UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CÉSAR AUGUSTO HASS LOUREIRO

Estudo e classificação de propostas e protocolos para provimento de mobilidade sobre IPv6

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Profa. Dra. Liane M. Rockenbach Tarouco
Orientador(a)

Porto Alegre, fevereiro de 2012

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Loureiro, César Augusto Hass

Estudo e classificação de propostas e protocolos para provimento de mobilidade sobre IPv6 / César Augusto Hass Loureiro. – Porto Alegre: PPGC da UFRGS, 2012.

96 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2012. Orientador(a): Liane M. Rockenbach Tarouco.

1. FMIPv6. 2. IPv6. 3. HIP. 4. HMIPv6. 5. LISP. 6. MIPv6. 7. Mobilidade. 8. PMIPv6. 9. SHIM6. I. Tarouco, Liane M. Rockenbach. II. Título.

AGRADECIMENTOS

Agradeço a minha orientadora Dra. Liane Tarouco pela oportunidade, ao meu coordenador da bolsa Leandro Bertholdo pelos ensinamentos.

Agradeço a minha noiva Taís Konrath e aos meus amigos pela paciência.

Agradeço as minhas primas Carine e Márcia pelo apoio.

Agradeço a Deus, colegas e familiares não mencionados mas que colaboraram com minha formação pessoal e profissional, que de certa forma contribuiriam para que este trabalho se concretizasse.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

AAA	Authentication, Authorization and Accounting
AH	Authentication Header
AP	Access Point
AR	Access Router
ARP	Address Resolution Protocol
AS	Autonomous System
CGA	Cryptographically Generated Addresses
CN	Correspondent Node
CoA	Care of address
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
ESP	Encapsulation Security Payload Header
EID	Endpoint Identifier
ETR	Egress Tunnel Router
Fback	Fast Binding Acknowledgment
FBU	Fast Binding Update
FMIPv6	Fast Handover for Mobile IPv6
FN	Foreign Network
FNA	Fast Neighboor Advertisement
HA	Home Agent
HAck	Handover Acknowledgment
HBA	Hash-Based Addresses
HI	Handover Initiate
HIP	Host Identity Protocol
HMIPv6	Hierarchical Mobile IPv6

HN	Home Network
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ID	Identifier
IETF	Internet Engineering Task Force
ITR	Ingress Tunnel Router
LBU	Local Binding Update
LCoA	Local Care-of Address
LISP	Locator/Identifier Separation Protocol
LMA	Local Mobility Anchor
LOC	Location
MAC	Media Access Control
MAG	Mobile Access Gateway
MAP	Mobility Anchor Point
MIPv6	Mobile IPv6
MN	Mobile Node
MR	Map Resolver
MS	Map Server
MTU	Maximun Transmit Unit
NA	Neighbor Advertisement
NAR	New Access Router
NAT	Network Address Translation
NCoA	New CoA
ND	Neighbor Discovery
NS	Neighbor Solicitation
OSPFv3	Open Shortest Path First
PAR	Previous Access Router
PBA	Proxy Binging Acknowlegment
PBU	Proxy Binding Update
PCoA	Previous CoA
PMIPv6	Proxy Mobile IPv6
PrRtAdv	Proxy Router Advertisement
QoS	Quality of Services
RA	Router Advertisement

RCoA	Regional Care-of Address
RIP	Routing Information Protocol
RIPng	Routing Information Protocol next generation
RIR	Regional Internet Registry
RLOC	Routing Locator
RN	Remote Network
RR	Return Routability Procedure
RS	Router Solicitation
RVS	Rendez-vous Server
RtSolPr	Router Solicitation for Proxy Advertisement
SPI	Authentication Data e Security Parameter Index
SHIM6	Site Multihoming by IPv6 Intermediation
TTL	Time to Live
ULID	Upper Layer Identifier

LISTA DE FIGURAS

RESUMO

A iminente implantação do IPv6, pode ser uma solução para o crescente uso de dispositivos móveis, para ensejar a mobilidade e para solucionar problemas derivados do esgotamento de endereços IPv4. Contudo, para o provimento de mobilidade é necessário garantir conectividade ao usuário, permitindo uma utilização continuada de seus dispositivos quando em movimento, sem que ocorra a perda de conexão, de forma segura e transparente. Isto não é possível com a arquitetura TCP/IP atualmente implementada nas redes que estão operacionais, mesmo que utilizem o protocolo IPv6. Neste sentido, este trabalho visa estudar as propostas mais expressivas no provimento de mobilidade sobre IPv6, com vistas a evidenciar suas características e funcionalidades. Apresenta como resultado, uma análise desses protocolos, em especial no que tange ao tempo de troca de rede (*handover*) e facilidade de implementação.

Palavras-chave: FMIPv6, IPv6, HIP, HMIPv6, LISP, MIPv6, Mobilidade, PMIPv6, SHIM6.

ABSTRACT

The imminent deployment of IPv6, may be a solution to the growing use of mobile devices, to bring mobility and to solve problems arising from the exhaustion of IPv4 addresses. However, to provide mobility is necessary ensure connectivity for the user, allowing continued use of their devices while on the move, without loss of connection, securely and transparently. This is not possible with the TCP/IP architecture currently deployed in networks that are operating, even if used the IPv6 protocol. Thus, this work aims to study the most significant proposals in the provision of mobile IPv6, in order to contrast their features and functionalities. Presents as result, an analysis of these protocols, especially with respect to time network exchange (*handover*) and ease of implementation.

Keywords: FMIPv6, IPv6, HIP, HMIPv6, LISP, MIPv6, Mobility, PMIPv6, SHIM6.

1 INTRODUÇÃO

O protocolo amplamente usado na Internet, conhecido como protocolo IP (*Internet Protocol*), ou atualmente IPv4, possui aproximadamente quatro bilhões de endereços para uso.

Entretanto, a proliferação de dispositivos móveis sem fio, como *netbooks, tablets e SmartPhones*, associada ao esgotamento de endereços IPv4, permite considerar que este protocolo não poderá ser usado também para prover mobilidade. Este fato pode ser observado no esforço feito por operadoras de redes celulares, em preparação para o novo padrão "‘4G", que em um futuro não muito distante, provavelmente terão de lidar com dispositivos IPv6-only (LIMONCELLI; CERF, 2011) (MORR, 2010). Esta possibilidade não se deve somente ao fato da escassez de endereçamento IPv4, mas estimulada pelas funcionalidades implementadas no IPv6 e em seus protocolos de autoconfiguração de endereços (*Neighbor Discovery e Stateless Address Configuration*), que formam uma base de protocolos apropriada para redes móveis (PERKINS, 2002a).

Contudo, simplesmente prover endereçamento IPv6 válido aos dispositivos móveis, não é sinônimo de mobilidade. Para o provimento de mobilidade é necessário garantir conectividade ao usuário, permitindo a utilização de seus dispositivos em movimento sem que ocorra a perda de conexão, de forma segura e transparente. Devido a estas necessidades, novas propostas estão sendo concebidas, onde relacionam-se como mais expressivas as propostas: MIPv6 (*Mobile IPv6*), PMIPv6 (*Proxy Mobile IPv6*), HMIPv6 *(Hierarchical Mobile IPv6)*, FMIPv6 *(Fast Handover for Mobile IPv6)*, HIP *(Host Identity Protocol)*, LISP *(Locator/ID Separation Protocol)* e SHIM6 *(Level 3 Multihoming Shim Protocol for IPv6)*.

Neste contexto, este trabalho visa estudar estas propostas com vistas a evidenciar suas características e funcionalidades, obtendo como resultado, uma análise destes protocolos, em especial no que tange ao tempo de troca de rede (*handover*) e facilidade de implementação, abrindo caminho para o estudo de novas propostas que complementem e/ou melhorem as alternativas existentes.

Para tanto, este estudo está organizado da seguinte forma: no capítulo dois é apresentando uma panorama sobre os esforços de implementação do protocolo IPv6, seguido das novas funcionalidades do protocolo IPv6 em comparação ao IPv4, que facilitam o provimento de mobilidade. Nos capítulos três e quatro é apresentado o referencial teórico sobre os protocolos de provimento de mobilidade estudados, realizando sua classificação. No capítulo cinco, um visão geral sobre as implementações existentes dos referidos protocolos, seguido pelos trabalhos relacionados a esta pesquisa. No capítulo seis são

apresentados os experimentos realizados sobre as implementações dos protocolos de mobilidade, os quais são seguidos de uma análise comparativa sobre os resultados obtidos no capítulo sete. Por fim, no capítulo oito, encontra-se a conclusão e os trabalhos futuros.

2 O PROTOCOLO IPV6 E SUAS FUNCIONALIDADES PARA MOBILIDADE

Neste capítulo serão abordados os esforços realizados para implantação do protocolo IPv6, seguidos das novas funcionalidades existentes neste protocolo em comparação ao protocolo IPv4. Por fim serão comentadas as funcionalidades que ensejam a mobilidade sobre IPv6.

2.1 Panorama da implantação do IPv6

A justificativa para criação de um novo protocolo a fim de substituir o IPv4 foi a escassez de endereços IPs, pois a quantidade de 2^{32} endereços não é suficiente para a escala na qual a Internet cresce (ARANO, 2010). Com IPv6 é possível endereçar 2^{128} dispositivos, isto é, dois bilhões de IPs para cada pessoa no planeta[1].

Todavia, mesmo com a disponibilização do IPv6 em 1998 e programas de alerta e conscientização sobre o término dos endereços IPv4, o crescimento do IPv6 não teve a adoção esperadas pelos órgãos reguladores da Internet no mundo. Em setembro de 2008, o IANA disponibilizou um cronograma informando que após o término dos blocos /8 existentes, ocorrido em fevereiro de 2011, foram distribuídos os últimos blocos conhecidos como **reservados**, sendo um bloco para cada RIR (*Regional Internet Registry*) e dependendo da demanda existente em cada RIR, seu consumo deverá ocorrer entre um e três anos.

A demora na adoção do IPv6 ocorre principalmente pela necessidade de investimento em serviços e na substituição de equipamentos, principalmente nos *Backbones* das operadoras de telefonia e provedores de Internet. Para contornar esta necessidade eminente, fabricantes como Cisco implantaram o protocolos *Carrier Grade NAT44* (CISCO, 2009) em seus equipamentos, que permite o uso de NAT na borda da rede, capacitando um Sistema Autônomo a trabalhar inteiramente com endereços inválidos, o que permite postergar ainda mais a migração para o IPv6, preservando investimentos feitos com os equipamentos existentes e postergando a necessidade de mudanças nos equipamentos, mas ao mesmo tempo impedindo que usuários alcancem outras vantagens oriundas deste novo protocolo.

Neste contexto, os governos e órgãos responsáveis pela distribuição de endereços es-

[1] Calculo considerando 2^{64} endereços para dispositivos e 2^{64} endereços para redes e uma população mundial de 7 bilhões.

tão tomando algumas atitudes. Nos Estados Unidos, em setembro de 2010, o Escritório Executivo da Presidência elaborou um memorando enviado a todos os órgãos do Governo Federal informando o engajamento do governo em realizar a transição para o IPv6, iniciando os trabalhos pelo Departamento do Tesouro, em novembro de 2010, e se comprometendo em iniciar a implantação em mais de vinte locais, incluindo o Departamento de Defesa Americano (VIVEK, 2010). No Brasil o incentivo para implantação do IPv6 é realizado pelo Núcleo e Informação e Coordenação do Ponto BR (NIC.br), através de treinamentos e eventos, estimulando as instituições a solicitarem blocos de endereços IPv6.

2.2 Novas funcionalidades do IPv6 em comparação ao IPv4

Com o IPv6 foram realizadas mudanças não só no tamanho do endereçamento, mas na implementação de um novo protocolo, com um novo cabeçalho e novas funcionalidades conforme descritas a seguir, as quais são utilizadas no provimento de mobilidade sobre o IPv6.

2.2.1 Cabeçalho

O cabeçalho IPv6 ficou mais simples em relação ao IPv4, pois apesar do aumento em quatro vezes no seu tamanho de endereçamento, seu cabeçalho possui apenas o dobro de tamanho. Para isto alguns campos foram retirados e outros tiveram seus nomes alterados. A Tabela 2.1 demonstra os campos que tiverem seus nomes alterados.

Tabela 2.1: Diferenças do cabeçalho IPv4/IPv6.

IPv4	**IPv6**
Tipo de Serviço	Classe de Tráfego
Tamanho Total	Tamanho de Dados
Tempo de Vida (TTL)	Limite de Encaminhamento
Protocolo	Próximo Cabeçalho

Conforme demonstrado na Figura 2.1, o cabeçalho IPv6 ficou com a seguinte estrutura:

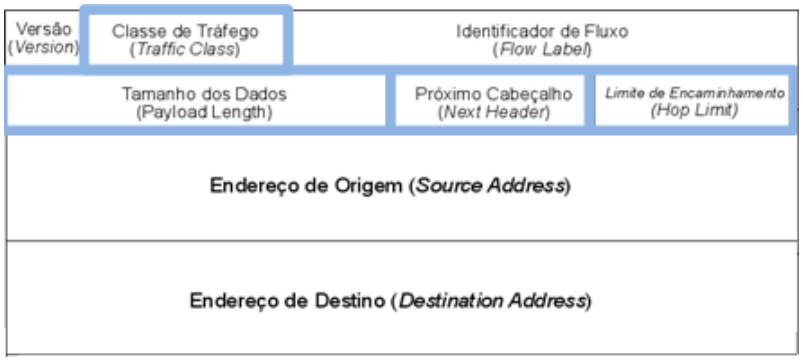

Figura 2.1: Cabeçalho IPv6

A grande alteração em comparação ao IPv4 no que diz respeito ao cabeçalho está no uso dos cabeçalhos de extensão. No IPv4 todas as informações do pacote estavam em

um cabeçalho de tamanho variável. No IPv6, havendo necessidade, múltiplos cabeçalhos de extensão podem ser incluídos em um único pacote, onde cada cabeçalho de extensão possui um campo próximo cabeçalho, que pode ser ou não processado pelos nós intermediários de uma rede. Segundo a RFC 2640 (DEERING; HINDEN, 1998), os cabeçalhos de extensão definidos são:

Hop-by-Hop - Informa aos roteadores a necessidade de analisar o restante dos cabeçalhos de extensão. Sem este cabeçalho, o pacote é encaminhado diretamente ao seu destino. Também utilizado para informar que o pacote trata-se de um *jumbogram* (acima de 64k octetos).

Destination Options - É utilizado em pacotes enviados por um nó móvel, enquanto estiver fora de sua rede, para informar ao destinatário seu *home address*.

Routing Header - Inicialmente utilizado para definir quais os saltos que o pacote deverá passar antes de chegar ao seu destino. Se tornou obsoleto por ser considerado um problema de segurança (ABLEY; SAVOLA; NEVILLE-NEIL, 2007).

Fragmentation - Informações sobre o pacote fragmentado, como a posição do fragmento atual em relação ao pacote original.

Authentication Header - Utilizado para a implementação de IPsec sobre IPv6, incluindo campos como *Authentication Data e Security Parameter Index* (SPI).

Encapsulation Security Payload Header (ESP) - É usado para garantir a confidencialidade, autenticidade da origem dos dados e integridade da conexão (KENT, 2005).

No ano de 2004, além das extensões de cabeçalhos definidas pela RFC2640 e suas alterações, foi regularizado um novo cabeçalho de extensão, o *Mobility Header*, que possui as mensagens: *Binding Refresh, Binding Update, Binding Acknowledgement, Binding Error Message*, as quais serão abordadas mais detalhamente no capítulo três (PERKINS; JOHNSON D.AND ARKKO, 2011).

Estes cabeçalhos, apesar de independentes, possuem uma ordem de uso conforme a Tabela 2.2, com isto os roteadores não necessitam ler todos os cabeçalhos de extensão para determinar quais deverão ser processados, eles podem processar até encontrar um cabeçalho endereçado ao destino e encaminhar o pacote. Isso facilita e diminui o processamento que precisa ser feito em cada roteador e contribui para reduzir a latência no trânsito dos pacotes, aumentando a performance dos roteadores, o que é especialmente importante no caso das redes de alta velocidade.

Tabela 2.2: Extensões do cabeçalho IPv6 e a sua ordem recomendada (DEERING; HINDEN, 1998) (CISCO, 2006).

Order	Header Type	Next Header Code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	135
	No next header	59
Upper Layer	TCP	6
Upper Layer	UDP	17
Upper Layer	ICMPv6	58

Além das alterações do cabeçalho, no protocolo IPv6 foram implementadas funcionalidades tais como: Descoberta de Vizinhança através da elevação da importância do *Internet Control Message Protocol* (ICMP), Descoberta Automática do *Maximun Transmit Unit* (MTU), alteração no protocolo de roteamento RIP para RIPng, OSPFv3, DHCPv6, DNS e outros. Dentre estas funcionalidades, serão abordadas nos próximos tópicos as que são necessárias para o entendimento da mobilidade sobre IPv6.

2.2.2 ICMPv6

O ICMPv6 tem fator crucial no funcionamento IPv6. Ele está localizado logo após o cabeçalho do IPv6 e suas extensões, possui apenas quatro campos (tipo, código, *checksum* e dados), mas sua importância está relacionada com os tipos de mensagens possíveis (RFC 4443) (CONTA; DEERING; GUPTA, 2006):

As **mensagens de erros** possuem as seguintes funcionalidades:

- *Destination Unreachable* - problemas em localizar destino;
- *Packet Too Big* - tamanho do pacote maior que o MTU;
- *Time Exceeded* - limite de encaminhamento;
- *Parameter Problem* - problema em algum campo dos cabeçalhos do IPv6.

Já as **mensagens de informações** contêm os conhecidos *Echo Request* e *Echo Reply* utilizados pelo comando *ping*.

Todas estas mensagens são definidas entre o tipo 1 e o tipo 129 do pacote ICMPv6, as diferenças estão nas mensagens entre os tipos 130 e 255 onde são definidos os tipos de mensagens para realização de ações como: Descoberta do Tamanho do MTU, Descoberta de Vizinhança (*Neighbor Discovery*), Gerência de Grupos *Multicast* e Mobilidade.

2.2.3 Neighbor Discovery

O protocolo *Neighbor Discovery* (ND) foi definido pela RFC4861 (NARTEN et al., 2007) o qual substitui o protocolo ARP e inclui as seguintes funcionalidades ao IPv6:

- Autoconfiguração de endereços;
- Localização de roteadores e vizinhos;
- Redirecionamento de pacotes;
- Descoberta de endereços duplicados.

2.2.3.1 Autoconfiguração de endereços

Ao contrário do IPv4, o IPv6 possibilita que um dispositivo gere automaticamente seus endereços através da configuração *Stateless* sem o uso de um serviço de DHCP, onde o *host* gera um IP com o prefixo FE80::/64 concatenado com seu *MAC Address*, chamado de endereço *link-local*, onde automaticamente passa a fazer parte dos grupos *multicast: solicited-node e all-node*. Posteriormente este *host* envia uma mensagem *Router Solicitation* (RS) para o grupo *multicast all-routers*, recebendo uma mensagem *Router Advertisement* (RA) do roteador padrão da rede, contendo as informações:

- MTU do enlace de rede;
- Rota *default*;
- Limite de encaminhamento;
- Prefixos da rede; e outras.

Através das informações recebidas, o *host* passa a ter um endereço *stateful* roteável na Internet. Outra forma de obtenção de endereços é através de um servidor DHCPv6, onde o *host* usando seu endereço *stateless* envia uma solicitação de endereço para o grupo *multicast* FF02::1:2, o qual é respondido por um servidor de DHCPv6, informações como servidor de DNS, NTP e outras. Estes processos também são utilizados quando um nó móvel entra em uma rede remota.

2.2.3.2 Descoberta de vizinhança

A mensagem *Neighbor Solicitation* (NS) é enviada do *host* a um grupo *multicast* da rede informando seu endereço MAC e obtém como resposta a mensagem *Neighbor Advertisement* (NA), contendo como parte da mensagem o endereço MAC de seus vizinhos. Estas mensagens substituem o protocolo ARP do IPv4 e inibem a necessidade de uso de *broadcast* na rede.

2.2.3.3 Redirecionamento de pacotes

Esta é outra funcionalidade do Descoberta de Vizinhança, onde roteadores que recebam pacotes de determinados *hosts*, possam enviar mensagens de *redirect* informando outro caminho de saída do enlace local.

2.2.3.4 *Endereços duplicados*

Após um *host* receber seu endereço, ele deve verificar se o mesmo já não está em uso na rede. Para isto, ele envia uma mensagem de *Neighbor Solicitation* informando no campo de destino o seu próprio endereço de origem, isto é, se ele receber uma resposta NA, significa que o endereço já está em uso. Este processo é sempre executado quando um nó móvel recebe um endereço ao entrar em uma rede estrangeira.

2.2.4 Endereçamento IPv6

Como referenciado anteriormente, o protocolo IPv6 utiliza 128 bits para formação de seu endereço, sendo estes números representados em hexadecimal divididos em 8 grupos de 16 bits separados pela pontuação ":", conforme a Figura 2.2 (HINDEN; DEERING, 2003).

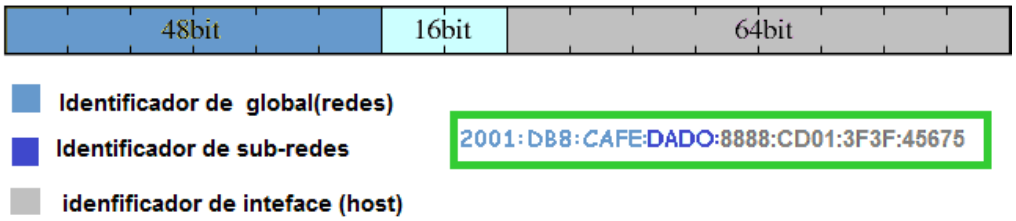

Figura 2.2: Endereçamento IPv6.

Devido ao seu tamanho, é permitido realizar uma abreviação dos endereços IPs para facilitar o seu uso. Caso ocorra uma sequência de zeros, ela pode ser substituída por "::". Por exemplo, o número IPv6 2001:0DB8:0000:0000:DADO:CAFE:FFDD:0051 pode ser representado pelo número 2001:DB8::DADO:CAFE:FFDD:51.

Além do formato de endereçamento, na arquitetura do IPv6 o *multicast* passou a ter papel fundamental no seu funcionamento, assim como o *Anycast* e o *Unicast* apresentados a seguir (HINDEN; DEERING, 2003).

2.2.4.1 *Unicast*

O Unicast no IPv6, assim como no IPv4, identifica o endereço de uma interface de rede de forma única e possui tipos (faixas reservadas) para certas funcionalidades. Por exemplo, no IPv4 utiliza-se o endereço de rede 192.168.0.0/16 para redes não roteáveis, no IPv6 este endereço é o FC00::/7. Esta mesma reserva de IPs existe para representar diferentes serviços, como:

- FE80::/10 endereço utilizado para a distribuição de IPv6 *stateless*;
- 2000::/3 faixa de endereços onde se iniciou a alocação dos endereços IPv6 roteáveis, isto é, a distribuição de endereços realizado pelo IANA utiliza inicialmente esta faixa;
- ::1/128 endereço de *loopback* (IPv4: 127.0.0.1).

2.2.4.2 Multicast

Utilizado para identificar um grupo de interfaces através de um endereço reservado FF00::/8, onde uma interface pode pertencer a mais de um grupo *multicast*. Os próximos oito bits 00 após o FF (FF**00**::/8) são utilizados para *flags* e delimitação da área de abrangência do grupo *multicast*, que pode variar da interface local até a rede externa. Sendo que a abrangência referente a rede externa é limitada pelo TTL (*Time to Live)* do pacote. Com base nesta abrangência, existem grupos *multicast* pré-definidos, tais como:

- FF01::1 Grupo *all-nodes*, referente a todas as interfaces do dispositivo;
- FF01::2 Grupo *all-routers*, referente a todos os roteadores do dispositivo;
- FF02::1 Grupo *all-nodes*, referente a todos os dispositivos do enlace da rede (link-local);
- FF02::2 Grupo *all-routers*, referente a todos os roteadores do enlace da rede (link-local);
- FF02::5 Roteadores OSPFv3;
- FF02::9 Roteadores RIPng; e outros.

2.2.4.3 Anycast

O *Anycast* é utilizado para identificar um grupo de interfaces como, por exemplo, uma subclasse de rede. Pode-se criar uma analogia ao *broadcast*, que também identificava um grupo de interfaces. A diferença é que no *broadcast* a mensagem é enviada para todas as interfaces e no *Anycast* a mesma é enviada apenas a uma interface escolhida por proximidade.

Este tipo de endereço é utilizado, por exemplo, no balanceamento de carga, na descoberta de serviços na rede (onde ocorrendo a primeira resposta considera-se satisfatório) e na mobilidade, no processo de descoberta do *Home Agent* (rede de origem do nó móvel).

2.3 Funcionalidades do IPv6 no ensejo da mobilidade

O processo de mobilidade inter-redes foi proposto por Perkins (2002) inicialmente para funcionar em IPv4 (PERKINS, 2002b) e foi chamado de *Mobile* IPv4.

No *Mobile* IPv4, a comunicação entre o nó móvel e o nó correspondente sempre passa pela rede de origem do nó móvel, existindo um roteamento triangular. No IPv6 é possível executar uma otimização de rota, criando um canal de comunicação direto entre o nó móvel e o nó correspondente, utilizando para isto o cabeçalho de extensão *Mobility*.

As mensagens de controle do protocolo de mobilidade em IPv4, necessárias para registrar e controlar o nó móvel, são enviadas em pacotes UDP, o que em IPv6 pode-se utilizar os cabeçalhos de extensão próprios para estas ações.

Uma das maiores vantagens da implementação de mobilidade em IPv6 é a possibilidade de um única interface de rede possuir *n* endereços, facilitando a troca e o retorno

do nó móvel entre as redes. Esta vantagem, aliada ao uso de endereços de grupos *multicast*, que permite a um nó móvel descobrir os agentes de mobilidade existentes na rede de forma dinâmica, capacitam o IPv6 a prover mobilidade de uma forma mais simples que o IPv4.

Em questões de segurança, o IPv6 possui uma implementação nativa de IPsec, facilitando o estabelecimento de uma comunicação segura inter-redes. Contudo, se a necessidade for aumentar o *throughput*, com o uso dos cabeçalhos de extensão do IPv6 é possível diminuir o *overhead* por não necessitar do tunelamento requerido pelo *Mobile* IPv4.

Por estas e outras vantagens do IPv6 em comparação ao IPv4, em 2004 foi criada a especificação de suporte a mobilidade em IPv6, através da RFC 6275 (PERKINS; JOHNSON D.AND ARKKO, 2011), que utiliza esses novos recursos para implementação de Mobilidade.

3 PROTOCOLOS DE MOBILIDADE IPV6

A utilização de acesso à Internet e serviços através de dispositivos móveis, demanda tecnologias que atendam a vários requisitos e entre eles a necessidade de velocidade, segurança e facilidade de implementação. Com isto, estudos estão sendo realizados para prover estas funcionalidades através de diferentes protocolos e implementações. Neste capítulo, será detalhado o funcionamento das propostas mais expressivas voltadas ao provimento de mobilidade sobre IPv6, as quais possuem ao menos uma implementação desenvolvida, um *Internet-Draft* proposto para o IETF e três citações às suas publicações.

3.1 *Mobile IPv6*

O *Mobile IPv6* (MIPv6), foi o primeiro protocolo de mobilidade sobre IPv6 concebido, permitindo que um nó móvel troque de rede preservando as características de acesso de sua rede de origem, sem a necessidade de agentes de mobilidade na rede estrangeira.

Para explicar o seu funcionamento é necessário apresentar alguns elementos que participam da solução (PERKINS; JOHNSON D.AND ARKKO, 2011):

Mobile Node (MN) - Refere-se ao nó móvel, que alterna de uma rede de origem a uma rede estrangeira, preservando a comunicação.

Home Network (HN) - Rede de origem do nó móvel.

Foreign Network (FN) - Rede remota onde se encontra MN após sair da sua HN.

Home Agent (HA) - Roteador da rede de origem responsável pela mobilidade.

Correspondent Node (CN) - Nodo externo a rede, que está realizando a comunicação com o nó móvel.

Care-of Address (CoA) - Endereço recebido pelo MN na rede remota.

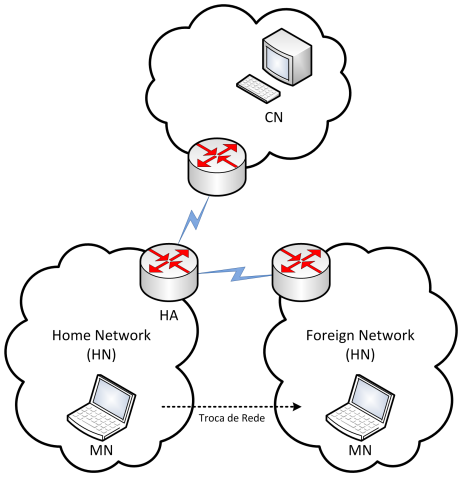

Figura 3.1: Agentes do MIPv6.

O processo de mobilidade ocorre quando um nó móvel (MN) (Figura 3.1) recebe um endereço na sua rede de origem, inicia uma comunicação com o nó correspondente (CN) e durante a comunicação realiza a troca da rede, isto é, movimenta-se para uma rede estrangeira (FN). Neste momento, ele recebe um novo IP através do sistema de configuração *stateful* ou *stateless*. Como MN continua com o seu endereço antigo, ele envia um pacote para seu roteador de origem (HA), registrando o seu novo endereço através da mensagem de *Binding Updates* e o HA responde com a mensagem *Binding Acknowlegement*. Este processo se repete toda a vez que o MN movimenta-se para uma nova rede, sendo que, no momento que o MN recebe uma mensagem de *Router Advertisement* (RA) de roteador contendo o prefixo da rede, ele compara com o endereço do seu HA. Se as informações forem as mesmas, MN sabe que voltou para sua rede de origem (LE; FU; HOGREFE, 2006).

Para o estabelecimento da comunicação entre o MN na rede estrangeira e o nó correspondente, pode-se trabalhar de duas maneiras: através de um tunelamento entre o MN e o HA ou diretamente do MN para CN, através de otimização de rotas. Na primeira forma, o CN não necessita saber que o *host* MN é um nó móvel, pois CN continua enviando seus pacotes para a rede de origem (HN) de MN, onde o HA fica responsável por encaminhar os pacotes para o MN através de um túnel bidirecional. Na segunda forma, CN precisa ter suporte à mobilidade, pois necessita conhecer a mudança de rede realizada por MN utilizando o cabeçalho de extensão *mobility* do IPv6. Este cabeçalho deve ser informado no campo *próximo cabeçalho* do pacote e possui o seguinte formato (PERKINS; JOHNSON D.AND ARKKO, 2011):

Payload proto - referente ao número do próximo cabeçalho, atualmente utilizado com o valor 59 para informar que não há próximos cabeçalhos;

Header len - tamanho do cabeçalho em múltiplos de 8 bytes;

MH Type - Tipos de mensagens;

Reserved - Reservado para uso futuro;

Checksum - Soma de verificação;

Message data - Dados do cabeçalho. Variável em tipo e tamanho de acordo com o campo *MH type*.

Os tipos de mensagens (MH Type) trocadas entre CN e MN durante o processo de negociação e estabelecimento da comunicação são:

Binding Update - Mensagem enviada pelo MN para o HA ou para o CN informando seu novo IP remoto (CoA);

Binding Acknowledgement - Confirmação de recebimento de uma mensagem de *Binding Update*;

Binding Refresh Request - Mensagem enviada pelo CN ao MN solicitando uma atualização de seus endereços atuais;

Binding Error - Utilizada pelo CN para informar a ocorrência de erros.

O processo de mobilidade completo pode ser observado na Figura 3.2.

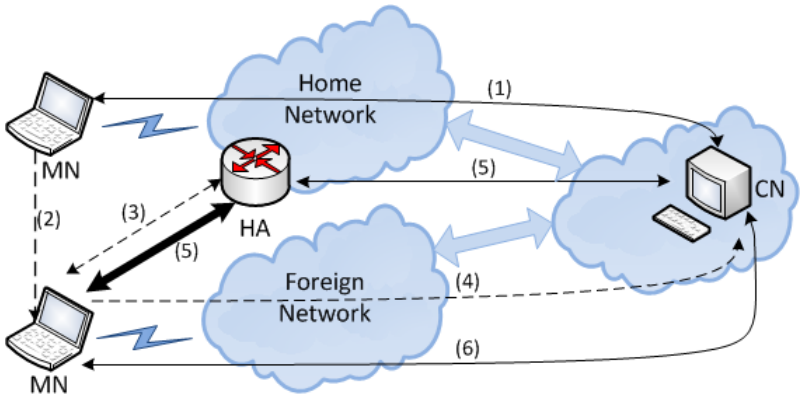

Figura 3.2: MIPv6, arquitetura e funcionamento (LE; LEI; FU, 2007).

(1) Existe uma comunicação normal entre o MN o CN (nó correspondente);

(2) MN movimenta-se para outra rede;

(3) MN registra seu novo CoA no seu HA;

(4) Ocorre o *Binding Update*(atualização das novas informações) entre o MN e o CN;

(5) A comunicação é realizada através de um túnel; ou

(6) A comunicação entre o CN e o MN é estabelecida diretamente.

3.2 *Fast Handover for Mobile IPv6*

O protocolo *Fast Handover for Mobile IPv6* (FMIPv6) foi padronizado pela RFC5268 em 2008 e atualizado pela RFC 5568 em 2009. Este protocolo tem o intuito de transformar o MIPv6 em um protocolo funcional, pois no MIPv6 existe uma latência muito alta durante o processo movimentação e registro do nó móvel na nova rede, chamado de tempo de *Handover*. Durante alguns segundos, o nó móvel fica incomunicável, isto é, não tem acesso à sua rede de origem e ainda não recebeu o IP na nova rede. Mesmo depois de possuir o novo IP, o nó móvel necessita aguardar a resposta referente ao seu *Binding Update* realizado na sua rede de origem. Com isto, neste capítulo serão abordadas as novas funcionalidades que tornaram o FMIPv6 mais eficiente que o MIPv6.

Neste protocolo, novos elementos fazem parte do processo de mobilidade (Figura 3.3), entre eles relacionam-se:

Access Point (AP): Dispositivo da camada dois que provê a conexão sem fio;

Access Router (AR): Roteador *default* do MN;

Previous Access Router (PAR): Roteador *default* do MN antes de realizar o *Handover*;

New Access Router (NAR): Roteador *default* do MN logo após realizar o *Handover*;

Previous CoA (PCoA): O endereço *Care-of Address* do MN na antiga rede;

New CoA (NCoA): O endereço *Care-of Address* do MN na nova rede.

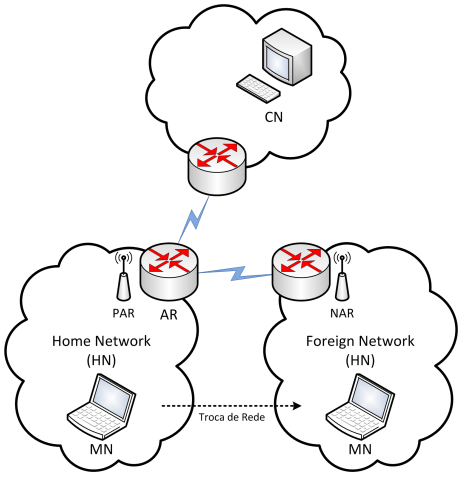

Figura 3.3: Agentes do FMIPv6.

3.2.1 Funcionamento no FMIPv6

O FMIPv6 aproveita informações da camada de enlace do modelo de referência OSI para sinalizar a troca de uma rede. Isto é, quando um dispositivo móvel reconhece que o sinal existente com o seu atual AP está enfraquecido e que existe um novo AP na área de cobertura, ele inicia o processo de conexão com esta nova rede, utilizando as seguintes mensagens introduzidas no FMIPv6:

Router Solicitation for Proxy Advertisement (RtSolPr);

Proxy Router Advertisement (PrRtAdv);

Fast Binding Update (FBU);

Fast Binding Acknowledgment (FBack);

Handover Initiate (HI);

Handover Acknowledgment (HAck) e

Fast Neighboor Advertisement (FNA).

Iniciado o processo de *Handover*, existem dois modos de operação possíveis: o modo preditivo e o reativo. A diferença está no momento que o nó móvel recebe a última mensagem do processo de *Handover*, antes ou depois de perder a conectividade com a sua rede atual.

3.2.1.1 *Modo Preditivo*

Neste modo, quando o MN realiza a negociação com o novo *Access Point*, ele envia ao seu AR uma mensagem RtSolPr, recebendo como retorno a mensagem PrRtAdv, inicia o processo de obtenção de endereço IP (*stateful* ou *stateless*) para configuração de um *New CoA*. Em posse de seu novo endereço, mas ainda se comunicando através de seu *Previous Access Router* (PAR), MN encaminha um *Fast binding Update* (FBU) a ele, solicitando que seu tráfego seja redirecionado através do *New Access Router* (NAR).

Imediatamente o seu PAR envia uma mensagem de *Handover Initiate* (HI) para o NAR informando o endereço *Previous CoA* (PCoA) e o endereço NCoA para validá-lo. Em resposta, o NAR envia ao PAR com uma mensagem HAck aceitando o endereço proposto inicialmente ou informando seu novo endereço válido. Terminada esta negociação, o PAR envia uma mensagem de *Fast Binding Acknowledgment* (FBack) em retorno a mensagem FBU anteriormente recebida e começa a encaminhar o tráfego para o NCoA. Neste momento, o nó móvel envia uma mensagem *Fast Neighboor Advertisement* (FNA) para o NAR informando sua presença na nova rede, permitindo que o tráfego encaminhado pela antiga rede ao NAR seja encaminhado ao MN. Por último, MN informa ao CN seu novo endereço para realizar uma possível otimização de rota, que permita uma comunicação direta sem a necessidade do tráfego passar pelo PAR (VIINIKAINEN et al., 2006). A troca de mensagens do processo está ilustrada na Figura 3.4.

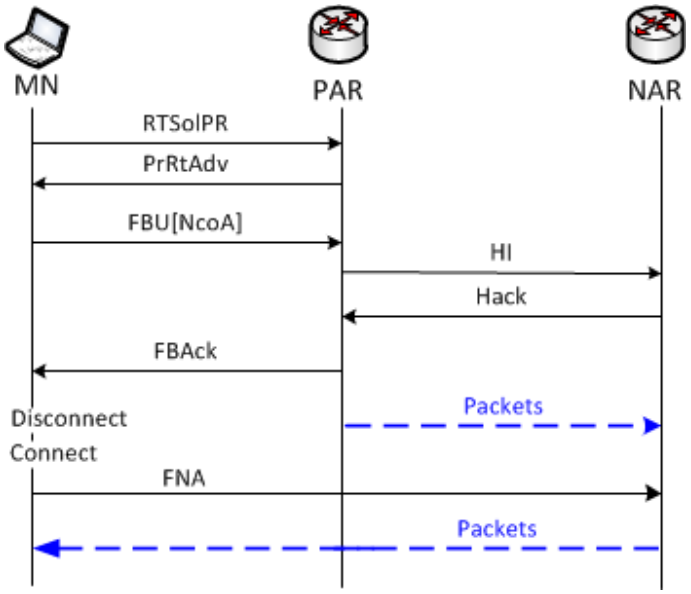

Figura 3.4: FMIPv6. Troca de mensagens no modo Preditivo.

3.2.1.2 *Modo Reativo*

No modo Reativo, o processo inicial é idêntico ao modo Preditivo até o momento do FBU. A diferença está na falta de comunicação devido à mobilidade do MN, onde a mensagem de resposta FBack, que deveria ser enviada do PAR ao MN não ocorre. Com isto, MN envia uma mensagem FBU para o NAR encapsulada em uma mensagem FNA, neste instante NAR verifica a validade do endereço de MN e encaminha uma mensagem de FBU para o PAR, recebendo um FBack de retorno. A partir desta ocorrência, o processo volta a ser igual, todas as mensagens endereçadas ao PAR são redirecionadas ao NAR e

encaminhadas ao MN no seu novo endereço IP (NCoA), como exposto na Figura 3.5.

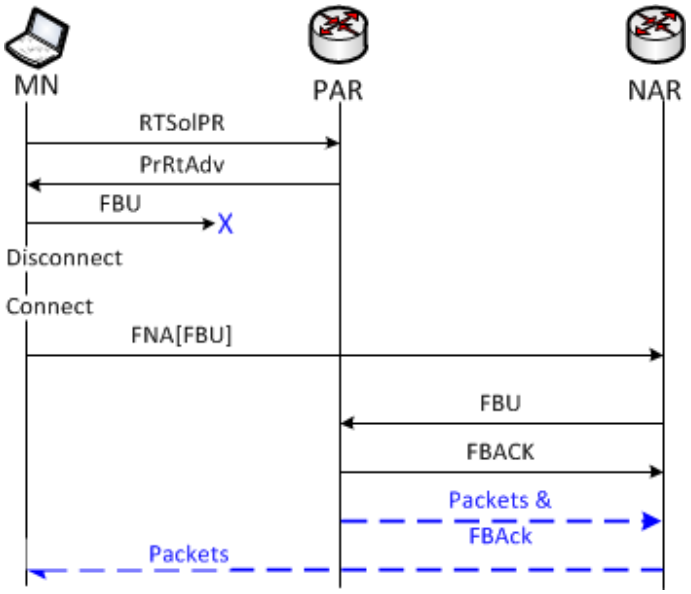

Figura 3.5: FMIPv6. Troca de mensagens no modo Reativo.

3.3 *Hierarchical Mobile IPv6*

No FMIPv6, a quantidade de sinalização existente no processo de *Handover* aumenta consideravelmente a complexidade do processo em comparação ao MIPv6. Para reduzir estas sinalizações foi desenvolvido o *Hierarchical MIPv6* (HMIPv6), que para atingir seu objetivo, incluiu mais um agente no processo, chamado de *Mobility Anchor Point* (MAP) (WANG; LI; YAN, 2009). Esse agente é responsável pelo controle da mobilidade existente no domínio da rede, isto é, possui o mesmo MAP para todo o *Autonomous System* (AS), independente do números de redes existentes. Com isto, passam a existir dois tipos de *Handover*: o local dentro do mesmo domínio e o externo quando ocorre a troca de domínios de rede.

No HMIPv6, o MN possui dois endereços de rede um *Regional Care-of Address* (RCoA) e um *Local Care-of Address* (LCoA). Quando MN conecta-se a uma rede, recebe uma mensagem de *Router Advertisement* contendo os endereços de um ou mais MAPs locais. Ocorrendo uma movimentação de MN dentro do mesmo domínio de rede, MN registra-se no novo *Access Point*, recebe um LCoA o qual deve ser informado ao MAP, mas permanece com o mesmo RCoA, que é utilizado para se comunicar com o CN.

Como mostrado na Figura 3.6, quando ocorre uma troca de domínio é necessário trocar os dois endereços. Para isto o MN envia um *Router Solicitation* (RS) para seu novo AR e recebe um *Router Advertisement* (RA), contendo os endereços dos MAPs locais. A seguir, o MN envia um *Local Binding Update* (LBU) para o novo MAP e um BU para seu *Home Agent* (HA), também conhecido por PAR, informando seu novo endereço LCoA e RCoA. Assim o tráfego endereçado para sua rede de origem é reencaminhado para o MAP, que encapsula as mensagens para o endereço LCoA do MN. Isto é necessário até que MN informe ao CN seu novo endereço e realize uma comunicação direta entre MN e CN, através do processo de otimização de rota.

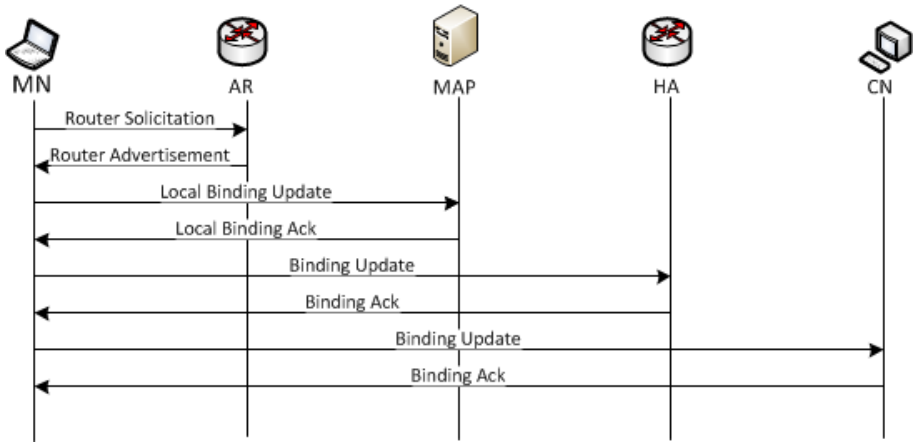

Figura 3.6: Troca de mensagens entre agentes no protocolo HMIPv6 (WANG; LI; YAN, 2009).

3.4 *Proxy Mobile IPv6*

O *Proxy Mobile IPv6* (PMIPv6) tem o intuito de incluir um ponto central no controle da mobilidade. Com isso MN não precisa realizar os controles de entradas e saídas de rede, esta responsabilidade passa a ser exercida por duas novas entidades: o *Mobile Access Gateway* (MAG), que está na rede pelo qual o MN está entrando e pelo *Local Mobility Anchor* (LMA), que se encontra na sua rede de origem. A Figura 3.7 ilustra o processo de troca de mensagens.

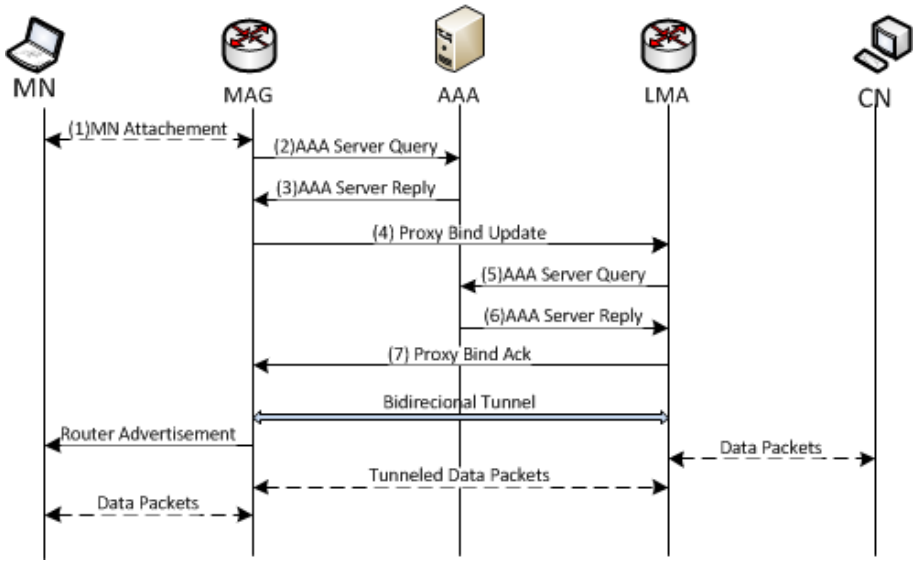

Figura 3.7: Processo de estabelecimento de conexão no PMIPv6 (KONG; LEE, 2008).

1 e 2) Quando um MN acessa a rede remota que possui um MAG, é realizado o procedimento de autenticação.

3) Após a autenticação o MAG obtém o perfil do MN, que contém: o *MN-identifier*, o endereço LMA e o modo de configuração suportado pelo MN, obtido a partir das políticas de segurança armazenadas em um AAA *Server* (*Authentication, Authorization and*

Accounting Server).

4) Em seguida o MAG envia um *Proxy Binding Update* (PBU) ao LMA, em nome do MN.

5 e 6) Uma vez que a LMA recebe a mensagem do PBU, ele verifica as políticas de segurança para assegurar se o remetente está autorizado a enviar o PBU. Se estiver autorizado, ele aceita a mensagem de PBU.

7) Por último, o LMA envia uma mensagem *Proxy Binging Acknowlegment* (PBA), incluindo o prefixo da rede do MN e atualiza a rota para a rede do MN sobre um túnel até o MAG.

Depois de realizado o túnel, MAG envia uma mensagem de *Router Advertisement* ao MN com suas configurações de rede, configurações estas pertencentes à nova rede local do MN. A partir deste ponto, o envio e recebimento de mensagens do MN será desempenhado pelo MAG e passará por um túnel até o LMA como se o MN estivesse na rede local, este túnel será utilizado por todos os MNs da *Home Network* que estiverem na *Foreign Network*, sendo desfeito quando não existirem mais MNs na rede estrangeira.

Como vantagens relativas ao PMIPv6 (KONG; LEE, 2008), relaciona-se:

- Não há necessidade de realizar qualquer tipo de configuração especial nos MNs;
- Bom funcionamento através de *wireless*, já que não é necessário o uso de túneis na rede local, que diminui o tamanho dos pacotes;
- Menor tempo para a entrega dos pacotes entre as redes.

Com a centralização dos controles de mobilidade nos MAGs e LMAs, o tráfego da rede relativo a MN pode ser mais facilmente controlado e gerenciado. Pois utilizando MIPv6 seria necessário identificar cada um dos MN para, por exemplo, conseguir estatística de uso de mobilidade.

4 PROTOCOLOS DE SEPARAÇÃO ENTRE A LOCALIZAÇÃO E IDENTIFICAÇÃO DE UM NODO

Os protocolos de mobilidade sobre IPv6 apresentados até o momento, realizam o processo de renovação de endereços na ocorrência de troca de rede, não sendo possível eliminar a latência de troca de endereços existente no processo de *Handover*. Com isto, uma nova abordagem está sendo estudada, a qual propõe a separação entre a identificação (ID) e a localização (LOC) de um nó na rede. Essa abordagem poderia ser comparada ao serviço de DNS, onde existe um mapeamento entre o nome e o endereço IP de um nó. Contudo, a separação entre localização e identificação propõe que este mapeamento ocorra de forma dinâmica durante a movimentação do nó, o que não ocorre com o DNS.

Os esforços nesta nova abordagem estão centrados nos protocolos: LISP (*Locator / Identifier Separation Protocol*), HIP (*Host Identity Protocol*) e SHIM6 (*Site Multihoming by IPv6 Intermediation*) como possíveis protocolos de provimento de mobilidade, os quais serão abordados a seguir.

4.1 *Locator/Identifier Separation Protocol*

O LISP possibilita a existência de redes *multihoming*, pois nodos utilizando LISP podem ser identificados sempre pelo mesmo endereço, independente de sua localização atual, sem a utilização de agentes de controle internos como HA, LMA ou a inclusão de novos cabeçalhos de extensão no IPv6.

Para isto, o LISP separa o endereçamento IP em duas partes, conhecidos por:

- *Endpoint Identifier* (EID): Referente ao endereço IP permanente de identificação de um nó na rede, sendo que este endereço não necessita ser um endereço roteável.
- *Routing Locator* (RLOC): Endereço IP roteável atribuído aos roteadores de borda da rede.

Basicamente, o LISP funciona encapsulando os pacotes entre dois endereços EID através dos roteadores de borda de rede, chamados de *Ingress Tunnel Router* (ITR) e *Egress Tunnel Router* (ETR). Estes dispositivos são responsáveis por armazenar o mapeamento de endereços entre EID e RLOC. Por exemplo, quando um ponto interno da rede necessita se comunicar com um site remoto, ele realiza uma consulta DNS que retorna o endereço EID do destino, com isto, o pacote é enviado até um ITR na borda da rede através de um

protocolo IGP *(Interior Gateway Protocols)*, que encapsula seu pacote em um novo pacote LISP, contendo o endereço RLOC de origem e o endereço RLOC do destino (Figura 4.1). Esse pacote chegando ao ETR de destino é desencapsulado e encaminhado ao EID (FARINACCI et al., 2011).

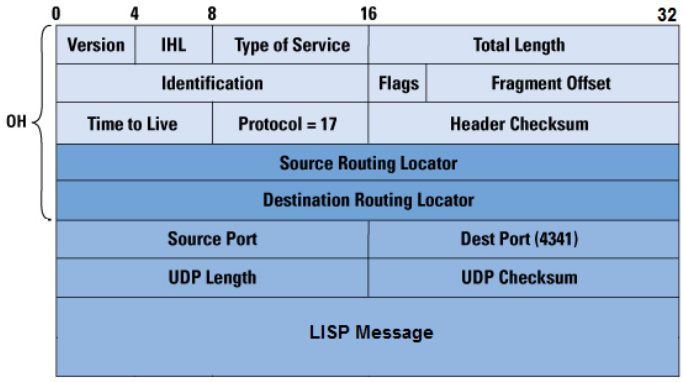

Figura 4.1: Formato do pacote LISP.

Com este protocolo é possível diminuir o tamanho das tabelas BGP (*Border Gateway Protocol*), pois os Sistemas Autônomos necessitam apenas conhecer os endereços RLOCs e não todos os blocos de endereços IP anunciados. Mas, para descobrir qual o RLOC de destino de um endereço possuindo apenas o seu EID, são necessárias mensagens de descobertas de endereços, *MAP-Requests* e *MAP-Replies*, que consistem em requisições enviadas aos ETRs para descobrir qual deles possui o endereço EID de destino. Após descoberto o endereço RLOC, o ITR mantém esta informação em cache para facilitar novas requisições, até a ocorrência de uma movimentação de dispositivos, que encadeia a necessidade de um novo mapeamento de endereços EIDs e RLOCs.

No intuito de agilizar este processo, FULLER at al. (2011) propuseram uma **Topologia Alternativa**, o LISP+ALT, que consiste em uma topologia lógica interligando os roteadores da Internet através de túneis GRE (*Generic Routing Encapsulation*), onde o protocolo BGP executado sobre estes túneis seriam responsáveis pelo controle dos endereços EIDs, os quais são distribuídos de forma ordenada, diminuindo assim o tempo de descoberta do mapeamento entre EID-RLOC. Outra proposta inclui na estrutura da rede dois novos componentes, os *Map Servers*, que aprendem o mapeamento entre EIDs e RLOCs, através de mensagens recebidas dos ETRs e os *Map Resolvers*, que respondem as consultas LISP realizadas pelos ITRs com o endereço RLOC dos EIDs ou com um *negative MAP-Reply* para endereços não EID (FULLER; FARINACCI, 2010).

Na ocorrência de comunicação entre redes LISP e redes *Non-LISP*, não é possível encaminhar o pacote LISP diretamente para a rede de destino, pois o endereço EID de origem não é roteável e seria descartado pelos roteadores na Internet. Com isto, o ITR precisa encaminhar o pacote LISP para um *Proxy* ETR (PETR) na borda da rede de destino, que aceita seu pacote, desencapsula o pacote original e entrega ao ETR. Na direção inversa, o pacote é entregue a um *Proxy* ITR (PITR) que encapsula a resposta em um pacote LISP e o entrega ao RLOC de origem, conforme ilustrado na Figura 4.2. (MENTH; KLEIN; HARTMANN, 2010).

Com estas características, o LISP pode ser utilizado para *multihoming* e também para mobilidade. Pois o MN implementa um ITR/ETR e, quando MN acessa uma nova rede, recebe um novo endereço de localização local, chamado de LLOC, assim, a mensagem

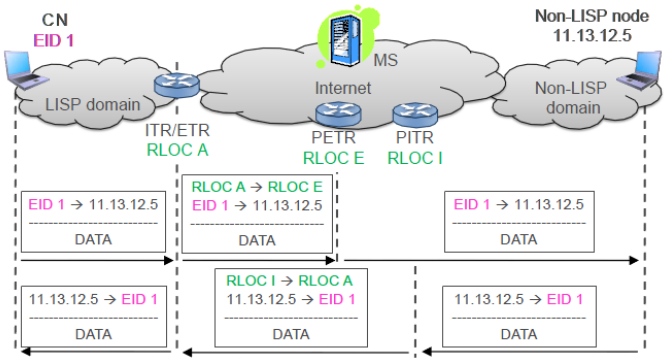

Figura 4.2: Comunicação entre redes LISP e Non-LISP (MENTH; KLEIN; HARTMANN, 2010).

destinada ao CN é encapsulada em um pacote LISP, que contém seu novo endereço de localização LLOC endereçado ao RLOC do destinatário. Esta mensagem, quando recebida pelo ITR da borda da rede, será novamente encapsulada e encaminhada a um PETR na borda da rede de destino, que realizará a primeira desencapsulação e encaminhará ao ETR, que por sua vez entregará ao CN. O nodo de destino, quando responder a mensagem ao MN, que realizou a troca de rede, obterá como resposta uma falha de mapeamento, obrigando seu ITR e realizar uma nova consulta em um *MAP-Revolver* para descobrir o novo endereço RLOC do MN. O fluxo deste processo pode ser observado na Figura 4.3.

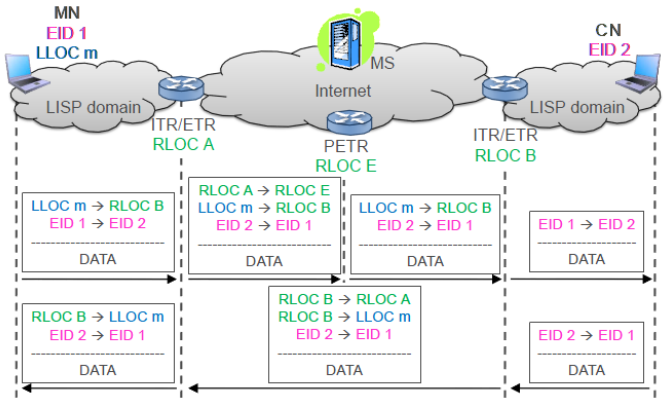

Figura 4.3: Comunicação entre MN e CN em redes LISP (MENTH; KLEIN; HARTMANN, 2010).

4.2 *Host Identity Protocol*

O HIP (*Host Identity Protocol*), reconhecido pelo protocolo de número 139, também foi proposto como uma solução à mobilidade (MOSKOWITZ et al., 2008). Neste protocolo de separação entre identificação (ID) e localização (LOC), é criada uma nova camada entre a camada de rede e a camada de transporte no modelo de referência OSI. Seu objetivo é apropriar ao *host* um identificador único, transformando a camada de rede em um localizador do *host* (GURTOV; PATHAK; KOMU, 2009).

Para realizar esta identificação, o *host* possui um par de chaves assimétricas (Pública / Privada) únicas. A chave pública é conhecida como HI (*Host Identifier*), mas para

identificar o *host* no protocolo HIP é utilizado o HIT (*Host identify Tag*), que se trata de um *hash* de 128 bits gerado a partir do HI, este tamanho de 128 bits é devido ao tamanho do campo de origem/destino do cabeçalho IPv6, para criar compatibilidade.

Para realizar a comunicação entre dois nodos utilizando o protocolo HIP, é necessário que ocorra uma negociação de autenticação em quatro passos (*handshake*) entre as partes. Esta negociação é chamada de *Puzzle Exchange*. Ela inicia no momento em que o nodo de origem (*Initiator*) envia um pacote contendo o seu HIT para o endereço HIT do nó de destino (*Responder*), que responde ao *Initiator* com um quebra-cabeça baseado no número HIT do *Initiator*. O *Initiator* por sua vez envia um pacote com a solução do quebra-cabeça, ao *Responder* que devolve um pacote contendo a assinatura e o cabeçalho HIP, finalizando o processo de autenticação (Figura 4.4). Este processo serve para evitar certos ataque DDoS (*Distributed Denial-of-Service Attack*) e aumentar a segurança do comunicação, pois permite que a chave pública utilizada para geração do HIT possa ser trocadas através do algoritmo *Diffie-Hellman* e utilizada para criptografar a comunicação entre as partes através do protocolo ESP *(Encapsulation Security Payload Header)*.

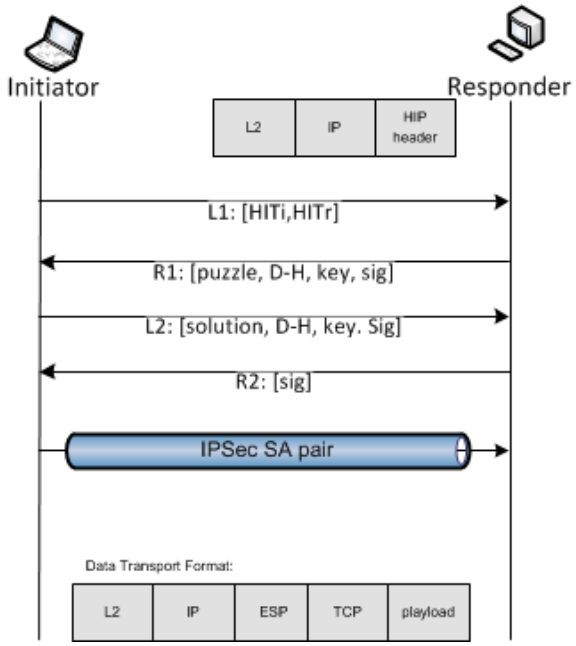

Figura 4.4: Criação de uma comunicação HIP (BOKOR; NOVáCZKI; JENEY, 2010).

Para realizar a comunicação entre as partes utilizando os endereços HIT, é necessário conhecer a localização do HIT de destino, o que pode ser realizado com o uso do campo *Resource Record* (RR) incluído nos servidores de DNS. Assim, quando um nó da rede necessita descobrir o IP de um *Correspondent Node* (CN), ele realiza uma consulta ao DNS pelo identificador HIT, obtendo como resposta o IP do *Correspondent Node*.

Porém, quando a comunicação a ser estabelecida é entre um MN e um CN, é necessária uma constante atualização do DNS para saber a atual localização do MN, o que acarreta em falhas de comunicação, devido ao tempo de convergência das informações na hierarquia de servidores DNS, que podem armazenar em cache as informações de endereçamento recebidas com vistas a evitar repetição de consultas frequentes.

No intuito de resolver esta dificuldade, foi proposta a inclusão de um novo agente, o *Rendez-vous Server* (RVS), seu objetivo é manter atualizada a informação de localização

dos nós móveis que utilizam HIP. Assim, conforme a Figura 4.5, quando um CN realizar a requisição do endereço IP de um MN ao DNS (1), receberá como resposta o endereço de um RVS (previamente cadastrado no DNS). No momento que o CN iniciar a comunicação com um RVS, este agente verificará em sua tabela a atual localização do MN e encaminhará o pacote a ele (2), que responderá diretamente ao CN (3), pois recebeu seu endereço no pacote redirecionado pelo RVS. Neste momento se inicia o processo de *handshake* entre o CN e MN (MOSKOWITZ et al., 2008).

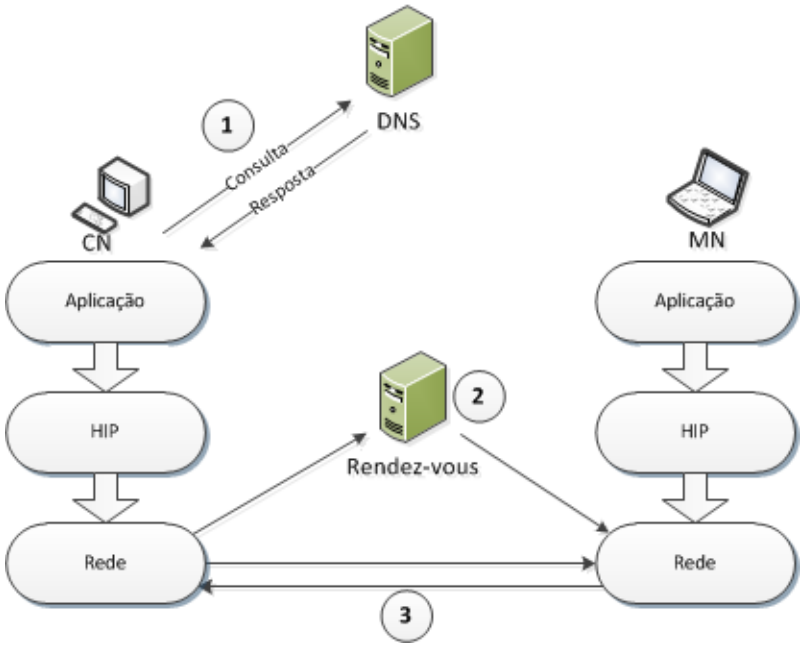

Figura 4.5: HIP. Comunicação entre CN e MN.

Após a descoberta dos endereços de localização, a comunicação ocorrerá através dos endereços HIT, já conhecidos por ambos os nodos. Ocorrendo a mobilidade de MN, o mesmo encaminhará um pacote de *readdress* ao RVS informando seu novo endereço e um pacote de *redirect* ao CN para que não ocorra falha na comunicação.

4.3 *Site Multihoming by IPv6 Intermediation*

SHIM6 é uma solução implementada diretamente nos *hosts* para utilização de *multihoming*, onde, por exemplo, um único *host* de uma rede pode prover serviço para vários ISPs possuindo *n* endereços IPs.

Conforme a Figura 4.6, ele é uma nova sessão entre a camada de rede (*layer-3*) e a camada de transporte (*layer-4*). Em seu funcionamento, um endereço IPv6 é usado para servir como Localizador e Identificador, chamado de ULID (*Upper Layer Identifier*).

Em SHIM6, uma sessão da camada de transporte contém os endereços dos *hosts* de origem e destino (ULID). Quando uma sessão inicia, a camada SHIM6 escolhe um par de localizadores de ambos os lados para configurar uma sessão de transmissão. Se ocorrer falha de conexão ou congestionamento, a camada SHIM6 fica responsável pela comutação do tráfego para um novo par localizador (contexto). Este processo ocorre sobre a camada IP e todo o processo é absolutamente transparente para aplicações das camadas superiores.

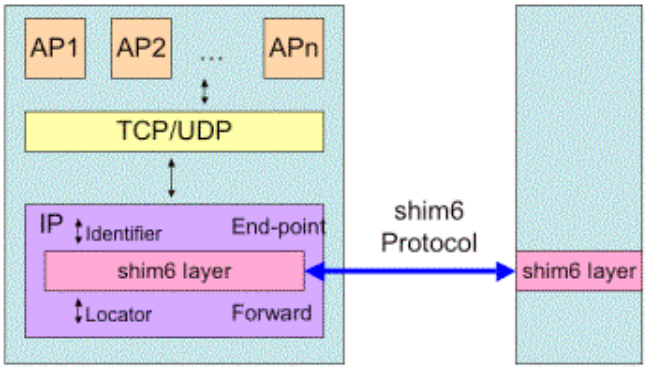

Figura 4.6: Arquitetura SHIM6 (BRAUN, 2009).

Na Figura 4.7, Host1 possui conexão com três provedores e obtêm três localizadores a partir deles, que são A1 do ISP A, B1 em ISP B e C1 do ISP C. Assim, o localizador do conjunto de Host1 é (A1, B1, C1). Da mesma forma, o localizador do conjunto de Host2 é (D2, E2), assim existem seis caminhos possíveis (A1D2, A1E2, B1D2, B1E2, C1D2, C1E2).

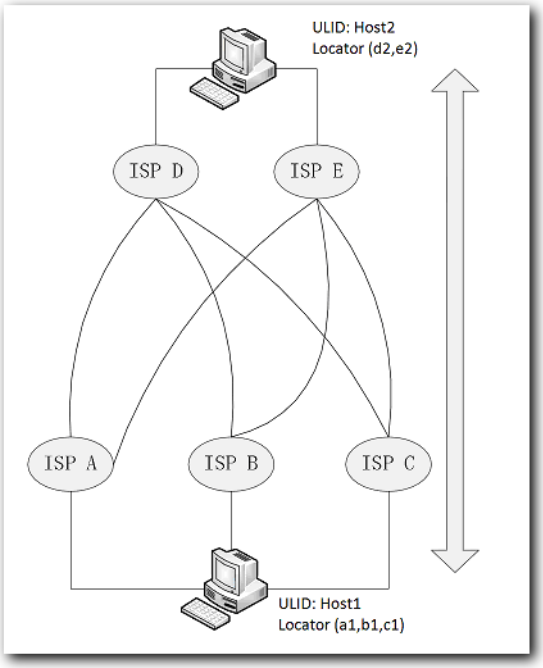

Figura 4.7: Funcionamento SHIM6 (LIU; BI; WANG, 2009).

Quando a conexão TCP entre o Host1 e Host2 é estabelecida, se um dos *hosts* possuir *multihoming*, este enviará um pacote de negociação para anunciar que o SHIM6 deve ser utilizado, para isto, é negociado um *handshake* em quatro vias, assim como no HIP. Nessa negociação serão trocados pacotes referente às configurações de *multihoming* onde será definido um *context tag* mantido pelos dois *hosts*, a fim de identificar a comunicação e prevenir certos ataques DoS. Durante a transmissão, o Host1 calcula estatísticas de desempenho da conexão através dos valores de RTT (Round-Trip Time) (LIU; BI; WANG, 2009). Ocorrendo um congestionamento ou falha no canal de comunicação, os *hosts*

trocam de rota durante a conexão. Também são avaliadas estatísticas através dos *n* IPs disponíveis em um nodo, possibilitando uma troca de rota entre as partes, na localização de um caminho mais rápido.

Outra característica importante do SHIM6 é a capacidade de realizar *fork*, isto é, durante uma conexão entre um par de *hosts* utilizando seus respectivos ULIDs é possível criar um novo canal de comunicação, utilizando localizadores diferentes em um novo *handshake*. Assim, é possível utilizar o protocolo SHIM6 para múltiplas conexões entre destinos iguais ou diferentes. Mas habilitando a capacidade de múltiplos localizadores para o mesmo ULID, é necessário garantir que estas informações oriundas de um CN, não se trata de uma tentativa de ataque de um terceiro *host* encaminhando seu endereço para se passar por CN. Para isto, o protocolo disponibiliza duas formas de proteção.

A primeira forma de proteção é utilizar a CGA (*Cryptographically Generated Addresses*). Refere-se a uma técnica de geração de *hash* utilizando uma chave privada e os 64 bits menos significativos do endereço ULID, o qual é transmitido junto com a chave pública durante o processo de *Handshake*, permitindo ao receptor decodificar o *hash* e comparar com o endereço de origem para garantir quem está enviando as informações. Essa técnica é utilizada sempre que ocorrer uma alteração de localizador entre os pares em comunicação.

A segunda possibilidade é a utilização do HBA (*Hash-Based Addresses*), que pode ser complementar a CGA, pois utiliza os prefixos /64 de todas as interfaces de localização de um *host* para a geração do *hash*, mas que pode ser um problema para a mobilidade, já que um MN tende a alterar seus prefixos durante a troca de rede. Isso limita o uso de HBA apenas para redes *multihoming*, que possui seus prefixos fixos durante a realização de uma comunicação.

No próximo capítulo serão apresentadas as soluções que implementam os protocolos que realizam a separação da identificação e localização de um nodo, assim como os protocolos que utilizam as funcionalidades do IPv6 para provimento de mobilidade.

5 VISÃO GERAL DAS IMPLEMENTAÇÕES DOS PROTOCOLOS DE PROVIMENTO DE MOBILIDADE

Neste capítulo serão apresentadas as funcionalidades implementadas para os protocolos de provimento de mobilidade, os quais foram classificados em IPv6 **puros** e **híbridos**. **IPv6 puros** são os protocolos que utilizam apenas as características disponibilizadas pelo IPv6, como *neighbor discovery* e cabeçalhos de extensão. **Híbridos** são os protocolos que separam a identificação da localização de um nodo, originalmente desenvolvidos para *multihoming*, mas que estão sendo destaque em soluções de mobilidade.

5.1 Protocolos de provimento de Mobilidade sobre IPv6 Puros

O MIPv6 por ser o primeiro protocolo desenvolvido. Possui todas suas mensagens implementadas nos cabeçalhos de extensão do IPv6, como a utilização do cabeçalho *destination options* (60) e *routing type2* (43), para encaminhar ao CN o endereço do seu *home agent* enquanto estiver em uma rede estrangeira, ou a utilização das mensagens ICMP de número 144 à 147, para a solicitação de prefixos na rede estrangeira e para a descoberta de um *home agent*. Funcionalidades estas implantadas no sistema operacional *Linux*, através do pacote UMIP *(Mobile IPv6 stack for the GNU/Linux Operating System)*[1].

Para a gerência de redes que utilizam *Mobile IPv6*, existe a MIB MOBILEIPV6-MIB (KEENI et al., 2006), a qual possui cinco grupos definidos:

- Mip6Core: grupo genérico, contendo objetos comuns a todas as entidades *Mobile* IPv6.
- Mip6Ha: grupo com modelos para o *home agent* com objetos referentes aos serviços e anúncio oferecidos pelo *home agent* a cada um dos links estabelecidos.
- Mip6Mn: grupo com modelos relativos ao nó móvel. Possui objetos para *Dynamic Home Agent Discovery* e objetos que registram o movimento do nó móvel.
- Mip6Cn: define os modelos para o CN, essencialmente registra a realização de Otimização de Rota entre o CN e MN.
- Mip6Notifications: define o conjunto de notificações utilizadas para monitoramento assíncrono das entidades envolvidas na *Mobile* IPv6.

[1] http://umip.linux-ipv6.org

O FMIPv6 não possui o suporte nativo em sistemas operacionais como existe no MIPv6. No *Linux* ele pode ser instalado através de um pacote compilado junto ao *Kernel* (IVOV et al., 2007) e por aplicativos que auxiliam na troca para novas redes. Esses aplicativos utilizam os cabeçalhos de extensão já definidos no IPv6 e o tipo 150 do protocolo ICMPv6, reservado especialmente para experimentos em mobilidade (KEMPF, 2005).

Para o protocolo PMIPv6 é necessário que os agentes LMA e MAG estabeleçam um túnel para a transferência de dados entre CN e MN, com isto, não há a necessidade de implementações a serem realizadas nos MNs e CNs, apenas a inclusão dos referidos agentes, conforme a RFC 5213 (LEUNG et al., 2008).

Já o HMIPv6 possui uma implementação para *Linux* realizada pela Universidade *Monash* (DALEY, 2004), mas que não possui novas versões desde 2004, apesar da RFC 5380 (*Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*) datar de outubro de 2008.

Com exceção do protocolo PMIPv6, todos os outros citados possuem compatibilidade com a MIB MOBILEIPV6-MIB por serem implementados sobre o pacote UMIP.

5.2 Protocolos Híbridos

O protocolo LISP é desenvolvido por três frentes de trabalho. O OpenLisp[2] versão 0.1.0, disponibilizado em setembro de 2010, implementa a draft-ietf-lisp-08 sob o Sistema Operacional FreeBSD. No Sistema Operacional *Linux* está disponível uma versão Alpha[3] onde é possível realizar o mapeamento entre RLOC e EID, mas não estão implementados os *MAP Servers*, o que impossibilita a utilização em mobilidade, por não existir um local para armazenar a localização do MN. Um terceiro projeto é mantido pela empresa Cisco Networks, que disponibiliza implementações para *Ingress Tunnel Router (ITR), Egress Tunnel Router, Proxy ITR (PITR), Proxy ETR(PETR), Map Resolver (MR), Map Server (MS)* e LISP+ALT, sendo possível criar uma estrutura completa para provimento de *multihoming*, utilizando a versão NXOS de seu Sistema Operacional. Provendo estruturas inteiras de contingência utilizando *Cloud Computing*, realizando o redirecionamento do roteamento de uma estrutura para outra através de LISP, em caso de necessidade (CISCO, 2011a).

O protocolo HIP foi disponibilizado inicialmente através do simulador OMNET++, em sua implementação do HIPSim++ (*Host Identity Protocol Simulation Framework for INET / OMNeT++*) desenvolvido pelo Departamento de Telecomunicações da *Budapest Univerity of Technology and Economics*, este *framework* possibilita a realização de experimentos para cálculo de tempo de troca de redes e análise das mensagens durante a comunicação, porém não possui implementado o algoritmo de troca de chaves para a criação de um canal seguro através de criptografia. Para *Linux* existe o HIPL (GURTOV; PATHAK; KOMU, 2009), desenvolvido com a colaboração de empresas como Nokia, Ericson, Elisa e Governo da Finlândia[4], mantido atualmente pela comunidade de software livre que implementa todas as funcionalidades definidas pela RFC 5201.

Por fim, o protocolo SHIM6 possui apenas uma implementação funcional desenvolvida por S. Barré (2011), pois as implementações OpenHIP (AHRENHOLZ; HENDER-

[2]http://www.openlisp.org
[3]https://github.com/aless
[4]http://infrahip.hiit.fi/index.php?index=about

SON, 2008) e MipShim6 (BARRé et al., 2009a) estão em fase de prototipação. Esta implementação identificada por LimShim6, foi desenvolvida para *Linux* e permite a utilização de múltiplos endereços de localização para um único UUID. Sobre esta proposta está sendo desenvolvido o *TCP Extensions for Multipath Operation with Multiple Addresses* (MPTCP) que propõe uma versão do TCP capaz de realizar múltiplas conexões simultâneas para a busca da mesma informação (FORD et al., 2011).

5.3 Trabalhos relacionados

Na comparação entre as implentações dos protocolos de mobilidade apresentados, a maioria dos autores utiliza o protocolo *Mobile* IPv6 como referência. Wang, Li e Yan (2009) em seu trabalho compararam o *Handover* HMIPv6 e MIPv6, não exaltando a metodologia utilizada em seus experimentos. Oliveira, Cascardo e Loureiro (2003) realizam a comparação de *Bind Updates* enviados pelos referidos protocolos utilizando simuladores. Kong e Lee (2008) em seu trabalho, compararam o tempo de *Handover* dos protocolos MIPv6, HMIPv6, FMIPv6 e PMIPv6, através de simulações e análise das mensagens dos protocolos, assim como Costa, Moreno e Hartenstein (2003), que através de simulações analisam o tempo de *Handover* e a quantidade pacotes perdidos dos protocolos FMIPv6, HMIPv6 e suas variações, levando em consideração em suas simulações o uso de quatro *Access Points* e cinquenta *Mobile Nodes*.

Menth, Klein e Hartmann (2010) demonstram as oito conjunções de agentes necessários para prover mobilidade sobre LISP, no relacionamento entre redes LISP e redes Não-LISP, Choi et al. (2009) analisaram o tempo de *Handover* no uso de mobilidade em LISP, obtendo resultados entre 1,469 e 1,679 segundos, desconsiderando o tempo de *Handover* físico.

Rahman et al. (2010) compararam o *Handover* do protocolo SHIM6 com o *Mobile* IPv6 através de implementações disponíveis, enquanto Oliva (2007) demonstra o *Application Recovery Time* utilizando SHIM6 através de simuladores, obtendo tempos de *Handover* entre 3 e 20 segundos em aplicações como Telnet e FTP, dependendo das variáveis utilizadas nos experimentos.

Na análise do protocolo HIP, Jokela et al. (2004) compararam o tempo de *Handover* entre os protocolos MIPv6 e HIP, obtendo respectivamente 8,05 e 2,46 segundos.

Estes e outros trabalhos demonstram o anseio pela comparação entre os protocolos de mobilidade, a fim de demonstrar suas funcionalidades e aplicabilidades. Contudo, a partir destes trabalhos não é possível classificar os referidos protocolos quanto ao tempo de *Handover* e suas funcionalidades, devido as diferentes metodologias utilizadas nos experimentos.

6 METODOLOGIA E EXPERIMENTOS REALIZADOS

Para atingir o objetivo de analisar as propostas e os protocolos de provimento de mobilidade, a fim de obter uma comparação quanto ao seu tempo de *Handover*, funcionalidades e agentes necessários, foi realizada uma série de experimentos com as implementações existentes sobre o mesmo ambiente, apresentados nos tópicos que seguem.

6.1 O Ambiente dos testes

Para análise do tempo de *Handover* e funcionalidades, foi usada a estrutura mostrada na Figura 6.1, composta de quatro computadores e dois *Access Points*. Nesta estrutura, foi utilizado: um computador Pentium 4 de 2,8 Ghz e 1Gb de RAM para atuar como CN (*Correspondent Node*), um Netbook Aton N450 com 2 Gb de RAM para atuar como MN (*Mobile Node*), dois *Access Points* e duas máquinas virtuais sobre o sistema de virtualização *Oracle VirtualBox* em modo *bridge*, que em conjunto, atuam como roteadores, auxiliando e/ou controlando a mobilidade de MN entre as redes.

As máquinas virtuais utilizadas em todos os experimentos possuem: um processador, 768 Mb de RAM e Sistema Operacional Ubuntu 10.04.2 LTS-32 bits ou FreeBSD 8.1 32 bits, virtualizados sobre um Intel i3 de 3,1Ghz e 4 Gb de RAM .

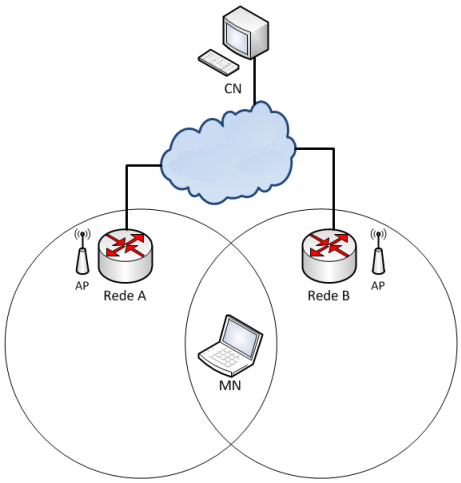

Figura 6.1: Estrutura de rede utilizada nos testes.

Sobre esta estrutura foram analisadas as seguintes implementações:

- MIPv6: *Mobile IPv6 stack for the GNU/Linux Operating System* (KUNTZAND et al., 2011).
- FMIPv6: fmipv6, versão 1.0 (IVOV et al., 2007).
- HMIPv6: Radvd 0.9.7 (DALEY, 2004) e hmip6d (SILVA; ALMEIDA, 2009a).
- PMIPv6: *Proxy Mobile IPv6*, http://www.openairinterface.org/ (OPENAIR3, 2010).
- HIP: hipl[1], versão 1.0.6-5193.
- LISP: OpenLISP[2] versão 0.1.0.
- SHIM6: LinShim6 (BARRé; RONAN; BONAVENTURE, 2011).

As medições do tempo de *Handover* das implementações dos protocolos em questão, foram realizadas durante o envio de pacotes ICMP e durante a transferência de dados por TCP, simulando assim, uma real utilização das implementações. Durante as medições, também foi coletada a quantidade de mensagens de controle introduzidas no canal de comunicação por estes protocolos, a fim de verificar seu volume perante os pacotes de dados. Todos os experimentos foram executados cinco vezes, após estabelecidos os parâmetros de configuração das implementações, e obtiveram um coeficiente de variação igual ou inferior a 0,02.

A análise das funcionalidades e tempos de *Handover* das implementações dos protocolos supracitados, foi realizada sobre o padrão *Wireless* 802.11n. O tempo de *Handover* físico, apesar de computado por fazer parte do *Handover* dos protocolos, não será ensejo de nossos estudos. Os mecanismos de *Handover* sobre *Delay-Tolerant Network* (DTN), como *Vehicular ad hoc network* (VANET) ou *Mobile ad hoc network* (MANET) (LUO et al., 2008), que possuem foco no *Handover* físico (ex.: 802.11p-*Wireless Access for Vehicular Environments*), podem ser utilizados em conjunto com os protocolos de camada de rede apresentados neste trabalho.

Outras análises como a escalabilidade dos protocolos, realizado por (PéREZ-COSTA; TORRENT-MORENO; HARTENSTEIN, 2003), e segurança das implementações, não foram abordados neste estudo devido a indisponibilidade de recursos para a realização dos mesmos.

6.2 Experimentos realizados

6.2.1 Experimentos com *Mobile* IPv6

O pacote MIPL (*Mobile IPv6 for Linux*) 2.0.2-umip-0.4 foi utilizado para a realização dos experimentos em *Mobile* IPv6. Este pacote necessita que seja habilitado no *Kernel* do *Linux* o suporte ao *Mobile IPv6* e instalado o serviço de *Router Advertisement* para atribuição automática de endereços *Stateful* aos dispositivos que entrarem na rede.

[1] http://infrahip.hiit.fi

[2] http://www.openlisp.org

Os procedimentos executados para realização destes e dos outros experimentos estão no Apêndice A.

6.2.1.1 Configuração dos experimentos com Mobile IPV6

Foram utilizados seis dispositivos na realização dos experimentos. Foi necessária a instalação do pacote MIPL no *Home Agent* (HA) e no *Mobile Node* (MN) para que, durante a movimentação do MN para a *Foreign Network* (FN), se constituísse um canal seguro entre MN e HA, utilizado para encaminhar os pacotes até CN. Para a configuração dos endereços foram instalados o serviço de *Router Advertisement* (RA) no HA e na FN, conforme a Figura 6.2.

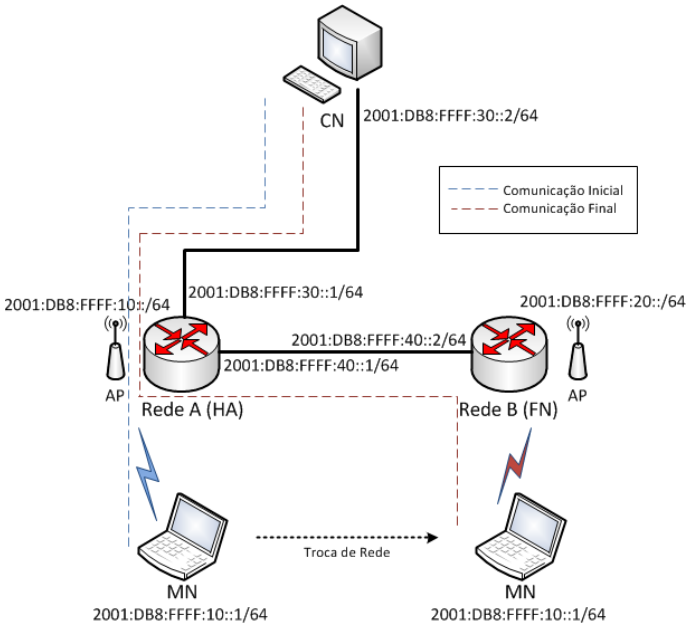

Figura 6.2: Configuração da rede nos experimentos com MIPv6.

6.2.1.2 Resultados obtidos com MIPv6

- **Pacotes de controle gerados pelo protocolo MIPv6:**

O primeiro procedimento foi averiguar qual a utilização do canal sem a transferência de dados, isto é, a utilização do canal de comunicação por pacotes inerentes ao protocolo estudado. Para isto realizou-se três capturas de MN de 120 segundos:

1. Na primeira captura, sem o uso de MIPv6, foram recebidos 37 pacotes, referentes a anúncios de *neighbor discovery*, gerando 0,308 pacotes por segundo e 3.182 *Bytes*.

2. Na segunda captura, foi habilitado o anúncio de *Router Advertisement* (RA) (ROQUE et al., 2011) realizado pelo *Home Agent* (HA), obtendo como resultado 96 pacotes, gerando 0,8 pacotes por segundo e 10.576 *bytes*. Esta quantidade de pacotes em apenas dois minutos é reflexo da configuração de *Router Advertisement* existente na *Home Network*, configurado para enviar mensagens de RA a cada três segundos no máximo. Isto possibilita uma rápida convergência, no momento em que o agente de mobilidade (HA ou

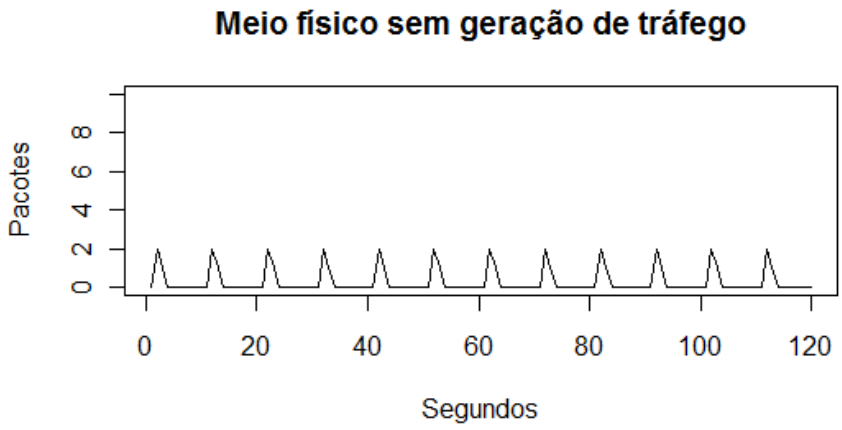

Figura 6.3: Pacotes por segundo sem geração de tráfego no protocolo MIPv6.

FN) percebe a existência do *Mobile Node* na rede. Este tempo de envio de *Router Advertisement* pode ser reduzido a um intervalo de 300 ms, porém não é usual esta configuração em redes IPv6.

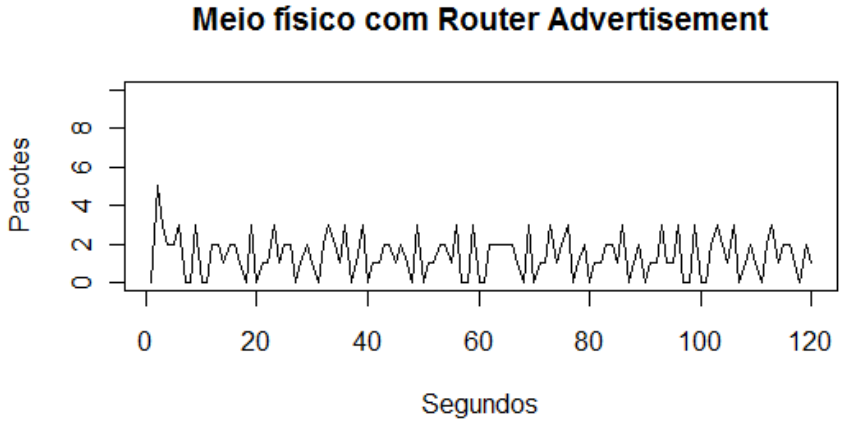

Figura 6.4: Pacotes por segundo com RA no protocolo MIPv6.

3. Por último foi habilitado o processo de mobilidade *mip6d* em MN. Com isto obteve-se um aumento inicial de pacotes referente ao registro do MN em seu HA de 142 pacotes, com uma média de 1,03 pacotes por segundo e 32.201 *bytes*.

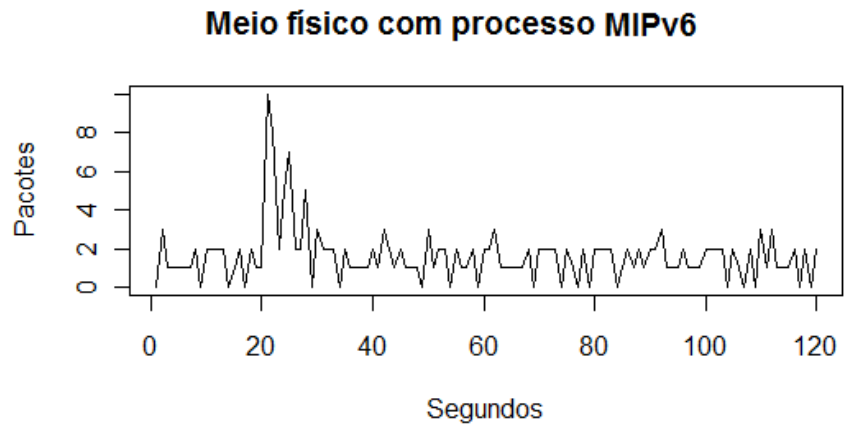

Figura 6.5: Pacotes por segundo com RA e MIPv6.

Esta quantidade de pacotes de controle representa 0,02 % da taxa de transferência de *bits/s* permitida neste canal de comunicação, conforme a mensuração da capacidade de transferência executada e demonstrada na Figura 6.6, abordada no tópico sobre taxa de transferência.

- **Cálculo de Round-Trip Time:**

O objetivo desta análise é verificar qual o *Round-Trip Time* (RTT) obtido entre MN e CN utilizando o MIPv6 em uma *Foreign Network*, em comparação a uma comunicação sem o uso de protocolos de mobilidade. Para isto foram realizadas coletas com e sem o uso do MIPv6, utilizando pacotes de 56 bytes realizadas em 3 amostras de 100 pacotes cada.

Tabela 6.1: RTT entre MN e CN (em ms).

RTT	**Com MIPv6**	**Sem MIPv6**
Mínimo	2,220	1,600
Mediana	2,575	2,240
Média	2,926	2,583

Na Tabela 6.1 pode-se observar que após a criação do túnel entre MN e CN, obteve-se uma elevação de 15% com o uso do MIPv6, devido ao canal de comunicação estabelecido entre MN e o HA utilizar IPsec.

- **Taxa de Transferência:**

Para mensurar a taxa de transferência de dados por TCP, foi utilizado o software Iperf[3], enviando tráfego de MN ao CN (Figura 6.6), sendo realizadas duas capturas: uma com MN em sua rede local e outra com MN na rede estrangeira. Nos testes foi utilizado o protocolo 802.11n nos *Access Points*, que permite uma taxa de transferência máxima de 300 Mbps, mas limitado pela interface de rede de HA e CN que possuem interfaces FastEthernet (100 Mbps).

Apesar de a estrutura física ser idêntica entre as redes, quando MN está na FN, ele realiza a comunicação por IPSec, reduzindo sua capacidade de transferência de dados devido a criptografia. Durante esta análise foi observado que a CPU teve uma elevação de apenas 5% em sua utilização, também devido à criptografia dos pacotes, o que não influenciou nos experimentos.

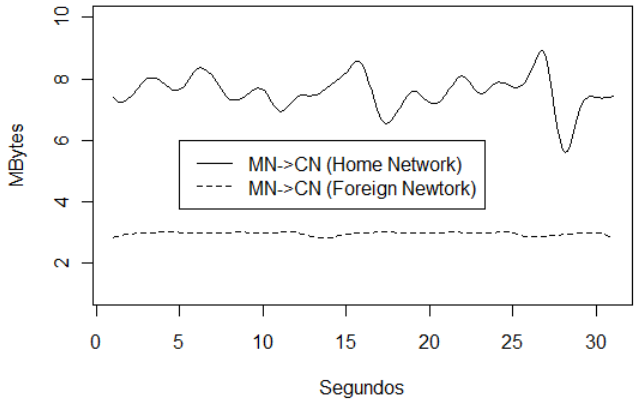

Figura 6.6: Taxa de transferência com MIPv6.

Conforme a Figura 6.6, foi obtida uma taxa de transferência de 7,9MB/s, quando realizada a medição na rede local e uma taxa de 2,95 MB/s na rede estrangeira, devido á utilização de IPsec e do aumento da rota dos pacotes.

- **Tempo de *Handover*:**

Os tempos de *Handover* durante a troca de Rede A (HA) para Rede B (FN), na transferência de dados por TCP, resultaram entre 13,101 e 15,613 segundos nos cinco experimentos realizados. Esta variação ocorreu devido ao tempo de recebimento do endereço IPv6 através de RA na FN, o qual estava configurado para ser realizado no intervalo de um a três segundos.

[3]http://sourceforge.net/projects/iperf/

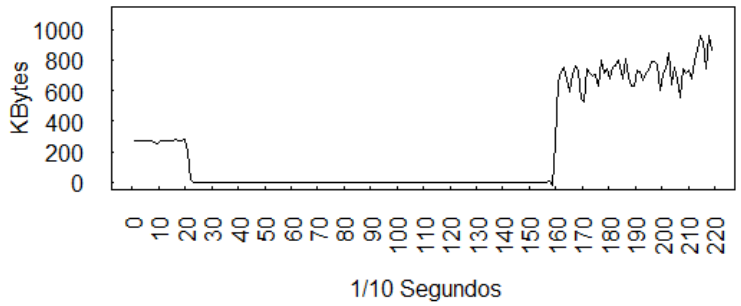

Figura 6.7: Tempo de *Handover* no MIPv6 na transferência por TCP.

Com isto, para desconsiderar o tempo de endereçamento durante a troca da rede (recebimento de RA), foi realizado o processo de mobilidade várias vezes entre as redes em um curto espaço de tempo, assim MN não necessitou solicitar um novo endereço, ou aguardar o recebimento de *Router Advertisement*, pois já possuía os endereços em sua interface.

Neste experimento, obteve-se 12,901 segundos de *Handover*, conforme a Figura 6.8, na transferência de dados por TCP entre MN e CN.

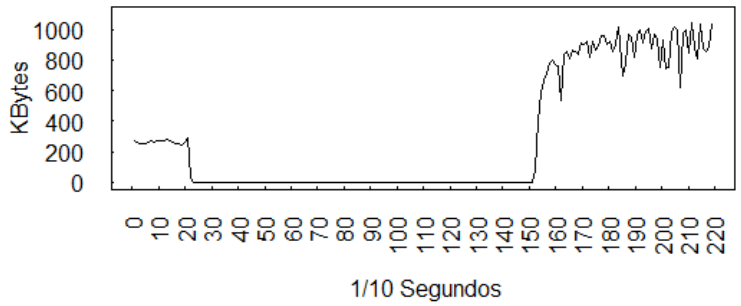

Figura 6.8: *Handover* sem a realização de endereçamento no protocolo MIPv6.

Em um terceiro teste foi analisado o tempo de *Handover* através do envio de pacotes ICMP a uma taxa de um pacote por milissegundo, com a mesma estrutura do experimento anterior. Nesse experimento, obteve-se uma melhora significativa no tempo de *Handover*, passando a ser realizado em 6,8 segundos (Figura 6.9), pois não ocorreu o processo de negociação de conexão necessário na comunicação por TCP.

Porém, incluindo o tempo de endereçamento (recebimento de RA) esse intervalo foi de 9,432 segundos.

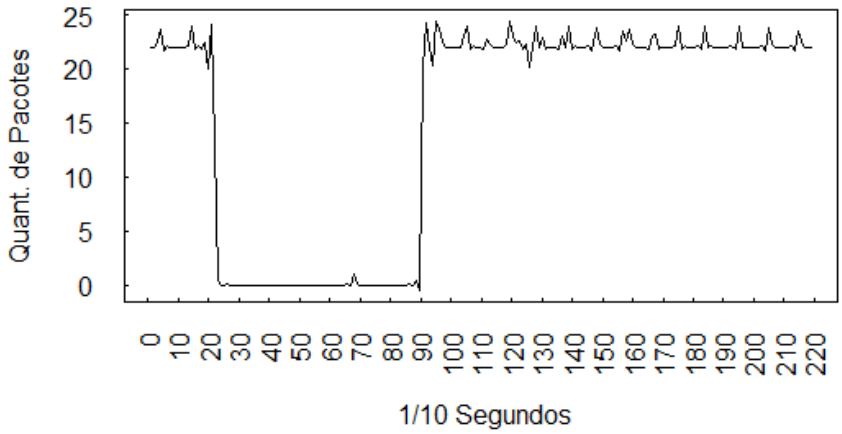

Figura 6.9: Quantidade de pacotes (ICMP) durante *Handover* no protocolo MIPv6.

Por último, foi mensurado separadamente o tempo de *Handover* da camada de enlace do MN, isto é, o tempo que o *hardware* a e pilha TCP/IP do Sistema Operacional demora para desconectar de um *Access Point* e conectar ao outro através da interface *Wireless*. Neste experimento obteve-se um tempo de 5,152 segundos.

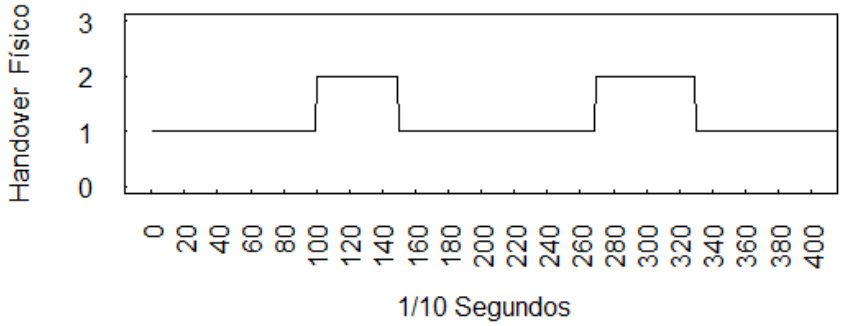

Figura 6.10: *Handover* físico da *interface Wireless.*

Com isto, conclui-se que o tempo de *Handover* é composto por três partes: *Handover* de enlace, de endereçamento e de registro no HA. Essas três partes, mensuradas com o envio de pacotes ICMP, representaram os seguintes tempos:

Handover de enlace: 5,152 segundos.

Handover de endereçamento: variação entre 0,839 e 3 segundos.

Handover de Registro no HA: 1,648 segundos.

6.2.2 Experimentos com *Fast Handover for Mobile* IPv6

Para realização do experimento com o FMIPv6 utilizou-se a implementação para *Linux* versão 1.0-rc1 (IVOV et al., 2007). O FMIPv6 tem por intuito minimizar o tempo de *Handover*, através da descoberta dos meios físicos existentes antes da realização do *Handover*. Isso é realizado através do processo *fmipv6-ar* instalado nos *Access Routers* e do *fmipv6-mn* instalado no *Mobile Node*, como ilustrado na Figura 6.11.

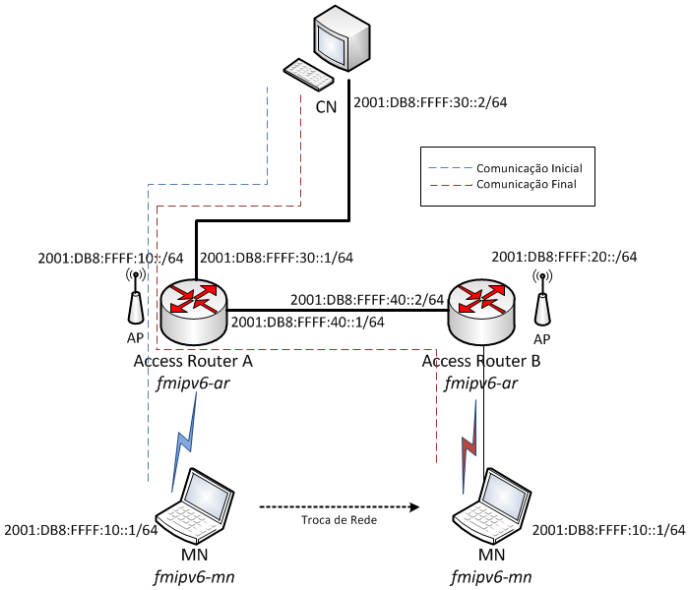

Figura 6.11: Estrutura da rede nos experimentos com FMIPv6.

O controle sobre o comunicação entre CN e MN continua sendo realizado pelo *Mobile* IPv6, isto é, o FMIPv6 é uma extensão do MIPv6. Nele, é necessário configurar os endereços MAC dos dispositivos que fazem parte do *Handover*: os endereços MAC dos *Access Routers* e dos *Access Points* que estão na *Home Network* e na *Foreign Network*. Este procedimento dificulta a escalabilidade desta implementação, já que as *Foreign Networks* precisam estar preparadas para o recebimento de um MN.

6.2.2.1 Resultados obtidos com FMIPv6

Como essa implementação do FMIPv6 utiliza o MIPv6, os tempos de RTT e de taxa de transferência permanecem inalterados se comparados com o MIPv6, pois ele é o responsável pela criação do canal de comunicação. Sendo assim, serão demonstrados os resultados referentes ao tempo de *Handover* e das mensagens de controle geradas pelo protocolo.

- **Mensagens de controle:**

Na medição dos pacotes de controle inseridos no canal de comunicação pelos protocolos FMIPv6 obteve-se 152 pacotes e 23.918 *bytes*, durante a amostragem de 120 segundos (Figura 6.12). Isso representa uma elevação de 7% na quantidade de pacotes gerados pelo MIPv6 com RA. Este pequeno aumento refere-se às mensagens de *Router Solicitation for Proxy* (RtSolPr), enviadas pelo MN ao *Access Router*, o qual recebe como respostas

uma mensagem de *Proxy Router Advertisement* (PrRtAdv) contendo informações sobre os APs disponíveis e previamente configurados nos *Access Routers*.

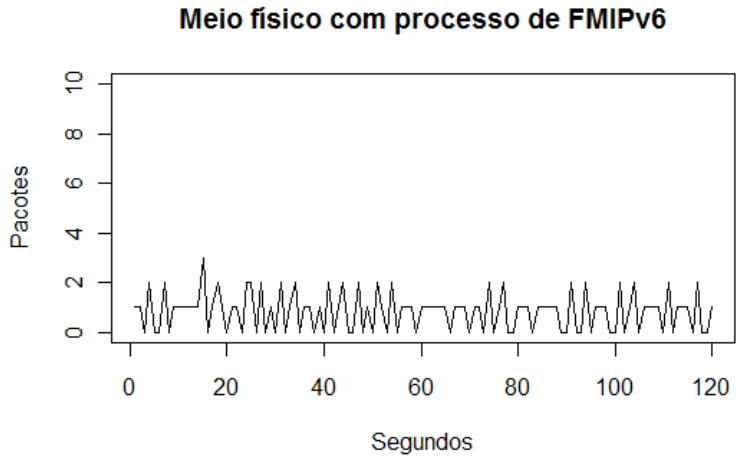

Figura 6.12: Pacotes por segundo com RA e FMIPv6.

- **Tempo de *Handover*:**

Nos experimentos de envio de pacotes ICMP a uma taxa de um pacote por milissegundo, obteve-se um tempo de *Handover* de 1,904 segundos (Figura 6.13), que é cinco vezes mais rápido que o mesmo teste em MIPv6 e deve-se ao fato de MN já possuir um endereço em cada interface de rede e já ser conhecido pelos dois *Access Routers*, necessitando apenas criar o túnel até o MN, processo o qual é realizado pelo MIPv6.

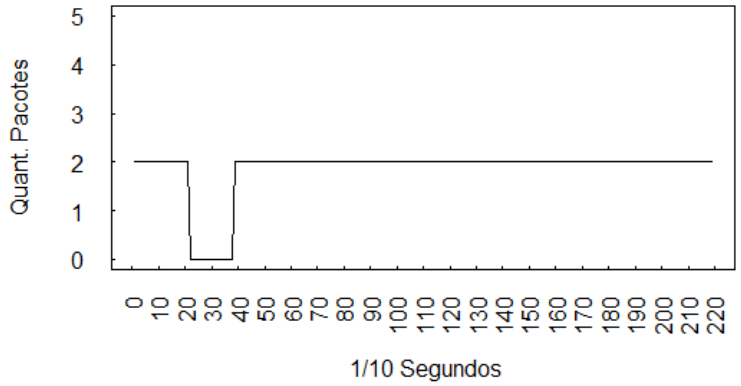

Figura 6.13: FMIPv6 - Tempo de *Handover* durante o envio de pacotes ICMP.

Nos experimentos realizados durante a transferência de arquivos, obteve-se também uma melhora no tempo de *Handover*, reduzindo o tempo de 12,901 segundos (MIPv6) para 6,114 segundos (FMIPv6).

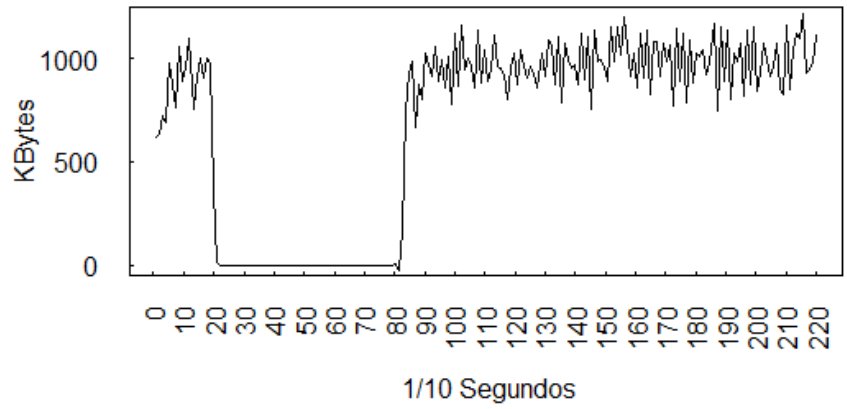

Figura 6.14: Tempo de *Handover* no FMIPv6 na transferência de arquivos.

6.2.3 Experimentos com *Hierarchical Mobile IPv6*

Para esta fase de testes foi usada a implementação disponibilizada pela Universidade *Monash*, que utiliza como base a implementação do MIPL 0.9.4, a qual possui compatibilidade com o Kernel 2.4.x do *Linux*. Como o MIPL foi praticamente reescrito em sua versão atual (2.0.2) e incorporada no Kernel 2.6.x do *Linux*, utilizou-se para análise do protocolo, a reimplementação realizada pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (SILVA; ALMEIDA, 2009b), a qual incorporou na versão 2.0.2 do MIPL as mensagens provenientes do protocolo HMIPv6 (SILVA; ALMEIDA, 2009a).

No protocolo HMIPv6 é necessário que as mensagens de *Router Advertisement* contenham os endereços dos *Mobile Anchor Points* (MAP), para o MN realizar o *Binding Update* no local correto. Para este propósito foi utilizada o RADvd versão 0.9.7, alterado pela Universidade *Monash* (DALEY, 2004).

No HMIPv6 validou-se a mobilidade local, isto é, a capacidade de realização de troca de redes dentro do mesmo domínio de rede, utilizando um *MAP Server* na rede estrangeira sobre a mesma estrutura dos experimentos do MIPv6.

6.2.3.1 Resultados obtidos com HMIPv6

Igualmente ao FMIPv6, o HMIPv6 utiliza como base o MIPv6 para o desenvolvimento desta implementação, com isto, o RTT e a taxa de transferência no HMIPv6 obteve os mesmos resultados que no MIPv6. Demonstrando-se então, o tempo de *Handover* e a utilização do canal pelas mensagens de controle do protocolo.

- **Mensagens de controle:**

Na verificação do tráfego de mensagens de controle, foi verificada a transferência de 23.102 bytes e 158 pacotes em 120 segundos (Figura 6.15). Ao contrário do esperado, não

ocorreu uma redução nas mensagens de controle, pois foi utilizada a mesma quantidade de envios de RA, para manter o mesmo tempo de endereçamento de MN na nova rede. Com isto a quantidade de mensagens de controle permaneceu próxima da quantidade verificada com o MIPv6, apenas com o acréscimo das mensagens de *Bind Updates* referente aos endereços *Regional Care-of Address* (RCoA) e *Link Care-of Address* (LCoA).

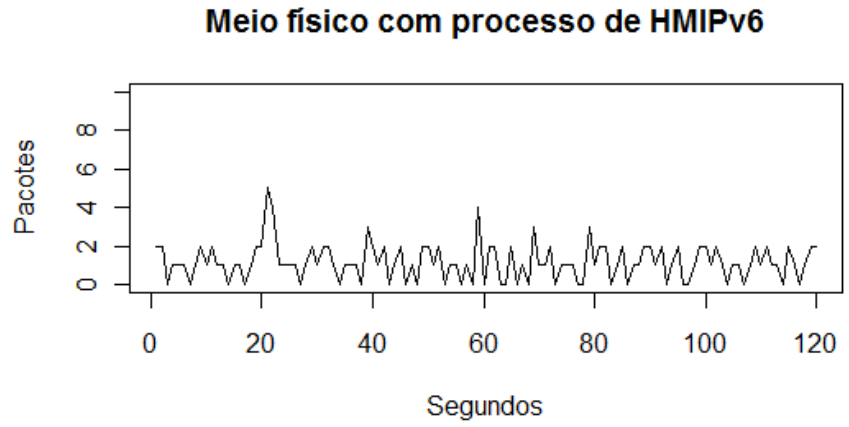

Figura 6.15: Pacotes por segundo com RA e HMIPv6.

- **Tempo de *Handover:***

No processo de *Handover* do protocolo HMIPv6 foram realizados experimentos sobre a mobilidade local. Nesta modalidade, o MN recebe um RA do MAP, contendo o endereço da rede e seu endereço LCoA, retornando para o MAP uma mensagem de *Bind Update*, informando seus endereços LCoA e o RCoA proveniente de sua rede de origem. Em um segundo momento, MN recebe do MAP um *Bind Ack* com a confirmação do processo. Nesta modalidade de *Handover* não há a necessidade de informar ao HA seu novo endereço LCoA.

Durante a realização do experimento não foi obtido sucesso na criação do novo roteamento entre o MAP e o CN após a mobilidade, impossibilitando a mensuração de tempo de *Handover* entre MN e CN através de transferência de arquivos, como realizado nos outros experimentos anteriores. Isto ocorreu devido a problemas na implementação utilizada, que não atualizou as tabelas de roteamento. Contudo, foi possível mensurar o tempo de *Handover* através da análise dos logs do processo *hmip6d*, como segue:

```
03:05:09.47 md_link_down: link down on iface wlan0 (3)
03:05:14.62 md_link_up: link up on iface wlan0 (3)
03:05:15.22 received RA from fe80:0:0:0:a00:27ff:fe6b:b4ff on iface 3
03:05:15.27 creating new prefix 2001:db8:ffff:20:0:0:0:0/64
03:05:15.29 creating new map 2001:db8:ffff:0:0:0:0:1000/3
03:05:15.30 creating new router fe80:0:0:0:a00:27ff:fe6b:b4ff on iface wlan0 (3)
03:05:15.43 looking for existing routers on iface wlan0 (3)
03:05:15.58 add new router fe80:0:0:0:a00:27ff:fe6b:b4ff on interface wlan0 (3)
03:05:15.72 adding default route via fe80:0:0:0:a00:27ff:fe6b:b4ff
03:05:15.77 add coa 2001:db8:ffff:20:4a5d:60ff:fe4a:37b3 on interface (3)
03:05:15.81 add rcoa 2001:db8:ffff:0:4a5d:60ff:fe4a:37b3
```

```
03:05:15.90 created tunnel from 2001:db8:ffff:...:37b3 to 2001:db8:ffff::1000
03:05:16.01 process_first_map_bu: New bule for MAP
03:05:16.04 Failed to del policy:
03:05:17.02 mh_send: sending MH type 5 from 2001:db8:ffff:...:37b3 to 2001:db8:ffff::1000
```

Por estes registros, é possível observar que o tempo de *Handover* físico utiliza 5,152 segundos, já mencionados, e que o processo de *Handover* do protocolo utilizou 2,4 segundos, incluindo o tempo de recebimento de RA. Como o tempo mensurado de RTT entre CN e MN é de 2,5 milissegundos, este valor se torna ínfimo, permitindo concluir que o tempo total de troca de rede desta implementação do HMIPv6 é de 7,551 segundos.

6.2.4 Experimentos com *Proxy Mobile IPv6*

Para avaliar o *Proxy Mobile* IPv6 foi utilizada a implementação PMIPv6 - v0.1 (OPENAIR3, 2010), disponibilizada pelo projeto *Open Air Interface*. Esta versão implementa o MAG e o LMA no mesmo aplicativo. Em seu projeto original foram utilizados *Access Points Cisco IRONET*, para realização da descoberta da movimentação de MN através dos *logs* enviados do *Access Point* para o MAG. Nesse experimento, se empregou o segundo recurso possível para a descoberta de movimentação: o envio de uma mensagem *Router Solicitation* (RS) realizada por MN. Para tanto, foi utilizado o aplicativo *ICMPv6 Router Discovery Tool* (rdisc6), na geração das mensagens de RS. Como este protocolo necessita a criação de dois novos Agentes, o LMA e o MAG, foi necessário alterar a estrutura da rede incluindo esses novos agentes, conforme a Figura 6.16.

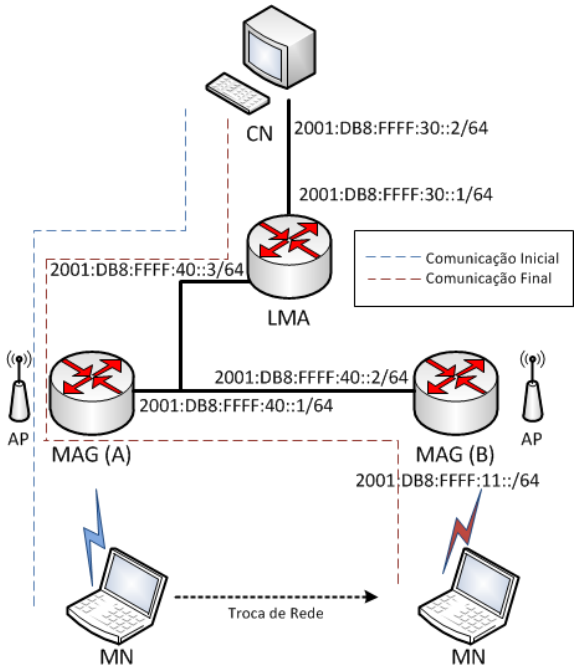

Figura 6.16: Estrutura da rede utilizada em PMIPv6.

6.2.4.1 *Resultados obtidos com PMIPv6*

- **Mensagens de controle:**

A implementação do protocolo PMIPv6 não utiliza o envio de RA para descobrir a movimentação de MN. O *Handover* é descoberto por mensagens de *Router Solicitation*, enviadas pelo dispositivo móvel em sua nova rede, ou pela análise dos *logs* enviados pelos *Access Points*. Com isto, neste teste foram coletadas apenas as mensagens de *Neighbor Solicitatation/Advertisement* entre MN e o MAG, que representam apenas 0,17 pacotes por segundo (pps) em 16.663 bytes (Figura 6.17).

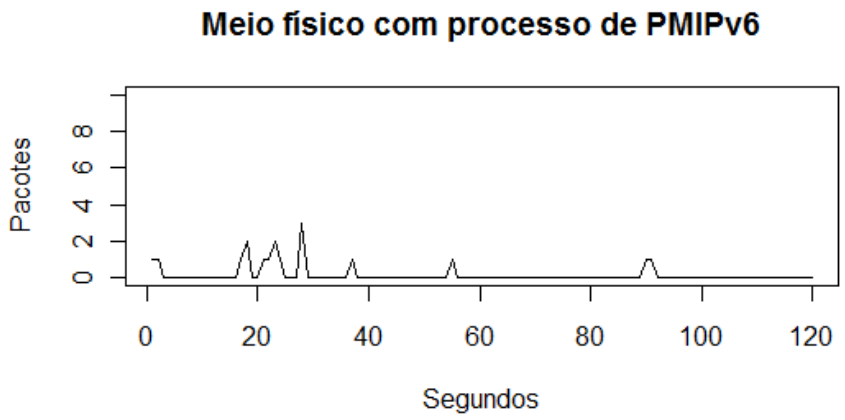

Figura 6.17: Pacotes por segundo no PMIPv6.

- **Tempo de *Handover:***

A troca de rede utilizando o protocolo PMIPv6 com envio de pacotes ICMP obteve um *Handover* total de 5,173 segundos, isto é, apenas 21 ms entre endereçamento e roteamento (Figura 6.15).

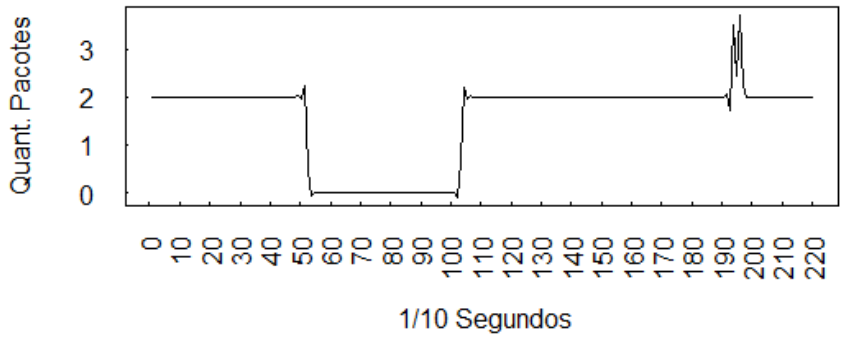

Figura 6.18: Tempo de *Handover* no PMIPv6 com envio de pacotes ICMP.

No experimento de transferência de arquivos, o tempo de *Handover* foi de 8,403 segundos, o melhor tempo dos protocolos estudados até o momento.

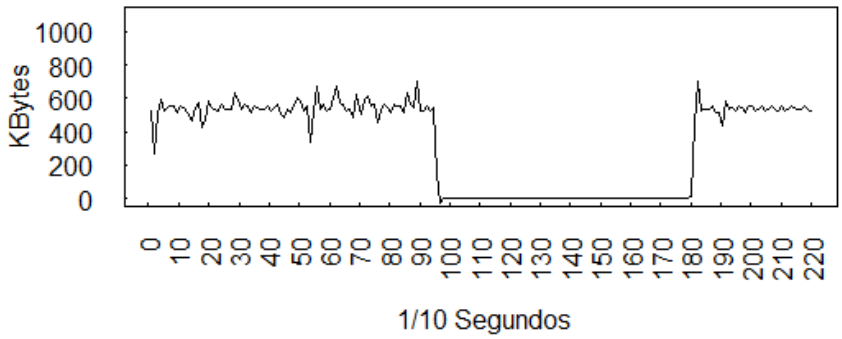

Figura 6.19: Tempo de *Handover* no PMIPv6 na transferência de arquivos,

6.2.5 Experimentos com *Host Identity Protocol*

Nos experimentos realizados com HIP foi utilizada a mesma estrutura definida no MIPv6, onde os *Access Routers* enviam mensagens de *Router Advertisement* para os *Access Points* com o intuito de fornecer o endereçamento IP para os dispositivos móveis entrantes. Ao contrário dos outros protocolos estudados até o momento, o HIP realiza um túnel fim-a-fim, entre CN e MN, independente da rede existente entre os pontos, pois utiliza o endereço HIT para estabelecer a comunicação TCP, o qual não é alterado durante a realização da troca de rede exercida pelo MN (Figura 6.20).

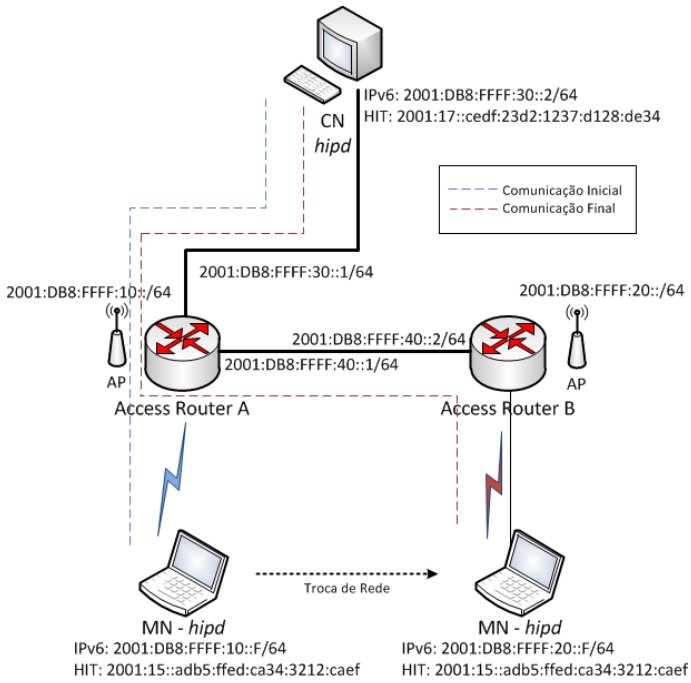

Figura 6.20: Estrutura da rede utilizada com HIP.

Para a realização dos testes com HIP foram utilizados os pacotes: hipl-daemon, hipl-dnsproxy, hipl-doc, hipl-firewall, versão 1.0.6-5193, detalhados no Apêndice A. Os procedimentos de instalação existentes em http://infrahip.hiit.fi/index.php?index=download para *Debian* não funcionaram, pois os pacotes desta versão possuem um erro em suas dependências, sendo necessário reconstruí-las para realizar a instalação.

Após a instalação foram disponibilizados os seguintes aplicativos:

hipd: responsável pela realização do *handshake*, criação dos túneis IPsec e comunicação utilizando endereços HIT;

hipl-dnsproxy: aplicativo de apoio que realiza a tradução de endereços HIT para endereços IP, utilizados na comunicação com a rede externa;

hipl-firewall: aplicativo que insere regras no *firewall* para liberar acesso entre os nodos utilizados na comunicação.

6.2.5.1 Resultados obtidos com HIP

Na implementação do protocolo HIP executado em CN e MN foi cadastrado o endereço IPv6 inicial de ambos no arquivo *hosts*, não necessitando assim a utilização do serviço de *dnsproxy*. A partir desta estrutura, foram realizados os experimentos para mensuração de tempo de *Handover* e da quantidade de mensagens de controle demonstrados a seguir.

- **Mensagens de controle:**

Inicialmente o processo *hipd* não envia nenhuma requisição à rede, ele aguarda o início de uma comunicação de um nodo correspondente para realizar o procedimento de *handshake* e estabelecimento da conexão através do endereço HIT. Com isto, conforme a Figura 6.21, a utilização do canal de comunicação pelos protocolos HIP é baixa, ocorrendo durante a realização do experimento apenas mensagens referente ao processo de *handshake* e mensagens de *neighbor discovery*.

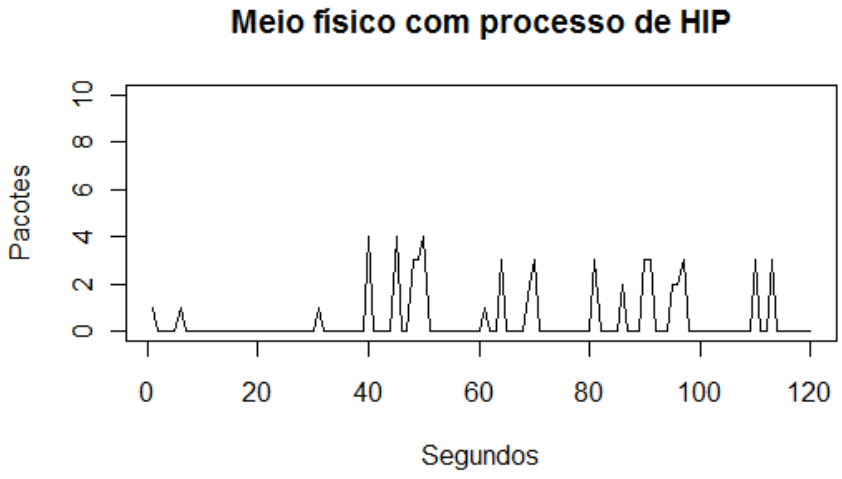

Figura 6.21: Pacotes por segundo no HIP.

- **Tempo de *Handover*:**

Na realização do *Handover* durante o envio de pacotes ICMP, com a captura sendo realizada em CN, obteve-se 5,364 segundos de *Handover*, sendo contabilizados o *Handover* físico, o endereçamento e o *handshake* do protocolo, até o estabelecimento da comunicação entre MN e CN.

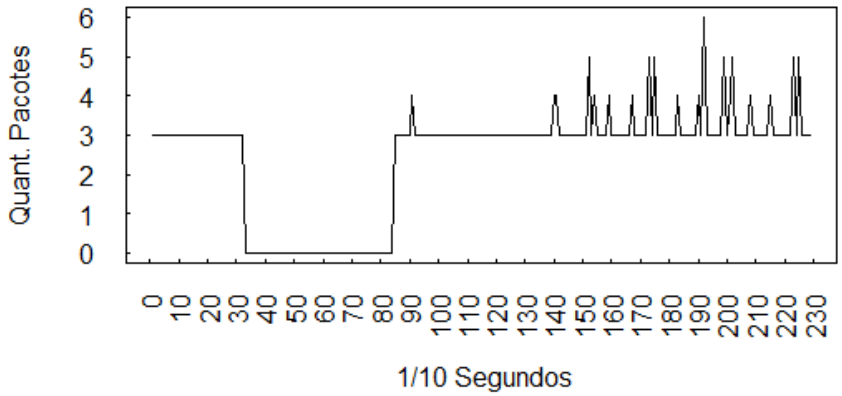

Figura 6.22: Tempo de *Handover* no HIP com envio de pacotes ICMP.

No experimento com envio de arquivo por TCP, obteve-se um tempo de 13,677 segundos de interrupção da transferência, capturado em CN, como demonstrado na Figura 6.23.

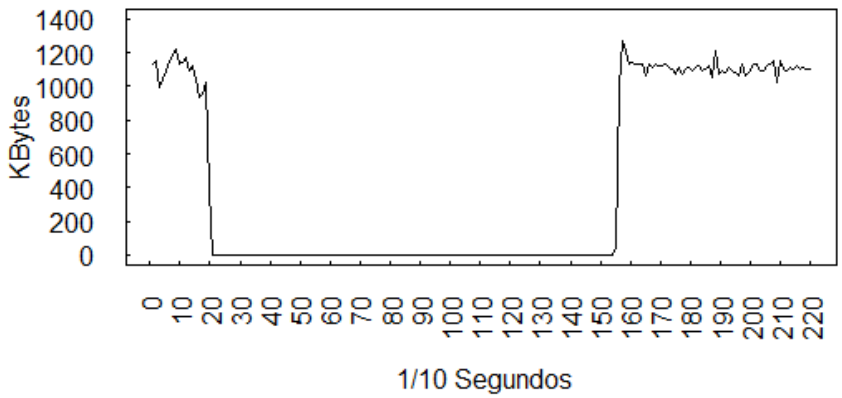

Figura 6.23: Tempo de *Handover* no HIP na transferência de arquivos.

6.2.6 Experimentos com *Locator/Identifier Separation Protocol*

Para análise do LISP foi utilizada a implementação OpenLISP versão 0.1.0 para o Sistema Operacional *FreeBSD*, pois a implementação do OpenLISP para *Linux* implementa funcionalidades apenas para IPv4. Esta implementação utiliza o encapsulamento IP-over-UDP na porta 4341, com isto todo o tráfego é realizado sobre UDP (IANNONE; SAUCEZ; BONAVENTURE., 2008).

O OpenLISP disponibiliza duas ferramentas: a primeira denominada *map*, que possibilita a criação do mapeamento entre EID e RLOC; a segunda denominada *mapstat*, que se trata de uma alteração do comando *netstat*, que demonstra as estatísticas de comunicação e os mapeamentos entre RLOCs e EIDs conhecidos pela estação em questão.

Como esta implementação não possui funcionalidades automatizadas para alteração de endereços RLOC na ocorrência da mobilidade, incluiu-se um *script* em MN que, no momento que a interface *wireless* realizou a ação de *"Down to Up"*, o referido *script* alterou o endereço RLOC de MN. Como o procedimento de *Handover* físico é executado em 5,152 segundos, neste período foi executa manualmente a inserção do mapeamento EID para o novo RLOC em CN.

A descrição da estrutura utilizada no experimento encontra-se na Figura 6.24. Para possibilitar o uso de um endereço RLOC e um endereço EID em um nó móvel, foi configurado o endereço EID na *interface Loopback* de MN e configurado os aplicativos para utilizar este endereço na geração de pacotes até CN.

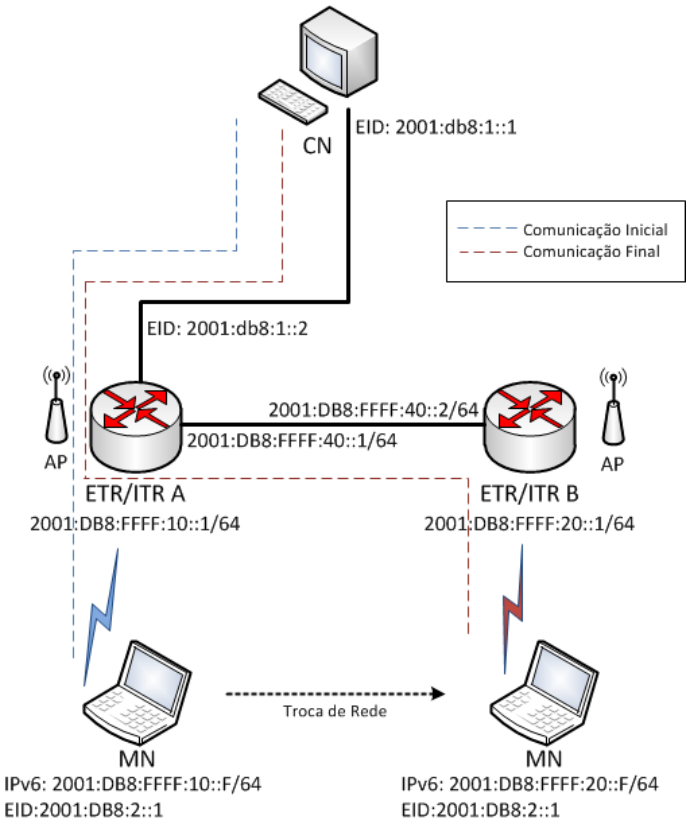

Figura 6.24: Estrutura da rede utilizada no LISP.

6.2.6.1 Resultados obtidos com LISP

Na configuração do OpenLISP verificou-se que todo o mapeamento é realizado localmente, com isto, não ocorreram trocas de mensagens realizadas pelo protocolo a serem demonstradas.

- **Tempo de *Handover:***

Durante a mensuração do tempo de *Handover* entre MN e CN, obteve-se 5,180 segundos de indisponibilidade de comunicação, isto é, 28 ms superior ao *Handover* físico de 5,152 segundos.

No experimento durante a transferência de arquivos, obteve-se um tempo de 5,186 segundos de *Handover*, 34 ms acima do tempo de *Handover* físico e com uma alta taxa de transferência, pois a implementação analisada não utiliza IPSec, ou qualquer outro protocolo de criptografia. Como ilustrado na Figura 6.25, a transferência ficou o dobro das implementações que utilizam IPSec.

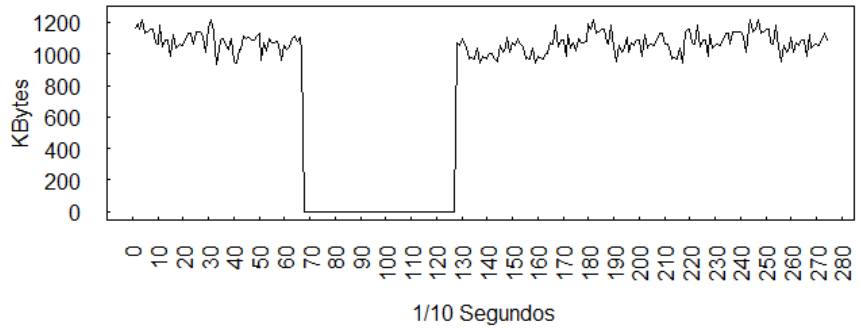

Figura 6.25: *Handover* na transferência de arquivos utilizando LISP.

6.2.7 Experimentos com *Site Multihoming by IPv6 Intermediation*

Existem três implementações para o protocolo SHIM6, a primeira desenvolvida em 2006 por K. Park et al (PARK et al., 2007) não possui uma versão atualizada. A segunda desenvolvida por J. Ahrenholz e T. Henderson (AHRENHOLZ; HENDERSON, 2008) em 2008 como uma parte do projeto do OpenHIP, não possui validação dos endereços ULID através de CGA ou HBA ou a detecção automática de perda de conexão (contextos SHIM6), com isto utilizou-se a implementação desenvolvida por S. Barré intitulada LinShim6 (BARRé; RONAN; BONAVENTURE, 2011), versão 0.9 de 2009.

Esta implementação realiza a geração dos endereços *Upper Layer Identifier* (ULID) através de *Cryptographically Generated Addresses* (CGAs) ou *Hash Based Addresses* (HBA) para criação dos pares de comunicação (contextos SHIM6). Para isto, o protocolo substitui ou acrescenta endereços IPv6 baseados no endereço de rede pré-existente na interface de rede, realizando assim a comunicação sobre estes endereços e alternando os pares de acordo com a necessidade, conforme o exemplo a seguir:

```
Information from user space daemon
_____________________-
Global state : established
local context tag : 50bf72fbfb61
peer context tag : 2d8d0f722c7e
Peer locator list :
        2001::300e:9c8f:4f34:e849
        2002::30a6:d07:ef7b:af98
Local locator list :
        2004::3c81:f00a:cd54:51ff (CGA)
Current local locator : 2004::3c81:f00a:cd54:51ff
Current peer locator : 2001::300e:9c8f:4f34:e849
Path array :
        src : 2004::3c81:f00a:cd54:51ff
        dest : 2001::300e:9c8f:4f34:e849
src : 2004::3c81:f00a:cd54:51ff
dest : 2002::30a6:d07:ef7b:af98
```

Neste experimento utilizou-se a estrutura de endereços da Figura 6.26, alternando o nó móvel entre a Rede A e a Rede B, sendo utilizada a implementação do protocolo SHIM6 no MN e no CN, o qual é composto pelos seguintes processos:

cgad - Transforma os endereços IPv6 previamente configurados em endereços CGA.

shim6d - Configura os pares de endereços disponibilizados pelos nodos.

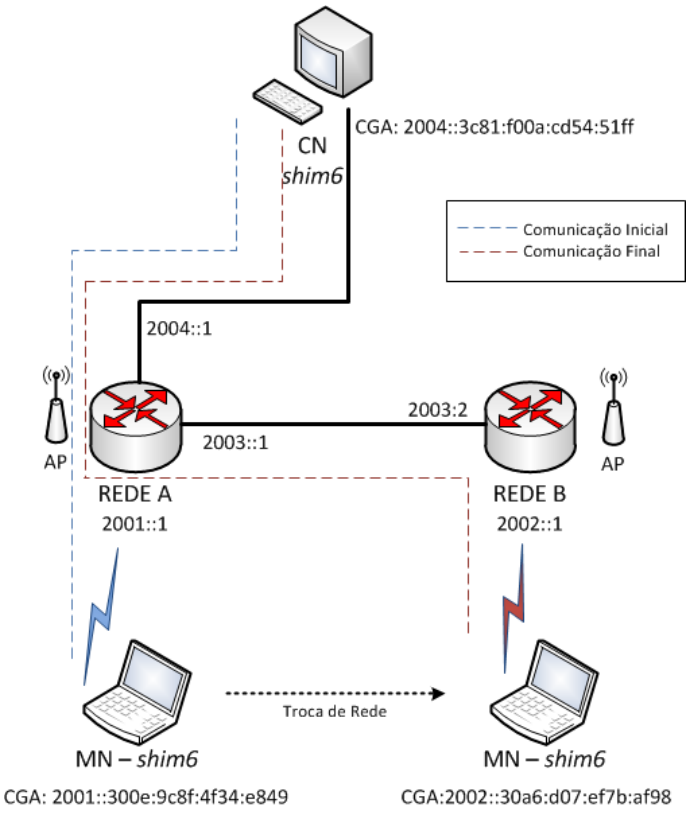

Figura 6.26: Estrutura da rede utilizada com SHIM6.

6.2.7.1 Resultados obtidos com SHIM6

- **Tempo de *Handover:***

Esta implementação não propicia a geração do endereço CGA no recebimento de um novo endereço através de *Router Advertisement*, sem a execução de processo *cgad* após o recebimento do endereço IPv6. Com isto, configurou-se previamente no *Mobile Node* dois endereços IPv6, uma para cada rede.

Nos experimentos realizados, a geração do endereço CGA empregou 3 ms, que foram adicionados ao tempo final de *Handover*.

Como o SHIM6 se trata de uma sessão entre a camada de rede e a camada de transporte (referenciando-se ao modelo OSI), não existe túneis, apenas a validação dos endereços CGA realizados durante a criação de um contexto entre os pares de nodos. Assim, quando um contexto é criado, o processo *shim6d* envia todos os endereços das interfaces de rede ao nodo correspondente para que o mesmo conheça quais as possibilidade para criação de novos contextos. A criação destes contextos, gerenciados pelo processo *shim6d*, só ocorrem após o estabelecimento de conexão entre os nodos através de TCP ou UDP, com isto, não é possível mensurar o tempo de *Handover* por ICMP.

No *Handover* por transferência de arquivos, conforme a Figura 6.27, obteve-se 15,105 segundos, pois o *shim6* apenas troca de contexto depois de enviar um *keepalive* pelo contexto atual, e, não obtendo resposta, envia um *probe* para todos os endereços conhecidos no nó correspondente até obter uma resposta e estabelecer um novo contexto.

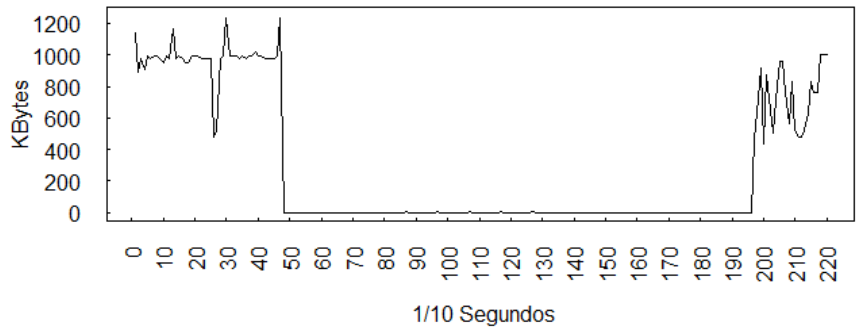

Figura 6.27: *Handover* na transferência de arquivos utilizando SHIM6.

7 ANÁLISE COMPARATIVA DOS RESULTADOS ENTRE OS PROTOCOLOS DE MOBILIDADE

Nas comparações do tempo de *Handover* durante o envio de pacotes ICMP entre os protocolos **puros** de provimento de mobilidade sobre IPv6 (Figura 7.1(a)), observa-se que a o FMIPv6 obteve o melhor resultado. Porém, este resultado é atribuído à utilização de duas interfaces de rede na realização do *Handover*, que se adicionados os 5,152 segundos, referente ao tempo de *Handover* físico, este tempo eleva-se a 7,056 segundos. Sendo assim, o PMIPv6 é o possuidor do melhor tempo total de *Handover*.

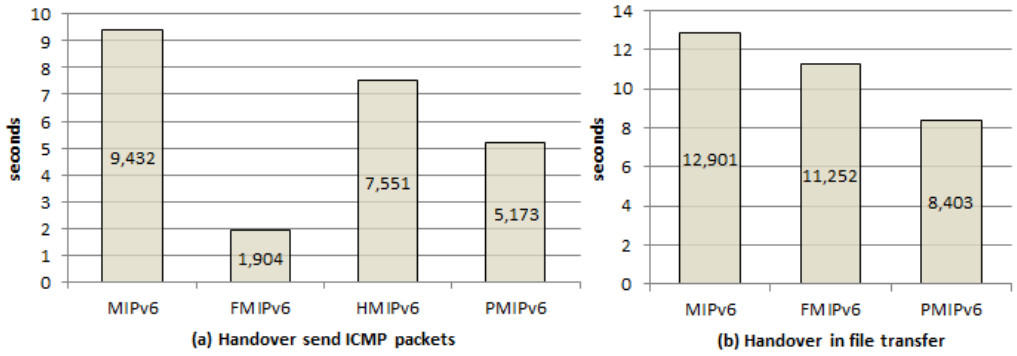

Figura 7.1: Tempos de *Handover* dos protocolos de mobilidade IPv6.

No quesito de transferência de arquivos (Figura 7.1(b)), o protocolo PMIPv6 também obteve o melhor tempo de *Handover*. Esses resultados confirmam que a estrutura de funcionamento do PMIPv6, que estabelece previamente os túneis entre o *Local Mobility Anchor* (LMA) e *os Mobile Access Gateway* (MAG), melhora em 65,13% o tempo total de *Handover* na transferência de arquivos, o que consequentemente proporciona uma melhor experiência do usuário no uso do protocolo.

Este mesmo protocolo, possui a menor quantidade de mensagens de controle, se comparado aos outros protocolos de mobilidade IPv6 **puros**, conforme mostrado na Figura 7.2.

Na análise sobre as funcionalidades das implementações estudadas, o PMIPv6 não dispõe do uso de criptografia em redes estrangeiras como as implementações de MIPv6, FMIPv6 e HMIPv6. Sua segurança esta baseada na prévia autorização de MN na rede estrangeira realizada através do cadastramento de seu endereço MAC junto ao MAG.

Na análise do tempo de *Handover* dos protocolos **híbridos**: HIP, LISP e SHIM6,

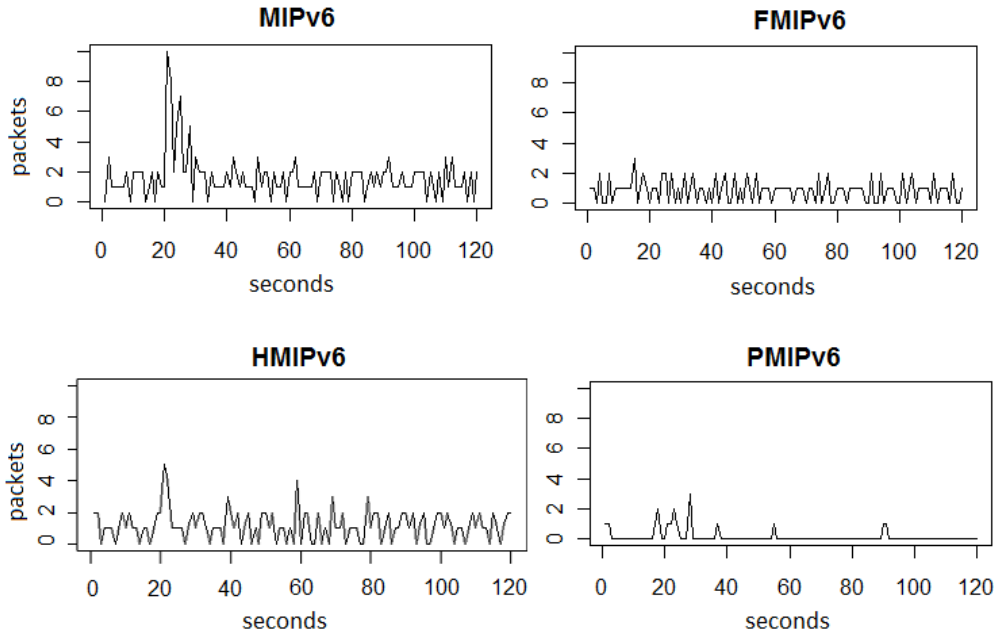

Figura 7.2: Resumo da quantidade de mensagens de controle dos protocolos de mobilidade IPv6 puros.

como demonstrado no gráfico resumo da Figura 7.3, o SHIM6, apesar de apresentar uma boa solução, onde MN possui múltiplos endereços de localização, que podem ser definidos previamente antes da realização da troca de rede, apresentou uma baixa performance de *Handover*. Ele só realiza a troca de contexto (endereçamento), após o tempo de *keepalive* definido no protocolo. Valor este definido em 15 segundos por sugestão dos desenvolvedores e que se torna instável com valores inferiores a 10 segundos. Nesse protocolo também não foi possível mensurar o tempo de *Handover* por ICMP, pois o procedimento de troca de contexto só é ativado depois de existir uma conexão TCP entre os nodos envolvidos. Uma possível utilização do SHIM6 para mobilidade foi proposta por Barré (BARRé et al., 2009b), na utilização em conjunto com o MIPv6, onde após processo de *Handover* executado pelo MIPv6, SHIM6 poderia ser utilizado para estabelecer uma conexão direta entre CN e MN, sem a necessidade de realizar um túnel pelo *Home Agent* do MN.

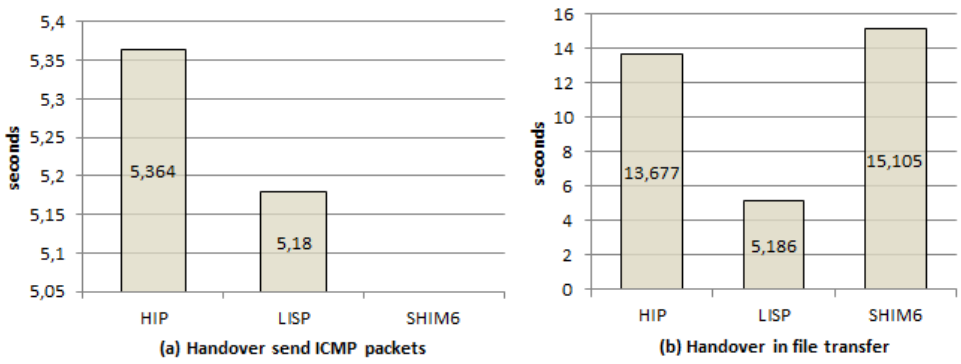

Figura 7.3: Tempos de *Handover* dos protocolos híbridos.

O protocolo LISP apresentou o melhor tempo na sua troca de rede, tanto nos experimentos utilizando ICMP, como nos experimentos de transferência de arquivos, pois utiliza um tunelamento *IP over UDP*, sem a realização de autenticações ou IPsec como nos protocolos MIPv6, FMIPv6 e HMIPv6. Contudo, é um protocolo que por definição em sua RFC, utiliza endereços inválidos para a identificação do nodo, necessitando a utilização de Proxy ou *Network Address Translation* (NAT) para o encaminhamento de pacotes sobre a Internet e, adicionalmente, uma nova estrutura de servidores para armazenar o mapeamento de endereços *(MAP Server)*, ou de uma nova tabela BGP de endereços inválidos (LISP+ALT), o que aumenta consideravelmente a quantidade de agentes na rede, dificultando sua gerência.

No protocolo HIP, o nível de segurança da comunicação são bem mais elevados se comparados com o LISP. Além de utilizar um endereço de identificação gerado através de um *hash* utilizando uma chave privada, utiliza *IPsec* para a transferência dos dados. Neste protocolo, ao ser identificada a troca da rede, é realizada uma nova autenticação entre os nodos, o que eleva o seu tempo de estabelecimento de conexão e, consequentemente, o tempo de *Handover* total. Mas se comparado aos protocolos de mobilidade sobre IPv6 **puros** que utilizam IPsec, seu tempo de *Handover* encontra-se no mesmo limiar, porém com as vantagens da validação de endereços que provê uma maior segurança.

Na análise sobre a facilidade de implantação dos protocolos de mobilidade e de seus agentes, os quais possuem seus procedimentos descritos no Apêndice A, foi verificado que nenhuma das implementações disponíveis até o momento pode ser considerada *"user friendly"*, pois nenhuma possui um *wizard* ou ambiente gráfico de configuração para seus agentes. Sendo assim, as implementações dos protocolos foram classificadas pelo número de agentes necessários a serem configurados e pela complexidade de configuração de cada um deles, utilizando para classificação da complexidade a quantidade de parâmetros e opções a serem alteradas em seus arquivos de configuração.

	Mobile Node		Home Network		Foreign Network		Correspondent Node	
Protocolo	**Agentes**	**Complexidade**	**Agentes**	**Complexidade**	**Agentes**	**Complexidade**	**Agentes**	**Complexidade**
MIPv6	1	Alta	2	Alta	1	Baixa	--	--
FMIPv6	2	Alta	3	Alta	3	Média	--	--
HMIPv6	1	Alta	2	Alta	2	Média	--	--
PMIPv6	--	--	1	Baixa	1	Baixa	--	--
HIP	1	Média	--	--	--	--	1	Média
LISP	--	--	2	Média	2	Média	--	--
SHIM6	2	Alta	--	--	--	--	2	Alta

Tabela 7.1: Complexidade de implantação dos Protocolos de Mobilidade.

Como demonstrado na Tabela 7.1, o MIPv6, FMIPv6 e HMIPv6 receberam um alto grau de complexidade devido às configurações do IPSec necessárias para seu funcionamento. Em contrapartida o PMIPv6 é simples de configurar e não necessita de configuração adicional no *Mobile Node*. Os protocolos HIP e SHIM6 possuem comportamentos parecidos, contudo o HIP possui agentes bem definidos e mais estáveis que o SHIM6. Por fim, o protocolo LISP apesar de estável, possui uma implementação que não percebe

automaticamente a ocorrência da mobilidade, não sendo indicado atualmente para este propósito.

7.1 Adoção dos protocolos IPv6 para mobilidade

A atual política de alocação de endereços IPv6 realizadas pelos RIRs oportuniza a utilização das lições aprendidas no passado, evitando a distribuição de endereços em demasia ou em blocos muito pequenos que não sejam passíveis de reagrupamento (LACNIC, 2011), isto permite um gestão organizada dos endereços IPv6.

Essa distribuição organizada de endereços e o aumento de dispositivos endereçados de maneira fim-a-fim, sem a utilização de NAT, permitem o crescimento na expectativa de vida de bens duráveis em aproximadamente um por cento, pois eles poderão ser monitorados preventivamente. Apenas esta ação representa três bilhões por ano em economia com o uso do IPv6 (COMMERCE, 2006). Com isto, empresas de tecnologia estão apoiando a expansão do uso de IPv6. Segundo a Microsoft, o IPv6 lida melhor com aplicações e serviços móveis, já a Motorola argumenta que o IPv6 oferece a oportunidade de criação de uma rede de sensores máquina-a-máquina, aumentando o uso de dispositivos conectados na Internet (COMMERCE, 2006). Este apoio das empresas também pode ser observado nas últimas versões dos sistemas operacionais mais usados por usuários finais, como: *Windows Seven, Linux 2.6 e Android 2.3*, que possuem suporte nativo ao IPv6. Contudo, nenhum destes sistemas possui o suporte à mobilidade sobre IPv6 de forma funcional. No *Linux* existe a capacidade de ativar esta funcionalidade com uma compilação de *kernel*. No *Android* existe um grupo de trabalho desenvolvendo a mobilidade sobre IPv6 (GOOGLE, 2011). Já o *Windows Seven* possui apenas a implementação para operar como um *Correspondent Node*, sem a possibilidade de realizar uma otimização de rota (HOGG, 2011).

Para ativos de rede, o fabricante *Cisco Networks* implementou os protocolos LISP e PMIPv6 para roteadores. Esses dois protocolos permitem que o processo de mobilidade ocorra somente sobre os agentes de mobilidade, sem a necessidade de alterações nos dispositivos móveis ou nos nós correspondentes (CISCO, 2011a) (CISCO, 2011b).

8 CONCLUSÃO

A implementação do MIPv6, como protocolo de provimentos de mobilidade, não obteve a inserção desejada no mercado devido à sua complexidade e alto tempo de *handover*, abrindo caminho para pesquisas de novas propostas de protocolos de provimento de mobilidade além das apresentadas neste trabalho. Neste trabalho foi feita uma análise das principais propostas a qual envolveu um levantamento bibliográfico, uma sintetização das características e funcionalidades de cada uma e um estudo comparativo. Neste estudo foram investigados e instalados alguns softwares que utilizavam as propostas de implementação de mobilidade com IPV6. A comparação envolveu uma medida do tempo de *handover* e uma avaliação da facilidade de instalar e utilizar estas soluções. O mesmo tipo de teste foi realizado com todas as soluções de mobilidade em IPv6 estudadas.

Como resultados foram classificadas as propostas de protocolos de mobilidade em IPv6 em **puros** e **híbridos**, sendo evidenciado o tempo de *handover* destes protocolos através de transferência de arquivos por TCP e de pacotes ICMP, assim como foi medida a ocupação do canal de comunicação com mensagens provenientes do controle de mobilidade. Na análise destes resultados, o PMIPv6 obteve o melhor tempo de *handover*, menor quantidade de mensagens de controle e menor complexidade da instalação de sua implementação entre os protocolos IPv6 **puros**, porém, sem a segurança advinda do IPsec implementada nos outros protocolos **puros**. Nos protocolos **híbridos**, o HIP se mostrou o protocolo mais viável a utilização, pois implementa segurança de dois níveis (de endereçamento e de comunicação), através de troca de chaves criptográficas, sem a utilização de NAT proposta pelo LISP e com um tempo de *handover* melhor que o SHIM6.

Nesta investigação foi possível comparar alternativas para implementar soluções que apoiem a mobilidade em uma rede IPv6. Todavia este estudo não esgota o assunto e em estudos futuros caberia ampliar a análise incluindo alternativas que utilizassem também outras camadas da arquitetura.

REFERÊNCIAS

ABLEY, J.; SAVOLA, P.; NEVILLE-NEIL, G. **RFC5095 - Deprecation of Type 0 Routing Headers in IPv6**. [S.l.]: The Internet Engineering Task Force -IETF, 2007.

AHRENHOLZ, J.; HENDERSON, T. **Shim6 Implementation**. Acesso em: 18 jul. 2011, openhip.org/docs/shim6.pdf.

ARANO, T. **IPv4 Address Report. Intec NetCore**. Acesso em: 21. set. 2010, http://www.potaroo.net/tools/ipv4/index.html.

BARRé, S.; DHRAIEF, A.; MONTAVONT, N.; BONAVENTURE, O. MipShim6: une approche combinée pour la mobilité et la multi-domiciliation. In: ACTES DU 14èME COLLOQUE FRANCOPHONE SUR L'INGéNIERIE DES PROTOCOLES (CFIP 09), 2009. **Anais...** [S.l.: s.n.], 2009. p.113–124.

BARRé, S.; DHRAIEF, A.; MONTAVONT, N.; BONAVENTURE, O. MipShim6: une approche combinée pour la mobilité et la multi-domiciliation. **CFIP**, [S.l.], p.113–124, October 2009.

BARRé, S.; RONAN, J.; BONAVENTURE, O. Implementation and evaluation of the Shim6 protocol in the Linux kernel. **Computer Communications**, [S.l.], 2011. http://dx.doi.org/10.1016/j.comcom.2011.03.005.

BOKOR, L.; NOVáCZKI, S.; JENEY, G. **Host Identity Protocol (HIP) Simulation Framework for INET/OMNeT++**. Acesso em: 25. mar. 2011, http://www.ict-optimix.eu/index.php/HIPSim.

BRAUN, M. B. **shim6 - IPv6 multihoming**. Acesso em: 28 mar. 2011, http://www.shim6.org/.

CHO, H.; CHOI, Y.; KWON, T.; YOU, T.; LEE, S. The Implementation of Layer-three Site Multihoming Protocol (L3SHIM). In: ADVANCED COMMUNICATION TECHNOLOGY INTERNATIONAL CONFERENCE, 2007. **Anais...** [S.l.: s.n.], 2007. v.1, p.234–237.

CISCO, I. S. **IPv6 Extension Headers Review and Considerations**. [S.l.]: Cisco Technology White Paper, 2006.

CISCO, I. S. **Cisco Carrier-Grade Services Engine-Delivering on the Future of the Internet**. Acesso em: 12 out. 2010, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/brochure_c02-560497_ns1017_Networking_Solutions_Brochure.html.

CISCO, I. S. **Cisco Locator/ID Separation Protocol and Overlay Transport Virtualization Data Center Infrastructure Solutions for Distributed Data Centers**. Acesso em: 10 abr. 2011, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/white_paper_c11-647157.html.

CISCO, I. S. **Proxy Mobile IPv6 Support for MAG Functionality**. Acesso em: 14 nov. 2011, http://www.cisco.com/en/US/docs/ios-xml/ios/mob_pmipv6/configuration/xe-3s/imo-pmipv6-mag-support-xe.html.

COMMERCE, U. D. of. **Technical and Economic Assessment of Internet Protocol version 6 (IPv6)**. [S.l.]: National Telecommunications and Information Administration, 2006.

CONTA, A.; DEERING, C.; GUPTA, M. **RFC4443 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification**. [S.l.]: The Internet Engineering Task Force -IETF, 2006.

DALEY, G. **Monash HMIPv6 Implementation**. Acesso em: 15 jun. 2011, http://www.ctie.monash.edu.au/ipv6/hmipv6.htm.

DEERING, S.; HINDEN, R. **RFC2460 - Internet Protocol, Version 6 (IPv6) Specification**. [S.l.]: The Internet Engineering Task Force -IETF, 1998.

FARINACCI, D.; FULLER, V.; MEYER, D.; LEWIS, D. **Locator/ID Separation Protocol (LISP)**. [S.l.]: draft-ietf-lisp-10.txt. The Internet Engineering Task Force -IETF, 2011.

FORD, A.; RAICIU, C.; HANDLEY, M.; BONAVENTURE, O. **STCP Extensions for Multipath Operation with Multiple Addresses**. INTERNET-DRAFT draft-ietf-mptcp-multiaddressed-03.txt.

FULLER, V.; FARINACCI, D. **LISP Map Server**. [S.l.]: draft-ietf-lisp-ms-06.txt. The Internet Engineering Task Force -IETF, 2010.

GOOGLE, A. **Android - Full IPv6 Android support**. Acesso em: 10 nov. 2011, http://code.google.com/p/android/issues/detail?id=3389.

GURTOV, A.; PATHAK, A.; KOMU, M. **Host identity protocol for Linux**. Seattle, WA, USA, 2009. v.2009.

HINDEN, R.; DEERING, S. **RFC3513 - Internet Protocol Version 6 (IPv6) Addressing Architecture**. [S.l.]: The Internet Engineering Task Force -IETF, 2003.

HOGG, S. **Windows 7 IPv6 Support**. Acesso em: 15 nov. 2011, http://www.networkworld.com/community/node/37947.

IANNONE, L.; SAUCEZ, D.; BONAVENTURE., O. **OpenLISP Implementation Report - draft-iannone-openlisp-implementation-01**. [S.l.]: The Internet Engineering Task Force -IETF, 2008.

IVOV, E.; ANDRE, M.; THERY, B.; VINCENT, S. **fmipv6.org**. Acesso em: 15 jun. 2011, http://www.fmipv6.org/.

KEENI, G.; KOIDE, K.; NAGAMI, K.; GUNDAVELLI, S. **RFC4295- Mobile IPv6 Management Information Base**. [S.l.]: The Internet Engineering Task Force -IETF, 2006.

KEMPF, J. **RFC4065 - Instructions for Seamoby and Experimental Mobility Protocol IANA Allocations**. [S.l.]: The Internet Engineering Task Force -IETF, 2005.

KENT, S. **RFC4303 - IP Encapsulating Security Payload (ESP)**. [S.l.]: The Internet Engineering Task Force -IETF, 2005.

KONG, K.; LEE, W. **Mobility management for all-IP mobile networks**: mobile ipv6 vs. proxy mobile ipv6. [S.l.]: IEEE Wireless Communications, 2008. 36-45p. v.15.

KUNTZAND, R.; TOMASCHEWSKI, M.; EBALARD, A.; CHEN, S.; LORCHAT, J. **Unix Mobile IPv6 - UMIP**. Acesso em: 10. jun. 2011, http://www.umip.org/.

LACNIC. **Manual de Políticas de LACNIC V. 1.7**. Acesso em: 15 nov. 2011, http://lacnic.net/pt/politicas/manual5.html.

LE, D.; FU, X.; HOGREFE, D. **A review of mobility support paradigms for the internet**. [S.l.]: IEEE Communications Society, 2006. 38-51p. v.8.

LE, D.; LEI, J.; FU, X. **A new decentralized mobility management service architecture for ipv6-based networks**. [S.l.]: ACM, 2007. 54-61p.

LEUNG, K.; DEVARAPALLI, V.; CHOWDHURY, K.; PATIL, B. **RFC5213 - Proxy Mobile IPv6**. [S.l.]: The Internet Engineering Task Force -IETF, 2008.

LIMONCELLI, T. A.; CERF, V. G. Successful strategies for IPv6 rollouts.: really. **Commun. ACM**, New York, NY, USA, v.54, p.44–48, April 2011.

LIU, S.; BI, J.; WANG, Y. **A Shim6-Based Dynamic Path-Selection Mechanism for Multi-homing**. [S.l.]: Evolving Internet 2009, INTERNET '09, 2009. 46-51p.

LUO, P.; HUANG, H.; SHU, W.; LI, M.; WU, M.-Y. Performance Evaluation of Vehicular DTN Routing under Realistic Mobility Models. In: WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE, 2008. WCNC 2008. IEEE, 2008. **Anais...** [S.l.: s.n.], 2008. p.2206 –2211.

MENTH, M.; KLEIN, D.; HARTMANN, M. **Improvements to LISP Mobile Node**. [S.l.]: Teletraffic Congress (ITC), 2010 22nd International. Amsterdam, 2010. 1-8p.

MORR, D. **T-Mobile is pushing IPv6. Hard.** Acesso em: 10 jul. 2011, http://www.personal.psu.edu/dvm105/blogs/ipv6/2010/06/t-mobile-is-pushing-ipv6-hard.html.

MOSKOWITZ, R.; NIKANDER, P.; JOKELA, P.; HENDERSON, T. **RFC5201 - Host Identity Protocol**. [S.l.]: The Internet Engineering Task Force -IETF, 2008.

NARTEN, T.; NORDMARK, E.; SIMPSON, W.; SOLIMAN, H. **RFC4861 - Neighbor Discovery for IP version 6 (IPv6)**. [S.l.]: The Internet Engineering Task Force -IETF, 2007.

OPENAIR3, M. C. D. **Proxy Mobile IPv6**. Acesso em: 23 jun. 2011, http://www.openairinterface.org/components/page1095.en.htm.

PéREZ-COSTA, X.; TORRENT-MORENO, M.; HARTENSTEIN, H. A performance comparison of Mobile IPv6, Hierarchical Mobile IPv6, fast handovers for Mobile IPv6 and their combination. **SIGMOBILE Mob. Comput. Commun. Rev.**, New York, NY, USA, v.7, p.5–19, October 2003.

PERKINS, C. **RFC3220 - IP Mobility Support for IPv4**. [S.l.]: The Internet Engineering Task Force -IETF, 2002.

PERKINS, C. E. Mobile IP. **IEEE Commun.**, [S.l.], May 2002.

PERKINS, C.; JOHNSON D.AND ARKKO, J. **RFC6275 - Mobility Support in IPv6**. [S.l.]: The Internet Engineering Task Force -IETF, 2011.

ROQUE, P.; FENNEBERG, L.; LUTCHANSKY, N.; SAVOLA, P.; METZ, C.; MYLLYNEN, M. **Router Advertisement Daemon for IPv6**. Acesso em: 21. jun. 2011, http://archive.ubuntu.com/ubuntu/pool/main/r/radvd/.

SILVA, C. M. da; ALMEIDA, F. M. de. **Mobilidade em Redes IP**: análise dos protocolos mipv6 e hmipv6. Acesso em: 15 jun. 2011, http://code.google.com/p/projfin-hmip/.

SILVA, C. M. da; ALMEIDA, F. M. de. **Plataforma para o Estudo de Mobilidade na Camada de Rede**. [S.l.]: Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, 2009.

VIINIKAINEN, A.; PUTTONEN, J.; SULANDER, M.; HäMäLäINEN, T.; YLöNEN, T.; SUUTARINEN, H. **Flow-based fast handover for mobile IPv6 environment - implementation and analysis**. Cited By (since 1996): 1.

VIVEK, K. **Memorandum for Chief Information Officers of Executive Departments and Agencies**. Acesso em: 11 out. 2010, http://www.ntia.doc.gov/press/2010/IPv6workshop_09282010.html.

WANG, Z.; LI, X.; YAN, B. **Fast inter-MAP handover in HMIPv6**. [S.l.]: ETCS 2009, 2009. 918-922p. v.3.

I APÊNDICE A

Neste Apêndice estão descritos os procedimentos realizados para a instalação e configuração dos protocolos de provimento de mobilidade estudados. Estes procedimentos podem ser utilizados para replicação dos experimentos realizados.

1.1 Instalação do *Mobile IPv6*

Para a instalação do protocolo *Mobile IPv6* foi utilizada a implementação MIPL existente em http://www.umip.org. Para isto, foi necessário realizar dois procedimentos distintos descritos a seguir: recompilar o *kernel* do Sistema Operacional *Linux* para habilitar as funções de mobilidade sobre IPv6 e instalar o pacote MIPL.

1.1.1 Compilação do *Kernel*

Antes de compilar o *kernel* foi necessário instalar os seguintes pacotes:

 # apt-get install kernel-package libncurses5-dev fakeroot wget bzip2

Depois, foi necessário entrar na pasta /usr/src e obter da Internet os programas fontes do *Kernel* para compilação.

 # cd /usr/src

 # wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.<versao>.tar.gz

Descompactado o *Kernel* e criado um link:

 # tar xvf linux-2.6.<versao>.tar.gz

 # ln -s /usr/src/linux-2.6.<versao> /usr/src/linux

 # cd /usr/src/linux

Salvo as configurações antigas e configurado o novo *kernel*:

 # make oldconfig

 # make menuconfig

Foi Habilitado o suporte ao MIPv6, selecionando as seguintes opções durante a execução do *make menuconfig*:

General setup

Prompt for development and/or incomplete code/drivers

System V IPC

Networking support

Networking options

Transformation user configuration interface

Transformation sub policy support

Transformation migrate database

PF_KEY sockets

PF_KEY MIGRATE

TCP/IP networking

The IPv6 protocol

IPv6: AH transformation

IPv6: ESP transformation

IPv6: IPComp transformation

IPv6: Mobility

IPv6: IPsec transport mode

IPv6: IPsec tunnel mode

IPv6: MIPv6 route optimization mode

IPv6: IPv6-in-IPv6 tunnel

IPv6: Multiple Routing Tables

IPv6: source address based routing

File systems

Pseudo filesystems

proc file system support

Salvo as configurações do *make menuconfig* e executada a compilação:

make-kpkg clean

fakeroot make-kpkg –initrd kernel_image kernel_headers

Após compilado o *Kernel* e incluindo o suporte a MIPV6, foi instalado o novo *Kernel* gerado na pasta /usr/src:

dpkg -i linux-image-2.6.<versao>.deb

dpkg -i linux-headers-2.6.<versao>.deb

Gerado o *Ram File System* para possibilitar o *boot* do Sistema Operacional:

cd /boot

mkinitramfs -k -o initrd.img-2.6.<versao> <versao>

Verificado no GRUB se o novo *Kernel* e o novo Initrd estão referenciados:

 # vi /boot/grub/menu.lst

1.1.2 Instalação do MIPL

Na instalação do MIPL, depois de recompilado o *Kernel*, foi realizado o seguinte procedimento:

 # apt-get install autoconf automake bison flex libssl-dev indent ipsec-tools radvd

 # cd /usr/src/

 # git clone git://git.umip.org/umip.git

 # cd umip/

 # autoreconf -i

 # CPPFLAGS='-isystem /usr/src/linux/include/' ./configure –enable-vt

 # make

 # make install

Para configuração dos arquivos: /usr/local/etc/mip6d.conf; /usr/local/etc/setkey.conf e /etc/radvd.conf, foram utilizadas as definições existentes em http://www.umip.org/docs/umip-mip6.html

Arquivos de configurações utilizados na implementação MIPL do *Mobile* IPv6:

Arquivo mip6d.conf (utilização do HA):

NodeConfig HA;
DebugLevel 10;
Interface "eth1";
BindingAclPolicy 2001:db8:ffff:0::1 allow;
DefaultBindingAclPolicy allow;
UseMnHaIPsec enabled;
KeyMngMobCapability disabled;
IPsecPolicySet { HomeAgentAddress 2001:db8:ffff:0::1000; HomeAddress 2001:db8:ffff:0::1/64;
IPsecPolicy Mh UseESP 10; IPsecPolicy TunnelPayload UseESP 11; }

Arquivo mip6d.conf (utilizado do MN):

NodeConfig MN;
DebugLevel 10;
OptimisticHandoff disabled;
DoRouteOptimizationMN disabled;
MnMaxHaBindingLife 60;
Interface "eth0"{
MnIfPreference 1;

Interface "wlan0"{

```
MnIfPreference 2; }
MnHomeLink "eth0"{
HomeAgentAddress 2001:db8:ffff:0::1000;
HomeAddress 2001:db8:ffff:0::1/64; }
UseMnHaIPsec enabled;
KeyMngMobCapability disabled;
IPsecPolicySet {
HomeAgentAddress 2001:db8:ffff:0::1000;
HomeAddress 2001:db8:ffff:0::1/64 ;
IPsecPolicy Mh UseESP 10;
IPsecPolicy TunnelPayload UseESP 11;
}
```

Arquivo radvd.conf (utilizado do aplicativo RADvd para envio de *router advertisement*):

```
interface eth1 { AdvSendAdvert on;
MaxRtrAdvInterval 3;
MinRtrAdvInterval 1;
AdvIntervalOpt on;
AdvHomeAgentFlag on;
AdvHomeAgentInfo on;
HomeAgentLifetime 1800;
HomeAgentPreference 10;
prefix 2001:db8:ffff::1000/64
{ AdvRouterAddr on;
AdvOnLink on;
AdvAutonomous on;
};
};
```

1.2 Instalação do *Fast Handover for Mobile IPv6*

O pacote do FMIPv6 necessitou que as configurações referente ao MIPv6, descritas anteriormente, fossem instaladas, inclusive a extensão do *Kernel* e o RAdvd. Após isto foram realizados os seguintes procedimentos:

Realizou-se o download dos fontes da aplicação:

```
# apt-get install cvs
```

```
# cvs -d:pserver:anonymous@fmipv6.cvs.sourceforge.net:/cvsroot/fmipv6 login
```

```
# cvs -z3 -d:pserver:anonymous@fmipv6.cvs.sourceforge.net:/cvsroot/fmipv6
co -P fmipv6
```

Para compilação do FMIPv6 foi necessário baixar os seguintes pacotes extras:

```
# apt-get install libiw-dev
```

```
# apt-get install libconfuse-dev
```

 # apt-get install libssl-<versao>

 # apt-get install libssl-dev

Editou-se o arquivo ./src/common.mh.c, no qual foi alterado o local correto dos arquivos .h existente nos *includes* deste arquivo. Em seguida, foi compilada a aplicação, gerando os executáveis *fmipv6-ar* e *fimipv6-mn*, através dos comandos:

 # aclocal

 # autoheader

 # autoconf

 # automake –add-missing

 # ./configure –prefix=/usr/ –enable-use-umip

 # make

 # make install

Após compilada a aplicação, foi necessário configurar os dois aplicativos.

O serviço fmipv6-ar nos dois *Access Routers* ficaram com a seguinte configuração:

Arquivo fmipv6-ar.conf:

```
Interface {
IfName = "eth4"
preference = 1
}
NIHover {
do_ni_hover = true
ap_lla = 54:e6:fc:ae:27:1c }
nap fmipar1 {
nap_lla = 54:e6:fc:ae:3a:30
nr_lla = 08:00:27:70:2b:fe
nr_addr = 2001:db8:ffff::1000/64
nr_pfx = 2001:db8:ffff::/64 }
nap fmipar2 {
nap_lla = 54:e6:fc:ae:27:1c
nr_lla = 08:00:27:6b:b4:ff
nr_addr = 2001:db8:ffff:20::1000/64
nr_pfx = 2001:db8:ffff:20::/64 }
```

No nó móvel (MN) foi utilizada a seguinte configuração para o serviço fmipv6-mn:

Arquivo fmipv6-mn.conf:

```
DetachFromTTY = false
DebugLevel = 3
PrimaryInterface {
IfName = "eth0"
}
```

```
ScanningInterface {
IfName = "wlan0"
}
LinkQuality {
UseLinkQualityTriggers = true
Threshold = -50
}
```

1.3 Instalação do *Hierarchical Mobile IPv6*

Para a implantação do HMIPv6 foram necessários dois pacotes: o RADvd para distribuição de *router advertisement* e o hmip6d, para controle da mobilidade. Utilizou-se o RADvd alterado pela Universidade de Monash, o qual inclui as mensagens de descoberta de MAP na rede, com as seguintes configurações:

```
interface eth1 { AdvSendAdvert on;
MaxRtrAdvInterval 3;
MinRtrAdvInterval 1;
AdvSourceLLAddress on;
AdvIfDown on;
FastRSResponse on;
MaxFastRS 10;
HackAdvLinkID 65535;
AdvIntervalOpt on;
AdvHomeAgentFlag off;
AdvHomeAgentInfo off;
AdvHmip6MapInfo on; # Advertise Local MAP options?
ReAdvHmip6MapInfo off; # Send Readvertised MAP options?
RecvHmip6MapInfo off; # Receive MAP options from Neighbours
HomeAgentLifetime 1800;
HomeAgentPreference 10;
map 2001:db8:ffff::1000
{
MapDistance 1;
MapPreference 7;
MapValidLifetime 80;
};
prefix 2001:db8:ffff::1000/64
{
AdvValidLifetime 200;
AdvPreferredLifetime 200;
AdvRouterAddr on;
AdvOnLink on;
AdvAutonomous on; }; };
```

Para o serviço HMIPv6, foi instalada a implementação disponível em http://code.google.com/p/projfin-hmip/, realizando os seguintes procedimentos:

 # apt-get install subversion subversion-tools subversion-helper-scripts

 # svn checkout http://projfin-hmip.googlecode.com/svn/trunk/ projfin-hmip-read-only

Foi acessado o diretório criado após a obtenção do pacote da Internet e executado:

 # ./configure

 # make

 # make install

Observação: Durante a compilação ocorrem erros em alguns fontes, necessitando a alteração dos mesmos para definir o objeto NULL, incluindo nos cabeçalhos dos arquivos a seguinte entrada:

```
# ifndef NULL
# define NULL (void *)0
# endif
```

1.4 Instalação do *Proxy Mobile IPv6*

Na implantação do PMIPv6 utilizou-se a implementação disponibilizada pelo projeto *Open Air Interface*, existente em http://www.openairinterface.org/docs/documents/PMIPv6/PMIP6D.v0.1.tar.bz2. Antes de iniciar a configuração do PMIPv6, foram instalados os seguintes pacotes:

 # apt-get install libpcap-dev indent bison flex iproute-dev

Depois de descompactado a arquivo PMIP6D.v0.1.tar.bz2, foi copiado o arquivo *match* para a pasta /etc, incluindo neste arquivo o endereços MAC do MN e sua rede /64. Este arquivo serve como uma autenticação de MN.

Instalou-se o *freeradius*:

 # cd freeradiusclient-1.1.6/

 # ./configure

 # make

 # make install

E por fim, compilou-se o *pmip6d*:

 # make

As configurações do pmip6d foram realizadas diretamente na execução do processo, sem o uso de arquivo de configuração, conforme a documentação disponível pela aplicação.

Para o experimento foram utilizadas as seguintes configurações:

LMA:

 # ./pmip6d -c -i -L 2001:db8:ffff:40::3

MAG A:

 # ./pmip6d -m -i -L 2001:db8:ffff:40::3 -N 2001:db8:ffff::10 -E 2001:db8:ffff:40::1

MAG B:

 # ./pmip6d -m -i -L 2001:db8:ffff:40::3 -N 2001:db8:ffff::11 -E 2001:db8:ffff:40::2

1.5 Instalação do *Host Identity Protocol*

Os experimentos com os protocolos HIP foram realizados com a versão 1.0.6-5193. Devido a problemas nas dependências dos pacotes para *Debian* 32 bits disponibilizados no *site* http://infrahip.hiit.fi/index.php?index=download, foi realizado o download dos pacotes .deb no endereço http://packages.infrahip.net/ubuntu/dists/lucid/main/binary-i386/, desempacotados os binários, acertadas as dependências dos arquivos e gerado novamente os pacotes para instalação.

Para instalação foi utilizado o comando:

 # dpkg -i hipl-daemon_1.0.6-5193_i386.deb hipl-doc_1.0.6-5193_i386.deb hipl-dnsproxy_1.0.6-5193_i386.deb hipl-firewall_1.0.6-5193_i386.deb

Este procedimento criou a pasta /etc/hip, a qual contém os arquivos de configuração dos pacotes.

Para realização dos experimentos, primeiramente gerou-se um endereço HIT nos dois nodos para realização da comunicação entre eles:

 # hipconf get hi default

Depois criou-se a relação entre o endereço HIT e o endereço IPv6 (o que também pode ser realizado através do aplicativo *hipdnsproxy*):

 # hipconf add map 2001:0015:a3a7:b9d3:484d:b65e:2f46:ad95

 # 2001:db8:ffff:0:4a5d:60ff:fe4a:37b3

Por último verificou-se o mapeamento, através dos comandos:

 # hipconf get ha all

Como resultado, este comando retornou o endereço HIT local e o endereço HIT do nodo remoto, que foram validados através do envio de pacotes ICMP (*ping6*). O processo de *handshake* entre os nodos somente é ativado na execução de alguma comunicação ou na troca de endereçamento proveniente da mobilidade. Importante ressaltar que o mapeamento manual é executado somente pelo fato dos nodos não estarem na mesma rede.

1.6 Instalação do *Locator/Identifier Separation Protocol*

A implantação do protocolo LISP foi realizada utilizando o OpenLISP, instalado no Sistema Operacional FreeBSD 8.1, pois foi desenvolvido para esta plataforma. Para tanto, foi recompilado o *kernel* do Sistema Operacional com a inclusão da linha a seguir no seu novo *Kernel*:

options LISP

Este procedimento foi realizado após executado o comando "install-lisp.sh"existente no pacote OpenLISP-0.1.0.tar.gz, disponível em http://www.openlisp.org/releases/ol-0.1.0-download.html.

Realizada a instalação do novo *Kernel*, compilou-se os aplicativos *map* e *mapstat*, utilizados para criação e verificação dos mapeamentos entre EID e RLOC, necessários para o funcionamento do protocolo.

 # cd /usr/src/sbin/map/

 # make depend

 # make

 # make install

 # cd /usr/src/usr.bin/mapstat/

 # make depend

 # make

 # make install

Após a instalação do *Kernel* e compilação dos aplicativos, foi necessário habilitar o roteamento IPv6 no FreeBSD, através da inclusão de duas linhas no arquivo /etc/rc.conf.

 # gateway_enable="YES"

 # ipv6_gateway_enable="YES"

Como exemplo de configuração, segue os comandos usados no MN para estabelecimento do mapeamento entre EID e RLOC, onde 2001:db8:2::1 é o endereço EID e 2001:db8:ffff:10::f é o endereço RLOC.

 # ifconfig lo0 inet6 2001:db8:2::1/64

 # ifconfig wlan0 inet6 2001:db8:ffff:10::f/64

 # map add -database -inet6 2001:db8:2::1/64 -inet6 2001:db8:ffff:10::f 1 100 1

 # map add -cache -inet6 2001:db8:1::1/64 -inet6 2001:db8:ffff:10::1 1 100 1

1.7 Instalação do *Site Multihoming by IPv6 Intermediation*

Os experimentos com o protocolo SHIM6 foram realizados com a implementação LIMSHIM6-0.9.1, disponível em http://gforge.info.ucl.ac.be/projects/shim6/. Esta implementação é composta por um *path* para o *Kernel* e por três aplicativos:

cgad - Processo responsável pela geração do *Upper Layer Identifier* (ULID), endereço IP utilizado pelo protocolo para identificação do nodo.

shim6d - Processo responsável pela controle dos pares de endereços existentes entre os nodos e pela troca de contexto entre eles.

shim6c - Aplicação que possibilita consultar os pares de endereços disponíveis entre os nodos, os contextos com conexão estabelecida e os endereços CGA existentes no nodo. Esta aplicação também possibilita alterar o tempo de *keepalive* do protocolo entre os

nodos e remover os contextos já pré-estabelecidos.

Para utilização, foi instalado o *kernel* 2.6.27, que na distribuição do Ubuntu, refere-se a versão 8.10. Os procedimentos de compilação e instalação do *kernel* são os descritos na instalação do MIPv6.

Depois de obtido o *kernel* 2.6.27, aplicou-se o *path* patch_linshim6_2.6.27_0.9.1.bz2, antes de realizar a compilação.

Na execução do *make menuconfig*, habilitou-se o SHIM6 conforme a documentação do pacote.

Para compilação dos aplicativos, executou-se na pasta do SHIM6:

```
# apt-get install libreadline-dev libssl-dev
# ./configure
# make
# make install
```

Para a execução do protocolo, configurou-se seus nodos com endereços IPv6 e executou-se o *cgad*, que realizou a geração dos endereços ULID através de um par de chaves criptográficas e pelo endereço IP /64 dos adaptadores de rede. Necessitando alterar estas chaves, utiliza-se o aplicativo *cgatool* que acompanha o pacote, sendo que os arquivos de configurações encontram-se na pasta /usr/local/etc/cgad.

A seguir encontra-se um relatório de configuração utilizada, gerado através do aplicativo *shim6c*.

```
LinShim6-0.9>cat *
Information from user space daemon
_____-
Global state : established
local context tag : 50bf72fbfb61
peer context tag : 2d8d0f722c7e
Peer locator list :
    2001::300e:9c8f:4f34:e849
    2002::30a6:d07:ef7b:af98
Local locator list :
    2004::3c81:f00a:cd54:51ff (CGA)
Current local locator : 2004::3c81:f00a:cd54:51ff
Current peer locator : 2001::300e:9c8f:4f34:e849
Path array :
    src : 2004::3c81:f00a:cd54:51ff
    dest : 2001::300e:9c8f:4f34:e849
    src : 2004::3c81:f00a:cd54:51ff
    dest : 2002::30a6:d07:ef7b:af98
```

II APÊNDICE B - ARTIGO ACEITO NO SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS - SBRC 2012

Título: Uma análise das implementações de protocolos IPv6 puros e híbridos para provimentos de mobilidade em IPv6.

Autores: César A. H. Loureiro, Liane M. R. Tarouco, Lisandro Z. Granville e Leandro M. Bertholdo.

Local de submissão: XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2012.

Uma análise das implementações de protocolos IPv6 puros e híbridos para provimento de mobilidade em IPv6

César A. H. Loureiro, Liane M. R. Tarouco, Lisandro Z. Granville, Leandro M. Bertholdo

Programa de Pós-Graduação em Computação – Instituto de Informática – Universidade Federal do Rio Grande do Sul, 91.501-970 – Porto Alegre – RS – Brasil

cahloureiro@inf.ufrgs.br, liane@penta.ufrgs.br, granville@inf.ufrgs.br, berthold@penta.ufrgs.br

Resumo. *O uso crescente de dispositivos móveis sem fio, associado ao esgotamento de endereços IPv4, nos levou a pensar que a mobilidade em redes IP será feita apenas utilizando o protocolo IPv6. No entanto, a diversidade de protocolos de mobilidade, pode ser interpretada como um indício de que as soluções de mobilidade utilizando IPv6 não estão maduras o suficiente para serem implantadas em larga escala. Neste artigo vamos analisar, testar e avaliar algumas implementações usadas no provimento de mobilidade em IPv6, analisando o desempenho, dificuldade de implantação e estabilidade destas implementações.*

Abstract. *The growing use of mobile wireless devices, associated with the depletion of IPv4 addresses, led us to think that mobility in IP networks will be done just by using the protocol IPv6. However, the diversity of mobile protocols in existence can be interpreted like a symptom that mobility´s solution using IPv6 are not mature enough to deploy on a large scale. In this paper we will review, test and evaluate several implementations used to provide mobility over IPv6, analyzing its performance, level of difficulty to deploy and stability of the solution.*

1. Introdução

A proliferação dos dispositivos móveis sem fio, associado ao esgotamento de endereços IPv4, levou-nos ao pensamento natural que a implementação da mobilidade nas redes IP ocorrerá apenas usando o protocolo IPv6. Este pensamento pode ser sentido no esforço realizado pelas operadoras de telefonia celular, em preparação para o novo padrão "4G", que em um futuro não muito distante, provavelmente terá que lidar com dispositivos *IPv6-only* [Limoncelli; Cerf, 2011] [Morr, 2010]. Além disso, IPv6 e seus protocolos de configuração de endereços (*Neighbor Discovery* e *Stateless Address Configuration)* formam uma base de protocolo apropriada para redes móveis [Perkins, 2002].

Por outro lado, existem vários protocolos propostos para lidar com a mobilidade, como: [Johnson; Perkins; Arkko, 2004] [Koodli, 2009], [Soliman; et al., 2008], [Leung; et al., 2008] [Menth; Klein; Hartmann, 2010] [Moskowitz; et al., 2008] e [Nordmark; Bagnulo, 2009]. Essa diversidade de protocolos pode ser interpretada como um indício que as soluções de mobilidade não estão maduras o suficiente para ser implantadas em redes IP, e que pesquisas ainda são necessárias.

Neste contexto, este trabalho tem por objetivo revisar, testar e avaliar as implementações mais importantes dos protocolos utilizados para prover mobilidade em IPv6, analisando seu desempenho, dificuldade de implantação e estabilidade das implementações.

Este artigo está organizado da seguinte forma: a seção (2) descreve uma classificação e uma visão geral sobre os protocolos de mobilidade avaliados, a seção (3) referencia trabalhos relacionados ao ensejo de nossa pesquisa, a seção (4) demonstra a metodologia utilizada nos experimentos, bem como uma descrição das implementações analisadas. Na seção (5) demonstramos os resultados obtidos e a análise dos mesmos. Finalmente, na seção (6) concluímos o artigo.

2. Protocolos de Mobilidade

Primeiramente, para ganhar uma melhor compreensão de conceitos de protocolos de mobilidade, é necessário obter algum *know-how* sobre a terminologia. A RFC 2002 define algumas entidades referenciadas neste artigo, incluindo: O *Mobile Node* (MN) como um *host* ou dispositivo que migra de uma rede para outra, mantendo a comunicação com um *Correspondent Node* (CN), que se refere ao par com o qual um nó móvel está se comunicando. Quando o nó móvel está trocando de rede, a partir de uma *Home Network* (HN) a uma *Foreign Network* (FN), a comunicação é controlada por um agente de mobilidade, conhecido como *Home Agent* (HA), responsável por receber e encaminhar todos os pacotes enviados entre o nó móvel e o nó correspondente. Esta comunicação, na maioria das vezes, é implementada por túneis IPSec ou *IP-over-IP*.

Existem dois tipos de protocolos de mobilidade classificados por nós, o primeiro chamamos de protocolo IPv6 "puros" e o segundo de protocolos "híbridos".

Os protocolos IPv6 "puros" são classificados dessa maneira porque utilizam apenas os recursos nativos oferecidos pelo IPv6. Entre eles vamos encontrar: *Mobile IPv6*, *Fast Handover for Mobile IPv6*, *Hierarchical Mobile IPv6* e o *Proxy Mobile IPv6*.

A segunda abordagem, que consideramos "híbridos", sugere a separação entre a identificação e a localização de um dispositivo na rede. Aqui encontramos protocolos como o *Locator / ID Separation Protocol*, *Host Identity Protocol* e *Site Multihoming by IPv6 intermediation*.

2.1. Mobile IPv6

O protocolo de mobilidade inicialmente concebido, MIPv6 [Johnson; Perkins; Arkko, 2004], permite o uso de mobilidade sem a necessidade de qualquer agente externo nas redes estrangeiras. Ao migrar para outra rede, o nó móvel (MN) solicita um endereço IPv6 na nova rede e informa ao seu *Home Agent* (HA) sobre o seu novo local. A partir deste momento, um túnel IPv6 é estabelecido entre o *Mobile Node* (MN) e seu *Home*

Agent (HA). Assim, a comunicação entre *Mobile Node* (MN) e *Correspondent Node* (CN) continua a fluir por este túnel.

A Figura 1 detalha todos os passos do processo de troca de rede, chamado de *handover*; no passo (1), o *Mobile Node* (MN) possui um endereço em sua *Home Network* e uma comunicação estabelecida com o *Correspondent Node* (CN). No passo (2), MN inicia o processo de troca de rede, movendo-se a uma *Foreign Network* (FN). Neste momento ele irá receber um novo endereço IPv6 chamado *Care-of Address* (CoA), esta designação é apenas para distinguir seus dois endereços IPv6.

Como o MN mantém seu antigo endereço, deve enviar um pacote para o seu *Home Agent* (HA) por meio da rede estrangeira, registrando o novo endereço (3) através de uma mensagem de *Binding Update*, onde o HA responde com *Binding Acknowledgement*. Neste momento, passo (4), MN atualiza seu endereço com o CN e, dependendo do suporte a mobilidade de CN, pode estabelecer a comunicação através do túnel, como podemos observar no passo (5), ou diretamente com MN, como no passo (6) [Le; Fu; Hogrefe, 2006].

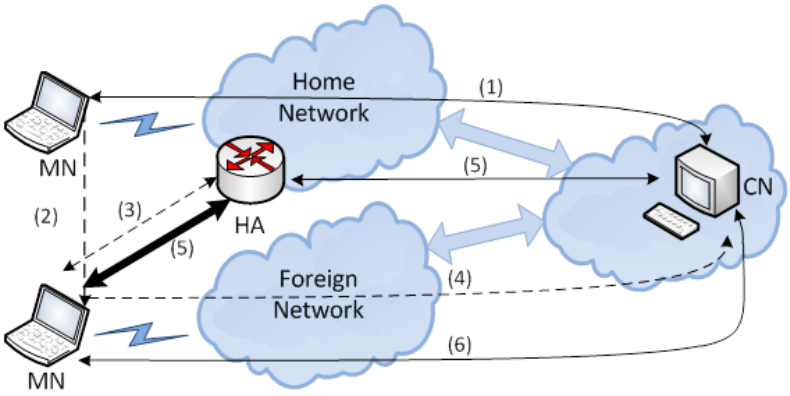

Figura 1. MIPv6 - Arquitetura e operação

2.2. Fast Handover for Mobile IPv6

O *Fast Handover for Mobile IPv6* (FMIPv6) se propõe a minimizar o tempo de configuração do túnel entre MN e HA durante a fase de troca de rede executada no MIPv6. Ele atinge seu objetivo realizando uma conexão com a nova rede, sem perder a conexão com a rede atual. Para realizar isso, o FMIPv6 utiliza informações de nível dois para sinalizar a mudança de uma rede, ou seja, quando um dispositivo móvel reconhece que o sinal do seu *Access Point* (AP) está enfraquecendo e que há um sinal mais elevado de outro AP, ele inicia o processo de conexão para a outra rede usando as novas mensagens introduzida no FMIPv6, mas ainda realizando a sua transmissão através do AP inicial, utilizando para isto duas interfaces de rede.

Neste protocolo, quando o MN realiza a negociação com o novo ponto de acesso (Figura 2), ele envia para o seu *Previous Access Router* (PAR) uma mensagem *Router Solicitation for Proxy Advertisement* (RtSolPr) (1), recebendo em troca uma mensagem *Proxy Router Advertisement* (PrRtAdv) (2), iniciando o processo de obtenção de um

endereço *stateful* ou *stateless*. Este endereço será usado para definir um novo *Care-of Address* (NCoA).

Na posse de seu novo endereço, mas ainda comunicando-se através do seu *Previous Access Router* (PAR), MN envia um *Fast Binding Update* (FBU) (3) para o PAR, solicitando o redirecionamento do tráfego através do *New Access Router* (NAR). Imediatamente PAR envia um *Handover Initiate* (HI) (4) para o NAR, informando os endereços *Previous Care-of Address (PCoA)* e *New Care-of Address* (NCoA) para validá-los.

Em resposta, o PAR envia ao NAR um *Handover Acknowledgment message* (HAck) (5), aceitando o endereço proposto ou indicando o seu novo endereço válido. Após essa negociação, o PAR envia um *Fast Binding Acknowledgment* (FBAck) (6) em retorno a mensagem *Fast Binding Update* (FBU) recebida anteriormente. Neste ponto, o nó móvel envia um *Fast Neighbor Advertisement* (FNA) (7) para o NAR, comunicando a sua presença na nova rede, permitindo que o tráfego transmitido pela rede antiga ao NAR, seja direcionado para o MN (8). Finalmente, o MN informa ao CN seu novo endereço para realizar uma possível otimização de rotas, que permite a comunicação direta, sem a necessidade de trafegar através do PAR (9) [Viinikainen; et al., 2006].

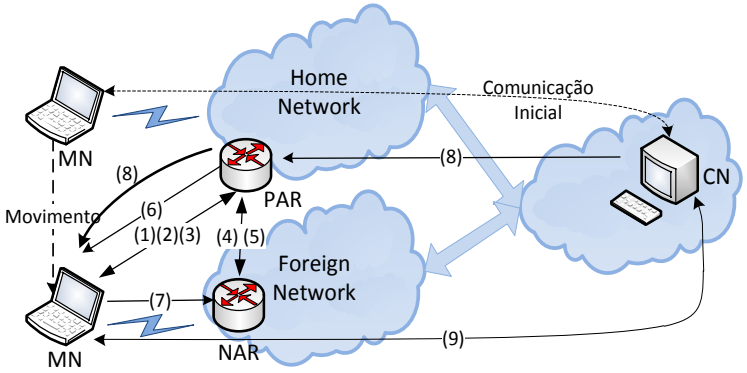

Figura 2. Configuração da conexão no FMIPv6

2.3. Hierarchical MIPv6

O HMIPv6 inclui um novo agente no processo: o *Mobility Anchor Point* (MAP). Este agente é responsável por controlar o domínio de toda a rede móvel, independentemente de quantas redes existem em cada domínio. Com isso, agora possuímos dois tipos de *handovers*: um local, dentro do mesmo domínio de rede e outro considerado externo. Esta segunda abordagem é utilizada cada vez que ocorre mobilidade entre diferentes domínios de rede.

No HMIPv6, MN sempre possui dois endereços de rede: o *Regional Care-of Address* (RCoA) e o *Local Care-of Address* (LCoA). Quando NM se conecta a alguma rede, ele recebe um *Router Advertisement* (RA) contendo os endereços dos MAPs locais. Assim, na mobilidade intradomínio evita-se que o nó móvel precise realizar um *Binding Update* para o *Home Agent*, minimizando o tempo de *handover* [Wang; Li;

Yan, 2009]. Em contraponto, o protocolo aumenta a quantidade de mensagens no *handover* externo, exigindo que o *Mobile Node* execute o processo de *Bind Update* no MAP, no HA e no CN.

Na Figura 3 demonstramos o processo de *handover* entre diferentes domínios. Na etapa (1), MN recebe um *Router Advertisement* contendo os endereços dos MAPs locais, no passo (2) MN envia um *Local Binding Update* (LBU) ao MAP e um *Binding Update* (BU) para o seu *Home Agent*, passo (3), informando seu novo *Local Care-of Address* (LCoA) e *Regional Care-of Address* (RCoA). Dessa forma, o tráfego dirigido à sua rede doméstica é encaminhado para o MAP, que encapsula os pacotes para o LCoA de MN, passo (4). Isso é necessário até que o NM informe o seu RCoA novo para o CN, possibilitando uma comunicação direta entre MN e CN, passo (5).

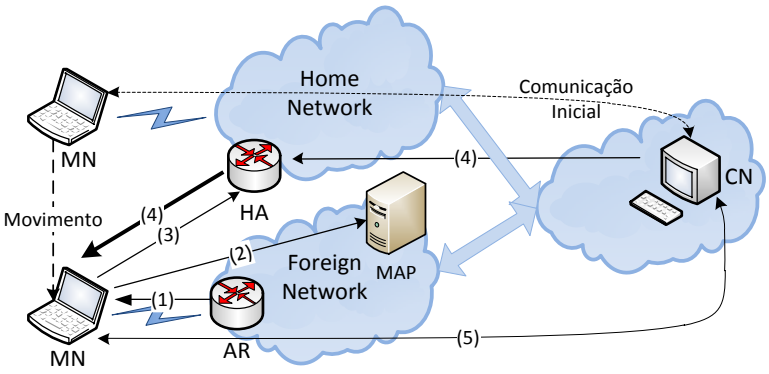

Fig. 3. Processo de configuração da conexão no HMIPv6

2.4. Proxy Mobile IPv6

Proxy Mobile IPv6 (PMIPv6) tem uma abordagem diferente, propondo criar um ponto central no controle da mobilidade. Utilizando esse conceito, o MN não precisa executar qualquer operação de troca de mensagens durante a migração de sua rede local para uma rede estrangeira. Esta responsabilidade será feita por duas novas entidades: o *Mobile Access Gateway* (MAG), localizado na rede estrangeira, e o *Local Mobility Anchor* (LMA), situado na sua rede local. Para explicar o funcionamento desta proposta, a Figura 4 a seguir, ilustra o fluxo da troca de mensagens.

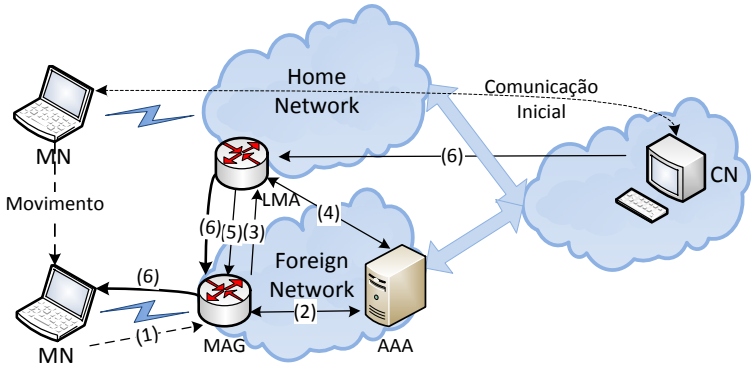

Fig. 4. Processo de configuração da conexão no PMIPv6

Na etapa (1), quando o MN acessar uma rede estrangeira, é realizado um procedimento de autenticação. No passo (2), o MAG obtém o perfil do MN em um *AAA Server (Authentication, Authorization and Accounting Server)*. Então, no passo (3), o MAG envia um *Proxy Binding Update* (PBU) para o LMA, em nome da MN.

Em (4), uma vez que a LMA recebe uma mensagem PBU e verifica as políticas de segurança, ele aceita a mensagem PBU. Em seguida, no passo (5), o LMA envia um *Proxy Binding Acknowledgment* (PBA) para MAG, incluindo o prefixo da rede do MN e atualiza a rota para a rede do MN sobre um túnel até o MAG.

Finalmente, em (6), após realizar a configuração do túnel, o MAG envia *um Router Advertisement* (RA) para o MN com as configurações de sua rede. A partir deste ponto, todas as mensagens enviadas e recebidas por MN será realizada através do MAG, utilizando o túnel até o LMA [Kong; Lee, 2008].

2.5. HIP - Host Identity Protocol

Em contraponto aos protocolos apresentados, o protocolo HIP assim como o LISP – *Locator / ID Separation Protocol* [Farinacci et al, 2011] e o SHIM6 - *Site Multihoming by IPv6 Intermediation* [Liu; Bi; Wang, 2009] propõem uma nova abordagem ao realizar a separação entre a identificação (ID) e a localização (LOC) de um nó na rede.

No HIP, o host possui um par de chaves assimétricas e, através de um *hash* da chave pública, obtém o *Host identity Tag* (HIT), utilizado para **identificar** o *host* enquanto o endereço IP é utilizado para **localizar** o *host*.

Para realizar a comunicação entre *hosts*, utilizando apenas os endereços HIT, necessita-se conhecer a localização do HIT de destino, o que pode ser obtido com o uso de um novo campo *Resource Record* (RR), incluído nos servidores de DNS. Assim, quando um nó da rede necessita descobrir o IP de *Correspondent Node* (CN), ele realiza uma consulta DNS através do identificador HIT, recebendo como resposta o IP do CN.

Quando a comunicação a ser estabelecida é entre um MN e um CN, necessita-se uma constante atualização do DNS para saber a atual localização do MN, o que resulta em falhas de comunicação devido ao tempo de convergência das informações na hierarquia de servidores DNS. No intuito de resolver esta dificuldade, foi proposta a inclusão de um novo agente, o *Rendez-vous Server* (RVS), com o objetivo de manter atualizada a informação de localização dos nós móveis que utilizam HIP. Assim, conforme a Figura 5, quando um CN realizar a requisição do endereço IP de um MN ao DNS (1), receberá como resposta o endereço de um RVS (previamente cadastrado no DNS). No momento que o CN iniciar a comunicação com um RVS, este agente verificará em sua tabela a atual localização do MN e encaminhará o pacote a ele (2), que responderá diretamente ao CN (3), pois recebeu o endereço de CN no pacote redirecionado pelo RVS [Moskowitz et al, 2008]. Após a descoberta dos endereços de localização, a comunicação ocorrerá através dos endereços HIT, já conhecidos por ambos os nodos.

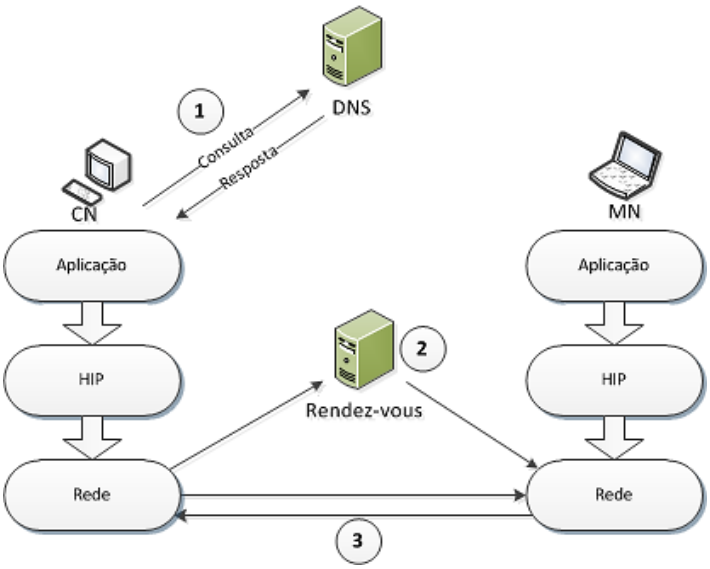

Figura 5. HIP-Comunicação entre CN e MN

2.6. LISP - Locator/ID Separation Protocol

O LISP é um protocolo que tem por objetivo prover *multihoming*, isto é, permitir que uma rede possa ser acessada independente de sua localização atual. Para isto, o LISP separa o endereçamento IP em duas partes: o *Endpoint ID* (EID), referente ao endereço IP permanente de identificação de um nó na rede e o *Routing Locator* (RLOC), um endereço IP roteável atribuído aos roteadores de borda da rede.

Basicamente, o LISP trabalha encapsulando os pacotes entre dois endereços EID através dos roteadores de borda de rede, chamados de *Ingress Tunnel Router* (ITR) e *Egress Tunnel Router* (ETR). Esses dispositivos são responsáveis por armazenar o mapeamento de endereços entre EID e RLOC. Por exemplo: quando um ponto interno da rede necessita se comunicar com um site remoto, ele realiza uma consulta DNS que retorna o endereço EID do destino, com isto, o pacote é enviado até um ITR na borda da rede utilizando um protocolo IGP (*Interior Gateway Protocol*), que encapsulará seu pacote em um novo pacote LISP, contendo o endereço RLOC de origem e o endereço RLOC do destino. Quando este pacote chegar ao ETR de destino ele será desencapsulado e encaminhado ao EID [Farinacci et al, 2011].

Com estas características, o LISP pode ser utilizado para mobilidade, pois MN pode possuir uma implementação de ITR/ETR e, na ocorrência de troca de rede, quando um MN entrar em uma nova rede, este receberá um novo endereço de localização que podemos chamar de LLOC (referente ao RLOC Local). Com isto, a mensagem a ser enviada ao CN será encapsulada em um pacote LISP contendo seu novo endereço de localização LLOC e endereçada ao endereço RLOC do destinatário. Esta mensagem quando recebida pelo ITR da borda da rede, será novamente encapsulada e encaminhada a um Proxy ETR, que realizará a primeira desencapsulação e encaminhará ao ETR, que por sua vez entregará ao CN. Quando o CN responder à mensagem ao MN, o ITR de

CN encaminhará a mensagem para o endereço LLOC de MN, encapsulada em um pacote LISP endereçado ao endereço RLOC do ETR, de onde se encontra MN. O fluxo deste processo pode ser observado na Figura 6.

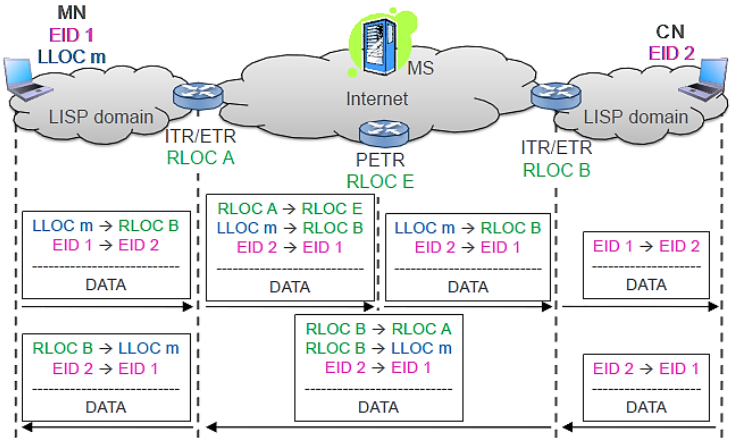

Figura 6: Comunicação entre MN e CN em redes LISP [Menth; Klein; Hartmann, 2010]

Assim, utilizando os endereços EID como origem / destino de mensagens, as camadas superiores da pilha TCP/IP não percebem a alteração de rede, garantindo a conectividade.

2.7. SHIM6 - Site Multihoming by IPv6 Intermediation

SHIM6 é uma nova subcamada entre a camada de rede (*layer*-3) e a camada de transporte (*layer*-4) que contém um ou mais endereços dos hosts de origem e destino, utilizados para servir como **localizador** e **identificador** de uma conexão.

Quando uma sessão inicia, a camada SHIM6 escolhe um par de localizadores de ambos os lados para configurar uma sessão de transmissão, utilizando os endereços ULID (*Upper Layer Identifier*) para estabelecer uma conexão. Na ocorrência de falha na conexão ou congestionamento, a camada SHIM6 fica responsável pela comutação do tráfego para um novo par localizador (contexto). Este processo ocorre sobre a camada IP e todo o processo é absolutamente transparente para aplicações das camadas superiores [Barré; Ronan; Bonaventure, 2011].

3. Trabalhos relacionados

Li e Yang [2009] comparam o *handover* do HMIPv6 e MIPv6, não explicitando a metodologia utilizada em seus experimentos. Oliveira, Cascardo e Loureiro [2003] realizam a comparação de *Bind Updates* enviados pelos referidos protocolos através de simulações. Kong e Li [2008], em sua pesquisa, comparam o tempo de *handover* dos protocolos MIPv6, HMIPv6, FMIPv6 e PMIPv6 através de simulações e análise das mensagens descritas nas especificações dos protocolos, assim como Costa, Moreno e Hartenstein [2003], que através de simulações analisaram o tempo de *handover* e a quantidade pacotes perdidos pelos protocolos FMIPv6, HMIPv6 e suas subversões,

levando em consideração nas suas simulações a escalabilidade dos protocolos em situações com até quatro *Access Points* e cinquenta *Mobile Nodes*.

Estes e outros trabalhos demonstram o anseio pela comparação entre os protocolos de mobilidade a fim de demonstrar suas funcionalidades e aplicabilidades sobre o mesmo viés. Contudo, mesmo a partir destes trabalhos não é possível classificar os referidos protocolos quanto ao tempo de *handover* e as funcionalidades implementadas devido as diferentes metodologias utilizadas nos trabalhos referenciados.

4 Metodologia

Para analisar os protocolos supracitados, utilizamos a estrutura mostrada na Figura 7, compostas de: um Pentium 4 de 2,8 GHz e 1 GB de RAM para atuar como CN; um Netbook Aton N550 com 2 GB de RAM para atuar como MN; dois *Access Points* e duas máquinas virtuais para atuarem como roteadores, auxiliando e/ou controlando a mobilidade do MN entre as redes. As máquinas virtuais utilizadas em todos os testes possuiam um único *core*, 768 MB RAM e o sistema operacional Ubuntu 10.04.2 LTS 32 bits, virtualizados sobre um computador Intel i3 de 3,1 GHz e 4 GB de RAM.

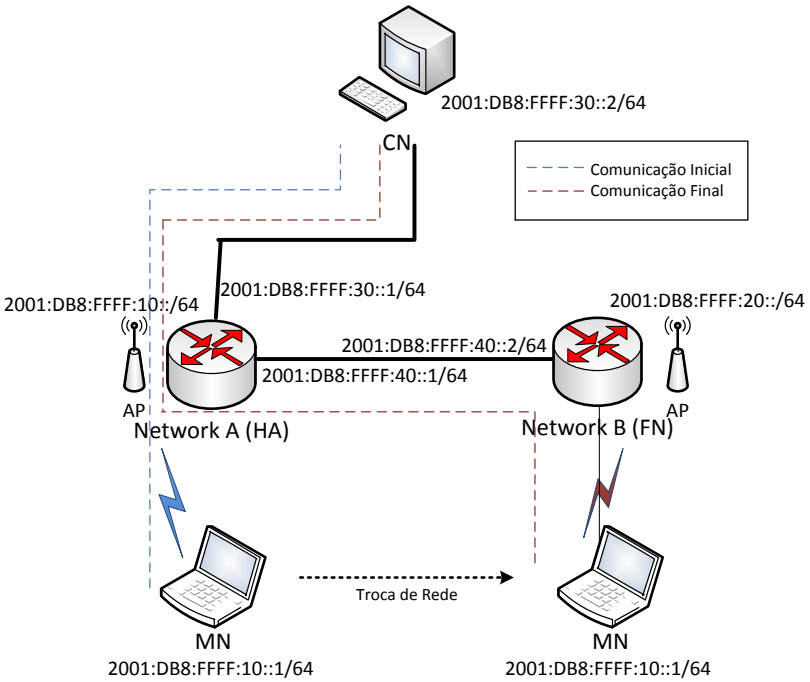

Fig. 7. Estrutura de rede utilizada nos experimentos

Nesta estrutura, analisamos as seguintes implementações:

- MIPv6: umip versão 2.0.2-0.4 [UMIP, 2011].
- FMIPv6: fmip versão 1.0-rc1 [FMIPV6, 2011].
- HMIPv6: versão 0.9.7 [Silva; Almeida, 2011] e radvd [Daley; 2011].
- PMIPv6: pmip6d [EURECOM, 2011].

- HIP: HIPL versão 1.0.6-5193. [InfraHIP, 2011].
- LISP: OpenLISP versão 0.1.0. [OpenLISP, 2011].
- SHIM6: LinShim6 [Barré; Ronan; Bonaventure, 2011].

Para analisar o tempo de *handover* foram utilizados dois métodos: o primeiro com envio de pacotes ICMP a uma taxa de um pacote por milissegundo e outro através da transferência de dados por TCP entre MN e CN.

5. Resultados

Primeiramente analisou-se o tempo de *handover* físico, ou seja, o tempo que o hardware e o sistema operacional demoram para desconectar de um *Access Point* e reconectar em outro. Este tempo é uma constante, pois é um valor independente do protocolo.

Neste experimento obteve-se um tempo de 5,152 segundos, tempo em que o nó móvel ficou desconectado de qualquer *Access Point (AP).* Todos os experimentos foram executados cinco vezes, e obtiveram um coeficiente de variação igual ou inferior a 0,02.

Nos experimentos, foi mensurado o tempo de *handover* total, isto é, o físico somado ao lógico. O *handover* lógico inclui o tempo gasto com endereçamento e estabelecimento de conectividade entre MN e CN. Na Figura 8(a), demonstramos o *handover* nos protocolos classificados como IPv6 "puros": MIPv6, FMIPv6, HMIPv6 e PMIPv6. Nestes casos foi obtido um *handover* total entre 1,904 e 9,432 segundos com o envio de pacotes ICMP. A grande diferença em favor do FMIPv6 é justificada pelo uso de duas interfaces *wireless,* necessárias para o funcionamento da implementação durante o processo de *handover* e não realmente uma melhora do *handover* lógico, pois somando o *handover* físico (5.152s), o *handover* total eleva-se em mais de três vezes (7.056s). O PMIPv6 obteve o melhor *handover, apenas* 21 milissegundos acima do *handover* físico, devido ao envio da mensagem de *Router Solicitation* executada pelo MN, que dispara o processo de mobilidade ao MAG e ao LMA, não necessitando de qualquer processamento adicional por parte de MN.

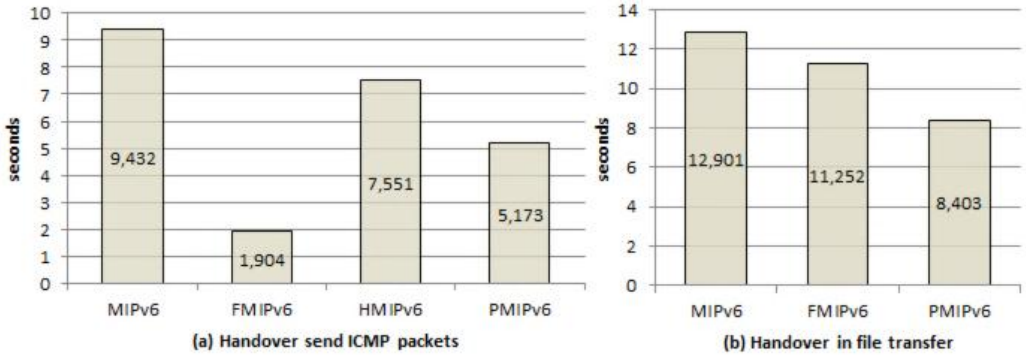

Figura 8. Tempos de *handover dos* protocolos puros

A fim de simular a utilização real dos protocolos foram realizados *handovers* durante a transferência de dados por TCP. Nestes experimentos foram obtidos *handovers* maiores devido à ocupação do canal de comunicação pela transferência de

dados, que, de acordo com a Figura 8(b), resultaram em tempos entre 8,403 e 12,901 segundos. Nestes experimentos, não foi possível mensurar a transferência de dados utilizando o protocolo HMIPv6 devido a problemas na implementação do mesmo (o túnel entre HA e CN não estabelece depois que o MN acessa a FN).

Na análise do tempo de *handover* dos protocolos híbridos (HIP, LISP e SHIM6), como demonstrado no gráfico resumo da Figura 9, o SHIM6, apesar de parecer uma boa solução, onde MN possui múltiplos endereços de localização, apresentou uma baixa performance de *handover*, pois só realiza a troca de contexto (endereçamento) após o tempo de *keepalive* definido na implementação do protocolo, valor este definido em 15 segundos por sugestão dos desenvolvedores, apresentando instabilidades com valores inferiores. Neste mesmo protocolo, não foi possível mensurar o tempo de *handover* por ICMP, já que o procedimento de troca de contexto só é ativado na pré-existência de uma conexão TCP entre os nodos envolvidos.

Entretanto, uma possível utilização do SHIM6 para mobilidade foi proposta por Barré, et al. [2009], na utilização em conjunto com o MIPv6, onde após o processo *handover* executado pelo MIPv6, SHIM6 poderia ser utilizado para estabelecer uma conexão direta entre CN e MN, sem a necessidade de criar um túnel passando pelo *Home Agent* do MN.

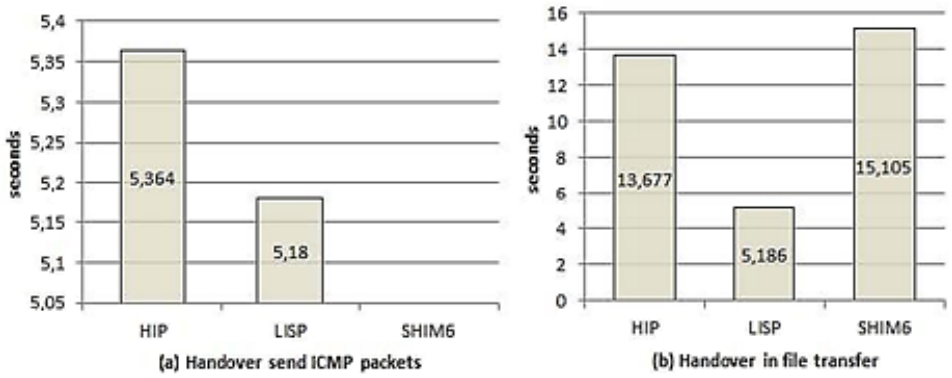

Figura 9: **Tempos de h*andover dos protocolos híbridos***

Na análise do protocolo LISP, este apresentou o melhor tempo de troca de rede, tanto nos experimentos utilizando ICMP como nos experimentos de transferência de arquivos, pois utiliza um tunelamento *IP-over-UDP*, sem a realização de autenticações ou IPsec como nos protocolos MIPv6, FMIPv6 e HMIPv6. Contudo, é um protocolo que por definição em RFC, utiliza endereços privados para a identificação de um nodo, necessitando a utilização de *Proxy* ou *Network Address Translation* (NAT) para o encaminhamento de pacotes na Internet e, adicionalmente, uma nova estrutura de servidores para armazenar o mapeamento de endereços (*MAP Server*) [Fuller; Farinacci, 2010] ou de uma nova tabela BGP formada com endereços EID e endereços RLOC (LISP-ALT) [Fuller at al., 2011], o que aumenta consideravelmente a quantidade de agentes na rede.

No protocolo HIP, o nível de segurança da comunicação são mais elevados se comparados com o LISP. Neste protocolo, ao ser identificada a troca da rede é realizada uma nova autenticação entre os nodos, o que eleva o seu tempo de estabelecimento de conexão e consequentemente o tempo de *handover* total. Entretanto, se comparado aos protocolos de mobilidade de IPv6 “puros” que utilizam IPsec, seu tempo de *handover* encontra-se no mesmo limiar, porém com as vantagens da validação de endereços, o que provê uma maior segurança.

Na análise sobre a facilidade de implantação dos protocolos de mobilidade e de seus agentes, verificamos que nenhuma das implementações disponíveis até o momento pode ser considerada "*user frendly*"', pois nenhuma possui um *wizard* ou ambiente gráfico de configuração para nenhum de seus agentes. Sendo assim, classificamos os protocolos pelo número de aplicações/agentes necessários a serem configurados e pela complexidade de configuração de cada um deles, utilizando para esta classificação a quantidade de parâmetros e opções a serem alteradas em seus arquivos de configuração.

Como demonstrado na Tabela 1, o MIPv6, FMIPv6 e HMIPv6 receberam um alto grau de complexidade devido as configurações do IPSec necessárias para seu funcionamento, em contrapartida o PMIPv6 é simples de configurar e não necessita de nenhuma configuração no *Mobile Node*. Os protocolos HIP e SHIM6 possuem comportamentos parecidos, contudo o HIP possui agentes bem definidos e mais estáveis que o SHIM6. Por fim, o protocolo LISP apesar de estável, possui uma implementação que não percebe automaticamente a ocorrência da mobilidade, não sendo indicado atualmente para este propósito.

Tabela 1. Complexidade das implementações de mobilidade analisadas

	Mobile Node		Home Network		Foreign Network		Correspondent Node	
Protocolo	**Agentes**	**Complexidade**	**Agentes**	**Complexidade**	**Agentes**	**Complexidade**	**Agentes**	**Complexidade**
MIPv6	1	Alta	2	Alta	1	Baixa	--	--
FMIPv6	2	Alta	3	Alta	3	Média	--	--
HMIPv6	1	Alta	2	Alta	2	Média	--	--
PMIPv6	--	--	1	Baixa	1	Baixa	--	--
HIP	1	Média	--	--	--	--	1	Média
LISP	--	--	2	Média	2	Média	--	--
SHIM6	2	Alta	--	--	--	--	2	Alta

6. Conclusão

Neste artigo testamos algumas propostas de provimento de mobilidade para IPv6, demonstrando a usabilidade de cada protocolo estudado. Na análise realizada por transferência de dados, não houve problemas de perda de conexão. Na análise do tempo de *handover*, FMIPv6 tem o menor tempo, no entanto, é incomum a existência de duas interfaces *wireless* em dispositivos móveis hoje em dia, o que elege a implementação do protocolo PMIPv6 como o melhor resultado a este respeito, com um tempo aceitável de

handover, baixa utilização do pacote de controle e compatibilidade com qualquer sistema operacional utilizado no *Mobile Node*, pois o *Mobile Node* não precisa realizar qualquer gestão sobre a mobilidade. Porém, este protocolo não implementa a segurança advinda do IPsec, implementada nos outros protocolos classificados como "puros".

Nos protocolos "híbridos", o HIP se mostrou o protocolo mais viável a utilização, porque implementa segurança de dois níveis (de endereçamento e de comunicação), através de troca de chaves criptográficas, sem a utilização de NAT proposta pelo LISP e com um tempo de *handover* melhor que o SHIM6. No entanto, é necessário analisar melhor o impacto da criação de túneis entre cada nó móvel e sua *home network*, como realizado por Costa, Moreno e Hartenstein [2003], pois uma grande quantidade de túneis podem gerar problemas de escalabilidade em grandes redes.

Concluímos que a mobilidade sobre IPv6 como uma solução fim-a-fim, precisa evoluir. Identificou-se que mais estudos são necessários para prover serviços utilizando mobilidade sobre IPv6 para o usuário final. Neste ponto pretendemos avançar os estudos sobre mobilidade utilizando protocolos de *layer*-2 do modelo de referência OSI, como MPLS e *OpenFlow*.

7. Referências

Barré, S., Dhraief, A., Montavont, N. and Bonaventure, O. (2009) "MipShim6: une approche combinée pour la mobilité et la multi-domiciliation". Actes du 14ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP), pp. 113-124.

Barré, S., Ronan, J. and Bonaventure, O. (2011) "Implementation and evaluation of the Shim6 protocol in the Linux kernel". Computer Communications Journal, pp. 1685-1695.

Costa, X. P., Moreno, M. T. and Hartenstein, H. (2003) "A performance comparison of Mobile IPv6, Hierarchical Mobile IPv6, fast handovers for Mobile IPv6 and their combination". Sigmobile, pp. 5-19.

Daley, G. (2011) "Hierarchical Mobile IPv6 Research at CTIE", http://www.ctie.monash.edu.au/ipv6/hmipv6.htm

EURECOM (2011) "Open Air Interface - Proxy Mobile IPv6", http://www.openairinterface.org/components/page1095.en.htm

Farinacci, D., Fuller, V., Meyer, D. and Lewis, D. (2011) "Locator/ID Separation Protocol (LISP)". draft-ietf-lisp-10.txt. IETF.

FMIPV6 (2011), "FMIPV6.org", http://www.fmipv6.org/

Fuller, V. and Farinacci, D. (2010) "LISP Map Server". draft-ietf-lisp-ms-06.txt. IETF.

Fuller, V., Farinacci, D., Meyer, D. and Lewis, D. (2011) "LISP Alternative Topology (LISP+ ALT)". draft-ietf-lisp-alt-05.txt. IETF.

InfraHIP (2011) "Infrastructure for HIP", http://infrahip.hiit.fi

Johnson, D., Perkins, C. and Arkko, J. (2004) “RFC3775 - Mobility Support in IPv6”. IETF.

Kong, K. and Lee, W. (2008) “Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6”. IEEE Wireless Communications, pp. 36-45.

Koodli, R. (2009) “RFC5268 - Mobile IPv6 Fast Handovers”. IETF.

Le, D., Fu, X. and Hogrefe, D. (2006) “A review of mobility support paradigms for the internet”. IEEE Communications Surveys & Tutorials, pp. 38-51.

Leung, K., Devarapalli, V., Chowdhury, K. and Patil, B. (2008) “RFC5213 - Proxy Mobile IPv6. IETF.

Limoncelli, T. A. and Cerf, V. G. (2011) “Successful Strategies for IPv6 Rollouts. Really”. Commun. ACM, vol. 54, no. 4, pp. 44-48.

Liu, S.; Bi, J. and Wang, Y. (2009) “A Shim6-Based Dynamic Path-Selection Mechanism for Multi-homing”. Evolving Internet 2009, pp. 46-51.

Menth, M., Klein, D. and Hartmann, M. (2010) “Improvements to LISP Mobile Node”. 22nd International Teletraffic Congress (ITC), pp.1-8.

Morr, D. (2011) “T-Mobile is pushing IPv6. Hard”, http://www.personal.psu.edu/ dvm105/blogs/ipv6/2010/06/t-mobile-is-pushing-ipv6-hard.html.

Moskowitz, R., Nikander, P., Jokela, P. and Henderson, T. (2008) “RFC5201 – Host Identity Protocol”. IETF.

Nordmark, E. and Bagnulo, M. (2009) “RFC5333 - Shim6: Level 3 Multihoming Shim Protocol for IPv6”. IETF.

Oliveira, E. R. de, Cascardo, T. L. de S. and Loureiro, A. A. F. (2003) “Análise dos Mecanismos de Gerenciamento de Mobilidade no IPv6”. XXI Simpósio Brasileiro de Redes de Computadores.

OpenLISP (2011) “The OpenLISP Project”, http://www.openlisp.org/

Perkins, Charles E. (2002) “Mobile IP”. IEEE Communications Magazine, Maio.

Silva, C. M. and Almeida, F. M. “Mobilidade em Redes IP: Análise dos Protocolos MIPv6 e HMIPv6”, http://code.google.com/ p/projfin-hmip/

Soliman, H., Castelluccia, C., ElMalki, K. and Bellier, L.“RFC5380 - Hierarchical Mobile IPv6 (HMIPv6) Mobility Management”. IETF.

UMIP (2011), “UMIP.org”, http://www.umip.org/

Viinikainen, A., Puttonen, J., Sulander, M., Hämäläinen, Ylönen, T. and Suutarinen, H. (2006) “Flow-based fast handover for mobile IPv6 environment – implementation and analysis”. Computer Communications, no. 16. vol 29. pp. 3051-3065.

Wang, Z., Li, X. and Yan, B. (2009) “Fast inter-MAP handover in HMIPv6”. First International Workshop on Education Technology and Computer Science, pp. 918-922.