 apresenta uma visão geral dos processos da arquitetura proposta na integração de mobilidade com qualidade de serviço, dentro da arquitetura PBNM.

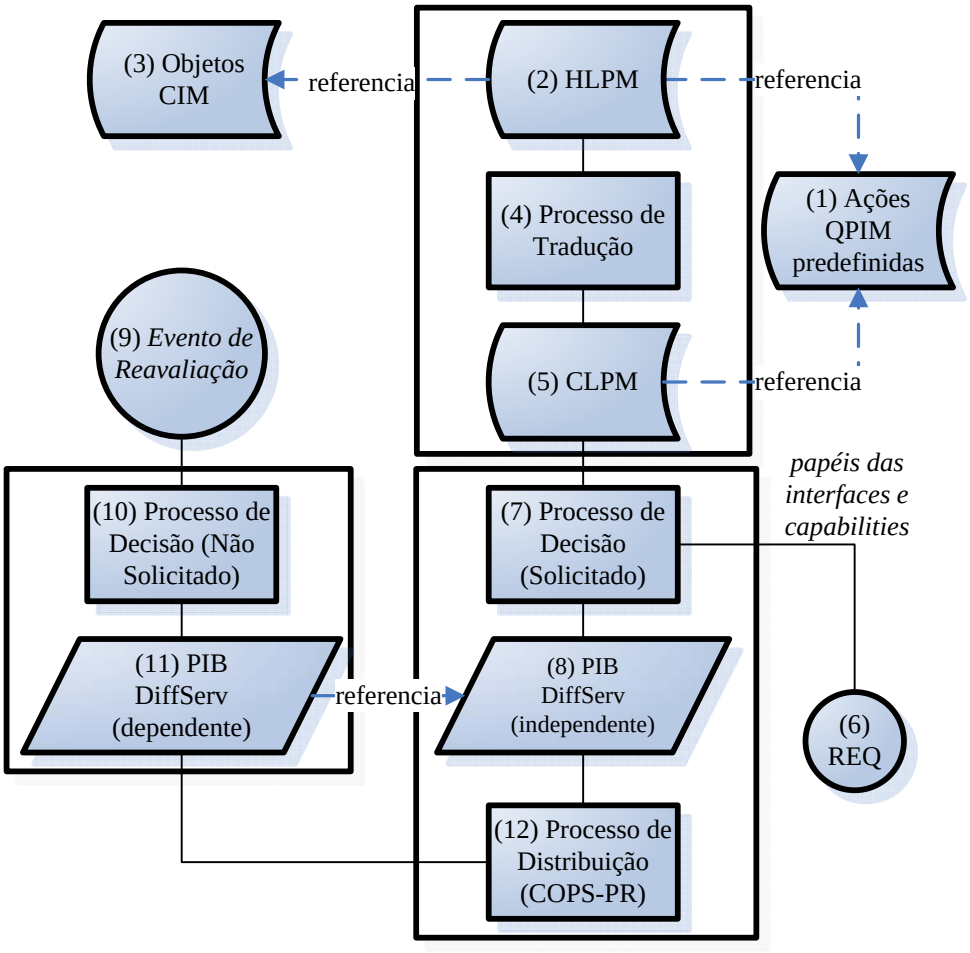

Figura 6.3: Processos da integração de mobilidade com qualidade de serviço

De acordo com a estratégia definida pela arquitetura, o administrador define uma biblioteca de ações QPIM (1) correspondentes aos SLSs (*Service Level Specifications*) que serão atribuídos aos usuários. Um SLS é definido em (Westerinen, 2001) como o controle do tráfego do usuário pelo provedor de serviço. Em um ambiente de serviços diferenciados, por exemplo, o SLS define *codepoints* e o perfil e o tratamento do tráfego para esses *codepoints* (isto é, o PHB para esses *codepoints*).

No modelo de política de alto nível (HLPM) (2), o administrador define os objetivos do negócio atribuindo SLSs (isto é, ações QPIM) aos clientes do ambiente gerenciado. O HLPM estende o modelo PCIM/PCIMe pelo provedor de serviço e suporta a semântica: *"usuário(s) acessando (uma) aplicação(ões) em (um) servidor(es) remoto, a partir de (uma) rede(s) de acesso recebe um nível de serviço específico"*. Usuários, aplicações e elementos de rede em uma política de alto nível são expressos em termos de objetos CIM (*Common Information Model*) (3). Os objetos CIM podem ser compartilhados entre diferentes arquiteturas de gerenciamento.

Dentro da arquitetura proposta, é neste ponto que são definidos os níveis de serviço recebidos pelos usuários móveis. A vinculação entre o serviço oferecido e o usuário móvel é realizada através do endereço de origem (*home address*) do nó móvel.

O processo de tradução (4) converte a informação de alto nível em políticas de configuração independentes de dispositivo (isto é, a configuração traduz os efeitos de QoS desejado sem especificar detalhes do mecanismo, como o tipo de escalonador ou o algoritmo de descarte). Por exemplo, *"o tráfego com ip_origem=200.10.10.5 e porta=21 recebe largura_de_banda=25%"*. O modelo de política de nível de configuração (CLPM) (5) é definido como uma combinação de classes PCIM/PCIMe e QPIM com o objetivo de suportar a representação de ambos os elementos em uma configuração de dispositivo: identificação de tráfego (condições) e tratamento de tráfego (ações). Condições são descritas em termos de filtros de pacotes baseados no cabeçalho IP e ações são descritas em termos de mecanismos de QoS, como escalonadores e algoritmos de descarte. O CLPM também inclui um mapeamento entre as políticas de configuração e os papéis das interfaces dos dispositivos. Esse mapeamento é deduzido da informação de topologia extraída do HLPM.

No processo de decisão (7), as políticas de configuração são transformadas em instâncias PIB *Diffserv* (8). Esse processo é executado quando o PDP recebe uma mensagem COPS-PR de requisição (REQ) (6) do PEP perguntando por uma configuração de provisão. A mensagem REQ possui dois conjuntos de informações que são usadas como parâmetros de entrada em um processo de decisão: (*i*) *RoleCombination*, que são rótulos associados às interfaces dos dispositivos gerenciados; (*ii*) *DeviceCapabilities*, que descrevem mecanismos de QoS específicos suportados pelo dispositivo gerenciado. Primeiramente, o PDP usa o *RoleCombination* para selecionar as políticas relevantes para a interface do dispositivo gerenciado e, após, o PDP converte as políticas de configuração em instâncias de provisionamento da PIB *Diffserv*, de acordo com o conjunto de *DeviceCapabilities*. A informação da PIB é gerada da configuração QPIM por um processo de transformação que leva em consideração as *capabilities* (mecanismos de QoS suportados) do dispositivo gerenciado.

Para suportar os *eventos de mobilidade* e, conseqüentemente, para suportar a mobilidade (9), o provisionamento inicial da PIB (8) gerado pela mensagem REQ (6) contém somente as definições independentes desse tipo de evento (no caso em particular, os eventos relacionados à mobilidade). As definições dependentes do *evento*

de mobilidade (11) são geradas em resposta a esses eventos e transportadas para o PEP através de mensagens de decisão não solicitadas (10).

Finalmente, o processo de distribuição (12) consiste em transmitir as PRIDs *da* PIB *DiffServ* utilizando o protocolo COPS-PR.

6.3 Cenário e Entidades da Proposta

Uma vez definida como se dará a integração do gerenciamento de mobilidade realizado pelo protocolo IPv4 Móvel com a metodologia de qualidade de serviço *diffserv*, cabe mostrar uma visão geral das entidades da proposta e como elas interagem para prover aos usuários móveis a qualidade de serviço previamente contratada. Inicialmente, a Figura 6.4 apresenta o cenário previsto para a implementação e validação da arquitetura.

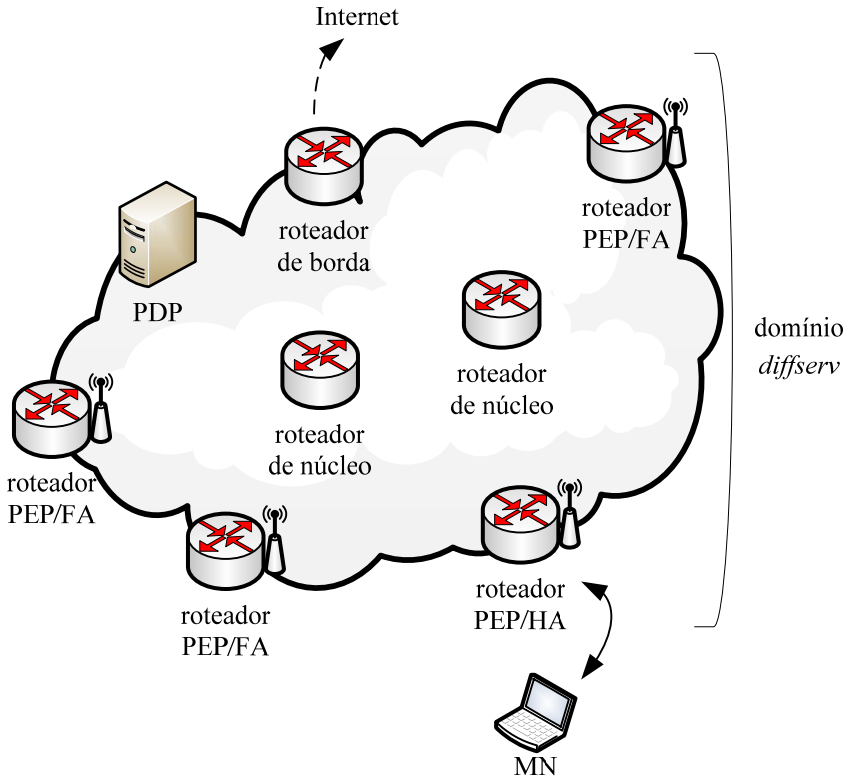

Figura 6.4: Cenário da arquitetura proposta

Para garantir a qualidade de serviço aos usuários móveis, o cenário da proposta consiste em um domínio (nuvem) *diffserv*, composto por roteadores de borda e roteadores de núcleo. Esse domínio coincide com o domínio administrativo no qual são aplicadas as mesmas políticas de provisão a todos os dispositivos de rede que o

compõem. Um dos nós de borda, por exemplo, pode consistir no acesso externo à Internet e/ou na integração a outros domínios *diffserv* sob outra administração.

A administração do domínio *diffserv* é responsável pela garantia de que os recursos adequados serão disponibilizados e reservados para suportar e cumprir com os SLAs oferecidos pelo domínio aos seus clientes. Em outras palavras, é neste ponto que é definida a biblioteca de ações QPIM correspondentes aos SLSs que serão atribuídos aos usuários. O SLS nada mais especifica que o tratamento dado ao tráfego do usuário na rede negociado previamente com os clientes do serviço oferecido pelo provedor.

Os roteadores de borda do domínio *diffserv* acumulam também as funções de PEP da arquitetura PBNM. É através desta arquitetura que as políticas são aplicadas aos roteadores de borda para a correta manipulação dos pacotes dos usuários móveis e, conseqüentemente, prover a qualidade de serviço. Como os PEPs necessitam ser configurados com as políticas adequadas, o domínio *diffserv* possui ao menos um servidor de políticas (PDP) para atendê-los.

Os roteadores de borda funcionam também como roteadores de acesso à rede e ao domínio *diffserv*. Para que haja o suporte à mobilidade dos usuários, esses roteadores, dentro da arquitetura de mobilidade, devem possuir as funcionalidades de um FA/HA.

No cenário proposto, a residência de um usuário móvel, por exemplo, pode consistir em sua rede de origem. Deste modo, o roteador com o qual o usuário realiza o acesso à rede de sua residência possui a função de um HA. Como observado na Figura 6.4, o usuário móvel está localizado em sua rede de origem conectado ao seu HA, e todos os demais roteadores de acesso comportam-se como FAs do seu ponto de vista.

A mudança de localização do usuário implica na notificação do PDP de tal mudança, na aplicação de novas políticas no roteador de borda ao qual o nó móvel conectou-se e na remoção das políticas no roteador a qual o nó estava conectado. Desse modo, há uma configuração dinâmica dos dispositivos, sem que seja necessário *a priori* que todos os roteadores de borda tenham o conhecimento de todos os usuários móveis que possam conectar-se a eles. Ademais, os recursos são utilizados de maneira mais eficiente, pois as configurações *diffserv* são aplicadas nos roteadores de borda somente em caso de necessidade.

Capítulo 7

Implementação da Proposta

A arquitetura proposta no Capítulo 6 independe de linguagem de programação. Contudo, para realizar a implementação do cenário proposto, algumas decisões de projeto devem ser tomadas. Este capítulo apresenta como é realizada a representação e o armazenamento das políticas utilizadas pelo PDP e aplicadas nos PEPs e a implementação dessas entidades. Ambas as questões são adaptadas a partir do trabalho de Beller (2005).

Este capítulo também mostra como é realizada a notificação de um *evento de mobilidade* e qual é a estratégia utilizada para atualizar a configuração roteadores de borda, os quais acumulam, de acordo com a arquitetura proposta, a função de FA da arquitetura IPv4 Móvel. É a partir de um *evento de mobilidade* que novas políticas de configuração *diffserv* são enviadas aos roteadores de borda.

Ademais, outra questão que surge a partir do cenário proposto para provisão de qualidade de serviço é o tratamento dos pacotes quando presentes em um túnel (abordagem utilizada para encapsular os pacotes destinados a um nó móvel fora de sua rede de origem no protocolo IPv4 Móvel). Uma abordagem deve ser definida para o tratamento do campo DS do cabeçalho dos pacotes IP quando do ingresso e do egresso dos pontos finais de um túnel IP-IP.

7.1 Representação e armazenamento de políticas

Com base em Beller (2005), as políticas são armazenadas em documentos XML e estes servem como repositório das informações utilizadas pelo PDP e pelo PEP, para definição e aplicação das políticas, respectivamente. Optou-se pela representação em XML pois a assimilação das políticas ocorre de maneira mais intuitiva durante sua

leitura. Ainda, segundo Beller (2005), esta estratégia facilita o entendimento dos complexos modelos (PCIM/PCIMe, QPIM, PIB *Framework*, PIB *DiffServ*) usados para representação das políticas nos diferentes níveis de abstração existentes dentro da arquitetura.

Para o mapeamento das informações em XML, foram consideradas as diretrizes do IETF e DMTF para armazenamento dos objetos CIM em LDAP, mais especificamente o mapeamento do modelo PCIM (STRASSNER, 2004) e PCIMe (PANA, 2005). O reaproveitamento de políticas é possível através da utilização de referências formadas por expressões *XML Pointer Language* (*XPointer*) (WORLD WIDE WEB CONSORTIUM, 2007).

A representação das políticas de alto nível (HLPM) segue a estrutura definida na Figura 7.1.

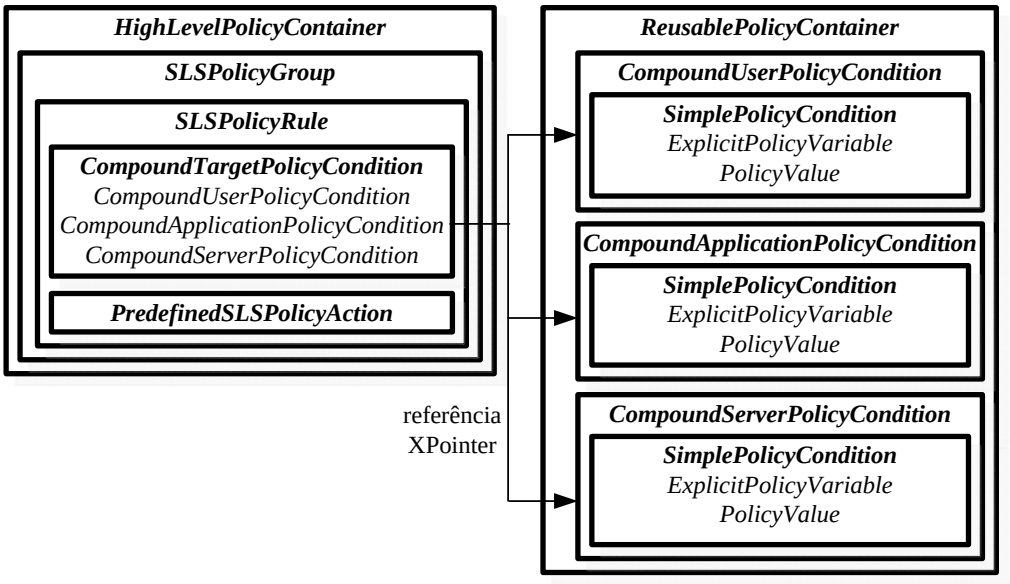

Figura 7.1: Representação do modelo HPLM

O primeiro contêiner na hierarquia é o *HighLevelPolicyContainer*, no qual são representados os grupos SLS chamados *SLSPolicyGroup*. Os grupos SLS são compostos por uma ou mais regras SLS chamadas *SLSPolicyRule*.

Em cada uma das regras SLS, as condições *CompoundTargetPolicyCondition* são definidas pela semântica *"usuários* acessando *aplicações* em *servidores"*. As condições de usuários *CompoundUserPolicyCondition*, aplicações *CompoundApplicationPolicyCondition* e servidores *CompoundServerPolicyCondition*

constituem contêineres reutilizáveis *ReusablePolicyContainer* em diferentes regras SLS. A reutilização dos contêineres é possível com o uso de referências XPointer.

As ações *PredefinedSLSPolicyAction* possuem um atributo chamado *PredefinedSLSName* que define o SLS selecionado para a regra.

A representação das políticas de configuração (CLPM) segue a estrutura definida na Figura 7.2.

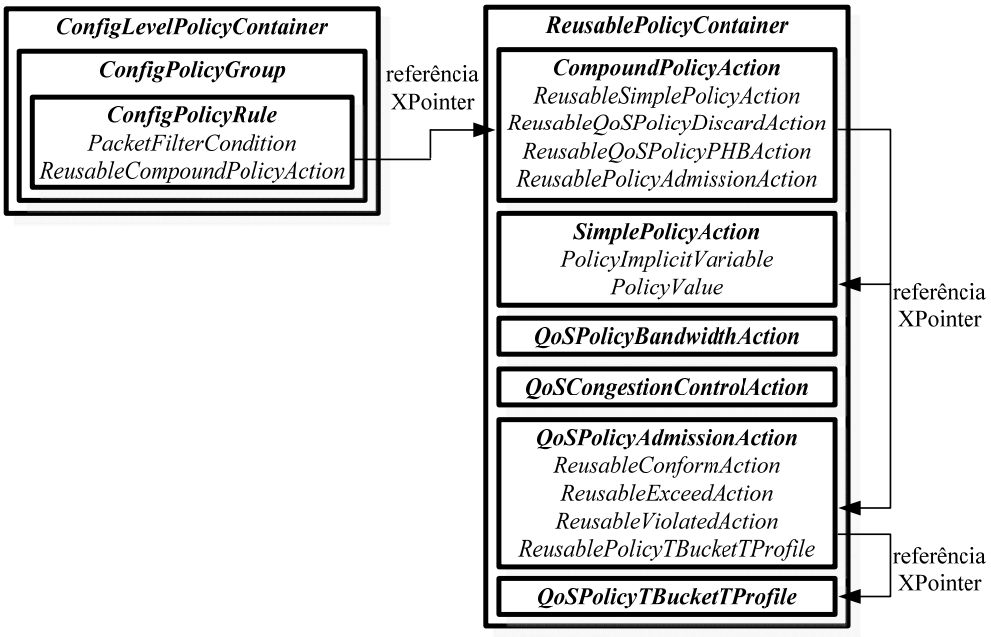

Figura 7.2: Representação do modelo CLPM

O primeiro contêiner definido para o CLPM é chamado *ConfigLevelPolicyContainer*, no qual estão presentes os grupos *ConfigPolicyGroup*, os quais representam diferentes grupos de dispositivos em um domínio administrativo *diffserv*. Os grupos são compostos por regras *ConfigPolicyRule* correspondentes à configuração dos dispositivos que desempenham o mesmo papel no domínio gerenciado.

Enquanto a condição *PacketFilterCondition* não é reutilizável, as composições de ações *CompoundPolicyAction* são reutilizáveis e referenciadas por ponteiros XPointer.

Todas as ações QPIM *SimplePolicyAction*, *QoSPolicyBandwidthAction*, *QoSCongestionControlAction* e *QoSPolicyAdmissionAction* são armazenadas no mesmo contêiner reutilizável *ReusablePolicyContainer* e descrevem os parâmetros dos SLS pré-definidos. As referências XPointer podem referenciar o próprio contêiner. Por

exemplo, *CompoundPolicyAction* reutiliza as ações definidas por *PolicyConformAction*, *PolicyExceedAction* e *PolicyViolateAction*, as quais são executas de acordo com o resultado obtido na comparação do também reutilizável *QoSPolicyTBucketTProfile*.

A tradução das políticas de alto nível (HLPM) para as políticas de configuração é descrita em detalhes em (Beller, 2005).

7.2 PDP e PEPs

A implementação do servidor de políticas (PDP) e os clientes (PEPs) foi aproveitada do trabalho presente em Beller (2005), o qual suporta a estratégia de provisão de políticas, mas modificada para o suporte aos *eventos de mobilidade*. De acordo com a estratégia de provisão de políticas, as requisições e as decisões ocorrem de forma assíncrona. A comunicação entre o PDP e o PEP é realizada através do protocolo COPS-PR. As configurações de ambas as entidades são representadas na linguagem XML e armazenadas em arquivos de configuração (Figura 7.3).

```
<Pdp>
        <Config>
                <PdpId>PdpId1</PdpId>
                <PdpPort>3288</PdpPort>
                <NextPdpAddr>127.0.0.1</NextPdpAddr>
                <NextPdpPort>3288</NextPdpPort>
                <Ka>10</Ka>
                <SupportedPep>2 4001 4002</SupportedPep>
                <AuthorizedPep> PepA PepB PepC PepD
                PepE PepF PepG PepH PepI PepJ PepK PepL
                PepM PepN PepO PepP PepQ PepR PepS PepT
                PepU PepV PepX PepZ PepY PepW
                </AuthorizedPep>
        </Config>
        <ActivePep>
        </ActivePep>
</Pdp>
<Pep>
        <Config>
                <PepId>PepB</PepId>
                <PdpAddr>192.168.0.5</PdpAddr>
                <PdpPorta>3288</PdpPorta>
                <ClientType>2</ClientType>
                <MaxHops>3</MaxHops>
        </Config>
</Pep>
```

Figura 7.3: Configurações do PDP e do PEP descritas em XML

As informações utilizadas para a ativação e configuração do PDP são as seguintes:

- <*PepId*>: corresponde à identificação do servidor de políticas;
- <*PdpPort*>: define a porta TCP pela qual o servidor de políticas recebe as solicitações dos clientes. De acordo com Durham (2000), a porta definida é a de número 3288 ;
- <*NextPdpAddr*>: define o endereço do próximo servidor de políticas caso o servidor atual não tenha a capacidade de prover as políticas necessárias. Deste modo, a requisição é encaminhada ao servidor apontado;
- <*NextPdpPort*>: define a porta TCP do próximo servidor de políticas, a qual segue o padrão definido em Durham (2000);
- <*Ka*>*:* define o intervalo de tempo para a geração de uma mensagem de controle, a qual objetiva informar o estado de funcionamento do servidor de políticas;
- <*SupportedPep*>: apresenta uma lista com todos os tipos de clientes PEP suportados por este servidor. Clientes que não estejam nesta lista não são suportados pelo PDP e são encaminhados ao próximo servidor de políticas;
- <*AutorizedPep*>: apresenta uma lista com todas as identificações de clientes PEP que têm permissão para se conectar a este servidor de políticas. Clientes não autorizados não têm sua conexão permitida;
- <*ActivePep*>*:* armazena os estados dos PEPs conectados a este PDP, na forma de uma cópia das instâncias (PRIs) enviadas para os PEPs.

As informações utilizadas para a ativação e configuração dos PEPs são as seguintes:

- <*PdpId*>: corresponde à identificação do cliente de políticas;
- <*PdpAddr*>: endereço IP do servidor de políticas ao qual o PEP conecta-se e obtém as políticas para sua operação;
- <*PdpPort*>: define a porta TCP pela qual o servidor de políticas recebe as solicitações dos clientes;

- *<ClientType>*: define o tipo de cliente de políticas do PEP. Este atributo define as características relativas à aplicação de políticas para o tipo de cliente especificado. Para os clientes de políticas de configuração *diffserv*, o valor deste atributo é 0x0002;
- *<MaxHops>:* número máximo de PDPs ao qual o PEP pode solicitar a conexão. Este atributo é utilizado quando um PDP não pode atender à solicitação de um PEP, encaminhando-a a outro servidor de políticas.

As funções implementadas pelo PDP e pelos PEPs são descritas em detalhes em Beller (2005).

7.3 Notificação do Evento de Mobilidade

A notificação dos *eventos de mobilidade* é realizada através de *web services*. Com a utilização de *web services*, garante-se a interoperabilidade de diferentes arquiteturas em diferentes linguagens de programação. Desta forma é possível ao HA comunicar ao PDP a mobilidade de um usuário móvel sem que seja necessário o conhecimento da implementação do servidor de políticas, e este então pode aplicar novas políticas nos PEPs.

Além de permitir a padronização do XML para troca de informações, a arquitetura *web services* permite que a comunicação entre diferentes entidades ultrapasse as barreiras impostas por *firewalls* presentes nas redes, pois o transporte de dados é realizado normalmente pelo protocolo HTTP (*HyperText Transfer Protocol*) utilizando a porta 80.

Para troca de informação entre o HA e o PDP, é necessário que o HA conheça o funcionamento do serviço disponibilizado pelo PDP. A descrição do serviço é realizada através da linguagem *Web Services Description Language* (WSDL), a qual descreve as interfaces dos serviços e a forma de ligação a protocolos específicos (HTTP ou SMTP – *Simple Mail Transfer Protocol*, por exemplo), isto é, como é realizado o acesso aos serviços oferecidos.

Para suportar os eventos de mobilidade na arquitetura PBNM e descrever o serviço disponibilizado pelo PDP, inicialmente é necessário definir quais são os tipos de eventos tratados pelo HA e que necessitam ser notificados ao servidor de políticas, isto é, definir quais são os eventos de mobilidade que devem ser notificados.

Como descrito na seção 6.1, o *evento de mobilidade* é gerado após o registro do nó móvel com seu HA. Através do registro, o nó móvel pode comunicar ao HA três situações distintas: *i)* informar ao HA qual é o seu atual CoA, no caso de mudar sua localização para outra rede estrangeira ou para sua rede de origem; *ii)* renovar um registro que está prestes a expirar; e *iii)* efetuar o *desregistro*.

Ao receber uma mensagem de registro (*registration request*), o HA pode aceitar ou negar o pedido com o envio de uma mensagem *registration reply*. Caso o pedido seja aceito, o processo de registro no HA cria, modifica ou extingue a associação de três informações relativas ao nó móvel: *i)* o endereço de origem do nó; *ii)* o CoA do nó, para o qual devem ser encaminhados os pacotes destinados ao nó móvel e interceptados pelo HA; e *iii)* o tempo de vida (*lifetime*) durante o qual o registro é válido.

Opcionalmente, é possível ao nó móvel realizar registros simultâneos e *desregistrar* um CoA específico. Através do campo S presente na mensagem de registro, o nó móvel pode solicitar ao HA que mantenha as associações anteriores relacionadas ao nó móvel. Caso o campo S não esteja habilitado, as associações anteriores relacionadas ao nó móvel são apagadas e substituídas pelas informações recebidas do novo pedido de registro. O registro simultâneo pode ser útil nos casos em que uma interface sem fio do nó móvel receba mensagens de *advertisement* de mais de um FA. Com o registro simultâneo, os pacotes destinados ao nó móvel são tunelados e encaminhados para cada CoA registrado no HA, e o nó móvel receberá múltiplas cópias dos pacotes a ele destinados.

Juntamente com a avaliação inicial das situações tratadas pelo HA no momento de um registro, é preciso considerar também o sistema de sinalização empregado no protocolo IPv4 Móvel e qual é a melhor abordagem a ser utilizada na arquitetura proposta.

Os sistemas de sinalização podem tender entre dois extremos: a abordagem *hard-state* e a abordagem *soft-state*. Entre esses dois extremos, há abordagens de sinalização que empregam tanto mecanismos da abordagem *hard-state* quanto mecanismos da abordagem *soft-state*. Na abordagem *hard-state* pura, uma vez sinalizado o estado, ele é mantido a menos que haja uma remoção explícita por parte daquele que sinalizou o estado anteriormente. Devido a essa necessidade de remoção explícita, é necessário um mecanismo para controlar os estados que não são removidos caso a entidade que sinalizou o estado apresente um problema ou caso a entidade seja desligada sem que tenha notificado a remoção do estado (estados órfãos).

Uma vez que a instalação e a remoção de um estado é realizada somente uma vez, é importante à entidade que notificou o estado saber quando um estado foi instalado ou removido. Desta forma, protocolos de sinalização confiáveis são tipicamente associados com protocolos *hard-state*. Tal abordagem é utilizada em protocolos como o ST2 (*Internet Stream Protocol Version 2*) (DELGROSSI, 1995) e Q.2931B (ITU-T, 2000).

Na abordagem *soft-state* pura, a sinalização do estado é inicialmente realizada e, se não houver uma atualização periódica deste estado, ele é removido (*timeout*). De maneira geral, a entidade responsável pela atualização do estado é a mesma entidade que comunicou o estado inicialmente. Uma vez que os estados não atualizados são removidos após um período de tempo, a abordagem *soft-state* não necessita de uma mensagem de remoção explícita nem de mecanismos para controlar estados órfãos caso ocorra algum problema com a entidade sinalizou o estado. Uma vez que há uma atualização constante do estado, não é necessária uma sinalização confiável. Alguns protocolos que usam a abordagem *soft-state* são os protocolos RSVP (BRADEN, 1997) e SIP (*Session Initiation Protocol*) (ROSENBERG, 2002b).

Entre as abordagens *hard-state* e *soft-state* puras, diversos protocolos adotam elementos de uma e de outra abordagem. Por exemplo, no protocolo RSVP as mensagens PATH e RESV são transmitidas sem que haja a confirmação do seu recebimento, pois se assume que, caso ocorra uma falha na sinalização, as mensagens serão corretamente aplicadas em uma sinalização de atualização futura. Uma extensão ao protocolo RSVP proposta em Berger (2001) introduz a confirmação do recebimento de uma mensagem de sinalização através de um *acknowledgement*. Além disso, é definido no protocolo RSVP a opção de explicitamente remover a reserva de recursos. O protocolo SIP também possui uma extensão para confiabilidade às mensagens de provisão (mecanismo da abordagem *hard-state*) (ROSENBERG, 2002b).

O protocolo IPv4 Móvel não adota uma abordagem *soft-state* pura, pois, além de utilizar as mensagens de atualização para manter o estado das associações ativas no HA, suporta, embora de maneira opcional, a remoção explícita de uma associação.

A abordagem empregada pelo protocolo de mobilidade influi na decisão de qual a abordagem utilizada na arquitetura proposta. A utilização de uma abordagem *hard-state* pura possui a vantagem de não necessitar de notificações periódicas dos eventos de mobilidade, reduzindo assim o *overhead* de sinalização na rede. Por outro lado, é necessário um mecanismo especial para a remoção de estados órfãos, isto é, um

mecanismo para a remoção de inconsistências entre os estados de quem gera um evento (neste caso, o HA) e de quem recebe o evento (neste caso, o PDP).

A arquitetura proposta segue a mesma abordagem utilizada pelo protocolo IPv4 Móvel, isto é, a abordagem *soft-state* com o mecanismo da abordagem *hard-state* para a remoção explícita de um estado.

Considerados os eventos tratados pelo HA e a abordagem de sinalização utilizada, é possível definir três formatos de eventos de mobilidade na arquitetura proposta. Estes formatos precisam ser descritos na interface *web service* disponibilizada pelo servidor de políticas.

O primeiro formato necessita de quatro parâmetros: *i)* o endereço do nó móvel; *ii)* o endereço do FA ou do HA ao qual o nó móvel está se conectado; *iii)* a lista de um ou mais FAs e/ou HA dos quais a configuração *diffserv* necessita ser removida; e *iv)* o período de validade das configurações *diffserv*. Este formato engloba as situações onde há a movimentação do nó móvel entre diferentes FAs.

O segundo formato definido necessita de três parâmetros e engloba as situações de registro inicial de um nó móvel, atualização do período de validade deste registro e um pedido de associação simultânea. Os parâmetros são: *i)* o endereço do nó móvel; *ii)* o endereço do FA ou do HA; e *iii)* o período de validade das configurações *diffserv*.

Por fim, a interface *web service* do PDP precisa de um terceiro formato para suportar as situações de remoção explícita de uma configuração. São necessários dois parâmetros: *i)* o endereço do nó móvel; e *ii)* o FA ou HA do qual a configuração *diffserv* necessita ser removida.

O endereço do nó móvel serve como identificador único do dispositivo, utilizado para selecionar quais políticas serão provisionadas ao PEP, o qual acumula também as funções de FA. Para identificar de maneira inequívoca o nó móvel, é utilizado seu endereço de origem (*home address*), o qual sempre permanece o mesmo e independe da localização do nó móvel. O endereço do FA ao qual o nó está conectado é obtido através das mensagens de *advertisement* recebidas pelo nó e enviadas em uma mensagem *registration request* (campo *care-of address*, conforme o tipo de endereçamento utilizado na arquitetura proposta).

A Figura 7.4 apresenta a interface do serviço disponibilizado pelo PDP com os elementos *reportMovement*, *reportRefresh* e *reportDeregistration* utilizados nos diferentes formatos de um evento de mobilidade definidos anteriormente.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://
webservices/mes" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" name="MobilityEventService" targetNamespace="http://
webservices/mes">
  <wsdl:types>
    <xsd:schema targetNamespace="http://webservices/mes">
      <xsd:element name="reportMovement">
        <xsd:complexType>
          <xsd:sequence>
                <xsd:element name="MNAddress" type="xsd:string" />
                <xsd:element name="NewFAAddress" type="xsd:string"></xsd:element>
                <xsd:element name="OldFAAddresses" type="tns:List"></xsd:element>
                <xsd:element name="Lifetime" type="xsd:integer"></xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
     ...
      <xsd:element name="reportRefresh">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element name="MNAddress" type="xsd:string"></xsd:element>
                        <xsd:element name="FAAddress" type="xsd:string"></xsd:element>
                        <xsd:element name="Lifetime" type="xsd:integer"></xsd:element>
                </xsd:sequence>
        </xsd:complexType>
      ...
      <xsd:element name="reportDeregistration">
        <xsd:complexType>
                <xsd:sequence>
                        <xsd:element name="MNAddress" type="xsd:string"></xsd:element>
                        <xsd:element name="FAAddress" type="xsd:string"></xsd:element>
                </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      ...
```

Figura 7.4: Descrição em WSDL do serviço

7.4 Configuração dos Roteadores de Borda

A proposta adota o padrão PIB *Diffserv* para representar a configuração distribuída dos roteadores de borda (CHAN, 2003). Um módulo PIB é uma estrutura de dados nomeados descrita como uma árvore conceitual, onde os galhos representam *Provisioning Classes* (PRCs) e as folhas representam *Provisioning Instances* (PRIs). As PRCs PIB *Diffserv* modelam um *Traffic Condition Block* (TCB), o qual é formado por zero ou mais classificadores, medidores, ações, algoritmos de descarte, filas e escalonadores de pacotes.

Como mostrado Figura 7.5, na PIB *Diffserv*, os elementos funcionais (classificadores, medidores, etc.) e seus parâmetros (filtros IP, parâmetros token-bucket) são representados por PRCs distintas. Estes elementos são organizados de acordo com a política de QoS e permanecem sempre na mesma ordem. O tráfego pode ser classificado (*ClfrElement*) e então medido (*Meter*). Cada fluxo de dados identificado pela combinação de classificadores e medidores pode sofrer uma ação (*Action*). Além disso, pode haver algoritmos de descarte (*AlgDrop*) ou o fluxo pode ser armazenado em uma fila (*Queue*) antes de ser escalonado (*Scheduler*) para o envio ao próximo destino. O

conjunto de operações realizadas pela PIB segue o atributo *next* presente nos diferentes elementos, o qual indica o próximo passo no processamento *diffserv*.

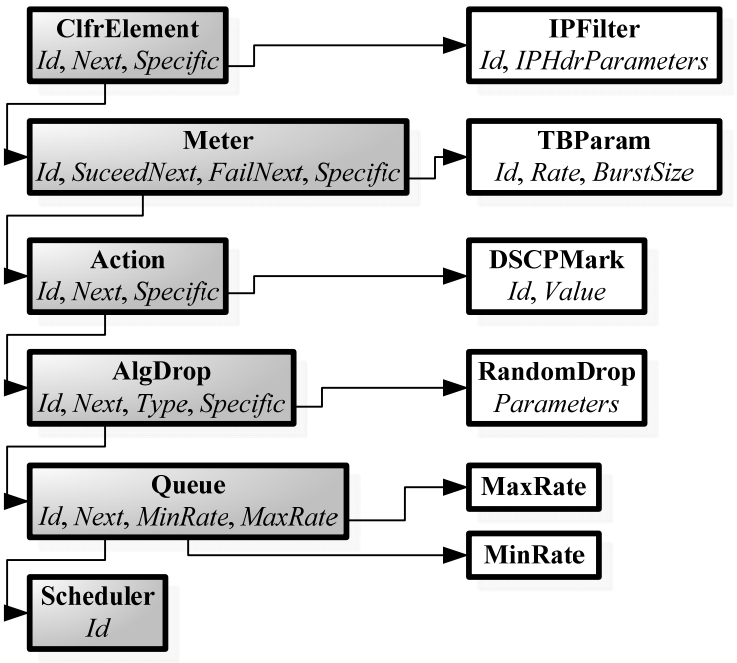

Figura 7.5: Visão geral do PIB *Diffserv*

A seqüência do tratamento *diffserv* aplicado a um pacote e a parametrização do tratamento são mantidas de forma separada na PIB *Diffserv*. O atributo *specific* presente nos elementos indica quais são os parâmetros determinados para determinado tratamento. Por exemplo, na Figura 7.5, o elemento classificador consiste em um filtro IP (*IPFilter*), a ação corresponde à marcação do campo DS do pacote (*DSCPMark*) e o algoritmo de descarte utilizado é o algoritmo de descarte aleatório (*RandomDrop*).

A Figura 7.6 mostra um exemplo de como os pacotes de um usuário são classificados (pelo filtro IP) e recebem uma política de tratamento específica (pelo medidor – *TBParam* e algoritmo de descarte) e ação de marcação específica (*DSCPMark*). A ação de marcação é responsável por atribuir uma classe agregada para os pacotes gerados pelo usuário. Os elementos e seus parâmetros são referenciados através de identificados chamados *object identifiers* (OID).

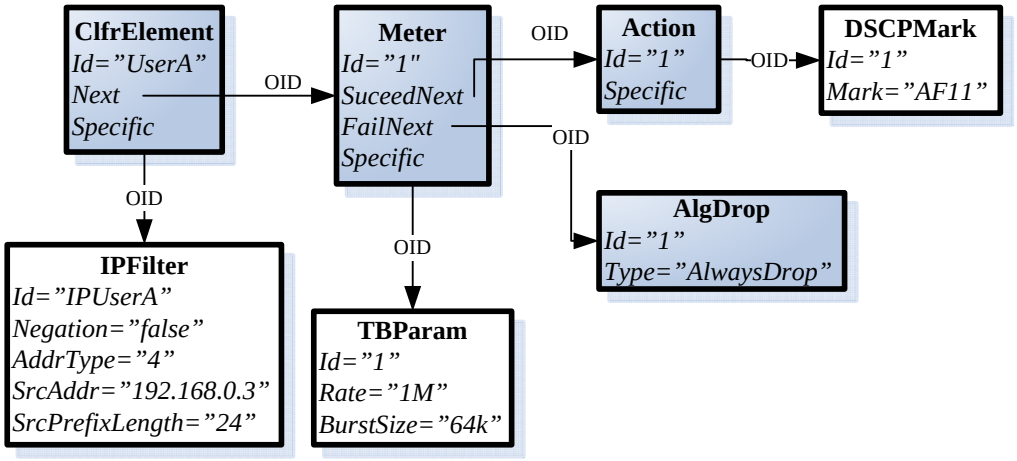

Figura 7.6: Exemplo de configuração da PIB *Diffserv*

Considerando o cenário da proposta, a solução simples seria um PEP conter a configuração de todos os possíveis usuários móveis que podem conectar-se em sua rede. Essa abordagem, entretanto, é, compreensivelmente, não prática. Deste modo, a arquitetura proposta adota uma configuração dinâmica dos PEPs, a qual é disparada pelos através de um evento de mobilidade gerado pelo HA. O evento de mobilidade é notificado a partir do momento em que o HA autentica a alteração de localização de um nó móvel.

Analisando a estrutura da PIB *Diffserv*, pode-se observar que o processo de configuração pode ser significantemente simplificado se os elementos da PIB forem classificados em informações estáticas e informações dinâmicas. Essa abordagem, presente na Figura 7.7, assume que a maioria dos usuários compartilhará um pequeno número de definições SLS. As definições SLS são representadas por PRCs “brancas” e são consideradas informações estáticas. As PRCs responsáveis pela atribuição do SLS (isto é, mapear um usuário para uma definição SLS) são representadas por PRCs “escuras” e são consideradas informações dinâmicas. Uma vez que é assumido que todos os túneis são criados entre o FA e o HA, o nó móvel é representado por seu endereço de origem.

A informação estática pode ser provisionada no momento da inicialização do PEP. A configuração dinâmica deve ser atualizada quando o nó móvel move-se de um domínio de um PEP para outro. Neste caso, o novo PEP do nó móvel deve receber novas informações de filtros para associar os pacotes gerados pelo nó móvel ao SLS atribuído ao usuário. Similarmente, a informação de filtros deve ser removida do PEP anterior.

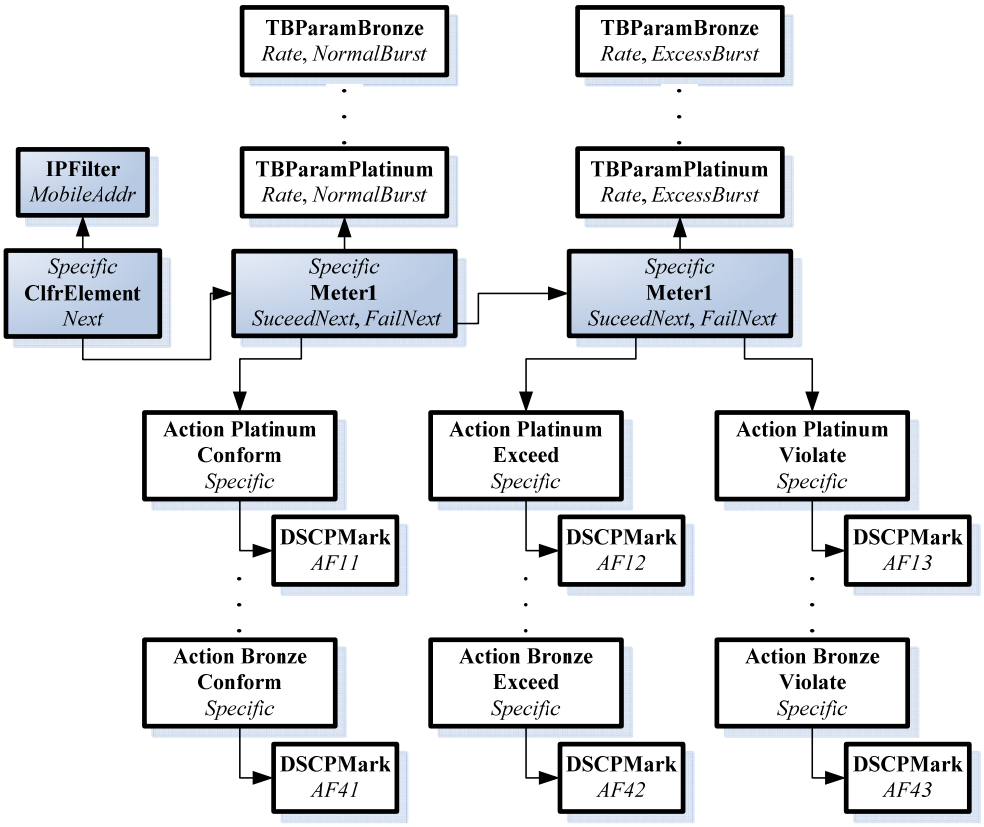

Figura 7.7: Configuração estática e dinâmica da PIB

As mensagens trocadas entre o PDP e os PEPs são ilustradas na Figura 7.8. O primeiro conjunto de mensagens (1 a 5) corresponde ao provisionamento inicial, no qual o PEP requisita a configuração estática da PIB. Depois disso, o PDP marca a *flag* da PIB *fullstate* = *false*, informando o PEP que as próximas mensagens DEC devem ser interpretadas como atualizações (isto é, o PEP não deve apagar a PIB *encarnation* anterior).

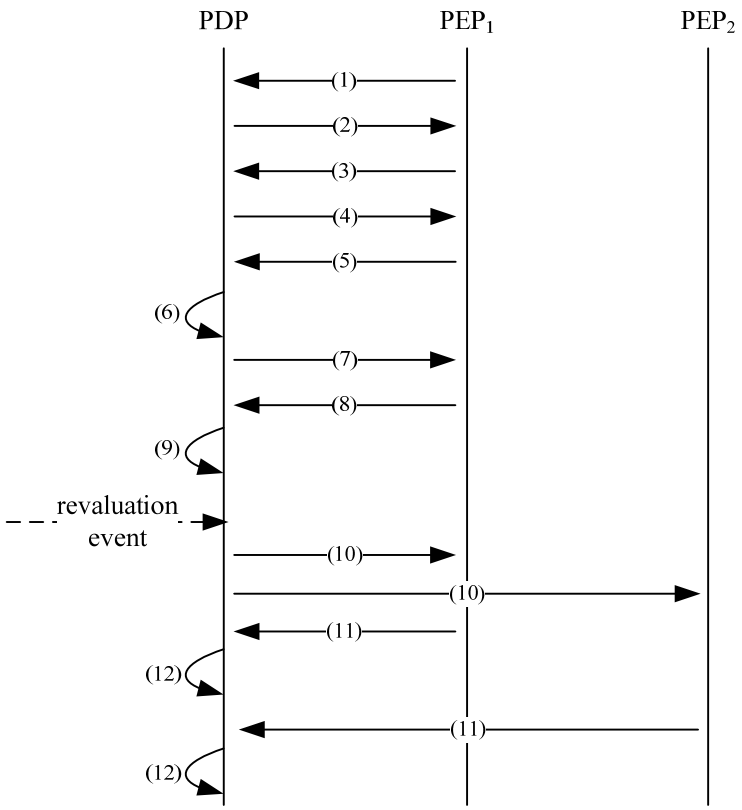

Figura 7.8: Troca de mensagens COPS-PR entre o PEP e o PDP durante o processo de provisão

(1) O PEP solicita uma abertura de conexão ao PDP através de uma mensagem *client-open* (OPN). Duas informações são transmitidas nessa mensagem: o tipo de cliente (*Client-Type*) e a identificação do PEP (*PepId*).

(2) A mensagem *client-accept* (CAT) enviada pelo PDP informa o intervalo de tempo no qual há a verificação da conexão entre o PEP e o PDP através de mensagens *keep-alive* (KA).

(3) O PEP, através de uma mensagem *request* (REQ), solicita uma configuração ao servidor de políticas. Nesta mensagem o PEP informa suas capacidades (*capabilities*), suas limitações e a combinação de papéis das interfaces do dispositivo.

(4) Após receber as informações a respeito do PEP, o PDP cria uma mensagem *decision* (DEC) contendo as configurações a serem adicionadas ou removidas do dispositivo. A configuração é formada de acordo com cada tipo de cliente (*Client-Type*).

(5) O PEP informa o sucesso ou falha na aplicação da configuração recebida pelo PDP através de uma mensagem *report state* (RPT).

(6) O PDP armazena internamente o estado atual do PEP para, nas mensagens futuras, somente enviar as atualizações da PIB, e não a PIB completa.

(7) O PDP notifica ao PEP que as próximas mensagens DEC devem ser interpretadas como atualizações. Essa notificação é realizada com a marcação da *flag* da PIB chamada *fullstate* com o valor *false*.

(8) O PEP informa o sucesso ou falha na aplicação da configuração recebida pelo PDP através de uma mensagem RPT**.**

(9) O estado atual do PEP é armazenado no PDP.

(10) Ao receber um *evento de mobilidade* informando a mobilidade de um dispositivo (evento disparado pelo HA), o PDP processa a informação e, com as informações armazenadas na etapa (9), envia uma mensagem DEC ao PEP somente com as atualizações a serem realizadas na PIB do dispositivo ao qual o nó móvel conectou-se. O PDP, com as informações também recebidas pelo HA, verifica o(s) roteador(es) anterior(es) ao(s) qual(is) o nó móvel estava conectado e também envia uma mensagem DEC a este(s) PEP(s), para que removam os filtros correspondentes ao nó móvel.

(11) O(s) PEP(s) informa(m) o sucesso ou falha na aplicação da atualização recebida pelo PDP através de uma mensagem RPT.

(12) Novamente o PDP armazena internamente o estado atual do(s) PEP(s) para utilização futura.

Como pode ser observado nas mensagens trocadas entre o servidor de políticas e um cliente, as decisões tomadas pelo PDP não são necessariamente respostas a requisições de um PEP. O PDP pode enviar uma mensagem de decisão como reação a um evento de mobilidade, por exemplo, ou a outro tipo de evento definido, como configurações para diferentes horários do dia. O processo de atualização é implementado por uma mensagem de decisão não solicitada transmitida do PDP para o PEP.

Caso não seja possível aplicar as políticas nos roteadores de borda devido à ausência de recursos, o tráfego do usuário móvel recebe o tratamento *default*, isto é, o encaminhamento por melhor esforço (*best-effort*).

7.5 Tratamento do Tunelamento dos Pacotes

O protocolo IPv4 Móvel utiliza a técnica de tunelamento para permitir que o nó móvel seja sempre acessível pelos outros nós e para que não haja interrupção da comunicação. Quando o nó móvel está em uma rede estrangeira, a comunicação entre o HA e o FA ao qual o nó está conectado é realizada através do encapsulamento de um pacote IP por outro pacote IP (PERKINS, 1996a).

Enquanto o cabeçalho do pacote IP interno é o pertencente ao tráfego original, o cabeçalho IP externo é acrescentado e removido nas pontas do túnel. No protocolo IPv4 Móvel, o cabeçalho interno contém o endereço de origem do nó móvel (*home address*) e o cabeçalho externo contém o CoA do nó (Figura 7.9).

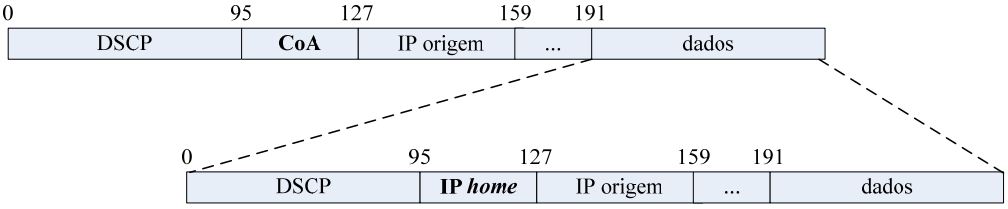

Figura 7.9: Tunelamento IP-IP com o protocolo IPv4 Móvel

O tratamento do tunelamento é importante porque, de maneira geral, os nós de rede intermediários que fazem parte de um túnel verificam somente o cabeçalho IP externo, e, dessa forma, os roteadores *diffserv* intermediários acessam e modificam somente o campo DS do cabeçalho IP externo.

Um dos modelos conceituais definidos em Black (2000) considera o tunelamento IP como um todo do ponto de vista do condicionamento do tráfego, isto é, o tunelamento não possui impacto no condicionamento do tráfego. Neste modelo, o qual é utilizado pela proposta, somente é considerado para o condicionamento do tráfego o campo DS do cabeçalho IP externo. No segundo modelo conceitual (modelo de tubo), o cabeçalho é ignorado na saída do túnel e somente o campo DS do cabeçalho IP interno é considerado.

Por trata-se de um único domínio *diffserv*, no qual o mapeamento do campo DS para um PHB específico nós intermediários é o mesmo, a solução utilizada pela arquitetura proposta consiste em aproveitar o condicionamento de tráfego realizado no ingresso do túnel, através da cópia do campo DS do cabeçalho IP interno para o cabeçalho IP externo, e no egresso, através da cópia do campo DS do cabeçalho IP

externo para o cabeçalho IP interno. Deste modo, o condicionamento de tráfego não é afetado pela presença do tunelamento que ocorre quando da utilização do protocolo IPv4 Móvel. É possível também com a metodologia de cópia do campo DS é possível manter as alterações sofridas pelo pacote (remarcação) quando em um tunelamento.

A Figura 7.10 sintetiza o tratamento do tunelamento dos pacotes na arquitetura proposta. Uma vez que o procedimento de registro utilizado na proposta para o IPv4 Móvel é o realizado através do FA/roteador de borda (utilização de um CoA ao invés de um *co-located* CoA), tanto o ponto de ingresso quanto o ponto de egresso do túnel são também pontos de ingresso e egresso do domínio *diffserv*.

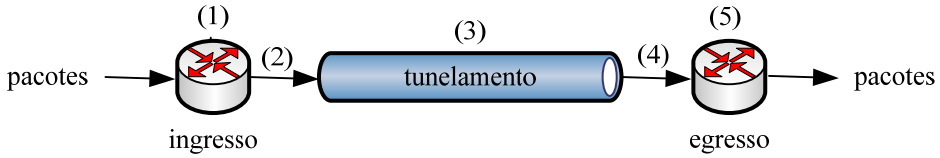

Figura 7.10: Tunelamento na arquitetura proposta

(1) No nó de ingresso do domínio *diffserv*, é realizado o condicionamento de tráfego.
(2) No momento do encapsulamento, o campo DS do cabeçalho IP interno é copiado para o cabeçalho IP externo.
(3) Ao longo do caminho, pode haver a alteração do valor do campo DS, o que implica em um mapeamento para um novo PHB para o pacote.
(4) No momento do desencapsulamento, o campo DS do cabeçalho IP externo é copiado para o cabeçalho IP interno. Dessa forma, as modificações sofridas pelo pacote quando encapsulado não são perdidas.
(5) O no de egresso do domínio *diffserv* realiza o condicionamento de tráfego.

Capítulo 8

Validação da Proposta

Para avaliar a arquitetura proposta no Capítulo 6 e as decisões de projeto tomadas no Capítulo 7, bem como averiguar o impacto da introdução de qualidade de serviço a um ambiente móvel, é proposto um cenário para testes e para a validação da proposta. Desta forma, procura-se avaliar a integração da arquitetura de gerenciamento de mobilidade IPv4 Móvel com a metodologia *diffserv* de qualidade de serviço, sendo que a configuração dos mecanismos *diffserv* dos roteadores é realizada através da arquitetura PBNM.

8.1 Cenário de Testes

Como apresentado no Capítulo 6, o cenário utilizado para a validação da proposta consiste em um domínio *diffserv* composto por roteadores de borda e roteadores de núcleo, conforme ilustrado na Figura 8.1. Por questões de simplificação, os endereços das interfaces dos roteadores de borda ligadas aos roteadores de núcleo foram omitidos, bem como os endereços das interfaces dos roteadores de núcleo. O PDP está presente na mesma rede do HA para reduzir a latência na notificação do *evento de mobilidade*. Os roteadores estão conectados entre si por uma rede de 100Mbps.

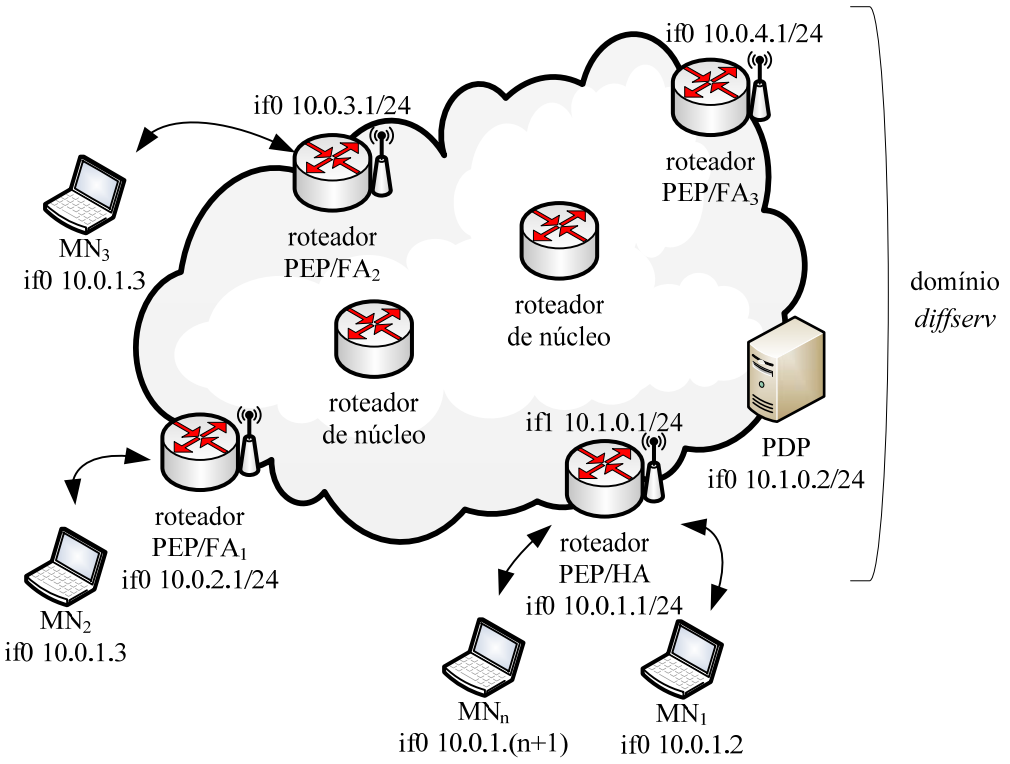

Figura 8.1: Cenário de testes

O endereço de origem dos nós móveis pertence à mesma rede da interface 0 (if0) do roteador de borda/HA. Isto significa que todos os nós móveis pertencem originariamente à mesma rede. Contudo, a utilização de uma arquitetura de mobilidade permite que esses nós continuem se comunicando mesmo quando conectados em redes distintas, isto é, conectados a diferentes FAs.

O PDP e os PEPs/FAs/roteadores de borda foram implementados em Java e hospedados em máquinas AMD Athlon 64 X2 3800+ com 1GB de memória RAM, rodando a distribuição de Linux Ubuntu 7.10 (*Gutsy Gibbon*).

O software de IPv4 Móvel, que corresponde ao *home agent*, ao *foreign agent* e ao nó móvel, é baseado no código disponível em Dynamics (2007). Originariamente, o software foi desenvolvido na Universidade de Tecnologia de Helsinki, para posteriormente ser disponibilizado sob a licença GNU GPL (*General Public License*).

Para adequar a implementação do protocolo IPv4 Móvel à proposta, algumas modificações no código do programa foram necessárias. Inicialmente, para aumentar a eficiência do protocolo de mobilidade e adequar ao proposto em Perkins (2002), foi necessário alterar o código do programa para diminuir o intervalo de tempo entre uma mensagem *agent advertisement* e outra, bem como para alterar o tempo de vida

(*lifetime*) durante o qual o nó móvel considera que está localizado na mesma rede, mesmo com a ausência de outras mensagens *agent advertisement*.

Através do arquivo de configuração disponível para as entidades de mobilidade HA e FA, a versão inicial do software somente permite a entrada de um número inteiro para o intervalo de tempo entre uma mensagem *agent advertisement* e outra. Deste modo, o menor intervalo de tempo possível entre tais mensagens é 1 (um) segundo na versão não modificada do código. Conforme define o padrão do protocolo IPv4 Móvel (PERKINS, 2002), o intervalo entre as mensagens *agent advertisement* não deve ser maior que 1/3 (um terço) do tempo de vida do *advertisement*, o que permite ao nó móvel perder três mensagens consecutivas antes de retirar a entidade (FA ou HA) de sua lista de agentes válidos. Conseqüentemente, o menor tempo de vida da mensagem *agent advertisement* definido originalmente na implementação é de 3 (três) segundos. Para melhorar o desempenho do *handover*, o código foi modificado para suportar *agent advertisements* com tempo de tempo de vida de um segundo.

Ademais, uma das alterações propostas para o protocolo IPv4 Móvel em relação à versão anterior do protocolo (PERKINS, 1996b), na qual foi baseado o código de em Dynamics (2007), consiste na possibilidade do envio de mensagens em um intervalo inferior a 1/3 do tempo de vida. Com a alteração implementação do protocolo, é possível que as entidades HA e FA emitam mais de 3 (três) mensagens *agent advertisement* por segundo, auxiliando também na melhora de velocidade do processo de *handover*.

Para notificar o servidor de políticas da autenticação dos nós móveis (notificar um evento de mobilidade), a implementação do HA foi alterada para realizar uma chamadas ao *web service* disponibilizado pelo PDP. Estas mensagens contêm as informações necessárias ao PDP para que ele selecione as políticas corretas e saiba onde aplicá-las.

Conforme o definido na seção 7.3, podem ser notificados ao PDP o endereço de origem do nó móvel, o endereço do FA/HA no qual está localizado (ou, no caso de uma associação simultânea, o provável FA/HA ao qual o nó irá conectar-se futuramente), a lista de um ou mais FAs não há mais a associação com o nó móvel, e o intervalo de tempo no qual o registro pode ser considerado válido. A utilização destes parâmetros na notificação do evento de mobilidade depende de qual é o tipo de mensagem *registration request* recebida pelo HA.

Inicialmente, é necessário verificar se o nó móvel está autorizado a realizar o pedido de registro com o HA (confirmação da mensagem *registration request*), para somente então sinalizar o evento de mobilidade ao PDP. Na implementação da arquitetura, este momento coincide com o envio da mensagem *registration reply* para o nó móvel, na qual o HA informa que o pedido de registro foi recebido e as informações de localização do nó móvel foram alteradas ou atualizadas.

Para o PDP receber as sinalizações enviadas pelo HA, o servidor de políticas disponibiliza um *web service* através da arquitetura Apache Axis2 (The Apache Software Fundation, 2007), a qual implementa o protocolo *Simple Object Access Protocol* (SOAP) e a leitura de dados no formato XML utilizado para a comunicação entre as diferentes entidades. O protocolo SOAP é responsável por encapsular os dados representados em linguagem XML.

Os mecanismos *diffserv* nos roteadores são implementados através da ferramenta *linux traffic control* disponível na plataforma Linux. O controle de tráfego consiste em um conjunto de sistemas de fila e mecanismos pelos quais os pacotes são enviados e recebidos em um roteador.

8.2 Procedimento dos Testes

Os testes foram propostos para avaliar o impacto da integração da metodologia de qualidade de serviço *diffserv* com a mobilidade de dispositivos. Para isso, é necessário avaliar a latência introduzida pelo processo de atualização da PIB *Diffserv* dos roteadores de borda através da arquitetura PBNM.

Para verificar a escalabilidade do sistema, é preciso verificar também o impacto de um número crescente de nós realizando o processo de *handover* na latência dos processos do protocolo IPv4 Móvel e na latência dos processos de decisão e aplicação das configurações *diffserv* nos roteadores de borda .

Para realizar a simulação de movimentação dos nós móveis e seu *handover*, foi utilizado um *script* em programação *bash*, onde era possível a criação ou extinção de nós móveis dinamicamente, isto é, a criação e a extinção de processos (*deamons*) de nó móvel do programa disponível em Dynamics (2007). Com isso, é possível simular a movimentação de um, dois ou mais nós móveis simultaneamente.

A Figura 8.2 apresenta a troca de mensagens entre as diferentes entidades da arquitetura proposta no processo de *handover* com os pontos onde é possível análise dos

resultados dos testes. Os testes somente consideram a mudança de localização do nó móvel sem a configuração antecipada dos roteadores de borda, isto é, dentro da arquitetura proposta, sem o suporte a múltiplas associações. Isto permite avaliar a arquitetura no pior caso, quando o roteador de borda para o qual o nó móvel desloca-se ainda não possui nenhuma configuração relativa a esse nó.

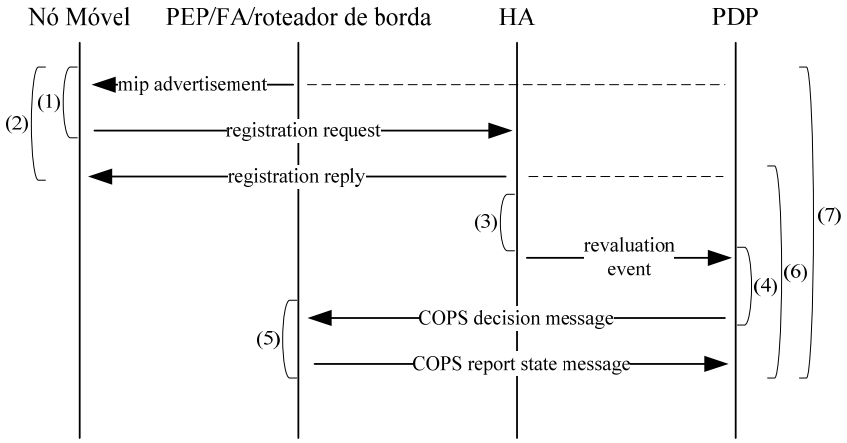

Figura 8.2: Avaliação da troca de mensagens entre as entidades

(1) Tempo entre o recebimento do primeiro *advertisement* do novo FA devido à mudança de rede e o envio de uma mensagem *registration request* pelo nó móvel.

(2) Tempo compreendido entre a mudança de rede, o pedido de registro de um nó móvel (*registration request*) e sua confirmação por parte do HA (*registration reply*) na arquitetura IPv4 Móvel. Este tempo reflete todo o processo pelo qual o protocolo IPv4 Móvel permite a movimentação dos usuários em diferentes redes sem interrupção da comunicação. Neste ponto, é possível verificar qual é o impacto da alteração do código fonte da implementação do protocolo IPv4 Móvel (DYNAMICS, 2007) no período de tempo necessário para o *handover* do nó móvel.

(3) A partir da confirmação da movimentação do nó, qual é tempo necessário para produzir a chamada em *web service*, isto é, tempo necessário para gerar o *evento de mobilidade*.

(4) Tempo de recebimento e processamento do evento de mobilidade por parte do servidor de políticas (PDP) para após gerar a mensagem de atualização não solicitada de políticas (COPS *decision message*).

(5) Tempo necessário para a aplicação das novas políticas nos roteadores de borda e confirmação das alterações por parte do PEP (COPS *report state message*).

(6) Tempo total necessário para a aplicação de novas políticas nos roteadores de borda. Este tempo reflete o impacto da presença do gerenciamento de QoS em um ambiente de mobilidade. Ressalte-se que, para um melhor desempenho da arquitetura proposta, utiliza-se a abordagem dinâmica de atualização discutida na seção 7.4. Este tempo corresponde à soma dos tempos de (3), (4) e (5).

(7) Atraso total da arquitetura proposta integrando a mobilidade do usuário com qualidade de serviço. É possível verificar neste ponto qual é o intervalo de tempo entre a mudança de rede de um nó móvel até que ele possua seus privilégios de qualidade de serviço corretamente configurados na rede à qual ele está conectado. Corresponde à soma dos tempos (2) e (6). Enquanto os mecanismos *diffserv* não forem configurados nos roteadores de borda, não é atribuída nenhuma classe AF ou EF para o tráfego do nó móvel, o qual é marcado somente como *best-effort* (BE).

8.3 Resultados

A Figura 8.3 exibe os atrasos médios obtidos (medidos em segundos) nos eventos mais importantes relacionados ao *handover*. Os resultados apresentados devem ser considerados somente como comparação entre os resultados obtidos nesse mesmo ambiente de avaliação, uma vez que, no ambiente simulado, há somente um salto entre o FA e o HA, e os roteadores estão ligados por enlaces de 100Mbps. Os dados da figura referem-se ao primeiro cenário, no qual há uma transição por segundo (1 *handover*/s). Para este e para os demais cenários de testes, foram realizadas 20 interações para cada, para então ser calculada a média e o desvio padrão de cada um dos pontos de análise da Figura 8.2.

As mensagens de *advertisement* e de registro do nó móvel (*registration request* e *registration reply*) foram capturadas na interface do nó móvel. O tempo entre a confirmação do registro (*registration reply*) e o envio de um *evento de mobilidade* foi obtido na interface pública do HA, isto é, na interface visível para os FAs e para o PDP. O recebimento de um *evento de mobilidade* e as mensagens do protocolo COPS-PR, as quais transmitem as atualizações da PIB *Diffserv*, foram capturadas no PDP. A sincronização dos tempos das mensagens capturadas em diferentes dispositivos foi

realizada através comparação entre o tempo de envio da mensagem em um dispositivo e o tempo de recebimento dessa mesma mensagem em outro dispositivo. O programa utilizado para a captura das mensagens e análise dos tempos foi o *Wireshark Network Protocol Analyzer*, versão 0.99.7, disponível em Wireshark (2008).

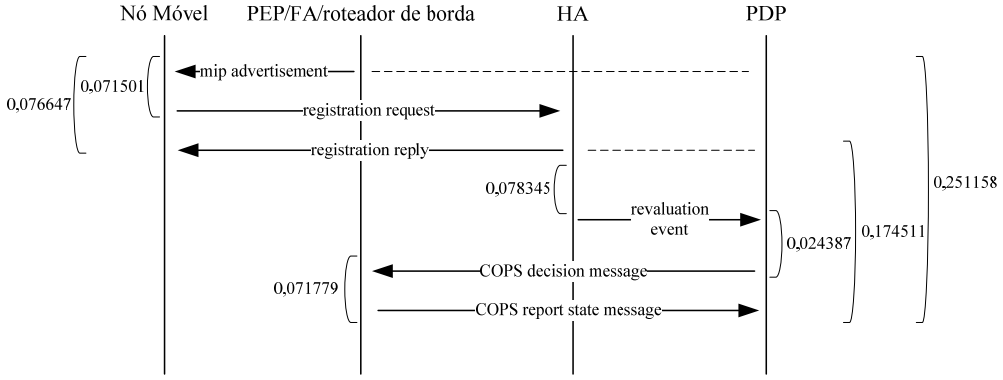

Figura 8.3: Mensagens relacionadas ao *handover* e seus atrasos médios (1 *handover*/s)

Para uma melhor análise da Figura 8.3, primeiramente serão tecidos comentários separadamente acerca dos pontos (1) a (5) da Figura 8.2, para após ser realizada a análise dos tempos obtidos nos pontos (6) e (7), sob uma ótica geral.

No que se refere ao ponto (1), observa-se que a modificação introduzida no software do protocolo IPv4 Móvel surtiu efeito na redução do tempo necessário para a realização do *handover* pelo nó móvel. Uma vez que o tempo de vida do pacote da mensagem de *advertisement* do protocolo IPv4 Móvel é de 1 (um) segundo, no pior caso, o tempo entre o recebimento de uma mensagem do FA de outra rede e o envio de uma mensagem *registration request* é o próprio tempo de vida do pacote ICMP (do qual o *mip advertisement* é uma extensão).

A outra modificação introduzida no software, qual seja, a redução do intervalo de emissão de mensagens de *advertisement*, também influencia no desempenho do *handover*, uma vez que é somente a partir do recebimento de uma mensagem de *advertisement* de uma nova rede que o nó móvel irá detectar que se encontra em outra rede e irá solicitar a atualização de sua posição ao seu HA. Na modificação introduzida na implementação do protocolo, são enviadas pelas entidades (HA e FA), em média, 10 *advertisements* por segundo.

Enquanto o nó móvel está em funcionamento, ele mantém o controle das mensagens de *advertisement* recebidas de diferentes FAs durante o intervalo de tempo no qual a mensagem é considerada válida (tempo de vida da mensagem). Depois de

esgotado o tempo de vida da mensagem, a informação obtida através dela também é descartada. Caso o nó móvel não receba mais as mensagens de *advertisement* da rede à qual ele está conectado, ele inicialmente verifica se há alguma outra rede à qual ele possa se conectar, isto é, se ele recebeu alguma mensagem de *advertisement* de outra rede e se essa mensagem ainda não expirou. Em caso positivo, de posse das informações recebidas, o nó conecta-se à nova rede.

Vistas as duas modificações introduzidas na implementação do protocolo IPv4 Móvel, adequando-o às alterações do padrão sugeridas em Perkins (2002), o tempo total médio do processo de *handover* – ponto (2) – no cenário da Figura 8.3, foi de 77ms.

O tempo registrado no ponto (3) refere-se à alteração introduzida na arquitetura de mobilidade para a notificação do *evento de mobilidade* definido nesta proposta. A partir da confirmação do registro do nó móvel por parte do HA, são necessários, em média, 78ms para a criação de uma mensagem *web services* para o serviço disponível no servidor de políticas (PDP).

Recebida a mensagem *web services* pelo PDP, conforme o ponto (4), são necessários 24ms para o servidor de políticas analisar o repositório de políticas e verificar quais são as políticas necessárias a serem aplicadas nos roteadores de borda. Uma vez que é utilizada na arquitetura proposta uma abordagem dinâmica para a aplicação das políticas, separando-as em informações estáticas e informações dinâmicas, não é necessário enviar todas as políticas novamente, mas somente as alterações sofridas pela PIB *Diffserv*. A seleção das políticas leva em consideração, entre outros fatores, qual é o endereço de origem do nó móvel, pois é a ele que estão vinculados os filtros e marcadores definidos na seção 7.4.

De acordo com o ponto (5), são necessários 72ms para o cliente de políticas (PEP) aplicar as atualizações da PIB recebidas em uma mensagem não solicitada ao PDP. Este tempo é menor do que aquele que seria necessário caso fosse necessário reaplicar toda a PIB *Diffserv* no PEP, e não somente sua atualização. A abordagem de somente enviar uma atualização da PIB importa tanto em um tempo menor para a aplicação da PIB pelo PEP quanto em um menor tráfego na rede devido às mensagens de decisão (COPS-PR *decision messages*) enviadas pelo PDP.

Uma vez analisados os impactos individuais de cada uma das trocas de mensagens e do tempo gasto para o processamento de algumas destas mensagens, cabe analisar os tempos obtidos na arquitetura proposta como um todo.

A introdução de qualidade de serviço à arquitetura de mobilidade introduz um atraso correspondente à aplicação das configurações pertinentes aos roteadores de borda, conforme a movimentação dos nós móveis. De acordo com o observado no ponto (6) da Figura 8.3, o tempo total médio necessário à arquitetura proposta para a configuração dos mecanismos *diffserv* dos roteadores de borda é de 174ms. Este tempo compreende desde a criação do evento de mobilidade pelo HA até o recebimento da mensagem dos roteadores de borda informando que as políticas foram aplicadas com sucesso.

Durante o intervalo de tempo em que o nó móvel registra-se em sua nova rede (seja ela uma nova rede estrangeira, ou a volta à sua rede de origem) e o tempo necessário para a aplicação das políticas no dispositivo ao qual o nó está conectado, este nó não possui as qualidades de serviços contratadas perante o provedor do serviço e acordadas em um SLA. Destarte, quanto mais rápido se dar a aplicação da configuração nos dispositivos, menor será a degradação do serviço oferecido. Esta degradação é prevista em um SLA através de porcentuais pré-definidos de degradação de serviço. Isto significa que, dentro do porcentual acordado, o serviço oferecido está de acordo com o contrato e não há punição (multa, por exemplo) ao provedor do serviço.

Finalmente, o tempo de 251ms – ponto (7) – corresponde ao tempo total médio necessário para um nó móvel detectar que está em uma nova rede e que seus pacotes sejam priorizados conforme o estabelecido em um SLA. Como já exposto anteriormente, as prioridades para determinados tipos de tráfego são definidas nos roteadores de borda com o uso de mecanismos *diffserv*. Deste tempo total, 30,7% corresponde ao tempo necessário para o nó móvel registrar-se em uma nova rede e 69,3% corresponde ao tempo para a aplicação dos mecanismos *diffserv* nos roteadores de borda com o uso de uma arquitetura PBNM.

Para verificar a escalabilidade do sistema, é necessário verificar o impacto de um número crescente de nós móveis e sua mobilidade entre diferentes redes. Para isto, a Tabela 8.1 apresenta os tempos e o desvio padrão obtidos no cenário ilustrado na Figura 8.3 e em outros cenários, nos quais há um acréscimo do número de *handovers* por segundo. Os índices presentes na Tabela correspondem aos intervalos de análise definidos na Figura 8.2.

	1 *handover*/s	2 *handover*/s	3 *handover*/s	4 *handover*/s	5 *handover*/s
(1)	$\overline{x}$ = 0,071501 σ = 0,135694	$\overline{x}$ = 0,190396 σ = 0,188564	$\overline{x}$ = 0,190995 σ = 0,182687	$\overline{x}$ = 0,235167 σ = 0,215338	$\overline{x}$ = 0,256447 σ = 0,205822
(2)	$\overline{x}$ = 0,076647 σ = 0,135790	$\overline{x}$ = 0,199420 σ = 0,189240	$\overline{x}$ = 0,201913 σ = 0,186136	$\overline{x}$ = 0,258765 σ = 0,213346	$\overline{x}$ = 0,286994 σ = 0,205632
(3)	$\overline{x}$ = 0,078345 σ = 0,001754	$\overline{x}$ = 0,076294 σ = 0,004906	$\overline{x}$ = 0,103981 σ = 0,023879	$\overline{x}$ = 0,121452 σ = 0,033047	$\overline{x}$ = 0,147581 σ = 0,042730
(4)	$\overline{x}$ = 0,024387 σ = 0,026254	$\overline{x}$ = 0,027016 σ = 0,021769	$\overline{x}$ = 0,028642 σ = 0,029352	$\overline{x}$ = 0,029254 σ = 0,030394	$\overline{x}$ = 0,029950 σ = 0,030630
(5)	$\overline{x}$ = 0,071779 σ = 0,029137	$\overline{x}$ = 0,071605 σ = 0,026113	$\overline{x}$ = 0,066454 σ = 0,033883	$\overline{x}$ = 0,064478 σ = 0,034285	$\overline{x}$ = 0,079647 σ = 0,032337
(6)	$\overline{x}$ = 0,174511 σ = 0,032947	$\overline{x}$ = 0,184670 σ = 0,048238	$\overline{x}$ = 0,198600 σ = 0,053983	$\overline{x}$ = 0,217251 σ = 0,070416	$\overline{x}$ = 0,257178 σ = 0,061750
(7)	$\overline{x}$ = 0,251158 σ = 0,136668	$\overline{x}$ = 0,384089 σ = 0,209844	$\overline{x}$ = 0,400513 σ = 0,189731	$\overline{x}$ = 0,495072 σ = 0,218227	$\overline{x}$ = 0,544172 σ = 0,217992

Tabela 8.1: Tempos obtidos nos cenários de teste da arquitetura proposta (valores em segundos)

Através da Figura 8.4, é possível observar o comportamento da arquitetura de mobilidade nos diferentes cenários presentes na Tabela 8.1. Esta e as demais figuras deste capítulo apresentam o intervalo de confiança, para média, de 95%, desconhecida a variância populacional.

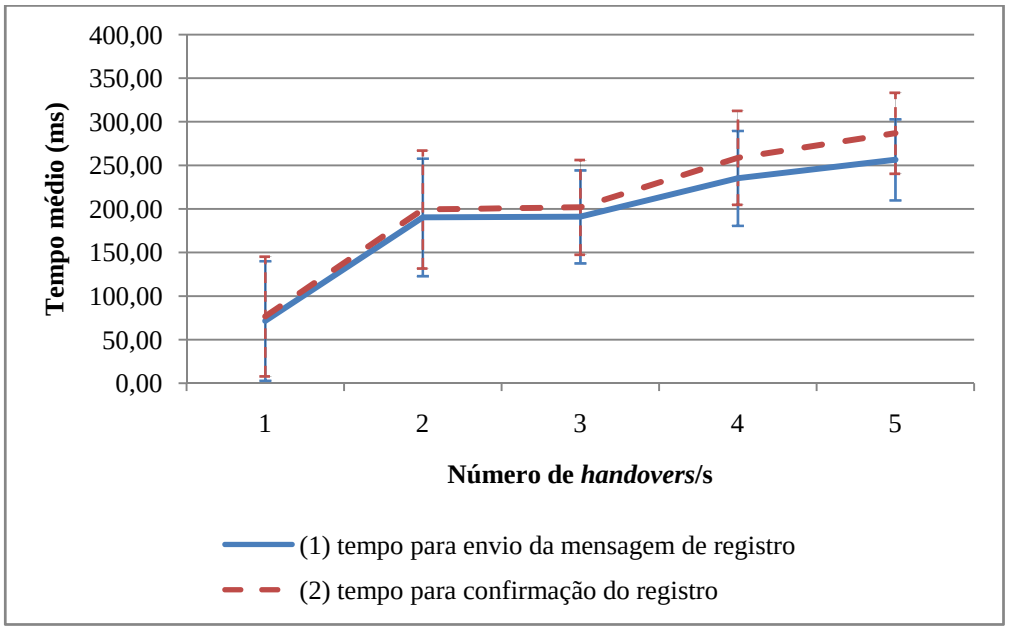

Figura 8.4: Tempos de resposta da arquitetura de mobilidade

Conforme há um acréscimo do número de *handovers* realizados durante um mesmo segundo, há um acréscimo no tempo médio tanto no envio de uma mensagem *registration request* pelo nó móvel quanto no envio de uma mensagem *registration reply* pelo HA. De acordo com o padrão IPv4 Móvel, infere-se que o intervalo de tempo

máximo esperado para o envio da mensagem de *registration request* é o tempo de vida da mensagem de *advertisement*, o qual é 1 (um) segundo nos testes realizados.

É possível verificar também que, conforme o acréscimo de *handovers*, o intervalo de tempo no qual o nó móvel aguarda uma confirmação do HA também sofre um acréscimo, visível pelo aumento da distância entre as linhas da Figura 8.4**.** Isto significa um aumento na carga do HA em gerenciar diversos usuários móveis.

Nos cenários testados, a entidade de gerenciamento de mobilidade HA é única para todos os usuários móveis. Em situações reais, diferentes usuários possuem diferentes redes de origem e, deste modo, possuem diferentes HAs.

A Figura 8.5 permite observar um crescimento linear no tempo médio necessário para a geração das mensagens *web service* de acordo com o aumento do número de *handovers*. A figura reflete o intervalo de tempo entre a confirmação da nova localização do nó móvel e o envio da mensagem ao servidor de políticas.

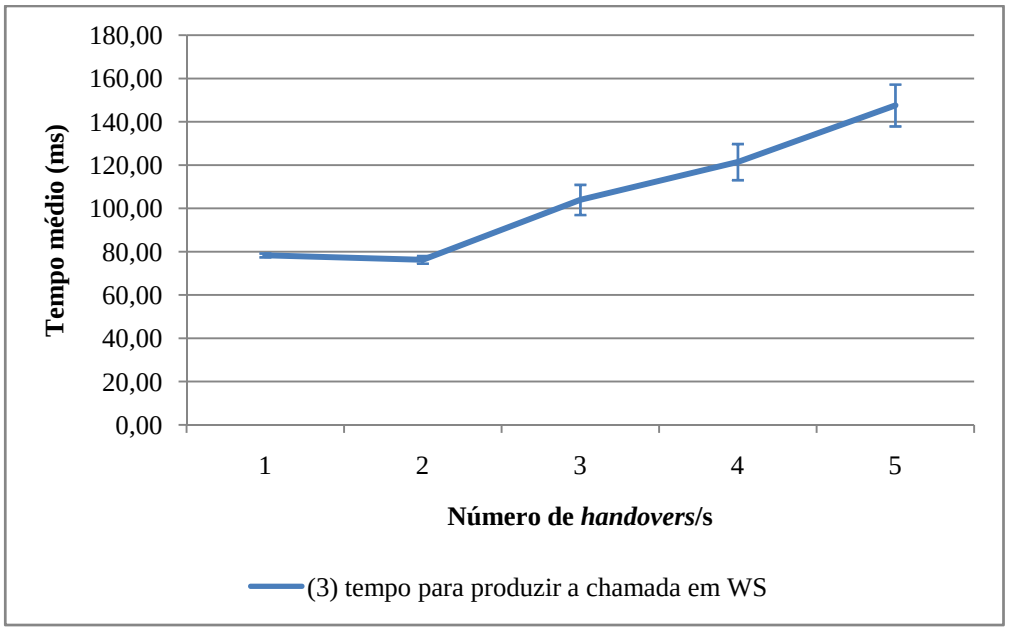

Figura 8.5: Tempo médio para a produção e envio da mensagem *web service*

A entidade responsável por gerar o *evento de mobilidade* e sinalizar a mobilidade de um dispositivo é o HA. Uma vez que HA é responsável pelo registro e encaminhamento das mensagens destinadas ao nó móvel, ele possui as informações necessárias ao PDP para reaplicar as políticas de configuração dos roteadores de borda. Novamente, cabe ressaltar que os cenários de teste consideraram somente a existência de um único HA para gerenciar todos os usuários móveis.

O tempo necessário para o processamento do *evento de mobilidade* e envio das novas políticas para os roteadores de borda nos diferentes cenários da Tabela 8.1 está presente na Figura 8.6.

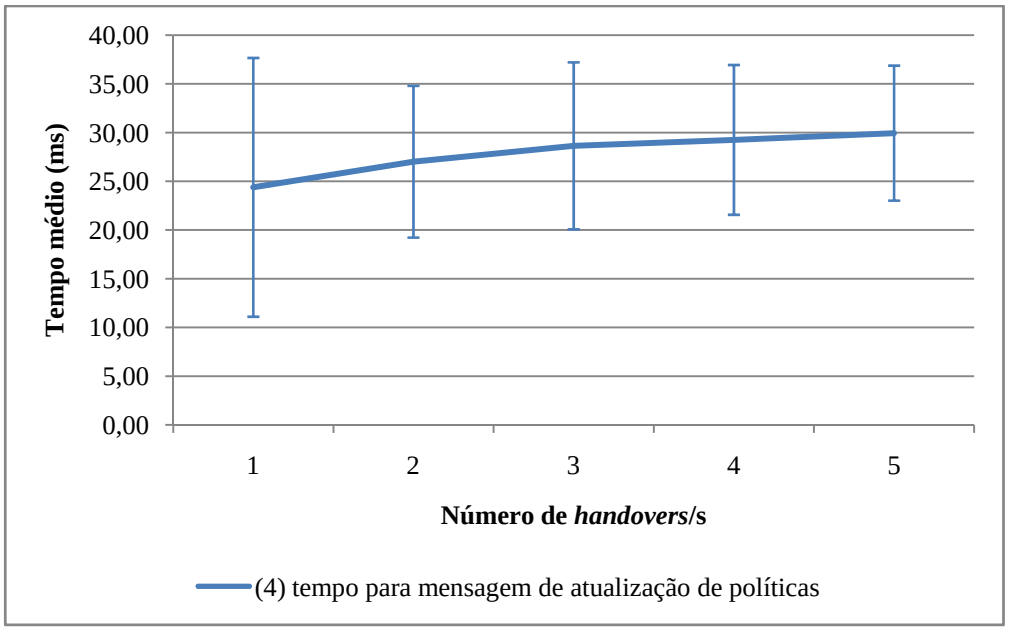

Figura 8.6: Tempo médio para o processamento do evento de mobilidade e envio de mensagem COPS-PR *decision*

De acordo com os resultados obtidos, o número crescente de *handovers* por segundo afeta em menor grau o servidor de políticas no que se refere à avaliação dos eventos de mobilidade recebidos, modificação das políticas existentes e envio das atualizações para os roteadores de bordas, os quais também acumulam a função de PEP da arquitetura PBNM.

Pode-se afirmar que o processamento dos *eventos de mobilidade* não produz um grande impacto no processamento do servidor de políticas. O tempo presente na Figura 8.6 reflete o tempo gasto pelo PDP para buscar as atuais políticas existentes para o PEP ao qual o nó móvel está conectado, inserção dos novos filtros na PIB e envio da mensagem não solicitada.

Como se observa na Figura 8.7, o tempo para a aplicação das políticas no PEP e o envio da mensagem de confirmação (*report state*) não sofre reflexo proporcionalmente ao número de usuários móveis nele conectados.

Cada usuário móvel é tratado de forma individual pelo PEP, isto é, a cada usuário que se conecta no roteador de borda, novas configurações devem ser recebidas e aplicadas. Os resultados demonstram que o número crescente de configurações

aplicadas em um intervalo de tempo não impacta no tempo necessário à aplicação dessas configurações.

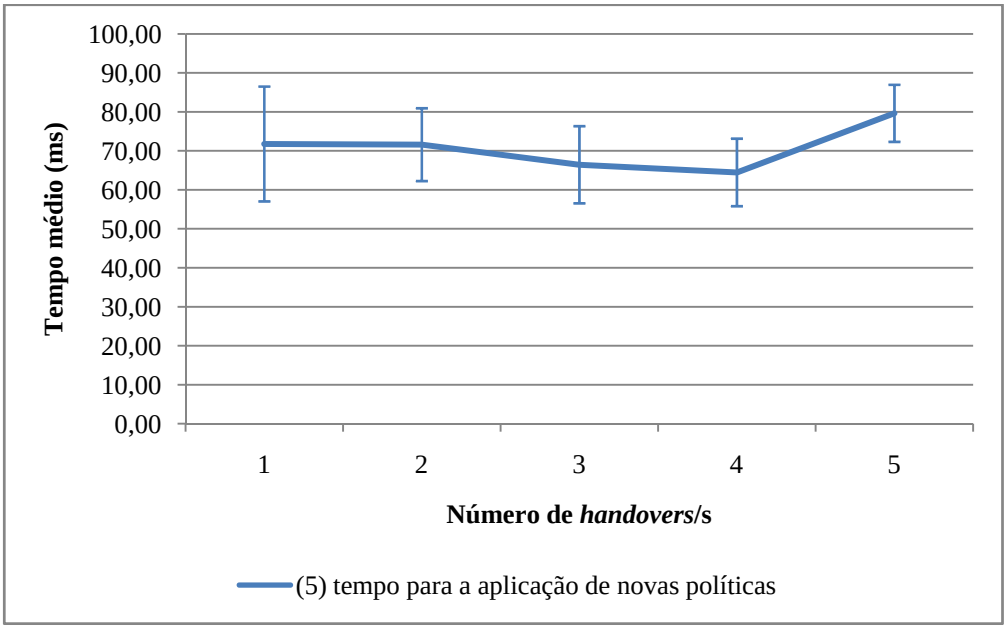

Figura 8.7: Tempo para a aplicação das políticas nos PEPs

A Figura 8.8 mostra o impacto da arquitetura PBNM com o suporte aos eventos de mobilidade no tempo total da arquitetura proposta. O crescimento mais acelerado do tempo total da arquitetura deve-se ao crescimento dos tempos obtidos na arquitetura de gerenciamento de mobilidade. Em outras palavras, o aumento do tempo total da arquitetura proposta deve-se mais ao crescimento do tempo médio obtido na arquitetura de mobilidade do que do crescimento dos tempos obtidos na arquitetura PBNM, conforme o aumento do número de *handovers* por intervalo de tempo.

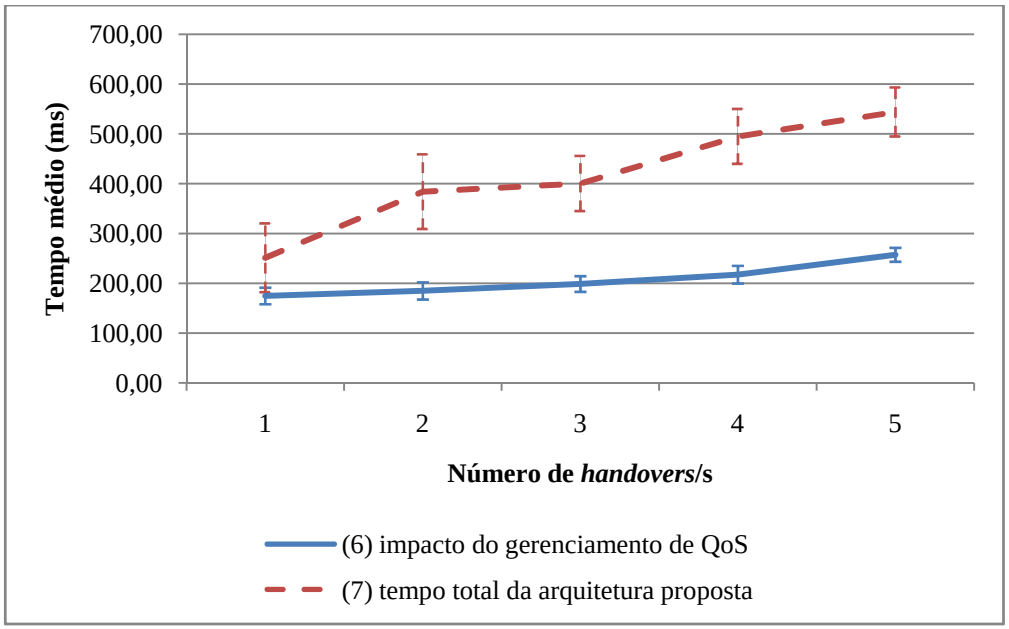

Figura 8.8: Impacto da configuração dos dispositivos no tempo total da arquitetura proposta

Evidencia-se que, inicialmente, uma melhora na arquitetura proposta é obtida de forma mais evidente com a melhora no gerenciamento de mobilidade. Por outro lado, os resultados obtidos na arquitetura de gerenciamento de mobilidade podem sofrer grande variação, pois o intervalo entre a detecção de uma nova rede e o registro realizado pelo nó (*registration request*) pode variar conforme o tempo de duração definido para as mensagens de *advertisement*.

A Figura 8.9 mostra mais claramente o impacto do gerenciamento de mobilidade na arquitetura proposta. O impacto do gerenciamento de mobilidade reflete em alto grau no impacto total da arquitetura proposta quando analisado um número crescente de *handovers*/s.

Por fim, através da Figura 8.10, é possível observar quem responde proporcionalmente, entre o gerenciamento de mobilidade e o gerenciamento de qualidade de serviço, pelos tempos médios obtidos na arquitetura proposta. Somente no primeiro cenário avaliado (1 *handover*/s), o gerenciamento de qualidade de serviço responde por mais da metade do tempo gasto pela arquitetura de integração proposta. A partir do cenário onde são considerados 2 *handovers*/s, o maior peso nos tempos médios obtidos cabe ao gerenciamento de mobilidade. Desta forma, evidencia-se que o custo da introdução do gerenciamento de qualidade de serviço em um ambiente móvel não é alto quando levado em consideração o atraso inerente à arquitetura IPv4 Móvel no momento do *handover* de um usuário móvel.

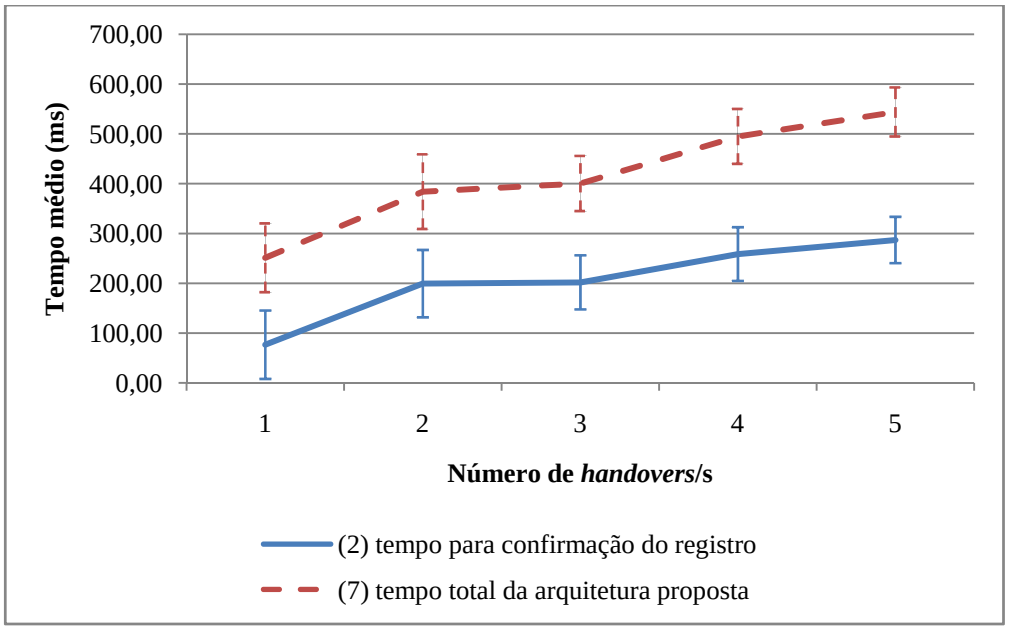

Figura 8.9: Reflexo do gerenciamento de mobilidade na arquitetura proposta

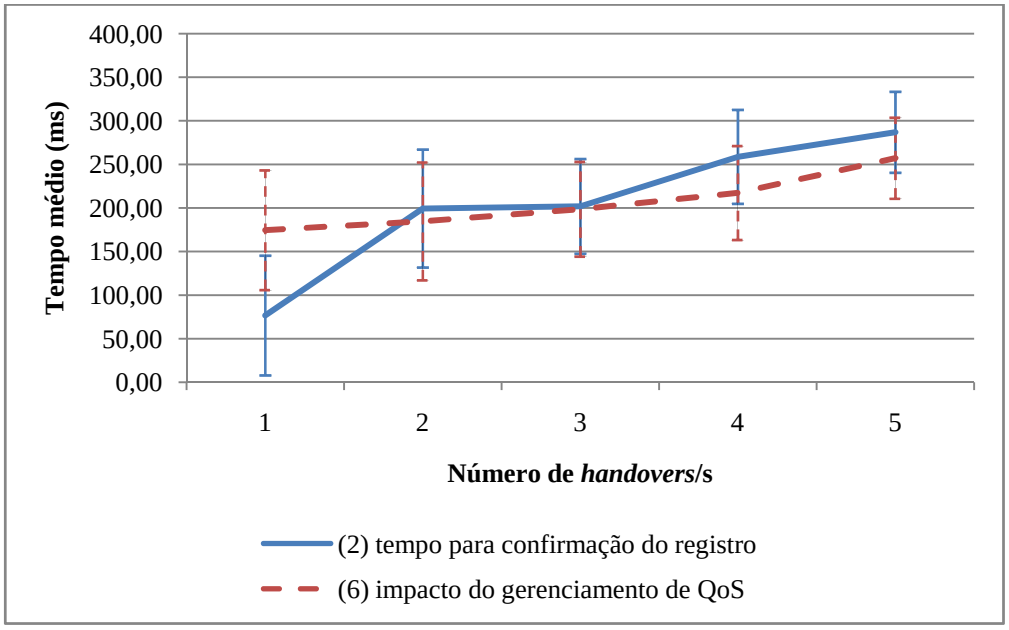

Figura 8.10: Impactos do gerenciamento de mobilidade e do gerenciamento de QoS na arquitetura proposta

Capítulo 9

Conclusão

A arquitetura proposta permite a manutenção da qualidade de serviço em um ambiente móvel. Este trabalho agrega-se às propostas já existentes, mas diferencia-se na utilização da arquitetura de gerenciamento PBNM para a aplicação das configurações nos roteadores de borda.

A arquitetura é baseada em padrões IETF relativos à configuração *diffserv*, onde os elementos mais importantes são a PIB *Diffserv* e o protocolo COPS-PR. Os resultados obtidos demonstram que PIB, através da diferenciação entre informações estáticas e informações dinâmicas, e o protocolo COPS-PR, com o uso de mensagens não solicitadas pelo cliente de políticas, fornecem flexibilidade suficiente para tratar do problema de mobilidade sem a introdução de novos protocolos ou modificações nos padrões já existentes.

Através deste trabalho, é definido um evento externo à arquitetura de gerenciamento chamado *evento de reavaliação*. Tal evento importa, dentro da arquitetura PBNM, a reavaliação das políticas existentes e a aplicação de novas políticas nos dispositivos gerenciados. Em particular, o *evento de reavaliação* toma o nome de *evento de mobilidade* quando utilizado na arquitetura proposta para notificar a mobilidade dos usuários em diferentes redes. As possibilidades de eventos gerados pela mobilidade de um dispositivo precisam ser notificadas de forma diferente ao servidor de políticas, o que é possível com a criação de um serviço *web services* no servidor de políticas e na disponibilização da interface para o HA realizar a notificação desses eventos.

O método de notificação da mobilidade de um nó através do evento de mobilidade mantém a independência dos protocolos relacionados à mobilidade das

metodologias de qualidade de serviço. Dessa forma, embora um *overhead* maior seja introduzido à proposta comparado a um método onde há a combinação da sinalização da mobilidade com a sinalização dos requisitos de qualidade de serviço, não é necessário a criação de um novo protocolo. Ademais, o resultado é um sistema de operação mais simples.

Os resultados dos testes realizados demonstraram que a arquitetura mostrou-se viável quando comparada ao atraso já introduzido pela arquitetura de mobilidade. O provisionamento somente das atualizações da PIB no momento do *handover* de um usuário permite reduzir o tempo necessário para a aplicação das novas configurações nos roteadores de borda bem como reduz o tamanho das mensagens que trafegam pela rede.

Com a flexibilidade dos *eventos de reavaliação*, novos tipos de eventos podem ser definidos e notificados ao servidor de políticas da arquitetura PBNM, inclusive no que se refere a políticas concernentes à segurança. Outra possibilidade é, através desses eventos, receber um *feedback* da rede e reaplicar novas políticas nos roteadores. Por exemplo, para que a não haja a quebra de um contrato e simultaneamente o usuário possa avaliar o serviço contratado, informações sobre o comportamento da rede podem ser notificadas ao servidor de políticas e armazenadas para posterior auditoria.

Referências Bibliográficas

BAKER, F. *Requirements for IP Version 4 Routers*. IETF RFC 1812, jun. 1995.

BELLER, André. *Uma Arquitetura para Gerenciamento de QoS Baseado em Políticas*. 2005. Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2005.

BERGER, L.; *et al*. *RSVP Refresh Overhead Reduction Extensions*. IETF RFC 2961, abr. 2001.

BERNET, Y.; *et al*. *An Informal Management Model for Diffserv Routers*. IETF RFC 3290, mai. 2002.

BLACK, D. *Differentiated Services and Tunnels*. IETF RFC 2983, out. 2000.

BLAKE, S.; *et al*. *An Architecture for Differentiated Services*. IETF RFC 2475, dez. 1998.

BRADEN, R.; CLARK, D.; SHENKER, S. *Integrated Services in the Internet Architecture: an Overview*. IETF RFC 1633, jun. 1994.

BRADEN, R.; ZHANG, L.; BERSON, S.; HERZOG, S.; JAMIN, S. *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*. IETF RFC 2205, set. 1997.

CAMPBELL, Andrew T.; *et al. Comparison of IP Micro-Mobility Protocols*. IEEE Wireless Communications, fev. 2002. p. 72-82.

CHAN, K.; *et al*. *COPS Usage for Policy Provisioning (COPS-PR)*. IETF RFC 3084, mar. 2001.

CHAN, K.; et al. *Differentiated Services Quality of Service Policy Information Base*. IETF RFC 3317, mar. 2003.

CHARNY, A.; *et al*. *Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior)*. IETF RFC 3247, mar. 2002.

CHASKAR, Hemant; KOODLI, Rajeev. *A Framework for QoS Support in Mobile IPv6*. Internet Draft, mar. 2001

CLARK, David D.; FANG, Wenjia. *Explicit Allocation of Best-Effort Packet Delivery Service*. 1998. IEEE/ACM Transactions on Networking, vol. 6, nº. 4, ago. 1998. p. 362-373.

CONTA, A.; DEERING, S. *Generic Packet Tunneling in IPv6 Specification*. IETF RFC 2473, dez. 1998.

DAVIE, B. *An Expedited Forwarding PHB (Per-Hop Behavior).* IETF RFC 3246, mar. 2002.

DEERING, S.; HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification*. IETF RFC 2460, dez. 1998.

DELGROSSI, L.; BERGER, L. *Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+*. IETF RFC 1819, ago. 1995.

DEVARAPALLI, V.; WAKIKAWA, R.; PETRESCU, A.; THUBERT, P. *Network Mobility (NEMO) Basic Support Protocol*. IETF RFC 3963, jan. 2005.

DURHAM, D. ; *et al*. *The COPS (Common Open Policy Service) Protocol*. IETF RFC 2748, jan. 2000.

DYNAMICS Mobile IP. Disponível em <http://dynamics.sourceforge.net>. Acesso em 10 dez. 2007.

FOGELSTROEM, E.; JONSSON, A.; PERKINS, C. *Mobile IPv4 Regional Registration*. IETF RFC 4857, jun. 2007.

GARCÍA-MACÍAS, J. Antonio; *et al. Quality of Service and Mobility for the Wireless Internet*. First ACM Wireless Mobile Internet Workshop, Roma, 2001. p. 34-42.

GROSSMAN, D. *New Terminology and Clarifications for Diffserv*. IETF RFC 3260, abr. 2002.

HEINANEN, J.; *et al*. *Assured Forwarding PHB Group*. IETF RFC 2597, jun. 1999.

ITU-T - INTERNATIONAL TELECOMMUNICATION UNION - TELECOMMUNICATION STANDARDIZATION SECTOR. *Digital subscriber signalling system No. 2 – Connection characteristics negotiation during call/connection establishment phase: Protocol Implementation Conformance Statement (PICS) proforma*. ITU-T Q.2962B. 2000.

JI, Ping; *et al*. *A Comparison of Hard-state and Soft-state Signaling Protocols*. SIGCOMM'03, Karlsruhe, Alemanha, 2003.

JOHNSON, D.; PERKINS, C.; ARKKO, J. *Mobility Support in IPv6*. IETF RFC 3775, jun. 2004.

MALINEN, Jari T.; PERKINS, Charles E. *Mobile IPv6 Regional Registrations*. IETF Internet Draft, mar. 2001.

MALKI, Karim El. *Low-Latency Handoffs in Mobile IPv4*. IETF RFC 4881, jun. 2007.

MANNER, Jukka; *et al. Evaluation of Mobility and QoS Interaction*. Computer Networks, Elsevier Science Publisher, vol. 38, fev. 2002. p. 137-163.

MCCANN, Pete; *et al. Transparent Hierarchical Mobility Agents (THEMA)*. IETF Internet Draft, mar. 1999.

MCCLOGHRIE, K.; *et al*. *Structure of Policy Provisioning Information (SPPI)*. IETF RFC 3159, ago. 2001.

MONTENEGRO, G. *Reverse Tunneling for Mobile IP*. IETF RFC 3024, jan. 2001.

MOORE, B. *Policy Core Information Model (PCIM) Extensions*. IETF RFC 3460, jan. 2003

MOORE. B.; *et al*. *Policy Core Information Model - Version 1 Specification*. IETF RFC 3060, fev. 2001.

NICHOLS, K.; *et al*. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. IETF RFC 2474, dez. 1998.

NOOR, Rafidah; EDWARDS, Christopher. *A Dynamic QoS Provisioning Model for Network Mobility*. The 7th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting, jun. 2006.

PANA, M.; *et al*. *Policy Core Extension Lightweight Directory Access Protocol Schema (PCELS)*. IETF RFC 4104, jun. 2005.

PASKALIS, Sarantis; *et al*. *An Efficient RSVP-Mobile IP Interworking Scheme*. Journal on Special Topics in Mobile Networking and Applications (MONET), jun. 2003.

PERKINS, C. *IP Encapsulation within IP*. IETF RFC 2003, out. 1996a

PERKINS, C. *IP Mobility Support for Ipv4*. IETF RFC 3344, ago. 2002.

PERKINS, C. *IP Mobility Support*. IETF RFC 2002, out. 1996b.

RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D. *The Addition of Explicit Congestion Notification (ECN) to IP*. IETF RFC 3168, set. 2001.

RAMJEE, R.; *et al*. *HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks*. Seventh International Conference on Network Protocols, Toronto, Canada, 1999. p. 283-292.

RAWLINS, D.; *et al*. *Framework Policy Information Base for Usage Feedback*. IETF RFC 3571, ago. 2003.

ROSEN, E.; VISWANATHAN, A.; CALLON, R. *Multiprotocol Label Switching Architecture*. IETF RFC 3031, jan. 2001.

ROSENBERG, J.; el al. *SIP: Session Initiation Protocol*. IETF RFC 3261, jun. 2002b.

ROSENBERG, J.; SCHULZRINNE, H. *Reliability of Provisional Responses in Session Initiation Protocol (SIP)*. IETF RFC 3262, jun. 2002a.

SAHITA, R.; *et al*. *Framework Policy Information Base*. IETF RFC 3318, mar. 2003.

SHELBY, Zach D.; *et al. Cellular IPv6*. IETF Internet Draft, nov. 2000.

SHEN, Charles Qi; *et al*. *An Interoperation Framework for Using RSVP in Mobile IPv6 Networks*. IETF Internet Draft, jul. 2001.

SHENKER, S.; PARTRIDGE, C.; GUERIN, R. *Specification of Guaranteed Quality of Service*. IETF RFC 2212, set. 1997a.

SHENKER, S.; WROCLAWSKI, J. *General Characterization Parameters for Integrated Service Network Elements*. IETF RFC 2215, set. 1997b.

SNIR, Y.; *et al*. *Policy Quality of Service (QoS) Information Model*. IETF RFC 3644, nov. 2003.

SOLIMAN, H.; *et al*. *Hierarchical Mobile IPv6 Mobility Management (HMIPv6)*. IETF RFC 4140, ago. 2005.

STATTENBERGER, Günther ; BRAUN, Torsten ; BRUNNER, Marcus. *A Platform-Independent API for Quality of Service Management.* IEEE Workshop on High Performance Switching and Routing, Dallas, Texas, EUA, 2001b.

STATTENBERGER, Günther ; BRAUN, Torsten. *QoS Provisioning for Mobile IP Users.* Conference on Applications and Services in Wireless Networks, ASW 2001, Paris, jul. 2001a.

STRASSNER, J.; *et al*. *Policy Core Lightweight Directory Access Protocol (LDAP) Schema*. IETF RFC 3703, fev. 2004.

TAHA, Abd-Elhamid; HASSANEIN, Hossam; MOUFTA, Hussein. *Extensions for Internet QoS Paradigms to Mobile IP*: a Survey. IEEE Communications Magazine, vol. 43, issue 5, mai. 2005. p. 132 - 139.

TALUKDAR, Anup Kumar; BADRINATH, B. R.; ACHARYA, Arup. *Integrated Services Packet Networks with Mobile Hosts:* Architecture and Performance. ACM Wireless Networks, vol. 5, issue 2, mar. 1999. p. 111-124.

TERZIS, A.; *et al*. *RSVP Operation Over IP Tunnels*. IETF RFC 2746, jan. 2000.

THE APACHE SOFTWARE FUNDATION. *Apache Axis 2*. Disponível em: <http://ws.apache.org/axis2/>. Acesso em 10 dez. 2007.

THOMAS, Michael. *Analysis of Mobile IP and RSVP Interactions*. IETF Internet Draft, out. 2002.

THOMSON, S.; NARTEN, T. *IPv6 Stateless Address Autoconfiguration.* IETF RFC 2462, dez. 1998.

TSENG, Chien-Chao; LEE, Gwo-Chuan; LIU, Ren-Shiou. *HMRSVP: A Hierarchical Model RSVP Protocol*. IEEE, 2003.

VASSILIOU, Vasos; PITSILLIDES, Andreas. *Supporting Mobility Events within a Hierarchical Mobile IP-over-MPLS Networks*. Journal Of Communications (JCM), vol. 2, nº. 2, mar. 2007. p. 61-70.

WESTERINEN, A.; *et al*. *Terminology for Policy-Based Management*. IETF RFC 3198, nov. 2001.

WIRESHARK Network Protocol Analyzer. Disponível em <http://www.wireshark.org/>. Acesso em 10 jan. 2008.

WORLD WIDE WEB CONSORTIUM (W3C). *XPointer Framework*. Disponível em : <http://www.w3.org/TR/xptr-framework/>. Acesso em : 10 dez. 2007.

WROCLAWSKI, J. *Specification of the Controlled-Load Network Element Service*. RFC 2211, set. 1997a.

WROCLAWSKI, J. *The Use of RSVP with IETF Integrated Services*. IETF RFC 2210, set. 1997b.

YAVATKAR, R.; PENDARAKIS, D.; GUERIN. R. *A Framework for Policy-based Admission Control*. IETF RFC 2753, jan. 2000.