Tomás Antônio Costa Badan

Uma Arquitetura de Mobilidade para Redes IP e sua Realização sobre o Protocolo MPLS

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Prof. Dr. Eleri Cardozo

Campinas, SP
2010

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Badan, Tomás Antônio Costa
B14a Uma arquitetura de mobilidade para redes IP e sua realização sobre o protocolo MPLS / Tomás Antônio Costa Badan. – Campinas, SP: [s.n.], 2010.

Orientador: Eleri Cardozo.
Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Internet sem fio. 2. Comunicações digitais. 3. Arquitetura de redes de computadores. 4. Linux. I. Cardozo, Eleri. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título

Título em Inglês: A mobility architecture for IP networks and its realization over MPLS protocol
Palavras-chave em Inglês: Wireless internet, Digital communications, Computer network architecture, Linux
Área de concentração: Engenharia de Computação
Titulação: Doutor em Engenharia Elétrica
Banca Examinadora: Marcos Rogério Salvador, Fábio Luciano Verdi, Leonardo de Souza Mendes, Mauricio Ferreira Magalhães
Data da defesa: 16/08/2010
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Tomás Antônio Costa Badan

Data da Defesa: 16 de agosto de 2010

Título da Tese: "Uma arquitetura de mobilidade para redes IP e sua realização sobre o protocolo MPLS"

Prof. Dr. Eleri Cardozo (Presidente): _____

Dr. Marcos Rogério Salvador: _____

Prof. Dr. Fábio Luciano Verdi: _____

Prof. Dr. Leonardo de Souza Mendes: _____

Prof. Dr. Maurício Ferreira Magalhães: _____

Resumo

A próxima geração da telefonia celular, 4G, será totalmente baseada no protocolo IP. Para o usuário final, a expectativa é estar constantemente conectado à esta rede, no qual a característica fundamental será a mobilidade transparente do dispositivo móvel, entre as várias subredes que compõem um domínio administrativo. Esta tese tem por objetivo propor uma solução para o problema da mobilidade transparente do dispositivo móvel em redes IP. Como consequência, duas contribuições são alcançadas. A primeira é a especificação de uma arquitetura que permita localizar e rastrear o dispositivo móvel em um domínio administrativo, que seja independente da tecnologia de túneis utilizada na camada de rede. A segunda é a proposição de um método de rastreamento do dispositivo móvel em redes MPLS, preservando as especificações do protocolo MPLS. É mostrado também como este método foi integrado com a arquitetura previamente definida. Por fim, é descrita como essa proposta de rastreamento em redes MPLS foi implementada sobre o sistema operacional Linux e os testes realizados para avaliar, tanto a implementação desta proposta, quanto a sua integração com essa arquitetura.

Palavras-chave: Internet sem Fio, Comunicações Digitais, Arquitetura de Rede de Computadores, Linux.

Abstract

The next generation of cellular telephony, 4G, is going to be totally based on the IP protocol. The end user expects to be constantly connected to this network, in which the key feature will be the seamless mobility of the mobile device among the various subnets within an administrative domain. This thesis has as objective to propose a solution to the problem of seamless mobility of the mobile device in IP networks. As such, two contributions are achieved. The first one is the specification of an architecture able to locate and track the mobile device inside an administrative domain, being independent of the tunnel technology used in the network layer. The second one is the proposition of a method to track the mobile device inside a MPLS networks, keeping intact the specifications of the MPLS protocol. It is also shown how this method was integrated with the previously defined architecture. Finally, it is described how the proposed method to track mobile devices inside a MPLS network was implemented on the Linux operating system, and the tests performed in order to assess both the implementation of this proposal and its integration with this architecture.

Keywords: Wireless Internet, Digital Communication, Computer Networks Architeture, Linux.

Agradecimentos

Ao meu amigo e orientador Prof. Eleri Cardozo, sou imensamente grato pela sua orientação sempre firme e determinada. Sou grato também pela oportunidade que me foi dada e pela paciência durante esta jornada.

À Ericsson do Brasil, sou grato pelo apoio financeiro deste trabalho.

Aos amigos da EEEC/UFG, sou grato pela licença concedida e por acreditar neste trabalho.

Aos meus amigos de doutorado Eduardo N. F. Zagari e Rodrigo C. M. do Prado, sou grato pelas discussões, incentivos e, principalmente, pela amizade estabelecida nesse período e pelo apoio em todos os momentos.

Aos meus amigos do LCA, sou grato pelas amizades e momentos de descontração.

Aos meus pais, irmão, irmãs, sobrinhos e sobrinhas, pelo apoio e por acreditar nesta jornada.

Aos meus filhos Guilherme e Thiago, sou imensamente grato por estarem sempre comigo, por esse amor quase que incondicional, o qual espero estar sempre retribuindo e por estarem sempre presentes, mesmo nos momentos mais difíceis, tornando-os sempre uma grande alegria.

Por fim e não menos importante, a minha mulher Márcia, sou imensamente grato pelo seu amor e dedicação dados a mim e, mesmo na minha solidão, por me apoiar em todos os momentos. Espero poder sempre lhe retribuir esse amor.

Aos meus pais Eudércio e Lucélia
Aos meus filhos Guilherme e Thiago
À minha mulher Márcia

Sumário

Lista de Figuras

Lista de Tabelas

Glossário

A

AC *Address Configuration*

ACP *Address Configuration Protocol*

AD *Administrative Domain*

AN *Access Network*

AP *Access Point*

API *Application Programming Interface*

AR *Access Router*

ARP *Address Resolution Protocol*

ATM *Asynchronous Transfer Mode*

B

BGP *Border Gateway Protocol*

BS *Base Stations*

BU *Binding Update*

C

CBR *Constant Bitrate*

CCoA *Co-Located Care-of-Address*

CDMA *Code Division Multiple Access*

CIDR *Classless Inter-domain Routing*

CIP *Cellular IP*

CLI *Command Line Interface*

CN *Correspondent Node*

CoA *Care-of-Address*

CoS *Class of Service*

CR-LDP *Constraint-Based Label Distribution Protocol*

D

DF *Don't Fragment*

DHCP *Dynamic Host Configuration Protocol*

DNS *Domain Name System*

DRR *Domain Root Router*

E

ESP *Encapsulating Security Payload*

F

FA *Foreign Agent*

FACoA *Foreign Agent Care of Address*

FCoA *Foreign Agent Care-of-Address*

FDA *Foreign Domain Agent*

FEC *Forwarding Equivalence Class*

FMIP *Mobile IPv6 Fast Handovers*

FN *Foreign Network*

G

GFA *Gateway Foreign Agent*

GMPLS *Generalized Multi-Protocol Label Switching*

GRE *Generic Routing Encapsulation*

GSM *Global System for Mobile Communications*

H

H-MPLS *Hierarchical Mobile MPLS*

HA *Home Agent*

HAP *Home Access Point*

HH *Handover Helper*

HIP *Host Identity Protocol*

HMIPv6 *Hierarchical Mobile IPv6*

HN *Home Network*

HoA *Home Address*

HSDPA *High Speed Downlink Packet Access*

HSP *Handover Signaling Protocol*

HTTP *HyperText Transfer Protocol*

HTTPS *Hypertext Transfer Protocol Secure*

I

I-LIB *Intermediate Label Information Base*

IAPP *Inter-Access Point Protoco*

ICMP *Internet Control Message Protocol*

IEEE *Institute of Electrical and Electronics Engineers*

IETF *Internet Engineering Task Force*

IP *Internet Protocol*

IPsec *Internet Protocol Security*

L

LCoA *on-link CoA*

LDP *Label Distribution Protocol*

LEMA *Label Edge Mobility Agent*

LER *Label Edge Router*

LIB *Label Information Base*

LMA *Local Mobility Anchor*

LSA *Link State Advertisement*

LSP *Label Switching Path*

LSR *Label Switch Router*

LU *Location Update*

M

MAC *Media Access Control*

MAG *Mobile Access Gateway*

MAN *Metropolitan Area Network*

MAP *Mobility Anchor Point*

MAR *Mobility Aware Router*

MIP *Mobile IP*

MIP-RR *Mobile IPv4 Regional Registration*

MIPv6 *Mobile IP version 6*

MM-MPLS *Micro Mobile MPLS*

MN *Mobile Node*

MPA *Mobility Plane Architecture*

MPLS *MultiProtocol Label Switching*

MR *Mobile Routing*

MRP *Mobile Routing Protocol*

MTU *Maximum Transmission Unit*

N

NAP *new FAP*

NAR *New Access Router*

NAT *Network Address Translator*

NCoA *new CoA*

NHLFE *Next Hop Label Forwarding Entry*

O

OO *Object Oriented*

OSPF *Open Shortest Path First*

P

P2MP *Point-to-Multipoint*

P2P *Point-to-Point*

PAP *Previous FAP*

PAR *Previous Access Router*

PCoA *Previous CoA*

PDA *Personal Digital Assistant*

PMIP *Proxy Mobile IPv6*

Q

QNM *Queueing Network Model*

QoS *Quality of Service*

R

RADIUS *Remote Authentication Dial In User Service*

RBT *Red-Black Tree*

RCoA *Regional CoA*

RSVP-TE *Resource ReserVation Protocol-Traffic Engineering*

S

SA *Security Associations*

SAR *Serving Access Router*

SIP *Session Initiation Protocol*

SNMP *Simple Network Management Protocol*

SO Sistema Operacional

SPI *Security Parameters Index*

SSL *Secure Socket Layer*

T

TCP *Transmission Control Protocol*

TLS *Transport Layer Security*

TM *Tunnel Management*

TMP *Tunnel Management Protocol*

U

UDP *User Datagram Protocol*

UM *User Mode*

UML *Unified Modeling Language*

V

VoIP *Voice over Internet Protocol*

VPN *Virtual Private Network*

W

WCDMA *Wideband Code Division Multiple Access*

WiMAX *Worldwide Interoperability for Microwave Access*

WLAN *Wireless Local Area Network*

WPA *Wi-Fi Protected Access*

Trabalhos Publicados Pelo Autor

1. Badan, T. A. C.; Prado, R. C. M.; Zagari, E. N. F.; Cardozo, E.; Magalhães, M. F. **Uma Implementação de MPLS para Redes Linux**. *XIX Simpósio Brasileiro de Redes de Computadores*, Florianópolis. Anais do SBRC´2001, 2001. p. 743-758.

2. Zagari, E. N. F.; Badan, T. A. C.; Prado, R. C. M.; Cardozo, E.; Magalhães, M. F. **Uma Plataforma para Engenharia de Tráfego com Qualidade de Serviço em Redes MPLS**. *Simpósio Brasileiro de Redes de Computadores*, Búzios - RJ. Anais do XX Simpósio Brasileiro de Redes de Computadores - SBRC, 2002.

3. Zagari, E. N. F.; Prado, R. C. M.; Cardozo, E.; Magalhães, M. F.; Badan, T. A. C.; Carrilho, J. A.; Pinto, R. P.; Moraes, D. H.; Barboza, D. H.; Johnson, T. M. S. M.; Westberg, L. **MPA: a Network-Centric Architecture for Micro-Mobility Support in MPLS Networks**. *IEEE/ACM Sixth Annual Conference on Communication Networks and Services Research (CNSR2008)*, Halifax, Canada. Proccedings of the CNSR2008, 2008.

4. Prado, R. C. M.; Zagari, E. N. F.; Cardozo, E.; Westberg, L.; Magalhães, M. F.; Badan, T. A. C.; Carrilho, A. C.; Pinto, R. P.; Berenguel, A.; Barboza, D. H.; Moraes, D. H.; Johnson, T. M. S. M. **A Reference Architecture for Micro-mobility Support in IP Networks**. *IEEE Symposium on Computers and Communications (ISCC'08)*, Marrakech, Marocco. Proceedings of the ISCC 2008, 2008.

5. Johnson, T.; Zagari, E.; Prado, R.; Badan, T.; Cardozo, E.; Westberg, L. **Performance Analysis of a New Architecture for Mobility Support IP Networks**. *IWCMC 2008 Mobile Computing Symposium*, Creta, Grécia. Proccedings of the IWCMC 2008, 2008.

6. Johnson, T.; Prado, R.; Zagari, E.; Badan, T.; Cardozo, E.; Westberg, L. **Considerations on Performance Evaluation of Micro-Mobility Architectures for IP Networks**. *IEEE PIMRC 2008 Mobile and Wireless Networks Track*, Cannes, França. Proccedings of the PIMRC 2008, 2008.

7. Badan, T.; Zagari, E. N. F.; Prado, R.; Cardozo, E.; Magalhães, M. F.; Carrilho, J.; Pinto, R.; Berenguel, A.; Moraes, D.; Johnson, T.; Westberg, L. **A Network Architecture for Providing Micro-mobility in MPLS/GMPLS Networks**. *WCNC 2009 IEEE Wireless Communications and Networking Conference*, Budapeste, Hungria. Proccendings of the WCNC 2009, 2009.

8. Zagari, E. N. F.; Prado, R.; Badan, T.; Cardozo, E.; Magalhães, M. F.; Carrilho, J.; Berenguel, A.; Moraes, D.; Dolphine, T.; Johnson, T.; Westberg, L. **Design and Implementation of a Network-Centric Micro-Mobility Architecture**. *WCNC 2009 IEEE Wireless Communications and Networking Conference*, Budapeste, Hungria. Proccendings of the WCNC 2009, 2009.

9. Johnson, T.; Prado, R.; Zagari, E. N. F.; Badan, T.; Cardozo, E.; Westberg, L. **Performance Evaluation of Reactive and Proactive Handover Schemes for IP Micromobility Networks**. *WCNC 2009 IEEE Wireless Communications and Networking Conference*, Budapeste, Hungria. Proccendings of the WCNC 2009, 2009.

10. Johnson, T.; Cardozo, E.; Prado, R.; Zagari, E. N. F.; Badan, T. **Mobility in IP Networks: From Link Layer to Application Layer Protocols and Architectures**. *Radio Communications*, Alessandro Bazzi (ed.), Intech, 2010.

Capítulo 1

Introdução

Sem dúvida vivemos um momento ímpar na história das comunicações. Observamos uma difusão em massa de aparelhos de comunicação sem fio, tais como: celulares, com um poder de processamento crescente, *notebooks* extremamente leves e versáteis, *tablets*, dentre outros dispositivos pouco populares.

O crescimento explosivo da utilização desses dispositivos pode ser explicado pelo amadurecimento de algumas tecnologias. Dentre elas, podemos citar: a tecnologia de fabricação de dispositivos de estado sólido, a tecnologia de acesso a redes sem fios e a tecnologia de redes de dados, em particular, a Internet.

A tecnologia de fabricação de dispositivos de estado sólido contribuiu com a miniaturização dos componentes, associada com o crescente poder computacional dos *chips* produzidos. Isso, em conjunto com um aumento constante nas vendas desses dispositivos, desonerou o custo final desses aparelhos. Com a miniaturização, tornou-se possível o surgimento de aparelhos de alta portabilidade, com baixa potência consumida e alto poder computacional.

A baixa potência consumida, pelos aparelhos, é interessante pois permite abastecê-los com uma fonte de energia portátil, ou seja, de baixo peso, o que se traduz, na vida cotidiana, em uma mobilidade física.

Em perfeita simbiose com a tecnologia de estado sólido está a tecnologia de redes sem fios. Impulsionada pela indústria da telefonia móvel e pela Internet, a tecnologia de redes sem fios tem invadido os aparelhos portáteis, oferecendo conectividade a praticamente todos os novos *gadgets*[1] que são lançados.

Podemos analisar a evolução desses dispositivos sob essas duas perspectivas. Sob a ótica da telefonia móvel, a cada geração tem aumentado a capacidade de transmissão de dados. Na geração chamada de 3G é possível usar não somente os serviços de voz, tradicionais da telefonia celular, mas também o acesso à Internet, graças à capacidade de transmitir dados

[1] Um *gadget* é um pequeno objeto tecnológico que possui uma função em particular, mas é frequentemente associado como sendo uma novidade.

a uma taxa média de pelo menos 100 kbit/s, com taxas máximas na faixa de mega bits por segundo[2] [1]. Isso é o suficiente para enviar e receber imagens digitais e trocar dados a altas velocidades. A próxima geração, chamada de 4G, promete velocidades ainda maiores, com uma integração total entre a tecnologia celular e a Internet. Na verdade, é esperado que o padrão de transmissão, desta nova geração, seja baseado totalmente no protocolo IP (*Internet Protocol*) [2].

Sob a ótica da Internet, o IEEE (*Institute of Electrical and Electronics Engineers*) tem lançados padrões de acesso às redes sem fio sob dois emblemas principais, o Wi-Fi que se refere à família relacionada às especificações do grupo IEEE 802.11 e o WiMAX (*Worldwide Interoperability for Microwave Access*) que se refere à série IEEE 802.16, o qual padroniza o acesso à rede sem fio em faixa larga.

Enquanto o Wi-Fi tem o seu nicho nas rede WLAN (*Wireless Local Area Network*), com características similares às encontradas por uma rede física do tipo *ethernet*, o WiMAX é mais apropriado para redes de banda larga, ou seja, do tipo MAN (*Metropolitan Area Network*), cuja característica chave é ser uma tecnologia voltada à conexão, que provê um suporte para qualidade de serviços – QoS (*Quality of Service*). O WiMAX é a tecnologia candidata para substituir as atuais GSM (*Global System for Mobile Communications*) e CDMA (*Code Division Multiple Access*) para a futura geração de redes celulares.

Por fim, a Internet é o padrão *de facto* para a rede de dados. Desde a sua abertura comercial, na década de 90, a Internet tem crescido de forma exponencial, agregando novos usuários continuamente. Alguns dos motivos desse sucesso, principalmente nesses últimos anos, são: o caráter descentralizado da rede, ofertas massivas de informações, vídeos sob demanda, mensagens instantâneas e as redes sociais.

Por outro lado, a Internet surgiu no final da década de 60, quando os atuais dispositivos de acesso ao meio eram inimagináveis. Com isso, várias das decisões de projeto, tomadas naquela época, tornaram-se um problema para a atualidade. Alguns desses problemas são: exaustão do endereçamento IP, ausência de diferenciação do tráfego transportado, roteamento *hop* a *hop* e alto acoplamento entre o endereço IP atribuído a um dispositivo e a sua localização física.

Para resolver esses e outros problemas, foi proposta a criação de uma nova versão para o protocolo IP, chamado de IPv6 [3]. Devido à demora em sua proposição e à necessidade de resolver o problema da exaustão dos endereços IP, soluções alternativas foram postas em prática, para aliviar este último problema. Assim, foram adotadas as seguintes soluções: a adoção de endereços sem vínculos com classe, ou seja, a CIDR (*Classless Inter-domain Routing*) [4] e a tradução de endereços NAT (*Network Address Translator*) [5]. Isso postergou a adoção do IPv6 por um tempo indeterminado.

[2] Na tecnologia WCDMA/HSDPA (*Wideband Code Division Multiple Access/High Speed Downlink Packet Access*) é possível usar uma taxa média entre 400-700 kbit/s.

O roteamento *hop* a *hop* é um grande empecilho para a otimização de recurso de uma rede de dados. Dentro desse contexto, associado à necessidade de se ter uma infraestrutura que permita a união das duas grandes redes existentes, a de telefonia e a da Internet, surge o protocolo MPLS (*MultiProtocol Label Switching*) [6]. O MPLS é uma solução elegante que atende aos requisitos de escalabilidade, qualidade de serviço e engenharia de tráfego. Em sua concepção, o protocolo MPLS foi planejado para ser aplicado no *backbone* de uma rede IP, mas nada impede que seja utilizado, também, em subredes IP, as quais dão acesso aos usuários.

Ao dissociar a função de classificação de pacotes da função de encaminhamento, nos roteadores da rede, por meio da atribuição de um rótulo que descreve a conexão criada entre o LER (*Label Edge Router*) de entrada e o LER de saída, o MPLS permite que vários pacotes, de protocolos distintos, sejam encaminhados sobre a mesma infraestrutura. Além disso, o MPLS é facilmente implementável sobre várias tecnologias de camada física e de enlace, permitindo um *framework* único de gerência. Portanto, é esperado que o protocolo MPLS exerça um grande papel na próxima geração de celulares, a 4G, quando a rede de transporte deverá ser totalmente baseada no protocolo IP.

Para concluir, vemos que a força motriz destas evoluções, ou podemos até mesmo chamar de revoluções, deve-se à necessidade de comunicação e aquisição de informações do ser humano, a qual é inesgotável. Apenas trocar informações audíveis, ou sentar-se à frente de um computador de mesa, em busca de informações, já não são suficientes. É preciso integrar todos esses serviços, ou seja, o de voz e o de dados, em uma única infraestrutura, que permita a mobilidade plena do usuário.

1.1 Objetivos

Os *gadgets* que surgem no mercado indicam que a nova tendência é a mobilidade total do usuário, em que os serviços de voz e dados deverão estar integrados em uma única infraestrutura. O conceito de mobilidade plena nos diz que a mesma experiência vivida por um usuário ao usar os serviços de voz e dados na ausência de mobilidade, deverá ser a mesma quando ele estiver em movimento.

Assim, dentro deste cenário heterogêneo, em que a tendência é ter uma infraestrutura de transmissão de dados totalmente baseada no protocolo IP e cujo papel do protocolo MPLS será importante, é necessário restringi-lo um pouco, para definir os objetivos desta tese. Vamos assumir que a rede de dados será baseada no protocolo IP, sob a gerência de um domínio administrativo qualquer. Isso é importante pois deixa a rede mais controlável, no sentido de que é possível realizar suposições sobre quais são os protocolos de rede que estão, ou estarão, em operação. Quanto ao nó móvel – MN (*Mobile Node*), é assumido que ele possua, além da capacidade de se conectar à Internet através de uma interface de rede aérea, uma

implementação do básico do protocolo IP, normalmente encontrada nesse tipo de dispositivo.

Neste contexto, os objetivos desta tese são dois: (a) contribuir com uma arquitetura que permita localizar e rastrear um MN em redes IP, mantendo a migração transparente ao usuário e independente da tecnologia de encaminhamento utilizada; (b) mapear essa arquitetura sobre o protocolo MPLS, utilizando-se apenas das facilidades já previstas em suas especificações. Isso permite uma adaptação natural da solução com a evolução do protocolo e futuros serviços.

1.2 Organização da Tese

Para alcançar e demonstrar os objetivos propostos, esta tese foi dividida em seis capítulos, estruturados como se segue:

- Capítulo 2, intitulado Paradigma de Mobilidade e Trabalhos Relacionados, tem por objetivo apresentar o paradigma de mobilidade e as várias alternativas de tratá-lo, propostas pela literatura;
- Capítulo 3, intitulado Arquitetura do Plano de Mobilidade, cujo acrônimo é MPA – (*Mobility Plane Architecture*), tem por objetivo apresentar a nossa arquitetura de mobilidade, o qual permite rastrear o MN independente da tecnologia de encaminhamento utilizada e sem que o usuário final perceba, ou participe, do processo de migração;
- Capítulo 4, intitulado Integração da Arquitetura MPA com o Protocolo MPLS , tem por objetivo mostrar como foi realizada esta integração, preservando as características primordiais do protocolo MPLS;
- Capítulo 5, intitulado Implementação do Protocolo MPLS e Validação, tem por objetivo mostrar como foi realizada a implementação deste protocolo sobre o sistema operacional Linux, adotando as práticas de engenharia de *software* aplicadas em conjunto com o paradigma de orientação a objetos. Tem por objetivo, também, apresentar os testes realizados, tanto sobre a implementação do protocolo MPLS, quanto sobre a integração dele com a arquitetura MPA;
- Capítulo 6, intitulado Considerações Finais e Extensões ao Trabalho, tem por objetivo apresentar as conclusões obtidas com o resultado deste trabalho de tese, comentar os trabalhos em andamento sobre a arquitetura MPA e propor algumas extensões como trabalhos futuros.

Capítulo 2

Paradigma de Mobilidade e Trabalhos Relacionados

Atualmente, vemos a explosão de dispositivos móveis individuais do tipo PDA (*Personal Digital Assistant*), *notebooks* e *video games* portáteis. Uma das características mais marcantes desses dispositivos é a crescente necessidade de se manter uma constante conectividade com a Internet através de enlaces aéreos, ou seja, de rádio frequência.

Este novo cenário exige uma nova configuração e organização dos elementos de rede e, por questões de otimização de recursos, o tamanho da célula empregada tende a diminuir em área de abrangência.

Tal tendência de diminuição do tamanho de célula, a qual a literatura normalmente chama de pico células, induz a uma crescente taxa na frequência de *handoffs*[1] que os dispositivos móveis sofrem quando estão em movimento.

Do ponto de vista da rede, o aumento na frequência de *handoffs* implica que frequentes trocas de sinalizações entre a rede visitada e a rede *home* irão ocorrer. Isso implica também em um aumento na latência no estabelecimento de novas conexões; no *jitter* percebido pelas aplicações de tempo real e na quantidade de pacotes perdidos.

Esses efeitos são causados devido ao modo como o protocolo proposto pelo IETF (*Internet Engineering Task Force*), para resolver o problema da mobilidade, foi concebido e opera, no qual otimizações locais não foram consideradas. Aproveitando dessa lacuna, várias propostas tentam otimizar esse protocolo, explorando formas de manter as informações de mobilidade as mais locais ou regionais possíveis.

O restante deste capítulo irá discutir essas propostas e dará ênfase ao protocolo MPLS como solução de mobilidade. As discussões serão divididas em quatro partes. A primeira, apresentada na Seção 2.1, irá discutir o paradigma de mobilidade e seus macro componentes.

[1]Termo técnico utilizado para designar o evento de mudar de uma célula para outra e a consequente reassociação com a nova antena.

A Seção 2.2 apresentará uma taxionomia dos principais grupos utilizados em mobilidade e as definições dos principais termos utilizados em sua classificação. A Seção 2.3 apresentará uma revisão bibliográfica das principais propostas de mobilidade. Serão apresentadas soluções pertinentes a cada camada que compõe a pilha de protocolos TCP/IP. Por fim, a seção 2.4 apresentará as considerações sobre este capítulo.

2.1 O Paradigma de Mobilidade

A expressão Paradigma de Mobilidade é uma definição bastante ampla e necessita de uma maior especificidade para podermos aplicá-la com mais rigor.

O termo paradigma é uma palavra de origem grega, bastante utilizada pela comunidade científica e que expressa a seguinte ideia, segundo o *The Free Dictionary* [7]: "Um conjunto de suposições, conceitos, valores e práticas, os quais constituem um modo de ver a realidade por uma comunidade e que lhe são compartilhadas, especialmente em um meio científico".

Por outro lado, o termo mobilidade possui também um sentido vago e pode ser aplicado em vários contextos. O *The Free Dictionary* define mobilidade por: "A qualidade ou estado de ser móvel".

Mesmo especializando-a para o contexto de redes de computadores, a ideia de mobilidade ainda continua vaga. Podemos desejar a mobilidade no nível de uma aplicação, onde a execução de um aplicativo que tenha se iniciado em um elemento de rede, possa ser continuada em um outro elemento de rede qualquer. Como exemplo, podemos citar uma conversa telefônica que se inicia em um terminal telefônico IP e deve ser terminada em um *notebook*. Esse seria um exemplo de uma mobilidade virtual, em que não há movimento físico entre os elementos de rede.

No outro extremo, podemos citar a mobilidade física[2], onde um elemento de rede pode se mover sobre uma área geográfica. Embora seja possível realizar este tipo de mobilidade utilizando-se elementos legados a um conector *ethernet*, vamos considerar aqui apenas os elementos que possuam uma conectividade através de uma interface aérea, chamada também de interface sem fio ou *wireless*.

A mobilidade realizada pelos elementos de rede sem fio pode ser vista sob dois aspectos. O primeiro é referente à mobilidade entre dois pontos de acesso dentro de uma mesma subrede, a qual é resolvida somente pelas camadas física e de enlace[3] e não envolvem a alteração do endereço IP do elemento móvel. Os pontos de acesso podem ser implementados com a mesma tecnologia de acesso ao meio, ou com tecnologias diferentes. Assim, após a reassociação do

[2]Entende-se por mobilidade física a capacidade de se mover fisicamente um dispositivo, ou um elemente de rede, de um ponto para outro.

[3]Os termos camadas e suas funções estão em conformidade com as definições de camadas do modelo de referência OSI-ISO.

elemento móvel com o ponto de acesso, a conectividade com a rede está concluída.

O segundo aspecto é referente à mobilidade entre dois pontos de acesso que estão em rede, ou subredes, IPs distintas. Isso acarreta em mudança no endereço IP do elemento móvel e, mesmo após a reassociação desse elemento com o ponto de acesso, a conectividade não está garantida enquanto o problema de atribuição de endereço IP não for concluído. É este tipo de mobilidade que estamos interessados nesta tese. Note que a situação descrita acima permanece inalterada, mesmo que os pontos de acesso possuam a mesma tecnologia de acesso ao meio, ou tecnologias diferentes.

A princípio, um elemento móvel pode ser tanto um elemento de rede do tipo PDAs, *notebooks*, ou similares, quanto um roteador móvel, o qual agrega uma subrede IP.

Conforme já elucidado por vários autores, como por exemplo Chiussi *et al* [8] e Akyildiz *et al* [9], a próxima geração de redes sem fios será totalmente baseada em redes IP. Assim, vamos restringir o conceito mais generalista do paradigma de mobilidade para uma conceituação mais específica e que chamaremos de paradigma de mobilidade de elementos de rede em uma rede Internet.

Existem dois objetivos bem específicos deste paradigma. O primeiro é manter uma sessão aberta, enquanto um nó migra de uma subrede, ou domínio, para um outro qualquer. A noção de sessão remete à ideia de uma conexão telefônica, da qual um fluxo de dados flui entre dois aplicativos, em máquinas distintas. A noção de manter uma sessão aberta remete à ideia de que este fluxo de dados não deve ser interrompido e nem mesmo reiniciado quando um nó se move entre redes diferentes.

O segundo é manter a experiência do usuário satisfatória, com relação à mobilidade, sem que ele perceba que o processo lógico de mobilidade esteja acontecendo. Isso implica em minimizar tanto as perdas de pacotes, quanto o atraso implícito ao processo de *handoff*.

A fim de normatizar uma linguagem comum para este paradigma, alguns conceitos e termos necessitam ser definidos [10]:

> **Agente de Mobilidade de Origem** – abreviado por HA (*Home Agent*), é um roteador pertencente à rede de origem do MN, com o qual o MN registrou o seu atual endereço da rede visitada. É papel desse agente interceptar os pacotes destinados ao MN, encapsulá-los e reenviá-los ao atual endereço do MN.
>
> **Agente de Mobilidade Estrangeiro** [4] – abreviado por FA (*Foreign Agent*), é um roteador pertencente à rede visitada pelo MN. Normalmente, é um elemento opcional e quando presente, participa do processo de encaminhamento/recebimento dos pacotes destinados/provenientes ao MN.

[4] Note que os termos estrangeiro e visitado estarão sendo usados de forma intercambiável e remetem à ideia de que não é a rede de origem do MN.

Domínio Administrativo – abreviado por AD (*Administrative Domain*), é a coleção de redes/subredes IP sob o mesmo controle administrativo.

Endereço de Origem – abreviado por HoA (*Home Address*), é o endereço IP atribuído ao MN e utilizado como seu endereço permanente.

Endereço da Rede Visitada – abreviado por CoA (*Care-of-Address*), é um endereço IP associado ao MN, quando ele está em uma rede visitada.

Handoff – também chamado de *handover*, é o processo pelo qual um MN muda o seu ponto de associação com a rede, ou quando um tentativa de mudança é realizada.

Mensagens de Atualização de Ligação – abreviada por BU (*Binding Update*), é uma mensagem que indica qual deve ser a atual ligação entre a mobilidade corrente do MN e seu CoA.

Nó Correspondente – abreviado por CN (*Correspondent Node*), é um nó IP que irá estabelecer, ou já possui, uma conexão de transporte com o MN. Ele reside, normalmente, em alguma rede IP diferente tanto da rede de origem do MN quanto da rede visitada;

Nó Móvel – abreviado por MN, é um nó IP, que é capaz de mudar o seu ponto de associação com a rede de dados. Pode tanto ser um *host* móvel que não possui as funções de encaminhamento de pacotes, quanto um roteador móvel que possui as funções de encaminhamento de pacotes.

Ponto de Acesso – abreviado por AP (*Access Point*), é um dispositivo de camada 2, o qual é conectado a um ou mais roteadores de acesso e oferece uma conexão sem fio para os MNs. São também chamados de estações bases – BS (*Base Stations*).

Rede de Acesso – abreviada por AN (*Access Network*), é uma rede IP que possui um ou mais roteadores de acesso.

Rede de Origem – abreviada por HN (*Home Network*), é a rede de origem do MN, o qual possui um contrato de conectividade com a rede visitada.

Rede Visitada – abreviada por FN (*Foreign Network*), é a rede na qual o MN está conectado temporariamente.

Roteador de Acesso – abreviado por AR – (*Access Router*), é o roteador que reside na borda de uma rede de acesso e conecta um ou mais APs. Um AR oferece uma conectividade IP para os MNs. Ele age como um roteador padrão para os MNs que estão correntemente sendo servidos. Os pontos de acesso podem ser de diferentes tecnologias.

A essência deste paradigma reside no conceito de gerência de mobilidade [9]. Ele é definido por um conjunto de quatro componentes principais, com funcionalidades bem definidas, mas que, normalmente, devem interagir entre si para se obter os objetivos iniciais. Esses componentes são: gerência de localização, gerência de *handoff* , gerência de endereços e sinalização.

A gerência de localização tem por objetivo habilitar o sistema para rastrear a localização do nó móvel entre comunicações sucessivas. Ele é composto de dois elementos: o registro/atualização de localização e o encaminhamento de pacotes.

O registro/atualização de localização é uma função realizada pelo MN, ou por um elemento de rede em seu nome, o qual informa o sistema, de tempos em tempos, sobre a sua atual localização. Essa função permite que o sistema seja atualizado, em seu banco de dados de localização, com os dados mais recentes do MN. Normalmente, a diferença entre registro e atualização é que a primeira é, usualmente, realizada quando o MN sai de sua rede de origem e ingressa em uma rede visitada, enquanto a segunda é realizada quando o MN migra entre redes visitadas.

A função encaminhamento de pacotes permite ao sistema determinar a atual localização do MN, através da consulta ao banco de dados de localização, quando uma comunicação com ele é iniciada. Normalmente é composta de dois passos: determinar qual banco de dados de localização está servindo o MN e qual é a subrede/célula que o MN está visitando.

Aqui, pode se inserir o conceito de *paging*, o qual, segundo Manner e Kojo [10], é:

> Um procedimento iniciado pela rede de acesso, para retirar um MN de um estado dormente e levá-lo para um estado ativo. Como resultado de um *paging*, o MN estabelece um SAR (*Serving Access Router*)[5] e a rota IP é estabelecida. (Tradução do autor)

Quando um MN está em um estado dormente, ele restringe a sua habilidade de receber o tráfego IP normal, ao reduzir o seu monitoramento dos canais de rádio. A rede de acesso conhece a área de *paging* do MN, mas o MN não possui um roteador de acesso associado. Assim, os pacotes não podem ser entregues a ele enquanto a rede de acesso não iniciar o *paging*. Normalmente, é chamado também de estado inativo.

Além de definirem o que é um estado dormente, Manner e Kojo [10] definem também o que é uma área de *paging*, ou seja:

> Uma parte da rede de acesso, tipicamente contendo um número de ARs/APs, os quais correspondem a uma certa área geográfica. A rede de acesso mantem e atualiza uma lista de todos os MNs dormentes que estão presentes nesta área. Se o MN está dentro de uma área de cobertura, ele será capaz de receber mensagens de *paging*, que são enviadas dentro de uma área de *paging*. (Tradução do autor)

[5] O SAR é o roteador que está atualmente oferecendo serviços de conectividade ao MN. Também é chamado de roteador de acesso.

A gerência de *handoff* é o processo pelo qual o MN mantem a sua conexão ativa, enquanto ele se move de um ponto de acesso para um outro qualquer.

Aqui, temos que ter o cuidado de restringir o escopo da gerência de *handoff*, a qual pode ser de dois tipos: o *handoff* que é tratado apenas pelas camadas físicas e de enlace, chamado também de *handoff* de camada 2, e o *handoff* que irá provocar mudanças no endereço de rede associado com o MN, o qual envolve a camada 3. O *handoff* de camada 2 está fora do escopo de discussão deste trabalho, pois, normalmente, é transparente para os roteadores de acesso. Por outro lado, o *handoff* de camada 3 é o de interesse para esta tese.

Duas métricas são de interesse quando se trata de *handover*. A primeira delas é a latência do *handover* que, conforme apontado por Manner e Kojo [10], é dada pela diferença entre o tempo em que o MN é capaz de enviar/receber o último pacote para um dado roteador de acesso e o tempo em que o MN é capaz de enviar/receber um pacote de seu novo roteador de acesso. A segunda é a perda de pacotes que correspondem aos pacotes que estão sendo enviados ao MN enquanto ele não está associado a nenhuma estação base e não podem ser entregues.

Dependendo de qual métrica é otimizada, o *handover* pode ser classificado em três categorias. A primeira se refere ao *handover* suave (*smooth handover*), no qual algumas propostas priorizam minimizar a perda de pacotes em detrimento aos atrasos de encaminhamento de pacotes.

A segunda se refere ao *handover* rápido (*fast handover*), no qual algumas propostas priorizam minimizar a latência do *handover*, sem um interesse explícito na perda de pacotes.

E, por último, o *handover* transparente (*seamless handover*), cuja premissa é de que outros protocolos, aplicações ou usuários finais, não devem perceber mudanças na segurança, capacidade ou qualidade dos serviços associados.

Em geral, essas métricas podem ser melhoradas quando existe uma realimentação entre as informações provenientes da camada 2 para as camadas superiores. A mais interessante delas é o conceito de *trigger*, que passa informações da camada de enlace para as camadas superiores, informando sobre os detalhes dos eventos envolvidos em um *handover*.

A gerência de endereços tem por objetivo permitir que o sistema administre os vários endereços necessários para realizar a mobilidade. Embora a forma como eles são atribuídos aos elementos seja ortogonal ao paradigma de mobilidade, isto é, se os endereços são obtidos de forma manual ou automática, através de um protocolo de distribuição de endereço, como o DHCP (*Dynamic Host Configuration Protocol*), a responsabilidade de quem deve ser a atribuição e como eles devem ser gerenciados, é de responsabilidade deste paradigma.

De um modo geral, é consenso que o MN seja capaz de reter dois ou mais endereços IP, simultaneamente. Normalmente, o primeiro se refere ao endereço IP de sua rede de origem e é considerado estável, ou permanente. O segundo é, normalmente, um endereço IP temporário

e serve para localizar o MN na rede visitada e manter a coerência topológica da rede. Quando o MN é incapaz de reter um segundo endereço IP, um agente de mobilidade, nesse caso o FA, age em seu favor no processo de registro e encaminhamento de pacotes e entrega as mensagens destinadas ao MN, via mecanismos de camada de enlace.

Por fim, a sinalização é responsável pela sincronização e distribuição de tarefas entre os vários elementos de mobilidade pertinentes ao sistema. Ela coordena quando e como devem ser as ações perante a ocorrência de alguns eventos específicos do sistema, como por exemplo, a ocorrência de um *handover*. Em geral, ela pode ou não necessitar da participação do MN neste processo. Sistemas que necessitam da participação do MN são ditos centrados no MN, enquanto que os que não necessitam da sua participação são ditos centrados na rede.

Devemos considerar ainda que toda essa dinâmica pode ser aplicada em dois escopos bem distintos. O primeiro é considerando um único domínio administrativo e normalmente recai no conceito de micromobilidade. O segundo é considerando vários domínios e normalmente recai no conceito de macromobilidade. Um terceiro cenário possível é considerar uma mescla dos dois anteriores mas que, em última instância, pode ser analisado de forma separada.

2.2 Parâmetros de Classificação em Mobilidade

Existem diversas formas em que se pode classificar uma proposta de mobilidade. Afim de se homogeneizar e obter um conjunto de termos comuns entre elas, iremos aplicar uma taxionomia baseada em quatro eixos diferentes: alcance de mobilidade, encaminhamento de pacotes de mobilidade, escopo da sinalização de mobilidade e camada de rede onde ocorre a mobilidade.

2.2.1 Alcance de Mobilidade

O alcance de mobilidade se refere como um nó móvel, em visita a uma rede estrangeira, se relaciona com ela e, também, como essa rede se relaciona com a rede de origem do nó móvel. O alcance de mobilidade pode ser dividido em dois grupos, conforme definido por Manner e Kojo [10] e Kempf [11].

Micromobilidade

Micromobilidade, também citado na literatura como mobilidade local, é definida como a mobilidade dentro de uma pequena área lógica, normalmente, dentro de um domínio IP. Note-se que, embora a rede lógica possa ser restrita, a rede geográfica pode ser extensa. Os protocolos de micromobilidade exploram a localidade do movimento ao confinar as mudanças relacionadas ao movimento e à sinalização do MN a essa rede de acesso.

Segundo Chiussi *et al* [8], para prover uma mobilidade de modo transparente ao usuário (*seamless mode*), os protocolos de micromobilidade devem satisfazer a alguns requisitos:

Fast Handover – resolve o problema do re-registro frequente entre o MN e o HA. Isso alivia o excesso de mensagens de sinalização, a latência do *handoff* e a perda de pacotes inerente ao processo, dentro de um domínio administrativo.

Projeto escalável – permite uma arquitetura flexível e distribuída da mobilidade local. Flexibilidade provê aos MNs a habilidade de escolher um ou mais agentes de mobilidade dentro de um conjunto de agentes disponíveis, evitando assim, os gargalos no sistema. Uma arquitetura distribuída refere-se à capacidade de espalhar na rede as informações de encaminhamento dos pacotes, ou seja, nem todos os agentes de mobilidade precisam conhecer as informações de encaminhamento de MNs que não estão sob sua responsabilidade.

Capacidade de QoS – provê os serviços de QoS para a micromobilidade.

Implementação gradual – permite uma evolução gradual da cobertura de micromobilidade. Isso implica na coexistência dessas funcionalidades, com nós que não as implementam.

Macromobilidade

Macromobilidade, também citada na literatura como mobilidade global, é definida como a mobilidade sobre uma grande área lógica, normalmente entre dois ou mais domínios IP. Um protocolo de macromobilidade usualmente altera o endereço *unicast* do nó móvel e o encaminhamento de pacotes fim a fim, quando a migração causa uma mudança de topologia, a fim de manter a continuidade da sessão estabelecida para as aplicações TCP (*Transmission Control Protocol*).

Como características dessas redes, podemos citar: a heterogeneidade do sistema, com relação às tecnologias de acesso ao meio, roteadores/*switches* e serviços oferecidos; e grandes atrasos nas mensagens de controle.

2.2.2 Encaminhamento de Pacotes de Mobilidade

O encaminhamento de pacotes de mobilidade define como as informações pertinentes à localidade do nó móvel são armazenadas e mantidas na rede e como os pacotes são encaminhados desde a rede de origem até a rede de destino. O encaminhamento de pacotes de mobilidade pode ser dividido em dois esquemas: os baseados no roteamento e os baseados no tunelamento.

Esquemas Baseados no Roteamento

É o esquema que explora a robustez do encaminhamento convencional do protocolo IP. Um banco de dados de localização do nó móvel é criado e mantido dentro de um domínio de rede. Esse banco de dados consiste de rotas específicas para cada nó móvel, ou roteador móvel, nos roteadores do domínio para encaminhar os pacotes. Essas rotas são atualizadas de acordo com a mobilidade do elemento móvel.

Esquemas Baseados em Tunelamento

É o esquema que aplica os conceitos de encapsulamento e registro hierárquicos, ou locais, para limitar o escopo das mensagens de sinalização relacionadas com a mobilidade, reduzindo, assim, o impacto de uma sinalização global e a latência do *handoff*. Isso é feito através da concatenação de, possivelmente, vários túneis locais.

2.2.3 Escopo da Sinalização de Mobilidade

O escopo da sinalização de mobilidade define qual elemento de rede é responsável por iniciar o processo de sinalização de mobilidade. Tal sinalização é necessária para o estabelecimento, manutenção e destruição do banco de dados de localidade do nó móvel. A arquitetura de rede de mobilidade local[6] pode ser dividida em duas partes mutuamente exclusivas: o núcleo da rede e o nó móvel; e o escopo, por consequência, pode ser dividido em dois grupos: os centrados na rede e os centrados no nó móvel.

Escopo Centrado no Nó Móvel

Nesse esquema, o nó móvel participa ativamente no processo de gerência de mobilidade, auxiliando nas etapas de sinalização de mobilidade. Ele deve implementar uma instância do protocolo de mobilidade da rede visitada, ou, pelo menos, um ou alguns de seu subsistemas. Dispositivos legados necessitam de atualização de seu *firmware* para poder usufruir dessas redes, assim como poder computacional suficiente ou adequado.

Escopo Centrado na Rede

Nesse esquema, o núcleo da rede é responsável por cuidar de todo o processo inerente à gerência de mobilidade, em nome do nó móvel. A premissa é que o nó móvel não possua qualquer protocolo de sinalização relacionado com a mobilidade, mas apenas os protocolos de redes já consagrados pela indústria e usualmente encontrados nesses tipos de dispositivos.

[6] A arquitetura de rede de mobilidade a que aqui se refere é a rede no qual o nó móvel está visitando.

2.2.4 Camada onde Ocorre a Mobilidade

A camada onde ocorre a mobilidade define a responsabilidade de cada camada de rede no suporte aos protocolos de mobilidade.

Camada de Enlace

Engloba as soluções de mobilidade, as quais se baseiam na melhoria ou alterações dos protocolos da camada de enlace, para preservar as informações da camada de rede e, por conseguinte, manter as conexões de transporte entre *handovers*. Ela utiliza as informações provenientes desta camada para manter o endereço IP inalterado.

Camada 2½

Engloba as soluções de mobilidade, que se baseiam no protocolo MPLS, para realizar o encaminhamento de pacotes e manter as conexões de transporte inalteradas entre *handovers*. O protocolo MPLS possui características tanto dos protocolos de enlace, ao prover informações pertinentes a essa camada, quanto aos protocolos de camada de rede, sendo, normalmente, classificado como protocolo de camada 2½. Note-se que essa classificação era inexistente na definição original da pilha de protocolos TCP/IP.

Camada de Rede

Engloba as soluções de mobilidade as quais utilizam-se dessa camada para manter as conexões de transporte intactas. O principal enfoque é manter toda a infraestrutura de rede legada inalterada.

Camada de Transporte

A mobilidade na camada de transporte mantem a conexão de transporte fim a fim e sua coerência semântica, enquanto permite o redirecionamento das pontas da conexão para uma outra sessão de transporte [12].

Camada de Aplicação

A mobilidade na camada de aplicação permite que sistemas finais de comunicação suportem mobilidade e heterogeneidade. A mobilidade de terminal também permite que um dispositivo se mova entre subredes IP, enquanto continua a ser alcançável para conexão entrantes e mantenha as conexões de transporte entre mudanças de subredes, como por exemplo, o SIP [13].

Mobilidade no nível de aplicação permite também que usuários mantenham a sessão mesmo quando mudam de terminais. Por exemplo, um usuário pode querer continuar uma sessão, que foi iniciada em um MN, em um *notebook* ao entrar em seu escritório. Como esperado, tanto o protocolo IPv4, quanto o protocolo IPv6 não suportam nativamente essa mobilidade de sessão [14].

2.3 Protocolos de Mobilidade

Embora os nossos interesses sejam em soluções de mobilidade que utilizem o protocolo MPLS como tecnologia de encaminhamento de dados, vamos apresentar nessa seção, uma revisão bibliográfica das principais soluções que proveem mobilidade para as redes Internet.

Descreveremos as várias soluções de mobilidade, tanto as consideradas de micromobilidade, quanto as consideradas de macromobilidade, englobando todas as camadas que compõem a pilha de protocolos TCP/IP. Obviamente, iremos dar mais ênfase às soluções que utilizam o protocolo MPLS como tecnologia de encaminhamento de pacotes e manter uma descrição mais resumida das outras soluções.

Assim, iniciaremos a nossa discussão com a proposta do IETF para prover mobilidade, conhecida como protocolo MIP (*Mobile IP*), tanto na versão 4 do protocolo IP, quanto na versão 6, por ser o protocolo no qual a maioria é baseada. Posteriormente, iremos tratar das soluções em MPLS e, por fim, iremos descrever as outras soluções de mobilidade.

2.3.1 IP Móvel

O IP Móvel, também conhecido por MIP, é uma solução de macromobilidade proposta pelo IETF, tanto para o protocolo IP em sua versão 4 [15], chamado também de MIPv4, quanto para a sua versão 6 [16], normalmente chamado de MIPv6. A seguir descreveremos estes dois protocolos, considerando cada versão do protocolo IP.

IP Móvel Versão 4

O MIPv4 resolve o problema da mobilidade, ao manter a conexão TCP, através da manipulação dos elementos de rede definidos na camada 3. Uma conexão TCP é identificada através de dois descritores, um que define a origem da conexão e o outro que define o destino da conexão. Cada descritor é definido pelo par endereço de rede e porta. Assim, para que uma conexão não seja interrompida, os quatro parâmetros (endereço de rede da origem e do destino e as portas de origem e destino) não podem ser alterados.

Um nó móvel, quando realiza um *handover* e muda a sua rede IP, requisita um novo endereço IP da rede no qual está se conectando, para que possa novamente estabelecer uma

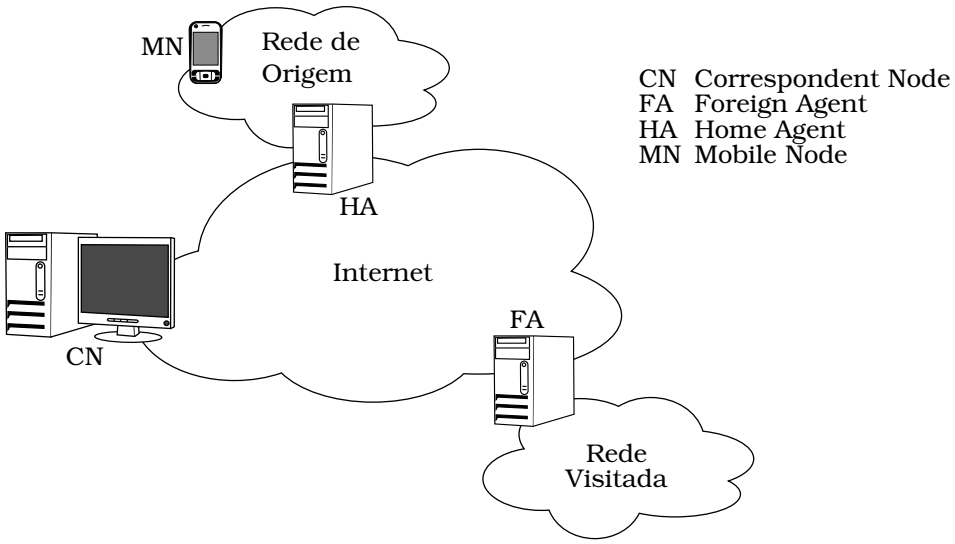


Figura 2.1: Arquitetura de referência para o MIP.

conexão com a Internet. O problema é que esse novo endereço adquirido, ou através de uma requisição automática de endereços (DHCP) ou através de entradas manuais, quebra a conexão TCP previamente estabelecida. Assim, serviços que exigem uma conexão contínua, como por exemplo VoIP (*Voice over Internet Protocol*) e mensageiros instantâneos, necessitam ser reinicializados. Isso faz com que a experiência de mobilidade do usuário seja bastante desagradável.

A solução proposta pelo MIP é manter inalterado o endereço que o nó móvel adquiriu em sua rede de origem, enquanto ele estiver em migração por outras redes IP. Assim, ele exige que o nó móvel seja capaz de reter dois ou mais endereços de rede associados com uma interface de rede (endereços de origem e de rede visitada).

Uma segunda restrição imposta ao nó móvel pelo MIP é que ele seja capaz de aceitar pacotes caracterizados pelos descritores antigos (endereço IP da rede de origem e porta) e entregá-los à aplicação associada.

A solução proposta pelo MIP pode ser melhor descrita pela arquitetura de referência, mostrada na Figura 2.1.

O MIP define dois novos agentes para auxiliar na mobilidade do nó móvel: o HA e o FA. O HA reside na rede de origem do nó móvel e é responsável por monitorar a localização atual do nó móvel. Por outro lado, o FA reside na rede visitada e é responsável em rastrear o nó móvel nesta rede. Tanto o HA quanto o FA podem ser um elemento de rede qualquer, mas, normalmente, são roteadores IPs com as respectivas funções (HA ou FA) adicionadas a eles.

O HA precisa manter, em seu banco de dados, a posição atual do nó móvel, pois, o papel exercido por este agente depende de onde o nó móvel está. Se estiver em sua rede de origem,

o HA não está ativo. Por outro lado, se o nó móvel estiver em *roaming*[7], o HA personifica o nó móvel e age em nome dele para receber os pacotes que lhe são destinados. Em outras palavras, ele age como um *proxy* para o nó móvel, recebendo os pacotes que são endereçados a ele e enviando-os para a sua rede IP atual. Conforme dito anteriormente, a posição atual do nó móvel é dada pelo endereço atribuído pela rede visitada e que coexiste com o endereço de rede de origem, ambos alocados em sua interface de rede.

Quando o HA personifica o nó móvel, em sua rede de origem, e necessita lhe entregar os pacotes que está interceptando, ele encapsula os pacotes em um novo pacote IP, cujo endereço de destino é o endereço da rede visitada. Esse processo também é conhecido por tunelamento [17, 18, 19].

Na outra ponta do túnel está o FA[8]. O seu papel é "destunelar" os pacotes, obtendo o pacote original e entregá-lo ao nó móvel. Este último, por sua vez, deverá estar conectado ao FA através da camada de enlace.

Dinâmica do Protocolo

O MIP chama o FA ou o HA de agentes de mobilidade e estabelece que esses agentes devem anunciar sua presença na rede, de tempos em tempos, através das mensagens de divulgação (*agent advertisement messages*). Elas são inseridas nas mensagens do protocolo ICMP (*Internet Control Message Protocol*) de descobrimento de roteador (*ICMP router discovery messages*), como extensões deste protocolo.

Para que um nó móvel se conecte com a Internet, é necessário que ele adquira um endereço IP. Este endereço pode ser é obtido, em sua rede de origem, através de vários métodos, como por exemplo, uma requisição DHCP ou via configuração manual. Um dos requisitos do MIP é que esse endereço seja mantido inalterado sempre que o nó móvel estiver em *roaming*. Portanto, é necessário que ele saiba quando está conectado em sua rede de origem ou quando está conectado em uma outra rede qualquer. Para descobrir isso, ele usa mensagens de divulgação emitidas periodicamente pelos agentes de mobilidade.

Se o nó móvel detecta que ele está em sua rede de origem, ele opera sem os serviços de mobilidade. Assim, se existe uma conexão estabelecida entre uma outra máquina na Internet, chamada pelo MIP de CN, ela será realizada da forma tradicional, ou seja, como definido no protocolo TCP/IP. A mesma situação acontece quando ele migra de uma rede qualquer, em que estava registrado, retornando para a sua rede de origem. Primeiramente, ele deve desfazer o registro perante o HA, através das mensagens *Registration Request* e *Registration Replay* e, em seguida, usar a rede sem os serviços de mobilidade.

[7] O termo *roaming* significa que o nó móvel é um visitante em uma rede IP qualquer.

[8] O FA é um elemento opcional. Se o nó móvel tiver recursos suficientes, pode atuar como o fim do túnel e receber os pacotes do HA diretamente.

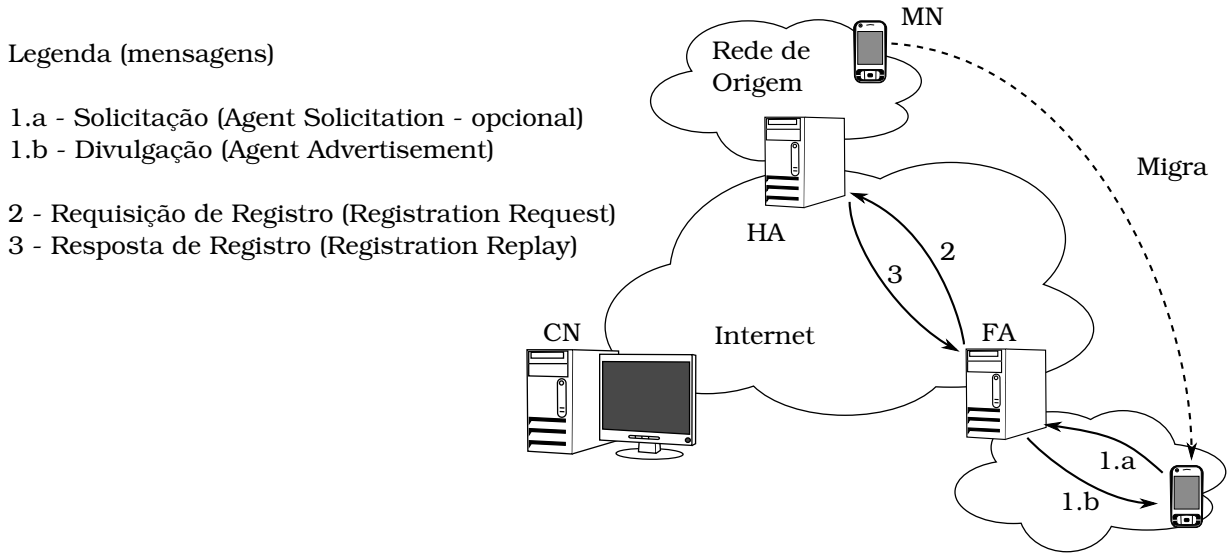


Figura 2.2: Processo de migração de um nó móvel de sua rede de origem para uma outra rede qualquer.

Por outro lado, se o nó móvel detecta que ele se moveu para uma outra rede qualquer, diferente de sua de origem, faz-se necessário obter um novo endereço IP, chamado de CoA, e deve coexistir com o endereço IP obtido, previamente, em sua rede de origem. Há dois tipos possíveis de CoA: o FCoA (*Foreign Agent Care-of-Address*) e o CCoA (*Co-Located Care-of-Address*). A Figura 2.2 ilustra o processo de migração e troca de mensagens do protocolo.

Após receber o CoA, ou através da troca de mensagens com o FA, via mensagens de divulgação, ou de um outro método qualquer, como por exemplo, o DHCP, o nó móvel deve iniciar o processo de registro deste novo CoA com o seu respectivo HA. Se o CoA foi obtido de um FA, o nó móvel delega a este FA o processo de registro com o HA, caso contrário, o nó móvel deve realizar todo o processo de registro, diretamente com o HA.

O CoA que será registrado no HA depende de quem é responsável em realizar o registro. Se for o FA, será registrado o endereço do FA; se for o nó móvel, será registrado o CCoA. É mais vantajoso o FA assumir essa responsabilidade pois permite que vários nós móveis compartilhem o mesmo CoA (o endereço IP do FA) e, por conseguinte, não impõe uma demanda maior nos já limitados endereços IPv4.

O FCoA é atribuído ao nó móvel pelo FA e permite que este faça a demultiplexação do pacote e entregue-o ao nó móvel correspondente. O CCoA é obtido pelo nó móvel através de outros meios, tais como DHCP ou configuração manual. A vantagem do uso do CCoA, em detrimento do FCoA, é a possibilidade de usar um nó móvel em uma rede que não suporta o protocolo MIP. Como é mais vantajoso utilizar o FA no MIPv4, vamos continuar a discussão considerando sempre a presença desse agente.

O HA é um dos elementos centrais no MIP e tem vários papéis a desempenhar, descritos a seguir:

- Personificar o nó móvel. Quando o nó móvel está em *roaming*, as mensagens enviadas a ele, na sua rede de origem, são interceptados pelo HA. A personificação acontece quando o HA responde às mensagens de ARP (*Address Resolution Protocol*), com o propósito de mapear o endereço IP do nó móvel com o seu endereço MAC (*Media Access Control*). Existem dois mecanismos que são utilizados para implementar essa funcionalidade. O primeiro deles ocorre toda vez que o nó móvel se registra, ou atualiza um registro, com o HA, o qual utiliza-se do artifício de enviar um pacote ARP (requisição ou resposta) para causar uma atualização nos *caches* dos elementos de redes presentes. Esse processo é conhecido por *Gratuitous ARP*. O segundo ocorre quando um elemento de rede deseja atualizar a sua tabela ARP. Quando ele envia uma requisição de mapeamento de ARP para o nó móvel, o HA responde com o seu MAC. Esse processo é conhecido como *Proxy ARP*;

- Redirecionar os pacotes para o nó móvel. Sempre que o CN envia uma mensagem ao nó móvel e o mapeamento ARP já está estabelecido, o HA intercepta essas mensagens e as encaminha para a sua atual localização, atuando em nome do nó móvel. Essa funcionalidade é continuação da função do *Proxy ARP*;

- Estabelecer um túnel com o FA. Para que seja possível entregar os pacotes enviados pelo CN ao nó móvel, sem alterações, o HA estabelece um túnel entre ele e o FA, no qual todas as mensagens são encapsuladas, ou seja, é criado um novo cabeçalho IP e a mensagem anterior se torna o *playload* desta nova mensagem.

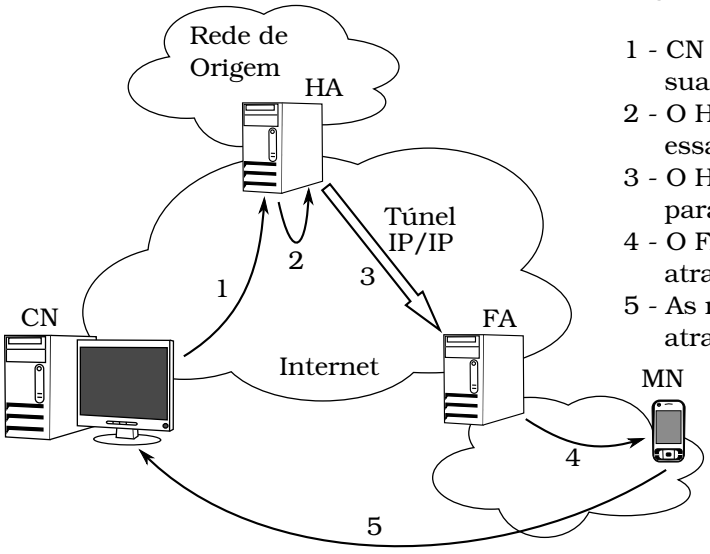
A Figura 2.3 ilustra um cenário típico do protocolo MIP, quando um nó móvel está em operação em uma rede que seja diferente de sua rede de origem.

Vemos nessa figura como o protocolo reage quando um CN mantem uma sessão com o nó móvel. Podemos resumi-lo através dos seguintes passos:

1. Para o CN, o nó móvel ainda está em sua rede de origem e, portanto, todas as mensagens são enviadas para o seu endereço IP, pertencente a esta rede;

2. O HA está personificando o nó móvel e, portanto, atuando em seu nome. Assim, ele recebe os pacotes que lhe são destinados e os encaminha ao FA que fora previamente registrado, durante a fase de sinalização;

3. O encaminhamento é realizado através do encapsulamento dos pacotes originais em um túnel IP/IP. Isso mantem o pacote intacto e íntegro até a sua entrega ao nó móvel;



Legenda:

1 - CN envia mensagens para o MN em sua rede de origem
2 - O HA personifica o MN e responde a essas mensagens
3 - O HA tunela as mensagens recebidas para o FA
4 - O FA entrega os pacotes para o MN através da camada de enlace
5 - As mensagens de resposta são retornadas através da Internet, do modo tradicional.

Figura 2.3: Um cenário de comunicação entre o CN e o MN, quando ele está em uma rede diferente do que a de origem.

4. O FA recebe os pacotes através do túnel entre o HA e o FA e retira o cabeçalho adicional, recuperando o pacote original. O pacote é entregue ao nó móvel através de um enlace de camada 2;

5. O nó móvel recebe os pacotes inalterados pelo processo, como se estivesse em sua rede de origem. A resposta enviada pelo nó móvel para o CN segue o roteamento tradicional do protocolo IP, uma vez que o nó móvel conhece o endereço do CN. O campo *source address*, no cabeçalho do pacote IP, deve ser o endereço IP da rede de origem do nó móvel.

Conforme elucidado pela RFC 3024 [20], existe um problema de coerência topológica quando o pacote enviado pelo nó móvel para o CN retorna por um caminho diferente do que ele fez quando chegou ao nó móvel.

O problema reside no fato de que o campo *source address* deve ser preenchido com o valor do endereço IP de sua rede de origem, pois o CN não deve saber qual a sua localização atual. Assim, os pacotes que emanam da atual rede em que o nó móvel reside são topologicamente incorretos, pois o campo *source address* possui um valor diferente de rede do que é esperado.

Embora o roteamento IP seja baseado somente no endereço de destino do pacote, pacotes topologicamente incorretos podem ser descartados como medidas preventivas, para aumentar a segurança, como por exemplo, ataques do tipo *IP spoofing*.

Para resolver o problema, a RFC 3024 [20] propõe um método para a criação de túneis reversos que são topologicamente corretos.

IP Móvel Versão 6

Com a atual expectativa de substituição da atual versão do IP pela sua próxima versão, surgiu também a proposta de atualização do protocolo de mobilidade para a nova versão do protocolo IP, o qual é conhecida por MIPv6 (*Mobile IP version 6*) [16].

O MIPv6 mantem as mesmas ideias básicas delineadas anteriormente, mas apresenta algumas mudanças significativas na estrutura de funcionamento da arquitetura, as quais listamos a seguir:

- Exclusão das funcionalidades do agente FA – o MIPv6 opera em qualquer localidade sem um suporte especial de um roteador local;
- Suporte para otimização de rotas – é uma parte fundamental do protocolo, ao invés de um conjunto não padrão de extensões usados pelo MIPv4;
  - Otimização de rotas pode ser operada com segurança, mesmo na ausência de associações de segurança pré-definidas;
  - Permite também que a otimização de rotas coexista eficientemente com roteadores que realizam filtragem no ingresso (*ingress filtering*);
- A maioria dos pacotes pode ser enviada para o nó móvel, enquanto ele não está em sua rede de origem, usando o cabeçalho de roteamento IPv6 (*IPv6 routing header*), ao invés do encapsulamento de túneis do IPv4. Isso reduz o *overhead* resultante, comparado com MIPv4;
- MIPv6 é desacoplado da camada de enlace, uma vez que ele utiliza a descoberta de vizinhança do IPv6 (*IPv6 neighbor discovery*) ao invés do ARP;

O protocolo MIPv6 impõe também que o nó móvel deve ser capaz de aceitar múltiplos endereços IP em sua interface de rede, ou seja, retendo vários CoAs simultaneamente. Isso permite, por exemplo, que o nó móvel continue a receber os pacotes que ainda estão sendo encaminhados para o seu enlace anterior, enquanto ele já está conectado em seu novo. O processo de se registrar o CoA no HA é conhecido, no MIPv6, como *binding*. Ele faz isso enviando ao HA uma mensagem de *binding update* e o HA envia como resposta, ao nó móvel, uma mensagem de *binding acknowledgement*.

O protocolo MIPv6 especifica dois modos nos quais um nó correspondente pode se comunicar com um nó móvel. O primeiro deles estabelece que a comunicação se dá da mesma forma que é descrita nas especificações do MIPv4. O HA mantem um banco de dados com as informações de localidade de todos os MN que estão atualmente registrados nele. Quando o CN envia uma mensagem para o nó móvel, ela será enviada para a rede de origem do MN.

O HA irá interceptar essa mensagem e tunelá-la para a atual localização do nó móvel. Este, por outro lado, irá responder a essas mensagens utilizando um túnel reverso, cujo destino é o HA, para que ele complete o encaminhamento das mensagens ao nó correspondente. O túnel reverso é importante para manter a coerência topológica da rede. Essa opção é interessante pois permite que o nó correspondente fique alheio à implementação MIPv6, ou seja, não há uma participação deste elemento na dinâmica do protocolo. Por outro lado, não é uma solução otimizada do ponto de vista de encaminhamento de pacotes.

A segunda forma permite que o nó móvel registre o seu endereço CoA perante ao CN. Assim, o nó correspondente pode enviar as mensagens diretamente ao nó móvel, evitando passar pelo HA, após o estabelecimento do primeiro contato. É a forma mais otimizada de troca de mensagens entre esses dois elementos, mas exige que o nó correspondente participe ativamente da dinâmica do protocolo.

Questões de Segurança no MIP

O grande diferencial entre o protocolo MIPv6 e o protocolo MIPv4 é a exigência do uso de mecanismos de segurança para proteger as mensagens. A lista inclui a proteção das mensagens de *binding updates* tanto para os HAs quanto para os CNs, da descoberta do prefixo móvel e dos mecanismos que o IPv6 utiliza para transportar pacotes de dados.

Em particular, as mensagens de *binding update* e *binding acknowledgement* são protegidas, em sua integridade e autenticidade, utilizando-se das extensões do protocolo de segurança IPsec (*Internet Protocol Security*) [21] e devem usar, minimamente, o cabeçalho ESP (*Encapsulating Security Payload*), no modo de transporte, conforme descrito na RFC 3775 [16].

Para qualificar uma conexão segura, o IPsec utiliza entradas, chamadas de Associações Seguras – SA (*Security Associations*), que descrevem completamente os seus parâmetros[9]. Como deve ser criado um SA, para cada conexão estabelecida, e as conexões são unidirecionais, deve ser especificado um modo para que seja possível obter o SA correspondente a um pacote, dentre os vários SAs possíveis. Essa função é chamada de Seletor e a RFC 4301 [21] especifica que existem três formas em que esse seletor pode ser usado (deve ser pesquisado na ordem dada a seguir):

- pesquisando por uma entrada em que a combinação do campo SPI (*Security Parameters Index*), endereço de origem e endereço de destino coincidem;
- pesquisando por uma entrada em que a combinação do campo SPI e do endereço de destino coincidem;
- pesquisando por uma entrada baseado somente no campo SPI.

[9]Vários são os parâmetros que definem uma conexão segura. Citamos a título de exemplo, a chave usada para autenticar/criptografar e o algoritmo de criptografia utilizado

Note-se que essa é uma das atualizações ocorridas na RFC 4301, a qual tornou obsoleta a RFC 2401. O Seletor anterior era baseado no campo SPI, endereço de destino e no protocolo IPsec utilizado [22].

Para evitar que um MN possa enviar uma mensagem de *Binding Update*, utilizando o SA correspondente, em nome de um outro MN, a RFC 3775 exige que o endereço *Home* do MN esteja no pacote. Assim, o HA pode selecionar corretamente qual o SA correspondente ao MN solicitante. Essa informação deve aparecer como um endereço de origem ou destino, como uma opção de endereço *Home* de destino, ou no cabeçalho de roteamento do tipo 2.

Considerações sobre o MIP

A operação do MIP envolve, basicamente, três diferentes atividades: processo de notificação de agentes, processo de registro dos MNs e processo de encaminhamento de dados [23]. É importante que operem de forma eficiente para que o MIP seja escalável com o aumento de MNs.

Toda vez que o MN muda de uma rede lógica para uma outra e ganha um novo endereço IP, faz-se necessário realizar o procedimento de registro da nova localidade, perante o HA. Como essa sinalização pode passar por vários *hops* até alcançar o HA, acaba por se tornar um processo demorado. Em redes em que a taxa de *handover* é alta, devido possivelmente ao tamanho reduzido das células da rede de dados, a latência devido à sinalização envolvida se torna elevada e a perda de pacotes se torna expressiva.

2.3.2 Protocolos de Mobilidade no MPLS

MPLS Móvel

O MPLS móvel, proposto no artigo *Integration of Mobile IP and Multi-Protocol Label Switching* [23], é um protocolo de macromobilidade que sugere uma implementação do MIPv4 usando o MPLS como protocolo de tunelamento, substituindo o tunelamento IP/IP como especificado inicialmente.

Ren *et al* especificam a arquitetura considerando que os agentes de mobilidade (HA e FA) são LERs e estão em uma mesma nuvem MPLS. Para realizar o rastreamento de localização do MN, eles sugerem que a tabela FEC (*Forwarding Equivalence Class*), que está localizada no HA, seja utilizada para este fim.

Cada MN possui uma entrada na tabela de FEC do HA[10]. Os descritores do segmento de saída desta entrada informam ao HA a localização atual do MN. Existem duas possibilidades para esses descritores:

[10]Pode existir mais de uma entrada na tabela FEC para um determinado MN. Isso pode ser necessário para satisfazer questões de QoS.

- Se forem nulos, o MN está em sua rede de origem;
- Se forem não nulos, o MN está em uma rede externa e os descritores informam qual deve ser o LSP (*Label Switched Path*) de saída para este pacote.

O LSP de saída é construído através da sinalização LDP (*Label Distribution Protocol*) e é disparado com o registro do MN junto ao FA da rede visitada. O FA, ao receber um pedido de registro por parte do MN, inicia o processo de criação de um LSP com o HA. Após a sua criação, o HA atualiza a entrada da FEC do MN para refletir, em seus descritores de saída, os descritores de saída deste novo LSP.

Sempre que o MN migra para um nova rede, o processo se repete e a entrada da FEC, deste MN, é atualizada no HA para refletir sua atual localização. Isso faz com que uma das responsabilidades do HA, que é a pesquisa na tabela de MN para saber a sua localização, seja incorporada de forma mais natural ao MPLS, já que é uma função nativa do protocolo.

Além do cenário discutido acima, os autores discutem também mais dois cenários, comuns de se encontrar na prática, sempre considerando que tanto o HA, quanto o FA são LERs:

- O HA e o FA se encontram em nuvens MPLS com domínio administrativos diferentes, interconectados por um LER. A solução proposta consiste em utilizar um protocolo de borda, como por exemplo, o BGP (*Border Gateway Protocol*), para trocar informações de rótulos e o procedimento acima se aplica completamente;
- O HA e o FA se encontram em nuvens MPLS separadas por uma nuvem IP que conecta as duas redes. A solução proposta consiste em ligar as duas nuvens MPLS por um túnel IP.

O MPLS móvel é uma solução de macromobilidade, baseada no protocolo MIP e, portanto, sofre das mesmas limitações do anterior, no que diz respeito à quantidade de *handoffs* e na latência em restabelecer as conexões. Apesar dos resultados apresentados pelos autores serem bastante satisfatórios com relação ao MPLS, deve-se observar que eles utilizaram uma implementação do MPLS no núcleo do SO (Sistema Operacional), enquanto que a implementação do MIP era executada no espaço do usuário, que poderia provocar resultados tendenciosos.

H-MPLS

O H-MPLS, proposto no artigo *Hierarchical Mobile MPLS: Supporting Delay Sensitive Applications Over Wireless Internet* [24], é um protocolo de micromobilidade. Sendo baseado no protocolo MPLS móvel, esta proposta trata da mobilidade de um nó móvel localmente a um domínio.

A justificativa em propor uma modificação do MPLS móvel reside no fato dele ser um protocolo de macromobilidade e, portanto, não se adequa bem quando o tamanho das células diminui e, por conseguinte, aumenta a quantidade de *handoffs*.

Para tratar da mobilidade localmente, Yang e Makrakis propõem a criação de um novo agente de mobilidade, chamado por eles de FDA (*Foreign Domain Agent*), responsável em prover a mobilidade dentro de um domínio. O papel exercido por esse novo agente se situa entre os atribuídos para o HA e os atribuídos para o FA. Deve existir apenas um único FDA por domínio MPLS.

Para um HA, o FDA atua exatamente como se fosse um FA, conforme definido pelo MIP. Como em um domínio podem existir vários subdomínios, ou rede lógicas, cada uma delas possui um FA. Para estes FAs, o FDA atua como se fosse um HA.

A dinâmica do protocolo é a seguinte. Quando um MN migra pelo primeira vez para o domínio, ele inicia o processo de registro ao receber uma mensagem de divulgação (*advertisement message*) do FA; o FA, ao receber a solicitação de registro do MN, encaminha essa solicitação ao seu FDA que, por conseguinte, inicia o processo de criação de um LSP com o FA, usando o endereço do MN como FEC. O FDA ao receber essa mensagem de registro, encaminha-a ao HA que, em seguida, inicia o processo de criação do LSP com o FDA, conforme descrito no MPLS móvel, para rastrear o MN.

Portanto, ao final do processo de registro irão existir dois LSPs: um entre o HA e o FDA e o outro entre o FDA e o FA. Se o MN se mover de uma subrede para um outra subrede, dentro do mesmo domínio, somente o LSP entre o FDA e o FA deverá ser atualizado e o HA não será notificado desta atualização de localidade. Assim, o processo de registro se repete como descrito anteriormente, mas, o FDA ao verificar que já existe uma entrada em sua tabela de MNs, não repassa esse pedido para o HA e somente atualiza o LSP com o novo FA.

Para evitar perda de pacotes no processo de migração, os quais seriam enviados para o FA antigo enquanto o novo LSP não é construído, os autores estabelecem também que o MN deve enviar ao FA antigo uma mensagem de *binding update*, informando-o qual é o atual FA em que ele está associado. Assim, se um pacote é enviado ao FA antigo e o MN não está mais associado a ele, este FA deve criar um novo LSP com o FA atual e enviar esses pacotes.

No artigo é especificado também um protocolo de sinalização, chamado de *fast signaling for mobile MPLS*, cujo mecanismo de estabelecimento de conexão é chamado de *make-before-break*. A ideia é estabelecer um LSP nas subredes que o MN irá visitar antes que ele efetivamente migre para lá. Esses LSPs, criados antecipadamente, são denominados de LSPs passivos e o LSP no qual o MN está trocando pacotes é chamado de ativo.

Assim, quando o MN migra de uma subrede para outra, basta apenas ativar o LSP da rede para a qual ele está migrando, colocando-a no estado ativo e, para o LSP que estava anteriormente ativo, colocá-lo no estado passivo. Segundo os autores, o processo de ativação

e desativação de um LSP é bem mais eficiente do que o processo de estabelecimento de um novo.

Conforme pode ser percebido, o papel do FDA é crucial para o funcionamento deste protocolo. É um elemento centralizador, no qual todos os MNs, dentro de um domínio, devem ter uma entrada FEC neste LER. Ele também deve atuar como um elemento de egresso e ingresso, pois ele é egresso para os LSPs que conectam os HAs com este FDA e deve agir como ingresso para os LSPs que rastreiam os MNs.

Assim, existe um problema potencial de escalabilidade com o aumento de MNs dentro de um domínio, pois o poder computacional deste elemento de rede deverá ser bastante elevado e possuir uma alta capacidade de encaminhamento de pacotes.

É assumido também que o MN implemente o protocolo proposto e possua um poder computacional condizente com o processamento necessário para executá-lo. Além de ser o responsável por enviar as mensagens de *binding update* para o FA antigo, é de sua responsabilidade também, conhecer a topologia da rede visitada, pois é ele quem mantem os LSPs passivos na sinalização proposta.

LEMA

LEMA, proposto no artigo *A Network Architecture for MPLS-Based Micro-Mobility* [25], é um protocolo de micromobilidade cujo domínio administrativo é uma rede MPLS. Chiussi *et al* consideram também que o protocolo de macromobilidade é implementado pelo MIP.

A arquitetura proposta pelo artigo consiste em se criar uma rede sobreposta (*overlay network*) à rede MPLS, composta de LERs especiais, chamados de LEMA (*Label Edge Mobility Agent*), com algumas funções especiais, as quais são:

1. Ser capaz de mapear o endereço IP de destino de um pacote em um par que contém o próximo *hop* e o rótulo de saída;
2. Ser capaz de aceitar e processar uma mensagem local de registro (*Registration message*), o qual causa um novo mapeamento a ser criado para um dado endereço IP com o par especificado acima;
3. Ser capaz de aceitar e processar uma mensagem de redirecionamento (*Redirect message*), o qual gera uma mudança dos valores de próximo *hop* e rótulo de saída para um dado endereço IP de destino.

A rede sobreposta é criada com LSPs estáticos, estabelecidos através de engenharia de tráfego, que conectam estes LEMA sobre uma determinada topologia. Todos os roteadores de acesso devem ser expandidos para atuar como um LEMA e são chamados de níveis mais baixos. Isso está em contraste com os LEMAs que estão mais próximos, logicamente, dos

HAs, os quais são chamados de níveis mais altos. É o LEMA de nível mais alto que tem seu endereço IP registrado no HA, enquanto os LEMAs de níveis mais baixos fazem a entrega do pacote ao MN.

A dinâmica do protocolo é a seguinte. Quando um MN entra em uma rede LEMA, recebe uma mensagem de divulgação e faz a associação com um endereço IP da rede visitada. Essa mensagem de divulgação contém, entre outros atributos, uma lista de LEMAs que estão servindo uma área geográfica e sua disposição hierárquica. De posse dessa lista, o MN ou o roteador de acesso, em nome dele, escolhe a lista de LEMAs que irão servir à conexão.

Assim, quando um CN deseja se corresponder com um MN, ele encaminha à rede de origem do MN os pacotes de dados e o HA tunela esses pacotes até o LEMA de mais alto nível, o qual atua como um FA, usando o protocolo MIP. O LEMA de nível mais alto encaminha esses pacotes, usando LSPs pré-estabelecidos através dos LEMAs escolhidos pelo MN, ou do roteador de acesso, em nome dele, até o LEMA de mais baixo nível. Nesse ponto o pacote deixa a nuvem MPLS e é entregue ao MN através da camada de enlace.

Quando um MN migra de uma subrede para uma outra qualquer, o processo é repetido. Nesse ponto, se existirem um ou mais LEMAs que são comuns com a conexão antiga, eles podem ser reutilizados na criação da nova conexão. Com isso, o MN[11] sinaliza qual é a lista de LEMAs que ele deseja que seja utilizada para servir a sua conexão e o processo de registro pára no primeiro LEMA que é comum às duas conexões, à antiga e à nova. Nota-se que a mensagem de registro não alcança o HA e, portanto, é mais eficiente do que o MIP. No pior caso, não existiriam LEMAs em comuns e todo o processo se repetiria, inclusive com a participação do HA em todo o processo.

Segundo os autores, essa arquitetura alcança quatro objetivos:

Fast Handover – resolve o problema do re-registro frequente entre o MN e o HA. Isso melhora a sobrecarga de sinalização e a perda de pacotes inerente ao processo, dentro de um domínio administrativo.

Projeto escalável – permite uma arquitetura flexível e distribuída da mobilidade local. Flexibilidade provê ao MNs a habilidade de escolher quais os LEMAs que irão compor uma conexão. Arquitetura distribuída refere-se à capacidade de espalhar na rede (LEMAs) as informações de encaminhamento dos pacotes, ou seja, nem todos os LEMAs precisam conhecer as informações de encaminhamento de MNs que não estão sob sua responsabilidade.

Capacidade de QoS – provê serviços de QoS para a micromobilidade. A ideia é utilizar os mecanismos nativos do MPLS para prover QoS no nível de serviços diferenciados.

[11] Lembrando também que é especificado que o roteador de acesso pode realizar essas funções em nome do MN.

Implementação gradual – permite a coexistência entre os elementos que implementam os serviços LEMAs e os LERs/LSRs (*Label Switch Router*) que apenas realizam as funções básicas da arquitetura do MPLS [6].

Como todos os LEMAs são LERs e alguns deles podem estar no meio da nuvem MPLS, esses LERs devem ser capazes de atuar como elementos de ingresso e egresso simultaneamente. Em outras palavras, se um LSP termina em um determinado LEMA, ele deve remover o rótulo do pacote, processar os dados de camada de rede, reclassificar em uma nova FEC e tunelar o pacote dentro de novo LSP de saída.

Outra questão que deve ser observada é o fato que o algoritmo que permite a escolha dos LEMAs, que compõe uma conexão, pode ser bastante complexo e não foi determinado ainda. Entre outros parâmetros, ele deve contemplar os padrões de mobilidade de um MN, a largura de banda disponível, dentre outros fatores.

MM-MPLS

O MM-MPLS, proposto no artigo *Fast Handoff Process in Micro Mobile MPLS Protocol for Micro-Mobility Management in Next Generation Networks* [26] e no artigo *Micro Mobile MPLS: A New Scheme for Micro-mobility Management in 3G All-IP Networks* [27], é um protocolo de micromobilidade. É assumido que a arquitetura MM-MPLS (*Micro Mobile MPLS*) é aplicada em um domínio administrativo, o qual implementa uma rede MPLS e tratará da mobilidade do MN localmente. É assumido também que o protocolo MIP está em execução para lidar com as questões de macromobilidade.

Langar *et al* propõem a hierarquização da rede em dois níveis. O primeiro é composto de elementos de rede que atuam como egresso de uma rede MPLS e que são os últimos que processam um pacote em camada 3. Tem as mesmas funcionalidades de um FA e são chamados de LER/FA. A partir deste ponto, o LER/FA se conecta com o MN através de uma conexão de camada 2. O segundo é um elemento de borda da rede e está mais próximo do HA. É chamado de LER/GW, ou simplesmente LERG. A função do LER/GW é registrar o seu endereço IP perante o HA e esconder a mobilidade local do MN. Deve haver apenas um LER/GW para cada domínio administrativo.

A dinâmica do protocolo é a seguinte: sempre que um MN entra em uma rede MM-MPLS e recebe uma mensagem de divulgação, ele inicia o registro com a nova rede. O LER/FA, ao receber o endereço do MN, armazena-o em suas tabelas e passa essa solicitação para o seu LER/GW. O LER/GW, ao receber essa solicitação de registro, inicia a criação de um novo LSP com o LSR/FA e, simultaneamente, registra o seu endereço IP junto com o HA. O LER/GW instala também uma FEC, em sua tabela de FECs, com a entrada do endereço IP do MN e cujos valores dos segmentos de saída, porta e rótulo de saída, são os mesmos atribuídos para o LSP com o LER/FA.

Assim, sempre que um CN envia dados ao MN, para sua rede de origem, o HA tunela esses pacotes e os envia para o LER/GW, usando o protocolo MIP. O LER/GW, ao checar o endereço atribuído ao MN, pesquisa em sua tabela FEC uma entrada com esse endereço e os encaminha para o LSP associado, ou seja, o que foi previamente criado entre do LER/GW e o LER/FA. O LER/FA, ao receber os pacotes, retira o cabeçalho MPLS e os encaminha, via camada 2, para o MN.

Quando o MN realiza um *handover*, mas continua no mesmo domínio administrativo, ao receber uma mensagem de divulgação, inicia o processo de registro, conforme explicitado anteriormente. Assim, quando o LER/GW recebe a mensagem de solicitação de registro, inicia a criação de um novo LSP com o LER/FA e verifica se já existe uma entrada para o MN em sua tabela FEC. Como, neste caso, já existe uma entrada, o LER/GW não passa a solicitação de registro para o HA e apenas atualiza os valores referentes ao segmento de saída do MN, para serem iguais ao que foi previamente estabelecido.

Eles estabelecem também que um *handover* mais eficiente pode ser obtido através da utilização de LSPs pré estabelecidos. A ideia é manter LSPs com as várias subredes que o MN poderia visitar e as deixam em um estado passivo, enquanto que o LSP, o qual o MN está conectado, fica em um estado ativo. Em uma migração, bastaria a troca de estados, de ativo para passivo e vice versa, para restabelecer a conexão.

Existe uma semelhança bastante forte entre essa proposta e a do H-MPLS [24], descrita anteriormente.

I-LIB

O I-LIB, proposto no artigo *Fast Handover over Micro-MPLS-Based Wireless Networks* [28], é um protocolo de micromobilidade. Ele é baseado no protocolo H-MPLS e estende suas funcionalidades ao expandir o conceito da LIB (*Label Information Base*), inserindo alguns campos a mais para lidar com a micromobilidade. Essa nova LIB é chamada pelo autores de I-LIB (*Intermediate Label Information Base*).

A ideia consiste em utilizar a mesma arquitetura proposta no H-MPLS, ou seja, cada domínio administrativo possui um FDA, o qual registra o seu endereço IP perante o HA e estabelece um LSP com este agente. O FDA é único por domínio administrativo e todos os FAs, os quais são os roteadores de acesso com a rede de enlace sem fio, devem criar um LSP para cada MN que está registrado neles.

Assim, quando um MN entra pela primeira vez em uma subrede de um domínio administrativo e recebe uma mensagem de divulgação, ele inicia o processo de registro com o FA. O FA, ao receber essa solicitação, passa essa mensagem ao FDA, o qual inicia a criação do LSP com o FA, tendo como FEC o endereço IP do FA e cria, também, uma entrada na FEC com o endereço IP do MN, fazendo com que os parâmetros de saída desta FEC sejam iguais aos

parâmetros de saída do LSP criado com o FA. Ao mesmo tempo, o FDA repassa essa solicitação ao HA, o qual inicia a criação de um LSP com o FDA. Note-se que o HA recebe, como endereço CoA do MN, o endereço IP do FDA.

Em uma migração, quando o MN move de uma subrede para uma outra subrede qualquer, desde que ainda esteja dentro do mesmo domínio administrativo, somente o LSP que liga o FDA ao novo FA deve ser refeito. O processo se inicia da mesma forma, com o MN recebendo uma mensagem de divulgação do novo FA. Assim, o MN inicia o processo de registro enviando uma mensagem de registro ao novo FA. Ele, ao receber essa mensagem, encaminha-a ao FDA que inicia a criação de um novo LSP com o FA, cuja FEC de saída é o endereço IP do FA. O FDA verifica também que já existe uma entrada na FEC com o endereço IP do MN e atualiza os seus parâmetros de saída para serem iguais aos parâmetros de saída da nova FEC.

Fowler e Zeadally [28] destacam que, durante o processo de encaminhamento de pacotes para cada LSR que compõe um LSP, com exceção do FDA que é um roteador de ingresso, a operação a ser realizada é sempre a mesma, ou seja, a consulta na tabela LIB para determinar qual deve ser o novo rótulo e sua interface de saída. Essa tabela é indexada pelo rótulo e a interface de entrada. Eles observam também que existe a possibilidade de que, após a migração do MN, o novo LSP pode ter vários LSRs em comum com o antigo LSP.

Dessa forma, ao invés de criar um novo LSP com o FDA, basta criar um segmento do LSP até o primeiro LSR que é comum aos dois LSPs; atualizar sua LIB para contemplar o novo segmento e se desfazer do segmento antigo que vai deste LSR até o antigo FA. Assim, a latência da sinalização e o tempo de estabelecimento do LSP é diminuído, pois não há a necessidade de criação de um novo LSP da forma tradicional.

Nota-se que quando o MN ingressa em uma nova subrede, ele recebe um novo endereço CoA e, uma vez que o LSP será reaproveitado parcialmente, o FDA não será notificado deste novo endereço IP. Para resolver esta dessincronização, os autores do artigo adicionam campos pertinentes à LIB do LSR de comutação[12], para que ele possa fazer a troca de CoAs, em tempo de encaminhamento, nos pacotes.

Portanto, o LSR deve conhecer, não somente o CoA atual do MN mas também o CoA antigo e o endereço original do MN (endereço *home*). Todos esses valores são presentes na LIB do LSR, incluindo alguns outros, para a gerência da mesma. Esses valores são passados via mensagens de sinalização MIP com a adição que ele deve conter também o endereço CoA antigo.

Assim, a dinâmica do protocolo passa ser a seguinte. Um CN envia pacotes para o MN em sua rede de origem. O HA, em nome do MN, intercepta esses pacotes e os encaminha para o FDA, usando o seu endereço IP como FEC. O FDA ao receber esses pacotes, extrai o endereço IP do MN, localiza sua FEC e os envia através do LSP que foi estabelecido para o FA antigo.

[12]Este é o LSR que era comum aos dois segmentos, o antigo e o novo.

Todos os LSRs do caminho analisam o conteúdo do pacote e os checam com sua I-LIB, para verificar se houve ou não uma comutação de segmentos. Quando o pacote chega ao LSR de comutação, ele verifica que necessita de atualizar os valores de CoA do pacote, troca o CoA antigo pelo Coa atual e os tunela novamente no LSP para o FA atual. O processo continua até chegar ao FA, que verifica a coerência topológica do pacote, retira o cabeçalho do MPLS e o encaminha, através da camada de enlace ao MN.

Quando o MN envia pacotes ao CN o processo inverso acontece, onde, no LSR de comutação, deve trocar o CoA atual pelo CoA antigo, pois é através deste endereço que o FDA conhece o MN.

Outra vantagem é a redução da carga sobre o FDA, pois existe um gargalo quando vários MNs estão sob a sua supervisão, ou vários *handovers* estão acontecendo simultaneamente. Assim, nesse esquema, a sobrecarga de criação de um novo LSP é espalhada pelo domínio MPLS.

Para minimizar a perda de pacotes em um *handover*, os autores especificam também que o MN deve ser capaz de sinalizar o FA antigo sobre o seu novo CoA. Assim, se porventura um pacote destinado ao MN for encaminhado ao FA antigo, ele deve reter esse pacote, estabelecer um novo LSP com o FA atual e, então, encaminhá-los. O FA antigo pode receber esses pacotes basicamente por dois motivos: porque o processo de estabelecimento do LSP é demorado e, enquanto ele não é criado, o LSP antigo continua válido; e devido a erros na construção da I-LIB.

Portanto, a proposta apresentada necessita alterar os pacotes IP para manter a sua coerência topológica. Isso viola uns dos princípios básicos de uma rede MPLS que é a opacidade dos dados transportados. Com isso, a rede MPLS só funcionaria para os dados que fossem transportados pelo protocolo IPv4. Nota-se também que todos os LSRs devem analisar o conteúdo dos pacotes que são encaminhados, para poder decidir se deve ou não haver comutação. Quanto mais LSRs forem aproveitados em um caminho, maior será o *overhead* associado com a troca de endereços.

Outro problema, associado à violação dos endereços do pacotes, é se eles estiverem sendo protegidos por algum protocolo de segurança, como por exemplo, o IPsec. Existe a possibilidade de que esses pacotes sejam descartados no destino por não existir uma associação segura que os discriminem corretamente. Note-se também que todos os LSRs devem responder à sinalização MIP expandida para o estabelecimento do LSP, que é mais uma fonte de *overhead* para o sistema.

Por fim, normalmente, os LSRs de núcleo são mais simples que os LSRs de borda, do ponto de vista de capacidade computacional, por não realizarem operações mais complexas, o que pode ser um problema para a implantação desse protocolo em redes legadas.

2.3.3 Protocolos de Mobilidade no Nível IP

Conforme será visto no Capítulo 3 – Arquitetura do Plano de Mobilidade, esta proposta lida muito bem com várias tecnologias de tunelamento, sendo as principais os túneis MPLS e os túneis IP/IP. Os túneis IP/IP são soluções que utilizam somente os protocolos de camada de rede para solucionar o problema da mobilidade.

Assim, nesta seção, vamos apresentar, resumidamente, as principais soluções de micro-mobilidade que focam na camada de rede para resolver esse problema.

Cellular IP

O Cellular IP (CIP), proposto no artigo *Cellular IP: A New Approach to Internet Host Mobility* [29], é um protocolo de micromobilidade. A arquitetura é composta de diferentes domínios de acesso sem fio, chamadas de redes de acesso CIP, conectadas com a Internet através de um *gateway*. O MIP gerencia a mobilidade entre essas redes de acesso CIP, enquanto que o *Cellular IP* gerencia a mobilidade dentro do domínio.

O endereço IP do *gateway* é usado como o CoA do MIP. É esse endereço que é registrado perante o HA. Assim, quando um CN envia dados para o MN, os pacotes são primeiramente roteados para o HA e, então, tunelados para o *gateway*, o qual extrai os pacotes e os encaminha para as estações bases, usando um caminho de roteamento específico para o MN.

As estações bases funcionam como pontos de acesso sem fio e realizam, também, o roteamento dos pacotes IP. O roteamento original IP é substituído pelo roteamento Cellular IP e gerência de localização, conforme especificado pelo artigo. As estações bases realizam o *cache* de roteamento do caminho utilizado pelo pacotes do MN para o *gateway*, por um período de tempo, e utilizam o caminho reverso para realizar o roteamento dos pacotes provenientes do *gateway* para os MNs.

Afim de rotear pacotes para MNs, que estão no estado inativo, o *Cellular IP* emprega o conceito de *paging*.

HAWAII

O HAWAII, proposto no artigo *HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Network* [30], é um protocolo de micromobilidade. O HAWAII divide a rede de dados em uma hierarquia de domínios. Todas as questões relacionadas com a gerência de mobilidade, dentro de um domínio, são geridas por um *gateway* chamado de DRR (*Domain Root Router*). Esse *gateway* utiliza um esquema especializado de estabelecimento de caminhos, o qual instala entradas de rotas baseadas no *host*, em roteadores específicos, para suportar a mobilidade intra-domínios.

Enquanto o MN estiver se movendo dentro de sua rede de origem, ele mantem o seu endereço IP. O MIP passa a ser usado quando o MN se move para uma rede externa. Porém, se a rede externa for baseada, também, no HAWAII, o MN recebe um endereço CoA do domínio externo, para o qual os pacotes para o MN são tunelados. O DRR roteia os pacotes para o MN utilizando-se de entradas de roteamento baseadas no *host*.

Quando o MN se move entre diferentes subredes, de um mesmo domínio, somente a rota que vai do DRR até a estação base, que está servindo o MN, é modificada. Os outros caminhos permanecem os mesmos e a conectividade é mantida utilizando-se de caminhos estabelecidos dinamicamente.

O protocolo contém três mensagens diferentes para estabelecer, atualizar e confirmar as rotas específicas dos MNs no DRR e nos roteadores intermediários, que estão no caminho até o MN. O protocolo possui, também, quatro diferentes esquemas de estabelecimento de caminhos, cujo objetivo é reduzir a interrupção do tráfego do usuário, durante o *handoff*. Esses esquemas são classificados em dois tipos, baseados no modo como os pacotes são entregues ao MN. No primeiro caso, os pacotes são encaminhados da estação base antiga até a nova e, no segundo tipo, eles são divididos no roteador comum aos dois segmentos.

MIP-RR

O MIP-RR, proposto na RFC 4857 intitulada de *Mobile IPv4 Regional Registration* [31], é um protocolo de micromobilidade.

O MIP-RR é uma extensão opcional ao protocolo MIPv4 e propõe um modo para os MN se registrarem localmente, dentro de um domínio visitado. Ao se registrar localmente, o número de mensagens de sinalização, para a rede de origem, é mantido ao mínimo e o atraso de sinalização é reduzido.

Esse protocolo introduz um novo nó de rede, chamado de GFA (*Gateway Foreign Agent*). Além das mensagens de registro do MIP, um novo par de mensagens de registro, ou seja, as requisições de mensagens de registro regional e suas respostas, são usadas entre os MNs/FAs/GFAs.

Há dois modelos de como o MN utiliza-se do registro regional. No primeiro modelo, os FAs de um domínio visitado anunciam o endereço do GFA e, quando o MN chega pela primeira vez neste domínio, ele deve realizar o registro em sua rede de origem. Nesse processo, o MN deve registrar o endereço do GFA, como sendo o seu CoA, no HA. Assim, quando o MN se move entre subredes, dentro do mesmo domínio, ele deve somente realizar um registro regional com o GFA.

No segundo modelo, o FA indica que a designação dinâmica do GFA deverá ser utilizada, sendo a responsabilidade do FA em escolher qual GFA deve ser usado, após receber uma requisição de registro do MN.

PMIP

O PMIP, proposto na RFC 5213, intitulada *Proxy Mobile IPv6* [32], é um protocolo de micromobilidade, cujo foco da proposta é centrar todo o protocolo de mobilidade na rede de dados, sem envolver o MN na dinâmica deste, e que se baseia em túneis, dentro de um domínio, para direcionar o tráfico para o MN.

O PMIP que tem como base a versão 6 do protocolo IP, reutiliza vários conceitos do MIPv6, como por exemplo, a funcionalidade do HA, e define duas novas entidades: o *Gateway* de acesso móvel – MAG (*Mobile Access Gateway*) e a âncora de mobilidade local – LMA (*Local Mobility Anchor*).

O MAG tipicamente é executado em um roteador de acesso. Ele é responsável pela detecção do ponto de associação do MN e, se as políticas de segurança são atendidas, pelo estabelecimento de túneis com o LMA, para o direcionamento de tráfego dos MN que são alcançáveis por este MAG.

Um MAG emula também, via mensagens de anúncio de roteador (*Router Advertisements messages*), a rede originária do MN, de tal forma que o MN possa mudar o roteador padrão em um *handover*, mas preserve os outros parâmetros de camada 3. Quando o MN se move dentro de um domínio, os túneis entre MAG e LMA e as rotas no LMA são atualizadas.

O LMA mantem uma entrada de *cache* para cada MN atualmente registrado, fornecendo uma alcançabilidade ao seu endereço IP. Quando um LMA recebe um pacote destinado ao MN, ele encaminha este pacote através de um túnel, cuja ponta terminal é o MAG no qual o MN está associado. O PMIP emprega mensagens de ligação local (*local binding update messages*) entre o MAG e o LMA com o propósito de sinalização e possui apenas uma única hierarquia de túneis.

Na realidade, o PMIP considera também um suporte ao protocolo IPv4, mas isto requer extensões ao protocolo original, uma vez que ele utiliza características próprias do IPv6, como por exemplo, a auto-configuração e os cabeçalhos de extensões.

HMIPv6

O HMIPv6, proposto na RFC 5380, intitulada de *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management* [33], é um protocolo de micromobilidade.

A proposta do HMIPv6 estende tanto o protocolo MIPv6 quanto o protocolo de descobrimento de vizinhança do IPv6 (*IPv6 Neighbor Discovery*) [34] e introduz o elemento de rede chamado de MAP (*Mobility Anchor Point*), o qual é um agente de mobilidade IPv6.

Um MN, ao entrar em um domínio HMIP, recebe uma mensagem de anúncio de roteador (*Router Advertisements message*) contendo informações sobre um ou mais MAPs locais e, em seguida, configura dois endereços CoAs: um relacionado ao enlace, chamado de LCoA (*on-link*

CoA); e o outro relacionado ao registro regional, chamado de RCoA (*Regional CoA*).

O LCoA é configurado na interface de rede do MN, baseado no prefixo anunciado pelo seu roteador principal. Ele é um endereço CoA, conforme especificado pelo MIP, e possui um nome diferente apenas para diferenciar do RCoA. O RCoA é configurado no enlace pertencente ao MAP e é obtido pelo MN, proveniente do MAP, empregando os mecanismo de endereçamento conforme descritos na RFC 4877 [35].

Após a configuração, o MN envia duas mensagens de BUs. A primeira, é uma mensagem de BU local para o MAP, cuja finalidade é associar o RCoA do MN com o seu LCoA e, também, estabelecer um túnel bidirecional entre eles. A segunda mensagem de BU é enviada ao HA, com a finalidade de associar o endereço IP de origem do MN com o seu RCoA.

O MAP recebe todos os pacotes em nome do MN que ele está servindo. Ele os encapsula e os envia diretamente ao endereço LCoA do MN. Quando o MN se move dentro de um mesmo domínio MAP, somente há a necessidade de se registrar o seu novo LCoA com o seu MAP. Isso limita a quantidade de sinalização do MIPv6 que sairia de um domínio local. O endereço RCoA permanece inalterado e o HA e CN não são notificados desta mudança de endereço LCoA.

2.3.4 Outros Protocolos de Mobilidade

O paradigma de mobilidade é bastante extenso e envolve soluções nas diversas camadas que compõem o protocolo TCP/IP. Nesta seção, apresentamos sucintamente as propostas de mobilidade nessas diversas camadas, em particular, nos protocolos de mobilidade na camada de enlace, na camada de transporte e na camada de aplicação.

FMIP

O FMIP, proposto na RFC 5268, intitulada *Mobile IPv6 Fast Handovers* [36] é um protocolo de micromobilidade, cuja solução é centrada na camada de enlace.

O objetivo do FMIP é reduzir a latência na detecção do movimento do MN e a latência na configuração do novo endereço CoA do MN, ao prover informações para o MN, enquanto ele ainda está conectado em sua subrede corrente.

Assim, o MN, após descobrir os pontos de acesso disponíveis, requisita informações da subrede destes APs, ou seja, o prefixo de rede, o endereço IP e o endereço de camada 2 de seus roteadores associados. Se o MN eventualmente se conecta a um destes APs, o atraso na detecção do movimento é reduzido, uma vez que o MN não necessita realizar o descobrimento do roteador.

O MN constrói um novo CoA – NCoA (*New CoA*), baseado no prefixo de sua nova subrede, e envia uma mensagem para o seu roteador de acesso atual, chamado de PAR (*Previous Access Router*), o qual se comunica com o futuro roteador de acesso – NAR (*New Access Router*) para

determinar se o NCoA é único.

O PAR estabelece também um túnel com o NAR, para redirecionar os pacotes destinados ao PCoA (*Previous CoA*) e enviá-los ao NCoA. Após o realização do *handover*, o MN anuncia sua conexão imediatamente, através de uma mensagem de anúncio de vizinhança não solicitada (*Unsolicited Neighbor Advertisement message*) [34], para contornar o atraso associado com a resolução de endereços de vizinhança.

O FMIP define também um comportamento alternativo do protocolo, quando o MN não recebe uma mensagem de confirmação antes do seu *handover*. Há também uma adaptação deste protocolo, o qual é definido para a versão 6 do IP, para ser utilizado na versão 4 deste protocolo, ou seja, o IPv4 [37].

IP-IAPP

O IP-IAPP, proposto no artigo *Fast and Efficient IP Handover in IEEE 802.11 Wireless LANs* [38], é um protocolo que pode ser considerado tanto de macromobilidade quanto de micromobilidade, cuja solução é focada na camada de enlace.

A proposta do IP-IAPP estende o protocolo 802.11F IAPP (*Inter-Access Point Protocol*) [39] para suportar *handover* inter-redes através de métodos específicos para a camada de enlace. O IP-IAPP define o elemento de rede HAP (*Home Access Point*), que é o AP no qual o MN esteve associado pela última vez dentro de sua rede de origem, similar ao HA do MIP.

Quando o MN se move para uma rede diferente, ele envia uma mensagem modificada de requisição de reassociação do IEEE 802.11 [40] para um AP, a qual é chamada pelo IA-IAPP de FAP (*Foreign Access Point*), informando o endereço IP do HAP, do MN e do FAP anterior (PAP – *Previous FAP*) e a mensagem dispara o procedimento de gerência de mobilidade.

O FAP se comunica com o HAP a fim de estabelecer um túnel bidirecional (HAP-FAP). Por outro lado, o HAP realiza o mapeamento do endereço IP do MN com o endereço IP do FAP, chamado de FACoA (*Foreign Agent Care of Address*).

Quando o MN reassocia com um novo FAP (NAP – *new FAP*), o mesmo procedimento descrito anteriormente é realizado, com a adição de uma comunicação extra entre o PAP e o NAP a fim de estabelecer um túnel unidirecional temporário entre eles. A proposta foi também melhorada com a provisão de serviços mais avançados: comunicação segura IP-IAPP inter-AP, alteração zero (*zero patching*) com o *software* do cliente e suporte a clientes que utilizam endereçamento dinâmico IP [41].

O comitê executivo do IEEE 802 abandonou a proposta IAPP em 2006, devido a: “o período de teste do 802.11F terminou, não houve uma aplicação significativa das implementações do 802.11F e as funcionalidades fornecidas pelo 802.11F estão sendo usadas por outros padrões.” (Tradução do autor) [42].

MSOCKS

O MSOCKS, proposto no artigo *MSOCKS: an Architecture for Transport Layer Mobility* [12], é um protocolo de micromobilidade, cuja solução é focada na camada de transporte.

O MSOCKS é uma arquitetura baseada em *proxy*, com conexão segmentada, que se utiliza da técnica de segmentação do TCP (*TCP Splice*) a fim de se conseguir as mesmas semânticas fim a fim de uma conexão normal TCP.

Um *host* especial, chamado de *proxy*, é posicionado no caminho de comunicação entre o MN e o CN. Uma conexão TCP fim a fim entre esses dois elementos é, então, dividida em duas conexões separadas: uma delas se refere à conexão entre o MN e o *proxy* e a outra se refere à conexão entre o *proxy* e o CN.

O protocolo MSOCKS estende o protocolo SOCKS [43] com a finalidade de redirecionar um *stream* de dados TCP para a localização, após migração, do MN. Assim, quando um MN muda o endereço IP de sua interface de rede, ele abre uma nova conexão com o *proxy* e envia uma mensagem MSOCKS, especificando o identificador de conexão da conexão original.

O *proxy* reagrupa a conexão antiga entre o MN e o *proxy*, com a conexão entre o *proxy* e o CN e a segmenta novamente com a nova conexão entre o MN e o *proxy*. Somente o *proxy* necessita estar ciente da migração do MN e as comunicações entre o *proxy* e o CN permanecem inalteradas. Outro ponto a ser ressaltado é o fato que essa técnica permite, também, que o MN mude a interface de rede utilizada para se comunicar com o *proxy*.

Migração TCP

Migração TCP, proposta nos artigos *An End-to-end Approach to Host Mobility* [44] e *The Migrate Approach to Internet Mobility* [45], é um protocolo de macromobilidade, cuja solução é focada na camada de transporte.

Migração TCP permite que uma aplicação, que está sendo executada em um MN, suporte uma conectividade transparente entre redes de dados que alteram o endereço IP. Assim, quando um MN muda o seu ponto de associação com a rede de dados, um novo endereço pode ser atribuído a ele, seja manualmente, seja através de um protocolo de auto-configuração, como por exemplo, o DHCP.

Quando se necessita localizar um MN, em uma nova rede, o DNS (*Domain Name System*) é usado para essa tarefa, incluindo aqui a sua habilidade de suportar atualizações dinâmicas seguras. Uma vez que a maioria das aplicações de Internet resolvem nomes de máquinas para um endereço IP, no início de uma transação ou conexão, está solução é viável para novas sessões que são iniciadas com o MN.

Assim, quando um MN muda o seu ponto de associação com a rede de dados, ou seja, o seu endereço IP, ele envia uma mensagem segura de DNS de atualização para um de seus

servidores de nomes, em seu domínio de origem, atualizando a sua atual localização. O mapeamento "nome para endereço", para os MNs, não está na *cache* dos outros domínios e, portanto, associações inválidas, ou dessincronizadas, são eliminadas.

No entanto, quando um MN se move durante uma conexão previamente estabelecida, pode-se suspender a conexão que está aberta e reativá-la a partir do novo endereço IP, enviando um pacote especial, chamado de *Migrate SYN*, para o CN. Tal pacote carrega, entre outras informações, um *token* que identifica a conexão anterior.

O pacote SYN sinaliza ao CN que ele deve ressincronizar a conexão com o MN em seu novo ponto de associação, isto é, o seu novo endereço de rede. Com isso, é possível prover um suporte à mobilidade como um serviço fim a fim, de acordo com os requisitos específicos de uma aplicação, sem alterações nos protocolos de camada de rede.

Mobilidade utilizando o SIP

O SIP (*Session Initiation Protocol*) [46] é um protocolo de sinalização, largamente utilizado para inicializar e terminar sessões de comunicação multimídias através da Internet. Ele pode ser utilizado por qualquer aplicação em que um protocolo de inicialização de sessão seja necessário.

O mecanismo de registro do SIP é considerado o equivalente, em camada de aplicação, ao mecanismo de registro do MIP. Entretanto, enquanto o MIP associa um endereço IP permanente (endereço IP de origem do MN) a um endereço IP temporário (CoA), o SIP associa um identificador, a nível de usuário, com um endereço IP temporário, ou ao nome de uma máquina [13].

Uma mensagem de `INVITE` é enviada pelo MN ao seu CN para inicializar uma sessão de comunicação. Os mecanismos para fornecer a mobilidade ao MN, durante uma sessão ativa, preveem que o MN necessita enviar uma outra mensagem de `INVITE` para o CN. Ela é necessária, a fim de abastecê-lo com informações acerca dos novos parâmetros da sessão de comunicação, após o *handover*, utilizando os mesmos identificadores de chamada, conforme foi inicializada na chamada original.

É uma solução que apresenta alguns inconvenientes [47]. A segunda mensagem de `INVITE` é enviada no contexto fim a fim e pode provocar um atraso relativamente alto. Além do que, o procedimento de *handover* se apoia na capacidade do CN manusear esse procedimento, aumentando assim o poder de processamento do MN. Um mecanismo auxiliar é necessário se o MN e o CN podem se mover ao mesmo tempo. Por fim, esse protocolo é considerado de macromobilidade, cuja solução é focada na camada de aplicação.

2.4 Considerações do Capítulo

Esse capítulo discutiu os conceitos envolvidos no paradigma de mobilidade. Como se trata de uma expressão bastante vaga, foi redefinido o escopo em que esse paradigma está sendo aplicado, ou seja, nos elementos de rede de uma rede Internet.

Foram discutidos também os termos mais comuns utilizados pelas propostas de mobilidades e que nos permitem homogeneizar as discussões.

Na sequência, discutiu-se os macros componentes que compõem uma solução de mobilidade e são pertinentes ao conceito de gerência de mobilidade. Esses macro componentes são: gerência de localização, gerência de *handoffs*, gerência de endereços e sinalização. Além das definições e funcionalidades de cada componente, foi indicado também, o inter-relacionamento entre eles.

Uma classificação dos conceitos de mobilidade em quatro eixos taxionômicos foi proposta. Ela nos permite classificar as soluções de mobilidade propostas na literatura e visualizar, com mais clareza, os escopos de cada uma.

Uma revisão bibliográfica foi apresentada, em que soluções pertinentes a cada camada da pilha de protocolo TCP/IP foram discutidas. A ênfase recaiu sobre as soluções que utilizam o protocolo MPLS como tecnologia de encaminhamento de pacotes. A Tabela 2.1 mostra como tais soluções podem ser classificadas segundo os quatro eixos taxionômicos propostos.

Tabela 2.1: Os protocolos de mobilidade divididos em quatro eixos taxionômicos.

Protocolo	Alcance	Encaminhamento	Sinalização	Camada
MIP	macro	roteamento/tunelamento	Centrada no MN	3
MPLS móvel	macro	tunelamento	Centrada no MN	$2\frac{1}{2}$
H-MPLS	micro	tunelamento	Centrada no MN	$2\frac{1}{2}$
LEMA	micro	tunelamento	Cent. no MN/rede	$2\frac{1}{2}$
MM-MPLS	micro	tunelamento	Centrada no MN	$2\frac{1}{2}$
I-LIB	micro	tunelamento	Centrada no MN	$2\frac{1}{2}$
Cellular IP	micro	roteamento/tunelamento	Centrada no MN	3
HAWAII	micro	roteamento/tunelamento	Centrada no MN	3
MIP-RR	micro	tunelamento	Centrada no MN	3
PMIP	micro	tunelamento	Centrada na rede	3
HMIPv6	micro	tunelamento	Centrada no MN	3
FMIP	micro	tunelamento	Centrada no MN	2
IP-IAPP	macro/micro	tunelamento	Centrada no MN	2
MSOCKS	micro	roteamento	Centrada no MN	4
Migração TCP	macro	roteamento	Centrada no MN	4
SIP	macro	roteamento	Centrada no MN	7

Como consideração final, foram apresentados, também, os objetivos essenciais de qualquer solução de mobilidade, ou seja:

1. manter uma sessão aberta enquanto um nó migra de uma subrede, ou domínio, para um outro qualquer;
2. manter a experiência do usuário satisfatória durante o processo de mobilidade.

Após os objetivos iniciais terem sido alcançados, podemos aumentar a lista e citar uma outra, relacionando os objetivos secundários, os quais, se forem atendidos, enriquecem uma solução de mobilidade. São eles:

Provisão de QoS – permitir que as aplicações possam usufruir de reservas de recursos da rede, diferenciando um tráfego multimídia de um tráfego de melhor esforço. O modo mais imediato de aplicar os mecanismos de QoS é, ao invés de prover QoS fim a fim para o usuário, utilizar-se dos mecanismos já presentes pelas camadas de enlace e/ou rede, como por exemplo, os LSPs criados com reserva de recursos e mapear as aplicações sobre eles.

Segurança – utilizar mecanismos de segurança, como por exemplo o IPsec [22], para proteger, tanto as mensagens de controle do protocolo, quanto as mensagens de dados do usuário. Note-se que o MIPv6 já estabelece que as mensagens de controle do protocolo devem ser protegidas. Isso inclui tanto a sua autenticação, quanto evitar que outros elementos, não pertencentes a uma conexão, possam inspecionar o seu conteúdo.

Aplicação gradual – permitir que a solução de mobilidade coexista com a rede já implantada. Isso permite que uma solução possa ser implementada sem que seja necessário se desfazer completamente de uma infra-estrutura já instalada e funcional.

Uso de MNs legados – permitir que dispositivos legados e que, principalmente, não implementam a solução de mobilidade possam usufruir também destas redes. Em geral, para se alcançar este objetivo, deve-se abster o MN de participar dos mecanismos de mobilidade. Chamamos essa característica de mobilidade centrada na rede.

Capítulo 3

Arquitetura do Plano de Mobilidade

Conforme foi visto no Capítulo 2, os protocolos de micromobilidade possuem como um de seus objetivos, melhorar a eficiência dos *handoffs* e, por consequência, diminuir tantos os atrasos, quanto as perdas de pacotes. Isso se dá pelo contingenciamento da quantidade de mensagens de sinalização dentro de um escopo bem definido, usualmente, um domínio administrativo.

Várias são as propostas que visam otimizar o protocolo MIP, ao provê-lo com as funcionalidades de micromobilidade. Todos alcançam os objetivos iniciais, como uma proposta acadêmica, mas falham em resolver o problema sob a perspectiva da indústria.

A afirmação é embasada pelas seguintes constatações:

Centrada no MN – a maioria das soluções assumem a participação do MN na dinâmica do protocolo. Excluem os MNs legados e forçam os fabricantes a adotarem essas soluções como padrões de mobilidades, em seus produtos. Exigem também que o poder computacional do MN seja condizente com as operações dos protocolos, o que pode encarecer o produto final.

Novas Sinalizações – exigem a introdução de uma sinalização nova no sistema, ou a alteração de alguma sinalização existente. Uma proposta de sinalização leva anos para maturar e tornar-se um padrão aceito. Uma proposta nova deve passar por todo o processo de adoção e padronização.

Aplicado sobre o IPv4 ou o IPv6 – contemplam somente, ou o protocolo de rede atual, o IPv4, ou a próxima versão do protocolo, o IPv6. Apesar das previsões da exaustão do endereço IPv4 terem sido estimadas para o final da década de 90 e da necessidade urgente de sua substituição pela nova versão, na prática isso não aconteceu. Os motivos foram: demora na definição e padronização do IPv6, a substituição da divisão inicial do protocolo IPv4, que era em classes, pelo CIDR [4] e no emprego de roteadores NAT [5], os

quais permitiram o uso de endereços não roteáveis, dando uma sobrevida ao IPv4 sem previsão de término.

Túnel Temporário – muitas propostas tentam melhorar o desempenho do *handover* criando um túnel temporário entre o último CoA (PCoA) do MN e o CoA atual (NCoA), se os pacotes destinados ao MN chegarem no PCoA. O PCoA deve reter os pacotes, sinalizar a criação de um túnel temporário entre o PCoA e o NCoA e encaminhar estes pacotes por este novo túnel. A ideia do túnel temporário reside no fato de que a sinalização pode gastar um tempo até que seja completada e, portanto, os pacotes podem seguir pelo caminho antigo, antes que todo o processo de atualização de estados nos elementos de rede se completem. O problema é que esta nova sinalização, entre o PCoA e o NCoA, pode ser demorada também, e algumas aplicações não toleram um atraso muito grande na chegada de pacotes. Outra questão, neste tempo o PAR deve ter recursos suficientes para armazenar os pacotes em trânsito do MN. Se existirem vários MNs em *handoffs*, a gerência de recursos pode ser bastante complexa, além de exigirem *hardwares* mais avançados.

Complexidade – exigem a interação entre vários componentes, aplicação de algoritmos de difícil implementação prática ou alteração no funcionamento de protocolos já existentes.

Alterações nos Elementos de Rede – exigem alterações radicais na dinâmica dos elementos de rede para implementar o protocolo proposto.

Com relação ao MPLS, as soluções propostas podem ser divididas em três grupos: as que criam um LSP entre o LER de ingresso e o LER de egresso, para cada MN; as que permitem a utilização de pequenos LSPs, dentro da nuvem MPLS, para rastrear o MN; e as que alteram a arquitetura do MPLS, ao introduzir novas funcionalidades.

As propostas do primeiro grupo possuem como característica comum o fato de se gastar um tempo relativamente longo para estabelecer os LSPs. Isso dificulta a obtenção de um *handoff* transparente. Tais propostas exigem que o LER de ingresso seja capaz de mapear os dados pertinentes ao próximo *hop*, do LSP criado entre o LSR de ingresso e o LSR de egresso, com os dados pertinentes ao próximo *hop* da FEC que representa o MN, ou seja, uma duplicidade na tabela de FECs. Por fim, possuem um único LER de ingresso, responsável em rastrear a localização do MN, o que acaba sendo um elemento centralizador, susceptível a falhas e com problemas de escalabilidade.

As propostas do segundo grupo possuem como característica comum o fato de inspecionarem o cabeçalho de camada 3, para decidir sobre o encaminhamento do pacote. Note-se que essa é uma função específica de um LER de ingresso. O problema é que tais propostas exigem inspeção do pacote também no interior da nuvem MPLS. A RFC 3031 [6] estabelece que:

"No paradigma de encaminhamento do MPLS, uma vez que o pacote é designado a uma FEC, não há mais análise dos cabeçalhos pelos roteadores subsequentes; todo o encaminhamento é realizado pelos rótulos".

Outro problema em se criar LSPs no interior da nuvem MPLS é fazer com que os LSRs de núcleo se comportarem como LSRs de egresso e ingresso simultaneamente. Normalmente esse LSRs são mais simples que os de borda, por não necessitarem de funções mais complexas, e, também, algumas tecnologias não permitem que estes LSRs inspecionem o conteúdo de um pacote, por exemplo, os que são abrangidos pela especificação do GMPLS (*Generalized Multi-Protocol Label Switching*) [48].

As propostas do terceiro grupo possuem como característica comum o fato de alterarem a dinâmica de estabelecimento do LSP, alterando, além da sinalização para a sua criação, o comportamento do LSR. Por exemplo, a proposta do I-LIB [28] consiste em alterar segmentos do LSP para rastrear o MN. Para fazer isso, essas propostas assumem que o protocolo de camada 3 é o IPv4 e alteram o LSR, aumentando a sua LIB para conter informações pertinentes ao MN quanto a dinâmica para a escolha do segmento de saída.

Mudar as especificações do MPLS é uma proposta de difícil aceitação pelos fabricantes e, além disso, limita os protocolos que a nuvem MPLS seria capaz de encaminhar pois, conforme a RFC 3031 especifica, "MPLS significa 'Múltiplos protocolos' encaminhados por rótulos, múltiplos protocolos porque estas técnicas são aplicáveis em *qualquer* protocolo de camada de rede".

Por fim, embora não seja uma limitação, mas uma característica, a grande maioria das propostas de micromobilidade assumem que o protocolo de macromobilidade é o MIP, ou o MIPv4, ou o MIPv6, mas dificilmente ambas simultaneamente. Ainda que o MIP esteja no centro das atenções do IETF, como solução de mobilidade, não é interessante desconsiderar outras propostas, como por exemplo, o HIP (*Host Identity Protocol*) [49].

A arquitetura MPA [50, 51, 52, 53, 54, 55] surgiu como uma solução para o problema de micromobilidade, proveniente de pesquisas geradas sob a orientação do Prof. Dr. Eleri Cardozo e dos seus doutorandos, além do autor desta tese, Eduardo N. F. Zagari e Rodrigo C. M. do Prado. O objetivo da arquitetura MPA é suprimir e, em alguns casos, minimizar as deficiências apresentadas pelas propostas de micromobilidade, discutidas anteriormente.

Portanto, as principais características da arquitetura MPA são:

Centrada na rede – toda a sinalização pertinente à mobilidade está localizada na rede. O MN pode participar do processo se for desejado um *handover* mais eficiente do sistema, o qual é chamado de *handover* pró-ativo, mas é uma funcionalidade opcional.

Uso de dispositivos legados – é assumido que o MN implemente apenas o básico para o acesso à Internet, ou seja, a pilha de protocolos TCP/IP e o cliente do protocolo DHCP [56, 57], para atribuição de endereços automaticamente.

Independência do protocolo de rede – foi concebida para ser uma solução em micromobilidade, tanto para o protocolo IPv4, quanto para o protocolo IPv6, ou em ambas, simultaneamente.

Escalabilidade – não concentra todas as funções de mobilidade em um único componente, mas distribui essas informações pela rede.

Implantação gradual – permite a coexistência de elementos de rede que estão operando de acordo com a arquitetura MPA e os que somente implementam as funções básicas de roteamento/encaminhamento de pacotes.

Uso somente de protocolos já estabelecidos – a sinalização se baseia em protocolos já consagrados e estáveis pela indústria. As informações pertinentes à MPA são inseridas como mensagens opacas nesses protocolos, ou como alterações mínimas em seu funcionamento, no sentido de agregação de valores.

Degradação suave – se a rede sofre de alguma restrição qualquer, como por exemplo, um elemento de rede que falha, ou um túnel que não pode ser compartilhado entre duas redes de acesso, a rede se degrada suavemente, até o pior caso, que é quando somente o protocolo de macromobilidade esteja em atividade.

Extensibilidade – pode ser facilmente estendida com a inclusão de novas funcionalidades, como por exemplo, melhorias no gerenciamento, otimizações, ou através de outras personalizações.

Este capítulo apresentará a arquitetura MPA. A Seção 3.1 discutirá o modelo de referência da MPA, apresentando alguns elementos e definindo o componente primordial da arquitetura. Serão mostrados ainda alguns conceitos básicos referentes a ela. A Seção 3.2 irá discretizar os blocos funcionais que compõem a arquitetura e suas interações. Serão mostrados, também, dois cenários típicos em mobilidade, o registro na rede e o *handoff*. A Seção 3.3 irá tratar das questões de implementação, mostrando algumas possibilidades de se implementar os protocolos propostos e, em especial, como foram implementados em nosso protótipo. Será mostrada também como a arquitetura pode ser implementada utilizando a tecnologia de tunelamento IP/IP. A Seção 3.4 irá discutir algumas variações sobre a arquitetura básica, incluindo como realizar um *handoff* mais eficiente, como realizar a agregação de endereços e considerar múltiplos pontos de entrada na rede MPA. Por fim, a Seção 3.5 conclui o capítulo com algumas considerações sobre a arquitetura.

3.1 Modelo de Referência da Arquitetura MPA

MPA, cujo acrônimo significa *Mobility Plane Architecture*, se propõe a prover micromobilidade em uma rede IPv4/IPv6. O modelo de referência, no qual a arquitetura é baseada, é mostrado na Figura 3.1.

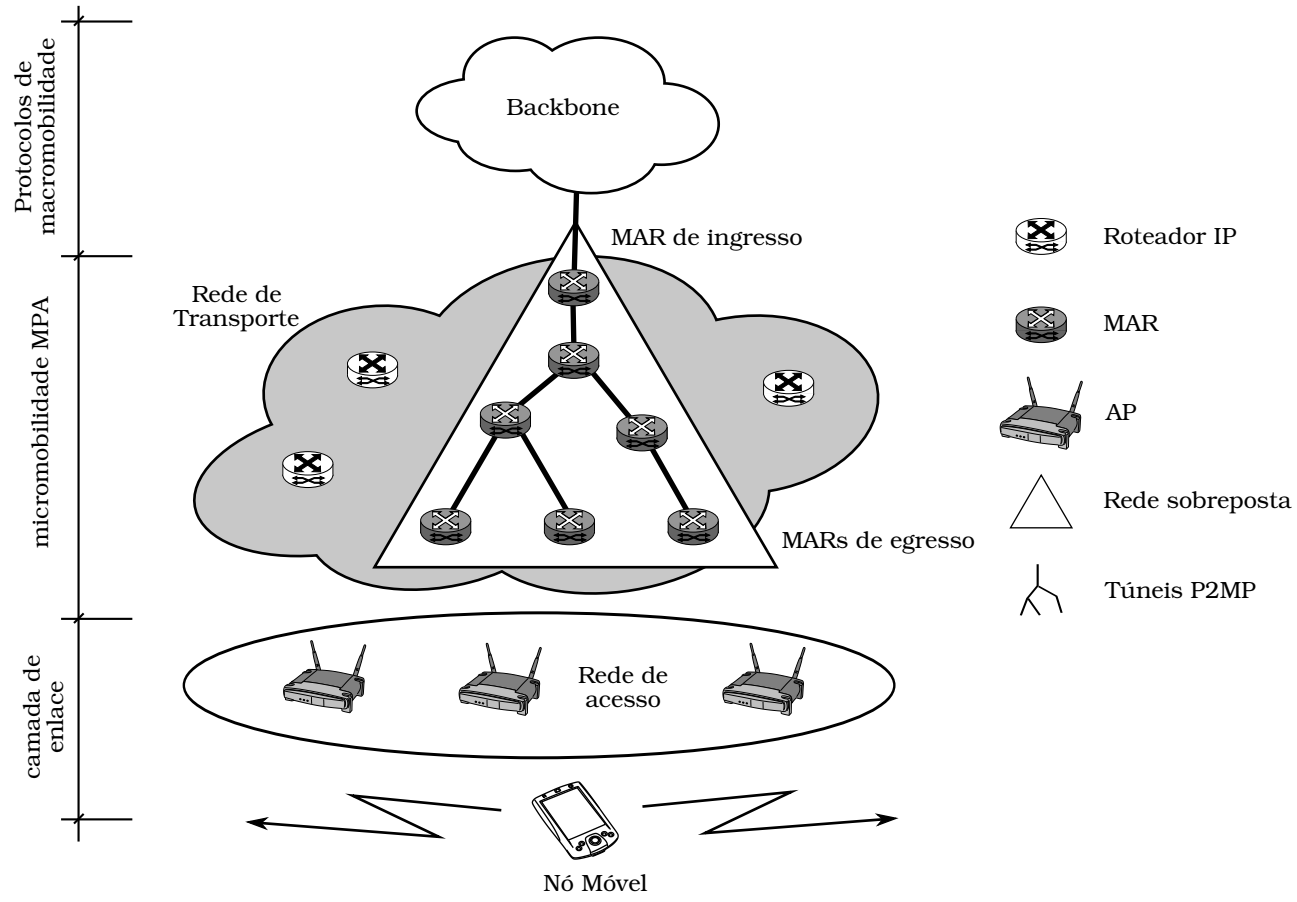


Figura 3.1: Modelo de referência para a arquitetura MPA.

MPA define alguns elementos que são pertinentes à sua arquitetura. Na base lógica da rede está o nó móvel (MN) e é o elemento que muda o seu ponto de associação, em camada de enlace, no tempo. Note-se que não se está restringindo se esse MN é um dispositivo do usuário, como um *notebook*, ou um elemento de rede móvel, como por exemplo, um roteador sem fio.

Acima, está o ponto de acesso (AP), o qual conecta o MN ao roteador de acesso. A sua função é prover um ponto de associação, em camada de enlace, para o MN. Associado a ele, está o roteador de acesso (AR) que é o primeiro elemento que processa informações de camada

de rede. Usualmente, é um MAR (*Mobility Aware Router*)[1] de egresso.

A rede de acesso (AN) é uma subrede IP composta pelos roteadores de acesso e os pontos de acesso. O prefixo IP desta rede é constante e único, em toda a sua extensão.

O núcleo da rede, chamada também de rede de transporte, é composto de uma rede IP, na qual o operador de rede deseja implantar os serviços de mobilidade. Normalmente, é um domínio administrativo, ou um subconjunto dele, em que se possui controle sobre os protocolos instalados.

O elemento principal desta arquitetura são roteadores IP, os quais foram expandidos com as funcionalidades previstas pela arquitetura MPA. Esses roteadores são chamados de MARs. A arquitetura MPA consiste de uma rede sobreposta (*overlay network*) à rede IP, cujos nós são MARs dispostos hierarquicamente e conectados entre si através de túneis do tipo ponto-multiponto, conhecidos também, por túneis P2MP (*Point-to-Multipoint*).

Separar, logicamente, a rede MPA da rede IP nos traz algumas vantagens:

1. Permite que a rede MPA seja implantada de forma gradual, ao adicionar novos MARs à rede já existente, aumentando a sua elasticidade e performance;
2. Permite ter um rede topológica diferente da rede de encaminhamento de pacotes;
3. Permite que a rede topológica seja alterada através de mecanismo de gerência, dinamicamente, sem alterar os mecanismos de roteamento IP;
4. Permite separar o plano de controle da arquitetura MPA do plano de controle do protocolo IP;
5. Permite que o tráfego pertinente à arquitetura MPA seja desacoplado do tráfego transportado pela rede IP.

Os túneis P2MP criam um grafo acíclico, cujos nós são compostos de MARs e os arcos são compostos de segmentos de túneis, conectando os MARs. Assim, tem-se um único MAR de ingresso, na raiz da árvore, em seu núcleo temos os MARs de ramificação, os quais são MARs que possuem interconexão com dois ou mais MARs, e, por fim, nas folhas dessa árvore, tem-se os MARs de egresso. O MAR de ingresso é o primeiro elemento que realiza a interface entre a Internet e a rede MPA e o MAR de egresso é o roteador mais próximo ao MN. Note-se que, nos MARs de ramificação, é possível que haja replicação de pacotes entre os vários ramos possíveis, dependendo das políticas implantadas nos MARs.

Um tráfego é dito *downstream* quando este flui do MAR de ingresso para o MAR de egresso, enquanto que o tráfego *upstream* flui na direção contrária.

MPA não faz restrição à tecnologia de tunelamento utilizada. Para que essa tecnologia seja elegível à arquitetura MPA, ela deve ser capaz de:

[1] O elemento MAR será definido a seguir.

- Criar redes sobrepostas: é a habilidade de conectar dois pontos distintos, escondendo a complexidade da rede que está entre eles. Isso pode ser alcançado, por exemplo, através do encapsulamento de dados, ou através do rotulamento de pacotes;

- Criar conexões ponto-multiponto: é a habilidade de criar múltiplas ramificações de segmentos de saída, em um dado MAR, para um determinado segmento de entrada de um túnel qualquer. A replicação de pacotes pode ocorrer nessas ramificações;

- Realizar realocação dinâmica de túneis: é a habilidade de mudar a topologia da rede por meio da adição ou remoção de segmentos de túneis.

Dessa forma, existe um conjunto de tecnologias de túneis que satisfazem essas exigências, dentre as quais, podemos citar o IP/IP [19, 17] e o MPLS [6].

Os túneis P2MP atendem ao tráfego *downstream* e são unidirecionais. O tráfego *upstream* pode seguir por túneis P2P (*Point-to-Point*) até o MAR de ingresso, ou seguir o roteamento IP padrão, ou seja, *hop* a *hop*.

Usualmente, os túneis P2MP são criados para aplicações de difusão de dados, que utilizam a faixa de endereços *multicast*, por exemplo, para a difusão de dados multimídia. No entanto, a arquitetura MPA define que os MNs, que serão rastreados pelo túnel P2MP, terão um prefixo de rede na faixa de endereços *unicast* e, portanto, os pacotes não serão replicados em todos os ramos deste túnel. Note-se que a rede de acesso deve ser capaz de aceitar os pacotes que são injetados dentro desse túnel. Portanto, uma outra característica desse modelo é que a rede de acesso possui um único prefixo de rede em toda a sua extensão, que implica que o endereço de rede do MN não se altera entre *handovers* sucessivos.

Tradicionalmente, a rede de acesso é dividida em subredes, cada uma com um prefixo de rede distinto, para que o tráfego fique contido localmente, evitando o gasto excessivo de largura de banda. Um outro motivo é diminuir o tamanho das tabelas de rotas nos roteadores, aumentando a escalabilidade da rede. Por fim, permite também a localização das máquinas finais e facilita a gerência de alocação de endereços.

Por outro lado, a arquitetura MPA determina que a rede de acesso possua apenas um único prefixo de rede, quebrando a divisão tradicional das redes em subredes. Note-se que isso não causa um impacto negativo na rede, pois:

- MPA é aplicado em um domínio administrativo, em que a alocação e gerência de endereços é administrada por um único gestor ou um grupo;

- Normalmente, um MN estabelece conexões com elementos externos à rede de acesso e, portanto, o tráfego local pode ficar contido apenas dentro do escopo de um ponto de acesso;

- Os túneis P2MP não são usados para difundir os pacotes em todos os MARs de egresso, mas, conforme será discutido, servem como uma possibilidade de conexão e rastreamento do MN. Portanto, não consomem largura de banda desnecessariamente;
- A escalabilidade também é mantida, pois a arquitetura permite a agregação de endereços, conforme será discutida na seção 3.4.2.

3.2 A Arquitetura MPA

Qualquer que seja a solução de micromobilidade, ela deve ser capaz de realizar quatro funções básicas: melhorar a eficiência do *handover* dos MNs; alocar endereços IPs aos MNs; rastrear a localização do MN e encaminhar os pacotes para a sua atual localização.

Assim, a arquitetura MPA, baseada em tais funcionalidades e em sua própria concepção, define quatro blocos funcionais, os quais são:

Bloco funcional TM (*Tunnel Management*) – possui o encargo de estabelecer, manter e alterar a topologia da rede sobreposta. Ele é responsável pela criação, remoção e reroteamento dos túneis que interligam os MARs, formando o grafo acíclico, ou seja, o túnel P2MP. Como esses túneis são de longa duração, é necessário que exista uma interface de comunicação, que pode ser usada tanto pelo sistema de gerência, quanto por um operador humano, para configurar a rede. Esse bloco funcional é implementado apenas nos MARs.

Bloco funcional MR (*Mobile Routing*) – possui duplo encargo: rastrear a atual localização do MN, ou seja, o seu ponto de associação com a rede e encaminhar os pacotes para o MN. As informações de localização são obtidas através de interações com o sistema de encaminhamento dos MARs (bloco TM), em que são extraídas as informações topológicas da rede. O encaminhamento de pacotes é obtido através do estabelecimento de conexões entre os segmentos dos túneis P2MP, que vão do MAR de ingresso aos MARs de egresso. Note-se que essas conexões estão contidas nos túneis estabelecidos pelo TM. Esse bloco funcional é implementado apenas nos MARs.

Bloco funcional AC (*Address Configuration*) – possui o encargo de suprir um endereço de camada 3 (um endereço IP) ao MN. Ele é ativado sempre que o MN se conecta, ou se reconecta, com a rede. Esse bloco funcional deve ser implementado, parte nos MARs, parte nos MNs.

Bloco funcional HH (*Handover Helper*) – possui o encargo de facilitar o processo de *handover* do MN. Ele realiza a sua tarefa através do uso de uma ou mais sinalizações relacionadas à camada de enlace, como por exemplo, do uso de notificações de camada

2, também chamadas de *triggers*; reassociação de camada 2; associação com segurança do MN e outras sinalizações relacionadas ao *handover*. Esse bloco funcional pode ser implementado entre vários componentes, ou seja, nos MARs, nos elementos de redes (*switches* sem fio) e nos MNs.

Assim, é através das interações entres esse blocos, os quais estão espalhados nos elementos que compõem a rede MPA, que a micromobilidade é alcançada. As seguintes interações foram identificadas e suas respectivas interpretações:

TM - MR – sempre que a topologia da rede sobreposta se altera[2], o TM deve avisar ao MR sobre essas mudanças. Isso é necessário, pois é sobre esta topologia que o MR cria as conexões utilizadas para encaminhar os pacotes ao MN.

HH/AC - MR – sempre que o HH ou o AC detecta a ocorrência de um *handover* e, isso pode ocorrer antes, durante, ou após ele ter sido completado, o HH ou o AC deve notificar o MR, para que este possa atualizar a real localização do MN e, por consequência, o encaminhamento de pacotes.

HH - AC – sempre que o bloco HH detecta a ocorrência de um *handover*, ele notifica o AC, para que este inicie o procedimento de configuração de endereços do MN.

MR - MR – os blocos MRs, entre os vários MARs, devem cooperar entre si para que seja possível a criação das conexões dentro do túnel P2MP. Eles realizam essa tarefa através da troca de mensagens pertinentes à estes blocos, a qual é chamada de protocolo de roteamento móvel - MRP (*Mobile Routing Protocol*).

TM - TM – os blocos TM, entre os vários MARs, devem cooperar entre si para gerenciar a topologia da rede sobreposta. Eles realizam essa tarefa através da troca de mensagens pertinentes à estes blocos, a qual é chamada de protocolo de gerência de túnel - TMP (*Tunnel Management Protocol*).

AC - AC – o bloco AC, do lado da rede, deve interagir com o bloco AC, localizado no MN, afim de estabelecer a negociação de um endereço IP. Eles realizam essa tarefa através da troca de mensagens pertinentes à estes blocos, a qual é chamada de protocolo de configuração de endereço - ACP (*Address Configuration Protocol*).

HH - HH – se a funcionalidade deste bloco está espalhada em vários componentes da rede MPA, eles devem interagir entre si para auxiliar no processo de *handover* do MN. Eles realizam essa tarefa através da troca de mensagens pertinentes à estes blocos, a qual é chamada de protocolo de sinalização de *handover* - HSP (*Handover Signaling Protocol*).

[2]A topologia da rede sobreposta pode ser alterar por alguns motivos, como por exemplo, uma ação de gerência ou na presença de falhas.

Note-se que, com exceção da interação HH - AC, ocorrida no MN, e do HH - HH que pode ocorrer entre vários elementos de rede, todas as outras interações ocorrem entre os MARs.

O Plano de Mobilidade é definido como sendo o conjunto de algorítmos e protocolos necessários para suportar as funções e interatividades explicitadas acima. A Figura 3.2 ilustra as interações existentes no plano de mobilidade.

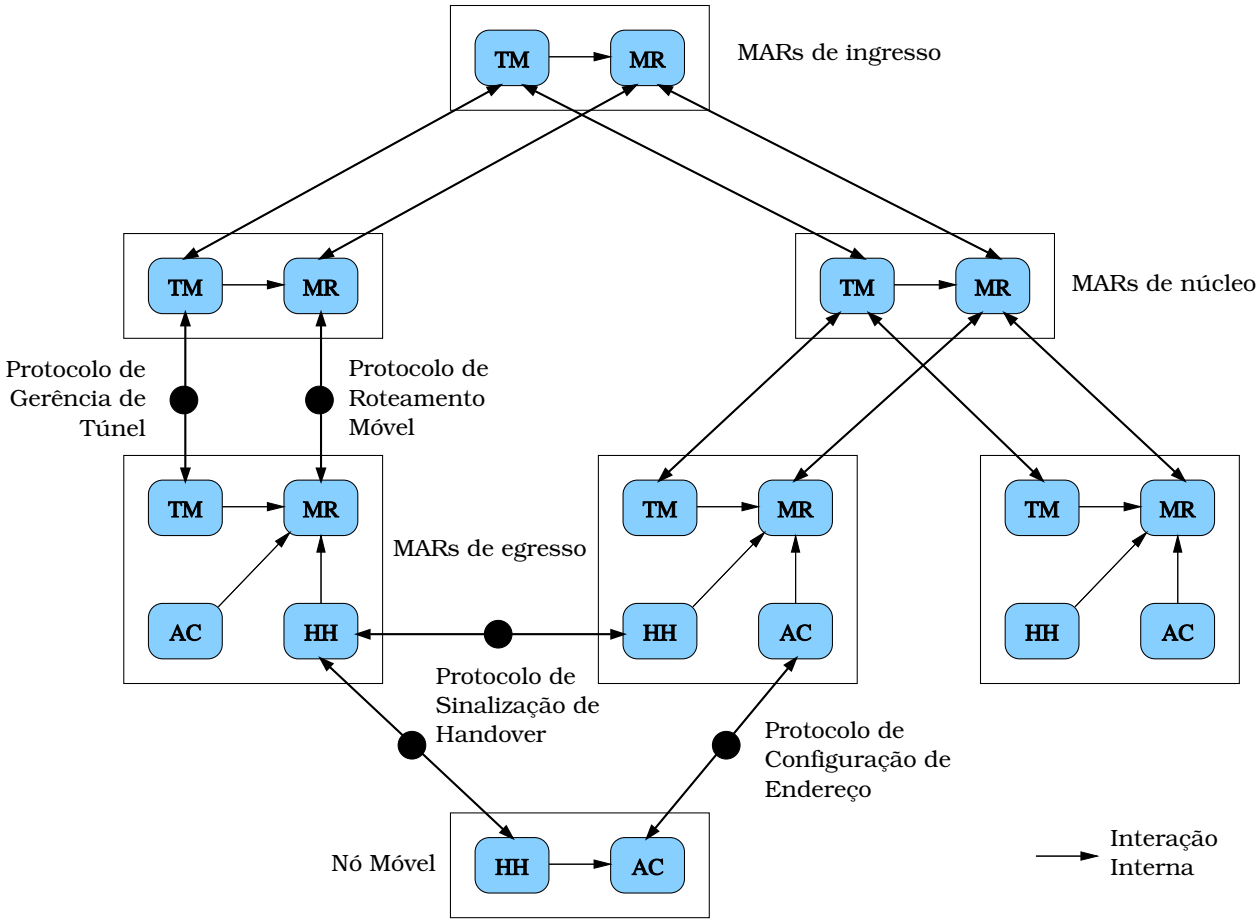


Figura 3.2: Interações entres os blocos funcionais.

O bloco funcional MR pode ser considerado como sendo o mais importante da arquitetura, por ser o mais ativo, ou, em outras palavras, por ser o responsável em receber os pacotes no MAR de ingresso e entregá-los ao MAR de egresso. Obviamente, a sua operação é regida pelo papel que o MAR está atuando em relação a um dado MN. Basicamente, ele é composto do módulo “Tabela de rotas móveis”.

Esse módulo tem por objetivo decidir qual o destino do pacote em um dado MAR. Reafirmando, a escolha do túnel de saída é limitada aos túneis que o bloco TM estabelece como sendo de longa duração. Após o pacote ter sido submetido às funções de roteamento / encaminhamento, existem dois destinos possíveis:

a) re-encapsulá-lo e encaminhá-lo a um segmento do túnel de saída, com destino a um outro MAR; ou

b) retirar as informações de túnel, pertinentes à arquitetura MPA, e submetê-lo ao roteamento IP normal até o seu destino final.

É importante observar que o conteúdo do pacote não se altera nesse processo de tunelamento.

3.2.1 Operação Básica

Podemos descrever a operação da arquitetura MPA sob dois cenários distintos. O primeiro refere-se à dinâmica envolvida quando um MN ingressa pela primeira vez na rede. O segundo se refere à dinâmica quando o MN realiza um *handoff*, sem se ausentar da rede MPA.

Registro na Rede MPA

Quando o MN entra pela primeira vez em uma rede MPA, a sua camada de enlace interage com o ponto de acesso da rede MPA para estabelecer uma associação com ela. Esse evento gera um *trigger*[3] de camada 2, o qual inicia o processo de renegociação de endereço de camada 3. Na arquitetura MPA, essa função é realizada através de interações entre o bloco AC do MN, com o bloco AC do MAR de egresso.

O bloco AC do MAR de egresso, ao receber o pedido de atribuição de endereço por parte o MN, informa o seu bloco MR sobre este evento, o qual incia o processo de atualização da localização do MN, na rede MPA. Assim, o MR envia mensagens de LU (*Location Update*) para os MARs *upstream* ao MN, até o MAR de ingresso. Em cada MAR que ela alcança, uma nova entrada na tabela de rotas móveis é criada, tendo como parâmetro de entrada um valor que identifica o MN e, em sua saída, qual é o segmento *downstream*, ou seja, na direção do MN. A Figura 3.3 ilustra esse processo para um MN que está migrando para uma rede MPA.

Note-se que ao finalizar o processo é criado um caminho comutado, desde o MAR de ingresso até o MAR de egresso, para um dado MN. Esse caminho é comutado no sentido de que, após concluída a sinalização, os MARs comutam os pacotes, retirando-os dos túneis de entrada e colocando-os nos túneis de saída, formando uma conexão fim-a-fim. A Seção 3.4.2 discute uma solução para agregar os MNs, evitando que a tabela de rotas móveis se torne muito grande.

A dinâmica do protocolo é a seguinte: o MAR de ingresso, ao receber um pacote destinado ao MN, pesquisa em sua tabela de rotas móveis por uma entrada que identifique para qual MN

[3] *Trigger* pode ser considerado como um evento que ocorre em camada 2, o qual informa, a quem estiver registrado para recebê-lo, sobre a ocorrência de um *handover*.

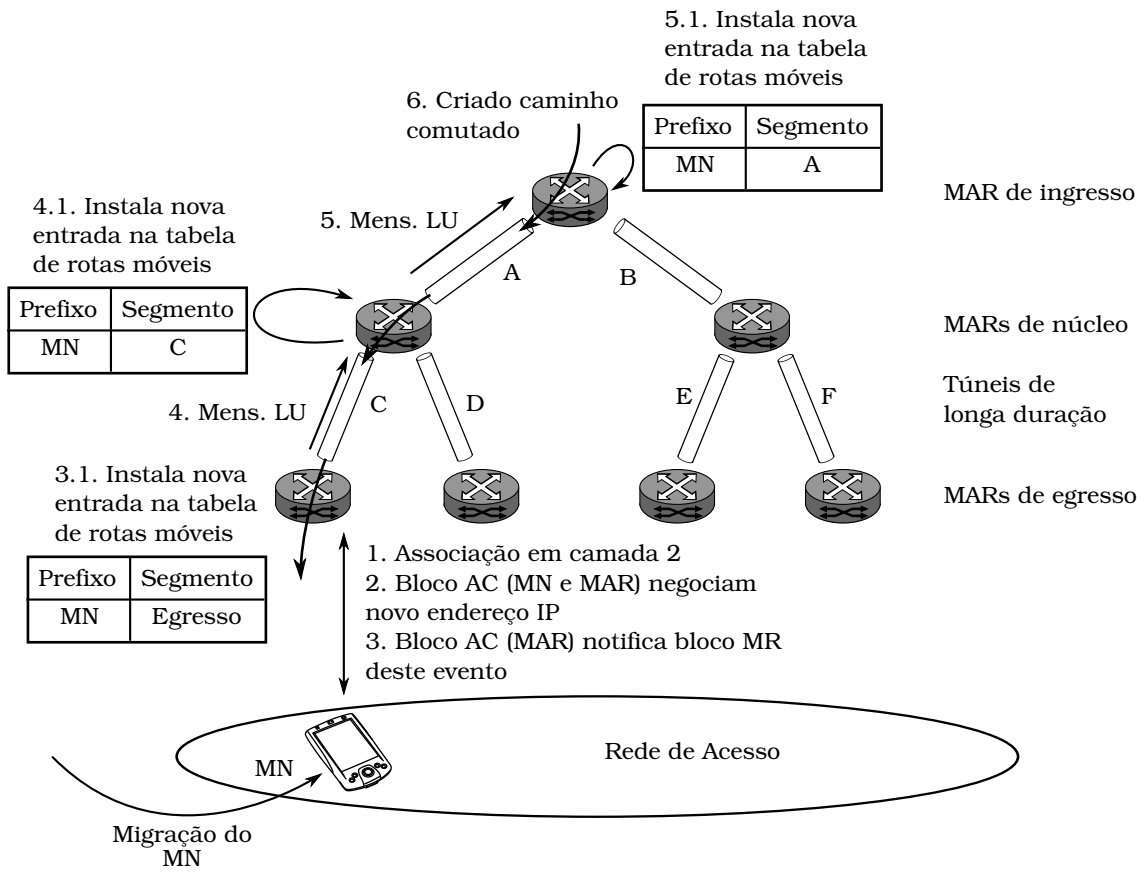


Figura 3.3: Processo de registro de um MN na rede MPA.

o pacote é destinado. Ao encontrar uma entrada, ele verifica qual é o túnel de saída associado, encapsula os pacotes, com um novo cabeçalho pertinente à tecnologia de túnel utilizada, e os entrega para a interface de saída, para que ela os encaminhe ao MAR *downstream*.

O MAR de núcleo, ao receber o pacote, procura por uma entrada, na tabela de rotas móveis, que o identifique e, a partir daí, realiza os mesmos procedimentos descritos para o MAR de ingresso.

O MAR de egresso, ao receber o pacote, procura por uma entrada em sua tabela de rotas móveis e conclui que é egresso para este pacote. Ele então retira o cabeçalho pertinente às informações do túnel, recupera o pacote IP original e, através de procedimentos de camada 2, o encaminha ao MN.

Handoff na Rede MPA

Considerando agora que o MN realiza um *handoff* dentro da rede MPA, as ações executadas, neste cenário, são bastante próximas às executadas quando o MN se registrava na rede MPA. A diferença ocorre quando uma mensagem de LU alcança um MAR no qual já existe uma

entrada para o MN, em sua tabela de rotas móveis. Neste caso, este MAR deve simplesmente atualizar as informações pertinentes ao segmento de saída do MN.

Observe-se que as mensagens de LU continuam seguindo até o MAR de ingresso, ou seja, no sentido *upstream* e isso ocorre, quando o MN está no processo de registro, ou quando o MN está no processo de *handoff*. O objetivo é ou criar uma entrada na tabela de rotas móveis, ou renová-las de tempo em tempo. O motivo desta renovação é devido ao fato delas serem do tipo *soft-state*. Isso significa que se uma mensagem de LU, relacionada a um MN, não chega em um MAR dentro de um certo período de tempo, a entrada desta tabela, pertinente a este MN, é removida automaticamente.

A Figura 3.4 ilustra o processo de atualização da tabela de rotas no MAR, mediante a ocorrência de um *handover*. A Figura 3.5 ilustra o mesmo cenário, sob o ponto de vista de diagrama de sequência, ilustrando as interações entre os vários blocos funcionais.

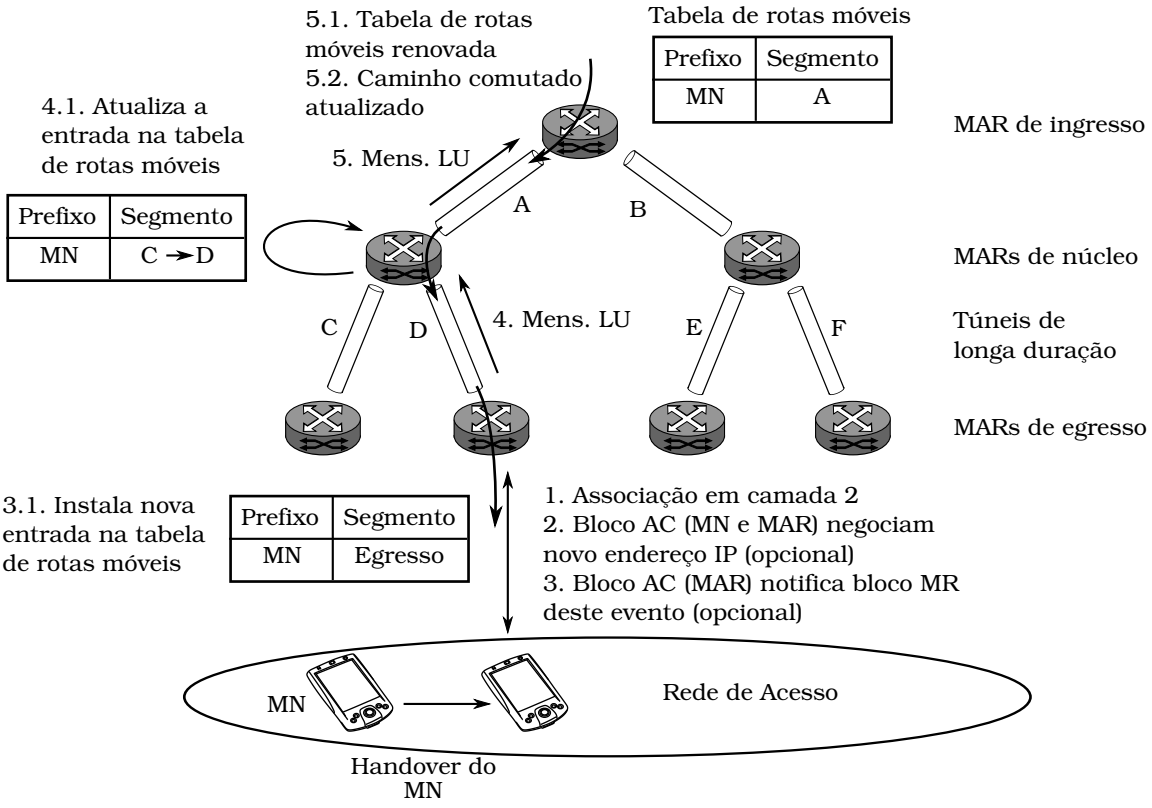


Figura 3.4: Processo de *handover* de um MN na arquitetura MPA.

Note como o caminho comutado se reconfigura para rastrear a atual localização do MN e como a quantidade de eventos[4] gerados na rede MPA se reduz, comparada ao processo de registro. Obviamente essa redução de eventos é altamente dependente de quais MARs estão

[4]Note-se que em um registro, um LU deve instalar uma nova rota para o MN e inicializar o seu temporizador, enquanto que para o *handoff* basta apenas atualizá-lo.

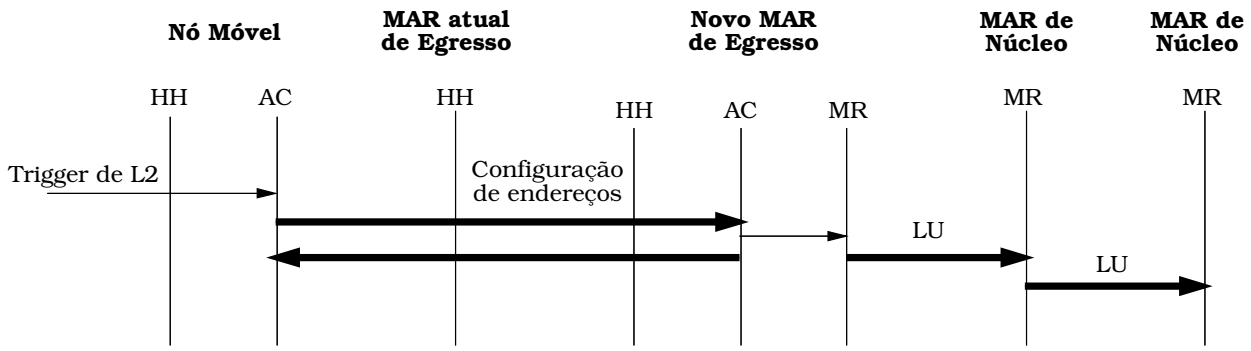


Figura 3.5: Diagrama de sequência, ilustrando as interações entre os vários blocos funcionais, quando ocorre a migração de um MN. As linhas finas representam interações intra-elementos.

envolvidos no processo de *handover*. O pior caso ocorre quando não existem MARs que podem ser compartilhados em uma conexão, degradando o processo na mesma quantidade de eventos de um novo registro. Na Figura 3.4 pode-se perceber essa situação, quando o MN migra do seu atual MAR de egresso para o MAR de egresso que está conectado pelo segmento E ou F.

A vantagem de se utilizar entradas do tipo *soft-state* é que elas mantêm a rede sempre com os estados mais atuais, sem ter entradas desatualizadas. Isso aumenta a escalabilidade, pois permite ter mais elementos presentes no sistema; a eficiência, pois mantem as estruturas de dados em seu tamanho mínimo; e facilita a sua gerência. A desvantagem é que manter esses estados demandam mensagens de controle, que devem trafegar na rede, e uma memória física condizente com a quantidade de estados que devem ser atualizados.

Para que esse esquema funcione, o MAR de egresso, no qual o MN está associado, deve gerar, de tempos em tempos, uma mensagem de LU, em seu nome, para disparar o processo de renovação de sua entrada na tabela de rotas móveis. Entretanto, uma mensagem de LU só deve ser gerada se o MN realmente estiver associado a este MAR. Assim, todo o processo deve ser iniciado pelo MN, quando o seu bloco AC inicia o processo de renovação de endereço com o bloco AC do MAR de egresso. Um modo de garantir que esta renovação, na alocação de endereços, ocorra antes que as rotas esgotem o seu tempo limite é impor que o tempo de alocação do endereço IP do MN seja menor que o tempo máximo de renovação dos estados das rotas.

Embora a versão da arquitetura MPA descrita nessa tese utilize as entradas de *soft-state*, a versão atual da arquitetura MPA emprega *trigger* L2 de conexão e de desconexão, que geram mensagens LU de conexão e de desconexão, respectivamente. Essas mensagens permitem a criação dos segmentos de rota atual e a destruição dos segmentos de rota antigos, sem a necessidade de *soft-state* e renovação frequente de endereços no nó móvel.

3.3 Questões de Implementação

Para que o plano de mobilidade seja de interesse prático, existe um conjunto de requisitos que são obrigatórios de serem atendidos. Por outro lado, existe também um conjunto de requisitos que são opcionais e que, se forem atendidos, enriquecem a implementação. Os requisitos obrigatórios são:

- Suportar tanto o protocolo IPv4, quanto o IPv6;
- Não necessitar de alterações na pilha de protocolos IP do MN;
- Ter uma coexistência pacífica entre os vários serviços IP e outros existentes, tanto os atuais quanto os futuros, por exemplo, o protocolo MIP, a rede VPN (*Virtual Private Network*) e a aplicação VoIP;
- Permitir uma associação segura do MN;
- Facilitar o *handover*.

Os requisitos opcionais são:

- Permitir múltiplos esquemas de tunelamento, como por exemplo, o IP/IP e o MPLS;
- Suportar qualidade/classes de serviços;
- Permitir o uso de funções de engenharia de tráfego.

Os requisitos obrigatórios são alcançados ao estabelecer que os protocolos pertinentes à arquitetura MPA utilizem de protocolos da Internet, que já são bem estabelecidos valendo-se, entretanto, de algumas extensões permitidas. A arquitetura MPA exige que o MN utilize-se apenas de protocolos usuais para este tipo de dispositivo.

3.3.1 Protocolos Pertinentes aos Blocos Funcionais

Nesta seção, vamos comentar sobre alguns protocolos, nativos à pilha de protocolos TCP/IP, que são possíveis de serem utilizados para transportar, ou realizar, as funções pertinentes a cada bloco funcional.

Protocolo de Gerência de Túneis

Este protocolo deve permitir a criação de túneis P2MP em um rede IP. Ele deve lidar com a criação, destruição e reconfiguração dos túneis. Deve lidar também com a adição e remoção de segmentos de túneis em um túnel já estabelecido. Um possível protocolo é o RSVP-TE

(*Resource ReserVation Protocol-Traffic Engineering*) [58], o qual foi adicionada a capacidade de gerência de túneis P2MP para os túneis LSPs [59, 60].

Outras alternativas possíveis são protocolos proprietários via aplicativos de linha de comando, chamados de CLI (*Command Line Interface*), o SNMP (*Simple Network Management Protocol*), ou o HTTP (*HyperText Transfer Protocol*).

Protocolo de Roteamento Móvel

Este protocolo deve permitir a criação de um caminho comutado dentro da rede MPA. Ele difere do protocolo de roteamento IP, no sentido de que somente deve ser executado na rede sobreposta. Assim, embora ele possa se utilizar dos protocolos usuais da camada de rede, ao inserir informações de localização do MN de forma opaca[5], como por exemplo usando um LSA (*Link State Advertisement*) opaco do OSPF (*Open Shortest Path First*), existem alternativas mais eficientes.

O problema do OSPF é que ele inunda toda a rede com informações de roteamento IP, para manter todos os roteadores de um domínio com o mesmo estado de rotas. Já o protocolo de roteamento móvel não precisa distribuir as informações de localidade do MN a todos os MARs da rede, mas somente aos que estão no caminho *upstream* a ele.

Assim, ao utilizar o OSPF toda vez que o MN realiza um *handover*, o OSPF teria que inundar toda a rede, com LSAs, para atualizar a localização do MN. Portanto, essa seria uma solução viável para rede pequenas, ou com baixos níveis de tráfego.

Uma solução mais eficiente é utilizar os protocolos de gerência de túneis para realizar também essa tarefa. A vantagem é que esses protocolos já conhecem os MARs que compõem a rede sobreposta e, ao inserir as mensagens do protocolo de roteamento móvel nestes protocolos, somente os MARs que compõem a rede sobreposta seriam atingidos. No melhor caso, somente os MARs no caminho *upstream* seriam atingidos por essas mensagens.

O RSVP-TE [58] pode ser utilizado para essa finalidade. Ele permite que as mensagens `RESV` carreguem um objeto opaco, que, nesse caso, seria a localização do MN. Uma segunda vantagem, é que as mensagens `RESV` percorrem o caminho *upstream* do MN, reduzindo a quantidade de mensagens de gerência na rede.

Protocolo de Configuração de Endereço

Existem pelo menos duas formas de realizar a alocação de endereços para o MN: pode ser do tipo *stateless* ou do tipo *statefull*. No modo de alocação *stateless* o MN se auto atribui um endereço IP e verifica se ele é único na rede em que está se associando [61]. No modo *statefull*, existe uma entidade na rede que é responsável em prover um endereço IP ao MN,

[5] O termo técnico para esse procedimento de inserir informações de forma opaca em mensagens é *piggyback*.

dentro de uma faixa de endereços disponíveis. A vantagem deste tipo de endereçamento é a possibilidade de manter no MN o mesmo endereço a cada *handover* efetuado. Isso preserva as conexões de camada de transporte ou superiores. A excessão à regra são as conexões TCPs, quando o MN não está usando um protocolo de macromobilidade e o seu *handover* é do tipo *break-before-make*[6]. O DHCP [56, 57] é o protocolo proposto pelo IETF para configuração de endereços.

O protocolo de configuração de endereço pode, então, se utilizar do DHCP para realizar a sua tarefa. Assim, dentro de um domínio teremos um servidor de DHCP e, em cada MAR de egresso, teremos um *proxy* para este servidor. Conforme já elucidado anteriormente, é responsabilidade do MN iniciar o processo de renovação de suas entradas nas tabelas de rotas móveis. Para garantir isso, o servidor de DHCP deve ser configurado para ter um tempo de alocação de endereços curto e menor do que o tempo de expiração dos *soft-states* (caso *triggers* de desconexão não sejam utilizados).

Protocolo de Sinalização de *Handover*

Este protocolo deve permitir que os blocos HH troquem informações entre si, para antecipar um *handover*. A RFC 4068 (*Fast Handovers for Mobile IPv6*) [62] estabelece um protocolo de *handover* de camada 3 para redes IPv6, suportanto o MIPv6. Embora não haja um equivalente deste protocolo para uma rede IPv4, não é complicado adaptá-lo para funcionar sobre ela.

Outras alternativas são os protocolos que empregam *triggers* de camada 2, que possam notificar as camadas superiores sobre as atividades de *handover*, ou até mesmo protocolos proprietários.

3.3.2 Protocolos Utilizados na Arquitetura

Dentre as várias possibilidades de protocolos, para cada bloco funcional, nesta seção, vamos apresentar aqueles que foram utilizados para implementar o protótipo da arquitetura MPA. Note-se que as soluções empregadas podem ser aplicadas para o IPv4 e para o IPv6.

RSVP-TE

O protocolo RSVP-TE, com as extensões P2MP, foi utilizado para implementar tanto o protocolo de gerência de túneis, quanto o protocolo de roteamento móvel.

[6]Existem, basicamente, dois tipos de *handover*, o *make-before-break*, no qual o MN se associa ao ponto de acesso de destino, antes de se desfazer da conexão antiga e o *break-before-make*, no qual o MN deve se desassociar do seu atual ponto de acesso, antes de se associar ao novo ponto de acesso. Do ponto de vista de conexões de camada 4, o *handover break-before-make* necessita fechar todas as conexões abertas, se desfazer do endereço IP associado em sua interface, antes de iniciar uma nova associação no novo ponto de acesso. Isso implica em quebra das sessões já estabelecidas. Para preservá-las, a interface de rede deve ter um endereço IP mais estável, normalmente proveniente de um protocolo de macromobilidade.

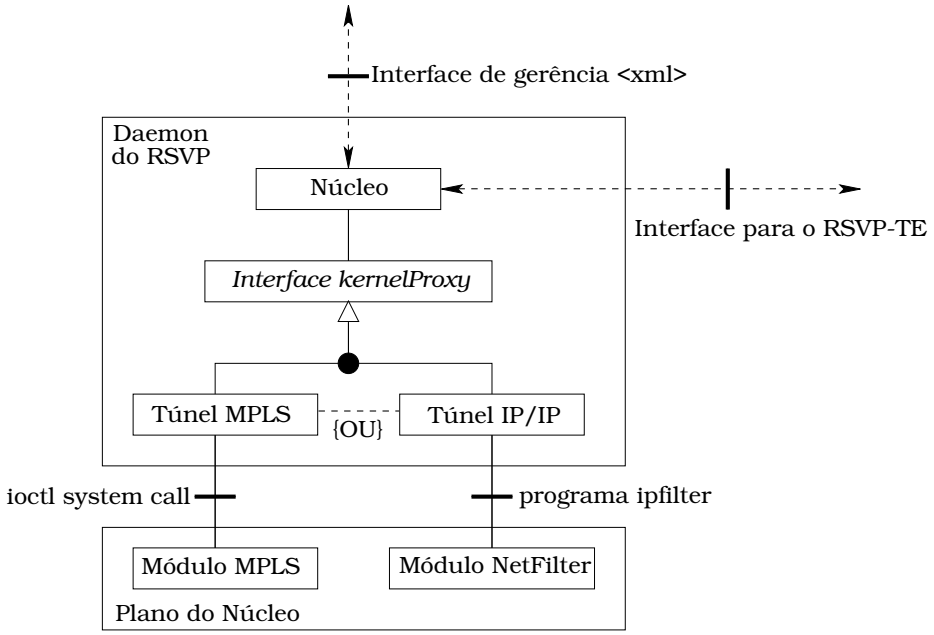


Figura 3.6: Integração entre os componentes principais da implementação da sinalização do RSVP, ou seja, das tecnologia de tunelamento e das interfaces de sinalização e gerência.

Com relação ao protocolo de gerência de túneis, a implementação da sinalização foi realizada para atender os seguintes requisitos:

1. Prover as extensões de engenharia de tráfego, permitindo roteamento baseado em restrições;

2. Prover as extensões para gerência de túneis ponto-multiponto;

3. Prover as extensões principais do protocolo GMPLS, em particular, rótulos genéricos, redução do *overhead* de renovação dos *soft-states* [63], interfaces não nomeadas [64] e o estabelecimento de LSPs bidirecionais [65].

Uma das características interessantes desta implementação é que ela desacopla as funções de sinalização das funções de encaminhamento, ao definir uma interface de ligação entre ambas. Com isso, ela permite que qualquer tecnologia de tunelamento seja utilizada. Em particular, as duas principais são: IP/IP e o MPLS. A Figura 3.6 ilustra essa divisão e mostra, também, os módulos chamados para cada tecnologia de túnel.

As funções de gerência providas por essa interface são: criação de túneis, remoção de túneis, inserção do novos segmentos de túneis e remoção de segmentos de túneis.

Com relação ao protocolo de roteamento móvel, foi utilizada uma extensão que permite a inclusão de objetos opacos, para rastrear a localização do MN. Este objeto é chamado de objeto

de localização do nó móvel e é transportado nas mensagens de RESV, as quais são geradas nos MARs de egresso.

Conforme pode ser visto na Figura 3.7, o objeto transporta o ID do MN, por exemplo, o endereço MAC de sua interface aérea, a versão do protocolo de camada de rede, o comprimento do prefixo de rede e o seu conteúdo, o identificador do MAR de egresso que está servindo o atual enlace do MN e um conjunto de *flags*.

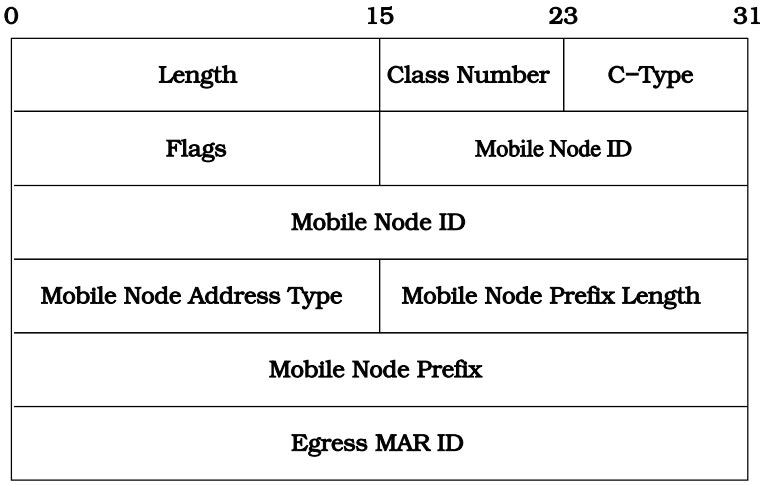


Figura 3.7: Objeto de localização do nó móvel.

Note-se que o uso de um prefixo de rede, ao invés de um endereço completo, permite que um objeto simples refira-se a um conjunto de nós móveis, se estiver sendo utilizado agregação de mobilidade.

DHCPv4 e DHCPv6

Tanto o DHCPv4 [56], quanto o DHCPv6 [57] foram utilizados para implementar as funções previstas no protocolo de configuração de endereços. A implantação destes elementos foi dividida em três partes: um servidor central, localizado em algum elemento de rede do domínio MPA; um servidor *proxy* instalado em cada MAR de egresso, para responder às requisições de atribuição de endereço, em nome do servidor central; e os clientes instalados nos MNs.

O uso de tais servidores traz algumas vantagens associadas, como por exemplo:

1. Fácil integração com os padrões de autenticação, por exemplo o RADIUS (*Remote Authentication Dial In User Service*) [66];

2. A associação dos MNs, na rede MPA, é mais controlável, por exemplo, o número de MNs e o espaço de endereços utilizados, em um enlace, pode ser limitado;

3. O MN pode informar ao servidor DHCP sobre as suas preferências, utilizando a opção `User Class` em uma mensagem de solicitação ou requisição.

Com relação às redes IPv4, a resposta do DHCPv4 supre ao MN o prefixo de rede utilizado e o roteador padrão de saída. É possível também utilizar uma comunicação segura, no processo de configuração, ao utilizar uma integração entre o DHCPv4 e o IPsec.

Com relação às redes IPv6, a resposta do DHCPv6 somente supre ao MN o seu endereço IP, sem informar o prefixo de rede ou mesmo o roteador padrão. Essas informações podem ser adquiridas através das mensagens de anúncio de descobrimento de roteador de rede (*Network Discovery Router Advertisement messages*) [34].

Note-se que a arquitetura MPA estabelece que existe uma comunicação entre o bloco AC e o bloco MR, em cada MAR de egresso, para que seja gerada uma mensagem de LU quando o MN realiza uma requisição de endereços. Essa comunicação foi implementada através do uso de um interceptador, instalado em cada MAR de egresso, o qual tem a função de capturar as mensagens de retorno, gerados pelo servidor de DHCP, antes que elas sejam entregues ao MN.

Este interceptador interage com o `daemon` do RSVP-TE, informando-o sobre o ID do MN, dos dados pertinentes ao endereço de camada 3 e dos *flags* associados. O `daemon`, de posse dessas informações, gera uma mensagem de LU e a envia aos MARs *upstream*, renovando as entradas pertinentes a este MN.

3.3.3 Tunelamento IP/IP

Conforme já mencionado anteriormente, uma das tecnologias aptas para implementar os túneis previsto na arquitetura MPA são os túneis IP, também chamados de túneis IP/IP.

O tunelamento IP/IP pode ser realizado através do GRE (*Generic Routing Encapsulation*) [19]. GRE é um protocolo de tunelamento bastante versátil e suporta túneis do tipo IPv4 sobre IPv4 ou IPv6 e IPv6 sobre IPv4 ou IPv6. Assim, a rede de acesso e a rede de transporte podem empregar diferentes versões do protocolo IP.

Os segmentos de túneis GRE são estabelecidos através das mensagens de sinalização do RSVP-TE. As mensagens de `PATH` estabelecem qual é o ponto de destino final do segmento para os MARs no caminho *downstream*, enquanto que as mensagens de `RESV` estabelecem qual é o destino final do segmento para os MARs no caminho *upstream*.

Os túneis IP/IP foram implementados em roteadores Linux e, para esse sistema operacional, os segmentos de túneis são representados por uma interface de rede virtual, chamada, por exemplo, de `tun0`. Assim, as entradas na tabela de rotas mapeiam os prefixos/endereços dos MNs para esta interface.

A vantagem do tunelamento IP/IP é que as funções de roteamento e encaminhamento do tráfego do MN não se diferem das funções de roteamento e encaminhamento do tráfego IP,

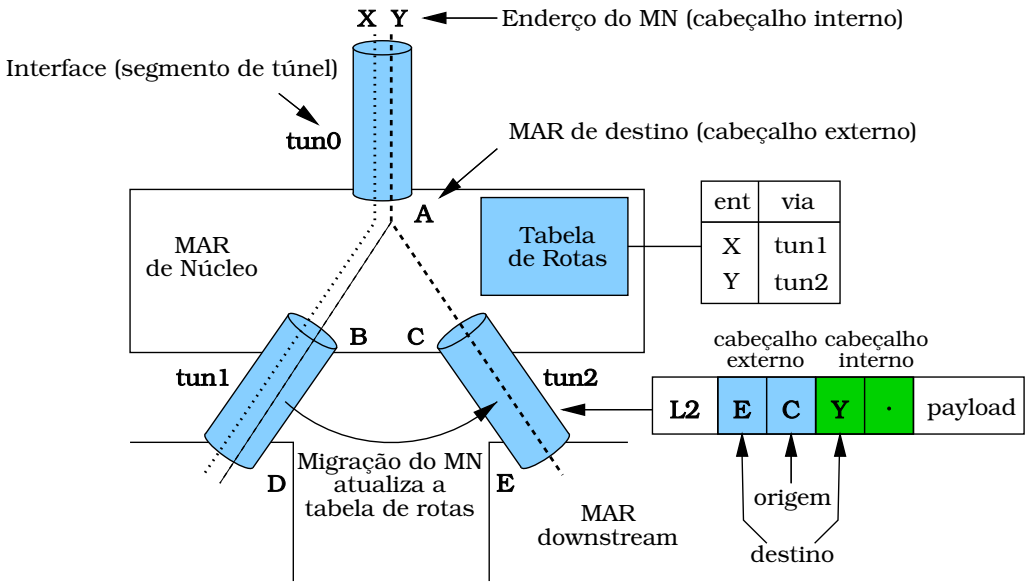


Figura 3.8: Esquema de tunelamento IP/IP.

dito regular. Mais precisamente, a tabela de rotas móveis é integrada com a tabela de rotas existente. Com isso, a função de encaminhamento de pacotes, já implementada pelo núcleo do roteador, permanece inalterada. A Figura 3.8 ilustra o esquema de tunelamento IP/IP. É uma descrição bastante concisa do esquema de tunelamento, pois não é foco principal desta tese.

3.4 Variações da Arquitetura Básica

O básico da arquitetura foi discutido nas seções anteriores e permite contemplar a maioria dos casos encontrados na prática. Nas próximas seções, serão discutidas alguns casos especiais, que darão à arquitetura uma maior flexibilidade ao retirar algumas restrições existentes na proposta básica.

Em particular, serão discutidas três cenários distintos, o primeiro que trata de como é possível aumentar a eficiência do *handover*, ao considerá-lo como um agente ativo no sistema; o segundo que considera a utilização de múltiplas faixas de endereços na rede de acesso e a possibilidade de agregação como forma de aumentar a escalabilidade; e, por fim, retirar a premissa que existe apenas um MAR de ingresso.

3.4.1 *Handover* Pró-Ativo

O modo como um *handover* é efetuado nos permite classificá-lo sobre dois grupos distintos, o *handover* reativo e o *handover* pró-ativo.

Um *handover* reativo ocorre quando a rede só sabe de sua realização após a sua ocorrência, ou seja, quando o MN tenta se associar ao ponto de acesso para o qual está migrando. O efeito desse cenário é que a rede MPA não tem como se preparar para receber o MN, em sua nova rede de acesso, uma vez que todas as ações são tomadas após a realização do *handover*.

Assim, o valor mínimo de latência que pode ser alcançada e a quantidade de pacotes perdidos é limitada pelo tempo que o MN leva para se associar à nova rede, adicionada com o tempo que a rede MPA leva para se reconfigurar diante do novo cenário. Percebe-se que esse efeito é causado devido a seriação inerente a este processo. Note-se que é esse tipo de *handover* que tem sido discutido nas seções anteriores.

Por outro lado, um *handover* pró-ativo ocorre quando algum elemento da rede, ou mesmo o próprio MN, avisa ao MAR de destino do MN sobre o seu iminente *handover*. Isso permite que a rede MPA se prepare para a chegada do MN, antecipando algumas ações que podem ser paralelizadas no processo. Isso implica que o tempo que o MN leva para se associar ao seu novo ponto de acesso é gasto, simultaneamente, tanto nesse processo de associação, quanto no processo de reconfiguração da rede MPA. O resultado acaba sendo em valores perto do mínimo para a latência e para as perdas de pacotes.

Existem várias formas de se realizar um *handover* pró-ativo e elas foram previstas, na arquitetura MPA, através das interações entre os blocos funcionais HH. A forma como um *handover* pode ser previsto e notificado para a rede MPA está fora do escopo desta tese, mas como ilustrações, pode-se citar:

- Um veículo em movimento. É fácil perceber quando vai ocorrer um *handover*, pois, normalmente, tem um trajeto bem definido sobre um trecho monitorado;
- Histórico de mobilidade. Se em um dado momento o MN está sempre em um determinado local, pode se antecipar a sua migração;
- O próprio MN possui recursos suficientes para se associar em mais de um ponto de acesso, portanto sabendo exatamente quando irá utilizar o novo ponto de acesso;
- A rede pode monitorar a potência do sinal de rádio do MN, entre vários pontos de acessos distintos, e prever para onde ele irá se associar, através da comparação entre os níveis analisados.

Considerando-se que o evento *handover* foi capturado com antecedência, a rede MPA pode se reconfigurar para receber o MN em seu nova localidade. Para fazer isso, as seguintes interações, entre os blocos funcionais, são realizadas, conforme pode ser visto na Figura 3.9.

1. Uma notificação de evento de *handover* é recebida pelo bloco HH do MN;

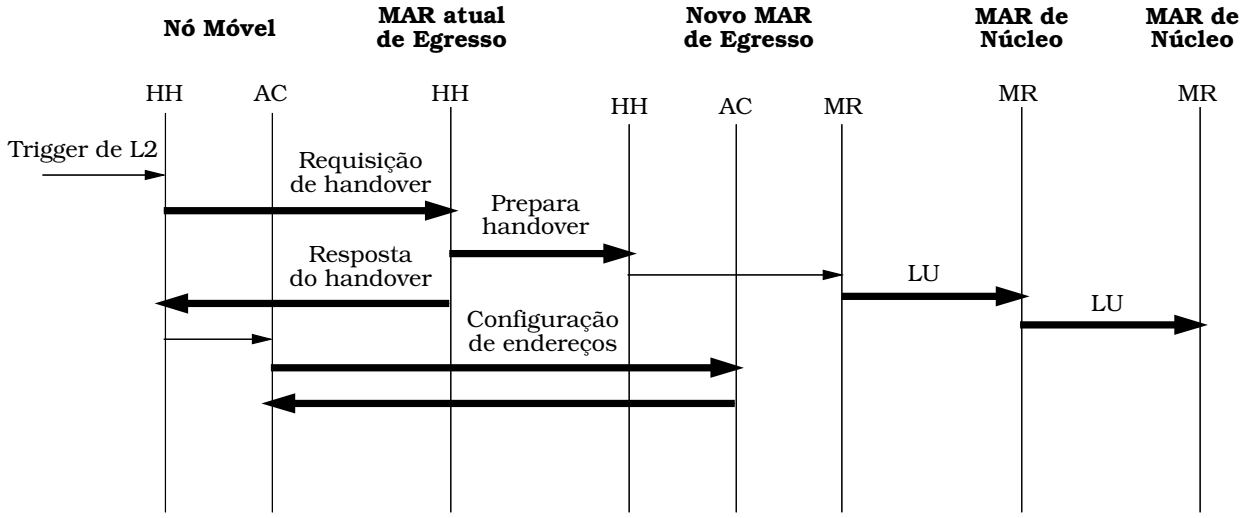


Figura 3.9: Diagrama de sequência mostrando as interações entre os blocos funcionais para realizar um *handover* pró-ativo. As linhas finas representam interações intra-elementos.

2. Esse bloco notifica o bloco HH do MAR de egresso atual informando para onde o MN está migrando;

3. O MAR de egresso atual então contacta o futuro MAR de egresso do MN, requisitando que ele se prepare para um *handover*;

4. Este MAR contacta o seu bloco MR para notificar a rede sobre a nova possível localização do MN e reconfigurá-la adequadamente. Mensagens de LU são enviadas na direção *upstream*;

5. Quando essa mensagem de LU alcança o MAR de núcleo, comum aos dois segmentos, a replicação de pacotes se inicia e o caminho comutado do MN, que era P2P, passa a ser um caminho comutado com replicação de pacotes. Isso é necessário pois o MN ainda é alcançável pelo MAR atual, mas a qualquer momento vai ser alcançável também pelo MAR futuro. Note-se que essa replicação de pacotes tem um tempo de vida curto, para não sobrecarregar as redes de acesso.

6. Quando o MN efetivamente se associar na nova rede, seu bloco AC irá interagir com o bloco AC do novo MAR de egresso;

7. Ao finalizar o processo de requisição de endereço, o MN já está recebendo os pacotes destinados a ele;

8. Devido à expiração do *soft-state* ou da mensagem LU de desconexão do segmento antigo, ele será removido do caminho comutado pertinente ao MN, tornando-o outra vez, um caminho comutado P2P.

A replicação de pacotes não é um problema para o MN, uma vez que o protocolo IP já contempla que pacotes podem ser duplicados e recebidos fora de ordem.

3.4.2 Agregação de Endereços

A arquitetura MPA assume que, em qualquer posição na rede de acesso, o prefixo de endereço é único. Isso facilita o projeto da arquitetura e sua implementação, ao estabelecer que somente um túnel P2MP precisa ser criado e mantido em um domínio. A desvantagem desta solução é a impossibilidade de agregar MNs que seguem o mesmo caminho comutado, desde o MAR de ingresso até o MAR de egresso. O motivo é que todos os roteadores de acesso compartilham o mesmo prefixo de rede e, portanto, o MN deve ser rastreado de forma individual. Obviamente, sem agregação, a escalabilidade do sistema é comprometida.

Uma possível solução é a utilização de um roteador NAT móvel, o qual expõe apenas um endereço de rede externamente e esconde todo um rol de MNs. Embora o objetivo inicial seja alcançado, que é diminuir o tamanho da tabela de roteamento móvel, a flexibilidade também é limitada. Essa solução obriga que todos os MNs ou fiquem confinados em uma área que seja coberta por este NAT; ou que caminhem juntos, impedindo que eventualmente um ou outro siga um caminho diferente. Um exemplo em que essa solução seria bem adequada é na implantação sobre transportes públicos, de longa duração, os quais possuem uma rota sempre bem definida.

Uma alternativa mais flexível é realizar a agregação de endereços na própria rede MPA. A construção do túnel P2MP se mantem, conforme descrito anteriormente, e o fato de que toda a rede de acesso tenha o mesmo prefixo de rede também. A ideia consiste em se criar regiões na rede de acesso em que um subconjunto de endereços seriam atribuídos aos MNs que se associassem nela. O único elemento que seria responsável em conhecer essa divisão da rede em regiões seria o servidor DHCP.

Vamos mostrar a dinâmica dessa solução através do exemplo ilustrado na Figura 3.10, no qual foi criado uma rede MPA, cujo prefixo da rede de acesso é 10.0.0.0/8 e foi dividida em quatro regiões.

O túnel P2MP é criado para o prefixo da rede de acesso e, devido à agregação, a tabela de rotas móveis foi dividida em faixas. As entradas dessas tabelas são estáticas e permanentes, não precisando de renovação para manter o seus estados. Assim, sempre que um MN entra na região 1, já está apto a receber pacotes provenientes do MAR de ingresso, sem a necessidade de reconfigurar a rede MPA. Em cada MAR de egresso existe um servidor *proxy* DHCP, que atende às requisições de obtenção de endereço por parte do MN e as remete a um servidor DHCP central, o qual libera um endereço dentro da faixa permissível à região solicitada.

Isso permite ter vários MNs confinados em uma região[7] de acesso, sem ter que alterar

[7]Uma região pode ser composta de vários pontos de acesso, cobrindo uma área física qualquer.

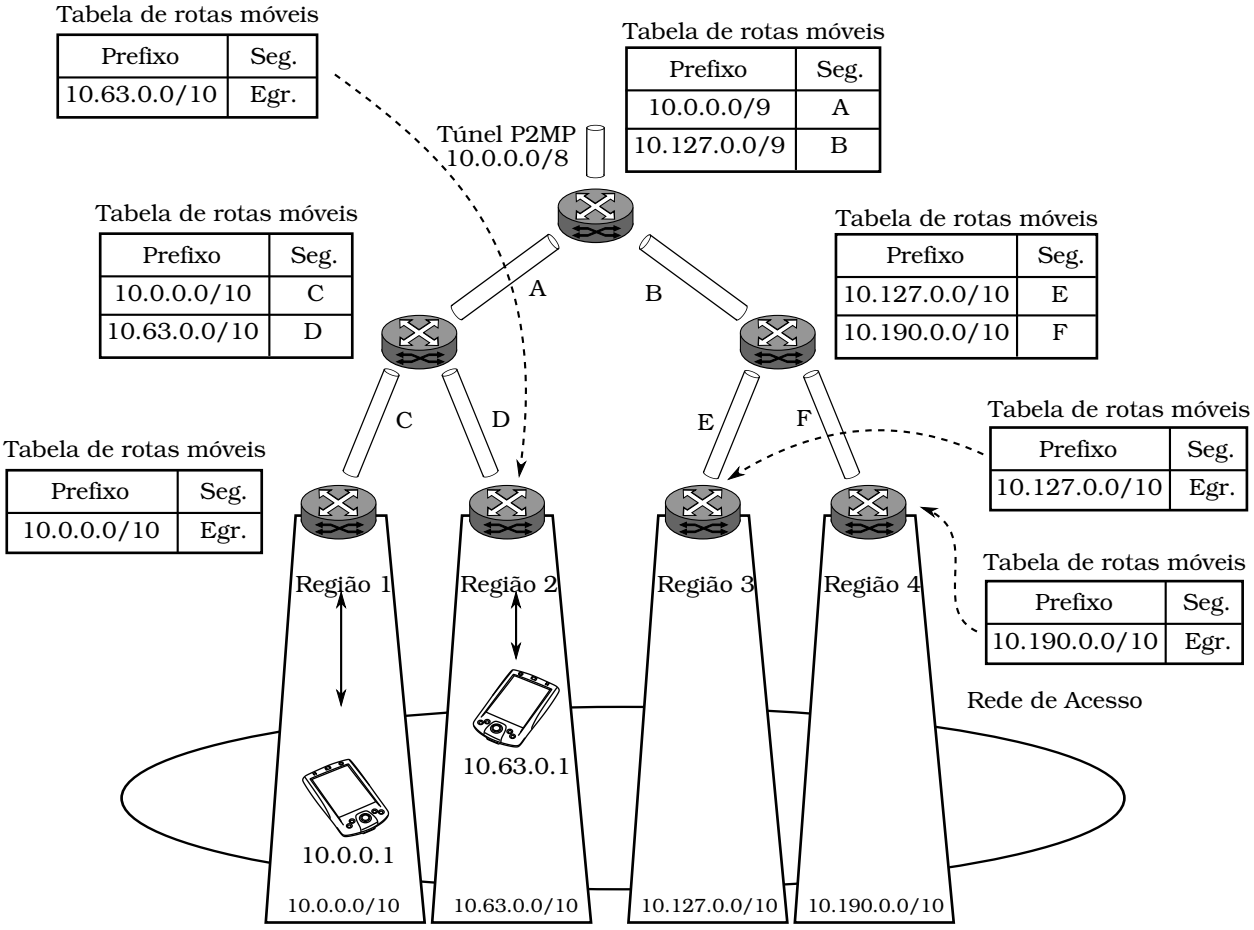


Figura 3.10: Agregação de rotas na arquitetura MPA. As entradas mostradas na tabela de rotas móveis são permanentes.

qualquer entrada nas tabelas de rotas móveis dos MARs *upstream*.

Eventualmente, alguns MNs irão migrar para outras regiões. Essa situação é ilustrada na Figura 3.11, onde um MN migra para uma região diferente da qual ele se registrou na rede MPA.

O endereço IP do MN não se altera devido à mudança de região. Isso é garantido devido à existência do servidor DHCP central. Uma vez que o MN já se associou a uma região qualquer, o DHCP irá manter esse endereço enquanto o MN estiver ativo na rede MPA.

Assim, quando o MN deixa a sua região original, ele perde a agregação e, para ser rastreado dentro da rede MPA, é necessário instalar rotas individuais nas tabelas de rotas móveis nos MARs *upstream*. Essas entradas seguem exatamente a mesma dinâmica já discutida anteriormente, ou seja, necessitam ser renovadas periodicamente, e estão representadas em negrito, para diferenciar das rotas estáticas.

A agregação de endereços tem um desempenho mais elevado quanto mais os MNs se man-

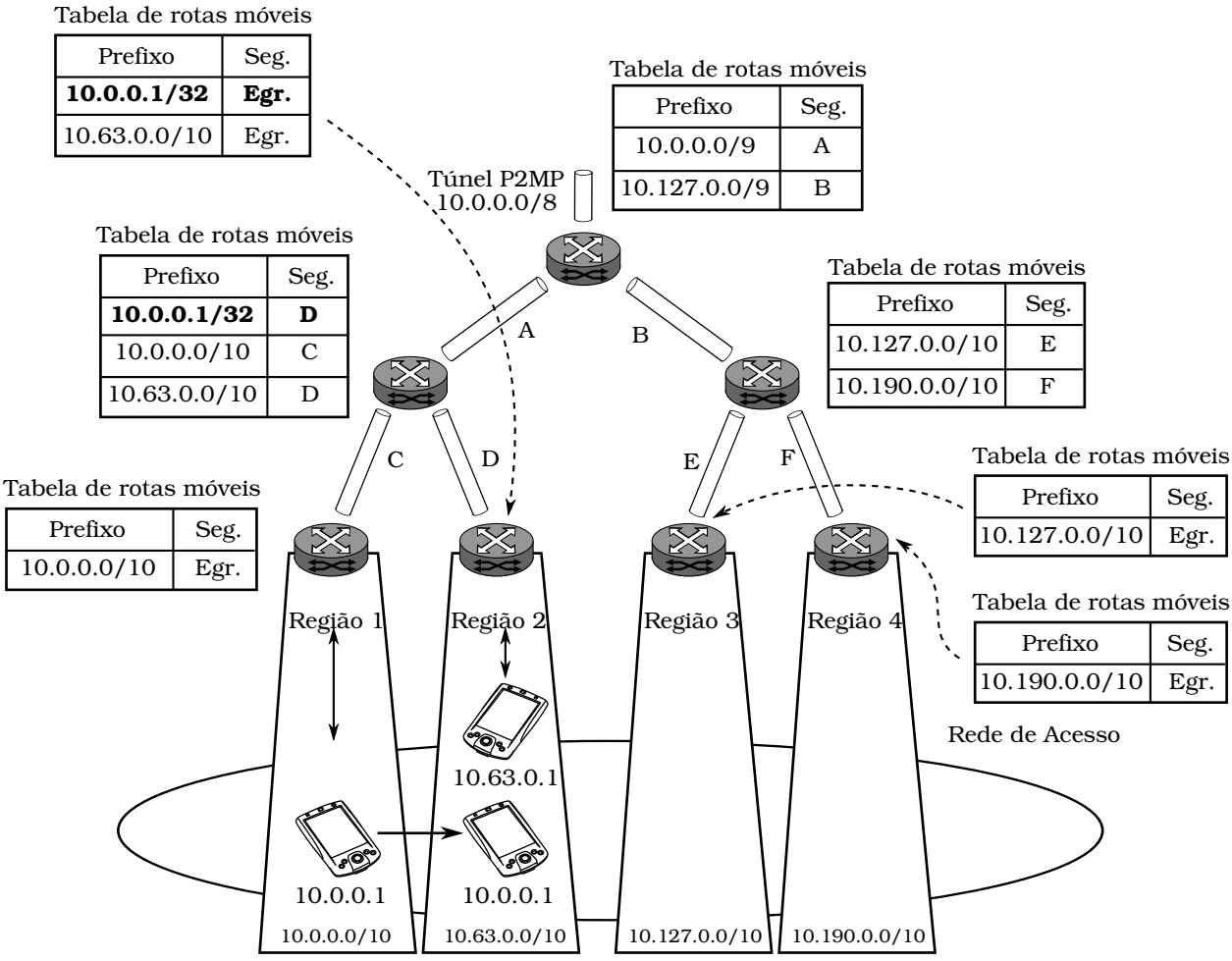


Figura 3.11: Migração de MN em um cenário de agregação de rotas na arquitetura MPA. As entradas mostradas em negrito, na tabela de rotas móveis, refletem as alterações efetuadas pelo sistema.

tiverem agrupados em suas regiões de inserção na rede MPA. O sistema se degrada para o cenário básico estudado anteriormente, quanto maior for a dispersão entre os MNs. Uma situação típica em que esse cenário é bem contemplado são as regiões onde existe uma grande concentração de elementos móveis, mas com baixa mobilidade, como por exemplo: eventos, congressos, exposições, aeroportos, etc.

Existem outros cenários igualmente interessantes, que acabam sendo variações da ideia principal, como por exemplo, permitir que um MAR de egresso seja móvel, que, por consequência, acarretaria em uma região móvel. Nesse caso, teríamos uma mobilidade agregada, no qual vários MNs seguiriam o movimento do MAR de egresso, mas permitiriam ainda que os MNs que deixassem a agregação não perdessem os serviços de micromobilidade. Por exemplo, um MAR de egresso dentro de um trem.

Por fim, em uma rede bem planejada, a mudança de uma região para a outra afetaria poucos MARs no caminho *upstream*. Essa situação é bem visível quando se observa a migração do MN que está na região 1 para região 2, enquanto que um MN que migra da região 2 para a região 3, afeta uma quantidade maior de MAR em seu caminho *upstream*. Como esses túneis de longa duração podem ser reconfigurados, como uma ação de gerência, é possível, através de procedimentos de engenharia de tráfego, otimizar os recursos da rede MPA.

3.4.3 Múltiplos MARs de Ingresso

Embora considerar que existe somente um único MAR de ingresso cubra o cenário sobre a operação de várias redes de dados na atualidade, isso não é a totalidade. Em alguns casos, pode-se ter vários MARs de ingresso para um mesmo domínio. O problema desse arranjo é que cada MAR de ingresso possui um túnel P2MP com os MARs de egresso e, como todos os MARs de egresso possuem o mesmo prefixo de rede, um pacote destinado a um MN que pertence a um dado túnel P2MP de um MAR de ingresso qualquer, pode chegar através de um outro MAR de ingresso, no qual o MN não está associado.

Para resolver o problema, deve existir uma forma para que os túneis P2MP existentes no domínio comuniquem entre si. Isso pode ser feito através da criação de túneis P2MP entre os MARs de ingresso ou entre os MARs de egresso. Como, normalmente, os MARs de ingresso são em menor número, é mais interessante a criação destes túneis nesses MARs.

A dinâmica da solução é a seguinte: cada MAR de ingresso, quando entra em operação, cria um túnel P2MP com os outros MARs de ingresso. Sempre que um pacote, destinado a um MN, chega em um MAR de ingresso qualquer e não possui uma entrada em sua tabela de rotas móveis, ele deve encaminhar o pacote através deste túnel.

Os outros MARs de ingresso, ao receber esse pacote, verificam se o MN está registrado em sua tabela de rotas móveis, caso negativo, simplesmente descartam o pacote. Em caso afirmativo, ele encaminha o pacote pelos MARs de *downstream* e envia uma mensagem de `ICMP redirect` para a fonte do pacote, solicitando que ele atualize a sua tabela de rotas. Talvez nesse ponto, haja a necessidade de interação, através de um módulo chamado, por conveniência, gerador de mensagens de ICMP, para que uma mensagem de `redirect` sempre seja enviada quando um pacote chegar por um destes túneis e houver uma entrada para ela na tabela de rotas móveis.

Isso evita que os pacotes que entrem na rede, através de um MAR incorreto, sejam replicados nesses túneis de forma desnecessária. O problema dessa solução ocorre quando se usa agregação de endereços. Não há como saber se o MN é alcançável, em um determinado P2MP, simplesmente observando as entradas na tabela de rotas móveis. Nesse caso, é necessário que haja uma interação entre o RSVP-TE e o gerador de mensagens ICMP para manter a simplicidade da solução.

3.5 Considerações do Capítulo

O capítulo apresentou uma nova arquitetura de micromobilidade, chamada de MPA, a qual possui como objetivos imediatos, suprimir as deficiências exibidas pelas principais soluções de micromobilidade, conforme descritas na literatura.

A discussão dessa arquitetura foi dividida em quatros partes: apresentação do modelo de referência; discriminação dos blocos funcionais e suas interações; dinâmica da arquitetura e variações da arquitetura básica.

O modelo de referência foi apresentado e nele foram mostrados os vários elementos que compõem a arquitetura e, em especial, foi introduzido o elemento primordial da arquitetura, chamado de MAR. Algumas particularidades também foram apresentadas, entre elas, a rede sobreposta que conecta os MARs entre si, através de túneis do tipo ponto-multiponto. Tal túnel forma uma árvore com um MAR de entrada e vários MARs de saída.

Foi mencionado também que os MARs de saída, também chamado de MARs de egresso, são os roteadores de borda pertinentes à rede MPA e que se conectam aos nós móveis através de uma rede de acesso. Essa rede de acesso possui, como característica fundamental, um prefixo de endereço único. Quando o MN se associa à rede MPA, recebe um endereço IP o qual permanece constante até que ele a deixe.

Em seguida, foram discriminados os blocos funcionais que realizam as operações básicas da arquitetura. Devido a sua natureza distribuída, foram mostradas também quais são as interações possíveis de serem realizadas e a sua localização entre os vários elementos de rede que compõem a arquitetura.

Além disso, foram mostradas as várias possibilidades de se implementar as interações dos blocos entre si e como eles foram implementados em nosso protótipo. Em particular, foi mostrado como o protocolo RSVP-TE foi estendido com a adição de extensões previstas na especificação do protocolo, para atender às especificações da arquitetura.

A dinâmica da arquitetura foi ilustrada em dois cenários típicos, contemplando os casos: a ocorrência de um primeiro registro do MN na rede MPA e a realização de um *handoff*. Foi mostrado também, de modo simplicado, como a implementação utilizando túneis IP atende aos requisitos da arquitetura.

Por fim, algumas variações da arquitetura foram discutidas, as quais fogem do cenário básico ilustrado anteriormente. Foram discutidas algumas ideias de como é possível fazer um *handover* mais eficiente, chamado de pró-ativo. Foi discutido, ainda, como realizar a agregação de endereços, para aumentar a escalabilidade da arquitetura. E, por último, foi discutida a possibilidade de se ter vários MARs de ingresso, para redes com vários pontos de interligação com a Internet.

Portanto, podemos resumir as principais características / vantagens da arquitetura MPA como sendo:

1. Foi concebida para ser utilizada tanto na versão atual do protocolo IP, ou seja, o IPv4, quanto na versão futura, ou seja, o IPv6. É possível, inclusive, utilizar uma mistura entre os protocolos, como por exemplo, o IPv4 na rede de acesso e o IPv6 na rede de transporte;

2. Uso somente de protocolos padronizados e de baixa complexidade. Não é exigido, por exemplo, o uso de protocolos complexos, como o MIPv6.

3. Suporte a todos os dispositivos móveis atuais, baseados, por exemplo, nos sistemas operacionais *Android*, *Windows Mobile*, *Symbian* e *PalmOne*. Ela permite também que melhorias realizadas nos MNs sejam utilizadas para melhorar o desempenho do *handover*;

4. Mantem o endereço de rede do MN constante enquanto ele migra dentro de sua rede. Isso permite manter as conexões de transporte operacionais, quando ele realiza um *handover*;

5. Não restringe a sua utilização com um determinado tipo de protocolo de macromobilidade, como por exemplo, o MIP ou o HIP. Pode ser usada, inclusive, sem a presença de um protocolo de macromobilidade;

6. Integra-se aos mecanismos de segurança relacionados à: camada 2, como por exemplo, o certificado WPA (*Wi-Fi Protected Access*); camada 3, como exemplo, o protocolo IPsec e camada 4 ou superiores, como por exemplo, o protocolo TLS (*Transport Layer Security*), o protocolo SSL (*Secure Socket Layer*) ou o protocolo HTTPS (*Hypertext Transfer Protocol Secure*);

7. Permite a integração de várias operações relacionadas à gerência de rede em um único protocolo. Assim, a gerência de túneis P2MP, a gerência de QoS/CoS (*Class of Service*) e o rastreamento de MNs podem ser realizados utilizando o RSVP-TE. Isso reduz os custos de implementação e de operação;

8. Não interfere nos serviços que estão em uso na camada de transporte, como por exemplo, o VPN ou o VoIP.

No capítulo seguinte, apresentaremos a integração da arquitetura MPA com redes MPLS.

Capítulo 4

Integração da Arquitetura MPA com o Protocolo MPLS

Existem várias tecnologias de tunelamento de mensagens disponíveis para o uso atualmente, mas nem todas passíveis de serem integradas com a arquitetura MPA. Para que uma determinada tecnologia de tunelamento possa ser utilizada em uma rede MPA, é necessário que satisfaça a um conjunto básico de requisitos. Esses requisitos foram expostos no Capítulo 3, página 46, e estão sendo repetidos aqui por conveniência:

1. Criar redes sobrepostas: é a habilidade de conectar dois pontos distintos, escondendo a complexidade da rede entre eles. Isso pode ser alcançado, por exemplo, através do encapsulamento de dados, ou através do rotulamento de pacotes;
2. Criar conexões ponto-multiponto: é a habilidade de criar múltiplas ramificações de segmentos de saída, em um dado MAR, para um determinado segmento de entrada de um túnel qualquer. A replicação de pacotes pode ocorrer nessas ramificações;
3. Realizar realocação dinâmica de túneis: é a habilidade de mudar a topologia da rede ao, incrementalmente, adicionar ou remover segmentos de túneis.

Conforme já exposto, o protocolo MPLS é uma dessas tecnologias que satisfazem a esses requisitos. O protocolo MPLS satisfaz o item 1 desde a sua concepção, pois é a operação nativa do protocolo. O item 2 é obtido através de novas extensões ao protocolo [59, 60], o qual incorpora novas extensões à sinalização, para prover túneis do tipo ponto-multiponto ao protocolo MPLS. Por fim, o item 3, para o protocolo MPLS, passa a ser uma extensão do item anterior, pois o mesmo protocolo que permite a criação, manutenção e destruição de túneis P2MP, também permite a inserção e remoção de segmentos de túneis à árvore principal.

Além dessas características essenciais ao protocolo de tunelamento, existem, também, algumas outras que são pertinentes ao modo como a arquitetura MPA foi projetada. Em primeiro

lugar, a arquitetura MPA assume uma convivência pacífica com dispositivos legados, para que haja uma implantação da solução de forma gradual. Assim, a solução de tunelamento em MPLS deve ser capaz de coexistir de forma transparente com os LSRs já instalados na nuvem.

Isso implica que, além dos LSRs legados serem capazes de encaminhar os pacotes do MN dentro da nuvem MPLS, eles não devem ser pertubados por quaisquer sinalização, que não sejam as nativas ao protocolo. Outra observação importante é que a idéia de multiprotocolo deve ser preservada. Assim, a solução em MPLS deve ser capaz de operar tanto com a atual versão do protocolo IP, quanto com a futura versão, o protocolo IPv6, sem fazer distinção entre as duas.

Em segundo lugar, a arquitetura MPA assume que ela é hierarquizada, ou seja, as informações sobre a localização de um MN estão distribuídas entre os MARs. Portanto, os MARs que compõem o núcleo da rede devem ser capazes de inspecionar o pacote e decidir para qual túnel de saída encaminhá-lo. Essa característica é bastante evidenciada pela criação da rede sobreposta, a qual permite, além da gerência da arquitetura MPA, a escolha dos possíveis túneis de saída que compõem a rota de localização do MN.

Para o tunelamento de pacotes, utilizando-se túneis do tipo IP/IP, inspecionar e extrair informações de classificação / encaminhamento de pacotes são operações básicas do protocolo, sendo, portanto, facilmente integradas nos roteadores. Por outro lado, o MPLS é uma tecnologia na qual a opacidade dos pacotes é preservada na rede. Para essa tecnologia, a solução deve contemplar tanto a funcionalidade prevista na arquitetura MPA, quanto evitar que o protocolo MPLS seja alterado.

Por fim, a arquitetura MPA foi concebida para permitir a sua evolução com o tempo, ao aceitar a agregação de novas funcionalidades e facilidades, como por exemplo, engenharia de tráfego, QoS, dentre outros serviços. Assim, a solução em MPLS deve permitir que esses novos serviços sejam incorporados de forma transparente sobre o protocolo. Umas das formas de garantir isso é restringir a solução às funcionalidades que já são previstas pelo orgão de padronização do protocolo, neste caso, o IETF.

Portanto, serviços já previstos e especificados para o MPLS devem ser facilmente integrados com a solução proposta, dentre eles, podemos citar as especificações dos serviços diferenciados para o MPLS [67], os procedimentos de engenharia de tráfego [68] e a recuperação rápida em situações de falhas [69].

O restante deste capítulo será organizado da seguinte forma: a Seção 4.1 apresenta as soluções propostas pela literatura, para realizar o rastreamento do MN e as dificuldades para empregá-las na arquitetura MPA. A Seção 4.2 apresenta a nossa proposta de rastreamento do MN, mostrando a dinâmica da solução e as questões de gerência de rótulos inerentes a ela. A Seção 4.3 apresenta o mapeamento de nossa proposta na arquitetura MPA. Por fim, a Seção 4.4 finaliza o capítulo, discorrendo sobre as considerações pertinentes a ele.

4.1 Soluções Propostas pela Literatura

Conforme discutido no Capítulo 2, existem, basicamente, três formas diferentes propostas pela literatura, de se utilizar o protocolo MPLS para rastrear o MN em uma nuvem MPLS. Devido a uma falta de nomenclatura mais adequada, vamos chamá-las de: LSPs entre LSRs de ingresso e egresso; vários LSPs que conectam os LSRs entre si e extensão da tabela de rótulos (LIB).

Conforme será discutido, nenhuma das três propostas é adequada para resolver o problema de rastrear o MN dentro de uma nuvem MPLS. Elas apresentam características negativas relacionadas, tanto em sua proposta original para o qual foram concebidas, quanto ao tentar adaptá-las à arquitetura MPA.

4.1.1 LSPs entre LSRs de Ingresso e Egresso

Este tipo de rastreamento do MN foi proposto originalmente por Ren *et al*, no artigo *Integration of Mobile IP and Multi-Protocol Label Switching* [23], considerando como o protocolo MIP poderia ser mapeado utilizando-se de túneis MPLS e, posteriormente, por outros autores [24, 26, 27], considerando a possibilidade de aplicar os conceitos de micromobilidade, dentro de um domínio administrativo MPLS.

A Figura 4.1 ilustra o princípio genérico de rastreamento do MN nessas propostas. Conforme pode ser observado, somente os LERs de ingresso e egresso são requeridos para participar ativamente da sua dinâmica. Observe-se, ainda, que a proposta assume que o MN está executando o MIPv4, como protocolo de macromobilidade. Assim, o MN possui, normalmente, dois endereços IPs: o primeiro que se refere ao endereço estático, recebido em sua rede de origem; e o segundo que se refere ao endereço CoA, adquirido na rede de acesso, o qual está associado.

A dinâmica é a seguinte: sempre que o MN se associa a um ponto de acesso, ele recebe um endereço CoA de seu LER de egresso. Com isso, o MN dá início aos procedimentos de registro deste endereço perante o seu HA. Esse LER, ao receber essa solicitação de registro, inicia o estabelecimento de um LSP com o LER de ingresso, cuja FEC é o endereço CoA do MN. O LER de egresso cria também uma entrada, em sua tabela de rotas, para o endereço IP da rede de origem do MN. Isso permite que os pacotes recebidos por este LER possam ser encaminhados ao MN através da camada de enlace.

Na proposta de Ren *et al* [23], a qual é a adaptação do MIPv4 para o MPLS e, portanto, um protocolo de macromobilidade, o LER de ingresso é o agente HA do MN. Para poder rastreá-lo, esse LER cria, em sua tabela de FECs, uma nova entrada, cujo parâmetro de entrada é o endereço IP do MN de sua rede de origem e os parâmetros de saída são os mesmos do LSP recém criado para o seu endereço CoA. Assim, quando um pacote é destinado ao MN e ele está

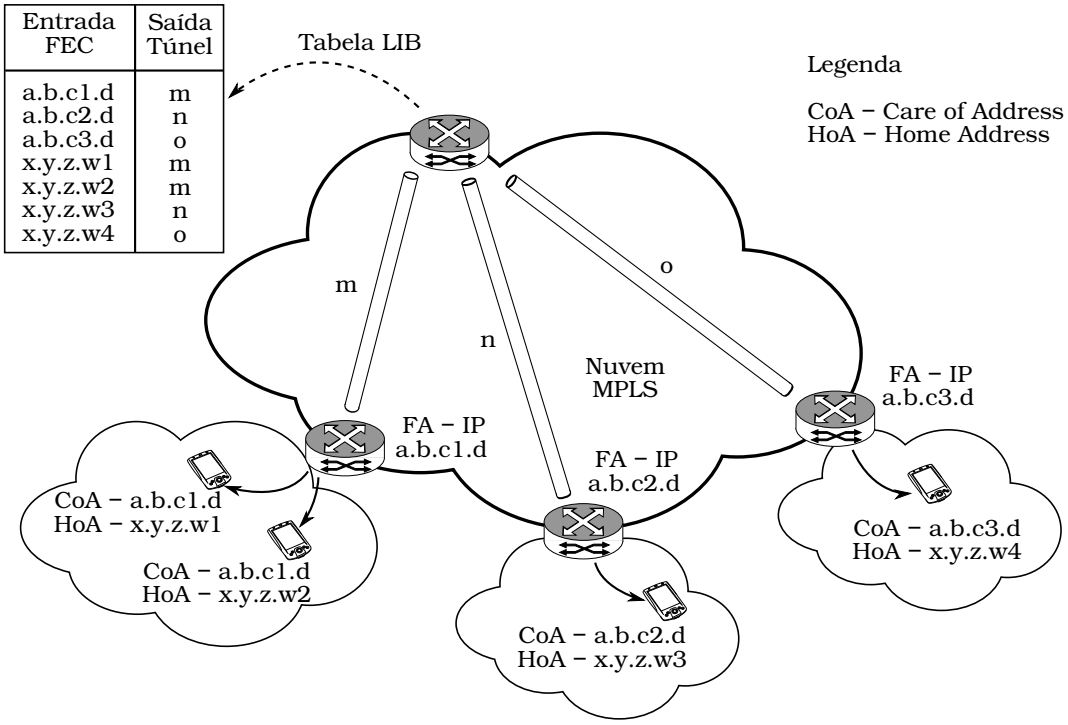


Figura 4.1: Túneis MPLS conectando os LSRs de ingresso aos LSRs de egresso, para rastrear os MNs.

ausente de sua rede de origem, o agente HA o personifica e, portanto, recebe o pacote em seu nome. Ele verifica em sua tabela de FEC por uma entrada correspondente ao MN e quando a encontra, tunela o pacote de acordo com os parâmetros que definem o túnel de saída, ou seja, o túnel que interliga o LER de ingresso ao LER de egresso.

Em uma migração, o MN repete todo o procedimento de registro para o seu novo CoA, adquirido na rede que está se associando. Um novo LSP é criado entre o HA (LER de ingresso) e o FA (LER de egresso) se não existir e no HA é necessário, apenas, atualizar os parâmetros de saída correspondentes à entrada do MN, na tabela de FEC, com os novos parâmetros de saída do novo túnel recém criado.

Uma variação deste esquema foi proposto por Yang e Makrakis [24] e posteriormente, também, por Langar *et al* [26, 27], o qual insere um novo componente, para cuidar da micromobilidade. Assim, a rede passa a ser dividida em duas partes, a primeira de macromobilidade MIPv4 até o LER de ingresso de um domínio administrativo e a segunda de micromobilidade entre este LER de ingresso, chamado por Yang e Makrakis de FDA, até o LER de egresso, ou seja, o FA.

A dinâmica é praticamente a mesma, com exceção de que enquanto o MN está dentro de um domínio é o agente FDA que registra o seu endereço IP junto ao agente HA, como

sendo o endereço CoA do MN. Assim, somente o LSP que conecta o FDA e o FA necessita ser atualizado quando o MN migra de uma subrede para uma outra, ou seja, os parâmetros de saída, da entrada referente ao MN na tabela FEC do FDA, necessitam ser atualizadas com os parâmetros de saída do LSP pertencente ao FDA - novo FA.

Portanto, quando um pacote destinado ao MN é recebido em sua rede de origem e ele está em migração, o agente HA o personifica e recebe este pacote em seu nome. Ele verifica em sua tabela FEC por uma entrada do MN e, caso exista, envia o pacote através do túnel de saída especificado pelos parâmetros de saída. O agente FDA, ao receber este pacote, procura por uma entrada do MN em sua tabela FEC. Se existir, tunela o pacote de acordo com os parâmetros pertinentes aos descritores de saída. O agente FA, ao receber os pacotes, retira as informações concernentes ao protocolo MPLS, verifica em sua tabela de rotas se o pacote é alcançável e se for, entrega-o ao MN.

Um dos atrativos dessa solução são as poucas alterações necessárias ao protocolo MPLS. Em particular, só é exigido que seja possível popular a tabela FEC do LER de ingresso com entradas pertinentes ao MN e atualizar os seus parâmetros de saída com os parâmetros de saída de algum outro LSP já estabelecido, toda vez que o MN migra. Portanto, é exigido que seja possível manipular a tabela FEC, deste LER, através de alguma interface de comunicação, não prevista originalmente nas especificações do protocolo. Isso implica, usualmente, em atualização do *firmware* dos LERs de ingresso.

Por outro lado, a função de rastrear o nó móvel é toda feita pelo LER de ingresso, não sendo distribuída pelos demais elementos de rede. Isso é um problema potencial de escalabilidade, pois qualquer alteração de localidade do MN deve ser reportada a esse LER, para que ele possa atualizar os dados pertinentes a sua localização. Assim, quanto maior for o número de MNs nesta rede, maiores serão as requisições de controle que ele deverá processar.

Uma outra deficiência que pode ser observada é a falta de hierarquização da nuvem MPLS, pois somente os elementos de borda da rede participam efetivamente no rastreamento do MN. Isso implica que qualquer *handoff* efetuado pelo MN sempre será tratado como o pior caso, no qual a sinalização deve ser sempre enviada até o LER de ingresso, para que possa ser atualizada a entrada na tabela FEC pertinente a sua localização. Se houver LSRs em comum entre o LSPs, ou seja, entre o antigo e o novo, eles não poderão ser compartilhados entre si.

Aplicar esse esquema de rastreamento do MN para a arquitetura MPA é um desafio ainda maior, pois a arquitetura MPA não restringe a sua aplicação a um protocolo de macromobilidade específico, podendo, ainda, funcionar na ausência dele. Assim, a arquitetura MPA não se apoia nesse tipo de sinalização para promover a localização do MN, o que significa atualizar toda a dinâmica de sinalização.

A falta de hierarquização é ainda mais problemática, pois a arquitetura MPA é fundamentada em sua capacidade de distribuir a função de rastrear o MN nos vários elementos de rede

e, ao se utilizar este esquema, estaria-se reduzindo a sua versatilidade. Isso pode ser observado tanto nos *handovers* reativos, pois não se poderia utilizar LSRs comuns entre dois LSPs para diminuir o tempo de latência; quanto nos *handovers* pró-ativos, pois não seria eficiente realizar a replicação de pacotes, uma vez que os pacotes replicados nunca teriam a chance de ficarem confinados em uma pequena região da rede. Eles sempre seriam replicados desde o LER de ingresso até os LERs de egresso.

4.1.2 Vários LSPs que Conectam os LSRs entre si

Proposto por Chiussi *et al*, no artigo intitulado *A network architecture for MPLS-based micro-mobility* [25], a ideia consiste em criar uma rede sobreposta à rede MPLS, na qual os nós são LSRs operando com as mesmas atribuições de um LER, ou seja, com a capacidade de classificação de pacotes e são chamados pelos autores de LEMA. Esses nós são interligados entre si através de LSPs permanentes, criados por algum procedimento de engenharia de tráfego.

A Figura 4.2 ilustra um cenário típico para esta rede. Somente a rede sobreposta é mostrada e os LSPs que interconectam esses LEMAs. Esta figura mostra também a tabela FEC para cada LEMA, considerando que existem quatro MNs sendo rastreados por ela. Por fim, ela assume, também, que os MNs executam o MIPv4 como protocolo de macromobilidade.

A dinâmica deste esquema é a seguinte: quando um MN ingressa na rede pela primeira vez, ele inicia o processo de registro do CoA, cujo valor é o endereço IP do LEMA de ingresso, perante o seu HA. Com isso, o LEMA de egresso cria uma entrada, em sua tabela de rotas, para o endereço IP do MN de sua rede de origem. Sequencialmente a este processo, o LEMA de ingresso, se não existir uma entrada para o MN em sua tabela FEC, cria uma entrada com o endereço IP de sua rede de origem e com os descritores de saída do LSP *downstream*. O LEMA de ingresso encaminha também, este registro ao HA do MN com o seu endereço IP como sendo o CoA do MN.

Esse processo de criação de uma entrada na tabela FEC é repetido para todos os LEMAs que estão no caminho *downstream* até o MN. Uma entrada na tabela FEC é composta pelo endereço IP da rede de origem do MN e os parâmetros de saída são os descritores do LSP *downstream*.

Assim, quando um pacote é destinado ao MN e chega em sua rede de origem, o agente HA o intercepta, consulta o seu banco de dados de localização do MN e verifica que ele é alcançável. Consequentemente, ele tunela esse pacote ao LEMA de ingresso. Quando o LEMA recebe o pacote, verifica se o endereço de destino tem uma equivalência em sua tabela FEC e, ao encontrar uma entrada pertinente ao MN, tunela-o com as diretrizes pertinentes ao LSP de saída.

Na saída desse LSP existe um outro LEMA, o qual é egresso para este LSP e, portanto,

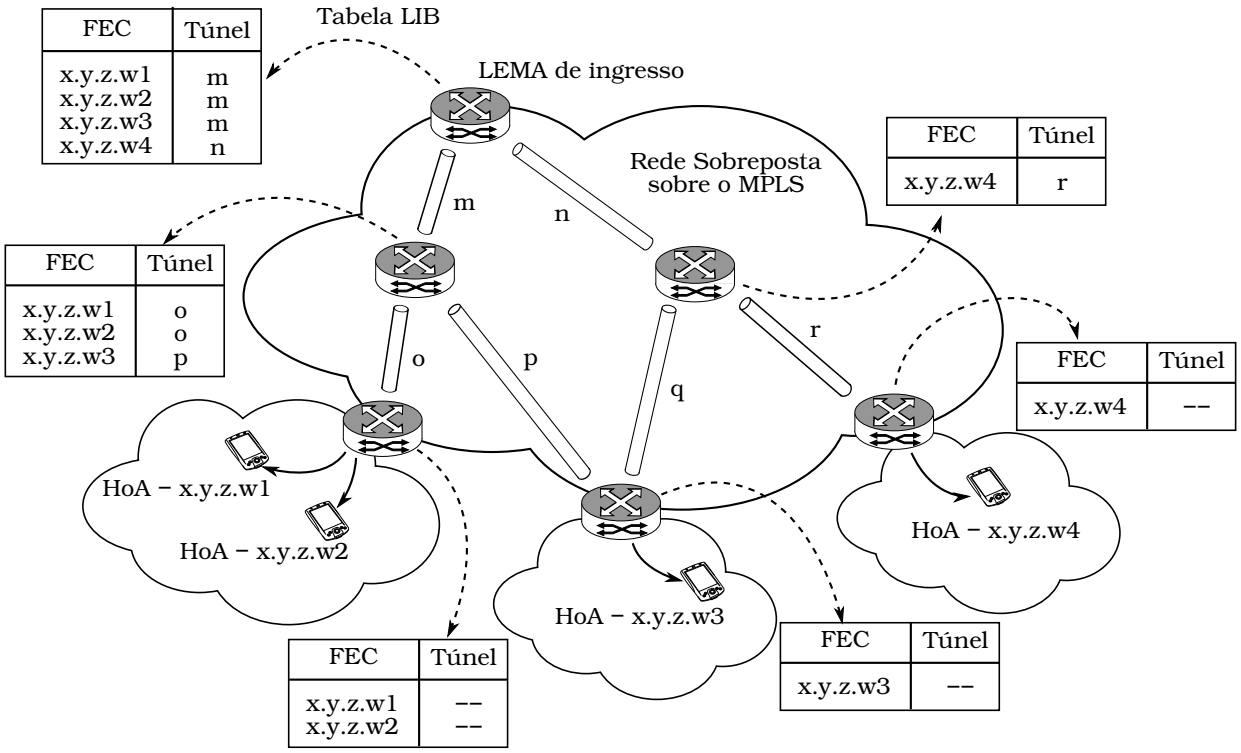


Figura 4.2: Rastreamento do MN utilizando-se de vários LSPs, interconectados entre si através da camada de rede.

obtém o pacote. Ele refaz essa classificação, procurando por uma entrada, na tabela FEC, pertencente ao MN. Ao encontrá-la, retunela este pacote, com as informações referentes ao novo LSP de saída. Este processo se repete até que um destes LEMAs seja o egresso para este MN. Nesse caso, esse LEMA entrega o pacote para a camada de rede, para que esta faça a entrega final do pacote ao MN. Usualmente, isso implica em verificar se existe uma entrada para o MN em sua tabela de rotas e que ele é alcançável dentro do escopo da camada de enlace.

Em uma migração do MN, desde que o MN permaneça dentro do domínio MPLS, ela será tratada como sendo de micromobilidade. Assim, quando o MN se associar a um novo LEMA, todo o processo descrito anteriormente se repetirá, exceto que o registro do CoA do MN junto ao seu HA permanece inalterada, prevenindo o HA de ter conhecimento desta migração.

Esta proposta de rastreamento do MN tem algumas propriedades interessantes. A primeira delas é o uso de túneis permanentes que interligam os LEMAs. Isso elimina o tempo gasto no estabelecimento de um LSP, uma vez que só é necessário realizar a ligação, ou religação, entre a FEC do MN e os atributos de saída pertinentes ao LSP *downstream*.

A segunda é a hierarquização da rede, pois a função de rastrear o MN é distribuída entre os LEMAs. Isso melhora a eficiência do *handover*, pois quanto mais LEMAs forem comuns

entre a antiga localização do MN e sua atual, mais LEMAs poderão ser aproveitados na nova interligação, diminuindo a sua latência. Melhora, também, a escalabilidade pois as informações pertinentes ao MN estão distribuídas na rede e estão contidas somente no caminho que liga a entrada na rede até o seu egresso.

A terceira é a rede sobreposta, pois permite a implantação gradual do esquema, permitindo a sua coexistência com os LSRs legados.

Por outro lado, este esquema exige que seja possível criar a uma nova entrada na tabela FEC nos LEMAs que associe os seus descritores de saída com algum descritor de saída de um LSP já estabelecido. Isso demanda uma nova interface de gerência e a atualização dos *firmwares* destes LEMAs.

Entretanto, o mais dispendioso é a necessidade de se reclassificar os pacotes no nível de camada rede, no núcleo da rede MPLS. Além de violar a premissa de que a classificação do pacote somente ocorre na entrada da nuvem, viola a premissa de que o MPLS possa transportar qualquer tipo de protocolo de camada 3. Quer dizer que para cada versão do protocolo IP, por exemplo, deva existir uma implementação equivalente que saiba tratar de suas especificidades. O segundo problema associado é o tempo gasto no processo de destunelar o pacote, classificá-lo e retunelar no LSP de saída.

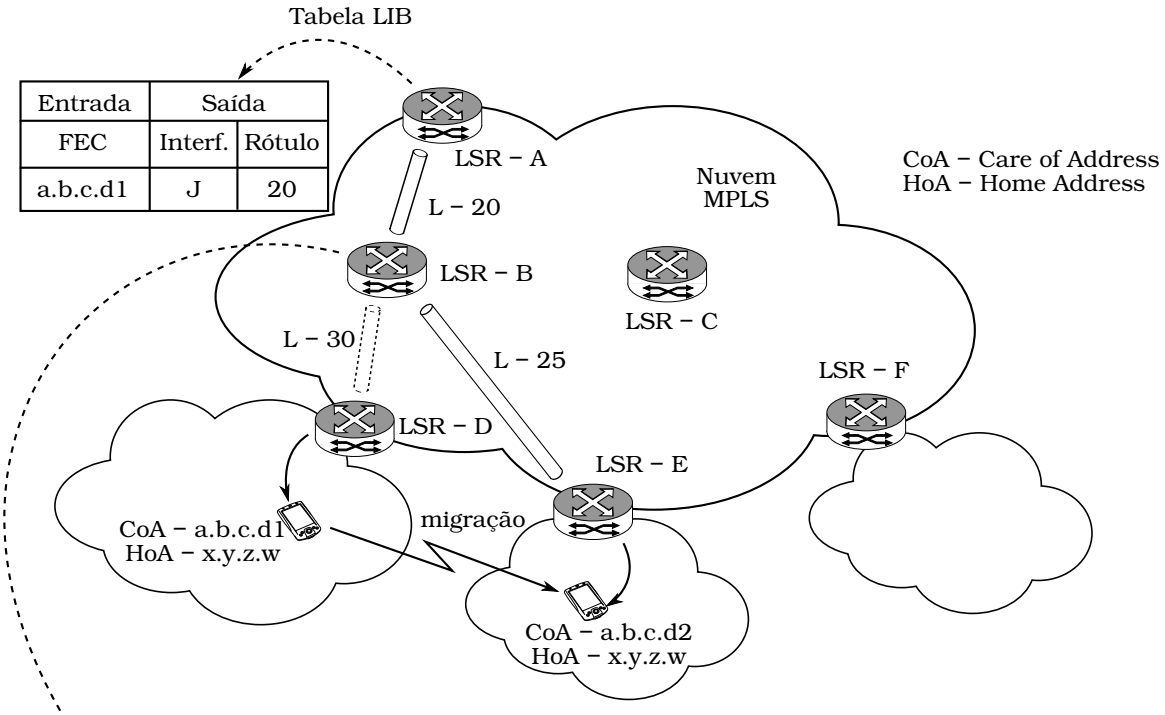
Embora esse esquema seja passível de se utilizar diretamente como uma solução de tunelamento de MN em MPLS, para a arquitetura MPA, a violação da filosofia básica de encaminhamento previsto no protocolo MPLS é quebrada. Portanto, não é uma solução adequada para se utilizar nesta arquitetura. Outras questões que previnem o seu uso são: necessidade de se ter implementações separadas para cada versão do protocolo IP, pois a arquitetura MPA pode operar indistintamente sobre qualquer uma delas e a possibilidade de replicar pacotes, pois não fica claro se este esquema permitiria isso. O fato dela ter sido especificada visando que o MN implemente o protocolo MIPv4, não impede que seja adaptada para a arquitetura MPA.

4.1.3 Extensão da Tabela de Rótulos

A extensão da tabela de rótulos foi proposta por Fowler e Zeadally, no artigo intitulado *Fast Handover over Micro-MPLS-Based Wireless Networks* [28]. A ideia consiste em modificar a tabela de rótulos, chamada de LIB, adicionando-lhe novas informações, a fim de poder rastrear a atual localização do MN. Esta nova tabela de rótulos estendida é chamada pelos autores de I-LIB.

A Figura 4.3 ilustra um cenário típico em que o MN, o qual está associado ao LSR-D, migra para o LSR-E. É assumido que o MN implementa o MIPv4 como protocolo de macromobilidade. Somente um LSP está sendo representado, composto, inicialmente, pelos segmentos L-20 e L-30. Note-se a presença dos LSRs C e F, os quais não participam da formação do LSP que tunela

os pacotes para o MN. Por fim, estão sendo mostradas a tabela FEC do LER de ingresso, ou seja, o LSR-A, e a tabela de rótulos do LSR-B, a qual é a tabela estendida pela proposta. Não foram mostradas as tabelas de rótulos dos LSRs D e E a fim de não sobrecarregar a figura.



Entrada		Saída						
Interf.	Rótulo	Interf.	Rótulo	Flag	novo CoA	antigo CoA	end. MN	temp.
J	20	K ~~L~~	25 ~~30~~	1 ~~0~~	a.b.c.d2 ~~a.b.c.d1~~	a.b.c.d1	x.y.z.w	150 s

Figura 4.3: Rastreamento do MN através da alteração da tabela LIB, com a agregação de novos descritores.

A dinâmica desta proposta é a seguinte: quando o MN ingressa pela primeira vez na rede MPLS, ele recebe um endereço CoA de seu roteador de acesso e inicia o processo de registro, conforme especificado no MIPv4, junto ao seu HA. O LER de ingresso, ao receber essa mensagem de requisição, registra o seu endereço IP com o HA do MN, para prover a micromobilidade, e cria um LSP, cuja FEC é o endereço CoA do MN, até o LER de egresso, no qual ele está associado. Nesse cenário, corresponde ao LSP que está compreendido nos LSRs A, B e D.

É durante o *handover* que a ideia do uso da tabela I-LIB é compreendida. Quando o MN se associa a um novo roteador de acesso, ele recebe um novo endereço CoA e inicia o processo de registro. Essa requisição caminha no sentido *upstream* até alcançar um LSR que seja comum tanto ao novo segmento do LSP, quanto ao antigo. Vamos chamá-lo, por conveniência, de LSR estendido.

Ao chegar nesse LSR, ao invés da mensagem continuar no seu percurso *upstream* até alcançar o LER de ingresso e recriar um novo LSP, ela para neste LSR e é criado um novo segmento, que compreende os LSRs que vão desde esse LSR comum até o LSR de egresso da nova localização do MN. Assim, o LSP antigo é atualizado com inserção deste novo segmento e, por consequência, a remoção do antigo. Essa situação é ilustrada na Figura 4.3 com a criação do segmento nomeado L-25 e a remoção do segmento L-30, o qual está em tracejado para reforçar essa ideia.

O LER de ingresso não é informado desta atualização de localidade do MN e, portanto, ainda o conhece através de seu antigo CoA. No entanto, o MN recebe um novo endereço CoA em sua rede de acesso. Logo, para que o pacote possa ser aceito, corretamente, na rede de destino, o LSR estendido deve corrigir o valor do endereço CoA no cabeçalho IP do pacote do MN.

Os descritores de saída extras, adicionados à LIB, permitem realizar essa operação. O campo `novo CoA` informa ao LSR estendido qual deve ser o valor do endereço de destino do pacote, para os que são enviados no caminho *downstream*, enquanto que o campo `antigo CoA` informa qual deve ser o valor do endereço de origem do cabeçalho IP para os pacotes *upstream*. Nem sempre essa troca é necessária, como por exemplo, no caminho composto pelos segmentos L-20 e L-30. Para que o LSR estendido saiba quando realizar essa comutação, o campo `Flag` é inicializado com zero ou um. Se for zero, não há a necessidade de comutação; se for um, realiza-se a comutação.

Um endereço CoA pode ser compartilhado entre vários MNs que estão sob o mesmo roteador de acesso. Isso permite multiplexar vários pacotes, de vários MN, no LSR de ingresso, diminuindo o tamanho da tabela FEC. Para que o LSR estendido possa demultiplexar os pacotes deste LSP, o campo `endereço MN` permite discriminar para qual MN o pacote pertence e, com isso, realizar a comutação de forma individualizada.

Por fim, para evitar que a tabela I-LIB se torne muito grande, quando o MN estiver em um estado inativo, o campo `temporização` diz quanto tempo essa entrada é válida, antes que seja removida automaticamente. Cada pacote que confirma uma entrada nesta tabela, reinicializa este contador, para esta entrada, o que dá uma característica de *soft-state* a essa tabela.

Embora a ideia de aproveitar segmentos comuns a dois caminhos distintos seja bastante interessante, pois diminui o tempo de sinalização e, por consequência, a latência do *handover*, esse esquema de rastreamento do MN possui alguns pontos negativos.

O primeiro deles é o fato de que o LSR estendido deve ser capaz de reconhecer um pacote IP e qual a sua versão. Isso viola a ideia de opacidade do MPLS, conforme já discutido anteriormente. Além disso, para aplicar a ideia na versão futura do IP, deve-se ter uma implementação que saiba reconhecer e processar qual a versão do pacote.

A segunda é a provável necessidade de realizar a comutação entre os endereços IP de des-

tino no pacote IP, em cada LSR estendido. Além de aumentar a latência de encaminhamento do pacote, pois deve-se alterar os valores do endereço IP e recalcular o campo de `checksum` do cabeçalho IP, existe um problema em potencial do pacote ser recusado na rede de destino, pois, se estiver sendo usado um protocolo de segurança, como por exemplo, o IPsec, o pacote não será autenticado.

A terceira é a necessidade dos LSR reconhecerem que está sendo executado um protocolo de macromobilidade, em particular, ou o MIPv4, ou o MIPv6, ou ambos, pois ele necessitam não somente realizar a comutação do endereço CoA do pacote, mas também reconhecer que o cabeçalho interno é pertinente ao MN, conforme especificado no MIP. Se as mensagens MIP estiverem protegidas (criptografadas) esse protocolo não irá funcionar.

Por fim, note-se que quanto mais LSR estendidos estiverem fazendo a comutação de pacotes, maior será a latência de encaminhamento de pacotes.

Aplicar esse esquema para a arquitetura MPA é no mínimo complicado. Como a arquitetura MPA assume que toda a rede de acesso possui um único prefixo de rede, a ideia de comutação de endereço, no LSR estendido, deixa de ter sentido e passa a ser equivalente a um subcaso dos túneis P2MP, em que somente a inserção, com a automática remoção de segmentos, é aplicada.

A arquitetura MPA utiliza-se também de uma rede sobreposta, com túneis P2MP permanentes entre os MARs. Adaptando esse esquema, para utilizar-se de túneis P2MP, verifica-se que ainda assim não é viável, pois, para demultiplexar o fluxo de pacotes dentro do túnel P2MP, é necessário estender a LIB, com um campo, referente ao endereço do MN da rede de origem. Isso implica que as deficiências apresentadas anteriormente estarão presentes nesta solução.

Por fim, a arquitetura MPA não obriga a utilização de um protocolo de macromobilidade específico, sendo aplicável até mesmo na ausência deles. Por outro lado, esse esquema exige que os LSRs estendidos saibam qual protocolo de macromobilidade está sendo utilizado, para que possam identificar quais pacotes pertencem a um determinado nó móvel, ou seja, a demultiplexação do fluxo de pacotes.

4.2 Proposta de Rastreamento do MN

Conforme foi mostrado na seção anterior, nenhuma das soluções apresentadas pela literatura satisfaz de forma eficiente o problema do rastreamento do MN, devido, principalmente, à falta de generalização de seus esquemas. Nessa seção, vamos propor uma nova forma de rastrear o MN, visando, além de preservar as especificações originais do protocolo MPLS, obter o máximo de generalidade, de forma a ter uma solução que possa ser aplicada, inclusive, nas propostas citadas na literatura.

Para que seja considerada generalista, a proposta deve ser capaz de rastrear a localização do MN, sem perder a essência da arquitetura do MPLS. Por essência, podemos citar alguns trechos da especificação de sua arquitetura [6]:

> No MPLS, a atribuição de um determinado pacote a uma determinada FEC é realizada apenas uma única vez, quando o pacote entra na rede. A FEC para o qual o pacote é atribuído é codificada como um valor de comprimento fixo e curto, conhecido por "rótulo".
> (...) Nos *hops* subsequentes, não há mais análise do cabeçalho da camada de rede do pacote.
> (...) No paradigma de encaminhamento do MPLS, uma vez que o pacote é atribuído a uma FEC, não existe mais análise de cabeçalho pelos roteadores subsequentes; todo encaminhamento é realizado através dos rótulos.
> (...) MPLS significa 'Múltiplos protocolos' encaminhados por rótulos, múltiplos protocolos porque estas técnicas são aplicáveis em *qualquer* protocolo de camada de rede. (Tradução do autor)

Como consequência deste conceito, um pacote deve ser classificado somente na entrada da rede, quando é permitido examinar os cabeçalhos de camada 3 ou superiores. Após o seu ingresso, somente as informações agregadas ao protocolo MPLS, ou seja, os rótulos, devem ser utilizadas para encaminhar os pacotes até o seu destino final. Os motivos estão elucidados na RFC 3031 e repetidos aqui por conveniência:

- O encaminhamento no protocolo MPLS pode ser feito por *switches*, os quais são capazes de realizar a busca e troca de rótulos, mas não são capazes de realizar, ou a análise dos cabeçalhos na camada de rede, ou a sua análise em velocidades adequadas;
- Uma vez que o pacote é atribuído a uma FEC, quando ele entra na rede, o roteador de ingresso pode usar, para determinar a atribuição, qualquer informação que ele possui sobre o pacote, mesmo que essa informação não possa ser obtida do cabeçalho de rede. Por exemplo, os pacotes entrantes em diferentes portas podem ser atribuídos a diferentes FECs. O encaminhamento tradicional somente pode considerar as informações que trafegam com o pacote, em seu cabeçalho;
- Um pacote que entra na rede em um determinado roteador pode ser diferentemente rotulado, do que se ele entrasse na rede por um roteador diferente. Como consequência, encaminhamentos que resultam de decisões que dependem do roteador de ingresso podem ser facilmente realizadas;
- As considerações que determinam como um pacote é atribuído a uma FEC podem se tornar cada vez mais complicadas, mas sem impacto algum sobre todos os roteadores que meramente encaminham pacotes rotulados.

Ainda, para a continuidade da discussão, é necessário definir o conceito de túnel LSP. A RFC 3031 o define no seguinte parágrafo:

> É possível implementar um túnel como um LSP e utilizar a troca de rótulos, ao invés do encapsulamento em camada de rede, para permitir que um pacote trafegue através deste túnel. O túnel poderia ser um LSP <R1, ..., Rn>, onde R1 é a ponta de transmissão do túnel e o Rn é a ponta de recebimento do túnel. Isto é chamado de "Túnel LSP". (Tradução do autor)

E, em seguida, o que é uma hierarquia de túneis, ou seja:

> Considere um LSP <R1, R2, R3, R4>. Vamos supor que R1 recebe um pacote não rotulado P e adiciona, em sua pilha de rótulos, um rótulo para permitir que ele siga este caminho e este é de fato o caminho *Hop* a *Hop*. Entretanto, vamos supor mais ainda que R2 e R3 não estão diretamente conectados, mas são "vizinhos" pela virtude de serem as pontas de um túnel LSP. Assim, a sequência atual de LSRs atravessados por P é <R1, R2, R21, R22, R23, R3, R4>.
> Quando P trafega de R1 para R2, ele terá um empilhamento de rótulos de profundidade 1. R2, trocando o rótulo, determina que P deve entrar no túnel. R2 troca, primeiramente, o rótulo de chegada com o rótulo que é significativo para R3. Então, ele adiciona um novo rótulo. Este rótulo de nível 2 tem um valor que é significativo para R21. O chaveamento de rótulos é realizado no nível 2 pelos LSRs R21, R22 e R23. R23, o qual é o penúltimo *hop* no túnel R2-R3, libera um nível da pilha de rótulos antes de encaminhá-lo para o R3. Quando o R3 observa o pacote P, P possui apenas o nível 1 de rótulo, tendo agora, saído do túnel. Uma vez que R3 é o penúltimo *hop* no nível 1 do LSP de P, ele libera a pilha de rótulos e R4 recebe o pacote P sem rótulo.
> O mecanismo de empilhamento de rótulos permite o tunelamento de LSP aninhar em qualquer profundidade. (Tradução do autor)

Diante disto, a proposta é realizar o rastreamento do MN utilizando-se da facilidade do MPLS que é o empilhamento de rótulos, ou seja, a capacidade de inserir túneis LSPs dentro de túneis LSPs mais externos.

Conforme será mostrado, um empilhamento de dois níveis é suficiente para realizar a função de rastrear o MN, mantendo as mesmas características de rastreamento apresentadas pela literatura, anteriormente discutidas e, também, satisfazendo o modo como a arquitetura MPA reorganiza os túneis para rastrear o MN [53, 54, 50, 51]. Note-se também que estarão sendo utilizados os termos túneis e LSP intercambiavelmente, para denotar o mesmo conceito, que é o caminho percorrido por pacotes em um domínio MPLS e que estão encapsulados por este protocolo.

Assim, os dois níveis de hierarquia serão chamados, por conveniência, de túnel externo e túnel interno. Os túneis externos permitem determinar os caminhos pelos quais um MN pode percorrer em uma nuvem MPLS, partindo desde o LER de ingresso até os seus possíveis LERs de egresso. Existem algumas variações de como esses túneis podem ser criados e gerenciados. A primeira delas é com relação ao seu tipo. Eles podem ser túneis ponto a ponto (P2P), ou túneis ponto-multiponto (P2MP).

Os túneis P2P devem ser criados entre dois LSRs que possuem alguma relação de vizinhança entre si. Por exemplo, um LSP para cada entrada da tabela de rota IP com o seu

correpondente *gateway*, ou a relação de vizinhança que existe entre dois MARs. Através da correta interligação entre esses túneis LSPs, pode-se estabelecer uma rota entre o LER de ingresso de um MN até o seu LER de egresso, na rede de acesso.

Por outro lado, os túneis P2MP devem ser criados entre o LER de ingresso de um MN com todos os seus LERs de egressos, estabelecendo, assim, os caminhos possíveis que o MN pode percorrer dentro um domínio MPLS.

Como o túnel P2MP possui um formato de árvore, pode-se defini-lo como sendo formado por segmentos de LSPs. Um segmento é um trecho de túnel LSP que conecta dois LSRs de ramificação entre si, ou entre os LERs de ingresso e egresso.

O segundo modo como se pode classificar esses túneis externos é com relação à duração. Eles podem ser de longa ou de curta duração. Um túnel de curta duração usualmente é criado para representar um estado momentaneo da rede e, assim, é normalmente criado através de um protocolo de distribuição de rótulos, como por exemplo, o RSVP ou o LDP.

Um túnel de longa duração normalmente é criado para representar um estado estático da rede, definido de acordo com alguma política de gerência. Assim, existe uma baixa probabilidade de que esses túneis serão alterados no futuro próximo. Eles são criados tanto através de procedimentos manuais de engenharia de tráfego, quanto através de protocolos de sinalização.

Por outro lado, o túnel interno identifica o MN dentro de um domínio MPLS. Assim, sempre que um MN entra em um domínio de mobilidade MPLS, um LSP é criado para rastreá-lo. Embora a criação deste LSP possa ser feita estaticamente, através do uso de procedimentos de engenharia de tráfego, não é esperado que seja a forma usual mais empregada. Assim, a criação do túnel interno deve ser realizada através do uso de um protocolo de distribuição de rótulos, ou seja, de forma dinâmica. O tempo de duração deste túnel é, em geral, equivalente ao tempo de permanência do MN no domínio MPLS, sendo considerado de curta duração.

Esse LSP é, então, tunelado através de uma lista de túneis externos. Essa lista é composta de túneis externos que permitem aos pacotes, pertinentes ao MN e que entram no LER de ingresso, alcançarem a rede de acesso. A composição dessa lista é dependente da forma como os túneis externos foram construídos.

Se forem criados através de túneis LSPs P2P, a lista é composta de LSPs que devem ser atravessados para alcançar o destino. Logo, na ponta final de cada túnel externo, o túnel interno deve ser extraído, para que o seu rótulo possa ser analisado e, consequentemente, a entrada pertinente a sua tabela LIB possa ser consultada para decidir em qual túnel externo o túnel interno deve ser novamente encapsulado.

Se forem criados através de túneis P2MP, a lista é composta de segmentos que compõem o túnel P2MP e que permitem alcançar a rede de acesso. Assim, em cada LSR de ramificação e nos LERs de ingresso e egresso, o túnel interno é extraído e o seu rótulo analisado para que a entrada correspondente, na tabela LIB, possa ser consultada para decidir em qual segmento

de saída o túnel interno deva ser encapsulado.

Reitere-se que, não importa como o túnel externo foi criado, a abertura do túnel externo só deve ocorrer, ou na ponta final de um túnel, ou nos LSRs de ramificação e nos LERs de ingresso e egresso que compõem a árvore P2MP. Isso implica que, nos LSRs que não estão nesse grupo, o encaminhamento é feito somente sobre o rótulo mais externo, ou em outras palavras, tais LSRs não conhecem que existe uma hierarquia de túneis presente na nuvem MPLS.

Uma questão importante, que deve ser ressaltada, é com relação à interpretação do rótulo atribuído a um LSP em cada LSR. Para o túnel externo, o rótulo atribuído é uma interpretação local, cujo significado é pertinente apenas a este elemento. Assim, dado um LSP qualquer, a concatenação desses rótulos, nos vários LSRs, definem o LSP para este domínio.

Para o túnel interno, o rótulo atribuído tem um significado global e o seu valor deve ser constante, dentro deste mesmo domínio. Isso é necessário, pois em qualquer LSR que se faça a extração do túnel interno, ao observar o valor deste rótulo, deve-se obter sempre a mesma associação com o MN ao qual ele pertence. Assim, para que isso seja possível, deve existir uma relação de 1:1 entre o rótulo atribuído a este LSP e o endereço IP do MN e essa interpretação deve ser a mesma em qualquer LSR deste domínio.

A Figura 4.4 ilustra esses conceitos para um cenário bem simples, composto de um LER de ingresso, alguns LERs de egresso e dois LSRs de ramificação. Somente estes LSRs estão sendo mostrados e que cada túnel externo pode ser formado pela composição de vários LSRs, pertencentes à nuvem MPLS. São mostrados, também, três túneis internos que estão rastreando o MN, onde cada um está em um LER de egresso diferente.

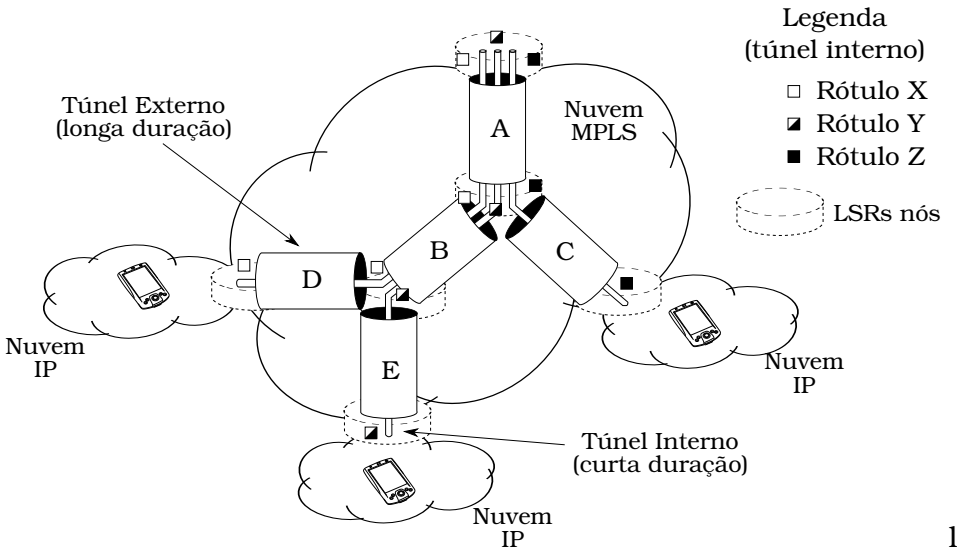


Figura 4.4: Relacionamento entre os túneis externos e os túneis internos em um domínio MPLS.

Conforme pode ser verificado, os túneis internos possuem rótulos de valor constante em toda a rede, onde no roteador de ingresso eles são tunelados dentro do túnel externo e encaminhados pela nuvem MPLS até o LSR que é a ponta final deste túnel. Neste LSR, o túnel externo é retirado e o rótulo do túnel interno é analisado para decidir qual ação deve ser tomada. Se for um LSR de núcleo, este *switch* deverá consultar a tabela LIB pertencente ao túnel interno e tunelar esse túnel em seu LSP de saída. Se for um LSR de egresso, para este túnel interno, este *switch* deverá liberar as informação de rótulo pertencente a este nível de túnel, recuperando o pacote IP do MN. Em seguida, ele deverá encaminhá-lo, através da camada de rede, até o MN.

Note a composição da lista de túneis externos para um determinado túnel interno. Por exemplo, para o túnel interno, cujo rótulo é X, a lista de túneis externos é composta dos túneis A, B e D. Note-se que somente nos LSRs de ramificação e nos LERs de ingresso e egresso que o túnel interno deve ser extraído e consultado.

A Figura 4.5 mostra como os túneis internos são reorganizados, quando ocorre um *handover*. Conforme pode ser observado, a migração envolve apenas reconstruir a lista de túneis externos pertencente a um dado túnel interno. Essa situação pode ser verificada nas figuras 4.5 (a) e (c), em que a lista parcial de túneis externos para o túnel interno, cujo rótulo é X, deixa de ser B e D e passa a ser B e E.

Para implementar o *handover* pró-ativo, há a necessidade de se realizar a replicação de pacotes, nas redes de acesso, conforme pode ser visualizado na Figura 4.5 (b). A replicação é simples de ser implementada e basta fazer o mapeamento do túnel interno em dois túneis externos, simultaneamente, de tal forma que a lista de túneis externos, para este túnel interno, tenha duas ramificações a partir de LSR-k. Esse túnel replicado tem curta duração e permite apenas antecipar a migração do MN. Portanto, a Figura 4.5 ilustra como seria a reconstrução da lista de túneis externos para o túnel interno, cujo rótulo é X, na ocorrência de um *handover* pró-ativo.

Note-se que as informações concernentes aos LSPs que compõem os túneis externos e a lista de túneis externos no qual estão contidos os túneis internos estão distribuídos nos LSRs da nuvem MPLS. Em cada LSR, as informações dos túneis que estão operantes e o papel deste LSR, em cada um destes LSPs, estão guardadas na tabela LIB. Portanto, é através do correto gerenciamento desta tabela que se criam / gerenciam os túneis externos; realiza-se a atribuição dos túneis internos aos túneis externos e, por consequência, realiza-se o rastreamento do MN[1].

A Figura 4.6 mostra uma versão reduzida da tabela LIB em um cenário composto de três

[1]Na realidade, a operação da arquitetura do MPLS recai no gerenciamento da tabela LIB. A criação, manutenção e destruição de LSPs, assim como a classificação e encaminhamento dos pacotes na nuvem MPLS, são baseadas no correto manuseio das entradas desta tabela. Portanto, é natural que a gerência da mobilidade também recaia em manusear corretamente as suas entradas.

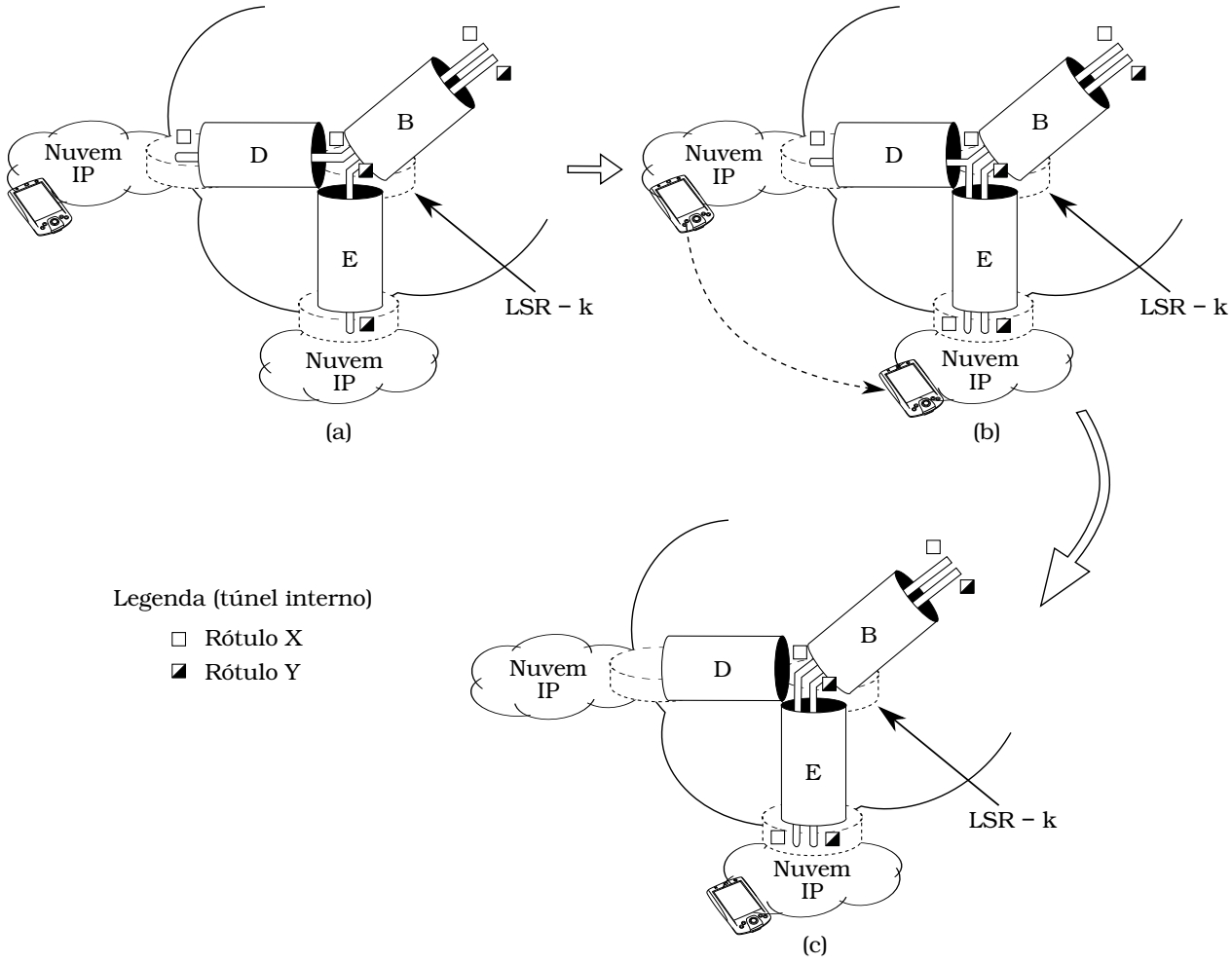


Figura 4.5: Reconstrução da lista de túneis externos, para um dado túnel interno, na ocorrência de um *handover*.

túneis externos, com dois túneis internos mapeados sob eles. Por ser um cenário simplicista, foram considerados apenas cinco LSRs, sendo que um é de ingresso (LSR1), dois de egresso (LSR4 e LSR5), um de ramificação (LSR3) e um de núcleo (LSR2), ou seja, apenas de comutação do túnel externo.

No LSR de ingresso é realizada uma dupla operação de inserção de rótulos. Assim, um pacote destinado ao MN1, por exemplo, irá receber a inserção de rótulo de primeiro nível, após a verificação de seu endereço de destino na tabela FEC, cujo valor será X. Conforme pode ser verificado, o valor do próximo *hop* é nulo. Isso obriga ao LSR reclassificar o "novo" pacote, na procura de uma nova entrada na tabela LIB. Ela será a entrada, cujo rótulo é X, a qual especifica que o pacote deve receber mais um nível de rótulo, cujo valor será A1, e deverá ser encaminhado para o próximo *hop* LSR2.

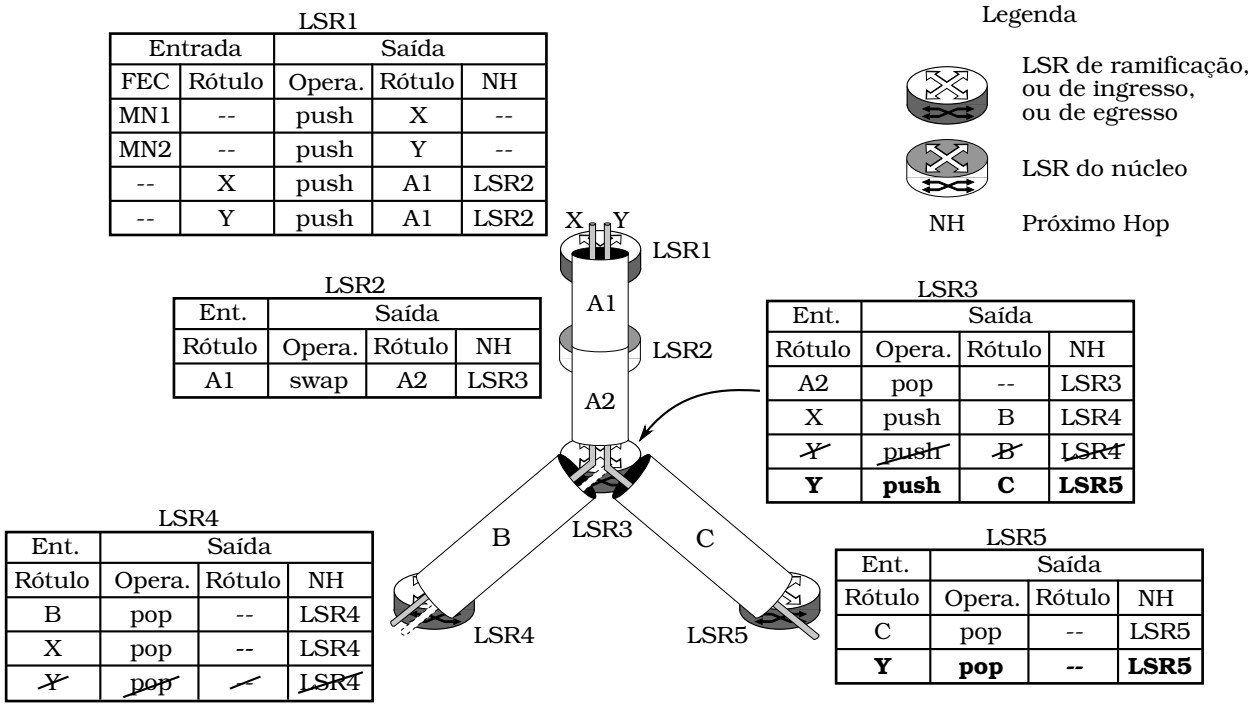


Figura 4.6: Atualização da tabela LIB para rastrear o MN, na ocorrência de um *handover*

Por outro lado, o LSR, que é a ponta final de um túnel externo, sabe que deve analisar o conteúdo do túnel interno devido ao fato de que, após a realização da operação `pop`, o valor atribuído ao próximo *hop* é o seu próprio endereço IP. Note-se que se for realizada uma outra operação de `pop` e o pacote resultante for um pacote IP, então este LSR é o egresso para este túnel interno, ou seja, o pacote está deixando a nuvem MPLS e irá seguir o roteamento IP tradicional. Essa operação está descrita na RFC 3031:

> Note que em um dado LSR, o próximo *hop* de um pacote pode ser o próprio LSR. Neste caso, o LSR poderia necessitar de liberar o rótulo do nível de topo e então "encaminhar" o pacote resultante para si mesmo. Ele poderia então fazer uma outra decisão de encaminhamento, baseado sobre o que restou na pilha de rótulos, após a liberação. Isso pode ser ainda um pacote rotulado, ou pode ser um pacote IP nativo.
> Isso implica que em alguns casos o LSR pode necessitar em operar sobre o cabeçalho IP afim de encaminhar o pacote.
> Se o próximo *hop* do pacote é o LSR atual, então a operação sobre a pilha de rótulos deve ser "liberar a pilha". (Tradução do autor)

No LSR de núcleo, o qual não é de ramificação, a operação do MPLS é baseada em apenas um nível de rótulo, o mais externo. Isso pode ser verificado na figura anterior, ao observar a tabela LIB do LSR2. Somente uma operação é especificada que é a troca de rótulos.

No LSR de ramificação LSR3 a ponta final do túnel mais externo, é realizada a abertura desse túnel. Isso pode ser verificado ao observar que a operação efetuada para a entrada, cujo

rótulo é A2, é o `pop` e o valor de seu próximo *hop* é o seu próprio endereço IP. Assim, ao se recuperar o valor de rótulo do túnel interno, uma nova pesquisa na tabela LIB é efetuada, para decidir quais ações devem ser executadas para encaminhá-lo ao seu destino. Continuando com o exemplo anterior, ao observar o túnel interno, cujo rótulo é X, nota-se que se deve realizar uma nova inserção de rótulo em mais um nível, ou seja, inseri-lo em um túnel mais externo, cujo rótulo de saída será B e destinado ao próximo *hop* LSR4.

Por fim, essa figura mostra também como a tabela LIB, em cada LSR, pode ser modificada para rastrear um MN que efetuou um *handover*. Nesse caso, é assumido que o MN1 se moveu do roteador de acesso LSR4 para o roteador de acesso LSR5. Isso implica que as tabelas LIB dos LSR3, LSR4 e LSR5 devem ser alteradas para poder rastreá-lo. As entradas que estão em negrito são novas inserções, enquanto que as entradas que estão cortadas são as excluídas.

4.2.1 Gerência de Rótulos

Conforme apontado anteriormente, o significado do rótulo tem interpretações diferentes quando analisado pelos diferentes níveis de empilhamento, ou seja, pelo túnel externo, ou pelo túnel interno. Para o túnel externo, ele tem um valor cujo significado é local ao LSR, enquanto para o túnel interno, o seu valor tem um significado que é global ao domínio MPLS.

Portanto, a gerência de rótulos possui duas funções bem específicas. A primeira é como realizar a divisão do espaço de rótulos entre os dois níveis de empilhamento de rótulos e a segunda é como realizar a alocação do rótulo para o túnel interno, de modo a garantir sua unicidade dentro do domínio.

Espaço de Rótulos

Conforme pode ser percebido, existem dois espaços de rótulos que são distintos entre si. Um pertencente ao túnel externo e o outro pertencente ao túnel interno. O modo mais simples de se realizar essa gerência de rótulos é, para sistemas que permitam a criação de múltiplos espaços de rótulos, alocar um espaço para cada um deste túneis. A consequência é a independência sobre os valores alocados e o aumento da quantidade de túneis possíveis em cada nível de empilhamento.

A especificação do MPLS [6] discrimina dois tipos de espaço de rótulos possíveis: o espaço pertinente à interface e o espaço pertinente à plataforma. Segundo a especificação, o espaço pertinente da interface seria apropriado para a tecnologia ATM (*Asynchronous Transfer Mode*), em que cada interface de rede seria dotada de um espaço de rótulos próprio. Já o espaço pertinente à plataforma seria único e disponível para os elementos de rede que não podem ser classificados no outro tipo.

Assim, para sistemas baseados em espaço de rótulos por plataforma e que tenham poder de

processamento condizente, é interessante aliviar essa restrição de que esse espaço seja único e permitir o uso de mais do que um espaço de rótulos. Com isso, cada nível de empilhamento de rótulos teria um espaço de rótulo próprio. A RFC 5331 [70] já permite o uso de outros tipos de espaço de rótulos, além dos dois previamente mencionados.

Uma segunda proposta de gerência de rótulos é considerar que somente um espaço de rótulos esteja disponível, ou que o espaço seja único por interface. Para efeito de discussão, embora as mesmas ideias possam ser aplicadas a outras tecnologias de camada de enlace utilizando codificações próprias de rótulos, tais como o ATM e o *frame relay*, vamos apresentar uma proposta considerando somente a codificação dos rótulos utilizando o cabeçalho *SHIM* [71].

O espaço de codificação de rótulos no cabeçalhos *SHIM* é planar, ou seja, não possui uma estrutura hierárquica que permita associar um grupo de valores de rótulos com o seu nível na pilha de rótulos. Portanto, para permitir que os dois tipos de túneis coexistam harmoniosamente, em que o túnel externo pode ter o valor de seu rótulo alocado aleatoriamente, pois o seu valor é significativo apenas localmente; e o túnel interno deve ter o seu valor controlado globalmente, o espaço de codificação deve ser reestruturado a fim de permitir a gerência de atribuição de rótulos.

Normalmente, a escolha de qual rótulo será utilizado em um dado LSP é uma decisão local. Assim, quando um LSR recebe uma mensagem de requisição de alocação de rótulos, ele escolhe um rótulo que esteja livre para alocação, ou escolhe um valor específico, de acordo com suas políticas locais. Um segundo ponto a ser observado é que normalmente o valor da FEC, para o qual está sendo requisitado um rótulo, também é conhecido.

Assim, a nossa proposta é dividir o espaço de rótulos em duas partes, uma pertinente aos túneis externo e outra pertinente aos túneis internos. Quando houver a requisição de alocação de rótulos, a escolha de qual região deve ser usada, pode ser baseada em máscaras de bits (*bitmask*) e, esta máscara, é aplicada dependendo do valor da FEC para o qual o LSP está sendo criado ou do nível do túnel que está sendo solicitado.

Alocação de Rótulo para o Túnel Interno

Conforme já mencionado, o valor de rótulo do túnel interno deve ser global, dentro do domínio MPLS, e é ortogonal ao modo como está sendo feito a divisão do espaço de alocação de rótulos. Basicamente, há três formas de garantir, ou pelo menos maximizar a probabilidade de unicidade do valor alocado. A primeira delas é utilizar um servidor central que permita requisitar um valor de rótulo para uma dada FEC. Essa não é uma solução muito atraente devido aos óbvios problemas de escalabilidade e confiabilidade inerentes.

A segunda consiste em permitir que os LERs de egresso do túnel mais interno escolham qual deverá ser o mapeamento entre a FEC e o valor do rótulo deste túnel. Também não é uma

solução muito interessante, pois, para garantir a unicidade do valor alocado, deve-se dividir o espaço de rótulos disponíveis para o túnel interno entre todos os possíveis LERs de egresso, para este túnel, de modo que não haja sobreposição de valores entre eles e, quando houver uma alocação de rótulos, deve-se publicar a escolha para todos os LSRs que são pontas do túnel externo, para que fiquem em sincronia entre si. É uma solução que apresenta problemas de gerência, principalmente quando a configuração da rede se altera, além de restringir a quantidade de MNs que podem estar sendo rastreados, simultaneamente. Outro problema é a geração de tráfego de controle bastante elevado, aumentando a latência do sistema.

A terceira consiste em obrigar que todos os LSRs, que são pontas dos túneis mais externos, rodem o mesmo algoritmo de alocação de rótulos. Essa é a solução mais interessante, pois a unicidade é garantida pelo determinismo do algoritmo escolhido e a distribuição da alocação é garantida por concepção.

A ideia consiste em criar uma função que recebe como entrada a FEC pertinente do MN e fornece como saída o rótulo associado. Ele deve ter como características principais, o determinismo da função matemática e a repetibilidade do resultado. A função *hash*[2] é uma boa candidata a satisfazer esse requisitos, utilizando-se um algoritmo que forneça uma baixa colisão entre os resultados de saída.

Um algoritmo bem simples, porém eficiente é, baseado no fato de que qualquer endereço IP é dividido em duas partes: o endereço de rede e o endereço de *host*. Assim, basta determinar que a faixa de endereços de *host* do MN contenha menos do que 20 *bits*. Como o espaço destinado ao *rótulo* na codificação do cabeçalho *SHIM* possui 20 *bits*, basta apenas considerar a parte do endereço IP, pertinente ao endereço de *host* do MN, como sendo o rótulo a ser atribuído ao LSP interno.

O *bit* mais significativo do rótulo pode ser usado, para os sistemas que utilizarem um único espaço de rótulos, como um separador. Por exemplo, se for 0, pertence ao espaço de rótulo do túnel externo. Se for 1, pertence ao espaço de rótulos do túnel interno. Assim, com esse exemplo, teríamos 19 *bits* para o rótulo interno, perfazendo um total de mais de 500 mil MNs rastreados simultaneamente. Se não for o suficiente, pode-se usar esquemas de agregação de MNs, conforme proposto pela arquitetura MPA.

4.3 Aplicabilidade desta Proposta na Arquitetura MPA

O modo como a arquitetura MPA foi concebida, com relação à criação e gerência de túneis, e a solução proposta de rastreamento do MN sobre o MPLS são bem próximas entre si, tendo, com isso, um mapeamento praticamente de 1:1.

[2]A função *hash* é qualquer procedimento bem definido, ou função matemática, que converte uma quantidade de dados, normalmente de grande volume, em um *datum*, usualmente um inteiro simples, de tamanho fixo, que serve como indexador de um vetor, ou matriz.

A arquitetura MPA estabelece que deva existir um túnel P2MP entre o MAR de ingresso e os MARs de egressos, criando uma estrutura em árvore. Este túnel tem como finalidade criar as possíveis rotas que o MN pode usar, quando estiver sendo rastreado por um domínio MPA.

Por outro lado, esse túnel P2MP não tem a ideia de replicar os pacotes em todas as saídas possíveis, tendo, portanto, uma aplicação um pouco diferente de um túnel P2MP padrão. Os pacotes dos MNs, ou seja, os túneis internos, são discretizados dentro deste túnel e roteados de forma individual, até o seu destino final. Conforme mencionado anteriormente, a replicação de pacotes só deve ocorrer quando estiver sendo usado o *handover* pró-ativo.

A vantagem de se utilizar um túnel P2MP é a já existente sinalização que permite a criação, manutenção e destruição de segmentos de túneis [59, 60], que minimiza as alterações necessárias nos *firmware* dos LSRs, além de se apoiar na implementação dos túneis P2MP fornecidos pelos fabricantes.

Assim, há duas formas de se criar esses túneis P2MP. A primeira consiste em criar túneis P2P entre os MARs e o túnel P2MP passa a ser, então, uma lista de interligação entre estes túneis P2P. Logo, qualquer túnel interno que fosse criado para rastrear o MN somente poderia ser feito se estivesse dentro de um conjunto de túneis externos que estivessem contemplados nesta lista.

A segunda forma consiste em se criar um túnel P2MP, entre o MAR de ingresso e os MARs de egresso, conforme explicitados por algum padrão de sinalização, mas impedindo que os túneis internos sejam replicados em cada MAR.

Com relação a forma de sinalização empregada na criação, manutenção e destruição destes túneis externos (P2MP) é do tipo estática, ou seja, de longa duração.

Por fim, as duas outras questões podem ser aplicadas diretamente, conforme foram especificadas na seção anterior, sem modificações, ou seja, a gerência do espaço de rótulos e o algoritmo de alocação de rótulos para o túnel interno.

4.4 Considerações do Capítulo

Esse capítulo discutiu como as principais propostas de rastreamento do MN, apresentadas pela literatura, poderiam ser aplicadas na arquitetura MPA. Conforme foi ilustrado, nenhuma delas poderia ser aplicada de forma satisfatória, devido, principalmente, à falta de generalidade de seus mecanismos de rastreamento propostos. Assim, foram mostradas, além das dificuldades em se adaptar essas soluções na arquitetura MPA, as deficiências apresentadas em rastrear o MN de forma eficiente.

Em seguida, foi apresentada a nossa proposta de rastreamento do MN, a qual possui como característica essencial a generalidade da solução. Ela é baseada em dois níveis de empilhamento de rótulos, sendo o papel do rótulo mais externo permitir o estabelecimento dos

possíveis caminhos que um MN pode percorrer em um domínio adminstrativo em MPLS e o papel do rótulo mais interno é rastrear o MN dentro de uma rede MPLS, ou seja, ele identifica o MN em um domínio adminstrativo MPLS.

Como a proposta de localização do MN envolve mais do que um nível de empilhamento de rótulos, há a necessidade de gerenciá-los. Assim, foi apresentada o escopo de interpretação dos rótulos, em cada nível de empilhamento, para cada LSR do domínio. Foi discutido também como o espaço de rótulos poderia ser dividido para atender as demandas de gerência. Por fim, devido ao caráter global do significado o rótulo do túnel mais interno, foi apresentada uma proposta de associação entre o endereço IP do MN e o rótulo do túnel interno, que possa ser aplicada de forma distribuída para atender os requisitos de unicidade deste rótulo.

Por último, foi mostrado como essa proposta pode ser utilizada para rastrear o MN na arquitetura MPA. O mapeamento entre a arquitetura MPA e a proposta apresentada é de aproximadamente 1:1 e a arquitetura utiliza apenas um subcaso das possibilidades apresentadas pela proposta.

Como a proposta não viola e nem altera as especificações do protocolo MPLS, as extensões do protocolo, divulgadas pelo IETF, são absorvidas naturalmente por ela. Por exemplo, questões de QoS e tolerância a falhas são facilmente incorporadas a esta proposta através da aplicação das RFCs 3270 [67] e 3469 [69], respectivamente.

Finalmente, a implantação desta proposta em redes MPLS legadas é bastante simples, pois as alterações exigidas são aplicadas apenas em LSRs especiais, que são as pontas dos túneis externo. Note-se que, com relação aos túneis P2MP, eles estão sendo utilizados para uma aplicação diferente da que foram originalmente concebidos, ou seja, a difusão de pacotes *multicast*, e, portanto, não há uma replicação de pacotes para todos os segmentos que compõem o túnel.

O próximo capítulo apresenta a implementação da proposta aqui apresentada.

Capítulo 5

Implementação do Protocolo MPLS e Validação

A necessidade da implementação de um protótipo do protocolo MPLS [6], como um produto de apoio às pesquisas realizadas na FEEC, surgiu em 2001. Embora, naquela época, existiam algumas soluções em *software* livre que nos permitiam utilizá-las para os propósitos acima mencionados, elas foram descartadas após uma avaliação preliminar, devido as suas limitações. Dessa forma, chegou-se à conclusão que seria mais interessante possuir uma implementação própria deste protocolo [72, 73].

Resgatando rapidamente o histórico dessa implementação, pode-se comentar, em linhas gerais, que ela foi concebida para sistemas Linux, dotados de interfaces de redes do tipo *ethernet* e pode ser dividida em dois subsistemas bem distintos: o plano de encaminhamento de pacotes, o qual foi implementado no espaço do núcleo e o plano de controle do sistema, o qual foi implementado no espaço do usuário. Eles são ortogonais e conectados através de uma interface simples e estável, a qual permite que cada um possa ser projetado e implementado isoladamente.

Com relação ao plano de encaminhamento foi implementado o básico da inserção de rótulos e, por consequência, o seu encaminhamento em uma rede MPLS, conforme descrito na RFC 3031 [6]. Em particular, a implementação se restringiu ao protocolo IPv4, para a atribuição de FEC no LER de ingresso e, como foi projetada para interfaces de rede do tipo *ethernet*, para a determinação do próximo *hop*. Foi implementado, também, a criação, alteração e destruição de LSPs, contendo um único nível de empilhamento de rótulos e considerando somente túneis do tipo P2P.

A proposta de integração da arquitetura MPA com o protocolo MPLS exige o uso de túneis do tipo P2MP, com a restrição de que a replicação de pacotes será controlada e somente ocorrerá nos segmentos de LSP de saída especificados. Ela prevê também a necessidade de realizar

empilhamento de rótulos em túneis MPLS. Infelizmente, nessa implementação do plano de encaminhamento, houve um acoplamento bastante forte entre as estruturas de dados que definem o segmento de entrada, em um LSR, com as estruturas de dados que definem os segmentos de saídas, para esse mesmo LSR, tornando a implementação dessas extensões bastante onerosa e de difícil manutenção.

Assim, devido aos problemas mencionados com a implementação anterior, decidiu-se que, ao invés de estender essa implementação, criar uma nova, agregando todas as funcionalidades já existentes e adicionando as novas exigidas. Com isso, este capítulo irá detalhar o projeto desta nova implementação, ou seja, a do plano de encaminhamento. A seção 5.1 detalhará o processo de análise, projeto e implementação do protocolo MPLS, adotando o paradigma de orientação a objetos. A seção 5.2 discorrerá sobre os testes executados nesta implementação, com o objetivo de verificá-la e validá-la, não somente com relação à correta realização da RFC 3031, mas também à correta integração com a implementação da arquitetura MPA. Por fim, a seção 5.3 tratará das considerações sobre este Capítulo.

5.1 Análise, Projeto e Implementação do Módulo MPLS

Conforme mencionado anteriormente, o módulo MPLS foi desenvolvido para o sistema operacional Linux, o qual é escrito em uma linguagem procedural, mais especificamente, a linguagem C. Apesar de existirem metodologias específicas para se projetar e implementar sistemas neste tipo de paradigma, nada impede que outros paradigmas de desenvolvimento de *software* sejam utilizados.

Devido a esta liberdade, para o desenvolvimento deste módulo foram empregadas as práticas recomendadas para o desenvolvimento de sistemas ditadas pela Engenharia de *Software*, em especial, as do paradigma de orientação a objetos – OO (*Object Oriented*).

O desenvolvimento de *software* consiste basicamente de três etapas [74, 75, 76]: a análise do sistema, o projeto do *software* e a sua implementação propriamente dita. A análise do sistema pode ser categorizada em três atividades, as quais são: a análise de requisitos, a especificação funcional e a arquitetura do *software*.

A análise de requisitos tem por objetivo estabelecer os limites do *software* que está sendo especificado, assim, ela define quem são os elementos externos que interagem com o sistema e quais são as ações executadas por ele. A especificação funcional é a documentação que descreve o comportamento requisitado de um sistema. Por fim, a arquitetura do *software* é a estrutura, ou estruturas, de um sistema, que contempla os componentes do *software*, as propriedades que são visíveis externamente e os relacionamentos entre eles.

O projeto do *software* tem por finalidade transformar o modelo obtido na análise de sistema em uma aplicação implementável, adicionando-lhe componentes que são dependentes

da tecnologia adotada. Isso inclui, não somente aplicar soluções de alto nível, como por exemplo, organizar o sistema em subsistemas, identificar elementos que são concorrentes entre si e alocar os subsistemas para processos e tarefas; mas também as soluções de baixo nível, como por exemplo, projetar algoritmos eficientes para serviços complexos, otimizar a solução para a aplicação em desenvolvimento e adicionar novos elementos pertinentes ao domínio da implementação.

Por fim, a implementação é o processo de escrever, testar, depurar e manter o código fonte de um programa computacional. Obviamente o código fonte é escrito em uma linguagem de programação e o código gerado pode ser o resultado da modificação de um programa já existente ou algo completamente novo.

Para auxiliar no projeto e documentação deste módulo, será utilizada a linguagem de modelagem de objetos, ou seja, a UML (*Unified Modeling Language*) [77], por ser uma linguagem universal, de fácil assimilação e bastante usada no paradigma OO. Com isso, serão aplicados esses métodos e metodologias para especificar e implementar o módulo MPLS.

5.1.1 Análise do Sistema

A fim de evitar que se tenha um sistema abrangente e de difícil implementação, o primeiro passo consiste em se restringir o domínio no qual o módulo será utilizado. Assim, ele será empregado em roteadores/*switches* cujo sistema operacional é o Linux, utilizando a versão mais recente, que é a 2.6.x [78]. Restringindo um pouco mais, a interface de rede suportada pelo sistema será a *ethernet*, cuja codificação do cabeçalho MPLS é a especificada pelo cabeçalho SHIM. Por fim, o protocolo de camada superior suportado será a versão atual do protocolo IP, ou seja, o IPv4.

O Linux foi escolhido por ser um sistema operacional livre, robusto, eficiente, de código aberto, cujo código fonte é disponível e de fácil alteração. A interface de rede do tipo *ethernet* foi escolhida por ser a mais comum de ser encontrada nos computadores comercializados atualmente e o protocolo de rede IPv4 por ser a versão do protocolo de camada de rede em uso. Dentre os vários modos em que se pode determinar se um pacote IPv4 pertence a uma dada FEC, foi escolhido o que a especifica através do endereço de destino, utilizando a notação do CIDR. É justificada essa escolha, pois casa perfeitamente com o modo como os atuais roteadores IPs determinam os caminhos de saída para estes pacotes.

Os requisitos do módulo MPLS podem ser melhor visualizados através do uso do diagrama de caso de uso, ilustrado na Figura 5.1. Neste diagrama, o sistema é dividido em duas partes lógicas: a primeira, que cuida da gerência do sistema; e a segunda, que cuida do processamento de pacotes, provenientes dos protocolos de rede.

Para a primeira parte do sistema, relativa à gerência, cujas interações externas estão representadas através do ator genérico `Gerenciador`, tem a função de configurar e atualizar os

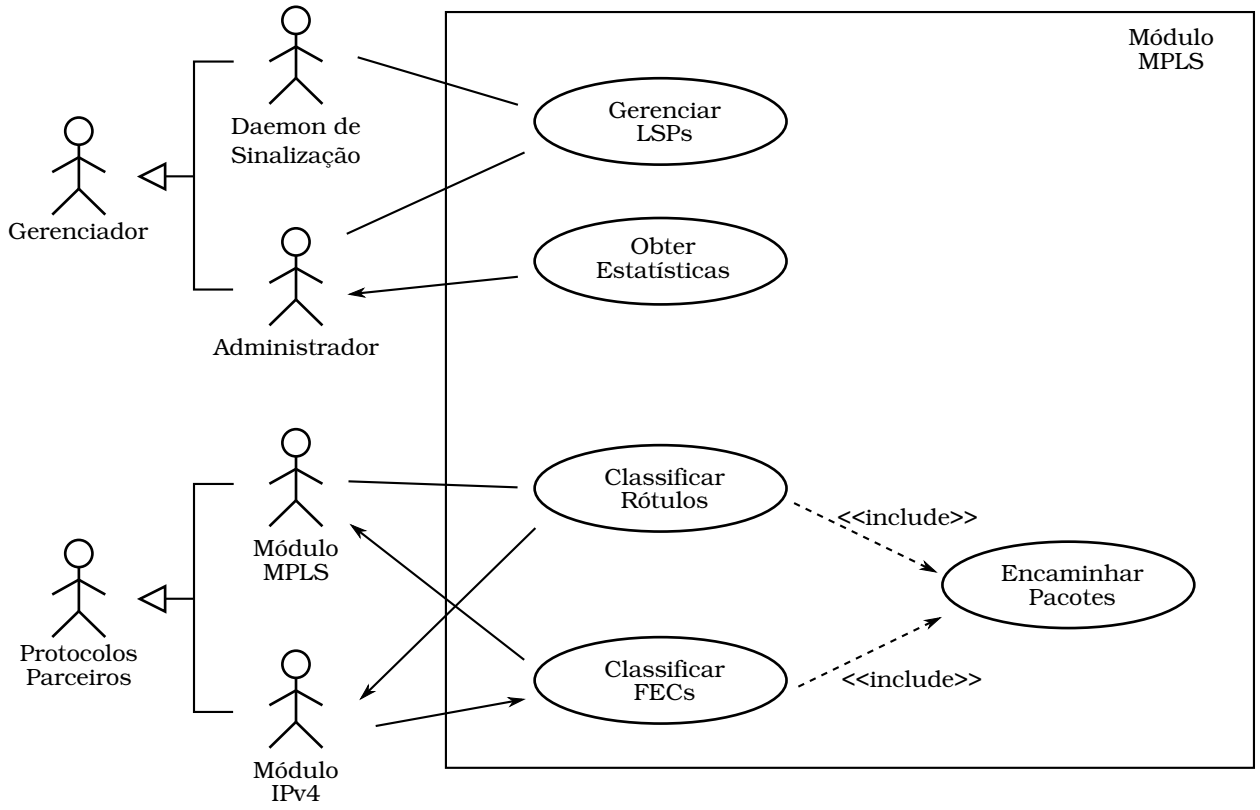


Figura 5.1: Diagrama de caso de uso mostrando as interações externas com o módulo MPLS.

LSPs que estão operacionais em um dado LSR. O seu papel é de criar, atualizar e destruir o trecho de um LSP que está apoiado neste LSR. Uma segunda interação prevista para este ator é a obtenção de estatísticas do sistema. Dois atores mais especializados estão previstos, embora não sejam limitados a eles, os quais são: o ator `Daemon de Sinalização` e o ator `Administrador`.

O ator `Daemon de Sinalização` representa a realização de um protocolo de sinalização qualquer, como por exemplo, o protocolo RSVP que está implementado na arquitetura MPA e permite a configuração dinâmica do sistema. Não é previsto que esse `daemon` obtenha dados estatísticos, uma vez que essa não é sua função precípua. O ator `Administrador` é um ator que idealiza o operador humano e permite a configuração estática do sistema. Esse é um tipo de ator que deseja obter dados estatísticos a fim de se obter dados de performance do sistema. Note-se que isso não impede que um software de gerência assuma o papel do ator `Administrador`.

As interações descritas acima são respondidas pelo sistema através dos casos de uso `Gerenciar LSPs` e `Obter Estatísticas`. O caso de uso `Gerenciar LSPs` tem o papel de manipular, ou seja, criar, atualizar e destruir, as entradas das tabelas da máquina MPLS.

Existem várias tabelas internas ao sistema, as quais podemos citar: a tabela de rótulos de entrada, a tabela de FECs e a tabela de rótulos de saída, também conhecida por NHLFE (*Next Hop Label Forwarding Entry*).

É seu papel também conectar as entradas dessas tabelas entre si, a fim de estabelecer um segmento do LSP. Essa conexão deve ser de tal forma a permitir que se tenha a possibilidade de criar vários níveis de empilhamento de rótulos e permitir, também, a replicação de um pacote em vários segmentos de saída.

De um modo genérico, se a tabela a ser configurada for a tabela de FEC, este LSR estará atuando como um LER de ingresso. Por outro lado, se a tabela a ser configurada for a tabela de rótulos e houver uma entrada na tabela NHLFE especificando um rótulo de saída, o LSR será considerado de núcleo. Por fim, se a tabela a ser configurada for a tabela de rótulos e a entrada associada na tabela NHLFE indicar a extração do rótulo, esse LSR estará atuando como um LER de egresso, no qual a ação esperada pode ser tanto reencaminhar o pacote através do protocolo MPLS, retirando um nível da pilha de rótulos, quanto encaminhá-lo para o processamento pelo protocolo IP.

O caso de uso `Obter Estatísticas` tem a finalidade de extrair os parâmetros de utilização dessas tabelas. Assim, para cada tabela, existem alguns parâmetros que podem ser contabilizados, como por exemplo, a quantidade de pacotes recebidos/enviados; a quantidade de *bytes* recebidos/enviados e a quantidade de pacotes descartados, devido a erros no pacote, ou devido a erros de processamento.

Para a segunda parte do sistema, relativa ao processamento de pacotes, cujas interações externas estão representadas através do ator genérico `Protocolos Parceiros`, tem a função de processar os pacotes entrantes. As ações executadas sobre eles dependem do tipo de protocolo de camada superior à camada de enlace que está sendo analisado. Se for um pacote IPv4, o sistema deverá procurar por uma entrada na tabela de FEC que case com as informações contidas no cabeçalho do pacote. Se for um pacote MPLS o sistema deverá procurar por uma entrada na tabela de rótulos que case com o rótulo pertinente ao cabeçalho SHIM.

Se o pacote for aceito para ser processado pelo sistema, após a sua computação, o pacote sainte poderá ser tanto encapsulado pelo protocolo MPLS, quanto pelo protocolo IPv4, cuja implementação já está presente no núcleo do Linux, dependendo apenas do papel deste LSR em um dado LSP. Com a finalidade de representar qual o protocolo de um dado pacote que está sendo encapsulado, foram criados dois atores mais especializados do ator `Protocolos Parceiros`: o ator `Módulo MPLS` e o ator `Módulo IPv4`.

Pelo lado do módulo MPLS, as ações iniciadas pelos atores `Módulo MPLS` e `Módulo IPv4` são executadas através dos casos de uso `Classificar Rótulos` e `Classificar FECs`, respectivamente. Note-se que, após o processamento inicial de cada um deles, é realizado o mesmo processamento para finalizar o caso de uso e ele é comum aos dois. Esse fato é repre-

sentado no diagrama pela associação estereotipada `<<include>>` de cada um desses casos de uso com o caso de uso `Encaminhar Pacotes`.

O caso de uso `Classificar Rótulos` é executado sempre que um pacote MPLS entra no sistema. Ele deve procurar por uma entrada na tabela de rótulos que case com o rótulo do pacote. Se não for encontrada, esse pacote deve ser descartado. Em caso de sucesso, as informações pertinentes ao cabeçalho SHIM devem ser extraídas do pacote, obtendo novamente um pacote com um nível acima. Uma lista com as possíveis entradas da tabela NHLFE são obtidas dessa tabela. Nesse momento, o caso de uso `Encaminhar Pacotes` assume o resto do processamento.

O caso de uso `Classificar FECs` é executado sempre que um pacote IPv4 entra no sistema. Ele deve procurar por uma entrada na tabela de FECs que case com o endereço de destino do pacote. Se não for encontrada uma entrada, esse pacote deve ser devolvido ao processamento IPv4, para que ele o entregue ao destino via roteamento *hop* a *hop*. Por outro lado, em caso de sucesso, uma lista com as possíveis entradas da tabela NHLFE são extraídas e o caso de uso `Encaminhar Pacotes` assume o resto do processamento.

O caso de uso `Encaminhar Pacotes` é executado como parte integrante dos outros dois e serve para finalizar as ações iniciadas por eles. Ele inicia com uma lista das possíveis saídas para um dado pacote. Para cada pacote, uma de cada três ações pode ser executada: empilhar um rótulo, encaminhar o pacote para o próximo *hop* ou diminuir o nível de empilhamento.

A ação de empilhar um rótulo, conhecida também por operação `push`, implica em criar um espaço entre o cabeçalho de camada de enlace e o cabeçalho de camada superior, para conter a codificação SHIM com as informações pertinentes à saída. Em seguida, deve-se obter a lista dos segmentos de saída associados, pois a ação de empilhar implica que haverá, pelo menos, dois níveis de rótulos em um pacote e deve-se reexecutar este caso de uso novamente, ou seja, o caso de uso `Encaminhar Pacotes`.

O fato de inserir o cabeçalho SHIM em um pacote pode extrapolar o tamanho máximo permissível a ele. Para que o sistema continue a funcionar a contento, existem algumas soluções que podem ser empregadas e dependem do papel do LSR em um LSP. Se for um LER de ingresso e o bit DF (*Don't Fragment*) do cabeçalho IPv4 estiver ativado, o sistema deve gerar uma mensagem ICMP do tipo *Destination Unreachable*, para a origem do pacote, informando o novo MTU (*Maximum Transmission Unit*) e descartar o pacote. Se o bit DF estiver desabilitado, o sistema pode fragmentar o pacote, ou descartá-lo. Vamos ficar com a primeira opção, a fim de garantir a máxima entrega de pacotes.

O maior problema surge quando se deve fazer a inserção de um novo cabeçalho SHIM no meio da nuvem MPLS, quando não se sabe, *a priori*, qual o protocolo que está sendo transportado. Nesse caso, há, pelo menos, duas soluções. A primeira consiste em se utilizar de extensões à sinalização, para informar ao LER de ingresso qual a MTU total do caminho. A

segunda, e mais fácil de se implementar, é assumir que o protocolo de passagem será somente o IPv4. Assim, basta destunelar o pacote, à procura do pacote IPv4, verificar o estado do bit DF e agir conforme o LER de ingresso.

A ação de encaminhar o pacote para o próximo *hop*, conhecida também por operação `swap`, implica que se deve criar um espaço entre o cabeçalho de camada de enlace e o cabeçalho de camada superior, para conter a codificação SHIM com as informações pertinentes à saída. Em seguida, deve-se encaminhar o pacote para as funções de camada 2 do núcleo para que este o entregue à interface de rede, cujo destino é o próximo *hop*.

A ação de diminuir o nível de empilhamento, conhecida também por operação `pop`, implica em analisar o conteúdo do campo próximo *hop* que está nesta entrada, proveniente da tabela NHLFE. Se ele apontar para o próprio LSR, o caso de uso `Classificar Rótulos` é executado para processar as informações do nível superior. Se o conteúdo do campo próximo *hop* for diferente do endereço IP do próprio LSR e o campo de rótulo for zero, o pacote deve ser entregue para o módulo IPv4, para que este dê o encaminhamento final ao pacote. Essa é a descrição comportamental dada pela RFC 3031. Para a nossa implementação, acrescentamos campos a mais na tabela NHLFE para descrever a ação que deve ser executada, sem a necessidade de ficar observando tais campos. Daremos mais detalhes desta solução adiante.

A RFC 3031 estabelece que um LSP é uma concatenação de vários LSRs, que vistos agrupadamente, formam um caminho entre o LER de ingresso e o LER de egresso. Ele possui como propriedades o fato de que, no LER de ingresso, existe uma tabela FEC que captura pacotes IP e, quando existe um casamento entre uma entrada nesta tabela e os dados pertinentes ao pacote IP, os envia para serem processados pela nuvem MPLS. Outra característica é que ela diferencia entre um LSP e um túnel MPLS. Um LSP possui apenas um nível de empilhamento de rótulos. Por outro lado, um túnel MPLS é um trecho de LSRs agrupados, dentro de uma nuvem MPLS, que permite a passagem de outros túneis ou LSPs em seu interior.

Conforme pode ser deduzido, de uma forma generalista, um LSP é um túnel MPLS cujo nível de empilhamento é unitário e que, na entrada deste túnel, a tabela FEC mapeia pacotes IP para dentro deste túnel. Assim, para o projeto do módulo MPLS, podemos considerá-los como sendo a mesma entidade, ou seja, um túnel MPLS.

Retornando à definição de túneis dada pela RFC 3031, vemos que um túnel é a união de vários segmentos, os quais são conectados em cada LSR que compõe o caminho. Uma definição trivial de segmentos é: “qualquer uma das partes nas quais alguma coisa pode ser dividida”. Assim, um túnel MPLS pode ser considerado como sendo composto de segmentos que unem os vários LSRs entre si, sendo o papel de cada LSR realizar o chaveamento entre um segmento anterior para o segmento posterior. A Figura 5.2 ilustra essa ideia.

Conforme pode ser visto, o LSP mostrado é composto pela união de vários segmentos. Normalmente, cada segmento é composto de duas pontas, exceção para o LSR de ingresso do

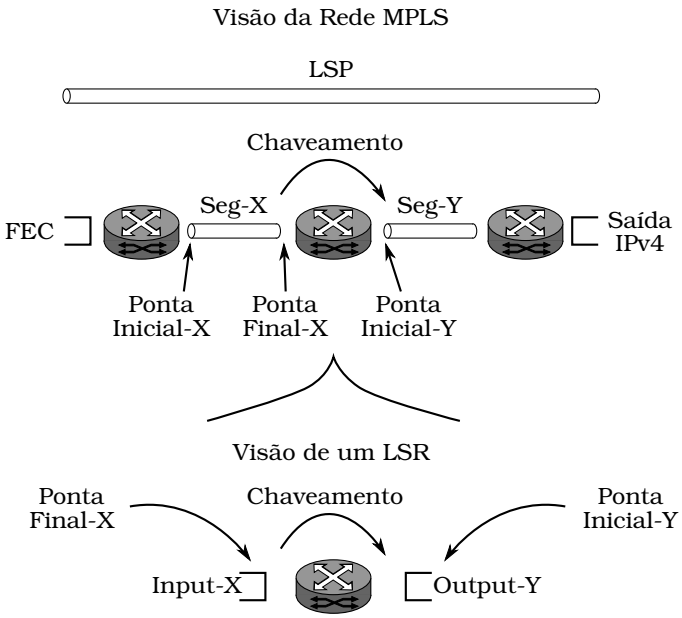


Figura 5.2: Composição de um LSP através de segmentos. Está sendo mostrado a visão da rede (acima) e a visão de um LSR (abaixo).

túnel, o qual o segmento de entrada possui apenas a ponta final e atua como um sorvedouro de pacotes IP; e para o LSR de egresso, no qual o segmento de saída possui apenas a ponta inicial e atua como um vertedouro de pacotes IP. Essa seria a visão que a rede possui, sobre como é a composição de um LSP.

Por outro lado, o papel do LSR é o de comutação de pacotes entre a ponta final de um segmento para a ponta inicial de um outro segmento. Assim, para ele, a ponta final é a entrada de dados, chamada aqui de `Input`, enquanto a ponta inicial é a saída de dados, chamada aqui de `Output`. Note-se que a replicação de pacotes ocorre na comutação, em que o LSR deve enviar o mesmo pacote para várias saídas, simultaneamente, e cada saída é um segmento, conforme a definição acima.

Portanto, para o projeto do módulo MPLS, usar a visão do LSR é mais interessante, pois lida diretamente como esses objetos devem ser implementados. A Figura 5.3 mostra, para um dado LSR qualquer, como deve ser a organização dessas pontas, chamadas, daqui para frente, de segmentos, por referir-se à composição de segmentos distintos, para comutar os pacotes da seção de entrada para a seção de saída.

Essa figura mostra vários exemplos de segmentos, em um LSR, e como eles podem ser configurados através da interligação entre os segmentos de entrada e os segmentos de saída. Em particular, na Figura 5.3 (a) pode-se ver, no LSR de núcleo, como é realizada a replicação de pacotes, quando um segmento de entrada é conectado a dois segmentos de saída. Note-se também que o ato de classificar o pacote IP em uma dada FEC, representada pelo segmento

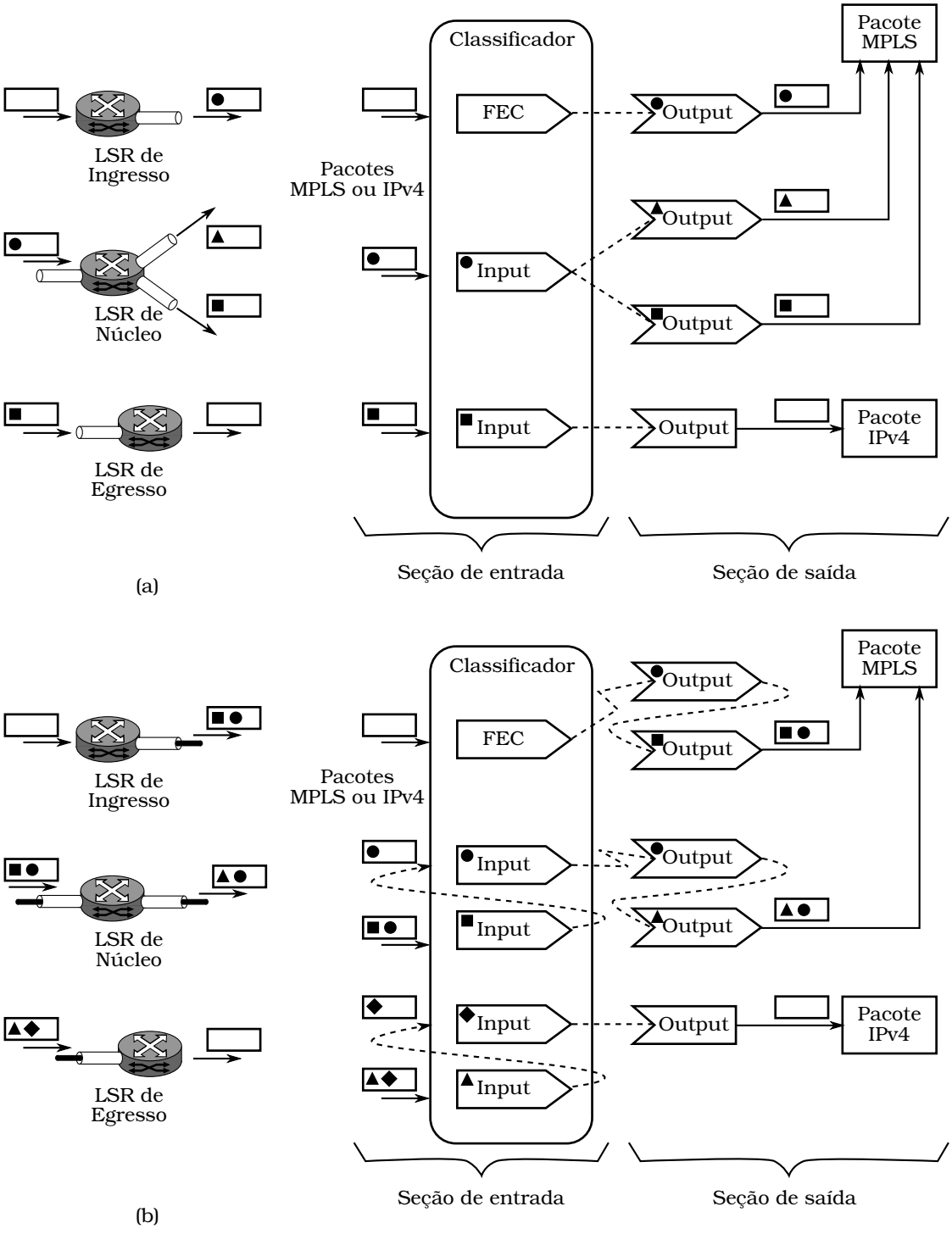


Figura 5.3: Um cenário de instanciação de túneis, em um LSR, considerando-o como um roteador de ingresso, de núcleo e de egresso. Em (a) são mostrados vários túneis P2P com um nível de empilhamento de rótulos, sendo que no roteador de núcleo é um túnel P2MP. Em (b) são mostrados vários túneis P2P com dois níveis de empilhamento de rótulos.

FEC, é também um segmento de entrada.

A Figura 5.3 (b) mostra como é realizado o empilhamento de rótulos, nos vários papeis que um LSR pode assumir, através da configuração entre os segmentos. Essa é uma configuração típica utilizada pela arquitetura MPA para rastrear o MN. Através do encadeamento de segmentos pode-se fazer a extração ou a inserção do pacote dentro de um novo túnel. Conforme pode ser notado, a saída do pacote de uma nuvem MPLS é representada por um segmento de saída que não permite o encadeamento com novos segmentos, como por exemplo, a extração do pacote IPv4.

Os segmentos de entrada e saídas podem ser visualizados como objetos, dentro do paradigma de análise aqui usado. Assim, a Figura 5.4 mostra o diagrama de classes para a nossa implementação, em que todos os conceitos são agrupados e são mostradas as possíveis interações entre eles.

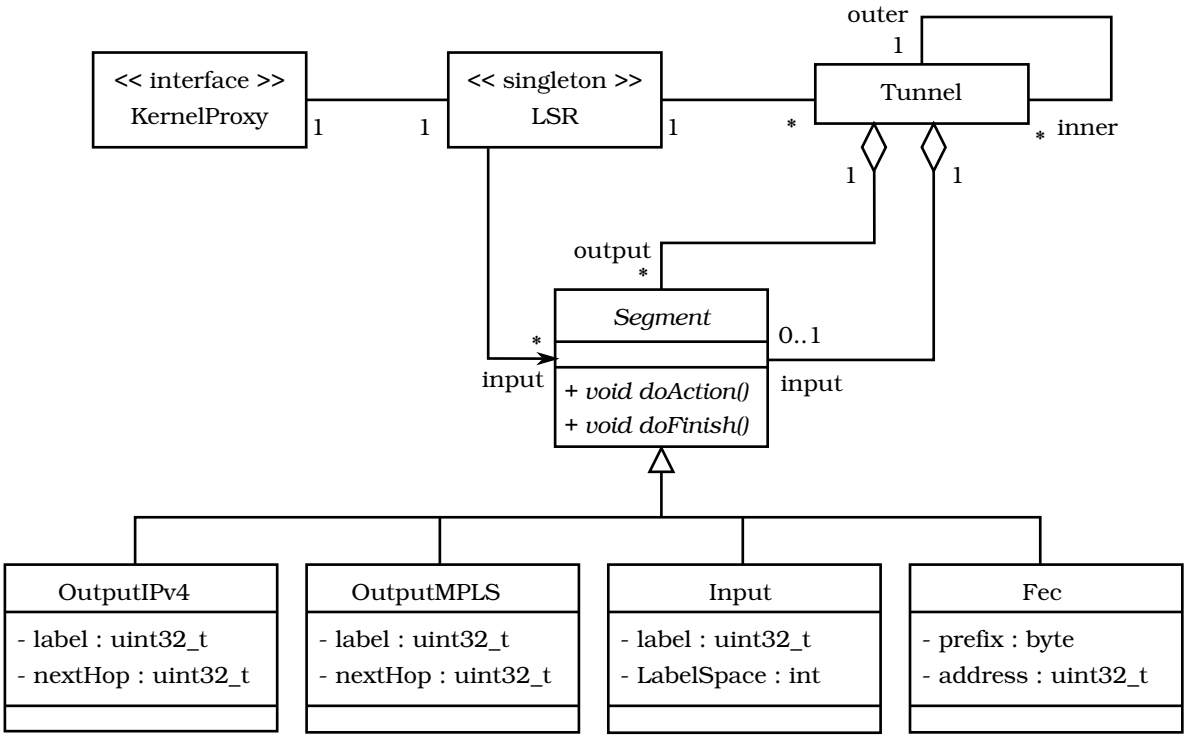


Figura 5.4: Diagrama de classes do módulo MPLS.

A classe LSR é um objeto único, representada pelo estereótipo <<singleton>>, o qual representa o *switch* MPLS. Ele é responsável em obter um pacote de rede, MPLS ou IPv4, e, de acordo com o seu protocolo, procurar o segmento de entrada que o representa, passando o processamento para ele. Essa associação não é bidirecional, pois o segmento de entrada não precisa saber que o LSR o está referenciando. Isso facilita a implementação. Ele é responsável também em guardar a lista de todos os túneis instalados no sistema. Obviamente, podem

haver vários túneis instalados no sistema, ou mesmo a total ausência deles.

A classe `Tunnel` representa os túneis que estão instalados neste LSR. Cada objeto `Tunnel` pode ter zero ou mais túneis internos, os quais permitem um nível de empilhamento de rótulos arbitrário. Cada objeto `Tunnel` possui também alguns parâmetros que o qualifica, ou seja, os segmentos que o compõem. Para os segmentos de entrada, pode-se ter apenas um único segmento, caso ele esteja descrevendo um LSP[1], ou não conterá segmentos de entrada, caso esteja descrevendo um túnel interno. Por outro lado, pode existir um ou mais segmentos de saída. Mais do que um segmento de saída implica em replicação de pacotes. É responsabilidade desse objeto realizar a interligação entre o segmento de entrada e os segmentos de saída e os parâmetros do segmento de saída de nível mais externo são os que devem ser utilizados para encaminhar o pacote para o próximo próximo *hop*.

A classe `Segment` é uma classe abstrata e representa os possíveis tipos de segmentos no sistema. Existem alguns métodos e alguns atributos que são comuns a todos os segmentos. Como exemplo, podemos citar: os atributos de coleta de estatísticas e as ações que devem ser efetuadas no recebimento do pacote e no seu envio. Essas ações são métodos virtuais, no sentido que devem ser polimórficos, ou seja, cada subclasse que instancia essa classe deve prover a sua própria implementação. As funções polimórficas são: `doAction` e `doFinish`.

A função `doAction` tem por objetivo realizar qualquer operação de checagem sobre o segmento e também realizar a coleta de estatística. A função `doFinish` tem por objetivo realizar as operações de finalização sobre o segmento e dependem do papel que o segmento está executando no LSP.

As classes `Input`, `FEC`, `OutputMPLS` e `OutputIPv4` são subclasses da classe `Segment` e representam os possíveis segmentos no sistema. Todas essas classes tem suas definições apoiadas na RFC 3031. A classe `FEC` é equivalente à tabela de FEC, a classe `Input` é equivalente à tabela de rótulos de entrada e as classes `OutputMPLS` e `OutputIPv4` são equivalentes à tabela NHLFE.

Para a classe `Input` a função `doFinish` deve remover as informações de empilhamento que estavam no pacote e realizar uma de duas ações possíveis, ou reaplicar a busca de rótulos, para uma nova extração do nível de empilhamento, ou processar a lista de túneis, dada pelo objeto `Tunnel`. O processamento da lista de túneis é um processo repetitivo, iniciado pelo objeto `Tunnel` que descreve o nível de empilhamento mais interno e finaliza com o objeto `Tunnel` do nível mais externo, em que, para cada nível, deve-se obter a lista de segmentos de saída e passar o controle para cada um deles, a fim de realizar o seu processamento.

Para a classe `FEC` a função `doFinish` deve procurar por uma entrada na tabela FEC que case com os dados do pacote. Se não encontrar, deve devolver o pacote para que ele continue o seu processamento pelo módulo IPv4. Se encontrar, deve realizar apenas a segunda parte

[1]Relembrando, um LSP é um túnel com um único empilhamento de rótulos.

descrita para a classe `Input`, ou seja, a de processar a lista de túneis, dada pelo objeto `Tunnel`.

Para a classe `OutputMPLS` a função `doFinish` deve criar, no pacote, o espaço necessário no final da camada de enlace para conter o cabeçalho SHIM, preenchê-lo com os dados pertinentes e realizar uma das possíveis finalizações: encaminhar o pacote para ser entregue na interface de saída, ou passar o controle para o segmento de saída que está em cascata. No fim desse encadeamento, o controle passará para o segmento de entrada, para que ele dê continuidade ao processo de replicação de pacotes. Para a classe `OutputIPv4` a função `doFinish` deve simplesmente entregar o pacote para ser processado e encaminhado pelo módulo IPv4, já implementado no sistema operacional Linux.

Por fim, a classe `KernelProxy` é uma interface que permite aos programas de nível de usuário configurar o sistema. Ela realiza essa função ao instalar os objetos `Tunnel` e configurar os segmentos associados, fornecendo-lhes a associação correta entre eles. Ela permite também, obter as estatísticas de utilização do sistema. Como era de se esperar é a classe `LSR` que expõe essa interface para os programas dos usuários.

É possível gerar combinações entre os segmentos que são inconsistentes, como por exemplo, conectar um segmento de entrada `FEC` com um segmento de saída `OutputIPv4`, ou conectar vários segmentos de saídas internos do tipo `OutputMPLS`, sendo o mais externo do tipo `OutputIpv4`. É responsabilidade da classe `LSR` verificar e evitar essas inconsistências. Essa operação é realizada no momento da configuração do sistema.

Note-se também, que a ligação entre a função `doAction` e `doFinish` e a sua real implementação é feita neste momento, ou seja, da configuração do sistema. A correta ligação, como descrita anteriormente, depende do papel que o segmento está realizando no momento da configuração do sistema.

Com a finalidade de ilustrar algumas interações entre os vários objetos que compõem o sistema, vamos mostrar o diagrama de sequência para o seguinte cenário: um pacote MPLS chega na entrada do módulo MPLS, considerando que o papel do LSR, para este LSP, é a de um roteador núcleo. A Figura 5.5 mostra este diagrama.

Conforme pode ser visto, o objeto `:LSR`, ao receber um pacote MPLS, procura na tabela de rótulos de entrada por uma entrada que corresponda ao rótulo que está codificado no cabeçalho SHIM. Ao encontrá-lo, obtém o segmento de entrada associado e executa duas funções virtuais a `doAction` e a `doFinish`. Supondo que existam dois níveis de empilhamento de rótulo e que, neste LSR, deva-se processar o túnel interno também, a função `doFinish` estará ligada a uma implementação que faça a extração deste túnel interno. Assim, ela irá realizar uma nova pesquisa na tabela de rótulo de entrada, por um novo segmento que case com o rótulo interno e chamar as funções `doAction` e `doFinish` deste novo segmento.

Em seguida, através do objeto `:Tunnel`, o segmento de entrada obtém uma lista dos segmentos de saída. Para cada segmento de saída, as funções `doAction` e a `doFinish` são

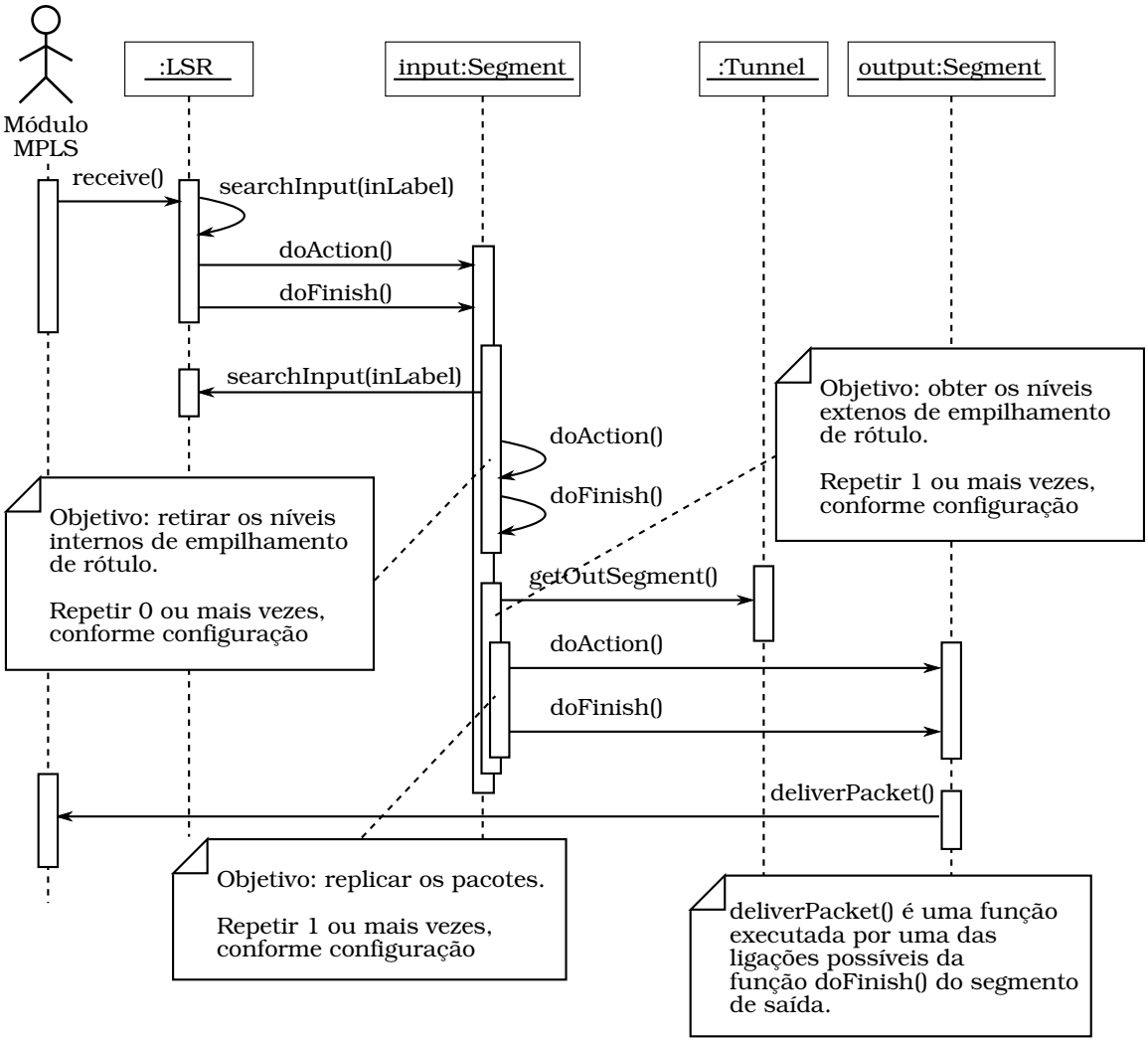


Figura 5.5: Diagrama de sequência, ilustrando os eventos realizados quando um pacote MPLS chega na entrada do módulo MPLS, considerando que o papel deste LSR, para este LSP, é a de um roteador núcleo.

executadas. Para o nível mais interno, a função `doFinish` está ligada a uma função que apenas insere a codificação SHIM no pacote. Em seguida, a mesma sequência é executada para o nível de empilhamento de um nível abaixo[2]. Neste caso, como este é o nível de empilhamento mais externo, a função `doFinish`, além de inserir a codificação SHIM no pacote, envia-o para a interface de saída, cujo `gateway` é o especificado por sua entrada na tabela NHLFE, representada pela função `deliverPacket`.

Existem algumas variações de cenários que empregam a mesma ideia básica, conforme delineada anteriormente. Pode-se analisá-los considerando o LSR atuando como um LER de ingresso ou como um LER de egresso. Para o LER de ingresso, o evento que inicia o processo é a chegada de um pacote IPv4, proveniente do módulo IPv4. Ao receber este pacote, o objeto `:LSR` realiza uma busca na tabela de FEC por uma entrada que case com os dados do pacote. Ao encontrá-la, obtém o segmento de entrada associado e executa as duas funções virtuais. Note-se que não exite túnel interno a ser processado pelo segmento de entrada, assim, a função `doFinish` deve consultar o objeto `:Tunnel`. A sequência de ações a serem executadas a partir desse instante é equivalente à descrita para o cenário em que o LSR está atuando como núcleo.

Para o cenário considerando o LER como egresso, a sequência de ações é a mesma descrita para o cenário em que o LSR está atuando como núcleo. A exceção ocorre com relação à ligação da função `doFinish` para o segmento de saída mais interno, se a ação a ser executada foi a entrega do pacote ao módulo IPv4, para que este realize o encaminhamento do pacote.

Para finalizar, note-se a ordem de ligação dos segmentos. Para os segmentos de entrada, a sequência de cascateamento deve ser do nível mais externo de empilhamento de rótulos em direção ao nível mais interno, enquanto que, para os segmentos de saída, a ordem deve ser a inversa, ou seja, do mais interno em direção ao mais externo.

5.1.2 Projeto do *Software*

O projeto do *software*, conforme já mencionado, consiste em adaptar a solução encontrada no passo anterior para o sistema operacional Linux. Isso pode ser realizado através das respostas a algumas perguntas. As principais são:

- Como será a inserção da solução no núcleo do sistema?
- Como será implementada a interface de comunicação entre o núcleo e os programas de usuários?
- Como será realizada a captura dos pacotes IPv4 para a pesquisa na tabela de FECs?

[2] O termo “abaixo” representa a ação em direção ao túnel mais externo.

- Como serão implementadas as associações descritas no diagrama de classes e as tabelas do sistema?

Obviamente existem outras perguntas que devem ser respondidas também, mas como são ligadas ao interfaceamento de "baixo nível" com o sistema operacional Linux, não valem a pena serem mencionadas aqui e, portanto, não serão discutidas.

Inserção da solução no núcleo do sistema

Há basicamente duas formas de se realizar a inserção da implementação MPLS para operar em conjunto com o núcleo do SO Linux. A primeira consiste em integrar a solução dentro da árvore principal de seu código fonte. Assim, ao se compilar o sistema operacional, a implementação MPLS seria compilada simultaneamente e estaria integrada a ele. A vantagem é poder usar, acessar e manipular qualquer estrutura de dados e utilizar qualquer função que seja necessária à implementação. A desvantagem é a necessidade de que, no momento da inicialização do sistema, o núcleo já esteja com o código MPLS compilado nele.

A segunda forma consiste em criar um módulo para a implementação MPLS. Um módulo equivale a uma biblioteca de carregamento dinâmico e permite que ele seja inserido no núcleo do SO em tempo de execução, sem a necessidade de ser compilado dentro da árvore principal do código fonte do SO. A vantagem é permitir que várias funcionalidades, inclusive as do MPLS, possam ser adicionadas em um sistema que já esteja em execução. A desvantagem é que somente as estruturas de dados e funções exportadas publicamente pelo núcleo podem ser utilizadas na programação do módulo.

Para esta implementação, empregou-se a forma de módulo, devido ao isolamento que ocorre entre o módulo e o núcleo do SO. Isso acarreta naturalmente em um baixo acoplamento entre eles, o qual permite que os sistemas evoluam distintamente entre si.

Interface de comunicação entre o núcleo e os programas de usuários

Há duas formas de se implementar a interface de comunicação entre o núcleo e os processos no espaço do usuário. A primeira consiste em utilizar a chamada de sistema `ioctl`, a qual permite inicializar ou obter as opções externas de um `socket`. A segunda consiste em utilizar o mecanismo de `socket` provido pela família `Netlink` [79, 80].

O uso da família `Netlink` é mais versátil e elegante do que o método da chamada de sistema `ioctl`. A desvantagem é a utilização de uma maior quantidade de código para se realizar esse interfaceamento. Por outro lado, o uso da chamada de sistema `ioctl` é mais simples de se utilizar, embora seja mais complexa para manipular os dados e pouco elegante.

Para o nosso sistema, apesar dos inconvenientes, a função `ioctl` foi a escolhida para implementar a interface de comunicação e configuração do sistema, devido às facilidades de uso

e implementação. Assim, foi criada uma nova família para o *socket* de rede (*network socket*)[3], chamada de `PF_MPLS`, ou `AF_MPLS`, com o intuito de se utilizar esta função. A vantagem da criação desta nova família é deixar que o próprio SO faça o isolamento e a correta demultiplexação entre as várias famílias já implementadas no SO e que usam essa função, com a implementação provida pelo módulo MPLS.

Captura dos pacotes IPv4 para a pesquisa na tabela de FECs

Há duas formas de se construir a tabela de FECs. A primeira delas é integrá-la com a tabela de rotas do protocolo IPv4. Assim, cada entrada na tabela de rotas pode ser considerada como uma FEC e os pacotes destinados a ela serem tratados pela implementação MPLS. Para isso, é necessário ter o acesso a esta tabela, a qual está disponível apenas quando se faz a implementação integrada com o código fonte do Linux.

Infelizmente, a escolha em se fazer a implementação através do uso da facilidade de módulo não nos permite o acesso direto a esta tabela e, portanto, não podemos utilizar essa solução.

A segunda alternativa é construir a tabela FEC isoladamente da tabela de rotas. Para isso, devemos capturar os pacotes IPs, retirando-os da cadeia de processamento usual realizado por este módulo, o IPv4, e procurar por entradas na tabela de FECs que casem com os dados dos pacotes. Se forem encontradas entradas que satisfaçam este casamento, deve-se excluir esses pacotes da cadeia de processamento IP, deixando-os serem tratados exclusivamente pela implementação MPLS. Caso não sejam encontradas entradas que satisfaçam este casamento, os pacotes devem ser devolvidos para a cadeia de processamento IP. Esta é a solução que está sendo utilizada para a implementação do MPLS em um módulo Linux.

Desde a versão 2.4.x do SO Linux, o *framework* `NetFilter` [81] está integrado na árvore principal do código fonte. Ele permite instalar pontos de captura de pacotes, chamados de *hooks*, dentro da cadeia de processamento de pacotes do módulo IP. Durante o processamento do pacote, quando um desses *hooks* é alcançado, o processamento é desviado para a função que foi instalada no momento da inicialização do *hook*. Essa função tem que decidir sobre o que fazer com o pacote. Das opções possíveis, duas nos interessam, que são: `NF_ACCEPT` e `NF_STOLEN`. A `NF_ACCEPT` diz que o pacote deve continuar sendo processado normalmente pelo módulo IPv4, enquanto que a `NF_STOLEN` diz que o pacote não mais retornará para a cadeia de processamento IPv4.

Existem cinco lugares possíveis de se instalar as funções *hooks*. Dentre as opções, somente dois lugares nos interessam: quando o roteador está encaminhando pacotes da interface de entrada, para a interface de saída, chamado de `NF_IP_FORWARD`, e quando os pacotes são gerados internamente, com destino a algum ponto na rede, chamado de `NF_IP_LOCAL_OUT`.

[3]Um *socket* de rede é a ponta final de um fluxo de comunicação bidirecional entre processos, em redes de computadores.

Por fim, foi considerado que uma FEC é dada pelo endereço de rede de um pacote IP, de acordo com a notação CIDR, e representa o destino de um pacote IP. Seguindo a ideia de que a FEC escolhida deve ser a que melhor representa o destino do pacote, a tabela FEC foi implementada utilizando-se um vetor de 33 posições, em que o indexador do vetor equivale à quantidade de bits utilizados como máscara, para o endereço IP de destino do pacote. Assim, cada entrada desse vetor aponta para uma lista encadeada linear das FECs instaladas no sistema e cujo índice é a máscara da rede.

O processo de pesquisa deve-se iniciar pelo índice de maior valor, que neste caso é o 32, analisar todas as FECs instaladas neste índice e, caso não encontre nenhuma entrada que case com o endereço de destino do pacote, realizar a mesma operação para os índices de menor valor, até o último, cujo valor é 0. Se achar uma entrada que case com os dados do pacote, a função *hook* deve retornar o valor `NF_STOLEN`, caso contrário, deve retornar o valor `NF_ACCEPT`.

Implementação das associações descritas no diagrama de classes

Conforme pode ser observado no diagrama de classes da Figura 5.4, existem várias associações que são do tipo "1 para várias". Para implementá-las, é necessário analisar cada ponta da associação separadamente. Para a classe que está associada com uma outra qualquer, cuja multiplicidade é 1, basta considerar que essa associação é representada por uma variável do tipo ponteiro.

Por outro lado, para a classe que está associada com uma outra qualquer, cuja multiplicidade é "várias", a forma mais fácil de implementá-las é utilizando contêineres conectados entre si, através de uma lista ligada linear e cujo conteúdo é uma variável do tipo ponteiro.

A exceção recai sobre a tabela de rótulos de entrada. Como é esperado que essa tabela possa ter várias entradas, o tempo de busca em uma lista linear é bastante elevado. A fim de reduzir esse tempo, a tabela de rótulos de entrada foi implementada utilizando-se de contêineres conectados entre si através de uma árvore binária do tipo RBT (*Red-Black Tree*)[4].

Esses dois tipos de listas já estão implementados no código fonte do Linux e suas APIs (*Application Programming Interface*) podem ser utilizadas por qualquer implementação que faça o uso do *framework* de módulos.

5.1.3 Implementação do Módulo MPLS

A implementação é a geração de código em uma linguagem de programação. Ela é o que realmente vai ser executada em um sistema computacional e não será detalhada nesta seção. Por outro lado, conforme já mencionado anteriormente, o código fonte do sistema operacional

[4]Uma árvore RBT é um tipo de árvore de pesquisa binária auto balanceável.

Linux é escrito em linguagem C, a qual é uma linguagem procedural. No entanto, o desenvolvimento do módulo MPLS empregou técnicas de análise e projeto voltados para o paradigma de orientação a objetos.

Assim, esta seção mostrará uma das possíveis alternativas de se mapear o paradigma de orientação a objetos na linguagem de programação C, conforme a referência [82]. O modelo de orientação a objeto estabelece alguns conceitos que são intrínsecos a este paradigma. Logo, a referência sugere a seguinte metodologia para mapeá-los em linguagem C:

1. Classe: transformar as declarações `class` em `structs`. Infelizmente perde-se a visibilidade nativa da programação OO, uma vez que todos os atributos passam a ser públicos.
2. Métodos: implementá-lo através de funções, em que o primeiro argumento deve ser uma referência para a estrutura que define os dados desta classe e, em seguida, a lista de argumentos do método em si.
3. Herança: quando uma classe deve herdar os atributos de uma classe mãe, basta declarar a classe mãe em primeiro lugar na definição da classe filha. Assim, a declaração de `struct` da classe filha deve conter, em sua primeira linha, a declaração de `struct` da classe mãe. Para processar a classe mãe, tendo apenas a classe filha, basta fazer um `cast` na classe filha para a classe mãe.
4. Polimorfismo: a linguagem C possui o conceito de ponteiro de função, o qual permite que uma função e a sua implementação possam ser associadas em tempo de execução do programa.
5. Encapsulamento: como o compilador não irá ajudar neste ponto, deve-se estabelecer uma metodologia de programação para amenizar essa deficiência, ou seja:
   - Evitar a utilização de variáveis globais;
   - Escrever os métodos de cada classe em arquivos separados;
   - Incluir estruturas que sejam realmente utilizadas;
   - Não acessar os campos diretamente.

Portanto, aplicando essa metodologia de programação, consegue-se uma implementação que se aproxima do modelo de programação orientado a objeto e, apesar do C não ser uma linguagem nativa desse paradigma, não impede que ela seja usada com este propósito.

5.2 Testes Efetuados

Os testes realizados na implementação do MPLS visam atender a dois objetivos bem específicos. O primeiro é verificar o correto funcionamento do módulo MPLS, não somente com

relação ao processo de codificação, mas também com a correta implementação da RFC 3031. O segundo é verificar a integração desta solução com a implementação da arquitetura MPA.

5.2.1 Testes Realizados com o Subsistema MPLS

Durante a fase de codificação, uma das melhores ferramentas disponíveis para testes e depuração, e isso vale para qualquer módulo escrito para ser executado no núcleo do sistema operacional, é o sistema de virtualização, integrado na árvore do código fonte do Linux, chamado de UM (*User Mode*).

Em tal modelo de virtualização, é criado um arquivo executável do SO Linux que pode ser executado como um processo do usuário. Observando-o pelo lado da aplicação, ele é uma máquina virtual completa, na qual o SO Linux está em execução, com todas as funcionalidades encontradas em uma distribuição Linux qualquer, excetuando-se o acesso direto ao *hardware*, uma vez que isto é feito pela máquina hospedeira. É possível, inclusive, interagí-lo com outras máquinas reais ou virtuais através do subsistema de acesso a rede de dados, utilizando-se das ferramentas de gerência normalmente encontradas nas distribuições Linux.

Observando-o pelo lado do usuário, ele é um processo como um outro qualquer, o qual nos permite paralisá-lo, depurá-lo e destruí-lo. Em particular, a capacidade de depurar esse processo, utilizando-se de ferramentas de depuração, como por exemplo, o `gdb`, permite-nos visualizar e analisar os estados das estruturas de dados internas da implementação, usando os mesmos procedimentos normalmente empregados na depuração de um programa qualquer.

Após a fase de codificação, existem alguns testes que podem ser efetuados com a finalidade de verificar, não somente as interações externas da implementação, mas também a sua estabilidade com o tempo. Para isso, foi montada uma rede simples, composta de três LSRs em cascata, conforme visualizada na Figura 5.6.

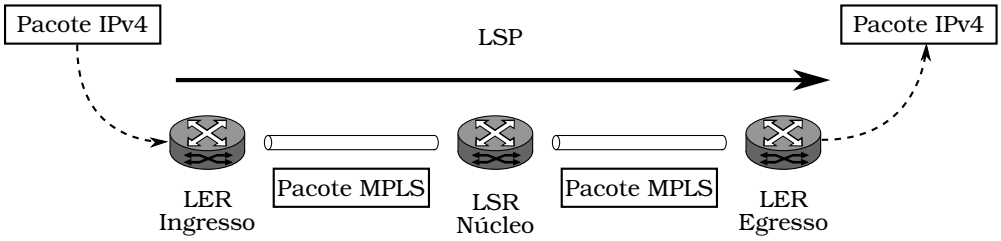


Figura 5.6: Topologia de rede trivial utilizada para os testes.

Sobre essa topologia, o programa `ping` foi utilizado com dois propósitos, o primeiro para verificar a correta construção do LSP, através da conectividade obtida entre o LER de ingresso e o LER de egresso para os pacotes ICMP. O segundo foi obter um comparativo de performance através do tempo médio de ida e volta, para os pacotes provenientes do LER de ingresso até o LER de egresso, utilizando os seguintes cenários: usando o subsistema MPLS com um

nível de empilhamento de rótulo, com dois níveis de empilhamento de rótulos, no qual o LSR de núcleo examina também o túnel interno, e somente com o uso do subsistema IPv4. A Tabela 5.1 mostra os resultados obtidos, considerando que os tempos médios foram obtidos sobre uma amostra de 200 pacotes, em um sistema com baixo tráfego de rede.

Tabela 5.1: Tempos medidos pelo programa "ping", para os tempos de ida e volta, utilizando tanto a implementação MPLS, quanto apenas o roteamento IPv4. D.P. significa desvio padrão.

Cenário	Tempo (μs)			
	min	médio	max	D.P.
Subsistema MPLS: um nível de empilhamento	375	488	1.126	126
Subsistema MPLS: dois nível de empilhamento	354	508	1.187	121
Somente subsistema IPv4	344	463	936	123

Conforme pode ser observado, a implementação MPLS tem um desempenho inferior ao roteamento IPv4 puro. Isso pode ser explicado por dois motivos:

1. A implementação IPv4 do Linux é extremamente otimizada e nela as decisões de roteamento, tomadas pelo sistema, residem em uma memória `cache` para que possam ser reaproveitadas nos roteamentos posteriores;

2. Devido ao fato da implementação MPLS ter utilizado o `framework` de módulo, no roteador de ingresso os pacotes devem ser processados, primeiramente, pela implementação IPv4 antes de se chegar ao módulo MPLS, para que ele realize a sua checagem com a tabela FEC. Isso acarreta um cascateamento entre os subsistemas IPv4 e o MPLS, aumentando o atraso. O mesmo fenômeno ocorre no roteador de egresso, quando o módulo MPLS primeiramente remove as informações de rótulo do pacote e, posteriormente, entrega-o à implementação do IPv4, para realizar o seu processamento.

Observe-se que o fato de se utilizar o duplo tunelamento aumenta, proporcionalmente, o atraso do sistema. Para averiguar os motivos deste incremento, foram obtidos os tempos médios gastos em cada LSR, considerando tanto um único nível de LSP, quanto um duplo tunelamento. Para essa medida foi considerado, apenas, o tempo gasto dentro do módulo MPLS, descartando os tempos gastos pelo subsistema IPv4. A Tabela 5.2 sumariza os resultados encontrados.

Observa-se que no LER de egresso, os tempos medidos são iguais para ambos os casos. Não há motivos para serem iguais, pois, para o segmento de entrada desse LER, a ação de desempilhar rotulos é serial por natureza e, portanto, o duplo tunelamento gasta um tempo maior de processamento. Deve-se notar que o sistema está com uma baixíssima quantidade de túneis instalados nesse teste, a qual, praticamente, não contribui na medição do tempo de atraso devido a seriação do sistema. Por outro lado, é esperado que a diferença entre

Tabela 5.2: Tempo médio de processamento do módulo MPLS, levando em consideração somente o tempo gasto pelo pacote dentro deste módulo.

Papel do LSR	Tipo do Túnel	Tempo Médio (μs)
Ingresso	Simples	72
	Duplo	78
Núcleo	Simples	49
	Duplo	62
Egresso	Simples	18
	Duplo	18

esses tempos sejam mínimas, pois não há segmentos de saída, do tipo MPLS, para serem processados. Outro motivo para a igualdade desses valores é devido aos erros de medida de tempo do próprio sistema operacional. Já no LER de ingresso e no LSR de núcleo os tempo são diferentes. Esse fato pode ser explicado pelos seguintes motivos:

1. No LER de ingresso, o processamento do segmento de entrada é o mesmo, não importando a quantidade de níveis de empilhamento de rótulos executados pelo subsistema. A diferença encontrada entre o túnel simples e o túnel duplo é no processamento dos segmentos de saída, vez que o túnel duplo possui um segmento a mais do que o túnel simples;

2. O processamento de um nível de túnel só pode ser realizado após o processamento do nível precedente, o que acarreta em uma seriação dos tempos;

3. A implementação do módulo MPLS não teve como objetivo principal a sua otimização. Por exemplo, no processamento do segmento de entrada, o segundo nível de empilhamento de rótulos deve passar pela mesma computação realizada pelo primeiro nível, não utilizando-se de atalhos para diminuir o atraso de processamento. Outro exemplo é a forma como a associação "1 para várias" foi implementada, para o objeto `Tunnel`, a qual requer alguns acessos à memória para contemplar o empilhamento de rótulos. Isso é mais visível no LSR de núcleo.

A segunda ferramenta utilizada, com o objetivo de testar a implementação, foi o programa `wireshark` [83], cujo propósito é capturar e examinar os pacotes que passam por uma ou mais interfaces de rede. Essa ferramenta foi usada com dois objetivos: verificar o correto posicionamento do cabeçalho SHIM e, por consequência, o correto preenchimento de seus campos, e verificar a correta resolução de endereços, através do protocolo ARP, para os próximos *hops* de cada segmento do LSP.

5.2.2 Teste de Integração do Subsistema MPLS com a Implementação da Arquitetura MPA

A segunda bateria de testes foi conduzida para verificar a integração do módulo MPLS com a implementação da arquitetura MPA. Para esta finalidade, foi utilizada a topologia de rede mostrada na Figura 5.7.

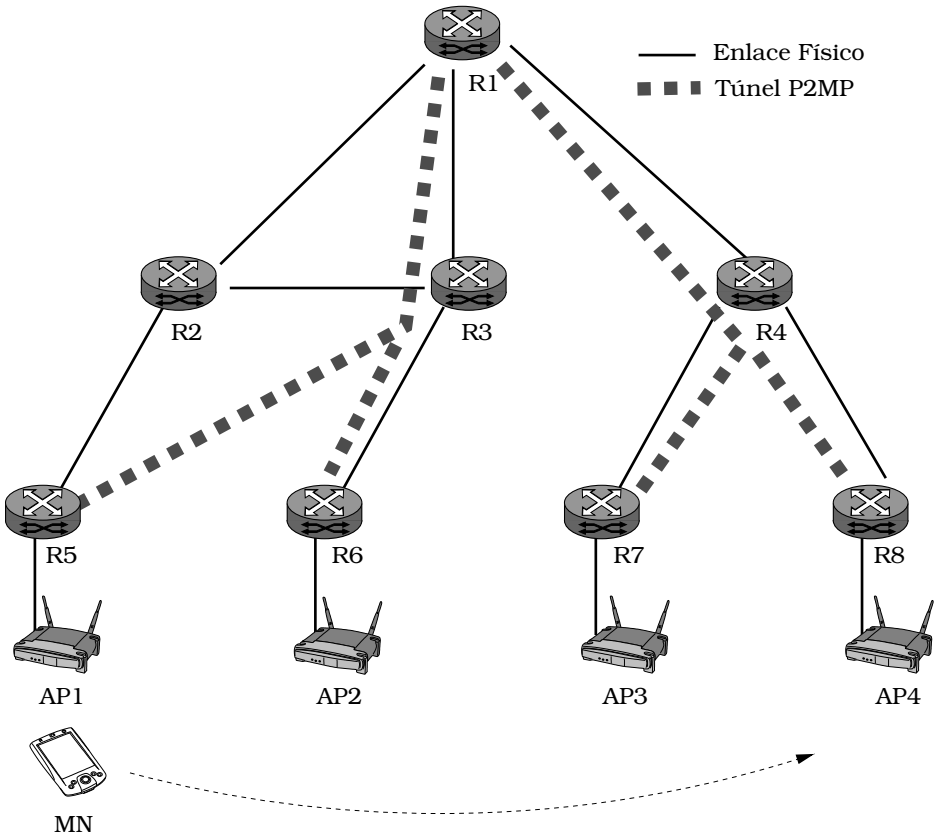


Figura 5.7: Topologia utilizada para testar a integração entre o subsistema MPLS e a implementação da arquitetura MPA.

A Tabela 5.3 mostra as métricas de medição obtidas quando ocorre um *handover* do ponto de acesso 1 (AP1) para o ponto de acesso 4 (AP4). Neste experimento, foram utilizado dois tráfegos do tipo CBR (*Constant Bitrate*), usando o protocolo UDP (*User Datagram Protocol*), cujo tamanho da carga foi de 340 *bytes* e cuja finalidade foi simular um tráfego multimídia. A taxa de transmissão foi de um pacote a cada 10 ms e a cada 20 ms, respectivamente. Note-se que esses são valores típicos utilizados em uma aplicação de áudio para a transmissão / recepção de pacotes. As métricas consideradas foram: intervalo de tempo entre o envio de pacotes; a perda de pacotes ocorridas entre os *handoffs*; e o atraso médio devido aos tempos de processamento das camadas de enlace e de rede e da implementação MPA.

Tabela 5.3: Métricas obtidas, considerando as sobrecargas das camadas 2 e 3 e também da implementação da arquitetura MPA, para um tráfego de *download*.

Intervalo de Tempo (ms)	Perda de Pacotes	Atraso Médio (ms)
10	124	1240
20	62	1240

Para este experimento, foram utilizados roteadores Linux, com o subsistema MPLS instalado e, no papel do MN, um *notebook* com um adaptador de rede sem fio do tipo Intel PRO/Wireless 2200BG, executando o sistema operacional Windows XP SP3. Os pontos de acesso utilizados foram do tipo Mikrotik RouterBoard 133 executando HostAPd sobre o *firmware* OpenWrt e configurados com o esquema de autenticação WPA-PSK + TKIP.

O objetivo dos testes foi medir os tempos de sobrecarga durante o *handover*. Esses tempos são compostos das sobrecargas das camadas 2 e 3 e pela implementação da arquitetura MPA. O tempo de sobrecarga da camada 2 é o necessário para realizar a associação nessa camada. Ele varia de 40 a 1500 ms e depende da placa de rede/*drivers* dos dispositivos instalados no MN e da direção do tráfego (*uplink*/*downlink*). A referência [84, 85] apresenta maiores detalhes sobre os motivos dessa variação.

O tempo de sobrecarga da camada 3 é o tempo necessário para configurar, através do DHCP, os parâmetros dessa camada, os quais são: prefixo de rede, endereço de rede e o roteador padrão. Esta sobrecarga varia, em nossa rede, de 15 a 25 ms.

O tempo de sobrecarga medido, da arquitetura MPA, foi em torno de 110 ms, gasto para atualizar um túnel interno em um MAR de ramificação. Esta sobrecarga inclui o processamento das mensagens do RSVP-TE e o redirecionamento do túnel interno, realizado pelo controlador MPLS. Esse tempo não depende das sobrecargas exibidas pelas camadas 2 e 3.

Por fim, os dados de temporização do subsistema MPLS foram utilizados para alimentar o modelo de filas de rede (QNM – *Queueing Network Model*), desenvolvido pela Doutora Thienne Johnson, para simular a vazão desta integração MPLS-MPA, ou seja, a quantidade de requisições de *handover* que o sistema consegue processar por segundo e o tempo de resposta, ou seja, o tempo gasto pelos pacotes dentro de tal sistema. Os resultados desta simulação podem ser verificados na referência [53] e um maior detalhamento sobre o modelo de simulação nas referências [52, 55].

5.3 Considerações do Capítulo

O capítulo discorreu sobre a implementação do protocolo MPLS, para ser executado no sistema operacional Linux. Com a finalidade de simplificar a implementação, foram adotadas algumas restrições no projeto, entre elas podemos citar: aplicada somente ao sistema ope-

racional Linux, considerado apenas a interface de rede do tipo *ethernet* e suportar apenas o protocolo de rede IPv4.

Assim, esse capítulo foi dividido em duas etapas, em que a primeira tratou da análise, do projeto e de sua implementação, enquanto a segunda tratou dos testes efetuados sobre este módulo, chamado também de subsistema MPLS.

Com relação à primeira etapa, foi empregado o paradigma de orientação a objeto, utilizando a linguagem UML para descrever os relacionamentos entre os vários objetos que compõem a solução. Na fase de análise, foram discriminados os atores externos a este módulo e, por consequência, foi definido também o escopo da aplicação. Ainda nesta fase, foram descobertos os objetos que compõem a solução e os seus relacionamentos.

Na fase de projeto foi mostrada como deve ser a integração da solução proposta na fase anterior com o código fonte do núcleo do Linux. Dentre as etapas realizadas, foi definido como deve ser o modo de inserção desta implementação com o SO Linux, em particular, utilizando-se do *framework* de módulo. Foi especificado como uma FEC foi definida nessa implementação, usando a notação CIDR e, na sequência, foi mostrado como capturar um pacote IPv4 para a procura de uma entrada, na tabela de FECs, que casasse com os dados do pacote. Foi descrito também como as associações entre os objetos foram implementadas.

Muitos detalhes técnicos da implementação foram omitidos, como por exemplo, a fragmentação de pacotes e a geração de mensagens ICMP do tipo *Destination Unreachable*, quando o bit DF está ligado, para contabilizar o MTU do caminho.

Finalizando esta fase, foi mostrado como o projeto, o qual foi desenvolvido pelo paradigma de orientação a objeto, pode ser implementado em uma linguagem orientada a procedimentos, ou seja, na linguagem C, utilizada para escrever o código fonte do núcleo do Linux.

Na segunda etapa foram apresentados os testes realizados sobre esse subsistema. Dois objetivos foram almejados: testar e validar esta implementação do protocolo MPLS e verificar a integração deste subsistema com a implementação da arquitetura MPA.

Várias foram as ferramentas utilizadas nesse processo, com objetivos distintos, para garantir o correto funcionamento deste módulo, conforme especificado pela RFC 3031.

Capítulo 6

Considerações Finais e Extensões ao Trabalho

Esta tese teve por objetivo prover uma solução de mobilidade para terminais IP que fosse transparente ao usuário final. Com a finalidade de contextualizar o ambiente, o cenário adotado foi o de uma rede Internet mantida por um único domínio administrativo que utiliza ou a versão atual do protocolo IP (IPv4), ou a versão futura (IPv6).

O foco em um único domínio administrativo se justifica por representar um ambiente controlado, onde pode-se definir, no núcleo da rede, quais os protocolos que devem ser executados e suas respectivas implementações. *A priori* é assumido que a rede permita a criação de túneis do tipo P2MP para uma tecnologia de tunelamento arbitrária. Dentre as possibilidades, escolhemos as mais utilizadas, que são os túneis IP/IP e os túneis MPLS. Em seguida é dada uma ênfase especial à tecnologia MPLS, mostrando como a nossa solução se integra em uma nuvem composta por elementos desta tecnologia.

Pelo lado do MN, foi assumido que ele necessita implementar apenas o básico da família de protocolos IP. Isso elimina a hipótese da introdução de produtos que adiram a nossa arquitetura, permitindo que um amplo rol de *gadgets* existentes, ou legados, possam usufruir da mobilidade oferecida pela arquitetura MPA.

Assim, podemos afirmar que existem duas contribuições oferecidas por esta tese na solução deste problema. A primeira delas foi a proposta de uma arquitetura que permita rastrear o MN dentro de um domínio administrativo, cuja solução é focada no núcleo da rede e é independente da tecnologia de tunelamento empregada. Ela foi idealizada por uma equipe de três doutorandos (incluindo o autor), orientados pelo Prof. Dr. Eleri Cardozo. Essa arquitetura é descrita no Capítulo 3.

Como principais atrativos desta arquitetura, podemos citar:

- é baseada em uma rede sobreposta – isola o tráfego de controle da arquitetura MPA do

tráfego de controle usualmente transportado pela rede IP;

- utiliza somente protocolos de rede bem estabelecidos pelos órgãos de padronização – evita o longo processo de especificação, maturação, implementação e estabilização de novos protocolos. Além disso, facilita a aceitação da arquitetura pelos administradores de redes, devido à confiabilidade dos protocolos já existentes;
- as funções de localização e rastreamento do MN são centradas no núcleo da rede – isso implica que não há restrições sobre os tipos de MNs que podem ser ancorados pela arquitetura MPA. A única exigência é a de possuir os protocolos básicos normalmente encontrados nesses tipos de dispositivos;
- é distribuída por concepção – a responsabilidade de localizar e rastrear os MNs estão espalhadas em vários elementos de redes, chamados de MARs. Isso aumenta a escalabilidade do sistema, uma vez que os MARs conhecem apenas informações parciais sobre os MNs; a disponibilidade, pois não possui um único ponto central de falha; e o desempenho, pois a carga de trabalho é divida entre vários MARs;
- mantém constante o endereço IP atribuído ao MN – enquanto o MN se mantiver dentro da rede MPA, o endereço atribuído a ele se manterá constante, não importando a quantidade de migrações que ele realize. A vantagem desse esquema é manter ativas as conexões de transporte, inclusive as conexões seguras (criptografadas);
- independe da tecnologia de tunelamento utilizada – pode ser utilizada em qualquer tecnologia de tunelamento, desde que ela satisfaça aos requisitos apresentados no Capítulo 3, pg. 46. Em particular, foi definido os mapeamentos para os túneis IP/IP e MPLS;
- interesse industrial – pode ser verificada através da patente registrada internacionalmente (WO/2008/147263).

A segunda contribuição foi uma proposta genérica de rastreamento do MN sobre o protocolo MPLS, para ser utilizada na arquitetura MPA. Essa proposta é descrita no Capítulo 4 e o modo como ela foi implementada e validada, tanto com relação à implementação em si, quanto à integração MPLS-MPA, no Capítulo 5.

As principais características desta solução são:

- baseada apenas nas especificações do protocolo MPLS, conforme RFC 3031 – isso evita quebrar a opacidade do protocolo MPLS, no qual é previsto que possa transportar qualquer tipo de protocolo. Assim, é garantido que pacotes criptografados podem ser transportados de forma transparente, sem quebrar as conexões. Por fim, facilita também a utilização desta solução nos LSRs já existentes, ao desobrigar a atualização de seus *firmwares* para incorporar funções não padronizadas;

- utilização de duplo tunelamento para rastrear o MN – o túnel externo é utilizado para indicar os possíveis egressos de um MN, dado um ponto de ingresso. O túnel interno permite rastrear e localizar o MN dentro de um domínio MPLS;
- pode ser utilizado tanto túneis P2MP, quanto túneis P2P que interligam os LSRs entre si – como a RFC 3031 permite a instanciação de túneis MPLS, os quais transportam outros túneis MPLS ou LSPs, a composição dos túneis externos pode ser feita através do uso de túneis P2MP, ou através desses túneis P2P, conectando os LSRs em um grafo acíclico (árvore);
- permitir a implantação gradual desta solução – nem todos os LSRs de uma nuvem MPLS necessitam abrir o túnel externo. Assim, nesses LSRs, em que essa abertura não é realizada, pode-se usar LSRs legados os quais processam apenas LSPs;
- permitir o uso dos *frameworks* de engenharia de tráfego, QoS e recuperação de falhas, já especificadas para o protocolo MPLS – como a nossa solução não quebra as especificações básicas do protocolo, as extensões adicionais ao protocolo, definidas pelo IETF, são facilmente incorporadas em nossa proposta;
- é escalável – o protocolo MPLS já possui uma alta escalabilidade e, uma vez que a nossa solução preserva o protocolo MPLS em sua totalidade, o túnel externo possui uma alta escalabilidade também. A preocupação recai sobre a necessidade de se manter, para o túnel interno, rótulos com uma abrangência global, que limita a quantidade de MNs em um domínio MPLS. Se esse espaço não for suficiente, é possível utilizar-se de mecanismos de agregação para aumentá-lo. Note-se que a quantidade de MNs que podem ser rastreados simultaneamente, em um domínio MPLS, é altamente dependente do tipo de cabeçalho utilizado para codificar os rótulos do sistema. O cabeçalho SHIM utiliza apenas 20 *bits*.

6.1 Trabalhos em Andamento e Futuros

Com relação à arquitetura MPA, existem alguns projetos que estão em progresso, cuja finalidade é explorar todas as suas potencialidades. Dentre esses trabalhos, podemos citar os seguintes tópicos, os quais são partes de teses de doutorado que estão em andamento:

- detalhamento do núcleo da arquitetura MPA, o seu mapeamento em túneis IP/IP e engenharia de tráfego;
- previsão de tráfego e sua aplicabilidade em *handover* pró-ativos, cujo objetivo é preparar a rede para o *handoff*, ao antecipar a próxima localização do MN em uma migração;

- aplicação de políticas de tráfego, cuja finalidade é dar tratamento especial para alguns tipos de pacotes em decorrência de políticas imposta na rede, como por exemplo, localização da fonte do tráfego e/ou horário de geração.

Em particular, já existem resultados publicados com relação à integração do *framework* de QoS com esta arquitetura, conforme pode ser visto no artigo intitulado *A Network Architecture for Mobile Robotics* [86].

Com relação a nossa solução de duplo tunelamento para rastrear no MN sobre o protocolo MPLS, temos algumas sugestões de trabalhos que podem ser realizados e que enriqueceriam a atual proposta. Dentre eles, podemos citar:

- criação de um protocolo de controle para ser executado sobre o protocolo MPLS – seria um protocolo que teria apenas os cabeçalhos do MPLS e os que estão abaixo dele e somente percorreriam os LSRs pertencentes a um dado LSP. Sob este protocolo poderiam ser aplicado as ideias do paradigma de agentes móveis, em que um agente de *software* percorreria este LSP agindo sobre os LSRs, conforme programado. Algumas aplicações seriam: determinação da quantidade de sobrecarga, em *bytes*, do cabeçalho MPLS em um caminho, coleta de estatísticas e avaliação de sobrecarga nos LSRs;

- simulação do impacto de se usar um duplo tunelamento em uma nuvem MPLS – desenvolver um modelo de simulação que permita avaliar qual é o impacto de se usar o duplo tunelamento, em várias condições de tráfego e cargas na rede;

- estender a implementação para que aceite FECs do protocolo IPv6 – atualizar a implementação para que possa ser utilizada também com a futura versão do protocolo IP;

- atualizar a implementação com o *framework* de QoS do Linux – atualizar a implementação para que possa fazer uso do já existente *framework* de QoS. Isso permitirá que o *framework* de QoS da arquitetura MPA possa ser mapeada nessa implementação do protocolo MPLS.

Bibliografia

[1] 3G: Tecnologias de celular. Teleco: Inteligência em Telecomunicações. http://www.teleco.com.br/3g_tecnologia.asp, acesso em 31/05/2010.

[2] What is a 4G phone? wiseGEEK. http://www.wisegeek.com/what-is-a-4g-phone.htm, acesso em 31/05/2010.

[3] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6) specification. RFC 2460, IETF, December 1998. http://tools.ietf.org/html/rfc2460, acesso em 31/05/2010.

[4] V. Fuller and T. Li. Classless inter-domain routing (CIDR): The internet address assignment and aggregation plan. RFC 4632, IETF, August 2006. http://tools.ietf.org/html/rfc4632, acesso em 07/03/2010.

[5] P. Srisuresh and M. Holdrege. IP network address translator (NAT) terminology and considerations. RFC 2663, IETF, August 1999. http://tools.ietf.org/html/rfc2663, acesso em 07/03/2010.

[6] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, The Internet Engineering Task Force (IETF), January 2001. http://tools.ietf.org/html/rfc3031, acesso em 23/10/2009.

[7] The Free Dictionary. http://www.thefreedictionary.com/mobility, acesso em 28/02/2010.

[8] Fabio M. Chiussi, Denis A. Khotimsky, and Santosh Krishnan. Mobility management in third-generation all-IP networks. *IEEE Communications Magazine*, pages 124–135, September 2002.

[9] Ian F. Akyildiz, Jiang Xie, and Shantidev Mohanty. A survey of mobility management in next-generation All-IP-based wireless systems. *IEEE Wireless Communications*, pages 16–28, August 2004.

[10] Ed. J. Manner and Ed. M. Kojo. Mobility related terminology. RFC 3753, IETF, June 2004. http://tools.ietf.org/html/rfc3753, acesso em 23/09/2009.

[11] Ed. J. Kempf. Problem statement for network-based localized mobility management (NETLMM). RFC 4830, IETF, April 2007. http://tools.ietf.org/html/rfc4830, acesso em 23/09/2009.

[12] D. A. Maltz and P. Bhagwat. MSOCKS: an architecture for transport layer mobility. In *Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1037–1045. IEEE, 1998.

[13] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. *SIGMOBILE Mobile Computing and Communications Review*, 4(3):47–57, 2000.

[14] A. Nasir and Mah-Rukh. Internet mobility using SIP and MIP. In *Proceedings of the Third International Conference on Information Technology: New Generations*, pages 334 – 339, 2006.

[15] Ed. C. Perkins. IP mobility support for IPv4. RFC 3344, IETF, August 2002. http://tools.ietf.org/html/rfc3344, acesso em 23/02/2009.

[16] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. RFC 3775, IETF, June 2004. http://tools.ietf.org/html/rfc3775, acesso em 15/10/2009.

[17] C. Perkins. IP encapsulation within IP. RFC 2003, IETF, October 1996. http://tools.ietf.org/html/rfc2003, acesso em 28/09/2009.

[18] C. Perkins. Minimal encapsulation within IP. RFC 2004, IETF, October 1996. http://tools.ietf.org/html/rfc2004, acesso em 28/09/2009.

[19] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. Generic routing encapsulation (GRE). RFC 2784, IETF, March 2000. http://tools.ietf.org/html/rfc2784, acesso em 28/09/2009.

[20] Ed. G. Montenegro. Reverse tunneling for mobile IP, revised. RFC 3024, IETF, January 2001. http://tools.ietf.org/html/rfc3024, acesso em 05/10/2009.

[21] S. Kent and K. Seo. Security architecture for the internet protocol. RFC 4301, IETF, December 2005. http://tools.ietf.org/html/rfc4301, acesso em 11/02/2010.

[22] S. Kent and R. Atkinson. Security architecture for the internet protocol. deprecated by rfc 4301. RFC 2401, IETF, November 1998. http://tools.ietf.org/html/rfc2401, acesso em 01/11/2009.

[23] Zhong Ren, Chen-Khong Tham, Chun-Choong Foo, and Chi-Chung Ko. Integration of mobile IP and multi-protocol label switching. In *Proceedings of the IEEE International Conference on Communications*, pages 2123–2127. IEEE, 2001.

[24] Tingzhou Yang and Dimitrios Makrakis. Hierarchical mobile MPLS: Supporting delay sensitive applications over wireless internet. In *Proceedings of the International Conferences on Info-tech and Info-net*, volume 2, pages 453–458. IEEE, 2001.

[25] F. A. Chiussi, D. A. Khotimsky, and S. Krishnan. A network architecture for MPLS-based micro-mobility. *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, 2:549–555 vol.2, March 2002.

[26] Rami Langar, Gwendal Le Grand, and Samir Tohme. Fast handoff process in micro mobile MPLS protocol for micro-mobility management in next generation networks. In *Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services*, pages 252–257, Washington, DC, USA, 2005. IEEE Computer Society.

[27] Rami Langar, Samir Tohme, and Gwendal Le Grand. Micro mobile MPLS: A new scheme for micro-mobility management in 3G All-IP networks. *ISCC*, 00:301–306, 2005.

[28] S. Fowler and S. Zeadally. Fast handover over micro-MPLS-based wireless networks. In *Proceedings of 11th IEEE Symposium on Computers and Communications*, pages 181–186. ISCC, IEEE, June 2006.

[29] A. G. Valko. Cellular IP: A New Approach to Internet Host Mobility. In *ACM Comp. Commun. Rev.*, January 1999.

[30] R. Ramjee, T. La Porta, S.Thuel, K. Varadhan, and S.Y. Wang. HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Network. *Proc. IEEE Intl. Conf. Network Protocols*, 1999.

[31] E. Fogelstroem, A. Jonsson, and C. Perkins. Mobile IPv4 regional registration. RFC 4857, IETF, June 2007. `http://tools.ietf.org/html/rfc4857`, acesso em 20/02/2010.

[32] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy mobile IPv6. RFC 5213, IETF, August 2008. `http://tools.ietf.org/html/rfc5213`, acesso em 23/02/2010.

[33] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier. Hierarchical mobile IPv6 (HMIPv6) mobility management. RFC 5380, IETF, October 2008. `http://tools.ietf.org/html/rfc5380`, acesso em 24/02/2010.

[34] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor discovery for IP version 6 (IPv6). RFC 4861, IETF, September 2007. `http://tools.ietf.org/html/rfc4861`, acesso em 23/02/2010.

[35] V. Devarapalli and F. Dupon. Mobile IPv6 operation with IKEv2 and the revised IPsec architecture. RFC 4877, IETF, April 2007. `http://tools.ietf.org/html/rfc4877`, acesso em 24/02/2010.

[36] R. Koodli. Mobile IPv6 fast handovers. RFC 5268, IETF, June 2008. `http://tools.ietf.org/html/rfc5268`, acesso em 23/02/2010.

[37] R. Koodli and C. Perkins. Mobile IPv4 fast handovers. RFC 4988, IETF, October 2007. `http://tools.ietf.org/html/rfc4988`, acesso em 23/02/2010.

[38] I. Samprakou, C. Bouras, and T. Karoubalis. Fast and efficient IP handover in IEEE 802.11 wireless LANs. In *Proceedings of the International Conference on Wireless Networks*, pages 249–255, 2004.

[39] IEEE recommended practice for multi-vendor access point interoperability via an inter-access point protocol across distribution systems supporting IEEE 802.11TM operation, March 2003.

[40] IEEE document P802.11/D6.1.97/5 wireless LAN, MAC and physical specifications, June 1997.

[41] I. Samprakou, C. Bouras, and T. Karoubalis. Improvements on IP IAPP: A fast IP handover protocol for IEEE 802.11 wireless and mobile clients. *Wireless Network*, page 497–510, April 2007.

[42] Approved minutes of the IEEE P802.11 full working group. IEEE P802.11 Working Group, 2005.

[43] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS protocol version 5. RFC 1928, IETF, March 1996. `http://tools.ietf.org/html/rfc1928`, acesso em 23/02/2010.

[44] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking*, pages 155–166. ACM/IEEE, 2000.

[45] A. C. Snoeren, H. Balakrishnan, and M. F. Kaashoek. The migrate approach to internet mobility. In *Proceedings of the Student Oxygen Workshop*, pages 14–17, 2002.

[46] J. Rosemberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, IETF, June 2002. `http://tools.ietf.org/html/rfc3261`, acesso em 24/02/2010.

[47] S. Salsano, A. Polidoro, C. Mingardi, S. Niccolini, and L. Veltri. SIP-based mobility management in next generation networks. *IEEE Wireless Communications*, 15(2):92–99, 2008.

[48] Ed. E. Mannie. Generalized multi-protocol label switching (GMPLS) architecture. RFC 3945, IETF, October 2004. `http://tools.ietf.org/html/rfc3945`, acesso em 08/03/2010.

[49] R. Moskowitz and P. Nikander. Host identity protocol (HIP) architecture. RFC 4423, IETF, May 2006. `http://tools.ietf.org/html/rfc4423`, acesso em 08/03/2010.

[50] Eduardo Zagari, Rodrigo Prado, Eleri Cardozo, Lars Westberg, Maurício Magalhães, Tomás Badan, José Carrilho, Rossano Pinto, André Berenguel, Daniel Barboza, Daniel Moraes, and Thienne Johnson. MPA: a Network-Centric architecture for Micro-Mobility support in IP and MPLS networks. In *Sixth Annual Conference on Communication Networks and Services Research*, Halifax, Nova Scotia, Canada, 5 2008.

[51] Rodrigo Prado, Eduardo Zagari, Tomás Badan, Eleri Cardozo, Maurício Magalhães, José Carrilho, Rossano Pinto, André Berenguel, Daniel Barboza, Daniel Moraes, Thienne Johnson, and Lars Westberg. A reference architecture for Micro-Mobility support in IP networks. In *13th IEEE Symposium on Computers and Communications*, Marrakech, Morocco, 7 2008.

[52] Thienne Johnson, Rodrigo Prado, Eduardo Zagari, Tomás Badan, Eleri Cardozo, and Lars Westberg. Considerations on performance evaluation of Micro-Mobility architectures for IP networks. In *IEEE PIMRC 2008 Mobile and Wireless Networks Track*, Cannes, France, 9 2008.

[53] Tomás Badan, Eduardo Nicola Ferraz Zagari, Rodrigo Prado, Eleri Cardozo, Maurício F. Magalhães, José Carrilho, Rossano Pinto, André Berenguel, Daniel Moraes, Thienne Johnson, and Lars Westberg. A network architecture for providing micro-mobility in MPLS/GMPLS networks. In *IEEE Wireless Communications and Networking Conference (WCNC 2009)*, Budapeste, Hungria, abril 2009.

[54] Eduardo Nicola Ferraz Zagari, Rodrigo Prado, Tomás Badan, Eleri Cardozo, Maurício F. Magalhães, José Carrilho, André Berenguel, Daniel Moraes, Tiago Dolphine, Thienne Johnson, and Lars Westberg. Design and implementation of a network-centric micro-mobility architecture. In *IEEE Wireless Communications and Networking Conference (WCNC 2009)*, Budapeste, Hungria, abril 2009.

[55] Thienne Johnson, Eduardo Zagari, Rodrigo Prado, Tomás Badan, Eleri Cardozo, and Lars Westberg. Performance Analysis of a New Architecture for Mobility Support IP Networks. In *IWCMC 2008 Mobile Computing Symposium*, Chania Crete Island, Greece, Greece.

[56] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, The Internet Engineering Task Force (IETF), March 1997. `http://tools.ietf.org/html/rfc2131`, acesso em 08/03/2010.

[57] R. Droms et al. Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315, The Internet Engineering Task Force (IETF), July 2003. `http://tools.ietf.org/html/rfc3315`, acesso em 08/03/2010.

[58] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP tunnels. RFC 3209, IETF, December 2001. `http://tools.ietf.org/html/rfc3209`, acesso em 13/03/2010.

[59] Ed. S. Yasukawa. Signaling requirements for point-to-multipoint traffic-engineered MPLS label switched paths (LSPs). RFC 4461, IETF, April 2006. `http://tools.ietf.org/html/rfc4461`, acesso em 13/03/2010.

[60] Ed. R. Aggarwal, Ed. D. Papadimitriou, and Ed. S. Yasukawa. Extensions to resource reservation protocol - traffic engineering (RSVP-TE) for point-to-multipoint TE label switched paths (LSPs). RFC 4875, IETF, May 2007. `http://tools.ietf.org/html/rfc4875`, acesso em 13/03/2010.

[61] S. Thomson, T. Narten, and T. Jinmei. IPv6 stateless address autoconfiguration. RFC 4862, IETF, September 2007. `http://tools.ietf.org/html/rfc4862`, acesso em 13/03/2010.

[62] R. Koodli. Fast handovers for mobile IPv6. RFC 4068, IETF, July 2005. `http://tools.ietf.org/html/rfc4068`, acesso em 13/03/2010.

[63] L. Berger et al. RSVP Refresh Overhead Reduction Extensions. RFC 2961, The Internet Engineering Task Force (IETF), April 2001. `http://tools.ietf.org/html/rfc2961`, acesso em 14/03/2010.

[64] T. Nadeau et al. Operations and Management (OAM) Requirements for Multi-Protocol Label Switched (MPLS) Networks. RFC 4377, The Internet Engineering Task Force (IETF), February 2006. `http://tools.ietf.org/html/rfc4377`, acesso em 14/03/2010.

[65] L. Berger. Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions. RFC 3473, The Internet

Engineering Task Force (IETF), January 2003. http://tools.ietf.org/html/rfc3473, acesso em 14/03/2010.

[66] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote authentication dial in user service (RADIUS). RFC 2865, IETF, June 2000. http://tools.ietf.org/html/rfc2865, acesso em 20/03/2010.

[67] Ed. F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen. Multi-protocol label switching (MPLS) support of differentiated services. RFC 3270, IETF, May 2002. http://tools.ietf.org/html/rfc3270, acesso em 27/03/2010.

[68] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS. RFC 2702, IETF, September 1999. http://tools.ietf.org/html/rfc2702, acesso em 27/03/2010.

[69] Ed. V. Sharma and Ed. F. Hellstrand. Framework for multi-protocol label switching (MPLS)-based recovery. RFC 3469, IETF, February 2003. http://tools.ietf.org/html/rfc3469, acesso em 27/03/2010.

[70] R. Aggarwal, Y. Rekhter, and E. Rosen. MPLS upstream label assignment and context-specific label space. RFC 5331, IETF, August 2008. http://tools.ietf.org/html/rfc5331, acesso em 18/08/2010.

[71] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta. MPLS label stack encoding. RFC 3032, IETF, January 2001. http://tools.ietf.org/html/rfc3032, acesso em 15/04/2010.

[72] Tomás Antônio Costa Badan, Rodrigo Chavez Monteiro do Prado, Eduardo Nicola Ferraz Zagari, Eleri Cardozo, and Maurício Ferreira Magalhães. Uma implementação de MPLS para redes linux. In *XIX Simpósio Brasileiro de Redes de Computadores*, pages 743–758, Florianópolis, SC, 2001. Anais do SBRC´2001.

[73] Eduardo Nicola Ferraz Zagari, Tomás Antônio Costa Badan, Rodrigo Chavez Monteiro do Prado, Eleri Cardozo, and Maurício Ferreria Magalhães. Uma plataforma para engenharia de tráfego com qualidade de serviço em redes MPLS. In *Simpósio Brasileiro de Redes de Computadores*, Búzios, RJ, 2002. Anais do XX Simpósio Brasileiro de Redes de Computadores - SBRC.

[74] Requirements analysis. Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Requirements_analysis, acesso em 11/05/2010.

[75] Richard C. Lee and Willian M. Tepfenhart. *UML and C++: A Practical Guide to Object-Oriented Development*. Prentice Hall, 2th edition, 2001.

[76] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGRAW-HILL International, 4th edition, 1997.

[77] Unified modeling language. OMG - Object Management Group. `http://www.uml.org/`, acesso em 22/05/2010.

[78] The linux kernel archives. Linux Kernel Organization, Inc. `http://www.kernel.org/`, acesso em 11/05/2010.

[79] Netlink. Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/wiki/Netlink`, acesso em 18/05/2010.

[80] Kernel korner - why and how to use netlink socket. Linux Journal. `http://www.linuxjournal.com/article/7356`, acesso em 18/05/2010.

[81] Linux netfilter hacking HOWTO. The netfilter.org project. `http://www.netfilter.org/documentation/HOWTO//netfilter-hacking-HOWTO.html`, acesso em 18/05/2010.

[82] Mapear objetos em C. Viva o Linux. `http://www.vivaolinux.com.br/artigo/Mapear-objetos-em-C`, acesso em 19/05/2010.

[83] Wireshark. Wireshark. `http://www.wireshark.org/`, acesso em 22/05/2010.

[84] Jon olov Vatn and Jon olov Vatn. An experimental study of IEEE 802.11b handover performance and its effect on voice traffic, 2003.

[85] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *SIGCOMM Comput. Commun. Rev*, 33:93–102, 2003.

[86] Paulo R. S. L. Coelho, Daniel H. Moraes, Eliane G. Guimarães, Eleri Cardozo, Thienne Johnson, and Fernanda C .A. Atizani. A network architecture for mobile robotics. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, volume CD-ROM, pages 1–6, Recife, PE, 2009. SBRC.