UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS IPv4 EM REDES SEM FIO AD HOC

GEORGES AMVAME-NZE

ORIENTADORA:
CLAÚDIA JACY BARENCO ABBAS

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.TD - 012/06
BRASÍLIA/DF: SETEMBRO/2006

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS IPv4 EM REDES SEM FIO AD HOC

GEORGES AMVAME-NZE

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA ELÉTRICA.

APROVADA POR:

CLAUDIA JACY BARENCO ABBAS, Dra, ENE/UNB
(ORIENTADORA)

ANTONIO JOSÉ MARTINS SOARES, Dr, ENE/UNB
(EXAMINADOR INTERNO)

RICARDO STACIARINI PUTTINI, Dr, ENE/UNB
(EXAMINADOR INTERNO)

JACIR LUIZ BORDIM, Dr, CIC/UNB
(EXAMINADOR EXTERNO)

MARIO ANTONIO RIBEIRO DANTAS, Dr, INE/UFSC
(EXAMINADOR EXTERNO)

BRASÍLIA, 29 DE SETEMBRO DE 2006

FICHA CATALOGRÁFICA

AMVAME-NZE, GEORGES.
RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS IPv4 EM REDES SEM FIO ADHOC
[DISTRITO FEDERAL] 2006.
xv, 147p., 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2006).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica.

1. MIPv4 (*Mobile IP version 4*) 2. RDAIPM
3. Recuperação de Falha 4. AD HOC
I. ENE/FT/UNB II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

AMVAME-NZE, GEORGES (2006). RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS IPv4 EM REDES SEM FIO AD HOC. Tese de Doutorado, Publicação PPGENE.TD - 012/06. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 147p.

CESSÃO DE DIREITOS

AUTOR: Georges Amvame-Nze

TÍTULO: Reconfiguração Dinânmica de Agentes Móveis IPv4 em Redes sem fio AdHoc.

GRAU: Doutor ANO: 2006

_____

Georges Daniel Amvame-Nze
Brasília/DF – Brasil

DEDICATÓRIA

A Deus, meus Pais e Irmãos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por colocar pessoas que me ajudaram durante o tempo que passei desde minha graduação na UNB.

Aos meus pais e irmões, em especial minha mãe, pela força e perseverança, que me ensinaram todos os dias a lutar para crescer nesse mundo instável.

A minha orientadora e professora Claudia Jacy Barenco Abbas, pela orientação, apoio e sobre tudo por sua paciência em me guiar no mundo de redes ao longo do desenvolvimento dessa tese. Agradeço ainda, pelas oportunidades de crescimento pessoal e profissional que me proporcionou.

Aos professores do curso de Pós-Graduação em Engenharia Elétrica pela contribuição na minha formação acadêmica, especialmente aos professores Humberto Abdalla Jr. e Antonio Martins Soares pelo apoio moral.

Aos meus amigos e companheiros da UnB, em especial aos meus bons amigos, do LABCOM (Eduardo Tommy Lopez Pastor, Priscila Solis América, Roque Lambert Filho), CASA MILITAR (Honório Crispim), ANATEL (Vladimir Daigele Barbosa), NMI (Flavio Ferreira Lima) e LABREDES que foram pessoas especiais durante a conclusão da tese;

Muito Obrigado,

Georges Daniel Amvame-Nze

RESUMO

RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS IPv4 EM REDES SEM FIO AD HOC

Autor: Georges Amvame-Nze

Orientadora: Claudia Jacy Barenco Abbas

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Setembro de 2006

Este trabalho apresenta um novo protocolo chamado "Reconfiguração Dinâmica dos Agentes do IP Móvel" (RDAIPM) que permite o uso do IP Móvel (MIP) em redes sem fio ad hoc de saltos múltiplos (MANETs). O protocolo, RDAIPM, é uma extensão do protocolo IPv4 móvel, oferecendo a continuidade de sessão entre redes distintas MANET, usando para isso os seus Agentes *nativos* (HA) e *estrangeiros* (FA). Os Agentes do IP móvel são deslocados para dentro da rede MANET infraestruturada. Esses agentes garantem a continuidade de sessão, entre os dipositivos em mobilidade, enquanto estiver ativos.

O algoritmo proposto elege também os agentes passivos, que substituem os agentes ativos no caso de falha. Essa recuperação de falha é especialmente importante para o nosso protocolo, porque os agentes dispõem agora de uma mobilidade total. O que não ocorre atualmente no cenário WLAN.

Nesta tese, apresenta-se a especificação funcional detalhada do nosso protocolo (RDAIPM). Faz-se, também, uma análise de desempenho do mesmo em ambiente experimental e simulação, em nós com comunicação sem fio, baseados no sistema operacional Linux.

O RDAIPM apresentou um excelente desempenho no geral e o mesmo ocorreu durante as faces críticas: a reconfiguração dos agentes passivos no caso de recuperação de falha.

Os resultados experimentais tiveram uma ótima aproximação dos dados teóricos, confirmando o bom funcionamento do RDAIPM.

O protocolo "Reconfiguração Dinâmica dos Agentes do IP Móvel" foi submetido na lista de discussão do IETF e está sendo analisado a fim de formalizar a tese proposta.

ABSTRACT

DYNAMIC RECONFIGURATION OF MOBILE IPv4 AGENTS IN WIRELESS AD HOC NETWORK

Author: Georges Amvame-Nze

Supervisor: Claudia Jacy Barenco Abbas

Programa de Pós-graduação em Engenharia Elétrica

Brasília, September of 2006

We propose a novel protocol called "Dynamic Reconfiguration of Mobile IP Agents Protocol" (RDAIPM) which provides Mobile IPv4 (MIP) capabilities to nodes in wireless ad hoc networks (MANETs). It extends the MIP protocol and offers session continuity for distinct MANETs through the dynamic reconfiguration of its primary MIPv4 *Home* (HA) and *Foreign* (FA) *Agents*. The MIP Agents are displaced inside MANETs. These agents provide MIP signaling mechanisms to their mobile nodes so that current data session exchanges are maintained.

Our algorithm also elects passive agents which replace active agents in case of failure, thus providing redundancy and fault recovery capabilities to our solution. This fault recovery capability is especially important in our protocol as our mobile IP agents are entirely mobile in MANETs. Currently, this is not the case in a WLAN scenario.

In this thesis we provide a technical specification of our protocol (RDAIPM). We also validate our proposal and analyze its performance through simulations and experiments. The experiments were performed in Linux based wireless networks running in two different scenarios: an infra-structured WLAN and MANET networks.

Our protocol showed a good overall performance, including in its most critical part: the reconfiguration of the passive agents in case of active agent's failure. The experimental results closely resembled the theoretical analysis, thus confirming the good functioning of DRAIPM.

The Dynamic Reconfiguration of Mobile IP Agents Protocol is currently being analyzed as draft submitted to the IETF group.

SUMÁRIO

LISTA DE TABELAS xi
LISTA DE FIGURAS xii
LISTA DE FIGURAS xii
ABREVIAÇÕES xiv
1 - INTRODUÇÃO 1
1.1 - MOTIVAÇÕES 2
1.2 - OBJETIVOS 3
1.3 - ORGANIZAÇÃO DA TESE 3
2 - MOBILIDADE EM REDES IP 5
2.1 - INTRODUÇÃO 5
2.2 - MOBILIDADE NO PADRÃO IEEE 802.11 6
2.3 - MOBILIDADE NA CAMADA IP 11
2.3.1 - Funcionamento básico 11
2.3.2 - Descoberta de Agentes 13
2.3.3 - Registro 13
2.3.4 - Entrega de Datagramas 15
2.3.5 - Detecção de Mobilidade 16
2.3.6 - Consideração de Segurança 17
2.3.7 - Mensagem do tipo registration request 17
2.3.8 - Mensagem do tipo registration reply 18
2.3.9 - Flooding dos Advertisements 19
2.4 - MOBILIDADE NA CAMADA DE TRANSPORTE 20
3 - REDE MOBILE AD HOC – MANET 22
3.1 - INTRODUÇÃO 22
3.2 - PROTOCOLOS DE ROTEAMENTOS 22
3.3 - O PROTOCOLO AD HOC ON-DEMAND DISTANCE VECTOR ROUTING (AODV) 25
2.4.1 - Estabelecimento de rota 25
2.4.2 - Gerência de conectividade local 26
2.4.3 - Descoberta de rotas 27
2.4.4 - Estabelecimento da Rota Reversa 28
2.4.5 - Estabelecimento da Rota Direta 29

2.4.6 - Manutenção de rotas 30
4 - PROPOSTA DE RECONFIGURAÇÃO DINAMICA DE AGENTES MÓVEIS IPV4 EM REDES MANET 31
4.1 - INTRODUÇÃO 31
4.2 - TRABALHOS RELACIONADOS 34
4.2.1 - IP MÓVEL E MOBILE NETWORK 34
4.2.2 - HIERARCHICAL MOBILE IPV6 35
4.2.3 - MOBILE IPV4 FOR MOBILE ADHOC NETWORK 35
4.2.4 - NETWORK MOBILITY SUPPORT IN IPV6 35
4.2.5 - CELLULAR IP 36
4.2.6 - COMPARAÇÃO DE ALGUMAS DAS PROPOSTAS 36
4.3 - PROBLEMAS DE PERDA DE CONECTIVIDADE DOS AGENTES NO MIP E PROPOSTA DE SOLUÇÕES 37
4.3.1- IP MÓVEL EM REDES AD HOC 40
4.4 - PROPOSTA DE SOLUÇÃO PARA RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS 42
4.5 - DIAGRAMA DE SEQÜÊNCIA PARA CONFIGURAR OS AGENTES MÓVEIS ATIVO E PASSIVO 45
4.6 - DIAGRAMA DE SEQÜÊNCIA PARA ELEGER UM AGENTE PASSIVO 47
4.7 - DIAGRAMA DE SEQÜÊNCIA PARA DETECÇÃO DA QUEDA DO AGENTE ATIVO E MUDANÇA DO AGENTE PASSIVO PARA AGENTE ATIVO 48
4.8 - SINALIZAÇÃO DE MUDANÇA DO AGENTE ATIVO PARA AGENTE PASSIVO 49
4.9 - DIAGRAMA DE ESTADOS DE UM NÓ GENÉRICO RDAIPM 52
4.10 - IMPLEMENTAÇÃO 54
5 - TUNELAMENTO 61
5.1 - TUNELAMENTO NA REDE RDAIPM 61
5.2 - ROTA DEFAULT NA REDE RDAIPM 63
5.3 - LOOSE SOURCE ROUTING NA REDE RDAIPM 64
6 - ANÁLISE DE DESEMPENHO 67
6.1 - ANÁLISE DE DESEMPENHO NO SIMULADOR DE REDE NS2 67
6.2 - DESEMPENHO EM AMBIENTE EXPERIMENTAL 74

6.2.1 - Cenário manet....76
6.2.2 - Comparativo wlan infraestruturada & manet 82
7 - CONCLUSÕES....85
7.1 - TRABALHOS FUTUROS....87
REFERÊNCIAS BIBLIOGRÁFICAS....89
APÊNDICES 96
A - IMPLEMENTAÇÃO BÁSICA EM AMBIENTE EXPERIMENTAL....96
B - PROPOSTA DRAFT-IETF-DRMIPA-00....111
C - TEMPOS DE RECONFIGURAÇÃO DO AGENTE PASSIVO PARA ATIVO....129
D - CONTRIBUIÇÕES DA TESE PARA NS2....131

LISTA DE TABELAS

Tabela 2.1 - Padrão IEEE 802.11 [1]. 9
Tabela 3.1 - Classificação dos Protocolos de Roteamento para MANET. 24
Tabela 4.1 - Propostas relevantes dos trabalhos relacionados 37
Tabela 5.1 - Resumo dos pontos relevantes para tunelamento na rede RDAIPM 63
Tabela 5.2 - Resumo dos pontos relevantes para Rota Default na rede RDAIPM 64
Tabela 5.3 - Resumo dos pontos relevantes para Loose Source Routing na rede RDAIPM 65
Tabela 6.1 - Parâmetros de Simulação. 69
Tabela 6.2 - Comparação de desempenho das mensagens MIP e RDAIPM. 71
Tabela 6.3 – Configuração das plataformas do cenário MANET e WLAN. 75
Tabela 6.4 - Total Pacotes Pass_Agent_REQ gerados em 1 salto. 79
Tabela 6.5 - Comparativo dos tempos médios para eleição do agente passivo entre WLAN e MANET 84
Tabela A.1 - Flag A. 97

LISTA DE FIGURAS

Figura 2.1 - Topologia básica do MIP. 6
Figura 2.2 - Topologia básica da rede WLAN padrão IEEE-802.11. 6
Figura 2.3 - Atribuição de canais no padrão IEEE 802.11b. 9
Figura 2.4 - Roteamento de datagramas para e de um nó móvel numa rede estrangeira. 12
Figura 2.5 - Processo de sinalização entre MN, FA e HA. 14
Figura 2.6 - Mobilidade na Camada de Transporte. 21
Figura 3.1 - Propagação dos RREQs e estabelecimento da rota reversa entre os nós de origem e destino A e F respectivamente. 29
Figura 3.2 - Propagação dos RREPs e estabelecimento da rota direta entre os nós de origem e destino, A e F respectivamente. 30
Figura 4.1 - Visão conceitual do RDAIPM 31
Figura 4.2 - Novo MIPv4 em MANET 38
Figura 4.3 - Estrutura MIPv4 atual 41
Figura 4.4 - Novo MIPv4 em MANET com acesso a gateways 41
Figura 4.5 - Eleição de um agente móvel na rede MANET 42
Figura 4.6 - Diagrama de seqüência para configurar os agentes móveis ativos (*MApri*) e passivo (*MAsec*) 46
Figura 4.7 - Diagrama de seqüência para eleger um agente passivo 47
Figura 4.8 - Diagrama de seqüência para detecção da queda do agente ativo e atualização da lista de visitantes 48
Figura 4.9 - Sinalização da mudança do agente ativo para agente passivo 50
Figura 4.10 - Sinalização da eleição do agente passivo 51
Figura 4.11 - Diagrama de estados básico para um Mnnodelect 53
Figura 4.12 - Algoritmo para eleição do agente passivo 55
Figura 4.13 - Algoritmo para mudança de um nó passivo em ativo 56
Figura 4.14 - Mensagem MIP de agent advertisement modificada (`Type=16, A=11`) 56
Figura 4.15 - Mensagem Passive Agent Request (`Type=81, A=00`) 56
Figura 4.16 - Mensagem MNnodelect Advertisement do MA ativo para nós da rede (Type=82, A=01) 57

Figura 4.17 - Diagrama de estados básico para autoconfiguração de um MN 58
Figura 4.18 - Exemplificação de um cenário para autoconfiguração dos nós RDAIPM 60
Figura 5.1 - Exemplificação do Tunelamento na rede RDAIPM 62
Figura 6.1 - Overhead das mensagens principais do RDAIPM em bytes.....72
Figura 6.2 - Tempo para eleição do agente passivo RDAIPM (em segundos).....72
Figura 6.3 - Cenário da simulação RDAIPM com 10 e 12 nós. 73
Figura 6.4 - Cenário MANET e WLAN. 74
Figura 6.5 - Tempo de eleição de agente passivo versus intervalo de tempo para anúncio de agente ativo.....76
Figura 6.6 - Total Pacotes AODV gerados em 1 salto.....77
Figura 6.7 - Total Pacotes ADV&SOL gerados em 1 salto. 78
Figura 6.8 - Total Pacotes MN_NodeElect_Ack gerados em 1 salto. 78
Figura 6.9 - Total Pacotes MN_NodeElect_ADV gerados em 1 salto. 79
Figura 6.10 - Comparativo entre os tempos de reconfiguração esperados teóricos e práticos em 1 salto.....81
Figura 6.11 - Total Pacotes capturados na WLAN em 1 salto.....83
Figura 6.12 - Total Pacotes capturados na MANET em 1 salto. 84
Figura A.1 - Estrutura do RDAIPM Agent Advertisement 99
Figura A.2 - Valores dos octetos FLAGS e RESERVED. 100
Figura A.3 - Visualização do RDAIPM Agent Advertisement no Ethereal 102
Figura A.4 - Estrutura do Passive Agent Request.....103
Figura A.5 - Visualização do Passive Agent Request no Ethereal. 105
Figura A.6 - Estrutura do MNnodelect Advertisement.....106
Figura A.7 - Visualização do MNnodelect Advertisement no Ethereal. 107
Figura A.8 - Estrutura do MNnodelect Acknowledgement. 109
Figura A.9 - Visualização do MNnodelect Acknowledgement no Ethereal.....110
Figura C.1 - Reconfiguração dinâmica do agente passivo após desativação do agente ativo, com *agent advertisements* enviados a cada 1s. 129
Figura C.2 - Reconfiguração dinâmica do agente passivo após desativação do agente ativo, com agent advertisements enviados a cada 2s.....130
Figura C.3 - Reconfiguração dinâmica do agente passivo após desativação do agente ativo, usando agent advertisements enviados a cada 6s. 130

ABREVIAÇÕES

AP *Access Point* — Ponto de acesso.

AODV *Adhoc On-Demand Distance Vector routing protocol* — Protocolo de roteamento Vector de Distância Adhoc Sobre Demanda

CCA *Clear Channel Assessment* — Sinal de Canal Livre

CN *Correspondent Node* — Nó correspondente

CoA *Care-of Address* — Endereço residente

CSMA/CA *Carrier Sense Multiple Access with Collision Avoidance*

CTS *Clear To Send* — Liberado para transmitir

CW *Contention Window* — Janela de Contenção

DCF *Distributed Coordination Function* — Função de Coordenação Distribuída

DHCP *Dynamic Host Configuration Protocol* — Protocolo de configuração dinâmica de Servidor

DRMIPA *Dynamic Reconfiguration of Mobile IP Agents* – Reconfiguração Dinâmica de Agentes IP Móveis

DSSS *Direct Spread Sequence Spectrum* — Espalhamento Espectral de Seqüência Direta

FA *Foreign Agent* — Agente de mobilidade estrangeiro

FN *Foreign Network* – Rede estrangeira

FHSS *Frequency Hopping Spread Spectrum* — Espalhamento Espectral por salto em Freqüência

GPL *GNU general public license* — Licença pública geral GNU

HA *Home Agent* — Agente de mobilidade nativo

HN *Home Network* – Rede nativa

ICMP *Internet Control Message protocol* — Protocolo de mensagem de controle da Internet

IEEE *Institute of Electrical and Electronics Engineers* - Instituto de Engenharia Elétrica e Eletrônica

IETF *Internet Engineering Task Force* — Força-tarefa de engenharia da Internet

IP *Internet Protocol* — Protocolo de Internet

ISI *Inter Symbol Interference* — Interferência Intersimbólica

ISM	*Industrial Scientific Medical* — Industrial, Científico, Médico
LAN	*Local Area network* — Rede de área local
MAN	*Metropolitan Area Network* — Rede de área Metropolitana
MFA	*Mobile Foreign Agent* – Agente de mobilidade móvel estrangeiro
MHA	*Mobile Home Agent* – Agente de mobilidade móvel nativo
MIPv4	*Mobile IPv4* — IPv4 Móvel
MN	*Mobile Node* — Nó móvel
NAT	*Network Address Translation* — Tradução de endereços de rede
NAV	*Network Allocation Vector* — Vetor de Alocação ao meio
NS-2	*Network Simulator 2* — Simulador de rede 2
OFDM	*Orthogonal Frequency Division Multiplexing* — Multiplexação por Divisão de Freqüência
OLSR	*Optimized Link State Routing* – Roteamento de estado de link otimizado
PCF	*Point Coordenation Function* — Função de Coordenação Pontual
RDAIPM	Reconfiguração Dinâmica de Agentes IP Móveis
RFC	*Request For Comments* — Chamada para comentários
RTS	*Request To Send* — Chamada para transmitir
SSID	*Service Set Identifier* — Conjunto Identificador de Serviço
TBRPF	*Topology Broadcast based on Reverse-Path Forwarding*
TCP	*Transmission Control Protocol* — Protocolo de controle de transmissão
TORA	*Temporally-Ordered Routing Algorithm* — Algoritmo de Roteamento Ordenado Temporariamente
UDP	*User Datagram Protocol* — Protocolo de datagrama de usuário
VLAN	*Virtual Local Area Network* — Rede de área local Virtual
WAN	*Wide Area Network* — Redes de área ampla
WLAN	*Wireless LAN* — Rede LAN sem fio
WMAN	*Wireless MAN* — Rede Metropolitana sem fio
Wi-Fi	*Wireless Fidelity* — Fidelidade sem fio

1 - INTRODUÇÃO

A demanda por computação e conectividade móvel vem crescendo significativamente nos últimos anos, em virtude da massificação das redes celulares e evolução dos equipamentos envolvidos, com queda dos custos de componentes e capacidade de processamento cada vez maior. A demanda por conectividade móvel em ambiente sem fio (*wireless*) [1] também fez surgir o conceito de redes Ad Hoc, de características temporárias e autoconfiguráveis, onde quaisquer dos nós dentro do alcance de transmissão de suas interfaces wireless podem se comunicar diretamente, sem a dependência de uma infraestrutura preestabelecida, como através de estações-base (*BS*) ou pontos de acesso (*AP*) fixos.

Nesse conceito, as conexões por saltos simples (single-hop), típicas das redes WLAN infraestuturadas, onde cada nó se comunica com uma estação-base central no mesmo enlace de conexão evoluem para conexões de saltos múltiplos (multi-hop), onde os nós adjacentes são utilizados para realizar o roteamento desde o nó de origem ao nó de destino quando estes não possuem conectividade direta.

Essas redes, de saltos múltiplos, têm grande flexibilidade e rapidez na sua implantação, sendo adequadas para muitas aplicações, como a comunicação em ambientes hostis, como os encontrados em campos de batalha ou situações de calamidade. Mas, é difícil ter a reconfiguração dinâmica em todas as camadas, devido à mobilidade, obtendo-se apenas um novo endereço IP via protocolo DHCP [76]. Isto porque, o mundo da Internet não estava preparado para um ambiente com mobilidade. Um nó tendo efetuado uma conexão TCP e querendo mantê-la enquanto for se deslocar para outra sub-rede, sem mudar o seu endereço de origem, tem sua conexão interrompida devido ao rompimento da sessão em curso:

```
<IP_Origem,Porta_Origem && IP_Destino,Porta_Destino>
```

Devido a esses problemas de mobilidade foi desenvolvido o protocolo da camada de rede o MIPv4, que é uma solução para o problema de transferência de informação entre nós fixos/móveis e nós em mobilidade. A camada física e de enlace são transparentes ao IP Móvel fazendo com que o nó móvel possa se comunicar com

outros nós sem alterar o seu endereço de origem, mantendo a conexão com suas aplicações e comunicações em andamento.

Esta tese apresenta uma nova proposta para a utilização do protocolo IP móvel em redes de comunicação do tipo Ad Hoc, em que é admitida a mobilidade de todos os nós, inclusive dos próprios Agentes de Mobilidade: HA e/ou FA, que são reconfigurados dinamicamente dentro da rede de acordo com a necessidade administrativa. Para isso, serão discutidos os problemas decorrentes dessa mobilidade e propostos algoritmos para a reconfiguração automática dos Agentes, dentro de um ambiente de micro-mobilidade, de modo a garantir a continuidade dos serviços associados ao MIP para todos os nós que deles se utilizam dentro do alcance da rede MANET.

1.1 - MOTIVAÇÕES

As implementações atuais do MIPv4 em redes MANET não suportam o uso de agentes móveis HA e FA para suprir as mesmas funcionalidades quando implementadas em redes infraestuturadas.

Até então, não existe uma solução quanto ao que poderia acontecer em caso de falha e/ou indisponibilidade dos recursos do agente de mobilidade no decorrer de suas transações entre os nós da rede MANET da qual ele faz parte e daqueles que forem visitantes.

Nos trabalhos relacionados em [6] e [10], foram propostas várias abordagens visando o estudo de recuperação de falha dos agentes de mobilidade. O trabalho [6], em particular, trata justamente da redundância no uso de agentes de mobilidade, mas na rede infraestuturada e caso um deles tenha alterado o seu funcionamento durante o tempo de atividade. O problema é que esses agentes cooperam entre si de maneira fixa, i.e. sem mobilidade nenhuma. Em [10] é abordado o mesmo assunto fazendo uso de vários agentes de mobilidade servindo de apoio ao agente principal caso tiver tido uma falha, também em redes infraestruturadas.

1.2 - OBJETIVOS

A presente tese tem por objetivo propor uma reconfiguração dinâmica dos agentes de mobilidade utilizando como suporte algum protocolo de roteamento em redes MANET.

A proposta também visa uma nova abordagem no que diz respeito ao uso de agentes de mobilidades dentro de uma rede MANET. Ao contrário das atuais implementações, em que esses agentes são estáticos e atuam em redes infraestuturadas, mas tendo pelo menos o MIP e o AODV para atender os nós MIP que estiverem dentro da rede MANET [45], [47].

Finalmente, será feita uma implementação das extensões do MIP e AODV em um ambiente real atendendo a proposta de reconfiguração dinâmica dos agentes de mobilidades.

Com o intuito de analisar o correto funcionamento da implementação a ser realizada, além da análise de desempenho, será utilizada a ferramenta NS2 de simulação de rede [19].

1.3 - ORGANIZAÇÃO DA TESE

O trabalho de pesquisa está organizado em sete capítulos e três apêndices, cujo conteúdo está descrito a seguir.

O Capítulo 2 mostra as possíveis soluções de mobilidade referentes às camadas de enlace, rede e transporte, realçando o uso do protocolo de rede MIP.

O Capítulo 3 apresenta brevemente o funcionamento de um dos principais protocolos de roteamento reativos da rede MANET, o AODV.

O Capítulo 4 começa com uma breve explicação dos problemas encontrados na implementação do MIP junto à rede MANET e suas implicações. A seguir, é apresentada uma discussão dos trabalhos relacionados na literatura com respeito ao RDAIPM, e uma contextualização da tese aqui proposta e uma proposta inicial de

mecanismo de Reconfiguração Dinâmica de Agentes Mobile IP em redes MANET (ou RDAIPM).

O Capítulo 5 aborda o mecanismo de tunelamento entre os nós RDAIPM num cenário MANET. Várias abordagens são analisadas, assim será possível propor o melhor mecanismo para auxiliar o RDAIPM de maneira a manter as propriedades de mobilide do MIPv4.

O Capítulo 6 trata da análise de desempenho do RDAIPM. O protocolo foi implementado primeiramente no simulador de rede NS2 e, em seguida, num ambiente de teste experimental utilizando o sistema operacional LINUX.

O Capítulo 7 apresenta as conclusões da tese proposta, discussão dos resultados obtidos e recomendações julgadas pertinentes para continuidade da pesquisa.

Os Apêndices contêm os cabeçalhos das principais mensagens do protocolo RDAIPM [A] usadas nesta tese, a proposta de um draft [B] submetido à análise do grupo MIPv4 do IETF, os tempos de raconfiguração [C] de um *agente passivo* previamente eleito para atuar como *agente ativo* e as contribuições da tese para o NS2 [D].

2 - MOBILIDADE EM REDES IP

2.1 - INTRODUÇÃO

O Mobile IP é um protocolo padrão que visa conectividade à nível de camada IP [2] independentemente da localização física do dispositivo móvel [3],[4],[5]. Um nó em *roaming* pode ter acesso a Internet pelo encapsulamento de seus dados via protocolo IP, sem, no entanto, mudar o endereço IP de sua rede de origem, e deixando a mobilidade transparente para a aplicação e protocolos da camada de transporte, TCP e UDP.

Sem o MIP, vários usuários de nós móveis tinham de se satisfazerem unicamente com a portabilidade do equipamento, i.e, sem ter mobilidade nenhuma. Isso porque quando um nó muda de uma rede para outra, as transações de dados aos quais ele tinha acesso são interrompidas e devem ser reinicializadas novamente na rede que ele for visitar. Com o advento do MIP, um nó pode quase que ininterrupta e continuamente ter conectividade com as aplicações a ele providas entre redes conservando o seu endereço IP de origem.

O principal problema técnico que tem de ser tratado para o suporte a mobilidade é a maneira pela qual o endereçamento IP tem de ser usado. O tráfego unicast Internet é roteado para a localidade especificada no campo de endereço destino do cabeçalho IP. O endereço IP de destino especifica o endereço de rede e, portanto, o tráfego é encaminhado para essa rede. Isso não é apropriado para um nó móvel que quer manter o seu endereço IP de origem independentemente da sua localização atual e sem que o nó da rede Internet tenha que saber a sua localização presente para possível envio de tráfego até ele.

O MIP resolveu o problema fazendo com que o nó móvel possa usar dois endereços IP, sendo eles o endereço do HA e *home address* que serão vistos em detalhe posteriormente.

Neste capítulo será feita uma breve descrição de como o Protocolo MIP na sua versão 4 funciona. A Figura 2.1 mostra a topologia básica do MIP representada por quatro entidades principais sendo elas o HA, FA, MN e CN. Serão descritos conceitos e

funcionalidades básicas, assim como os detalhes específicos que são importantes nesta tese.

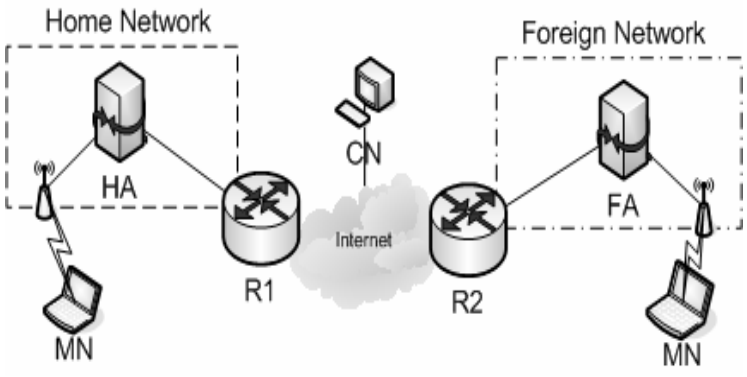


Figura 2.1 - Topologia básica do MIP.

2.2 - MOBILIDADE NO PADRÃO IEEE 802.11

Quando se usa a mobilidade no padrão IEEE 802.11, para redes WLAN infraestruturada, o ponto de conexão do dispositivo para a rede Internet permanece o mesmo durante a sua mobilidade, i.e, um nó móvel se deslocando dentro de sua rede infraestuturada, com vários pontos de conexão fazendo uso dos mesmos parâmetros de configuração, pode ter os seus dados recebidos naquela nova localização [37]. Esta mobilidade visa aumentar o raio de ação de um nó dentro de determinada área administrativa, Figura 2.2.

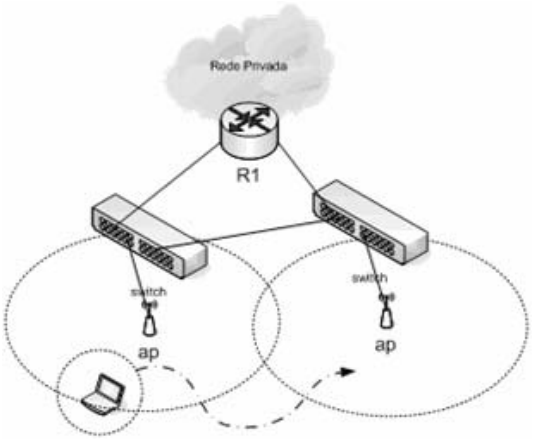


Figura 2.2 - Topologia básica da rede WLAN padrão IEEE-802.11.

Nas redes MANET, o mesmo ocorre para o nó móvel que estiver dentro da mesma área de cobertura. Caso um nó saia dessa área de cobertura, ou seja, não possua sinal de rádio freqüência, a mobilidade nesse padrão seria perdida. Por isso, é necessário o

uso de várias regiões de coberturas de radiofreqüência MANET interligadas entre si a fim de manter essa conectividade.

Em ambos cenários, WLAN infraestruturada e MANET, não existe a garantia da continuidade de sessão das aplicações em curso, quando o dispositivo se desloca de um domínio a outro. A Figura 2.2 mostra um dispositivo em *roaming* realizado na camada 2 entre diferentes domínios e com o mesmo SSID. O nó tem de manter o seu endereço de camada 3 inalterado durante esse processo (esse cenário reflete um ambiente do tipo VLAN).

Caso os endereços de camada 3 dos APs forem distintos (cada switch de camada 2 conectado com um roteador distinto do outro), a estação descartaria todo tipo de sessão previamente estabelecida em um dos domínios, após a realização do *roaming*.

Os dados da Tabela 2.1 mostram um resumo da capacidade de cada padrão IEEE 802.11a/b/g quanto a sua freqüência de operação, tipo de acesso, número de canais, duplexagem, largura de banda e taxa de transmissão máxima de dados. O padrão IEEE 802.11b foi usado nesta tese por possuir a menor taxa de transmissão de dados, entre outros aspectos técnicos. Dessa forma, se garante que o sistema proposto seja funcional usando-se o primeiro padrão da família 802.11 conhecido como Wi-Fi.

Este padrão é especificado para operar na banda de ISM de 2,4 GHz utilizando espalhamento espectral de seqüência direta DSSS atingindo taxa de até 11 Mbps com alcance típico em ambientes fechados de 50 a 100 metros, dependendo da quantidade de paredes e número de obstruções. O sistema pode operar com 14 canais superpostos e 3 não superpostos de 22 MHz de largura de banda. O 802.11 pode também utilizar o espalhamento espectral por salto em freqüência FHSS.

Definido apenas para sub camada MAC da camada 2, o 802.11 oferece dois tipos de controle de acesso, um assíncrono e outro síncrono, livre de contenção [59]. Quando as estações se comunicam diretamente umas com as outras (como numa rede MANET), o controle é assíncrono e a coordenação da rede acontece de forma distribuída. O controle assíncrono é realizado por uma função de coordenação distribuída DCF, que utiliza a técnica CSMA/CA, na qual o nó que deseja transmitir

ativa seu receptor para detectar a presença de portadora no meio antes de iniciar sua transmissão. Este método é semelhante ao CSMA/CD utilizado nas redes Ethernet padrão 802 e também se baseia na detecção de portadoras, mas diferentemente deste, o CSMA/CA evita colisões em lugar de detectá-las.

O método assíncrono é utilizado, nesse ambiente, porque os terminais do padrão 802.11 são *half-duplex* para reduzir seu custo de complexidade, não podendo transmitir e receber simultaneamente. A detecção de portadora é feita de duas maneiras: utilizando o sinal de canal livre CCA, cuja implementação é obrigatória, ou utilizando pacotes RTS/CTS que implementam um esquema de detecção de portadora virtual utilizando um vetor de alocação do meio NAV, cuja implementação é tambem opcional. Caso o meio esteja livre e permaneça nesse estado por um tempo maior que um intervalo de tempo bem definido (*Interframe Space*), o nó pode transmitir. Caso contrário, a transmissão é atrasada por um intervalo de espera aleatório, uniformemente distribuído em uma janela de contenção CW, limitada por valores CWmin e CWmax.

Em redes WLAN os nós se comunicam entre si passando por um AP, sendo que, neste caso, o controle é síncrono. Esse controle síncrono é fornecido pela função de coordenação pontual PCF que, basicamente, implementa o polling como método de acesso. O modo PCF é uma função opcional construída sobre o modo DCF e implementada através de um mecanismo de acesso ordenado ao meio que proporciona a oportunidade de transmitir sem contensão.

De acordo com a Tabela 2.1, para se ter um *roaming* entre domínios, é necessária a implementação de APs com no mínimo 3 canais não sobrepostos (no padrão IEEE 802.11b), Figura 2.3. Esses canais têm de pertencer a faixa de freqüência de operação do padrão escolhido. Essa técnica permite o reúso de freqüência ao longo do domínio da rede infraestuturada. Caso os raios de operação de cada AP, também conhecido como célula, não forem sobrepostos, a perda de sinal na camada 1 inviabiliza a mobilidade na camada 2.

Tabela 2.1 - Padrão IEEE 802.11 [1].

	IEEE 802.11a	**IEEE 802.11b**	**IEEE 802.11g**
Freqüência de Operação	5,15 – 5,35 GHz 5,47 – 5,825 GHz	2,4 – 2,485 GHz	2,4 – 2,485 GHz
Modulação	BPSK, QPSK, 16QAM, 64QAM, OFDM	BPSK, DQPSK no cabeçalho; BPSK, QPSK no payload (CCK, PBCC)	BPSK, DQPSK, QPSK, 16QAM, 64QAM, OFDM, CCK, PBCC
Múltiplo Acesso	OFDM CSMA/CA	CSMA/CA	OFDM CSMA/CA
Duplex	TDD	TDD	TDD
Largura de Banda do Canal	22 MHz	22 MHz	22 MHz
Número de Canais	12 (não sobrepostos)	14 (sobrepostos) 3 (não sobrepostos)	14 (sobrepostos) 3 (não sobrepostos)
Taxa máxima de transmissão de dados	54 Mbit/s	11 Mbit/s	54 Mbit/s

Essa perda de sinal tende a ser um sério problema para sistemas de comunicação sem fio. A propagação de sinal pelo espaço livre causa degradações nos dados que estão sendo transportados, através dos efeitos de múltiplo percurso e efeito Doppler. O efeito de múltiplo percurso consiste na recepção do mesmo sinal por um receptor em instante de tempos diferentes, ou seja, o sinal transmitido é refletido por objetos no ambiente entre as antenas transmissora e receptora [58].

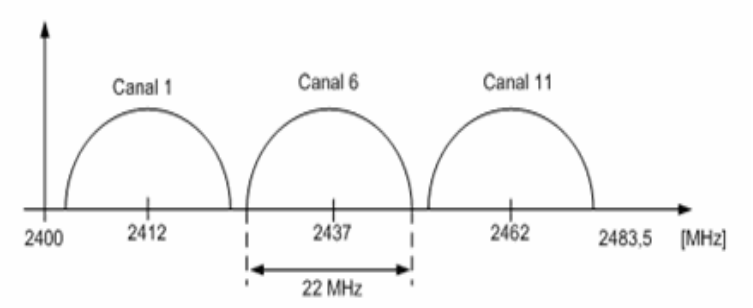


Figura 2.3 - Atribuição de canais no padrão IEEE 802.11b.

Como resultado, várias réplicas do mesmo sinal chegam ao receptor com pequenos intervalos de tempo, configurando o espalhamento de retardo (*delay spread*), que consiste na diferença de tempo entre a recepção do primeiro sinal e a última réplica do mesmo, ocasionando interferência intersimbólica, conhecida como ISI. Assim, em 1966 surgiu uma técnica de modulação e multiplexação denominada OFDM, com o objetivo de minimizar os problemas ocorridos na transmissão.

A técnica OFDM é um esquema de transmissão (*modulação e multiplexação*) utilizado por vários sistemas de comunicação digital tais como as redes locais e metropolitanas sem fios (*WLAN*, *WMAN* e *MANET*). Ela é uma técnica de modulação de múltiplas portadoras, isto é, para transmitir uma dada seqüência de bits, a técnica OFDM utiliza dezenas ou até milhares de portadoras paralelas, ao invés de uma única portadora como fazem as técnicas de transmissão convencionais. Isso dá à técnica OFDM as seguintes vantagens:

- maior tolerância aos efeitos degradantes da propagação multipercurso sobre o sinal recebido;
- maior tolerância à interferência de canal adjacente ou a uma interferência qualquer que afete o sinal desejado de forma não uniforme na freqüência, isto é, que não afete igualmente toda a faixa de freqüência ocupada pelo sinal desejado.
- uma flexibilidade inerente que permite que se transmita informações específicas em subfaixas de freqüência diferentes dentro da banda passante do canal e usando modulações distintas.

Pode-se concluir que, para se ter mobilidade no padrão 802.11, não basta garantir o correto funcionamento da camada 2. Deve-se garantir também o bom funcionamento dos mecanismos providos pela camada física do dispositivo móvel e amenizar problemas decorrentes da perda de sinal na propagação do espaço livre, desvanecimento seletivo, entre outros problemas técnicos mencionados anteriormente.

2.3 - MOBILIDADE NA CAMADA IP

2.3.1 - Funcionamento básico

O MIP usa três passos principais para prover mobilidade aos nós móveis: descoberta do Agente; registro e entrega de datagramas.

O mecanismo de descoberta do agente é usado para que um nó móvel possa saber a respeito do seu novo ponto de conexão, i.e, que ele adquira um novo endereço IP devido ao seu deslocamento através da Internet a fim de manter conectividade com a sua rede de origem. Quando o nó móvel descobre o endereço IP do seu novo ponto de conexão, ele pode se registrar com o seu HA que, por seguinte, o representará na sua rede local, conhecida como HN. O MIP também define mecanismos simples de entrega de datagramas ao MN, pelo uso de protocolos de tunelamento estabelecidos entre o HA e FA quando esse estiver ausente do seu HN.

Para manter as conexões existentes da camada de transporte, como as conexões TCP, a cada nó móvel é atribuído um home address estático. O home address é um endereço IP que permite o nó móvel de sempre receber dados fora de seu ponto de conexão como se ele estivesse na sua rede de origem HN.

Quando o nó móvel é conectado ao seu novo ponto de rede (chamado de foreign network), ele usa um endereço IP conhecido como care-of address (CoA). O CoA é um endereço IP válido na rede do FA que o nó móvel está visitando. Qualquer que seja a rede que o nó móvel for visitar durante o seu deslocamento, ele tem de adquirir um novo care-of address referente à nova rede a fim de ter esse endereço IP cadastrado no seu home network pelo home agent.

Para receber os datagramas que lhe forem destinados, o nó móvel tem de registrar o seu care-of address atual com o seu HA. Para isso acontecer, o nó móvel tem de se registrar através do foreign agent localizado na rede estrangeira, i.e., no FN. Uma vez que o registro tenha ocorrido sem falha com o home agent passando pelo foreign agent, cada datagrama enviado para o nó móvel no seu home address é recebido pelo

home agent e encaminhado ao care-of address, sendo o foreign agent que, por vez, os encaminha ao nó móvel visitando a sua rede.

A Figura 2.4 exemplifica como os datagramas são roteados para e do nó móvel, i.e., o MN, que se registrou com o seu HA pelo FA. Inicialmente, o CN, um usuário da rede Internet, envia datagramas ao MN. Esse datagrama é roteado para o home network do nó móvel e interceptado pelo HA. O HA encapsula o datagrama encima de um novo datagrama e que é tunelado para o FA (esse processo será detalhado na sessão 2.4). Quando o datagrama tunelado é recebido pelo FA, ele é desencapsulado (processo inverso ao ocorrido no início do tunelamento no HA) e entregue para o MN. Caso o MN queira enviar um datagrama para o CN, os mecanismos do protocolo IP entrarão diretamente em ação.

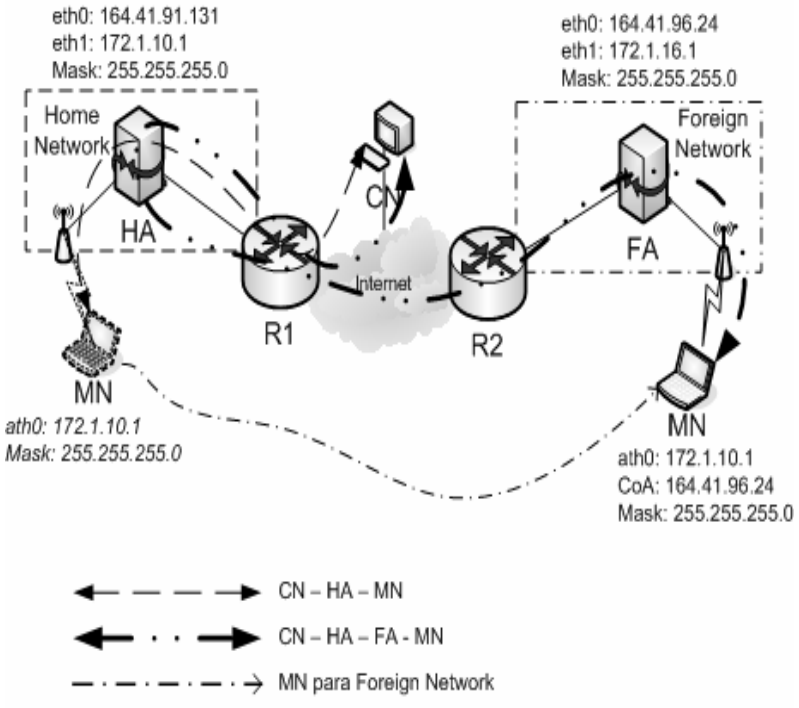


Figura 2.4 - Roteamento de datagramas para e de um nó móvel numa rede estrangeira.

Se o CoA for do FA, ele é conhecido como Care-of Address do foreign agent. Se o nó puder adquirir um CoA por outros meios a não ser pelo FA, esse último se torna transparente na entrega de um CoA ao MN. O protocolo DHCP é um exemplo de como se obter um CoA na rede estrangeira sem ajuda do FA [16,17]. O home agent pode então encaminhar os pacotes diretamente até o nó móvel (sendo neste caso a outra ponta do túnel MIP) na rede estrangeira sem passar pelo FA, visto que o novo

CoA é do próprio MN; esse novo CoA é então chamado de co-located Care-of Address.

Esta tese visa usar os FAs como CoA dos nós móveis que forem visitar sua rede pelas razões que serão explicadas no Capítulo 4. Também se usará o processo básico do MIP tendo os home agent e foreign agent como peças fundamentais para tráfego das mensagens MIPs.

2.3.2 - Descoberta de Agentes

O processo de descoberta de agentes de mobilidade, i.e, de um home agent ou foreign agent, do MIP, utiliza o protocolo ICMP Router Discovery. Uma extensão é aplicada ao formato da mensagem ICMP para facilitar a descoberta da presença dos agentes de mobilidade pelos nós que tenham implementado o protocolo MIP.

Cada agente de mobilidade faz o Flooding periódico dos agent advertisements dentro da rede a qual ele estiver diretamente conectado, para avisar a sua presença na rede. Esses advertisements contêm os Care-of Address `<CoAddress_addr>`. Os nós móveis escutam esses advertisements a fim de escolher o agente de mobilidade ao qual eles tem de se registrar com o home agent.

O nó móvel, no caso de querer um CoA com urgência e não esperar por um advertisement do agente de mobilidade, envia por Flooding ou multicast um agent solicitation. Qualquer agente de mobilidade que receber a mensagem de agent solicitation dará ou não permissão ao nó móvel para que esse possa usar os seus serviços. Caso seja aceita a mensagem, envia um agent advertisement ao nó móvel via unicast.

2.3.3 - Registro

O nó móvel envia um registration request para o seu home agent para lhe informar a respeito:

- do seu CoA atual,
- do tempo que ele deseja manter esse CoA,

- dos parâmetros e flags que indicam como os datagramas deveriam ser encaminhados, i.e. tipo de tunelamento entre HA e o CoA,
- das características especiais disponíveis no FA.

Quando o home agent recebe o registration request (assumindo que esse registro tenha sido aceito), ele faz uma associação do endereço de casa do nó móvel com o care-of address especificado no campo da mensagem do registration request. Ele envia um registration reply de volta ao nó móvel para que esse possa saber que o registro foi aceito. Esse registro somente é válido por um determinado tempo conhecido como lifetime. Caso o nó móvel deseje guardar o registro atual, ele tem de refazer o processo de registro com o home agent antes do término do lifetime atual.

- A Figura 2.5 mostra o processo de sinalização entre o MN, FA, HA e CN. Em todo esse processo, o foreign agent é considerado passivo, i.e, ele só encaminha as mensagens de registration request e registration reply tanto na ida quanto na volta entre o nó móvel e o home agent. Cada foreign agent tem de manter uma lista dos nós móveis visitantes na sua rede com quem tiver se registrado durante a propagação do agent advertisement.

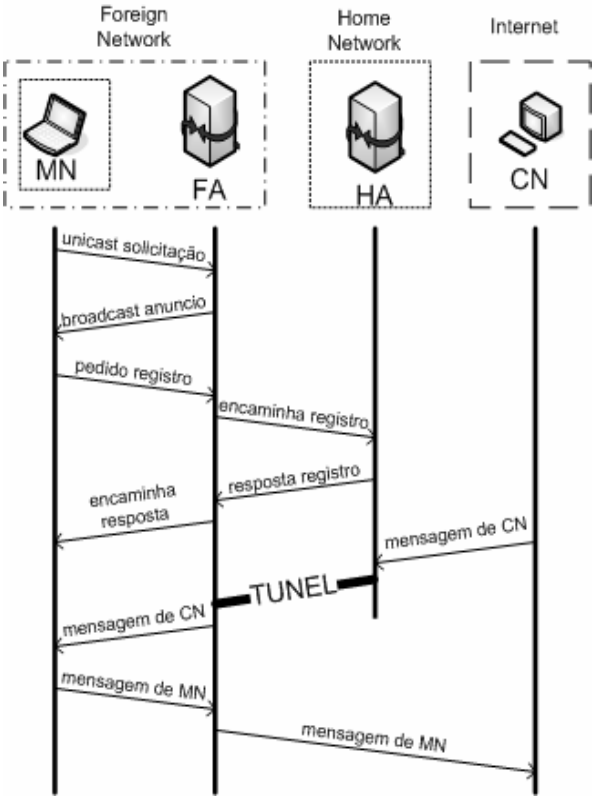


Figura 2.5 - Processo de sinalização entre MN, FA e HA.

A lista dos nós visitantes contém entre outras opções os:

```
<LL_node_addr, Home_addr, HomeAgent_addr,>
```

- `<LL_node_addr>`: o endereço MAC do MN da rede nativa.
- `<Home_addr>:` o endereço IP de origem do MN da rede nativa.
- `<HomeAgent_addr>`: o endereço IP do agente de mobilidade, neste caso sendo o HA

2.3.4 - Entrega de Datagramas

Um nó móvel conectado a sua rede de origem pode receber os datagramas a ele destinados a partir de um nó correspondente da Internet ou da mesma rede sem haver necessidade de passar pelo HA. Uma vez esse nó tiver se deslocado para outro ponto de conexão, agora estando numa rede estrangeira, onde tiver um FA, esse lhe fornecerá um CoA a fim de se cadastrar junto ao seu HA de origem como visto anteriormente.

Um nó correspondente querendo enviar datagramas para o nó ausente na rede de origem, terá os seus datagramas capturados pelo HA dessa rede por ser o único que sabe da localização do nó móvel (o HA tem uma lista dos nós a ele associados dentro de sua tabela de roteamento). No próximo passo o HA encapsula esses dados via tunelamento IP-in-IP (outros mecanismos de tunelamento existem para o MIP) e encaminha até o FA onde estiver conectado o nó móvel [13,18,23,24,30,33,34]. Os datagramas são em seguida desencapsulados no final do túnel pelo FA que fará a entrega ao MN com o campo `<Dest_addr,Source_addr>` contido no campo do protocolo IP que foi enviado do CN ao HA. É claro que os datagramas também podem ser desencapsulados pelo MN caso esse tiver um Co-Located care of address. E por roteamento IP normal, o MN pode enviar dados ao CN diretamente pela Internet, mas passando pelo FA, esse último sendo seu gateway padrão na rede estrangeira. Neste caso o `<Dest_addr==IP address do CN>`, e o `<Source_addr==IP address do MN>` tem de ser de um CN da Internet. Dessa forma os datagramas podem ser roteáveis pela rede pública.

Agora, no caso do MN enviar datagramas para o CN via HA, o método do *tunelamento reverso* é usado para tal efeito e o ponto de conexão lógico atual do MN é transparente para o CN.

2.3.5 - Detecção de Mobilidade

Uma das formas de saber se um nó móvel se deslocou de uma rede a outra é primeiramente através do seu *Care-of Address*. Lembrando que o CoA indica o endereço IP do agente MIP, podendo ser tanto o HA como o FA, ao qual o nó móvel estiver conectado ou registrado logicamente.

O MIP define três mecanismos para a detecção de mobilidade do nó móvel [3]:

- **Lazy Cell Switching (LCS)**

O LCS faz com que um nó móvel que não receber um *agent advertisement* numa rede estrangeira com determinado *lifetime* válido, informe o nó de sua possível perda de contacto com o agente de mobilidade. Isso implicará na sua impossibilidade de registrar-se com o seu HA de origem. Caso ele for receber um *agent advertisement* do FA, ele pode iniciar o processo de registro junto ao seu HA, passando pelo FA, salve a aceitação do HA. O HÁ, dessa forma, pode saber a nova localização lógica do seu MN. Caso nenhuma das opções acima tiver resultado, o MN tem de enviar um *agent solicitations* para receber algum *agent advertisement* de um agente de mobilidade da rede na qual ele estiver conectado.

- **Prefix Matching**

Neste caso, o nó usa um processo comparativo do endereço de rede (ou prefixo da rede) da sua rede com o da rede a qual ele estiver conectado para definir a sua atual localização. Isso porque os prefixos de rede dos agentes de mobilidade (HA/FA) são diferentes e implica numa mudança do ponto lógico do MN.

- **Eager Cell Switching (ECS)**

O ECS é um método usado para detecção de mobilidade do nó móvel que for passar por várias células contendo cada uma pelo menos um agente de mobilidade e recebendo vários agent advertisements deles. Receber vários agent advertisement

complica a tarefa do nó que quer se registrar junto ao agente de mobilidade que for lhe enviar essas mensagens. Isso porque caso as células não forem muito distantes uma da outra, o nó pode querer se registrar nas células que estiverem enviando esses agent advertisements e, assim sendo, dificultando a sua localização pelo HA de origem. Tem-se aqui um fenômeno de ping-pong de registros entre células. Para esse efeito, o nó que for entrar numa célula e seguir adiante, tem de se registrar com o mais novo CoA e terminar a sua conexão com o antigo CoA, mesmo se ele for continuar recebendo os agent advertisements do antigo FA.

2.3.6 - Consideração de Segurança

Por ter mensagens de registro e detecção de mobilidade, é importante ter um mecanismo de segurança para que tanto o MN, FA e HA sejam protegidos de qualquer ameaça maliciosa local ou da Internet. Mudança de registro por um FA ou HA desconhecido pode acontecer, assim como de um MN obter um CoA ilegalmente; ou seja, várias possibilidade de intrusão seriam possíveis para corromper o funcionamento do MIP.

Para evitar problemas de ataques alheios, o MIP usa a mensagem MD5 para o cômputo de uma única assinatura digital em cada mensagem de registro [3,25,26,29,30,38]. No processo geral, o MN e HA trocam senhas secretas e o campo da mensagem de registro incluiu um campo de identificação que muda a cada registro.

2.3.7 - Mensagem do tipo registration request

Um nó móvel tem de usar a mensagem *registration request* caso queira se cadastrar no seu HA, a fim de ter a sua associação de `<Home_addr, CoA_addr,Lifetime>` atualizados na tabela de roteamento do HA. Essa mensagem é encapsulada no cabeçalho do protocolo UDP e tendo `<Src_port=variavel>` e `<Dst_port=434>.` As mensagens para requerimento de registros fazem uso dos seguintes campos principais:

`<Type, Flags, Lifetime, Home_addr, HA_addr, CoA_addr, ID>`

- `<Type>`: recebe o valor de 1 por se tratar de um request.

- <Flags>: campo tendo um conjunto de bits permitindo ao nó móvel alterar os parâmetros de *tunelamento*, *binding* entre outras opções.
- <Lifetime>: indica o número de segundos restantes antes do registro expirar.
- <Home_addr>: o endereço IP de origem do MN da rede HN
- <HA_addr>: o endereço IP do agente de mobilidade, neste caso sendo o HA.
- <CoA_addr>: o endereço IP do agente de mobilidade sendo o FA (ou MN em caso de obtê-lo via DHCP).
- <ID>: campo de 64 bits implementado pelo MN e usado para comparar e validar as mensagens de *registration request* e *registration reply*. Isso garante de certa forma a integridade das mensagens *registration reply*.

Essa mensagem sofre uma pequena mudança nesta tese para que o agente passivo da rede MANET possa responder ou não para o nó enviando o(s) *request(s)* e contendo os campos que ele possa usar para atualizar a sua tabela de nós registrados no ativo e enviar um *reply* ao(s) nó(s) em questão caso o agente ativo tiver problemas de recursos, por exemplo.

2.3.8 - Mensagem do tipo registration reply

Após a recepção da mensagem de *registration request* e ter verificado a autenticidade da mesma mostrando que provém do MN de sua rede, o HA cria uma mensagem *registration reply* que será encaminhada via tunelamento até o <CoA_addr> de destino do MN.

Essa mensagem é encapsulada no cabeçalho do protocolo UDP, tendo <Src_port=variavel> e <Dst_port==Src_port_rqst>. Neste caso a porta de destino recebe o valor copiado da porta de origem contida na recepção da mensagem do *registration request*.

Caso,

<Lifetime#_IN_RQ> > <Lifetime#_IN_RP>

- <Lifetime#_IN_RQ>: indica o número de segundos restantes antes do registro contido no *registration request* expirar.

- <Lifetime#_IN_RQ>: indica o número de segundos restantes antes do registro contido no *registration reply* expirar.

o <Lifetime> contido no *registration reply* tem de ser usado pelo MN. Por outro lado, o <Lifetime> do *registration request* seria usado caso seja menor que o do *registration reply*. As mensagens para resposta de registros fazem uso dos seguintes campos principais:

<Type, Code, Lifetime, Home_addr, HA_addr, CoA_addr, ID>

- <Type>: recebe o valor de 3 por se tratar do reply.
- <Code>: campo indicando o resultado do processo do registro caso tenha sido aceito ou não pelo agente de mobilidade.
- <Lifetime>: indica o número de segundos restantes antes do registro espirar e tendo associação direta com o campo <Code>.
- <Home_addr>: o endereço IP de origem do MN da rede HN
- <HA_addr>: o endereço IP do agente de mobilidade, neste caso sendo o HA.
- <CoA_addr>: o endereço IP do agente de mobilidade sendo o FA (ou MN em caso de obtê-lo via DHCP).
- <ID>: campo de 64 bits implementado pelo MN e usado para comparar e validar as mensagens de *registration request* e *registration reply*. Isso garante de certa forma a integridade das mensagens *registration reply*.

2.3.9 - Flooding dos Advertisements

Para um nó MN saber da existência de agentes de mobilidade durante a sua mobilidade ou não, ele tem de receber o Flooding de *agents advertisements* disseminados pelos agentes de mobilidade presentes na rede em questão. Essas mensagens contêm um TTL=1 para redes infraestuturadas.

Entretanto, as redes MANET são do tipo saltos múltiplos e requerem uma modificação dessas mensagens para ter o TTL=N (N sendo o número total de nós na rede MANET), cujo valor é decrementado a cada salto e usa o endereço IP

<Dst_addr==255.255.255.255>. Essas mensagens fazem uso dos seguintes campos principais:

<Type,Seq#,Reg_Lifetime,Flags,CoA_addrs>

- <Type>: recebe o valor de 16 por se tratar de um advertisement.
- <Seq#>: campo indicando a contagem dos agent advertisements desde o início de sua disseminação pelo agente de mobilidade.
- <Reg_Lifetime>: indica o tempo máximo em segundos que um agente deseja aceitar um registration request do MN.
- <Flags>: campo tendo um conjunto de bits permitindo ao agente de mobilidade alterar os parâmetros de *tunelamento*, *registro* entre outras opções.
- <CoA_addrs>: o(s) endereço(s) IP do agente de mobilidade sendo o FA pelo qual o MN terá de se registrar.

Essa mensagem foi alterada para que os agentes (ativo e passivo) de mobilidade da rede MANET possam saber a respeito de seus estados de atividades, contendo um campo de 32 bits a mais indicando o endereço do agente que está propagando a mensagem [ver apêndice B].

2.4 - MOBILIDADE NA CAMADA DE TRANSPORTE

Para um nó ter mobilidade no nível da camada de transporte, principalmente no que diz respeito a aplicações orientadas a conexão, as mais comumente usadas, ele tem de ter o conjunto <Src_addr,Src_port,Dst_addr,Dst_port> inalterado para manter a conexão com as aplicações em curso. Então, essa camada sempre teria de prover de forma dinâmica uma associação de novos endereços IPs às conexões do nó em mobilidade. Como o SCTP possibilita que um *endpoint* SCTP suporte mais de um endereço IP, acessos redundantes podem ser utilizados para reforçar a comunicação. Os dados retransmitidos usam os endereços alternativos para aumentar a probabilidade de alcançar o ponto remoto destino.

Esses endpoints SCTP têm de trocar listas de endereços durante o início da associação (*binding*). Cada endpoint deve ser capaz de receber mensagens de qualquer dos endereços associados ao endpoint remoto, Figura 2.6. Isso seria então uma maneira de

garantir a mobilidade na camada de transporte como descrito em [36,37]. Em [36], por exemplo, o MSOCKS gera dois tipos de sinalização para manter a conexão TCP: a primeira sendo do nó fixo até o Proxy e a segunda sendo do Proxy até o nó móvel. O Proxy dá ilusão de uma comunicação TCP fim a fim entre nó fixo e móvel. Para se ter um correto funcionamento do MSOCKS, o Proxy, o nó fixo e o nó móvel devem ter as bibliotecas do MSOCKS.

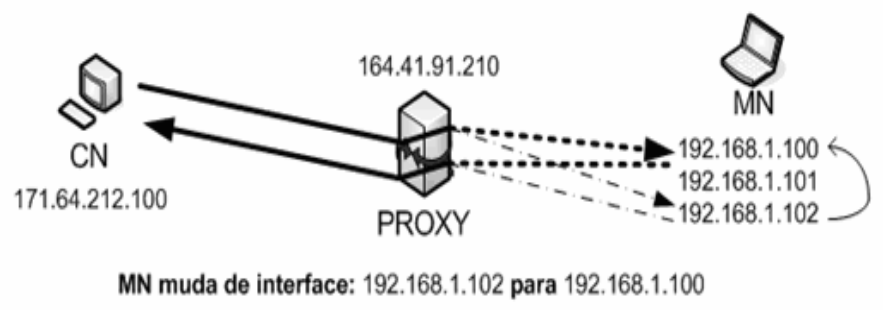


Figura 2.6 - Mobilidade na Camada de Transporte.

3 - REDE MOBILE AD HOC – MANET

3.1 - INTRODUÇÃO

As redes Mobile Ad hoc NETwork (MANET) são redes sem fio nas quais os nós móveis têm possibilidade de trocarem informação sem o auxílio de uma rede infraestuturada. Essas redes são também conhecidas como redes espontâneas, onde os nós se comunicam diretamente entre eles, ponto a ponto.

Por não serem de característica infraestuturadas, os serviços de roteamento são estabelecidos de maneira cooperativa e cada nó participante da rede MANET atua como um possível roteador. Dessa forma, caso um nó deseje se comunicar com outro dentro da área de cobertura da rede, ele encaminha seus pacotes com o auxílio dos nós vizinhos até chegar ao nó de destino. Nesse contexto, os pacotes podem passar por vários nós antes de chegarem ao destinatário [50].

Numa rede MANET, os nós podem continuamente e a qualquer momento entrar e sair de sua área de cobertura. Como resultado, a adesão dos nós com a MANET é mantida dinamicamente, o que faz com que a topologia da rede esteja sujeita à mudanças freqüentes e imprevisíveis.

Portanto, as MANETs são consideradas redes móveis com características de salto múltiplos onde a conectividade entre os nós é assegurada através de um protocolo de roteamento colaborativo podendo ser de tipo: pró-ativo ou reativo.

3.2 - PROTOCOLOS DE ROTEAMENTOS

Os protocolos tradicionais conhecidos como vetor de distância (*distance-vector)* e estado de enlace (*link-state)* são dito pró-ativo no sentido de sempre manter as rotas de todos os nós da rede, até mesmo aqueles que não forem receber pacotes atualizados, para que quando um pacote necessite ser encaminhado, a rota seja definida de maneira imediata. Eles possuem a vantagem de ter um atraso mínimo quando um nó for solicitar determinada rota para enviar um pacote até outro nó, já que as rotas de toda rede constam em sua tabela de roteamento.

A desvantagem desses tipos de protocolos é que essas constantes mensagens de controle implicam em deterioração dos escassos recursos dos nós, como energia e banda de transmissão ao longo do seu deslocamento através da rede, para se manter a consistência e a topologia da rede atualizada.

Como protocolo pró-ativo, o OLSR permite uma redução do uso dos recursos da rede de forma significativa [54]. Cada nó faz o Flooding periódico de mensagens *Hello* com informações específicas de nós dentro da rede para troca de informação de vizinhança. Estas informações contêm o endereço IP de cada nó, número de seqüência e uma lista da informação de distância até os nós vizinhos. Após recepção desta informação, o nó começa a montar a sua própria tabela de roteamento e pode assim determinar a distância para o nó com o qual ele deseja se comunicar usando um algoritmo de menor caminho. Se o nó for receber um pacote com o mesmo número de seqüência duas vezes, ele tem de descartá-lo. A tabela de roteamento é atualizada quando: uma mudança na vizinhança é detectada, uma rota até o destino expirou ou quando um melhor caminho é encontrado até o destino desejado.

Outro protocolo que vem chamando atenção é o pró-ativo TBRPF que provê um algoritmo de caminho curto para roteamento salto a salto para cada nó destino. Cada nó que for executar o protocolo TBRPF faz o cômputo da árvore relacionada com a informação parcial da topologia guardada dentro da sua tabela topológica, com o uso do algoritmo de Dijktra modificado. Para minimizar o overhead das mensagens, parte da tabela é repassada para os nós vizinhos. O protocolo faz uso de uma combinação de atualizações periódicas e diferenciais para que cada nó na rede possa receber a árvore relacionada com a informação parcial da topologia da rede guardada dentro da sua tabela. Esse mecanismo de atualizações diferenciais é também usado para descoberta de vizinhos pelas mensagens *Hello* e receber avisos quanto a mudanças de estado no nó. Cada nó na rede pode também contribuir, opcionalmente, na adição de uma informação topológica e, assim, prover certa confiabilidade em redes altamente móveis.

De outra maneira, os protocolos reativos somente operam no caso da necessidade de um nó querer se comunicar com outro nó. Essa abordagem faz com que os nós procurem por rotas a serem estabelecidas sob demanda ou utilizar as já ativas. O

mecanismo do protocolo reativo inicia o processo de descobrimento da rota somente quando esta for requerida.

A vantagem do protocolo reativo é de possibilitar uma economia de energia e banda de transmissão. Mas esse processo de descobrimento de rotas pode vir aumentar o atraso na criação da rota ativa caso a rede seja extremamente grande, em número de nós.

Como exemplos de protocolos reativos, têm-se o AODV [43]. Devido aos bons resultados comparativos obtidos em [53], [55] e por já ser uma RFC de caráter experimental [43], o protocolo reativo AODV será descrito na sessão 3.4 a seguir, por fazer parte do protocolo de roteamento da rede MANET e MIP escolhido para ser analisado nessa tese. A Tabela 3.1 apresenta uma classificação dos protocolos de roteamento para MANET. Os protocolos em itálico são atualmente RFCs experimentais e as demais drafts do IETF.

Tabela 3.1 - Classificação dos Protocolos de Roteamento para MANET.

	Roteamento	
Característica técnica	Pró-Ativo	Reativo
Estado de Enlace	**OLSR** ***(RFC-3626)*** **TBRPF** ***(RFC-3684)***	
Vetor de Distância		**AODV** ***(RFC-3561)***

3.3 - O PROTOCOLO AD HOC ON-DEMAND DISTANCE VECTOR ROUTING (AODV)

O AODV é um protocolo de roteamento de vetor de distância reativo [43]. A característica desse protocolo faz com que ele requisite uma rota somente quando for desejado por um dos nós da rede e, dessa maneira, evita que os nós não participantes mantenham as rotas para outros destinos quando a requisição não for passar por eles. Assim, o AODV reduz a necessidade de difundir as mensagens de roteamento através da rede MANET, o que aumenta sua escalabilidade e evita loops de rotas desnecessárias graças ao uso de números de seqüências que permitem indicar o quanto uma rota é atual.

2.4.1 - Estabelecimento de rota

O AODV requer de cada nó uma atualização da rota de entrada para o destino ao qual ele for se comunicar. Cada entrada de rota guarda certos campos que facilitam o descobrimento de rotas entre os nós de origem e destino, seu formato é:

`<Flags,Hop_Count,Flooding_ID,Source_addr,Source_Seq#,Dest_addr,Dest_Seq#>`

- `<Flags>`: característica da rota; *up* (válida), *down* (inválida) ou *repair* (manutenção).
- `<Hop_Count>`: o número de saltos desde o endereço IP do nó de origem até o endereço do nó de destino.
- `<Flooding_ID>`: o número do Flooding.
- `<Source_addr>`: o endereço IP do nó de origem pelo qual é iniciada a requisição de rota.
- `<Source_Seq#>`: o número de seqüência associado a rota de origem.
- `<Dest_addr>`: o endereço IP do nó de destino que for receber a rota requisitada.
- `<Dest_Seq#>`: o número de seqüência associado a rota de destino.

Tradicionalmente, o endereço IP <255.255.255.255> de broadcast limitado é usado para enviar um Flooding dentro de um enlace local. Os datagramas destinados ao endereço de broadcast limitado nunca são encaminhados pelo roteador. No AODV, o Flooding é determinado pelo processo de disseminação, ou seja, datagramas destinados ao endereço de broadcast são encaminhados por todos os nós da rede MANET de tal forma que todos possam receber esse Flooding.

Já que o nó usa uma interface sem fio, ele é sujeito à recepção de várias cópias do mesmo pacotes de Flooding provenientes dos nós vizinhos. Desse fato, um mecanismo de detecção de mensagens duplicadas tem de ser incluído dentro do IP. Para identificar unicamente cada Flooding, os parâmetros <Source_addr,ID>, do cabeçalho do protocolo IP, para Flooding são usados. Cada nó guarda no *cache* os logs de cada Flooding que ele tiver encaminhado e quando ele receber Floodings duplicados, esses são simplesmente descartados.

O AODV usa o broadcasting para entrega de informações de roteamento dentro da rede MANET. Para não sobrecarregar a rede com Floodings, define o Time To Live, TTL com valor igual a 1. O AODV implementado em cada nó decide se a informação de roteamento tem de ser encaminhada ou não. Se o nó decidir encaminhar a informação, ele envia um *novo broadcast* próprio contendo as informações de roteamento. Esse novo broadcast é um novo pacote IP e os parâmetros <Source_addr,ID> usados por outros nós no broadcast não são somente para a informação de roteamento, mas também para broadcast local. Para o AODV ser capaz de diferenciar várias cópias de uma mesma informação de roteamento, parâmetros similares ao anterior são usados: <Source_addr,Broadcast_ID>. Cada nó mantém um contador de broadcast_id que aumenta cada vez que o AODV envia um broadcast.

2.4.2 - Gerência de conectividade local

Cada nó mantém as informações de suas conectividades locais (vizinhança) guardando uma lista dos vizinhos que ele tenha detectado. Essa lista é atualizada cada vez que um pacote for recebido. Para mantê-lo atualizado, mesmo na ausência da entrega de pacotes, mensagens *hello* são usadas. Se um nó não tiver enviado um

broadcast dentro de determinado intervalo de tempo, ele envia um broadcast (com TTL = 1) de uma mensagem *hello* a fim de avisar os seus vizinhos sobre a sua presença. Um nó vizinho é considerado inalcançável caso o broadcast de mensagens *hello* consecutivas, vindo dele, não forem recebidas por outro nó. O uso das mensagens *hello* é controvérsia já que elas quebram a natureza reativa do AODV e isso sobrecarrega a rede mesmo na ausência de roteamento de pacotes dentro da rede. Maior discussão pode ser encontrada em [45].

Além de rastrear os nós vizinhos, cada nó mantém informação dos nós para os quais ele encaminha pacotes. Esses nós são considerados *nós ativos*. Como será visto mais adiante, o AODV usa essa informação para enviar informações sobre perda de conectividade nas camadas inferiores.

3.3.3 - Descoberta de rotas

Cada vez que um nó quer se comunicar com um destino ao qual ele não tem informação de roteamento na sua tabela de rotas, ele inicia uma *descoberta de rota*. O princípio da descoberta de rotas é a de montar uma rota bidirecional do nó origem até o nó destino, i.e., ambas rotas *direta* e *reversa*. A descoberta de rota é inicializada enviando um broadcast do *route request* (RREQ) para todos os nós vizinhos. O RREQ contém as seguintes informações:

```
<Hop_Count,Broadcast_ID,Source_addr,Source_Seq#,Dest_addr
,Dest_Seq#>
```

Cada nó que for receber o RREQ procura na sua tabela de roteamento pela existência de uma nova rota para o destino requerido ou se ele próprio é o destino da mensagem. Caso uma das suposições anteriores seja certa, ele envia um *route reply* (RREP) para o nó que originou o RREQ ou faz o broadcast do RREQ para os seus vizinhos, após ter incrementado o *hop_count*. Dessa forma, cada nó estabelece uma *rota reversa* temporária até o nó de origem. Uma rota da tabela de roteamento é considerada suficientemente atualizada se o,

```
<Dest_Seq#_IN_TABLE> >= <Dest_Seq#_IN_RREQ>
```

- `<Dest_Seq#_IN_TABLE>`: o número de seqüência associado a rota de destino na tabela de roteamento.
- `<Dest_Seq#_IN_RREQ>`: o número de seqüência associado a rota de destino no campo da mensagem Route Request.

Na existência de uma provável rota entre origem e destino, alguns nós vão enviar uma mensagem RREP. Qualquer nó intermediário que tiver uma rota nova dentro de sua tabela de roteamento pode gerar um RREP. Se nenhum nó intermediário for gerar o RREP, o RREQ se propaga até o destino, que gerará o RREP. O RREP contém as seguintes informações:

```
<Hop_Count,Lifetime,Source_addr,Dest_addr,Dest_Seq#>
```

Vários nós na rede podem ter um rota nova para o destino nas suas tabelas de roteamento o que implica na geração de vários RREPs. Por regra, o primeiro RREP é sempre encaminhado ao longo da *rota reversa*, mas o restante dos RREPs podem vir a ser descartados para reduzir inúmeros RREPs circulando dentro da rede MANET. O nó que receber um RREP a mais, encaminha esse RREP somente se ele tiver um maior *dest_seq#* ou tiver o mesmo *seq#* com menor número de *hop_count* para o destino.

O nó de origem pode iniciar o envio de dados para o nó destino assim que for receber o primeiro RREP. Caso souber de uma melhor rota para o destino depois ter recebido outros RREPs, ele pode atualizar a sua tabela de roteamento com as mais novas informações recebidas.

3.3.4 - Estabelecimento da Rota Reversa

Como o RREQ se propaga através da rede MANET, cada nó que receber o RREQ estabelece uma *rota reversa* até o nó origem. O propósito imediato do estabelecimento dessas rotas reversas é a de encaminhar o RREP diretamente para o nó origem. Visto que nem todas as rotas reversas serão usadas, o campo *lifetime* que nela figura é menor do que o encontrado nas rotas RREP. Isso evita que a tabela de roteamento não seja preenchida com rotas inutilizadas.

A Figura 3.1 mostra uma rede MANET com topologia com poucos nós. O nó A deseja descobrir a rota para o nó F e envia um RREQ. Todas as rotas reversas são estabelecidas após propagação dos RREQ.

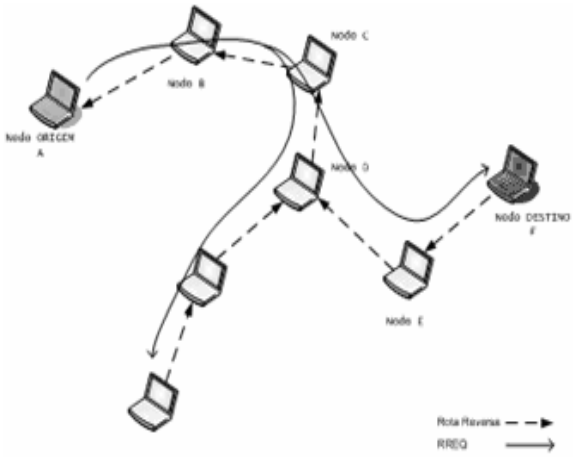


Figura 3.1 - Propagação dos RREQs e estabelecimento da rota reversa entre os nós de origem e destino A e F respectivamente.

3.3.5 - Estabelecimento da Rota Direta

Após ter estabelecido a rota reversa, o RREP trafega até o nó origem pela rota reversa e cada nó intermediário estabelece por si uma rota direta, colocando-a na sua tabela de roteamento para encaminhamento de dados, do nó de origem até o nó de destino. A tabela de entrada da *rota direta* para o próximo salto é estabelecida para o nó vizinho que tiver enviado o route reply e o `<Lifetime,Dest_Seq#>` é definido a partir do RREP.

A Figura 3.2 repete o cenário da Figura 3.1, com a diferença de ter o nó F enviado um RREP ao nó A. O RREP foi encaminhado ao longo da *rota reversa* até o nó A e a *rota direta* de A até F foi estabelecida. As rotas reversas estabelecidas fora do caminho de A e F passaram do tempo de ativação e são removidas da tabela de roteamento; e a *rota direta* é inserida na tabela de roteamento do nó C com *lifetime* da *rota reversa* estendida. Esse é o processo comum dentro do AODV e entre os demais nós que façam parte do caminho da *rota reversa*.

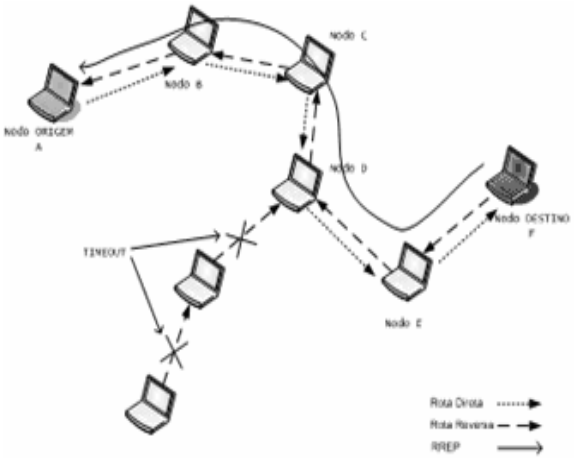


Figura 3.2 - Propagação dos RREPs e estabelecimento da rota direta entre os nós de origem e destino, A e F respectivamente.

3.3.6 - Manutenção de rotas

Quando há falha de um enlace numa determinada rota, os nós que se encontram na parte depois da falha, sentido downstream, invalidam todas as rotas que eram usadas pelo enlace interrompido. Em seguida, os nós enviam um broadcast de *erro de rota* (RERR) para os seus vizinhos, com o *TTL* igual a 1. A mensagem de RERR contém o endereço IP, `<Dest_addr>`, dos nós que se tornaram inalcançáveis devido a falha no enlace. Após recepção da mensagem RERR, o nó procura dentro de sua tabela de roteamento por uma ou várias rotas de destino que lhe permitiam se comunicar com o nó que originou o RERR, listado na mensagem RERR. Caso tais rotas forem existir, elas são invalidadas e o nó receptor da mensagem RERR envia broadcast de uma nova mensagem RERR para os seus vizinhos. Esse processo continua até o RERR chegar ao nó de origem. Uma vez chegada ao nó origem, esse invalida todas as rotas que tinham destino pelo enlace interrompido e inicializa uma nova mensagem de broadcast RREQ caso seja necessário e caso não haja uma rota ativa naquele momento até o nó de destino.

4 - PROPOSTA DE RECONFIGURAÇÃO DINAMICA DE AGENTES MÓVEIS IPv4 EM REDES MANET

4.1 - INTRODUÇÃO

Neste Capítulo, apresenta-se uma nova proposta para o uso dos Agentes do protocolo MIP sendo totalmente implementados dentro das redes MANET [51]. Mostra-se na Figura 4.1 uma visão conceitual da proposta, na qual se tem duas redes MANET implementando a solução proposta, que se chama RDAIPM. Em cada rede encontra-se um agente de mobilidade móvel MHA e MFA. Um nó origem da rede RDAIPM1 quer iniciar uma sessão FTP, por exemplo, com um nó destino da rede RDAIPM2, ver Figura 4.2.

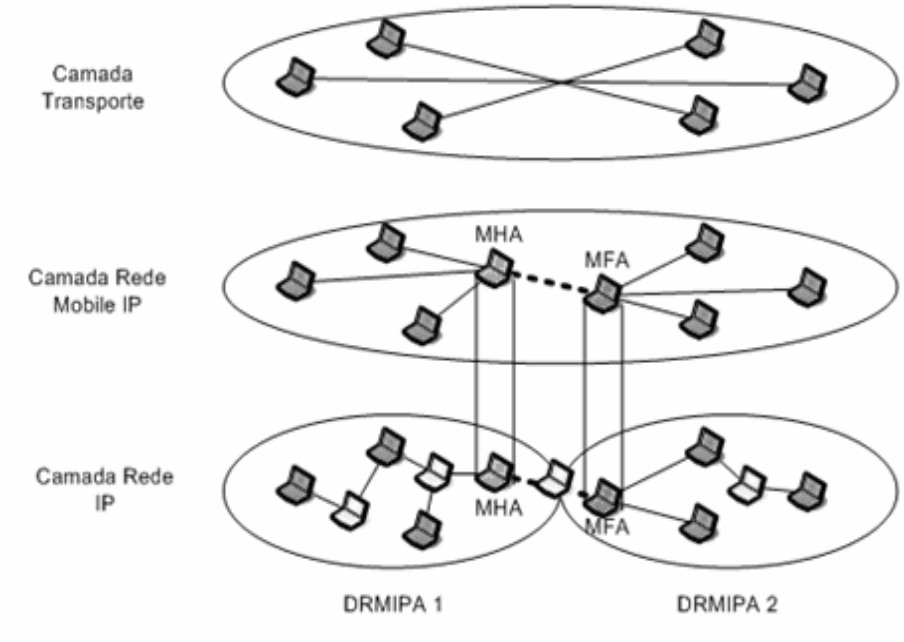


Figura 4.19 - Visão conceitual do RDAIPM.

A mobilidade traz problemas de endereçamento IP que podem ser resolvidos via protocolo DHCP, além do problema de localização e manutenção de roteamento do fluxo em curso do nó móvel, cujas soluções são atendidas pelo MIP. Outro problema será a continuidade da conectividade das camadas de transporte e aplicação, por exemplo, um nó tendo efetuado uma conexão TCP e querendo mantê-la enquanto for se deslocar para outra sub-rede, necessitando, com isso, mudança do seu endereço de origem, terá sua conexão interrompida devido à mudança de parâmetros da sessão de transporte em curso.

As implementações atuais do MIPv4 em redes MANET não suportam o uso de agentes móveis HA e FA para suprir as mesmas funcionalidades quando implementadas em redes infraestuturada. Além disso, não encontra-se soluções no caso acontecer falha e/ou indisponibilidade dos recursos do agente de mobilidade no decorrer de suas transações entre os nós da rede MANET da qual ele faz parte e daqueles que forem visitantes. Nos trabalhos relacionados em [6,8,9,10,11], foram propostos várias abordagens visando o estudo de recuperação de falha dos agentes HA e FA. O trabalho de [6], em particular, trata justamente da redundância no uso de agentes de mobilidade na rede infraestuturada, caso um deles tenha o seu funcionamento alterado durante o tempo de atividade. O problema é que esses agentes cooperam entre si de maneira estática, i.e, sem mobilidade nenhuma. A proposta de [10,11] aborda o mesmo assunto fazendo uso de vários agentes de mobilidade servindo de apoio ao agente principal caso ocorra uma falha, mas seu foco também é em redes infraestuturadas.

Esta tese tem como proposta a utilização do protocolo MIPv4 em redes de comunicação do tipo Ad doc onde é admitida a mobilidade de todos os nós, inclusive dos próprios Agentes de Mobilidade (agora sendo móveis): HA e FA, que são reconfigurados dinamicamente dentro da rede de acordo com a necessidade administrativa. Para isso, são discutidos os problemas decorrentes dessa mobilidade e propostos algoritmos para a reconfiguração automática dos agentes, dentro de um ambiente de micro-mobilidade, de modo a garantir a continuidade dos serviços associados ao MIP para todos os nós que deles se utilizam dentro da faixa de alcance da rede MANET. Essa solução permitirá ao mesmo tempo a recuperação de falha dos agentes de mobilidade no decorrer de suas transações entre os nós da rede MANET já que haverá sempre um agente de mobilidade móvel servindo de reserva.

A seguir, serão discutidos os problemas de integração entre o MIP e o AODV em redes MANET sendo propostos algoritmos para a efetiva implementação da reconfiguração dinâmica dos Agentes MIP.

No contexto da conectividade sem-fio, o conceito de redes Ad hoc apareceu com características temporárias e mecanismos de configuração automática entre os nós móveis. As redes Ad hoc tornaram possível a comunicação direta entre os nós móveis,

dentro do alcance de sua área de trabalho, eliminando a necessidade de uma infra-estrutura fixa conhecida, como aquelas utilizadas em WLANs empregando AP ou estações rádio base.

As redes Ad hoc possuem grande flexibilidade e responsividade após sua implementação, o que as tornam extremamente adequadas em muitas aplicações, tais como em ambientes hostis vistos em cenários militares de batalha, situações de desastre natural, conferências de trabalho ou estudo, pesquisas de campo, redes de sensores, entre outros. Entretanto, pode haver a necessidade dos nós móveis, em sua nova rede, fazerem uso de uma conexão com a Internet e ainda serem alcançados por sua *Home Network* de uma rede fixa ou de outra rede Ad hoc distante. Em vista desse assunto, o grupo de pesquisa sobre *MIP* contemplou a necessidade de um móvel conectado a sua *Home Network* ter a habilidade de se mover para outras redes (chamadas de *Foreign Networks*) mantendo seu *Home Address* (endereço lógico obtido na *Home Network*) e tendo a possibilidade de manter os serviços que ele possuía na sua rede local original. Para solucionar tal questão, dois endereços IP devem ser atribuídos ao nó móvel de maneira que ele possa ser alcançado na *Home Network* por um endereço IP permanente e em uma *Foreign Network* através do chamado COA, que representaria seu novo ponto de conexão fora de sua rede original, como descrito na RFC-3222 [3].

As implementações atuais do MIPv4 em redes MANET não suportam o uso de agentes móveis HA e FA para suprir as mesmas funcionalidades quando implementadas em redes infraestuturadas. Até então, não existe uma solução quanto ao que poderia acontecer em caso de falha ou indisponibilidade dos recursos do agente de mobilidade no decorrer de suas transações entre os nós da rede MANET da qual ele faz parte e daqueles que forem visitantes.

Dadas as restrições anteriores, apresenta-se uma proposta que permite mobilidade dentro do cenário *MIPv4* onde os agentes móveis HA e FA são ambos móveis em uma MANET. Foram introduzidos novos agentes *MIP*, denominados agentes ativos e passivos, em que o agente ativo é o real responsável pela provisão do serviço, enquanto o agente passivo se encontra num estado ocioso de espera. O sistema desenvolvido é denominado RDAIPM e suas principais contribuições são:

- A inserção dos agentes móveis (HA e FA) dentro de uma rede Ad Hoc para se obter a mobilidade dos mesmos, num cenário de micro-mobilidade, onde o deslocamento dos nós é realizado dentro de uma área de abrangência pequena, restrita a uma única rede como, por exemplo, o campus de uma universidade. Conseqüentemente, as funcionalidades dos agentes MIP serão realocadas de uma rede fixa cabeada conectada a um AP para agentes MIP móveis em uma MANET. Esta técnica irá contribuir para a implementação de um mecanismo de recuperação de falhas para o agente ativo.

- A eleição de um agente passivo (HA e FA) para prover o mecanismo de recuperação de falhas se o agente ativo deixar a rede ou finalizar a sua participação. Com isto, os nós móveis serão supridos com conexões de rede contínuas seja na *Home Network* ou na *Foreign Network.*

- A proposta de novos algoritmos para assegurar a efetividade da mobilidade dos agentes e sua reconfiguração dinâmica.

- A extensão do *MIP* na versão 4 com objetivo de possibilitar a interação de múltiplas MANETs com uma arquitetura MIP livre de falhas por parte dos agentes.

4.2 - TRABALHOS RELACIONADOS

Nesta sessão, faz-se um levantamento dos trabalhos relacionados ao MIP, tanto nas versões *IPv4* e *IPv6* [4]. Apesar de permitir a mobilidade na camada de rede em qualquer ponto de conexão da Internet, e que tenha os agentes de mobilidade IP configurados, nenhuma implementação do *IPv4* e *IPv6* tem estudado esses agentes num cenário de mobilidade Ad Hoc. Nesse contexto, esses nós se tornariam agentes de mobilidade móveis e não fixos como na rede infraestruturada atual.

4.2.1 - IP Móvel e Mobile Network

Quando se faz referência a uma rede com *MIP*, fala-se de quatro componentes principais: o MN, HA, FA e CN. O *MIPv4* e *IPv6 (esse último não trás o agente de mobilidade FA)* permitiram a introdução da mobilidade para os *mobile nodes* com relação aos seus agentes de mobilidade HA e/ou FA. Com a ajuda desses agentes, o MN pode manter o seu endereço IP mesmo estando em uma rede estrangeira,

mantendo ativa a sua conexão com o servidor ou usuário da Internet. Todos os mecanismos associados ao funcionamento do MIPv4 estão descritos no item 2.3.

4.2.2 - Hierarchical Mobile IPv6

O HMIPv6 (*Hierarchical Mobile IPv6*), ou *IPv6 móvel hierárquico*, permite a micromobilidade dos MNs dentro de determinado domínio da rede Internet [80]. Esse processo evita o registro freqüente do nó MN em seu agente de mobilidade HA. Com isso, uma redução do tráfego no domínio é introduzida.

Mas essa solução não discute os possíveis problemas que seriam gerados pela perda do agente de mobilidade HA. A proposta do HMIPv6 também não visa a mobilidade dos seus agentes de mobilidade.

4.2.3 - Mobile IPv4 for Mobile Adhoc NETwork

O MIPMANET (*Mobile IPv4 for Mobile Adhoc NETwork*) trás a proposta de integração de uma rede Ad Hoc com o MIPv4 [45]. A proposta visa também a conexão dos MNs na rede Ad Hoc com o seu agente de mobilidade IP através de um Gateway de acesso à Internet. Dessa forma, o sistema apresentado permitiria ao nó móvel de entrar e sair em uma rede Ad Hoc.

O MIPMANET, em contrapartida, não trata da inclusão de um agente de mobilidade IPv4 dentro da rede Ad Hoc. Nenhum estudo com relação à recuperação de falha desses agentes e, portanto, de uma possível mobilidade na rede Ad Hoc inteira, implementando o MIP, é apresentada.

4.2.4 - NETwork Mobility support in IPv6

O Network Mobility (*NEtwork Mobility support in IPv6*) pretende ser utilizado para redes móveis a fim que elas possam se conectar em diferentes pontos da rede Internet. O protocolo é uma extensão do MIPv6 e permite a continuidade da sessão entre todos os nós da rede enquanto ela for se deslocando através da rede Internet. Esse protocolo também permite que cada nó em sua rede seja alcançável pelo resto dos nós da rede Internet. Ele faz uso de uma entidade denominada Mobile Router (MR) que permite transação de dados entre a sua rede e a Internet [56].

Diferentemente da proposta aqui apresentada, o NEMO não considera a possibilidade de que o MR seja um roteador móvel ou que ele possa ser desconectado da rede a qual ele é responsável. Isso porque o NEMO não é direcionado a redes Ad Hoc como o caso aqui estudado. Ou seja, apesar da presente proposta mencionar especificamente a Reconfiguração Dinâmica dos Agentes MIPv4 dentro da rede Ad Hoc, a rede RDAIPM acaba sendo também uma Rede com Mobilidade quando se desloca em grupos entre pontos de conexões da rede Internet. E para finalizar, todos os nós são roteadores, ao contrário do NEMO, que tem unicamente um roteador chamado MR.

4.2.5 - Cellular IP

O Cellular IP [63] é uma proposta que oferece a mobilidade e handoff para nós cuja mobilidade é freqüente numa região de micro-mobilidade, como num campus universitário, por exemplo. Os nós sem fio recebem e enviam os datagramas IP dentro da rede cellular. Por essa razão, o Cellular IP pode interagir com o MIP [3] para dar suporte à mobilidade entre redes Cellular IP distintas.

Como se pode ver, a proposta do Cellular IP não apresenta nenhuma solução quanto à inclusão dos agentes de mobilidade IP dentro da rede cellular. É mais um cenário no qual o RDAIPM poderia dar suporte para permitir a mobilidade e reconfiguração dos *agentes de mobilidade* caso a rede cellular não permita a inclusão de uma rede MIP infraestruturada.

4.2.6 - Comparação de algumas das propostas

Das propostas citadas na Tabela 4.1, nenhuma delas faz menção de um nó MIP se tornar um *Agente de mobilidade móvel*. Também não é citado a possibilidade desse *Agente* se desconectar de sua rede e assumir outras funcionalidades, por exemplo a de um MN. As três últimas implementações mostradas na Tabela 4.1 são as que mais se aproximam da solução RDAIPM para redes Ad Hoc IEEE 802.11, mas de novo, sem permitir a mobilidade e reconfiguração dos agentes.

Assim, a proposta desta tese visa mudar o conceito de *agentes de mobilidades* como sendo unicamente fixos, para torná-los móveis. Os algoritmos aqui propostos irão contribuir na mobilidade desses agentes de tal forma que novas redes possam ser implementadas dinamicamente com o protocolo RDAIPM em redes wireless Ad Hoc.

Tabela 4.1 – Propostas relevantes dos trabalhos relacionados.

PROPOSTAS RELEVANTES	STATUS	MOBILIDADE EM ÁREA DENSA	RECONFIGURAÇÃO DINÂMICA DOS AGENTES COM MOBILIDADE
Mobile IPv4 [3]	RFC-3222 *(Janeiro 2002)*	Não escalável	Não
Mobile IPv6 [81]	RFC-3775 *(Junho 2004)*	Não escalável	Não
CELULLAR IP [63]	Internet-Draft *(expirou Junho 2000)*	Sim	Não
INRIA HMIPv6	RFC-4140 *(Agosto 2005)*	Sim	Não
MIPMANET [45,67]	-	Não escalável	Não
NEMO [56]	RFC-3963 *(Janeiro 2005)*	Sim	Não

4.3 - PROBLEMAS DE PERDA DE CONECTIVIDADE DOS AGENTES NO MIP E PROPOSTA DE SOLUÇÕES

Nas redes MANET, não existe uma rede infraestuturada que possa permitir um maior controle dos seus nós quando esses forem se deslocar de uma rede para outra, mantendo o acesso aos serviços que eles tinham na rede MANET de origem na rede MANET de destino (ou estrangeira) ininterruptamente, ou de forma transparente como oferecido pelo protocolo MIP.

A Figura 4.2 apresenta o sistema sugerido possuindo um caminho virtual entre os nós de origem e destino (o nó móvel de destino representado era originalmente da RDAIPM1). Os dados deixam a origem passando pelo MHA e MFA ativos; ambos servirão como as extremidades do túnel até o nó de destino desejado. Quando fora da *Home Network,* todas as transações de dados entre os MNs precisam ser entre os agentes ativos, de modo que os mecanismos MIP se comportem da mesma maneira quando eram parte de uma topologia WLAN convencional. O processo de roteamento é suportado pelo protocolo de roteamento usado, nessa proposta usa-se o AODV, por todo o caminho da MANET.

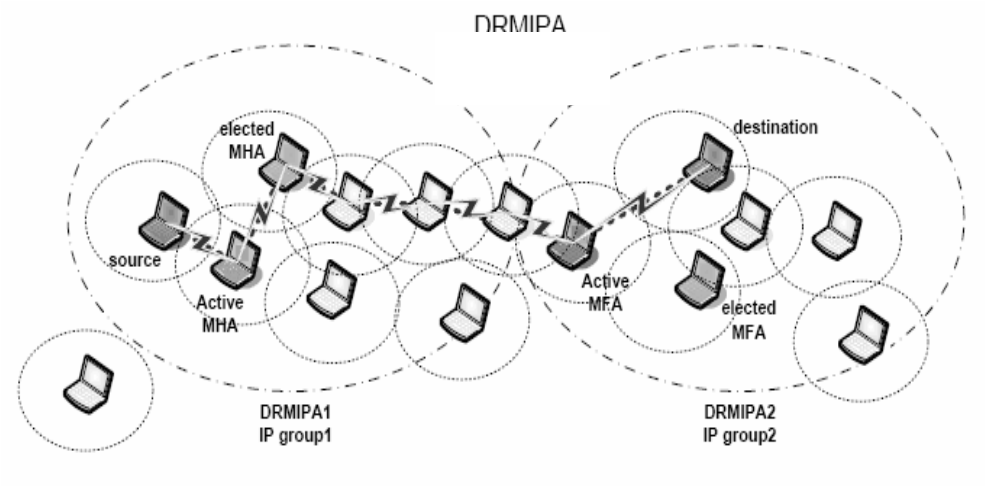


Figura 4.20 - Novo MIPv4 em MANET.

A necessidade de se ter um acesso à Internet para esses tipos de nós foi estudada em vários trabalhos de pesquisas, tais como o MIPMANET [45] no qual os nós MIP podem ter acesso à Internet através de Gateways, sendo eles agentes FA, que implementam tanto os protocolos MIP quanto algum protocolo de roteamento, como o AODV. Daí se originou parte da proposta desta tese de se integrar o MIP e protocolos de roteamento, como o AODV, para reconfiguração dinâmica dos agentes de mobilidade em redes MANET, permitindo o registro dos nós que entram e saem de sua área de cobertura, sem os HA e FA mantidos na rede infraestuturada, e sim dentro de cada rede MANET.

Diferentemente do que foi proposto em [31, 35, 45, 47], aqui se ressalta o uso de Agentes Móveis (MA) MIP, tendo a possibilidade de se deslocar dentro da área de cobertura MANET de origem. Em tais condições, os serviços previamente associados ao MHA e MN devem ser mantidos quando o MN estiver fora de sua rede de origem. No caso de uma falha no MHA, os serviços disponibilizados ao MN seriam afetados.

Uma solução para tal problema seria a duplicação dos MAs dentro de cada rede MANET garantindo ao nó acesso aos serviços a ele providos pelo MFA e MHA. Sendo assim, cada nó nessa nova abordagem se tornaria possível candidato para ser um futuro MA. Por isso, a proposta de ter-se um *MA ativo* e um *MA passivo* cujos mecanismos de seleção serão discutido na sessão 4.3. Dessa forma, sempre haverá um MA disponível para os demais nós da rede.

Entre outros problemas associados à coordenação e funcionamento de ambos protocolos aqui utilizados, MIP e AODV, o trabalho realizado em MIPMANET revela fatores aos quais a presente tese também terá de lidar, tais como:

- No ambiente MIP, o foreign agent e o nó visitante móvel sempre mantém uma conexão através da camada de enlace. A rede MANET, por sua vez, permite uma conectividade por saltos, ou seja, sem haver a necessidade de ter uma conexão de enlace entre um nó de origem e de destino para envio e recepção de pacotes. Assim sendo, tem-se um dos problemas de se incorporar o MIP dentro da rede MANET. Os mecanismos de encaminhamento dentro dessa rede devem ser revistos, os nós e os agentes MIP têm de fazer uso do protocolo de roteamento da rede multihop para poderem trocar mensagens de sinalização e de dados como ocorre na rede infraestuturada.

- O *broadcast* das *mensagens de anúncio* do MIP numa rede WLAN infraestuturada é menos danoso quando comparada ao cenário MANET. Isso porque o *broadcast* na infraestuturada é realizado somente no enlace ao qual todos os nós estiverem conectados. Mas, nas redes MANET, a propagação de tais mensagens é feita de nó a nó, i.e, o nó que for receber um Flooding tem de encaminhá-lo para os(s) nós(s) vizinhos e isso até chegar aos últimos nós da área de cobertura. Uma provável solução seria diminuir o envio desse Flooding não impactando o mecanismo de funcionamento do MIP, ou então fazer o *unicast* dessas mensagens para o nó que for solicitar o serviço do MA. Como a proposta é trabalhar com redes MANET totalmente MIP, o uso do *Flooding* é preferível, junto com o aumento do tempo entre cada Flooding.

- O protocolo AODV sempre inicia o roteamento de dados de uma fonte de origem `<Source_addr>` para uma fonte de destino `<Dest_addr>` da rede. Tomando a Figura 4.1 como exemplo, um MN da rede RDAIPM1 tendo se deslocado até a rede RDAIPM2 e querendo manter um serviço TCP oferecido por outro MN da RDAIPM1, terá os seus datagramas perdidos pelo fato do AODV rotear os pacotes entre ambos MNs sem passar pelos MAs (*MFA* e *MHA*). Isso inviabiliza o funcionamento dos MAs em tal ambiente

MANET. A solução proposta, igual à apresentada em [45] para descoberta de *Gateways* a fim de prover acesso a Internet, é de usar rotas `<rt_DEFAULT>` para encaminhamento dos dados referentes aos nós MIP.

Dessa maneira, os MHA e MFA seriam os default gateways para cada nó MIP e os mecanismos de tunelamento seriam novamente incorporados independentemente de quantos nós estiveram entre o grupo `{<MN,MHA>;<MHA,MFA>;<MFA,MN>}`.

4.3.1 - IP Móvel em redes Ad Hoc

A proposta MIP atual consiste em suportar a mobilidade IP em diferentes ambientes. O HA, sendo um roteador localizado na *Home Network*, provê informação de localização de serviço para o nó móvel, quando o mesmo se encontra fora de casa, e funciona como uma das pontas do túnel a ser formado durante a transação de dados. Localizado na sua *Home Network*, o nó móvel usa seu *home address* para receber e enviar dados para outros nós em sua área (atuando como nós correspondentes). Ao se mover para uma rede estrangeira, o nó recebe um COA na sua nova localização e estará apto a manter sua comunicação de dados enquanto em *roaming*, Figura 4.2. Para facilitar a transação de dados entre o HA e o MN, um FA é usado como a outra ponta do túnel durante o processo de envio das mensagens *registration request* e *registration reply* e durante o intercâmbio de dados.

O protocolo de roteamento a ser utilizado nesta tese para análise de desempenho é o AODV [6], por ser mais confiável e de melhor escalabilidade que o DSDV. O AODV é um protocolo reativo que utiliza mensagens *Route Request* (RREQ) e *Route Reply* (RREP) para descoberta de rotas entre os nós Ad hoc.

Nas redes Ad hoc não há uma infra-estrutura que permita um grande controle dos MNs. Quando eles se movem de uma rede IP para outra, seus serviços correntes deveriam permanecer disponíveis quando requisitados na rede Ad hoc estrangeira, de maneira similar ao que acontece numa rede infraestuturada. Para isso, o MIP e a MANET deveriam ser combinados para permitir o registro de qualquer nó móvel entrando e saindo de sua área de propagação e forçando todas MANETs a implementar um *Mobile Home Agent* e um *Mobile Foreign Agent* (chamados nesta

tese de MHA e MFA), para estabelecer a conectividade dos nós. A Figura 4.3 apresenta uma rede infraestuturada com implementação do protocolo MIPv4, sendo a topologia mais utilizada para quem for implementar a mobilidade na camada 3 dentro de sua rede privada [1,3,5,7-12,14,18-21,35,45].

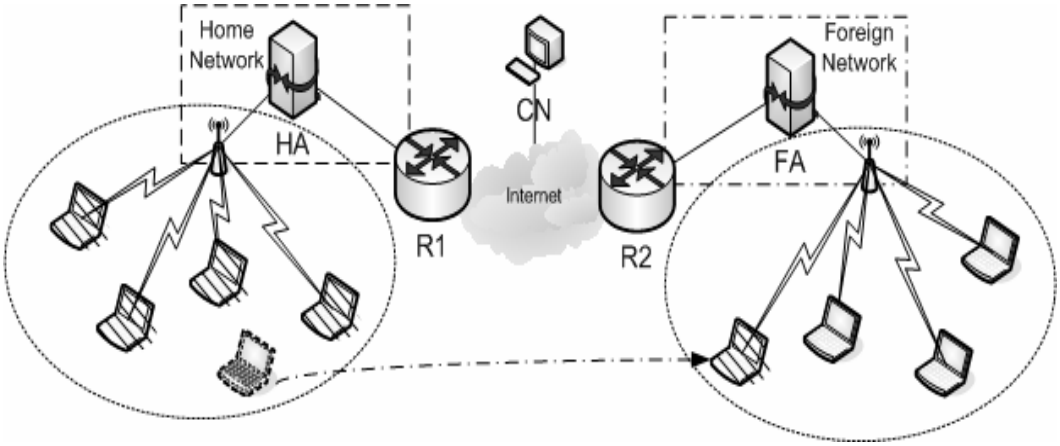


Figura 4.3 - Estrutura MIPv4 atual.

Na Figura 4.4, tem-se uma primeira visualização do que seria o MIPv4 na MANET: duas redes móveis implementando RDAIPM (*possuindo agentes de mobilidade ativos e passivos*) e se comunicando pela Internet, através de acesso a *gateways*. Como podem ser percebidos, os agentes de mobilidade estão completamente alocados dentro da MANET [51,61,62,64,65]. Os mecanismos de configuração e eleição desses agentes serão apresentados no item 4.6.

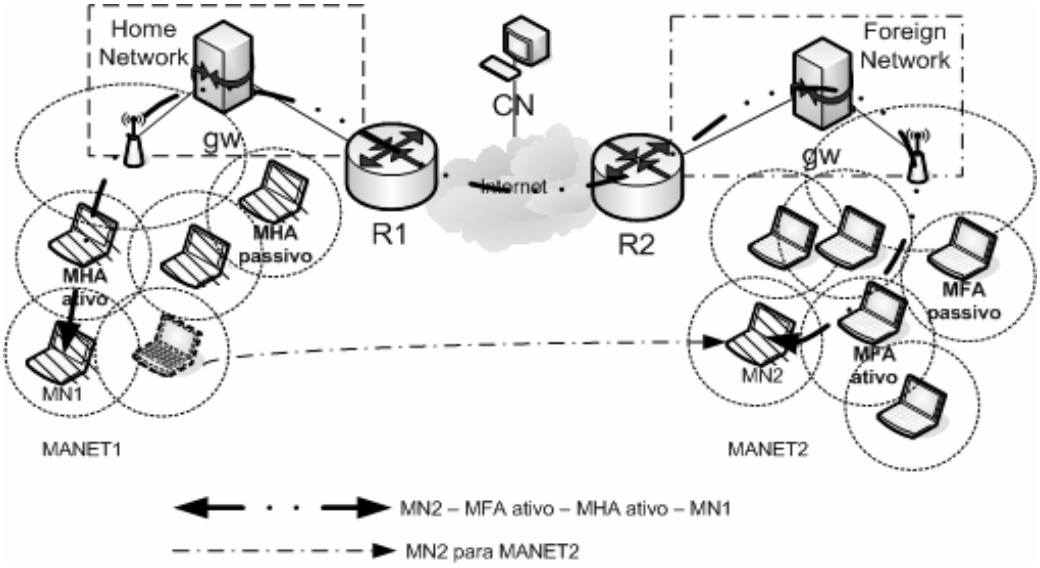


Figura 4.4 - Novo MIPv4 em MANET com acesso a gateways.

4.4 - PROPOSTA DE SOLUÇÃO PARA RECONFIGURAÇÃO DINÂMICA DE AGENTES MÓVEIS

A rede MANET e sua representação em grafo planar é apresentada na Figura 4.5, [51]. Pode-se aplicar, nesta rede, um grafo G tendo um conjunto de vértice (nós) *V* com arestas (enlaces) *E* [77] e assim, mostrar que sempre haverá um Agente ativo e passivo dentro da rede.

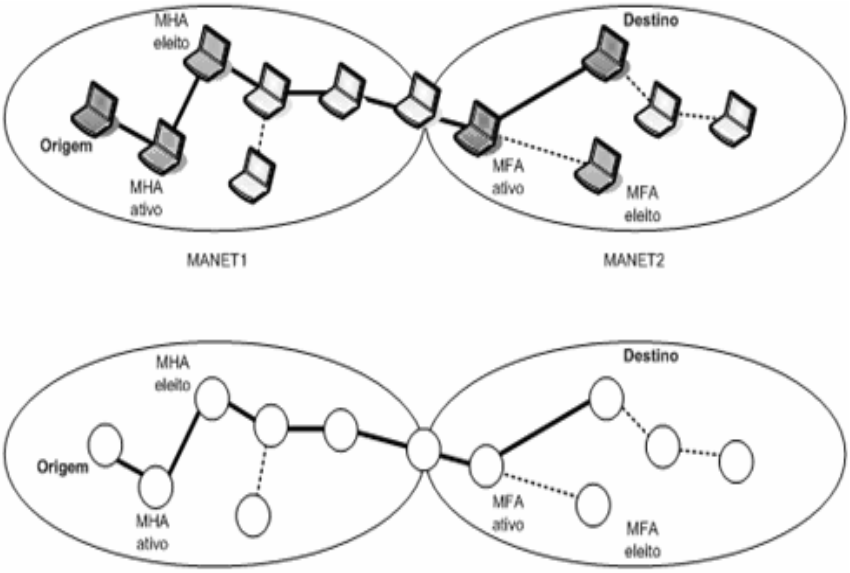


Figura 4.5 - Eleição de um agente móvel na rede MANET.

Seja um conjunto de nós $V = \{v_1, v_2, v_3, ..., v_n\}$, Figura 4.5, onde cada nó MIP $\in V$ e as arestas *E* entre nós seriam um conjunto combinatório tal que $\{v_1, v_2\} \Leftrightarrow \{MA, MN\}$ onde, por e.g v_1 representa um MA, tal como o MHA, e v_2 um MN.

A seqüência de enlaces que vão do nó origem ($SH_{n'}$), passando pelos MHA e MFA ativos até o nó de destino (DH_n), onde *n* é o *n*-iésimo nó da rede MANET2 no caminho reconhecido como rota ativa [43], pode ser representada como *m*-enlaces (neste caso 7-enlaces entre origem e destino) no caminho da rota ativa do protocolo AODV após disseminação de mensagens RREQ e RREP.

Assim, o caminho é estabelecido desde o `<Source_addr>` $n' \in V'$ ($SH_{n'}$) para o `<Dest_addr>` $n \in V$ (DH_n) onde teria-se $SH_{n'} \Leftrightarrow MN_{n'}$ e $DH_n \Leftrightarrow MN_n$. Esta

proposta mostra que qualquer nó da rede MANET que estiver na rota ativa, dada pela equação 5.1, é considerado um candidato para ser *Agente Móvel Passivo*.

$$P = [\{SH_{n'}, v_2\}, \{v_4, ..\}, \{.., DH_n\}] \tag{4.1}$$

Nesta tese, são eleitos os nós que fazem parte da rota ativa mais recente e que sejam alcançáveis em um único salto, `<TTL=1>`, a partir do MA ativo. Então estendendo a relação entre caminhos e nós, um nó MIP de uma rede estrangeira poderia fazer parte da *n*-iésima rede MANET como mostrado na Figura 5.5.

Como visto anteriormente, um túnel tem de ser estabelecido para o envio e a recepção dos datagramas entre nós móveis da rede MANET tendo como *gateways* os MFA e MHA. Um novo caminho P^1 surgiria na rede local sendo,

$$P^1 = [\{SH_{n'}, v_2\}, \{v_4, ..\}, ..., \{.., MHA\}] \tag{4.2}$$

onde $SH_{n'} \Leftrightarrow MN_{n'}$ é o nó de origem e MHA é o agente móvel HA como início do túnel.

Em seguida, um túnel entre MAs seria criado no caminho P^2 dado pela

$$P^2 = [\{MHA, ..\}, \{v_n, v_{n'}\}, \{.., MFA\}] \tag{5.3}$$

onde MHA é o agente móvel HA como início do túnel e MFA é o agente móvel FA como final do túnel.

Um último caminho P^3 inicia a partir do final do túnel criado entre o MHA e MFA sendo pela equação

$$P^3 = [\{MFA, v_2\}, \{v_4, ..\}, ..., \{.., DH_n\}] \tag{5.4}$$

onde $DH_n \Leftrightarrow MN_n$ é o nó de destino e MFA é o agente móvel FA como final do túnel.

A rota ativa P_a seria, então, a soma dessas rotas parciais passando pelos MAs, como pode ser vista pela equação

$$\begin{aligned} \mathrm{P}_a &= \sum_{i=1}^{n} P^n \\ &= P^1 + P^2 + P^3 \\ &= [\{SH_{n'}, v_2\},..,\{MHA,..\},\{..,MFA\},..,\{..,DH_n\}] \end{aligned} \tag{5.5}$$

Pode-se concluir que, em caso de existência de um conjunto de nós em um caminho ativo MANET, sempre haverá dois nós em que um deles será um MA considerando que tenham conectividade entre eles.

Na presente tese, faz-se uso das mensagens do protocolo AODV a fim de garantir a correta entrega das disseminações dos *agent advertisements*. Isso porque a atual versão do AODV implementado no software de simulação NS2, usado para análise de desempenho [19], não estabelece rotas reversas entre nós da rede MANET, depois do Flooding do protocolo MIP. Essa funcionalidade foi incorporada no simulador de rede para garantir o correto funcionamento do mesmo, criando um tipo de "*tunelamento*" entre o MA e o(s) MN(s) para futuras operações de registros [45].

Para garantir uma correta interação entre os agentes ativo e passivo, uma Flag A foi incluída no campo *reserved* da mensagem *agent advertisements* original [ver Apêndice A e B]. Essa flag facilita a análise da situação de atividade ou falha entre ambos agentes. A Flag A com nível '11' indica que o agente ativo está ativo e que os nós RDAIPM podem enviar suas requisições para participar da eleição de *agente passivo*.

Caso o agente ativo deixe de enviar os *agent advertisements*, ele mesmo deixa de suprir as funcionalidades de agente ativo e passa a ser um nó comum. A não recepção periódica dos *agent advertisements* pelo agente passivo após determinado tempo pode ativar o processo de mudança de estado do agente passivo para ATIVO, sendo esse tempo relacionado ao *lifetime* da última mensagem recebida. Um campo de 32 bits foi incluído na mensagem [ver Apêndice A e B] para que o agente passivo saiba do verdadeiro endereço do agente ativo que está disseminando os *agent advertisements*, a fim de usá-lo como endereço de destino no campo de 32 bits reservado para tal no

cabeçalho do protocolo IP. Esse processo viabiliza a troca de mensagens entre *agentes ativo e passivo* caso haja mais de um agente ativo implementado na rede MANET. Apesar de não ser o foco desta tese, o uso de mais de um agente ativo permitiria um balanceamento de carga da rede aliviando a posteriori os processos realizados em cada agente ativo para atender as requisições dos nós MIP.

Propõem-se uma nova mensagem de associação (*binding*) do endereço IP do agente ativo com o(s) endereço(s) dos nós MIP que nele tenham se registrado. O agente ativo coloca os endereços IP dos nós visitantes dentro da mensagem de *binding*. Essa mensagem é enviada ao agente passivo para atualização da tabela dos nós cadastrados pelo agente ativo. Caso o agente ativo esteja inoperante e o agente passivo tenha passado do estado passivo para o estado ativo, envia via *unicast* as mensagens de *agent advertisements* para os nós de sua nova lista e dessa forma evita uma inundação (*flooding*) dessas mensagens através da rede MANET. Se um novo nó desejar se registrar na rede MANET, a metodologia descrita no item 2.3.2, do capítulo 2, quanto à descoberta de agente de mobilidade, seria usada. A principal modificação da mensagem é constituída por um novo campo `<LENGHT>` de um byte, o endereço IP do agente ativo de quatro bytes e o(s) endereço(s) de nós MIP de quatro bytes cada [ver Apêndice A e B]. Para transmissão segura dessa mensagem, um campo *Identification* de 32 bits é incluído para comparar e validar a trocar de mensagens entre agentes ativo e passivo já definido em [3,14].

4.5 - DIAGRAMA DE SEQÜÊNCIA PARA CONFIGURAR OS AGENTES MÓVEIS ATIVO E PASSIVO

Em função das características não estruturadas de uma rede MANET, é esperado que a mesma possa conter desde um até uma quantidade arbitrária de nós, e que as funções de MA (MHA e/ou MFA) possam ser assumidas por qualquer estação móvel que entre na área de cobertura dessa rede e tenha implementado o MIP. É feita inicialmente uma configuração estática dos MA (*MApri* ou agente primário) em cada rede que tiver o protocolo MIP implementado a fim de ter iniciado a eleição do MA secundário (*MAsec* ou passivo) caso mais de um nó com MIP esteja presente na rede MANET.

Por determinação administrativa, Figura 4.6 [79], é escolhido o nó MIP que for servir de MA ativo dentro da rede em questão, caso já não esteja presente. São instaladas as configurações do MIP contendo as novas mensagens RDAIPM.

Uma vez configurado o MA primário e tendo outros nós MIP presentes na rede MANET, o agente ativo verifica se tem um agente passivo previamente eleito na rede e o coloca em espera. Caso contrário, inicia o algoritmo de eleição de agente passivo. Durante as transações de mensagens *agent advertisements* entre agente ativo e passivo, verifica-se a atividade do agente ativo com o auxílio da flag A discutida anteriormente (ver item 4.3).

Em caso de falha ou inatividade do agente ativo, o algoritmo para mudança do estado passivo para ativo do agente passivo é acionado. O novo agente ativo tem de repetir os processos anteriormente citados para procura de um novo agente passivo. Supõe-se que os novos nós MIP da rede MANET tenham implementados o RDAIPM, para permitir os processos acima descritos, mesmo sem terem sido escolhidos pelo agente ativo.

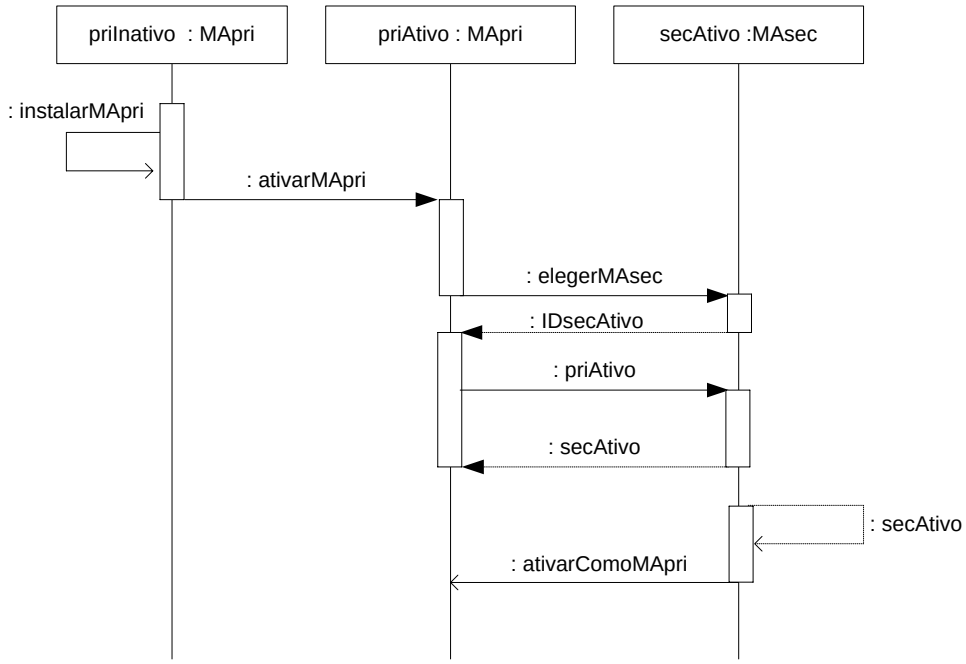


Figura 4.6 - Diagrama de seqüência para configurar os agentes móveis ativos (*MApri*) e passivo (*MAsec*).

4.6 - DIAGRAMA DE SEQÜÊNCIA PARA ELEGER UM AGENTE PASSIVO

O nó sendo agente ativo (primário ou *MApri*) tem de procurar na sua tabela de cadastro por nós MIP que sejam possíveis candidatos à eleição do agente passivo usando o `<table_Nd_Lookup>`, Figura 4.7. Esse cadastro é montado à medida que os nós forem se registrando através do envio de mensagens registration requests vista no Capítulo 2.

Se pelo menos um nó estiver cadastrado no agente ativo e ou pelo menos um nó estiver presente nas mais recente rotas ativas, criadas pelo protocolo de roteamento, nesta tese o AODV, procura-se pela rota que tiver o menor número de seqüência `<DSN>` e menor número de saltos `<HOP_COUNT>` até ele. Nesta tese, usa-se um `<HOP_COUNT=1>` para escolha do agente passivo que esteja o mais próximo possível do agente ativo. Esse processo é tido como base na hora da eleição. Depois serão escolhidos aqueles que tenham o mesmo endereço de rede do agente ativo.

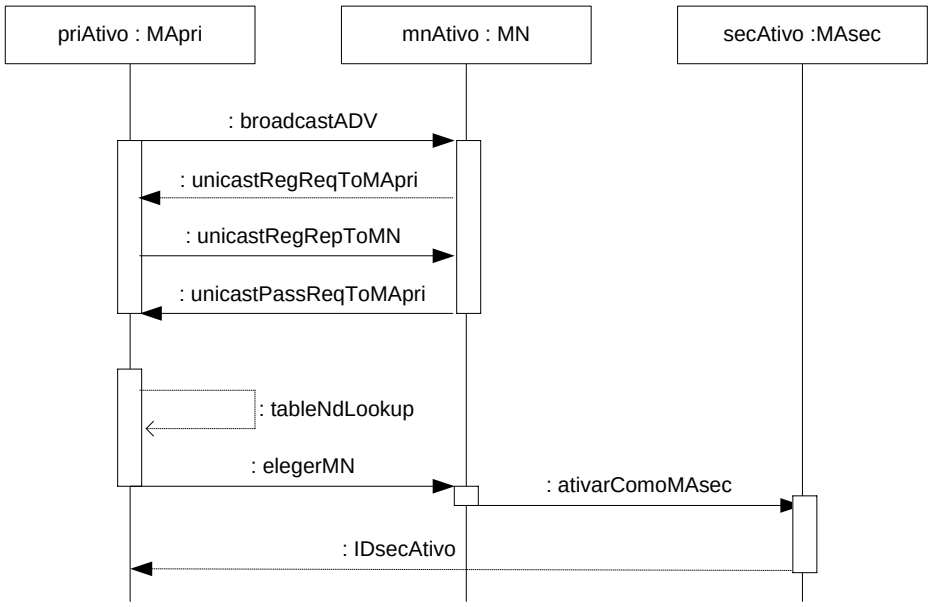


Figura 4.7 - Diagrama de seqüência para eleger um agente passivo.

Se o melhor nó MIP for encontrado, obedecendo aos parâmetros anteriores, o agente ativo o elege como agente passivo. Monta-se a tabela de cadastro para eleição dos agentes passivos que poderia ser consultada para futuras escolhas e assim, diminui-se o tempo usado para eleição desses agentes, se um deles fosse deixar a rede MANET repentinamente. Caso contrário, procura novamente pelo melhor nó MIP cadastrado

no agente ativo, tirando os processos de eleição dos nós MIP que seriam candidatos à agente passivo.

O protocolo MIP se encarrega das mensagens de registros dos nós MIP na rede MANET e a escolha das melhores rotas é realizada através do protocolo de roteamento.

4.7 - DIAGRAMA DE SEQÜÊNCIA PARA DETECÇÃO DA QUEDA DO AGENTE ATIVO E MUDANÇA DO AGENTE PASSIVO PARA AGENTE ATIVO

Após implementação e eleição do novo agente passivo, esse mesmo fica de aviso para assumir a posição de agente ativo, Figura 4.8.

A seguir tem-se uma explicação do fluxograma mostrado na Figura 4.8. O agente passivo fica recebendo a disseminação dos agent advertisements na rede verificando os campos <TYPE>, <CODE> e <FLAG_A> [ver apêndice B].

Se o agente ativo ainda estiver no estado ativo, o agente passivo atualiza a sua tabela de agentes ativos na tabela de roteamento do AODV com o <rt_resolve> e <rt_lookup>.

Caso contrário inicia o algoritmo de mudança de primário para secundário do agente passivo [item 4.6].

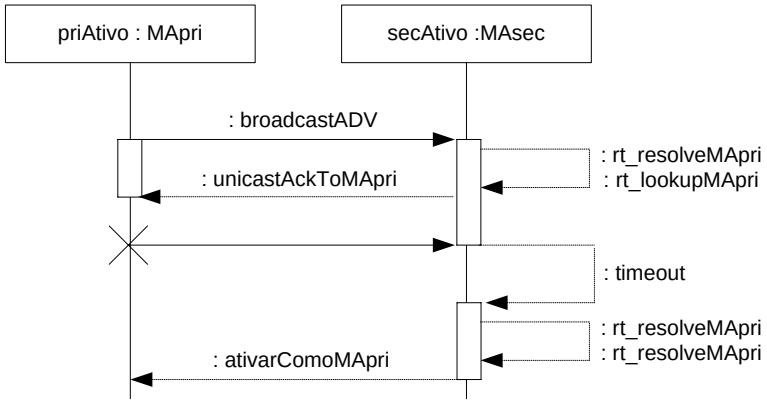


Figura 4.8 - Diagrama de seqüência para detecção da queda do agente ativo e atualização da lista de visitantes.

4.8 - SINALIZAÇÃO DE MUDANÇA DO AGENTE ATIVO PARA AGENTE PASSIVO

Caso o MA passivo esteja a vários saltos do MA ativo, as mensagens de sinalização do RDAIPM têm de ser encaminhadas através dos nós intermediários até chegar nele, via roteamento. Essas mensagens são os *agent advertisements* <agt_adv>, modificados nesta proposta, para facilitar o correto funcionamento da sinalização RDAIPM [ver apêndice B]. O MA que for naquele momento ativo tem de enviar os <agt_adv> com a <FLAG_A=11>. Essa mensagem é processada por cada nó da rede MANET com MIP e usada unicamente pelo nó a fim de saber para qual MA enviar futuros requerimento de registros.

Quando a mensagem for recebida pelo MA passivo, este identificará o campo contendo a <FLAG_A>. Se ela for igual a 11, significa que o agente ativo não tem problemas, caso contrário ele terá de assumir como agente ativo. Vale lembrar que nesse processo os <agt_adv> podem revelar, através das flags H e F, que a mensagem provém de um *Home agent* ou *Foreign Agent*.

A Figura 4.9 mostra como os processos de sinalização ocorrem unicamente entre os nós MN e o MA ativo e não com o MA passivo [51]. Caso ocorra uma falha no MA ativo e o *Lifetime* da mensagem for vencido após determinado tempo, o MA passivo assume o estado ativo e começa a disseminação dos <agt_Adv>, a fim de se anunciar na rede MANET como o novo MA ativo. Esse processo garante o registro dos MN previamente cadastrados no antigo MA para serem registrados junto a ele, com o auxílio de mensagens <reg_rq> e <reg_rp>.

O nó no papel de agente primário deve procurar na sua tabela de cadastro por nós RDAIPM que sejam possíveis candidatos à eleição do agente secundário. Esse cadastro é montado à medida que os nós efetuam seu registro na rede (através da troca de mensagens *registration request* e *registration reply*) e solicitam sua participação no processo de eleição por meio do envio de *passive agent requests*.

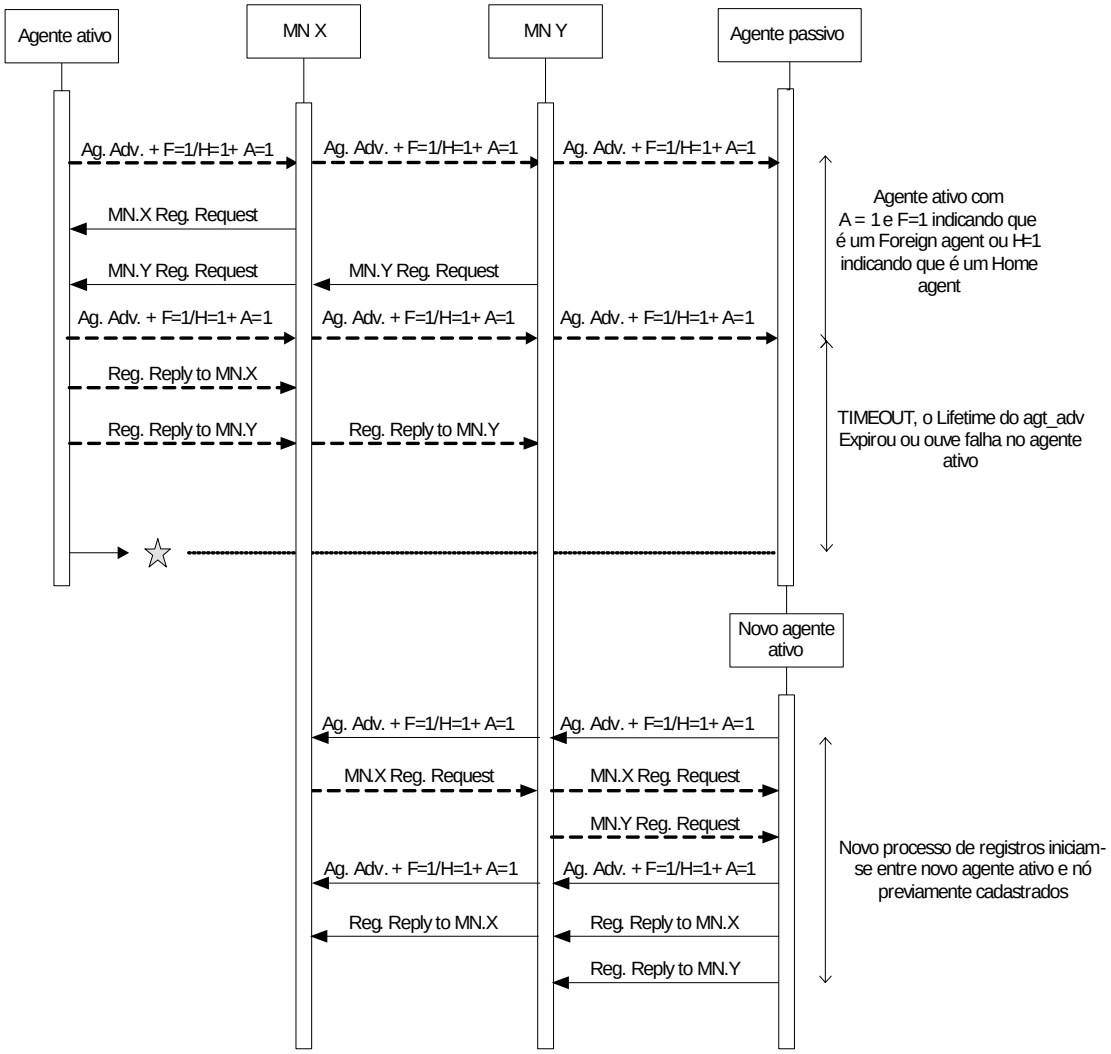


Figura 4.9 - Sinalização da mudança do agente ativo para agente passivo.

A escolha do *agente passivo*, dada a existência de pelo menos um nó cadastrado no *agente ativo*, seria feita com base na procura pela rota AODV de menor número de seqüência `DSN` (*Destination Sequence Number*) e menor número de saltos `HOP_COUNT` até o nó. Se o melhor nó RDAIPM for encontrado obedecendo aos parâmetros anteriores, o agente primário o elege como agente secundário. Monta-se a tabela de cadastro para eleição dos *agentes passivos*, a ser consultada para futuras escolhas com o objetivo de se reduzir o tempo usado para eleição desses agentes, desde que o tempo de vida da tabela seja menor que o tempo da próxima atualização da tabela de roteamento AODV.

Em termos de implementação real, como não se chegou a lidar com o protocolo de roteamento AODV mas somente com o MIP, utilizou-se a seguinte regra substituível:

a primeira mensagem passive agent request que chega no MA é utilizada para se eleger o MN como agente passivo; os demais são armazenados em um array para o caso de uma nova escolha futura.

Como exemplo, segue a sinalização da eleição do agente passivo [65], conforme Figura 4.10:

1: O MA ativo, tem de enviar os <agt_adv> com a <FLAG_A=11>. Essa mensagem é processada por cada nó da rede MANET com RDAIPM. É lembrado no processo que os <agt_adv> podem revelar através das flags H e F que a mensagem provém de um MFA ou MHA e não de ambos.

2: Essa mensagem é usada unicamente pelo nó a fim de saber para qual MA enviar o requerimento de registros como candidato para agente passivo. Neste caso o nó envia por unicast a mensagem de solicitação ao MFA.

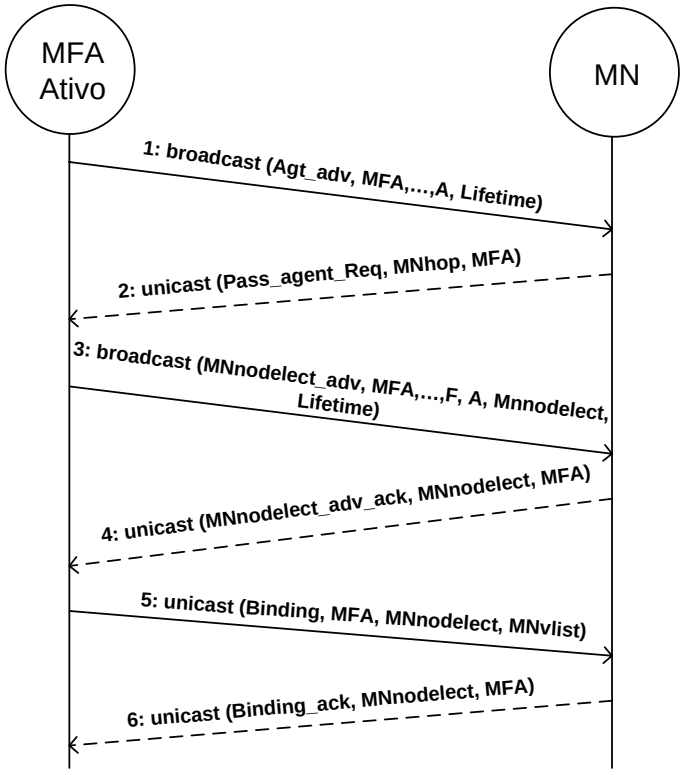


Figura 4.10 - Sinalização da eleição do agente passivo.

3: Quando a mensagem for recebida pelo agente ativo, este identificará os nós que podem ser candidatos através das restrições administrativas. Os nós têm

de estar previamente cadastrados no agente ativo da rede nativa via sinalização padrão MIPv4 e não serem nós com constante mobilidade fora da rede nativa. O melhor nó achado para ser agente passivo, o MFA envia a todos os nós o aviso de parar o envio de requerimento de registros como candidatos para agente passivo. A mensagem contém um campo com o endereço lógico do nó escolhido como passivo.

4: O nó eleito deve enviar por unicast uma mensagem ao agente ativo confirmando a recepção da mensagem anterior.

5: Já que o agente passivo deve ter de antemão as configurações contidas nas mensagens registration request dos nós RDAIPM enviados ao agente ativo, essa mensagem é copiada e encaminhada do MA para o MNnodelect.

6: O MNnodelect envia por unicast uma mensagem ao agente ativo confirmando a recepção da mensagem anterior.

4.9 - DIAGRAMA DE ESTADOS DE UM NÓ GENÉRICO RDAIPM

Todos os MNs que implementam o protocolo RDAIPM devem apresentar o diagrama de estados apresentado na Figura 4.11 [61,65,75]. O estado `EXIT` ocorre sempre que uma falha no algoritmo é detectada. No estado `LISTEN`, todas as mensagens trafegando na rede são analisadas. Se não aparece uma *flag* A no *agent advertisement*, é porque o MN em questão está numa rede MIP comum. Isto serve para garantir uma funcionalidade transparente para o MIP e, assim, nenhum processo RDAIPM será iniciado, [ver apêndice B].

Se o MIP *agent advertisement* contém uma *flag* A, dois processos distintos podem ser iniciados, dependendo se a mensagem que veio de um MFA (*flag* F setada) ou de um MHA (*flag* H setada). O MN eleito, chamado de MNnodelect, entrará no estado de MFA (MHA) passivo. Enquanto o *lifetime* dos *agent advertisements* for menor que determinado *treshold time*, o MNnodelect irá monitorar qualquer falha eventual do MFA (MHA) ativo. Ao mesmo tempo, o MFA (MHA) sincroniza sua lista de MNs (visitantes ou fora de casa, dependendo do tipo de MA) para o novo MNnodelect.

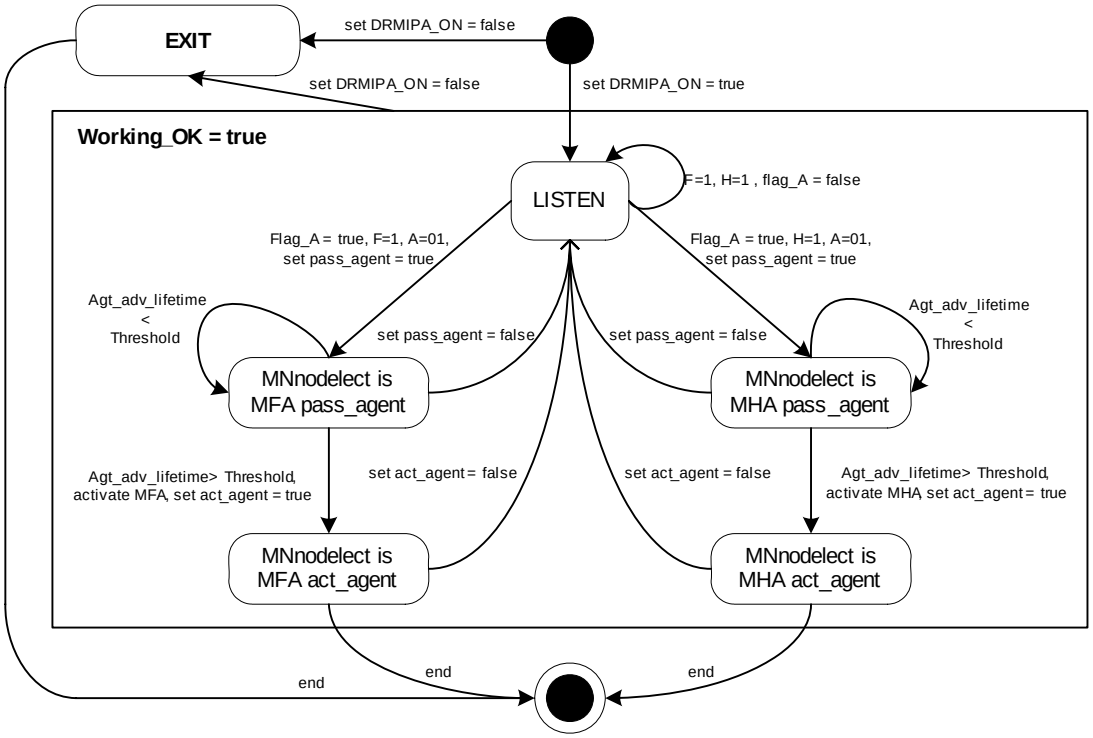


Figura 4.11 - Diagrama de estados básico para um Mnnodelect.

Se o *lifetime* dos *agent advertisements* exceder o *treshold time*, o nó vai para o estado de agente ativo e suas listas previamente obtidas serão incluídas na nova lista do MFA (MHA). Neste ponto, o antigo agente ativo cessa suas atividades. Desse modo, o novo agente ativo estará informado dos MNs previamente registrados com a agente ativo anterior. Agora o algoritmo do agente ativo está sendo executado e um mecanismo de autoconfiguração se inicia para eleição do agente passivo. Se algum MNnodelect deixa a rede ou finaliza sua operação como *agente ativo (ou passivo)*, ele necessita retornar ao estado `LISTEN` inicial.

A presente tese enfatiza o fato de que um nó pode ser um agente ativo ou passivo e ao mesmo tempo manter suas funcionalidades MIP normais como um simples MN. Isto é devido ao fato de que se, por exemplo, um MN possui uma transmissão de dados em andamento com outro MN e é eleito como agente passivo, ele não deve cessar suas operações como um nó MIP normal. Isso pode ser visto como o mesmo caso de um MN funcionando como um servidor de impressão ou de páginas Web e sendo, ao mesmo tempo, um MN normal na rede.

4.10 - IMPLEMENTAÇÃO

Em função das características não estruturadas de uma rede MANET, é esperado que a mesma possa conter desde um até uma quantidade arbitrária de nós, e que as funções de MHA e MFA possam ser assumidas por qualquer estação móvel que esteja na área de cobertura dessa rede e tenha implementado o RDAIPM. É feita inicialmente uma configuração estática dos MA em cada rede que tiver o protocolo RDAIPM, implementado a fim de ter iniciado a eleição do MA passivo caso mais de um nó esteja presente na rede MANET [62]. Uma vez configurado o MA ativo e tendo outros nós RDAIPM presentes na rede MANET, o agente ativo verifica se existe um agente passivo previamente eleito na rede e o coloca em espera [51]. Caso contrário é iniciado o algoritmo de eleição do agente passivo, Figura 4.12 [78].

O algoritmo de escolha do agente passivo é tratado da seguinte maneira:

1: Primeiramente, o nó ativo manda mensagens na rede para avisar sua presença na rede.

2: A escolha do nó a ser passivo tem que passar por pelo menos 3 etapas principais:

- Não estar na tabela que faz associação dos MNs que estão fora da rede.
- Não estar na tabela que faz associação dos MNs que estão de visita na rede.
- E ter o menor número de saltos até o MA, no caso do protocolo de roteamento AODV.

3: Caso o nó queira sair da rede, ele envia uma mensagem para comunicar ao agente passivo que pode se tornar agente ativo.

4: Em caso de falha ou inatividade do agente ativo, o algoritmo para mudança do estado passivo para ativo do agente passivo é acionado e o novo agente ativo tem de repetir os processos anteriormente citados para procura de um novo agente passivo.

```
/* A=11 (act_agent), A=00 (pass_agent), A=01 (act_agent to MNhop in Tablehop ) */

        MNelect := false;
        act_agent := true;
        receive (Binding, MHA, MFA, MNmha ) message from MHA;
        if act_agent ∧ MNelect then begin
                set Flag-A=11 ∧ broadcast (Agt_adv, MFA,…, A, Lifetime) message to
                Network{MFA ∧ MHA} nodes;
                Tablehop [MNhop] := initialize table;
                while (Agt_adv Lifetime < Threshold)
                        receive (Pass_agent_Req, MNhop,MFA) messages from MNhop;
                        Tablehop [MNhop] := MNhop ∈ Network{MFA ∧ MHA}
                        sending Pass_agent_Req messages;
                end
                discard next (Pass_agent_Req, MNhop,MFA) messages from MNhop;
                MNnodelect := MNhop ∉ {(MNmha ∪ MNvlist)} ∧ {at min(MNmip
                Hop-Count)};
                set MNelect := true;
        end
        else if act_agent ∧ ~MNelect then
                set Flag-A=01 ∧ braodcast (MNnodelect _adv, MFA,…, F, A,
                MNnodelect , Lifetime) message to MNhop;
                receive (MNnodelect _adv_ack, MNhop, MFA) message from MNhop;
                unicast (Binding, MFA, MNnodelect, MNvlist) message to MNnodelect;
                receive (Binding_ack, MNnodelect, MFA, MNvlist) message from
                MNnodelect;
        end
        else if ~act_agent then
                error (activate MFA agent);
        end
        else if Gratuitous_exit then
                unicast (Gratuitous_exit, MFA, MNnodelect) message to MNnodelect ;
        end
```

Figura 4.12 - Algoritmo para eleição do agente passivo.

Supõe-se que os novos MNs da rede MANET tenham implementado o RDAIPM para permitir os processos acima descritos, mesmo sem terem sido escolhidos pelo agente ativo.

Na Figura 4.13 tem-se o algoritmo da mudança de um nó passivo para ativo. Caso seja enviado uma mensagem de desabilitação do nó ativo para o nó passivo, é feita a confirmação da mensagem e do recebimento da mensagem.

Caso seja verdadeiro, o nó passivo manda uma mensagem unicast (*ack*) para o MHA (MFA) e inicia o algoritmo de substituição de agente passivo para ativo. Caso o agente ativo não mande a mensagem de desabilitação da rede, o campo de *Lifetime* da mensagem sendo maior do que o limiar permitido irá executar todo o procedimento anterior. A Figura 4.14 apresenta uma mensagem MIP agent advertisement modificada com a inserção da Flag A no campo Reserved [3].

```
Gratuitous_exit := true
pass_agent := true
receive (Gratuitous_exit, MHA/MFA, MNnodelect ) message from MHA/MFA;
        if Gratuitous_exit ∧ pass_agent then begin
    unicast (Gratuitous_exit_ack, MNnodelect, MHA/MFA) message to MHA/MFA;
                activate MFA/MHA algorithm;
    set Flag-A=11 ∧ broadcast (Agt_adv, MHA/MFA,…, A, Lifetime) message to
    Network{MFA ∧ MHA} nodes;
                set pass_agent := false;
                set act_agent := true;
        end
        else if (Adv_Lifetime > Threshold) ∧ pass_agent then
                activate MFA/MHA algorithm;
    set Flag-A=11 ∧ broadcast (Agt_adv, MHA/MFA,…, A, Lifetime) message to
    Network{MFA ∧ MHA} nodes;
                set pass_agent := false;
                set act_agent := true;
        end
```

Figura 4.213 - Algoritmo para mudança de um nó passivo em ativo.

Na Figura 4.15 pode-se ver uma mensagem MIP de *registration request* modificada. Nessa mensagem é inserido um novo campo contendo uma Flag A [51]. Com essa flag os nós da rede enviam uma mensagem ao MA solicitando o requerimento de registro como prováveis candidatos para serem agente passivo. O resto da mensagem é igual ao descrito em [3].

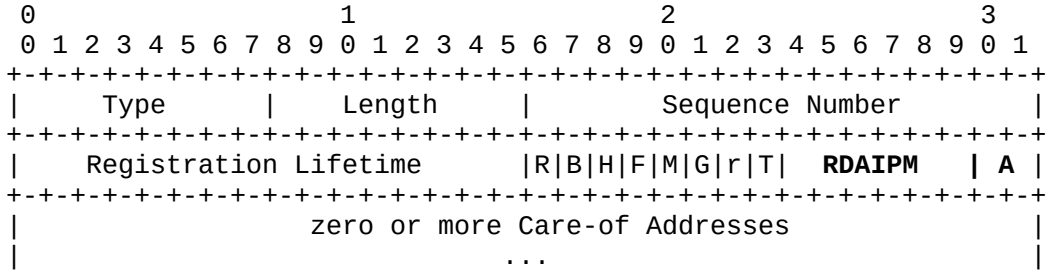


Figura 4.14 - Mensagem MIP de agent advertisement modificada (Type=16, A=11).

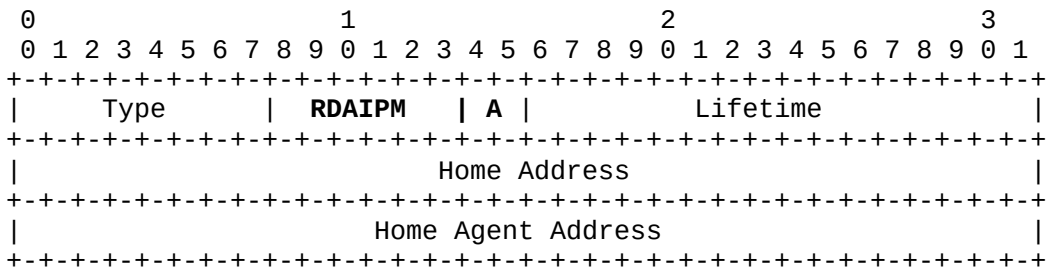


Figura 4.15 - Mensagem Passive Agent Request (Type=81, A=00).

Uma nova mensagem é criada, mostrada na Figura 4.16, para que todos os MNs da rede cessem o envio do requerimento de registro como candidatos para agente passivo. Um campo de 32bits é usado para que cada MN saiba o endereço lógico do nó escolhido pelo MFA (MHA).

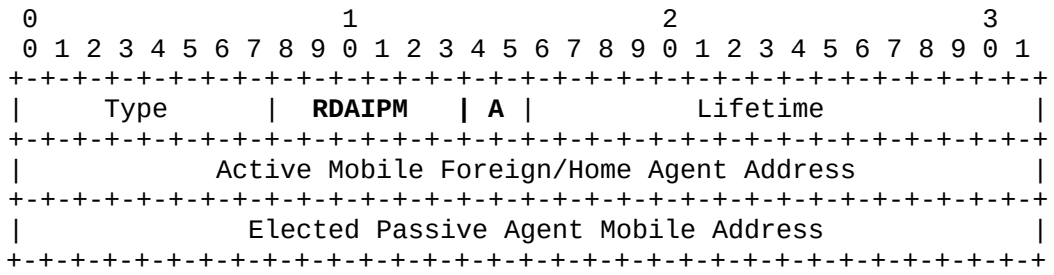


Figura 4.16 - Mensagem MNnodelect Advertisement do MA ativo para nós da rede `(Type=82, A=01)`.

Como o agente passivo tem de estar preparado para assumir como agente ativo, e precisa da cópia da lista dos nós associados (*binding*) ao MA ativo, uma nova mensagem é criada contendo o endereço lógico do MA. Essa mensagem é enviada junto com a lista de nós em visita na rede local.

Um dos problemas encontrados no decorrer da pesquisa diz respeito à possibilidade de se ter os nós ativo e passivo desconectados da rede nativa. Qual seria então o mecanismo para se ter a rede RDAIPM de volta em pleno funcionamento, sem ação direta nos nós da rede nativa? Um dos mecanismos encontrados para resolver esse problema foi de usar o algoritmo de chaveamento de agente passivo para ativo no nó cujo estado estiver em `LISTEN`, e que nunca teve a chance de ser escolhido pelo agente ativo como agente passivo antes de se desativar. Mas, após análise, percebeu-se que essa metodologia não seria suficiente para saber quais dos nós que fazem parte da rede nativa seriam elegíveis para serem automaticamente futuro agente ativo.

Então se definiu que somente o nó com o menor endereço IP na rede, e que tenham implementado o RDAIPM (*neste caso a versão do protocolo RDAIPM teria de ser disponibilizada a posteriori para aqueles que desejam implementar esse mecanismo*) teria de passar para o estado de agente ativo, Figuras 4.17-4.18. Na hora de levantar cada nó RDAIPM, o comando ping broadcast, “`ping –b -c10 [broadcast`

`da rede]" (realizado no sistema operacional LINUX)`, tem de ser executado automaticamente em cada nó, após falha do agente ativo e passivo (detectado pelos nós no caso da não recepção consecutiva de três agent advertisements), para saber quais endereços IP estão presentes na rede. Através de uma comparação de endereço IP do nó que gerou o comando junto ao resultado do `ping` anterior sabe-se quais nós têm o primeiro e segundo menor endereço IP da rede nativa. O nó que tiver o primeiro e segundo menor endereço IP da rede nativa tem de ativar seu algoritmo de agente ativo e disseminar os *agent advertisements* com as *Flag A/H* e *Flag A/F* respectivamente setadas, e o processo de escolha de agente passivo teria início como mostrado nas Figuras 4.11-4.17.

Vale lembrar que esse mecanismo para autoconfiguração dos agentes RDAIPM numa MANET prevalece nos seguintes casos:

1: O agente ativo morrer antes de escolher o nó para ser agente passivo;

2: Os nós ativos e passivos morrerem ao mesmo tempo.

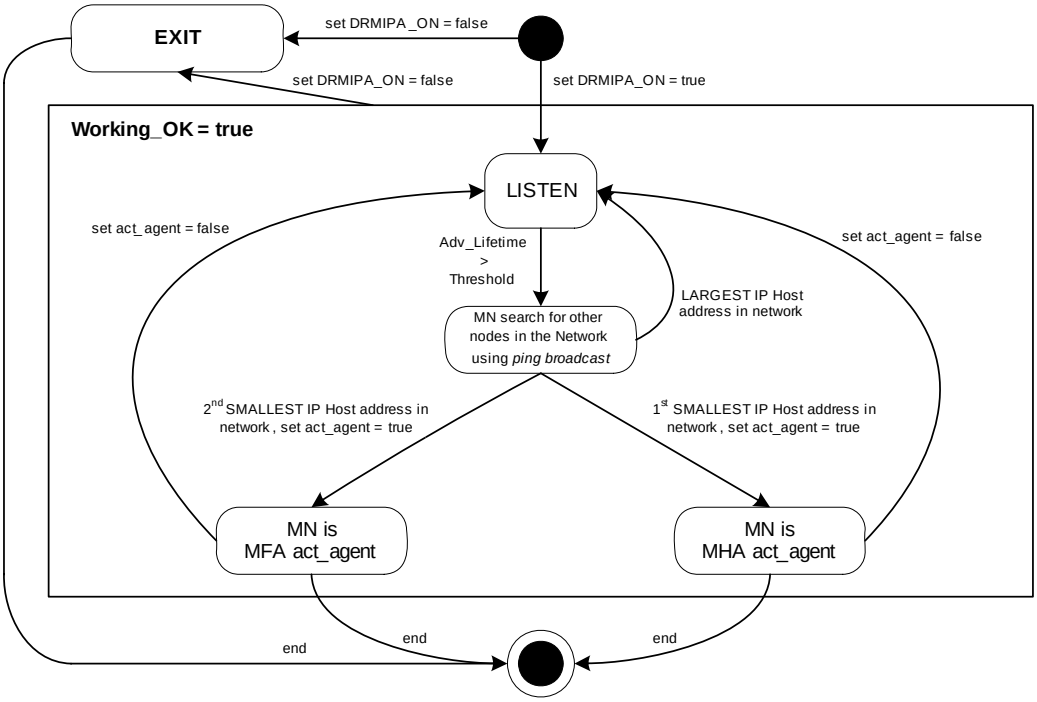


Figura 4.17 - Diagrama de estados básico para autoconfiguração de um MN.

Em ambos os casos, nenhum ex-agente ativo ou passivo têm de ter os seus estados anteriores reativados como mostrado nas Figuras 4.11 e 4.17, só se não for

estabelecido administrativamente, para evitar de se ter vários agentes ativos e passivos na mesma rede, o que aumentaria drasticamente o *flooding* na rede.

Outro ponto importante diz respeito à administração da rede que se deseja que seus nós tenham acesso a Internet. Para se ter internet através de um gateway, que neste caso seria o próprio agente ativo, o nó tem de possuir duas interfaces de redes (uma com o modo de configuração em *Ad Hoc* e outra para *WLAN infraestruturada*). Portanto, a parte administrativa deverá seguir o passos de configuração de cada nó onde o uso de duas interfaces de rede (*NICs*) sem fio se torna mandatório e descrito em [ver apêndice B].

A Figura 4.18 apresenta quatro cenários nos quais é exemplificado o início e término de uma autoconfiguração dos nós RDAIPM após uma desativação ou mal funcionamento do *agente ativo* previamente configurado pelo processo definido na Figura 4.6. Supondo que se queira configurar uma rede MANET executando o RDAIPM, um nó é escolhido e configurado para ser *agente ativo* (cenário 1). Após certo tempo inferior ao requerido para se ter um *agente passivo* eleito na rede, o agente ativo entra em falha (cenário 2). Por não receberem três *agent advertisements* consecutivos na rede, os demais nós iniciam automaticamente um *ping broadcast* entre todos para saber quais deles têm o primeiro ou segundo menor endereço IP da rede (cenário 3). Quando o mecanismo anterior estiver terminado, o nó cujo endereço IP é o primeiro menor host da rede se torna o *MHA* e entra no estado `ATIVO` (cenário 4). Da mesma maneira, o nó sendo o segundo menor endereço IP entra no estado `ATIVO` para agente *MFA* (cenário 4).

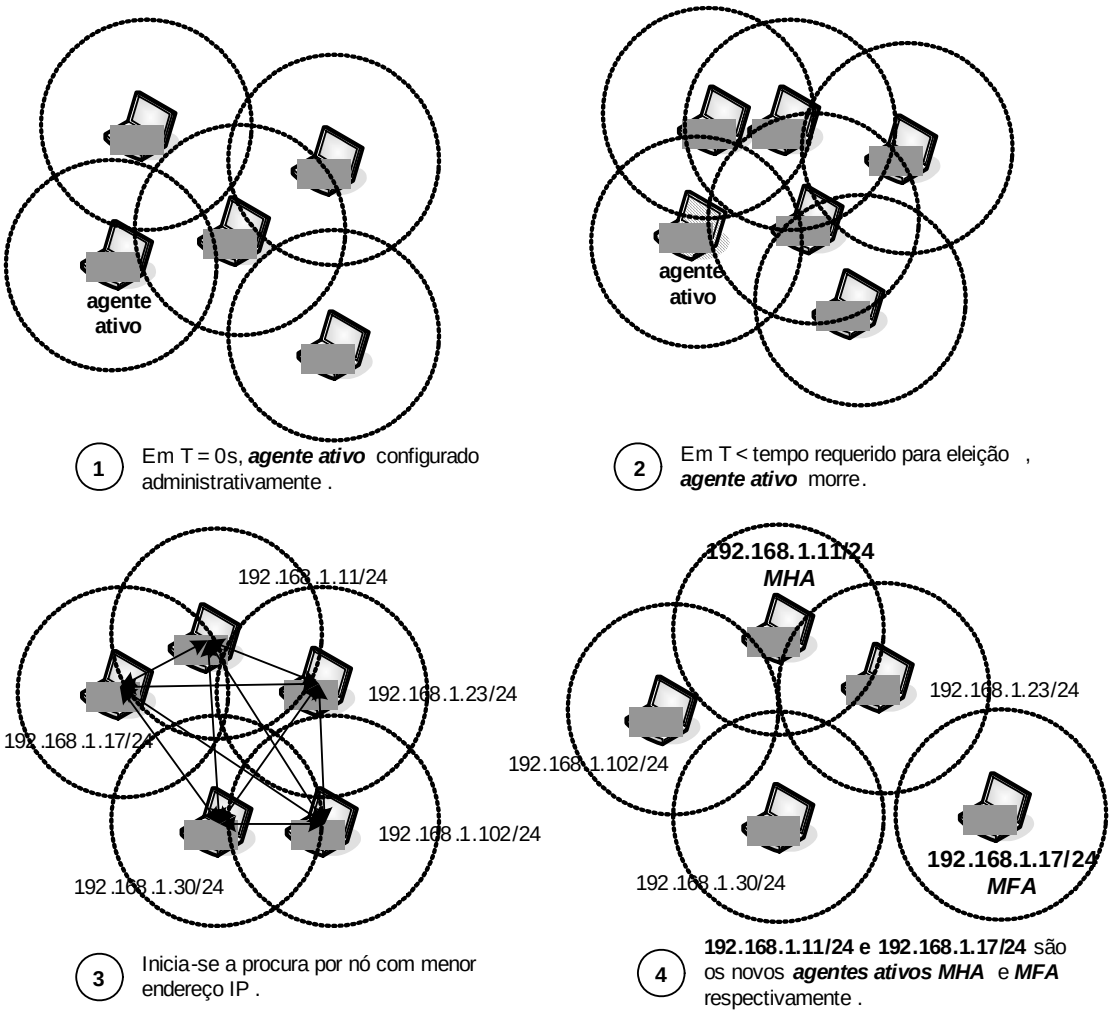


Figura 4.18 - Exemplificação de um cenário para autoconfiguração dos nós RDAIPM.

5 - TUNELAMENTO

Neste Capítulo serão discutidos os pontos relevantes quanto ao uso de um mecanismo de entrega de pacotes entre os nós da rede RDAIPM conhecido como tunelamento. Esses nós, sendo eles MA e MN, devem conservar os mecanismos de mobilidade descritos em [3,23,24,66]. Ou seja, para se ter o RDAIPM e MIPv4 atuando transparentemente dentro ou fora da MANET para os demais nós da rede, os seus pacotes têm de ser encaminhados entre os agentes de mobilidade, sem passar por nós que tenham rotas diretas para o destino desejado. Assim, existe a necessidade de existir mecanismos que criem um caminho do nó origem até o nó destino sem passar pelos agentes de mobilidade IP, e assim sobrepassando os mecanismos de mobilidade sugeridos pela RFC-3222.

Por causa dos problemas mencionados anteriormente, os itens a seguir irão esclarecer o porquê do uso de um mecanismo de tunelamento para o RDAIPM, podendo não somente em uma rede MANET MIPv4, como também em uma rede WLAN infraestuturada.

5.1 - TUNELAMENTO NA REDE RDAIPM

Com o tunelamento, um pacote tendo destino à Internet passando pela rede RDAIPM é enviado para o gateway de acesso a Internet passando sempre pelo agente ativo. Os mecanismos de encaminhamento dos pacotes IPs vindo da rede RDAIPM dentro do gateway são independentes do protocolo de roteamento usado nessa rede. Todo pacote chegando ao gateway será decapsulado para ser encaminhado na Internet, Figura 5.1. Os mecanismos são basicamente os mesmos descritos em [12,13,23,24,47].

Quando um pacote chegar à Internet através do gateway com destino na rede RDAIPM, ele terá de passar pelo agente ativo, apesar do endereço de destino ser roteável na rede RDAIPM (o gateway envia o pacote até o agente ativo que por sua vez o envia até o nó destino) [45]. Todos os estados para encaminhamento do pacote até o gateway são mantidos no nó de origem. Nenhum estado é replicado entre os nós intermediários da rede RDAIPM. Só a rota para o gateway é mantida dentro dos nós intermediários. É necessário a implementação no NAT e DNAT no gateway a fim de permitir a resolução de endereço privados e públicos.

Por outro lado, no protocolo RDAIPM, o nó de origem, tendo como destino um nó fora da rede nativa, sempre terá de enviar os seus datagramas passando pelo agente ativo que, por sua vez, os encaminhará por tunelamento até o agente estrangeiro da rede vizinha MANET [61]. Assim, os mecanismos de tunelamento do MIPv4 serão mantidos pelo RDAIPM entre redes MANETs, Figura 5.1.

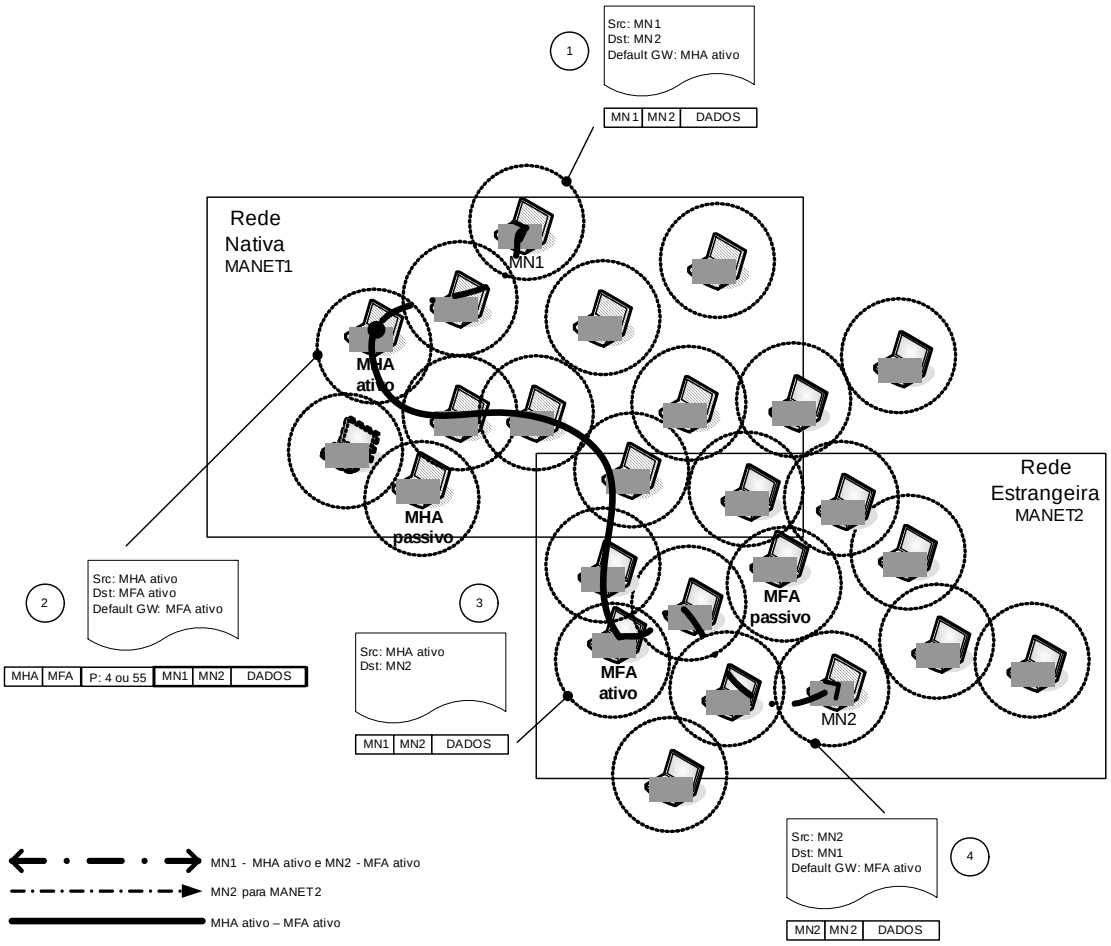


Figura 5.1 - Exemplificação do Tunelamento na rede RDAIPM.

A Tabela 5.1 reflete as vantagens e desvantagens do uso de tunelamento na rede RDAIPM. Apesar de causar um overhead mínimo entre nós envolvidos na rota ativa até o agente ativo, não deixa de ser uma solução interessante por permitir a integridade dos mecanismos de sinalização descritos em [3].

Tabela 5.1 - Resumo dos pontos relevantes para tunelamento na rede RDAIPM.

VANTAGEM	DESVANTAGEM
Todos os estados para encaminhamento do pacote até o gateway são mantidos no nó de origem. Nenhum estado é replicado entre os nós intermediários da rede RDAIPM. Em caso de ruptura de link, o nó origem pode ter outra rota para tunelar o pacote até o gateway.	Overhead das mensagens ao longo da rota ativa. Pelo fato de todos os nodos terem como gateway intermediário o agente ativo, esse último teria uma sobrecarga maior no que diz respeito ao encaminhamento de todos os túneis. Recursos do nó podem diminuir com rapidez. Sempre haverá um tunelamento com rota default passando pelo Agente Ativo, para um destino na rede MANET ou Internet.

5.2 - ROTA DEFAULT NA REDE RDAIPM

O uso de rotas default nas redes LANs é muito mais comum do que em redes MANETs. Isso é devido ao fato de se ter um gateway de acesso à rede mundial a um único salto. Em contrapartida, redes MANETs fazem uso de múltiplos saltos e podem ter vários gateways para permitir redundância no acesso a rede externa [45].

Não se pode ter rastreio de todos os gateways tendo unicamente uma rota default na tabela de entrada para um destino na Internet. O uso de protocolos reativos na rede MANET pode acarretar mudança de requisição de rotas a qualquer momento para determinado gateway [66]. Os nós intermediários só teriam uma entrada para nova rota default, sem portanto informar a mudança de gateway de acesso aos demais nós da rede.

O seu uso dentro da rede RDAIPM pode não causar tantos problemas pelo fato de se ter um único agente ativo pelos quais os pacotes devem passar, enquanto ele for ativo. Mas, como dito anteriormente, caso haja uma ruptura de link para se chegar até o novo agente ativo, uma nova tabela tem de ser entrada para o IP do novo agente ativo. Ou seja, todos os nós deverão atualizar suas tabelas de roteamento contendo o novo agente ativo como rota default para se chegar até o gateway.

A Tabela 5.2 também reflete as vantagens e desvantagens do uso de rota default na MANET para o correto funcionamento do RDAIPM.

Tabela 5.2 - Resumo dos pontos relevantes para Rota Default na rede RDAIPM.

VANTAGEM	DESVANTAGEM
Funciona bem caso exista um único agente ativo permanente na rede, sem mudar de endereço IP. A rota default seria sempre mapeada até o principal agente ativo.	Aumento do número de acesso à tabela de roteamento para encaminhamento dos pacotes até o gateway, em cada nó. Ineficaz para o suporte de vários gateways de acesso a Internet. Recursos do nó podem diminuir com rapidez. Sempre haverá uma rota passando pelo agente ativo, para um destino na rede MANET ou Internet. A rede RDAIPM pode ter cada nó como futuro agente passivo/ativo e isso seria problema no uso de rotas default predefinidas.

5.3 - LOOSE SOURCE ROUTING NA REDE RDAIPM

Também conhecido pelo campo opcional Loose Source Record Route (LSRR) do protocolo IP, ele permite à fonte geradora do datagrama IP fornecer um roteamento explícito que deve ser usado pelos roteadores da rede. A informação contida no LSRR é usada para o encaminhamento do datagrama até seu destino e, da mesma forma, gravando a rota usada do nó de origem até destino. Os dados sobre a rota são compostos por séries de endereços IP de 32 bits.

Então, dependendo do número de nós na rede, esse campo pode ser grande demais e causar overhead para encaminhamento de um só datagrama até o seu destino. Com o uso de protocolos reativos na rede RDAIPM, isso talvez causasse congestionamento nos nós da rede. Cada nó sendo um roteador na rede RDAIPM deveria olhar esse campo a fim de saber qual caminho deve ser usado para envio do datagrama e assim permitir envio de mensagens RREQ e RREP do protocolo AODV, ou qualquer mensagem enviada por outro protocolo de roteamento usada para atualização da tabela de rotas, até o IP destino contido no LSSR.

No caso da proposta RDAIPM, o nó de origem sempre terá de enviar os seus datagramas passando pelo agente ativo (cujo agente poderia ser a qualquer instante outro nó da rede RDAIPM com outro endereço IP) e, dessa forma, permitir ao agente saber mais a respeito dos nós na rede. Só o agente ativo teria como usar as

informações contidas no campo LSSR do nó de origem para encaminhá-lo até o nó destino (podendo ser um nó localizado em outra MANET, RDAIPM ou Internet).

Essa maneira de usar o LSSR faria com que se tenham somente três endereços a serem usados para se chegar até o destino, sendo eles: IP agente ativo, IP gateway, IP nó Internet/MANET/RDAIPM. O ponto positivo desse método é de se ter uma rota predefinida para envio do pacote sem preocupação de quais nós intermediário foram usados para esse fim. Um problema seria a necessidade de avisar o nó origem sobre a troca de agente ativo ocasionado mudança de IP a ser incluído no LSSR (recebendo informação sobre mensagens Advertisement, o nó origem teria de remontar o pacote colocando o endereço IP do novo agente no campo LSSR). A Tabela 5.3 mostra as vantagens e desvantagens do uso de rota Loose Source Routing na MANET para o correto funcionamento do RDAIPM.

Tabela 5.3 - Resumo dos pontos relevantes para Loose Source Routing na rede RDAIPM.

VANTAGEM	DESVANTAGEM
Funciona bem caso exista um único agente ativo permanente na rede sem mudar de endereço IP. Seriam necessários apenas três endereços IP para criar um túnel até a fonte destino.	Aumento da lista de endereços IP contidos no campo LSSR para encaminhamento dos pacotes até o gateway, em cada nó. Ineficaz para o suporte de vários gateway de acesso a Internet. Isso aumentaria o tamanho do campo LSSR. Recursos do nó podem diminuir com rapidez. Sempre haverá uma rota passando pelo agente ativo, para um destino na rede MANET ou Internet.

Neste capítulo foram abordados os possíveis requisitos que viabilizam a manutenção dos mecanismos de encaminhamento na MANET para prover a mobilidade IPv4 descrita em [3]. Em [66], uma análise de desempenho é feita sobre qual mecanismo para encaminhamento de pacotes tem de ser usado dentro da MANET até um *gateway* da rede Internet. Os testes comprovam que tem te ser usado um mecanismo de tunelamento para encaminhar os pacotes, na rede Ad Hoc e com o auxílio de um protocolo de roteamento reativo. A proposta se torna mais eficiente e flexível do que fazer o uso do mecanismo de rota default.

Dessa forma, a escalabilidade do roteamento para Internet e inter MANET é mantida, já que cada nó não necessita conhecer a posição geográfica de nenhum dos agentes RDAIPM. Mesmo que todos os nós RDAIPM/MIP/MANET possam receber pacotes diretamente, somente os dois agentes ativos têm de saber que o tunelamento está ocorrendo para viabilizar a mobilidade IP fora dessa rede. O único requisito do nó intermediário é de verificar para qual nó tem de ser encaminhado o pacote de origem, que neste caso seria um dos agentes ativos que tratará de realizar o desencapsulamento e encapsulamento do mesmo até o nó de destino.

6 - ANÁLISE DE DESEMPENHO

Neste Capítulo serão apresentados os resultados do RDAIPM no cenário de testes de um simulador de rede, bem como num ambiente real para validação do protocolo proposto. Os cenários terão o protocolo MIP [A] modificado e refletindo as funcionalidades do RDAIPM. O protocolo de roteamento escolhido foi o AODV [60], pelas razões já descritas anteriormente. As discussões julgadas pertinentes sobre os mesmos serão realizadas após exposição de cada cenário. O total de pacotes analisados na rede é dado pela equação,

$$\begin{aligned} \text{Total Pacotes na rede} &= \sum_{i=1}^{n} P^n \\ &= P^1 + P^2 + P^3 \\ &= [\{SH_{n'}, v_2\},..,\{MHA,..\},\{..,MFA\},..,\{..,DH_n\}] \end{aligned} \tag{6.1}$$

onde P^n representa a soma dos pacotes gerados em cada rota ativa, ou seja, somente os nós envolvidos nessa rota têm os seus pacotes computados.

Na equação (6.1), o nó *SHn* pode representar um nó origem ou agente passivo (nativo ou estrangeiro). O *MHA* um agente ativo (nativo ou estrangeiro), o *MFA* um agente ativo (nativo ou estrangeiro), e o DH_n, um nó destino ou agente passivo (nativo ou estrangeiro). O sentido usado no caminho, de ou para rede nativa/estrangeira, não tem influência no cômputo do total de pacotes na rede.

6.1 - ANÁLISE DE DESEMPENHO NO SIMULADOR DE REDE NS2

O protocolo RDAIPM foi implementado e compilado no simulador de rede NS2 versão 2-27/2-28/2-29 [19], nas linguagens de programação C++ e OTcl, com as extensões dos protocolos MIP e AODV [19], e a extensão desenvolvida nesta tese do RDAIPM [A].

O arquivo *Trace* (cmu-trace.cc) foi parcialmente modificado para receber o rastreio do novo formato das mensagens do RDAIPM e MIP (até a data desta tese, nenhuma implementação do NS2 fornecia o *Trace* das mensagens MIP).

O cenário da simulação contém 10 nós em uma rede ad hoc IEEE 802.11b com nós estáticos usando o modelo de canal sem fio Two-Ray Ground [71], numa área plana de 670x670 metros.

O objetivo é analisar os fatores de atraso e utilização de banda que ocorrem entre o MA e seus respectivos agentes passivos, quando for comparado ao envio e recepção das mensagens MIP e RDAIPM simultaneamente. A idéia é que os resultados possam refletir o desempenho e confiabilidade do protocolo proposto quando tiver de ser eleito um agente passivo.

A Tabela 6.1 mostra os parâmetros usados na simulação deste cenário. O restante dos parâmetros de configuração básicos do protocolo MIP que estão implementados em código OTcl e C++, como tamanho de pacote, não foram alterados. Um tempo para aquecimento da simulação (*warm up*) de 5 s foi usado para correta coleta dos dados a serem analisados. O tamanho máximo das mensagens é de 54 bytes (432 bits), correspondentes ao protocolo MIP, já que as mensagens do RDAIPM são bem menores que as do MIP. Cada cenário teve a simulação repetida cinco vezes.

Conforme equação (6.2), a porcentagem de banda utilizada entre o agente ativo e passivo pode ser obtida fazendo-se `<dados_enviados> = total dados do protocolo RDAIPM enviados` e `<dados_recebidos> = total dados do protocolo RDAIPM recebidos,` a partir do arquivo trace `RDAIPM.tr`, que reflete os pacotes RDAIPM trocados durante a sinalização. Devido ao overhead do IEEE 802.11b, a taxa de transmissão de 5,5 Mbps foi usada ao invés dos 11 Mbps de taxa de transmissão teórica [73].

$$\%Utilização = \frac{(dados\_enviados + dados\_recebidos)*8}{(banda*tempo)}100 \tag{6.2}$$

Dos dados do arquivo *Trace*, obteve-se uma utilização de banda de 1,4%. Dado esse valor, pode-se ver que as mensagens que permitem a eleição do agente passivo, a partir de um grupo de nós RDAIPM, são bem insignificantes em termos de ocupação de banda, durante toda a simulação.

Tabela 6.1 - Parâmetros de Simulação.

Nome dos Parâmetros	Valores
Nr. De nó	10
Distribuição dos nós	Aleatória
Tamanho das mensagens (incluindo cabeçalhos)	54bytes
Tamanho da grade	670x670m
Características do canal	Two-Ray
Antena	Omni direcional
Padrão de mobilidade	Sem-mobilidade
Lifetime dos Agent advertisements	15s
Tempo para reenvio	5s
Tempo para warm up da simulação	5s
Simulação inicia em	0s
Simulação finaliza em	30s

O atraso médio apresentado na Tabela 6.2 foi medido usando o atraso médio de ida de um agente ao outro. Isso foi feito para se ter uma boa análise comparativa do tempo gasto para envio e recepção das mensagens MIP e das mensagens RDAIPM [61].

Para o cômputo do atraso médio, foi usada a equação:

$$\begin{aligned} Atraso = {} & Agt\_Ativo\,(mensagem\_criação\_e\_envio) \\ & + Agt\_Passivo\,(mensagem\_processada) \\ & + Agt\_Passivo\,(mensagem\_criação\_e\_envio) \end{aligned} \tag{6.3}$$

O *Agt_Ativo(mensagem_criação_e_envio)* reflete o tempo gasto para uma mensagem deixar a interface do agente ativo após ter sido criada.

O *Agt_Passivo(mensagem_processada)* trata do tempo gasto pelo agente passivo para processar a mensagem vinda do agente ativo.

O *Agt_Passivo(mensagem_processada)* reflete o tempo gasto para uma mensagem deixar a interface do agente passivo.

A Tabela 6.2 apresenta os resultados de atraso médio obtidos durante a simulação, assim como os nós envolvidos na troca destas mensagens. Para rastreio das transações trocadas pelos agentes ativos, passivos e nós da rede, são utilizados os tipos de

mensagem MIP/RDAIPM, as Flag-A/H/F e para onde estas mensagens são enviadas, mostradas respectivamente na primeira, segunda e terceira coluna da Tabela 6.2 cujo detalhamento pode ser visto em [Apêndice B].

Como a modificação da mensagem *RDAIPM_Agt_adv* teve apenas a inclusão de uma Flag-A de 2 bits, para diferenciá-la da mensagem original *MIP_Agt_adv*, nenhuma diferença no atraso médio entre ambas mensagens foi notada. Por serem as mensagens dos agentes de mobilidade MIP/RDAIPM pequenas, o tempo de atraso médio de 1,89 ms é o melhor entre todas as outras mensagens. Um tempo de 2,11 ms é observado na troca de mensagens *Pass_agent_req (do RDAIPM)* e *Reg_Request (do MIP, cuja terminologia foi definida no nosso arquivo Trace)*. Isso se explica pelo fato de serem mensagens com o mesmo tamanho em bytes. A diferença é que a mensagem do RDAIPM tem de ser enviada após recepção de um *RDAIPM_Agt_adv* , com as *Flag-A* e *Flag-H/F* configurada como 1.

O *MNnodelect_adv* registrou um dos maiores tempos, *4,31 ms*, na troca de mensagens porque o agente ativo teve de receber o *Pass_agent_req* de todos os nós da rede, que fossem desejar participar na eleição como futuro agente passivo, procurar na tabela de futuros agentes passivos aqueles que são parte da rede atual e escolher o melhor entre eles utilizando parâmetros definidos em [Apêndice B]. Assim, de posse do endereço lógico do novo agente passivo, a mensagem *MNnodelect_adv* é criada e disseminada dentro da rede para que todos os nós possam saber qual entre eles é o nó escolhido para ser agente passivo.

Por outro lado, a mensagem *MNnodelect_adv_ack* teve um tempo de 2,64 ms abaixo do tempo registrado anteriormente por ser uma mensagem de confirmação do novo agente passivo para o agente ativo. Essa é uma forma usada por ambos agentes para sinalizarem os seus estados de atividades. Cabe ressaltar que somente o nó eleito envia por unicast esta mensagem para o agente ativo e, dessa forma, evita o *flooding* desnecessário na rede toda como explicado em [Apêndice B].

Tabela 6.12 - Comparação de desempenho das mensagens MIP e RDAIPM.

	Flags	Atraso Médio (ms)	De - Para
MIP_Agt_adv	H=1/F=1	1,89	
RDAIPM_Agt_adv	A=11	1,89	Active MFA (MHA) - MN
Reg_Request	-	2,11	
Pass_agent_req	A=00	2,11	MN - Active MFA (MHA)
Reg_Reply	-	2,16	
MNnodelect_adv	A=01	4,31	Active MFA (MHA) - Passive MFA (MHA)
MNnodelect_adv_ack	A=10	2,64	Passive MFA (MHA) - Active MFA (MHA)

A Figura 6.1 apresenta o comportamento do overhead das mensagens RDAIPM do cenário estudado, conforme aumenta-se o número de nós [61,65]. A mensagem com maior overhead é a *MNnodelect_ADV*, porque todos os nós da rede MANET têm de saber qual dos nós presentes é o agente ativo. Dessa forma existe um flooding inevitável dentro da rede, caso contrário nenhum nó teria os mecanismos do RDAIPM em correto funcionamento. Uma vez que o agente ativo envia o Flooding da mensagem *MNnodelect_ADV*, os nós podem fazer o unicast da mensagem RDAIPM_pass_REQ para o agente ativo. Pode-se ver que o overhead dessa mensagem chega a ser 4,5 vezes menos que o Flooding dos advertisements do agente ativo. Julgamos então, que fazer o unicast e não Flooding do *RDAIPM_pass_REQ* se torna a melhor solução para evitar flooding da rede já que nem todos os nós na rede terão o RDAIPM implementado.

Em uma rede MANET pode haver todo tipo de nó, sendo eles nós MIP, RDAIPM ou comuns. Ou seja, nem todos os nós devem saber quem vai participar da eleição e sim quem foi eleito. Partindo desse princípio, o agente passivo eleito tem de manter contato constante junto ao seu agente ativo fazendo unicast da mensagem *MNnodelect_ACK*. O overhead do *MNnodelect_ACK*, apesar de pequeno, segue o comportamento da curva da mensagem *MNnodelect_ADV* porque o agente passivo só a envia quando o agente ativo dissemina os advertisements.

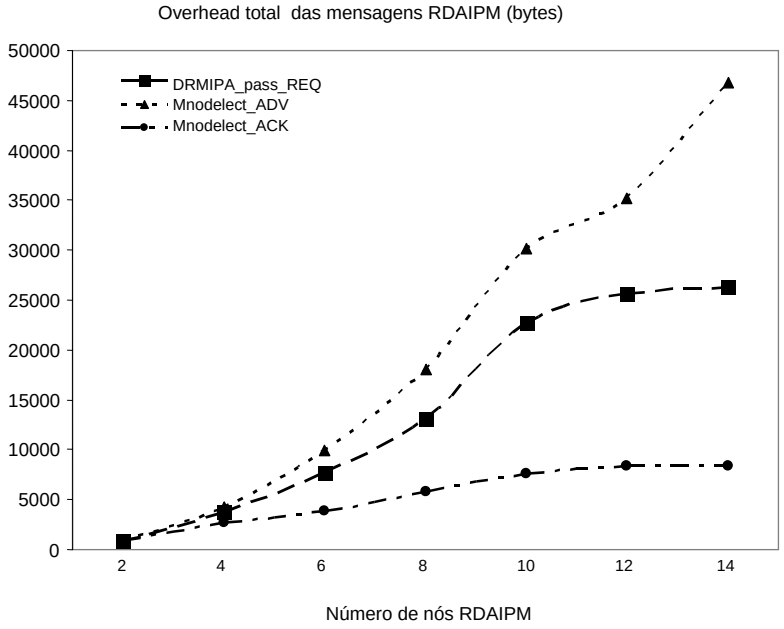


Figura 6.1 - Overhead das mensagens principais do RDAIPM em bytes.

A Figura 6.2 nos permite compreender a distribuição dos valores de tempo de eleição para escolha do agente passivo. Não se observaram nos sete experimentos realizados (contendo respectivamente 2, 4, 6, 8, 10, 12 e 14 nós), resultados atípicos, a não ser pelo tempo percebido na eleição do nó 1 no primeiro experimento, com dois nós, sendo de ~0,83 s. Esse tempo é maior que dos outros seis experimentos, tendo respectivamente quatro a quatorze nós, devido à posição aleatória de uns 40 m entre os nós 1 e 0 na área de simulação.

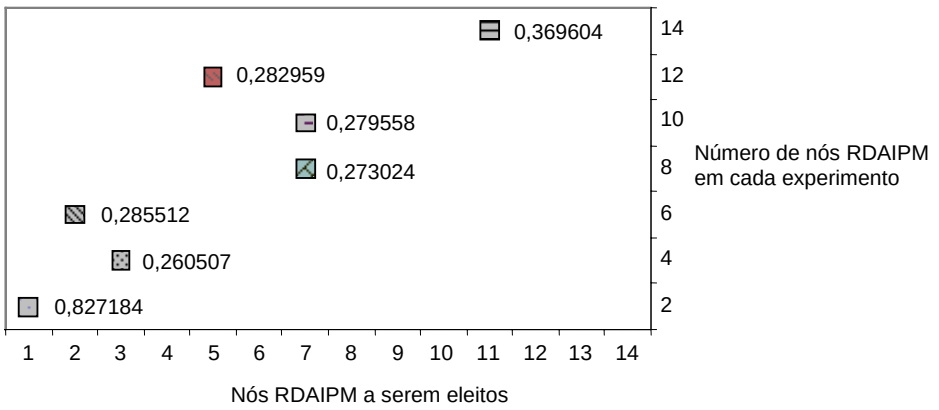


Figura 6.1 - Tempo para eleição do agente passivo RDAIPM (em segundos).

Os nós eleitos foram escolhidos a partir de uma lista tabelada de nós RDAIPM previamente estabelecida na hora de receber as mensagens *RDAIPM_pass_REQ*. Os

demais tempos de eleição estiveram uma média de 0,30 s por terem sido os nós escolhidos que estavam mais perto do agente ativo. Um dos parâmetros utilizados na simulação foi escolher o nó que fisesse parte da rede do agente ativo e estivesse a menos saltos dele, conforme descrição da proposta RDAIPM. Esse resultado comprova que é uma boa forma de eleição do agente passivo.

A Figura 6.3 mostra os detalhes do cenário dos experimentos 5 (com dez nós) e 6 (com doze nós) respectivamente. Pode-se ver no cenário do experimento 5 que o nó 7 foi eleito como agente passivo, porque ele está a 1 salto do agente ativo (nó 0), segundo tabela de roteamento do protocolo AODV. No cenário do experimento 6 o processo de eleição foi o mesmo e neste caso, o nó 5 foi escolhido como agente passivo.

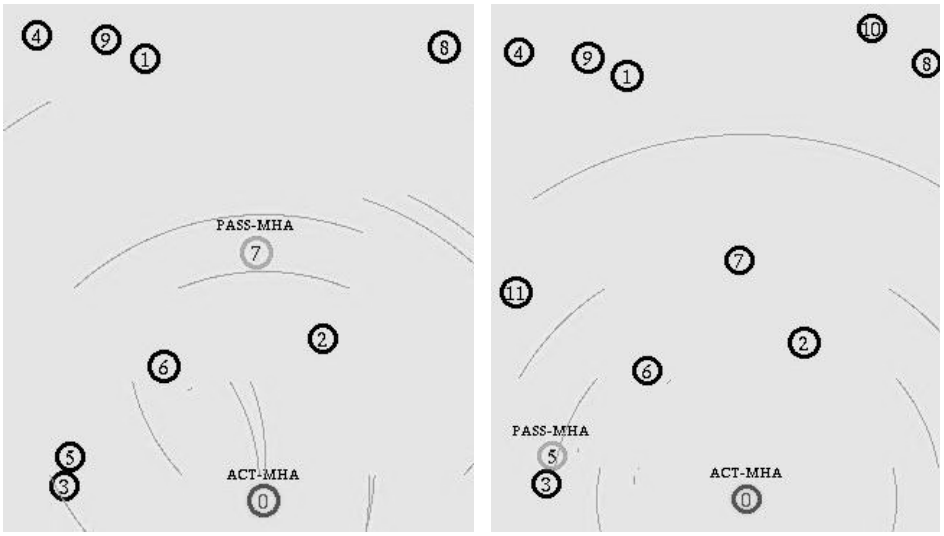


Figura 6.1 - Cenário da simulação RDAIPM com 10 e 12 nós.

Percebeu-se que na maior parte dos experimentos realizados, há distintas escolhas dos nós para agente passivos, o que torna este mecanismo flexível em situações de mobilidade ou inexistência temporária do nó.

Nesta seção, foram apresentados os resultados de desempenho do protocolo RDAIPM num ambiente de simulação de rede. Os resultados da eleição do *agente passivo* apresentados mostram um bom desempenho do modelo proposto levando em consideração que entre o envio e recepção das mensagens MIP e RDAIPM, outras mensagens de sinalização como a de camada de enlace e do protocolo de roteamento

AODV também trafegam na rede e que estamos simulando uma rede Ad Hoc IEEE 802.11b com uma capacidade efetiva de 5,5 Mbps.

6.2 - DESEMPENHO EM AMBIENTE EXPERIMENTAL

Os resultados colhidos nesta seção foram inicialmente avaliados num ambiente empírico com três plataformas, tendo desempenhos computacionais distintos. Esse método foi escolhido para entender melhor os mecanismos de funcionamento do RDAIPM numa rede MANET e WLAN infraestruturada, Figura 6.4.

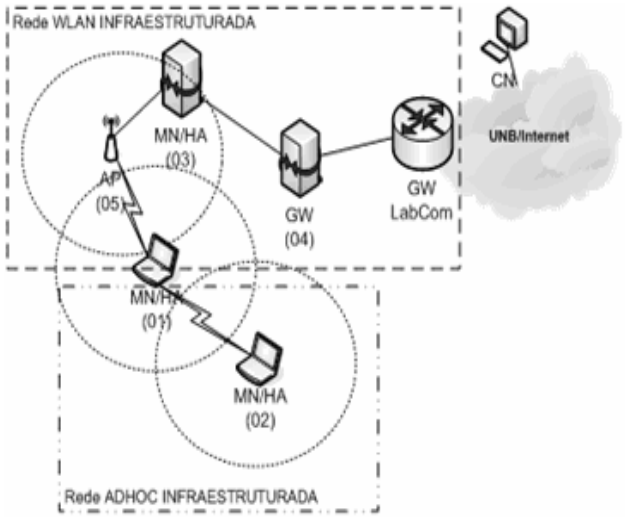


Figura 6.14 - Cenário MANET e WLAN.

A primeira plataforma é um laptop Pentium IV com tecnologia HT (Hyper Threading) de 2,8GHz de freqüência de núcleo, 1Gbytes de memória rodando o sistema operacional Fedora Core 4. A segunda plataforma é um laptop Pentium III de 850MHz de freqüência de núcleo, 512Mbytes de memória rodando o sistema operacional RedHat 9.0. A terceira plataforma é um PC Pentium III de 850MHz de freqüência de núcleo, 512Mbytes de memória rodando o sistema operacional Fedora Core 2, Tabela 6.3.

A implementação e instalação do RDAIPM é descrita em [Apêndice A]. Os tempos dos experimentos foram de 3.600 s para permitir um estudo mais detalhado do estado da MANET junto ao RDAIPM. A implementação de protocolo de roteamento escolhida foi o AODV-UU, *Uppsala University Ah Hoc Implementation Portal*, version 9.0 [60] por permitir tunelamento entre os nós da rede e encontrar automaticamente gateway para Internet.

Tabela 6.13 – Configuração das plataformas do cenário MANET e WLAN.

Plataforma	Endereço MAC	Endereço IP	Tipo de Nó	Padrão IEEE
Laptop Pentium IV, 2.8GHz, 1GB (01)	PCMCIA CISCO - ARTHEROS ar5001x 00:90:96:67:7A:FF	172.1.16.3 ou 192.168.1.100 - 172.1.10.1	MN/MA	eth1: 802.11b - ath0: 802.11b/g
Laptop Pentium III, (02)	PCMCIA ENTERASYS 00:01:F4:96:60:08	172.1.10.2	MN/MA	eth0: 802.11b
Desktop Pentium III, 850MHz, 512MB (03)	USB DWL-G122 D-Link 00:13:46:EA:37:CC - Ethernet: 00:E0:7D:F1:8D:B8	192.168.1.20 - 192.168.2.2	MN/MA *(rede nativa)*	usb0: 802.11b/g - eth0: 802.3
Access Point AirLAN (05)	Wireless: 00:90:96:30:D9:87 - Ethernet: 00:90:96:2D:02:E6	172.1.16.1-100	rede nativa	802.11b/g - 802.3 (x4)

Algo importante, percebido durante os testes, é o fato de a RFC3222 pedir que cada configuração do MIP tenha um tempo mínimo de 1s (visto que o tempo mínimo do *Lifetime* é de 3 s) entre envio de anúncio de agentes de mobilidades *(Agent Advertisements*), mostrada abaixo [3].

"..If sent periodically, the nominal interval at which **Agent Advertisements are sent SHOULD be no longer than 1/3 of the Advertisement Lifetime given in the ICMP header.** This interval MAY be shorter than 1/3 the advertised Lifetime. **This allows a mobile node to miss three successive advertisements before deleting the agent from its list of valid agents**..."

As análises descritas a seguir irão mostrar que o desempenho do RDAIPM quanto à detecção de falha no agente ativo, depende e muito do valor dado ao tempo mínimo para envio de anúncio de agentes de mobilidade na MANET e WLAN infraestruturada. Como descrito anteriormente na RFC3222, um nó MN tem de perder no mínimo três anúncios consecutivos antes de decidir se perdeu contato com o agente de mobilidade. O método escolhido para chaveamento do agente passivo em agente ativo é dado pela equação abaixo [ver Apêndice B], caso

```
actual.time_sec > (ha.current_adv->last.time_sec+3*agent_adv_value)
                                                                    (6.4)
```

então, deve-se chavear o agente passivo para agente ativo.

6.2.1 - Cenário manet

O experimento realizado neste cenário faz uso dos dois *laptops* descritos anteriormente. O primeiro é configurado como agente ativo e o segundo como MN. O código do RDAIPM é executado em ambas as plataformas. Foram realizados seis testes experimentais com valores de tempo de anúncios distintos sendo eles de 1, 2, 4, 6, 8 segundos. A Figura 6.5 apresenta os resultados obtidos para intervalos de tempo de anúncio de agentes de mobilidade versus o tempo necessário para eleição de um agente passivo após ter recebido um *RDAIPM_pass_REQ* [64].

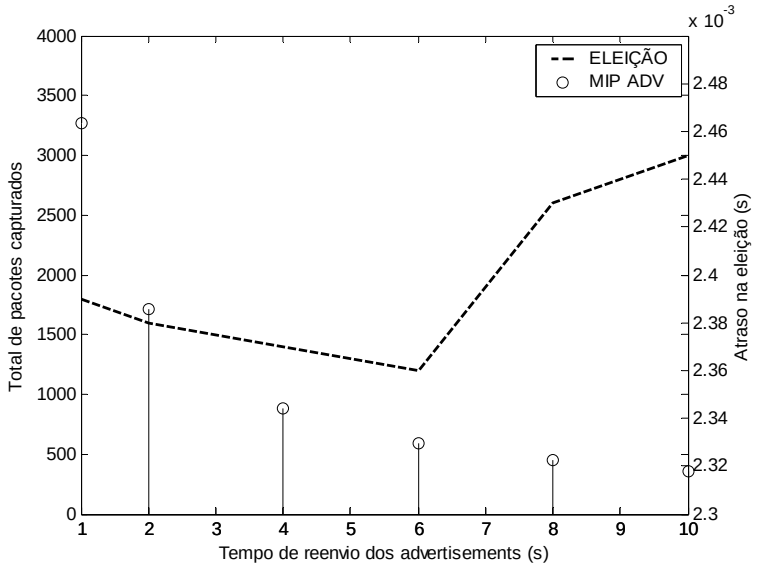


Figura 6.15 - Tempo de eleição de agente passivo versus intervalo de tempo para anúncio de agente ativo.

Percebe-se que quanto maior o tempo de intervalo de anúncio, menor é o overhead das mensagens dentro da MANET. Isso porque os intervalos de tempo sendo maiores, as mensagens *RDAIPM_Agt_adv* não irão ocasionar um *flooding* da rede. De posse dessas análises, pode-se definir a melhor relação entre o custo computacional, *flooding* e intervalo de tempo para anúncio de agente de mobilidade RDAIPM.

Em paralelo, foi realizada a eleição do agente passivo para intervalos de tempo de anúncio distintos e nenhuma discrepância foi encontrada quanto aos resultados já

obtidos. Assim, para esta faceta a solução desenvolvida apresentou um resultado excelente, abaixo de 2,5 ms, para eleição do agente passivo. Tem-se como melhor escolha os anúncios entre intervalos de tempo de 4 e 6 s por não ocasionar *flooding* excessivo na MANET e por permitir chaveamentos do agente passivo para ativo na ordem de 8 a 18 s respectivamente.

Na Figura 6.6 observa-se o total de pacotes AODV gerados, durante o experimento, entre as duas plataformas, em intervalos de tempos de anúncios distintos. À medida que o tempo vai passando, as duas plataformas tem de manter contato via troca de mensagens *Hello* do protocolo AODV que em muito depende da implementação disponível em [60].

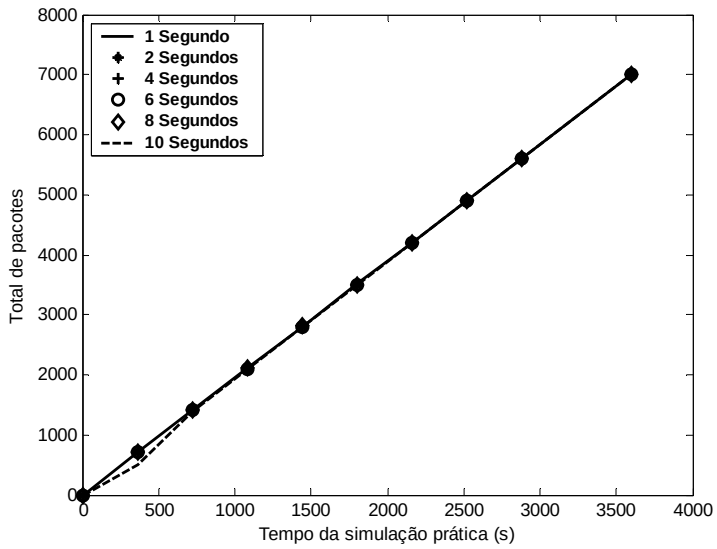


Figura 6.16 - Total Pacotes AODV gerados em 1 salto.

E como as rotas são mantidas ativas durante a troca de mensagens sinalização entre agente ativo e passivo, não há necessidade de disseminar mensagens *RREQ* e *RREP*, conforme descrito no capítulo 3.

As Figuras 6.7 e 6.8 apresentam respectivamente o total de pacotes *ADV&SOL* (*agents advertisements & agents solicitations*) e *MN_NodeElect_Ack* (*mensagem sendo enviada do agente passivo ao agente ativo após ter sido escolhido*) gerados entre o nó ativo e nó passivo.

Nas Figura 6.7 e 6.8 têm-se um total de pacotes capturados abaixo de 1000 pacotes, para intervalos de tempo de anúncio superiores a 4 s, comparáveis aos apresentados na Figura 6.5, pelo fato de ter-se a mensagem *MN_NodeElect_Ack* como resposta ao envio de um *ADV&SOL (sinalização realizada entre agente passivo e ativo).*

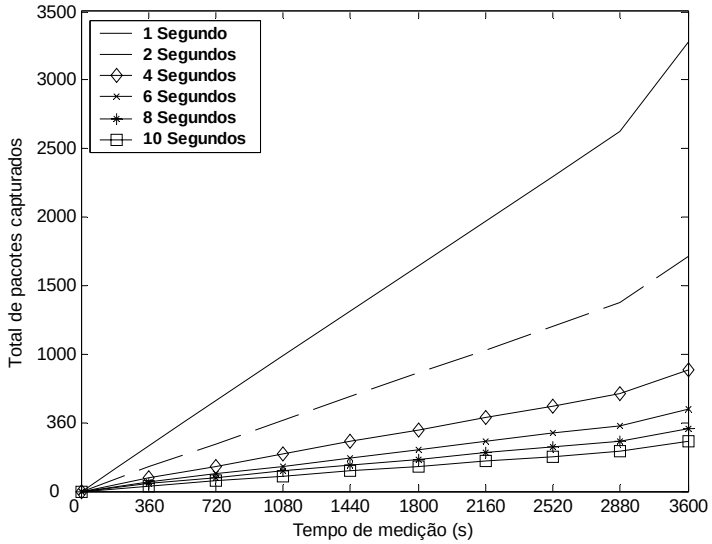


Figura 6.17 - Total Pacotes ADV&SOL gerados em 1 salto.

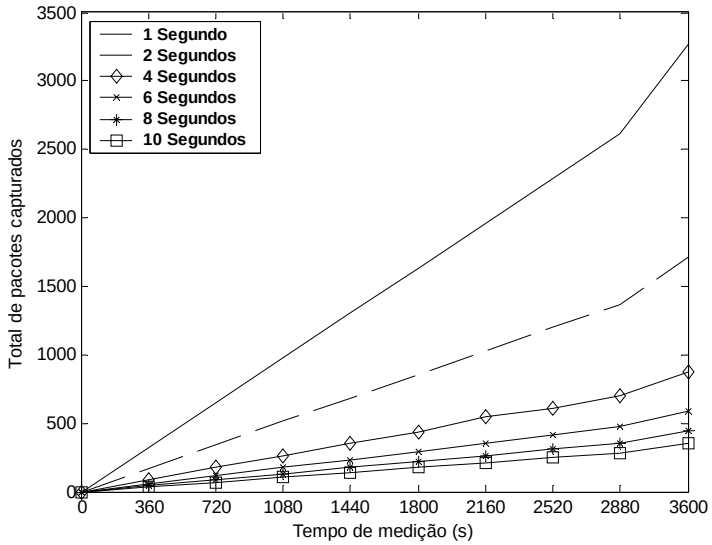


Figura 6.18 - Total Pacotes MN_NodeElect_Ack gerados em 1 salto.

Como descrito anteriormente, a fim de se evitar o flooding da rede, é necessário ficar abaixo do limiar de 4 s para intervalos de anúncio de agentes. Como esperado, os resultados obtidos para o total de pacotes na Figura 6.8 foram muito próximos aos da Figura 6.7 porque o agente ativo e passivo tem de manter um mecanismo para

verificar que ambos estão funcionando. Para isso, a cada mensagem de anúncio gerada pelo agente ativo, uma é gerada pelo agente passivo para avisar sobre sua disponibilidade para ser um futuro agente ativo.

Os resultados apresentados na Figura 6.9 e Tabela 6.4 são próximos porque o *MN_NodeElect_ADV* e *Pass_Agent_REQ* são enviados via Flooding e unicast respectivamente uma única vez na MANET. Então, o total de pacotes sempre será de 1, caso não haja ruptura de link entre ambas plataformas durante a simulação.

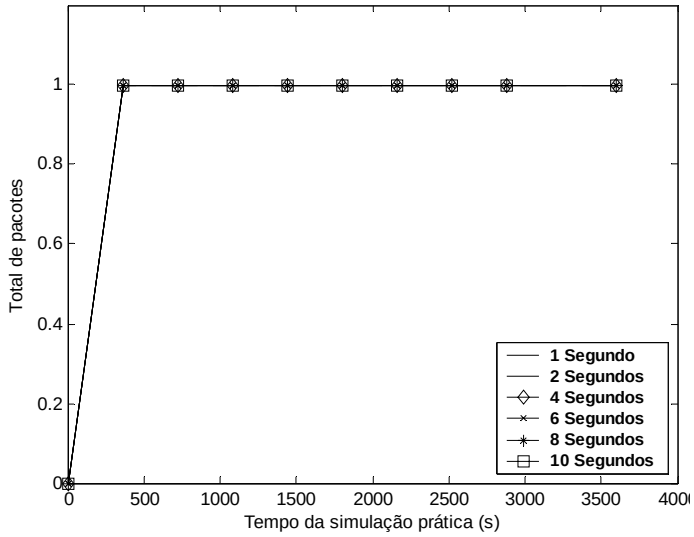


Figura 6.19 - Total Pacotes MN_NodeElect_ADV gerados em 1 salto.

Tabela 6.14 - Total Pacotes Pass_Agent_REQ gerados em 1 salto.

Tempo Para Reenvio dos Agent Advertisements [segundos]						
1	2	4	6	8	10	Tempo Simulação Prática [segundos]
0	0	0	0	0	0	0
1	1	1	1	1	1	360
1	1	1	1	1	1	720
1	1	1	1	1	1	1080
1	1	1	1	1	1	1440
1	1	1	1	1	1	1800
1	1	1	1	1	1	2160
1	1	1	1	1	1	2520
1	1	1	1	1	1	2880
1	1	1	1	1	1	3600

Outro excelente desempenho é dado pelo mecanismo de escolha do agente passivo tendo mostrado tempos inferiores a `2,5 ms` quando comparado ao tempo de processamento para se calcular a média do *RTT*, realizado pelo envio e recepção de uma mensagem do tipo *ping* (com a requisição de somente *(56+8) = 84 bytes*).

Como pode ser visto a seguir, a título de exemplo, (agente ativo com IP: `172.1.10.1`, e agente passivo com IP: `172.1.10.2`) o *RTT* do *ping* é na ordem de `4,2 ms` (cabe lembrar que o MN só saberá que foi eleito após análise do pacote *MN_NodeElect_ADV* contendo o seu endereço IP e, dessa forma, passar a enviar a mensagem *MN_NodeElect_Ack* para o seu agente ativo),

Comando utilizado no sistema operacional LINUX (*Fedora Core 4*):

```
#ping -s56 -c5 IP

PING 172.1.10.2 (172.1.10.2) 56(84) bytes of data.
64 bytes from 172.1.10.2: icmp_seq=0 ttl=64 time=4.37 ms
64 bytes from 172.1.10.2: icmp_seq=1 ttl=64 time=5.34 ms
64 bytes from 172.1.10.2: icmp_seq=2 ttl=64 time=3.89 ms
64 bytes from 172.1.10.2: icmp_seq=3 ttl=64 time=3.44 ms
64 bytes from 172.1.10.2: icmp_seq=4 ttl=64 time=4.01 ms

--- 172.1.10.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 3.449/4.213/5.341/0.639 ms, pipe 2
```

Os tempos de chaveamento, por estarem ligado ao tempo de intervalo de anúncio, parecem ser altos para detecção de falha no agente ativo [ver Apêndice C]. Se o sistema tiver de trabalhar com tempos de intervalos de anúncios inferiores a 1 s, o *flooding* da MANET seria severo e, assim, outros mecanismos de sinalização do RDAIPM e protocolo de roteamento a ser utilizado sofreria uma degradação gerando perdas de pacotes e atraso no envio dos mesmos.

Na Figura 6.10 têm-se o comparativo entre os tempos de reconfiguração esperados, teóricos e práticos. Para cada tempo de simulação de 1, 2, 4, 6, 8 e 10 segundos, espera-se uma reconfiguração dos nós RDAIPM após 3, 6, 12, 18, 24 e 30 segundos respectivamente. Os resultados experimentais se aproximam bem dos dados teóricos, até porque os pontos que se afastem da curva teórica mostram que a rede está recebendo mais pacotes de sinalização do AODV. Nessa análise, nota-se que, caso a rede tenha outros dados trafegando em direção ao agente passivo, haverá um

acréscimo no tempo computado para o agente passivo se tornar ativo. É bom lembrar que esses tempos, para reconfiguração do agente passivo em ativo na detecção de falha, são atrelados aos tempos de anúncios de agentes definidos em [3], [ver Apêndice C para visualização dos tempos de reenvio dos *agent advertisements* a cada 1, 2, 6 s e 3, 6, 18 s para reconfiguração do agente passivo em ativo].

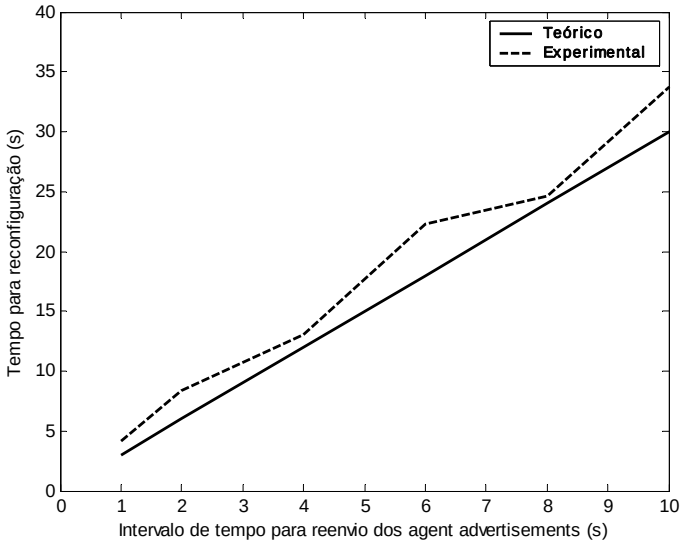


Figura 6.10 - Comparativo entre os tempos de reconfiguração esperados teóricos e práticos em 1 salto.

Como discutido no final do item 4.11, o sistema proposto não serve só para eleger um *agente passivo* e sim também fazer com que ele se auto configure se houver falha no agente ativo (considerando que não houve problemas com o *agente passivo*). Todos os nós na rede têm de ouvir os *agent advertisements* vindo do agente ativo e configurar o endereço IP da rota default para o nó que está disseminando essas mensagens. Então, cada vez que um novo agente ativo entra em atividade, os demais nós da rede autoconfiguram suas rotas default para esse novo agente com o comando "`route add default gw <endereço agente ativo>`". Essa foi a forma mais eficiente encontrada para que a continuidade de sessão fosse transparente para o usuário.

Nesta seção, foram apresentados os resultados de desempenho do protocolo RDAIPM num ambiente MANET real com as implementações descritas em [ver Apêndices A e B] e [60]. Os resultados apresentados mostram de novo um bom desempenho do modelo proposto tomando em conta que entre o envio e recepção das mensagens MIP

e RDAIPM, outras mensagens de sinalização como a de camada de enlace e o protocolo de roteamento AODV, entre outras, trafegam na rede. Para finalizar, é importante lembrar que o RDAIPM é uma extensão do MIPv4 [3] e tem de seguir rigorosamente os detalhes de implementação e configuração dos agentes de mobilidade em qualquer cenário.

6.2.2 - Comparativo wlan infraestruturada & manet

Uma análise de desempenho do RDAIPM foi realizada em cenário WLAN infraestruturado e MANET para avaliar a possibilidade de se ter o protocolo como possível implementação de agentes ativo e passivo numa rede WLAN infraestuturada e, assim, garantir a recuperação de falha dos agentes de mobilidade MIPv4. Desta vez, a rede WLAN infraestruturada conta com um laptop e um PC, conforme foram descritos no início do item 6.2 e na Figura 6.4. A seguir, tem-se a configuração básica que tem de ser implementada antes da realização dos testes,

- Todos os nós RDAIPM e MIPv4 [Apêndice A],[60,67]:
  - `[..@..]#aodvd –l` *{deamon para protocolo AODV}*
  - `[..@..]#dynmnd` *{deamon para protocolo RDAIPM/MIP}*

- Gateway MANET [60]:
  - `[..@..]#aodvd –l –w –I  eth1` *{deamon para protocolo AODV com gateway de accesso para Internet via interface de rede eth1}*
  - `[..@..]#vi /etc/rc.d/rc.local` *{addicionou-se: /sbin/iptables –t nat –A POSTROUTING -o [interface para rede internet] -j MASQUERADE}*

A Figura 6.11 apresenta os resultados obtidos entre um PC executando o agente ativo e um *laptop* executando o RDAIPM nativo, que será futuramente agente passivo. Os dados transmitidos e recebidos por ambas as plataformas trafegam por um AP (Access Point). Percebe-se que as mensagens do *agent advertisement* (MIP ou RDAIPM) representam 37,82% do total de pacotes capturados na NIC (*Network Interface Card*), na rota PC-*laptop*, que é superior às mensagens RDAIPM Node Elected Ack representando 24,53%. Esse valor é ligeiramente inferior aos dos *agents advertisement*, por ser enviado somente em tempo superiores, a fim de evitar *flooding* na rede.

A Figura 6.11 também mostra o quão pequeno é o overhead da mensagem *RDAIPM Node Elect Ack* quando comparado ao MIP Adv sendo de 60 e 67 bytes respectivamente [61].

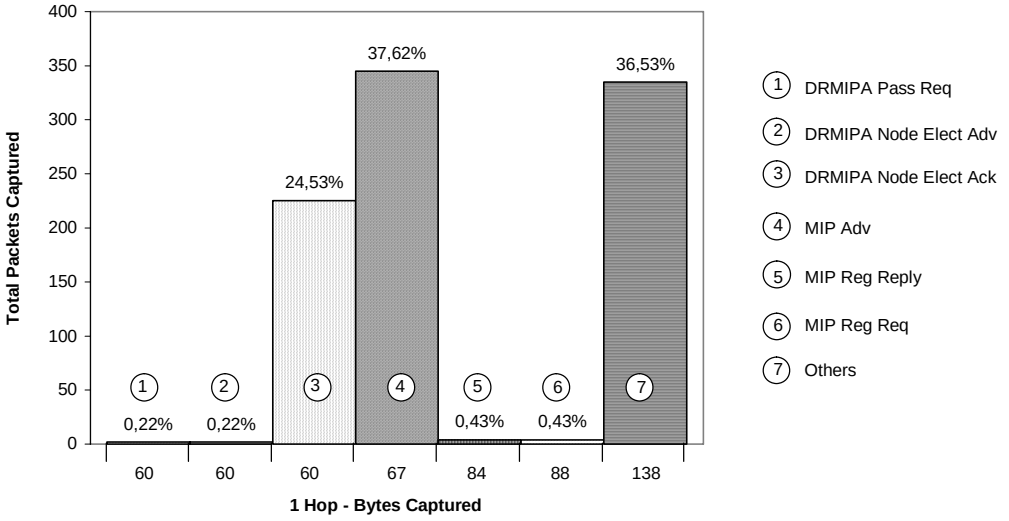


Figura 6.11 - Total Pacotes capturados na WLAN em 1 salto.

O maior overhead coletado durante os testes veio de mensagens de sinalização das outras camadas representando 36,53% do total de pacotes capturados com overhead de 138 bytes. O restante das mensagens do RDAIPM (*com overhead de 60 bytes*) e MIP (*com overhead de 84 e 88 bytes*) não ultrapassa os 0,22% e 0,43% respectivamente, do total de pacotes capturados, por serem enviadas uma única vez no início da sinalização de cadastro e eleição do agente passivo.

Na Figura 6.12 observa-se um comportamento totalmente diferente ao da WLAN infraestuturada, no cômputo dos pacotes gerados na MANET [61]. Isso se deve ao fato de ter-se um protocolo de roteamento com um mecanismo de sinalização ativo praticamente o tempo todo, através o envio de mensagens *Hello*. Os resultados mostram que as mensagens *RDAIPM Node Elect Ack* representam apenas 3,06% do total de pacotes capturados, contrariamente aos 80,71% gerados pelo protocolo AODV. A razão do total de pacotes capturados entre protocolo RDAIPM e MIP ser maior que na WLAN infraestruturada é porque cada nó na MANET é um roteador e as rotas têm de ser criadas para estabelecer rotas ativas entre agente ativo e passivo. O overhead dos outros protocolos presentes na MANET é similar aos presentes na WLAN infraestruturada, mas com um total de pacotes capturados de somente 9,15%.

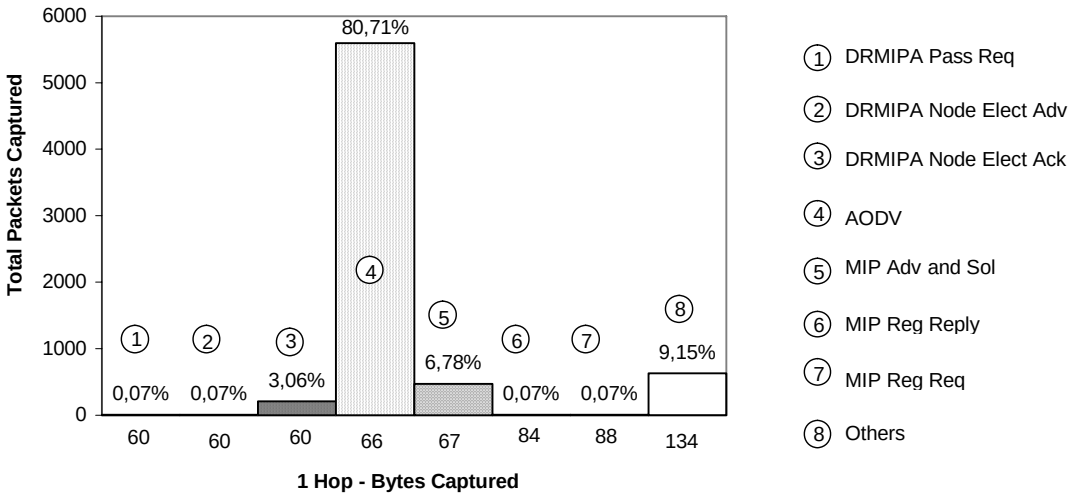


Figura 6.122 - Total Pacotes capturados na MANET em 1 salto.

Na Tabela 6.5 é mostrado o tempo médio no envio das mensagens RDAIPM configurado num cenário WLAN e MANET [64].

Tabela 6.15 - Comparativo dos tempos médios para eleição do agente passivo entre WLAN e MANET.

Mensagem: enviada - recebida	**Flag-A**	**Atraso Médio (ms)**	**Nó: De - Para**
Pass_agent_Req - MNnodelect_adv	A=10 - A=01	*3,10(wlan)* – 60,84*(manet)*	MN – Agente Ativo
MNnodelect_adv - MNnodelect_adv_ack	A=10 - A=01	*0,75(wlan)* – 2,40 *(manet)*	Agente Ativo - Agente Passivo

Como esperado, o RDAIPM responde como na WLAN infraestruturada, 0,75 ms contra 2,40 ms para eleição do agente passivo, por não ter um protocolo de roteamento em cada nó de mobilidade. Esse protocolo de roteamento sempre irá causar o flooding da MANET para manter suas rotas ativas (*seja ele reativo ou pró-ativo*). Todos os testes foram realizados cinco vezes. Nesta seção mostrou-se a versatilidade do RDAIPM quanto ao seu funcionamento numa rede WLAN infraestruturada e MANET. O protocolo RDAIPM pode garantir os mecanismos de auto configuração e recuperação de falhas dos agentes, em ambos cenários e apresenta excelentes resultados, podendo-se afirmar que o RDAIPM ajuda em ambos aspectos citados, quando implantado tanto numa rede infraestuturada ou ad hoc.

7 - CONCLUSÕES

Com o advento do MIP, um nó pode manter conectividade com as aplicações existentes, no nível de camada IP, mesmo estando em uma rede distinta e ainda conservar o seu endereço IP de origem, tanto em sua rede WLAN infraestruturada nativa ou estrangeira [3]. Com a possibilidade de manutenção de conectividade na camada IP, torna-se importante a implementação desse mecanismo também em redes que não sejam do tipo WLAN infraestruturadas, já que o usuário final sempre desejará manter conectividade independente da infraestrutura de rede existente.

Nesse contexto, uma das possibilidades são as redes MANETs, por serem de características temporárias e autoconfiguráveis, em que quaisquer dos nós dentro do alcance de transmissão de suas interfaces sem fio podem formar uma rede de comunicação, sem a dependência de uma infraestrutura preestabelecida.

Essas considerações motivaram o uso do MIPv4 também nas redes MANET para que os nós vindos de outras redes pudessem usufruir a mobilidade IP [45], mas nenhuma das propostas acadêmicas estudadas procuraram levar os agentes de mobilidade a um cenário de rede Ad Hoc, apesar da grande necessidade de manutenção de conexão no nível de camada IP.

Com base nas argumentações apresentadas acima, o principal objetivo desta tese foi propor o uso dos agentes de mobilidade IP num cenário de mobilidade total com um mecanismo de reconfiguração dinâmica desses agentes [51,61,62,64,65]. Com isso, esta nova abordagem trouxe inúmeros desafios decorrentes da mobilidade dos agentes, que antes eram considerados fixos e não havia qualquer solução de recuperação de falha.

A partir da proposta de reconfiguração dinâmica dos agentes de mobilidade IP surgiu o acrônimo RDAIPM, sendo este responsável por manter as funcionalidades dos agentes do protocolo MIPv4, de forma transparente, tanto em uma rede nativa ou estrangeira. Então, sempre haverá a disponibilidade dos agentes de mobilidade móveis e a manutenção dos serviços providos por estes em uma rede MANET, mesmo em situações de falhas ou mobilidade dos agentes.

Um grande desafio da presente tese foi manter as funcionalidades dos agentes de mobilidade dentro da MANET com relação às regras estipuladas na RFC-3222 do MIPv4 [3], por causa da reconfiguração dinâmica dos agentes. Ou seja, qualquer nó que for requerer os mecanismos MIP, independentemente da infraestrutura que exista, tem de manter a continuidade de sessão da transação de seus dados e, em situações de mobilidade na camada IP, necessitam das funções dos agentes MIPv4 para manter essa continuidade. Nesse aspecto, foram também analisadas diversas possibilidades para encaminhamento das mensagens e, sendo assim, o mecanismo de tunelamento descrito em [23] foi escolhido por viabilizar a manutenção do encaminhamento da sinalização e mobilidade RDAIPM/MIPv4 na MANET [3,65].

Dessa forma, novas mensagens de sinalização foram propostas para garantir a escolha e reconfiguração do atual e futuro agente de mobilidade [51,61,62,64,65]. Entre elas, a primeira mensagem de sinalização é uma extensão do *agent advertisement*, com a inclusão de uma Flag A de dois bits. Com essa nova Flag, os nós saberão da existência de um agente MIPv4 e RDAIPM ao mesmo tempo, o que garante as funcionalidades descritas na RFC3222 quanto ao anúncio dos agentes de mobilidade na MANET. A segunda mensagem, *Passive_agent_Request*, surge dos nós da rede e é enviada ao agente ativo, após recepção de um *agent advertisement*. O nó escolhido como agente passivo (sendo o nó que irá ficar na espera de qualquer falha eventual do agente ativo) é previamente eleito pelo agente ativo, e tem de enviar a quarta mensagem do tipo *Mnnodelect_Ack* para confirmar o recebimento da terceira mensagem *Mnnodelect_Adv*. Foi também proposto um mecanismo para autoconfiguração dos agentes ativo e passivo, caso o agente ativo não consiga eleger um agente passivo antes de um tempo preestabelecido, por causa de uma falha repentina ou não.

Para validar o protocolo proposto, testes foram realizados em ambiente de simulação e plataformas experimentais. No ambiente de simulação, NS2 [19], foi modificado a versão do MIPv4 e outros arquivos de grande importância para rastreio e análise da performance das novas mensagens de sinalização. Em seguida, fez-se uma modificação do código do MIPv4 distribuído pela universidade da HUT [67]. As contribuições desta tese foram implementadas nesse código para dar suporte às mensagens do RDAIPM, seguindo rigorosamente as premissas descritas em [3]. Os

resultados obtidos em ambos ambientes mostraram um bom desempenho do RDAIPM, de acordo com os objetivos propostos, sendo eles a eleição e reconfiguração dos agentes ativo e passivo sem gerar *flooding* excessivo na rede. Os tempos de envio e recebimento das mensagens RDAIPM, incluindo os tempos de processamento, também tiveram um excelente desempenho quando comparados às mensagens de sinalização do MIPv4 em ambos ambientes de testes. Os dados de desempenho referentes à reconfiguração do agente passivo em ativo (item 7.2.1) tiveram uma ótima aproximação dos dados teóricos, confirmando o bom funcionamento do mecanismo proposto. Cabe apontar que os tempos de reconfiguração são atrelados aos tempos de reenvio dos *agent advertisements*, predefinidos na RFC3222. Então, caso se deseje configurar o tempo para reenvio dos *agent advertisements* abaixo de 1s [3], e, com isso, ter uma reconfiguração abaixo dos 3 s (tempo mínimo definido em [3] para que um nó MIPv4 possa definir se perdeu ou não a sinalização do agente de mobilidade), ter-se-á o risco de degradar a performance da mobilidade IP com um aumento do *flooding* na rede.

Para finalizar, todos os mecanismos referentes ao funcionamento, configuração e implementação do RDAIPM foram reunidos e proposto na forma de um draft a ser submetido na lista de discussão do IETF [B], a fim de formalizar a tese proposta.

7.1 - TRABALHOS FUTUROS

A seguir serão descritos possíveis trabalhos que possam dar continuidade a esta tese, com o intuito de melhorar e aplicar o RDAIPM em outros entornos de rede.

Um estudo terá de ser feito para verificar a escalabilidade do RDAIPM para áreas densas fazendo uso de novas tecnologias de rede de acesso como WiMAX. Espera-se um bom desempenho por parte do protocolo tendo como principal métrica a eleição do agente passivo a somente 1 salto do agente ativo. Com o advento da disponibilização no mercado de interfaces PCMCIA Wimax, no mais tardar final de 2006 [83], poder-se-á, em todas as plataformas móveis que forem implementar o protocolo RDAIPM, ter uma maior cobertura da MANET/RDAIPM. Com duas interfaces dessas, ter-se-á também acesso à rede Internet e, assim, testar melhor os

mecanismos de reconfiguração dos agentes RDAIPM entre redes nativas e estrangeiras.

Novos parâmetros, como os tempos para envio dos agent advertisements, números de sequência da manutenção das rotas ativas do protocolo de roteamento, entre outros, terão de ser estudados, implementados e configurados quando da eleição, reconfiguração e autoconfiguração de um agente ativo e/ou passivo em momento de falha, viabilizando o correto funcionamento do RDAIPM na MANET. Esses parâmetros poderão ser atrelados dinamicamente ou não ao protocolo de roteamento escolhido pelo usuário da MANET ou administração da rede na qual ele estiver presente. Isso porque sempre surgirá a dúvida de quem irá escolher o protocolo de roteamento a ser usado na MANET, caso o usuário seja de uma rede estrangeira sem ter conhecimento do tipo de roteamento usado naquele momento. Será que cada usuário terá de ter em seu equipamento a disponibilização de todos os protocolos de roteamento da MANET e assim garantir a sua inclusão no grupo alheio?

Terão de ser realizados mais testes de desempenho do RDAIPM usando outros tipos de protocolos de roteamento Ad Hoc (podendo ser eles reativos, pró-ativos e até mesmo híbridos), e dessa forma garantir uma perfeita interoperabilidade entre redes e protocolos de roteamento distintos. A idéia aqui é obter transparência nos mecanismos de sinalização do RDAIPM, independentemente do protocolo de roteamento utilizado.

A busca por novos mecanismos de segurança (autenticação das mensagens RDAIPM e dos agentes) terá de ser estudada e estes implementados para garantir a integridade e autenticidade das mensagens trocadas entre agentes de mobilidade. Corre-se aqui o risco de um nó malicioso infiltrar-se na rede nativa ou estrangeira e forjar os endereços IP (entre outras ameaças) dos agentes ativo e passivo e assim se tornar um deles.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] IEEE Document P802.11/D6.1.97/5,*"Wireless LAN, MAC, and PHY Specifications,"* June 1997, Alpha Graphics #35, 10201 N. 35th Ave., Phoenix, AZ 85051.

[2] J.B. Postel, ed.,*"Internet Protocol"*, IETF RFC 791, Sept. 1981.

[3] C. Perkins, ed.,*"IP Mobility Support for IPv4"*, IETF RFC 3222, January 2002.

[4] C. Perkins, *"Mobile IP: Design Principles and Practice"*, Addison-Wesley Longman, Reading, Mass., 1998.

[5] C. Perkins, *"Mobile IP,"* IEEE Comm., Vol. 35, No. 5, 1997, pp. 8499.

[6] R. Ghosh and G. Varghese, *"Fault-Tolerant Mobile IP"*, *MReport WCUCS-98-11. Washington Univ*, Apr. 1998

[7] Khurana S., Kahol A., Gupta S.K.S., Srimani P.K., *"An Efficient Cache Maintenance Scheme for Mobile Environment"*, Distributed Computing Systems, 2000. Proceedings. 20th International Conference on , 10-13 April 2000 pp.530 – 537.

[8] J. Ahn; C.Sun Hwang, *"Low-cost fault-tolerance for mobile nodes in mobile IP based systems"*, Distributed Systems Workshop, 2001 International Conference on , 16-19 April 2001 pp. 508 – 513.

[9] J. Ahn; C.Sun Hwang, *"Efficient fault-tolerant protocol for mobility agents in mobile IP"*, Parallel and Distributed Processing Symposium., Proceedings 15th International , 23-27 April 2001, pp.1273 – 1280.

[10] J. W. Lin, J. Arul, *"An Efficient Fault-Tolerant Approach for Mobile IP in Wireless Systems"*, IEEE Transactions on Mobile Computing , July-September 2003 (Vol. 2, No. 3).

[11] J. W. Lin, J. Arul, *"An Efficient Fault-Tolerant Approach for Mobile IP in Wireless Systems"*, Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on, Volume: 1 , June 28 - July 1, 2004 pp.556 – 561.

[12] J. Solomon, *"Mobile IP: The Internet Unplugged"*, Prentice Hall, Englewood Cliffs, N.J., 1998.

[13] P. Calhoun and C. Perkins, *"Tunnel Establishment Protocol (TEP),"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-mobileip-calhoun-tep-00.txt, Aug. 1997 (work in progress).

[14] C. Perkins, *"Mobile-IP Local Registration with Hierarchical Foreign Agents,"* ftp://ftp.ietf.org/internet-drafts/draft-perkins-mobileip-hierfa-00.txt, Feb. 1996 (work in progress).

[15] C. Perkins and J. Tangirala, *"DHCP for Mobile Networking with TCP/IP,"* Proc. IEEE Int'l Symp. Systems and Comm., June 1995, pp. 255261.

[16] R. Droms, *"Dynamic Host Configuration Protocol"*, IETF RFC 2131, Mar. 1997.

[17] S. Alexander and R. Droms, *"DHCP Options and BOOTP Vendor Extensions,"* IETF RFC 2132, Mar. 1997.

[18] J. Solomon and S. Glass, *"Mobile-IPv4 Configuration Option for PPP IPC,"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-pppext-ipcp-mip-02.txt, July 1997 (work in progress).

[19] NS-2, *http://www.isi.edu/nsnam/ns/tutorial/*, Creating Wired-cum-Wireless and Mobile IP Simulations in ns, 2003.

[20] C. Perkins, ed., *"IP Mobility Support Version 2,"* draft-ietf-mobileip-v2-00.txt, Nov. 1997 (work in progress).

[21] P. Bhagwat, C. Perkins, and S.K. Tripathi, *"Network Layer Mobility: An Architecture and Survey,"* IEEE Personal Comm., Vol. 3, No. 3, June 1996, pp. 5464.

[22] S.E. Deering, ed., *"ICMP Router Discovery Messages,"* IETF RFC 1256, Sept. 1991.

[23] C. Perkins, *"IP Encapsulation Within IP,"* IETF RFC 2003, May 1996.

[24] C. Perkins, *"Minimal Encapsulation Within I,"* IETF RFC 2004, May 1996.

[25] V.L. Voydock and S.T. Kent, *"Security Mechanisms in High-Level Networks,"* ACM Computer Surveys, Vol. 15, No. 2, June 1983, pp. 135171.

[26] R.L. Rivest, *"The MD5 Message-Digest Algorithm,"* IETF RFC 1321, Apr. 1992.

[27] S. Bradner and A. Mankin, *"The Recommendation for the IP Next Generation Protocol,"* IETF RFC 1752, Jan. 1995.

[28] C.E. Perkins and D.B. Johnson, *"Route Optimization in Mobile-I,"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-mobileip-optim-07.txt, Nov. 1997 (work in progress).

[29] S. Kent and R. Atkinson, *"IP Authentication Header,"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-ipsec-auth-header-03.txt, Nov. 1997 (work in progress).

[30] S. Kent and R. Atkinson, *"IP Encapsulating Security Payload (ESP),"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v2-02.txt, Nov. 1997 (work in progress).

[31] C. Perkins and P. Bhagwat, *"A Mobile Networking System Based on Internet Protocol (IP),"* Proc. USENIX Symp. Mobile and Location-Independent Computing, Aug. 1993, USENIX Assoc., pp. 6982.

[32] D.B. Johnson, *"Scalable and Robust Internetwork Routing for Mobile Hosts"*, Proc. 14th Intl. Conf. Distributed Computing Systems, June 1994, pp. 211.

[33] G. Montenegro, *"Reverse Tunneling for Mobile I,"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-mobileip-tunnel-reverse-04.txt, Aug. 1997 (work in progress).

[34] G. Pall et al., *"Point-to-Point Tunneling Protocol-PPT,"* ftp://ftp.ietf.org/internet-drafts/draft-ietf-pppext-pptp-02.txt, July 1997 (work in progress).

[35] S. Cheshire and M. Baker, *"Internet Mobility 4x4,"* Proc. ACM SIGCOMM Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm., Vol. 26, No. 4, ACM SIGCOMM Computer Comm. Rev., ACM Press, New York, 1996, pp. 318329.

[36] David A. Maltz, P. Bhagwat, *"MSOCKS: An Architecture for Transport Layer Mobility"* INFOCOM'98, IEEE 1998.

[37] Wesley M. Eddy, *"At what Layer Does Mobility Belong?"*, Topics in Internet Technology, IEEE Communications Magazine, Oct. 2004, pp. 155-159.

[38] J. Zao et al., *"A Public-Key Based Secure Mobile IP,"* Proc. ACM Mobicom 97, ACM, New York, Oct. 1997, pp. 173184.

[39] W. Palter et al., *"Layer Two Tunneling Protocol L2TP,'"* ftp://ftp.ietf.org/ internet-drafts/draft-ietf-pppext-l2tp-08.txt, Nov. 1997 (work in progress).

[40] R.H. Katz, *"Adaptation and Mobility in Wireless Information Systems"*, IEEE Personal Comm., Vol. 1, No. 1, 1994, pp. 617.

[41] D.B. Johnson and D.A. Maltz, *"Protocols for Adaptive Wireless and Mobile Networking,"* IEEE Personal Comm., Vol. 3, No. 1, Feb. 1996, pp. 3442.

[42] D.B. Johnson and D.A. Maltz, *"Dynamic Source Routing in Ad Hoc Wireless Networks,"* in *Mobile Computing*, T. Imielinski and H. Korth, eds., Kluwer Academic Publishers, 1996, pp. 153181.

[43] C. E. Perkins, E. M. Belding-Royer, e S. Das, *"Ad Hoc On Demand Distance Vector (AODV) Routing"*, RFC-3561 (*experimental*), IETF, July 2003.

[44] D. B. Johnson, D. A Maltz, Y. Hu, e J. G. Jetcheva, *"The Dynamic Source Routing Protocol for mobile adhoc networks"*, IETF Internet draft, draft-ietf-manet-dsr-07.txt, Fevereiro 2002.

[45] U. Jönsson, F. Alriksson, T. Larsson, P. Johansson , e G. Q. Maguire Jr., *"MIPMANET: Mobile IP for mobile ad hoc networks"*, Proceedings of IEEE MohiHoC 2000, pp. 75-85, Novembro 2000.

[46] J. Broch, D. Maltz, e D. Johnson, *"Supporting Hierarchy and Heterogenous Interfaces in multihop wireless adhoc networks"*, Proceedings of 4th International Symposium on Parallel architectures, algorithms and networks (I-SPAN), pp. 370-375, Junho 1999.

[47] Y. Sun, E. M. Belding-Royer, e C. E. Perkins, *"Internet Connectivity for Ad hoc Mobile Networks"*, International Journal of Wireless Information Networks special issue on Mobile Ad hoc Networks, 9(2), Abril 2002.

[48] M. Marina e S. R. Das, *"On-Demand Multipath Distance Vector Routing"*, Proceedings of ICNP 01, pp. 14-23, Novembro 2001.

[49] Haas, Z.J., Pearlman, M.R., Samar, P., *"The Zone Routing Protocol (ZRP) for Ad Hoc Networks"*,IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt, Junho 2002.

[50] S. Corson and J. Marker, *"Mobile ad hoc networking (MANET) "*, Routing protocol performance issues and evaluation consideration. RFC-2501 (informational), IETF, January 1999.

[51] G. Amvame-Nze**,** C. Jacy Barenco Abbas, L. J. García Villalba, *"Novel Dynamic Reconfiguration of Mobile IPv4 Agents Fully Integrated in MANET"*, Proceedings of the 4th International Information and Telecommunication Technologies Symposium - I2TS/IEEE R9, 2005, pp. 46-51.

[52] Padmini Misra, *"CBRP (Cluster Based Routing Protocol)"*, Publicado Online at the The Ohio State University, Computer Science and Engeneering, 1999, (acessado no dia 29 de agosto 2006).

[53] Broch, J. et al., *"A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols"*, MOBICOM'98, pp. 85-97, Texas, Estados Unidos, 1998.

[54] T. Clausen, E. P. Jacquet, *"Optimized Link State Routing Protocol (OLSR)"*, RFC-3626 (experimental), IETF, October 2003.

[55] Das, S. R. et al., *"Performance Comparison of Two On-demand Routing Protocols for AdHoc Networks"*, INFOCOM'2000, Tel-Aviv, Israel, March 2000.

[56] V. Devarapalli, A. Petrescu and P. Thubert, *"Network Mobility (NEMO) Basic Support Protocol"*, RFC-3963 (standard track), IETF, January 2005.

[57] Chien-Fu Cheng, Shu-Ching Wang, Tyne Liang, *"Multi-agent schema of Mobile IP protocol for mobile environment"*, *ACM SIGOPS Operating Systems Review*, Volume 39, October 2005, pp. 46-65.

[58] Silva, L. M. Lima, F. F. Lopes, D. Junior, H. A. *"Radiodifusão Sonora Digital Terrestre: Sistemas Existentes e suas Principais Características"*. Apostila, Fevereiro 2005.

[59] *http://grouper.ieee.org/groups/802/11/*, acessado no dia 10 de Agosto 2006.

[60] AODV-UU, *Uppsala University Ad Hoc Implementation Portal*, version 9.0, http://core.it.uu.se/adhoc/ImplementationPortal (acesso em 20 agosto 2006).

[61] G. Amvame-Nze**,** C. J. Barenco Abbas, L. J. García Villalba. *"Evaluation of The Dynamic Reconfiguration of Mobile IPv4 Agents in MANET"*. The IASTED International Conference on Communications Systems for WNET-Wireless Networks and Emerging Technologies, 2006, Banff, Alberta. The Sixth IASTED International Multi-Conference on Wireless and Optical Comunications. Calgary, CANADA. ACTA Press, Julho 2006, pp. 352-357.

[62] G. Amvame-Nze**,** C. Jacy Barenco; L. Regal Dutra. *"Reconfiguração Dinâmica de Agentes Mobile IPv4 em Redes 802.11b MANET"*. IADIS Conferencia Ibero-Americana WWW/Internet 2005, 2005, Lisboa, p. 630-634.

[63] A. Campbell,*"Cellular IP"*, draft-ietf-mobileip-cellularip-00, expirou Junho 2000.

[64] G. Amvame-Nze, Flavio E. de Deus, Roque Lambert, C. Jacy Barenco Abbas and L. J. García Villalba. *"Performance of WLAN and MANET Networks for New Auto-Configured Móbile IP Agents"*. IEEE-ITS2006, The International Telecomunications Symposium, Stembro 2006, Fortaleza, Brasil. IEEE Xplore, 2006.

[65] Claúdia J. Barenco Abbas, Georges Amvame-Nze and L. Javier Garcia Villalba. *"Performance Evaluation of Mobile IP Agents' Auto-Reconfiguration Mechanisms in MANET"*. EUC2006, The 2006 IFIP International Conference on Embedded and Ubiquitous Computing, Seoul, Korea. Lecture Notes in Computer Science – LNCS**,** Springer-Verlag, 2006, pp. 844-853.

[66] Erik Nordstrom, Per Gunningberg, Christian Tschudin, *"POSTER: Comparison of Forwarding Strategies in Internet Connected MANETs"*, The Fifth ACM International Symposium on Móbile Ad Hoc Networking and Computing, MOBIHOC'04, Tokyo, Japan.

[67] The Dynamics Mobile IP system, *Helsinki University of Technology (HUT)*, version 0.8.1, http://dynamics.sourceforge.net/ (acesso em 20 agosto 2006).

[68] Fedora Core 2/3/4, RedHat 9.0, *Fedora Project Board*, http://fedora.redhat.com/ (acesso em 5 setembro 2005).

[69] Ethereal version 0.10.12, *Ethereal Network Analyzer*, http://www.ethereal.com/ (acesso em 5 setembro 2005).

[70] Sockets, *GNU facilities for interprocess communication using sockets*, http://www.gnu.org/software/libc/manual/html_node/Sockets.html (acesso dia 5 de setembro 2005).

[71] Barreto, P. Solis; Amvame-Nze**,** G. ; Lambert, R. et. al. *"Open Source Software for Evaluation of Applications and Traffic Measurement in an Experimental Testbed for Converged Networks"*. IEEE-TridentCom2006, Bracelona, Spain, 2006. IEEE Xplore, 2006.

[72] Rappaport 2002, *"Wireless Communications: Principles and Practice"*, Second Edition. Prentice Hall, NJ, 2002, USA.

[73] Jun J., Peddabachagari P. and Sichitiu M., *"Theoretical Maximum Throughput of IEEE 802.11 and its Applications"*, 2nd International Symposium on Network Computing and Applications, NCA-03, 2003, USA.

[74] Douglas E. Comer, David L. Stevens, *"Internetworking with TCP/IP Vol. II: ANSI C Version: Design, Implementation, and Internals"* , Prentice Hall, 3rd edition, June 15, 1998, USA.

[75] W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Richard W. Stevens, *"UNIX Network Programming, Volume 1: The Sockets Networking API, Third Edition"*, Addison-Wesley Professional, 3rd edition, October 22, 2003, USA.

[76] Andrew S. Tanenbaum, *"Computer Networks"*, 4.ed. Englewood cliffs, Prentice-Hall, 2003, USA, pp. 912.

[77] Reinhard Diestel, *"Graph Theory"*, 3rd Edition. http://www.math.uni-amburg.de/home/diestel/books/graph.theory/download.html, 2005, New York, USA.

[78] Paulo Blauth Menezes, *"Linguagens Formais e Autômatos"*, 5ª Edição, Instituto de Informática da UFRGS, Editora Sagra Luzzatto, 2005, Porto Alegre, RS, Brasil.

[79] Tiarajú Asmuz Diverio, Paulo Blauth Menezes, *"Teoria da Computação"*, 2ª Edição, Instituto de Informática da UFRGS, Editora Sagra Luzzatto, 2005, Porto Alegre, RS, Brasil.

[80] HMIPv6, *"Hierarchical Mobile IPv6"*, INRIA (Institut National de Recherche en Informatique et en Automatique), RFC-3963 (experimental), IETF, August 2005, http://www.inrialpes.fr/planete/people/bellier/hmip.html, (acesso dia 29 de agosto 2006).

[81] C. Perkins, ed., *"Mobility Support for IPv6"*, IETF RFC 3775, June 2004.

[82] R. Ramjee, *"IP micro-mobility support using HAWAII"*, draft-ietf-mobileip-hawaii-00, expirou Dezembro 1999.

[83] http://www.wimax.com/commentary/blog/blog-2006/blog3-15-2006fo1, (acesso dia 5 de setembro 2006).

APÊNDICES

A - IMPLEMENTAÇÃO BÁSICA EM AMBIENTE EXPERIMENTAL

A implementação do RDAIPM (*Dynamic Reconfiguration of Mobile IP Agents*) é baseada no sistema *Dynamics Mobile IP*, um software em linguagem C para o sistema operacional Linux [68,74,75], desenvolvido originalmente na *Helsinki University of Technology*. A versão HUT do *Mobile IP* foi escrita em conformidade com a RFC-3222 (*a RFC-2002 tornou-se obsoleta*), mas ela inova ao permitir uma hierarquia em árvore dos *Foreign Agents* entre o *Home Agent* e o FA mais próximo do MN (no nível mais baixo) [3]. A idéia dos desenvolvedores era reduzir as latências no caso de se ter um MN se deslocando de uma sub-rede a outra (ao invés de sempre informar a *home network* de sua localização atual, o sistema *Dynamics* informa apenas o FA mais próximo na hierarquia do túnel).

Efetuou-se o *download* do código *Dynamics Mobile IP* (versão 0.8.1) [67] e deu-se início ao procedimento de modificação do mesmo como demandado pelo algoritmo proposto. Iniciou-se a implementação das mensagens RDAIPM nos agentes de mobilidade. HA sendo a peça chave para cadastro e encaminhamento dos pacotes fora da rede nativa suscitou a nossa maior dedicação.

O software *Mobile IP* consiste basicamente em rodar um *daemon* para cada uma das três entidades MIP: *Home Agent*, *Foreign Agent* e *Mobile Node* (*daemons* dynhad, dynfad e dynmnd, respectivamente). O diretório fonte contém nove pastas e um *makefile*. Os três sub-diretórios principais são */ha*, */fa e /mn*. Outro diretório importante (que possui um papel de suporte ao funcionamento das entidades MIP) é o */other*, que possui um arquivo de biblioteca (*message.h*) responsável pela declaração de todas as estruturas das mensagens MIP (de acordo com a antiga RFC-2002) e outras mensagens de extensão próprias do *Dynamics*. Foi neste arquivo que se inseriu as declarações das estruturas das mensagens RDAIPM.

Antes da execução dos *daemons* do *Mobile IP* em cada máquina, deve ser realizada a configuração dos mesmos através dos arquivos *".conf"* correspondentes (*dynhad.conf, dynfad.conf, dynmnd.conf*). No arquivo de configuração do HA, por exemplo, deve-se entrar com o endereço da interface de rede na qual trafegarão as

mensagens MIP, o intervalo de envio dos *agent advertisements*, o valor da porta UDP para as mensagens de registro (o *default*, segundo a RFC3222, é 434), o número máximo de *bindings*, o tipo de tunelamento, a lista dos MNs autorizados (pode ser um *range* de endereços de rede), parâmetros de segurança, entre outros. No arquivo de configuração do MN deve-se fornecer o *home address*, o endereço do HA e alternativos (se existentes); pode-se habilitar ou não o desencapsulamento das mensagens pelo FA (caso contrário seria feito pelo próprio MN), além de outras configurações como o modo de tunelamento, o *lifetime* do túnel e o envio de *agent solicitations* em um determinado intervalo pré-definido.

Durante o período de implementação do algoritmo RDAIPM, tomou-se a Figura 5.10 (*sinalização da eleição do agente passivo*) como elemento norteador. As mensagens trocadas entre MA e MN foram implementadas na ordem em que aparecem na figura.

Um elemento de fundamental importância no funcionamento do código é a *flag* de dois bits de controle denominada simplesmente *flag* A. Essa aparece repetidamente em todas as mensagens RDAIPM implementadas e possui a finalidade de promover a identificação dos tipos de mensagem, de acordo com sua origem e destino, pelos componentes da rede (*agentes móveis e nós comuns*). A *flag* A permite, por exemplo, identificar se uma mensagem *agent advertisement* provém de um agente RDAIPM, Tabela A.1. A tabela com os possíveis valores e significados assumidos pela mesma se encontra abaixo.

Tabela A.1 - Flag A.

Valor (binário)	DE	PARA
00	MN	AGENTE ATIVO
01	AGENTE ATIVO	AGENTE PASSIVO
10	AGENTE PASSIVO	AGENTE ATIVO
11	AGENTE ATIVO	MN

ADAPTAÇÃO DO ETHEREAL

À medida que se prosseguiu o processo de criação das mensagens do RDAIPM, tornou-se necessária a visualização das mesmas assim como dos seus campos. Desse modo, pôde-se auditar as alterações do código e a eficácia das mesmas. Para esse fim, decidiu-se alterar o código fonte de um *sniffer* (software que possibilita a visualização

dos pacotes trafegados na rede) a fim de fazer com que ele reconhecesse as mensagens implementadas. O programa escolhido foi o *Ethereal* versão 0.10.12 [69].

Identificou-se nos arquivos do *Ethereal* onde era realizado o reconhecimento das mensagens de tráfego. Na pasta */epan/dissectors* dentro do diretório fonte do Ethereal encontram-se os arquivos que cuidam deste processo. Foram modificados os arquivos *packet-ip.c* e *packet-mip.c*. O primeiro é responsável por reconhecer e tratar as mensagens que utilizam o protocolo ICMP; nele foram feitas as modificações necessárias para reconhecer a inserção da *flag* A no *agent advertisement*. No segundo, está o reconhecimento das mensagens que utilizam o protocolo UDP [74,75] (*registration request* e *registration reply*); realizaram-se alterações no mesmo a fim de se obter o reconhecimento das mensagens *passive agent request, MNnodelect advertisement* e *MNnodelect acknowledgement*.

Nos arquivos citados no parágrafo anterior existem estruturas condicionais que, de acordo com o campo TYPE da mensagem, montam a estrutura de visualização (em forma de árvore) específica da mesma na tela do *Ethereal*. O código foi aprimorado para reconhecer os *TYPE 81, 82 e 83* (mensagens do RDAIPM para eleição de agentes secundário) e imprimir na tela as informações convenientes (nome das mensagens, novos campos, etc).

A seguir, serão descritos os procedimentos necessários para a implementação de cada uma das mensagens do algoritmo RDAIPM, de acordo com a Figura 5.10. Apresentar-se-ão, ao final do tópico explicativo de cada mensagem implementada, as capturas de tela referentes às mesmas (obtidas com o software *Ethereal* modificado).

RDAIPM AGENT ADVERTISEMENT

O primeiro passo na modificação do código *Dynamics Mobile IP* consistiu em adicionar a *flag* A nas mensagens *agent advertisement* enviadas pelo MA (mensagem 1 na Figura A.1). Convém relembrar que se decidiu iniciar a mudança do código pelos arquivos do *Home Agent*, devido ao fato dos mesmos apresentarem uma maior simplicidade se comparados aos arquivos presentes no diretório do *Foreign Agent*.

O objetivo da inserção dessa *flag* de controle é de tornar o código para o algoritmo RDAIPM transparente aos nós que implementam o *Mobile IP*. Se o nó móvel que

recebe a mensagem de *agent advertisement* com a *flag* A setada com o valor binário `11` não implementa o RDAIPM, então ele irá ignorar a *flag* e enviará simplesmente uma mensagem de *registration request* para o agente, caso ainda não tenha sido registrado em sua rede. Caso contrário, se o mesmo implementa o RDAIPM, ele entenderá que, ao efetuar com sucesso o seu registro na rede (após receber um *registration reply* com *code* 0), deve enviar uma mensagem *passive agent request* de modo a participar do processo de eleição do agente secundário.

No intuito de efetuar o processo descrito anteriormente de modo simples, preferiu-se não alterar a estrutura da mensagem *agent advertisement* original. A RFC-3222 define um campo de um *byte* de nome *reserved*, que não é utilizado pelo *Mobile IP* (o mesmo é enviado com o valor zero e ignorado na recepção pelo MN). Observado este fato, decidiu-se inserir nesse octeto a *flag* de controle. Em vista disto, não foi declarada uma nova estrutura no arquivo *message.h*; simplesmente aproveitou-se a estrutura original já existente para o *agent advertisement*, cuja visualização é dada abaixo.

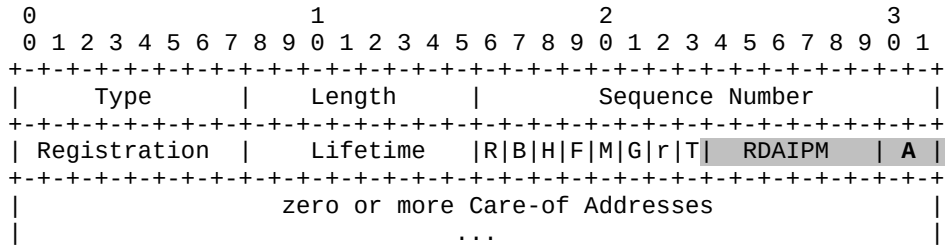


Figura A.1 - Estrutura do RDAIPM Agent Advertisement

O arquivo principal para o funcionamento do *daemon* do HA é o *ha.c*. Este arquivo possui dentro de sua função *main* um *loop* “infinito”, no qual se encontram todos os procedimentos relacionados ao tráfego de mensagens na rede (enviadas e recebidas pelo agente). Dentre esses procedimentos destaca-se a função responsável pelo envio periódico dos *agent advertisements*, cujo nome é *send_agent_advs*. No arquivo *dynhad.conf*, existe a opção de se configurar, para cada interface de rede, o envio das mensagens *agent advertisement*. O parâmetro *agentadv* define se as mensagens serão enviadas:

- regularmente, independente de solicitação por parte do MN - valor 1

- somente mediante solicitação (via mensagem *agent solicitation*) - valor 0
- nunca, mesmo sob solicitação - valor -1

Existe ainda o valor *interval* que atribui o tempo de espera para o envio periódico de *agent advertisements* (o valor *default* para o mesmo é de 10 segundos).

A função *send_agent_advs* recebe como único parâmetro um ponteiro para o tempo de envio do próximo *agent advertisement*. Essa função é chamada uma vez antes do *loop* da função *main*, de modo a iniciar o processo de envio dos *advertisements* assim que o *daemon* do HA é levantado. Já no interior do *loop*, a função *send_agent_advs* é chamada sempre que esgotado o intervalo de envio entre dois *advertisements* consecutivos. Essa função também, sempre que chamada, atribui o valor do tempo em que o próximo *advertisement* deve ser enviado.

Dentro de *send_agent_advs* é chamada a função *ha_send_agent_adv_RDAIPM*, que foi declarada de modo a substituir a função original *ha_send_agent_adv*. A *ha_send_agent_adv_RDAIPM* chama outra função denominada *set_agent_adv_data_RDAIPM*, a qual atribui os valores dos campos do *agent advertisement*, inclusive o valor do conjunto FLAGS + RESERVED, ao qual se atribuiu o valor AGENT_ADV_HOME_AGENT_RDAIPM, definido em message.h como sendo o hexadecimal 0x2003. O valor 20 significa que apenas a *flag* H do primeiro *byte* é setada (a flag identificadora do *Home Agent*). O valor 03 seta (atribui valor 1) os dois últimos bits da *flag* A. A representação da descrição é dada a seguir, Figura A.2.

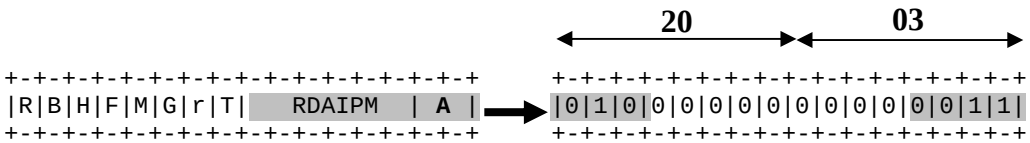


Figura A.2 - Valores dos octetos FLAGS e RESERVED.

Ainda na função *ha_send_agent_adv_RDAIPM*, após a atribuição dos valores dos campos da mensagem *agent advertisement*, é feita uma chamada à função *send_agent_advertisement_RDAIPM*, similar à original *send_agent_advertisement*. A mesma se encontra declarada dentro do arquivo *agentadv.c* no diretório */other*, o qual

é de suma importância, em se tratando do envio e recebimento efetivo das mensagens *agent advertisement*. Essa função monta a mensagem *agent advertisement* no *buffer* de saída e envia a mesma, por *Flooding*, utilizando uma conexão por *socket*. O protocolo utilizado é o ICMP [22]. A última função chamada (em nível mais baixo) é a *sendto* a qual envia de fato a mensagem para o endereço e porta de destino especificados como parâmetros de entrada [70].

Uma vez enviadas pelo agente móvel, as mensagens *agent advertisement* serão recebidas pelos nós móveis presentes na rede. No diretório do MN, existe um arquivo de nome *mn_agentadv.c*, que representa o módulo de tratamento dos *agent advertisements* recebidos pelo mesmo, seja de um HA ou de um FA. A principal função dentro deste arquivo é a *handle_icmp*, a qual lida com o recebimento das mensagens que utilizam o protocolo ICMP (o que é o caso dos *agent advertisements*). Ela verifica se a *flag* H ou a *flag* F estão setadas, para determinar se o *agent advertisement* provém de um *Home Agent* ou de um *Foreign Agent* (dentro da mesma são chamadas as funções *handle_home_adv* e *handle_fa_adv*, respectivamente).

No arquivo *agentadv.c* se encontra a função *handle_icmp_adv*, que é chamada antes de *handle_home_adv* ou *handle_fa_adv*, dentro da função *handle_icmp* citada no parágrafo anterior. Essa função é responsável pelo recebimento efetivo das mensagens *agent advertisement* via função *recvfrom*. A função *recvfrom* recebe um conjunto de bits que é tratado e particionado dentro da função *handle_icmp_adv*. Essa última atualiza um ponteiro para uma estrutura *adv_extensions*, na qual é armazenada a mensagem propriamente dita. Cabe comentar que no arquivo em questão também existem as funções que tratam do envio e recebimento de mensagens *agent solicitation*.

A Figura A.3 ilustra a mensagem *agent advertisement* modificada, onde se nota sublinhada a *flag* A sublinhada, cujo valor é `11`, de acordo com a Tabela A.1. O arquivo principal para o funcionamento do *daemon* do MN é o *mn.c*. Assim como no *ha.c*, este arquivo possui dentro de sua função *main* um *loop* “infinito”, no qual se encontram os procedimentos relacionados ao tráfego de mensagens na rede. A função *handle_icmp* aparece neste *loop* dentro de uma estrutura de repetição que “varre”

todas as interfaces de rede disponíveis para verificar a chegada de uma nova mensagem ICMP.

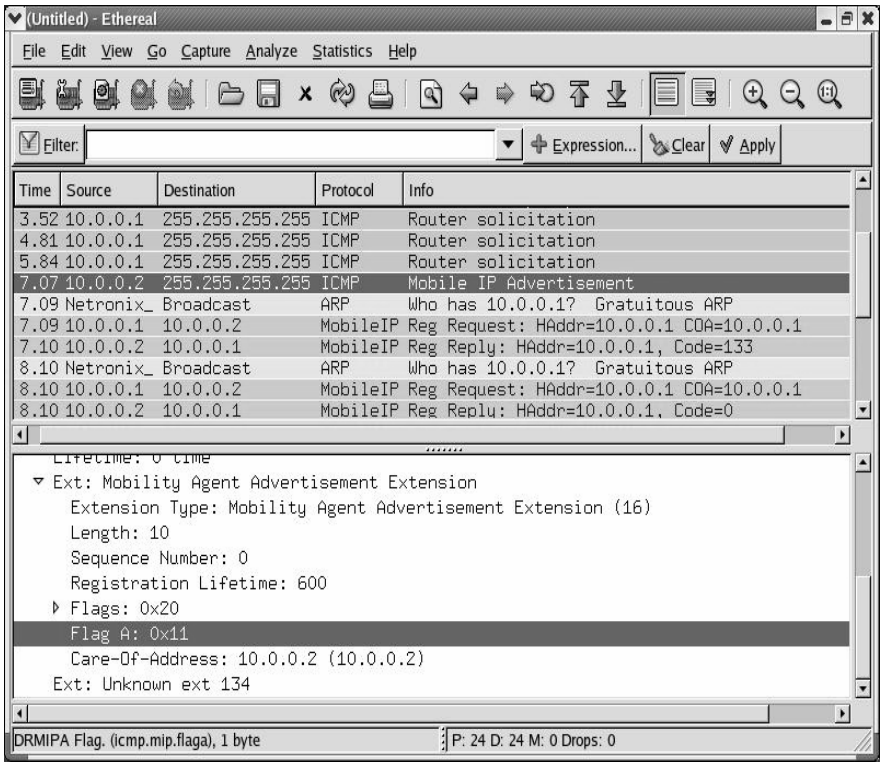


Figura A.3 - Visualização do RDAIPM Agent Advertisement no Ethereal

PASSIVE AGENT REQUEST

O próximo passo consistiu em criar a segunda mensagem da Figura 5.10, denominada *passive agent request*, através da qual o nó móvel envia ao MA uma solicitação que o torna candidato a agente secundário na rede, Figura A.4. É importante salientar que essa mensagem só é enviada caso tenha sido detectada no *agent advertisement* a *flag* A setada com o valor 11.

Inseriu-se no arquivo *message.h* a declaração da estrutura da mensagem *passive agent request*. A mesma foi baseada na estrutura da mensagem *registration request*, sendo que se retirou dessa os campos de 32bits destinados ao *care-of address* e ao *identification*. Como as *flags* próprias da mensagem *registration request* não são necessárias na nova mensagem *passive agent request*, utilizou-se o octeto que contém as mesmas para se inserir a *flag* A, do mesmo modo ocorrido na modificação do *agent advertisement* (onde o octeto correspondente ao *reserved* foi substituído pela *flag*).

Como visto anteriormente, o valor dessa *flag* no *passive agent request* é 00. A estrutura da mensagem pode ser vista abaixo.

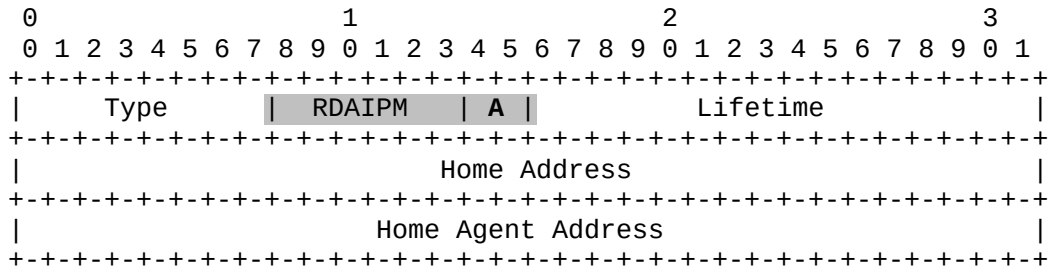


Figura A.4 - Estrutura do Passive Agent Request.

Dentro do diretório do MN se encontra o arquivo *mn_reg.c*, que é responsável pelo processo de registro do nó com os agentes móveis. Neste arquivo existem as funções para envio da mensagem *registration request* e tratamento da chegada de um *registration reply* oriundo do MA em questão.

A priori, a mensagem *passive agent request* estava sendo enviada pelo MN assim que este recebia um *agent advertisement* com a *flag* A setada em `11`, simultaneamente com o envio da mensagem *registration request*. Entretanto, depois de um certo tempo, percebeu-se que tal procedimento não era logicamente correto, uma vez que o conveniente é que o MN envie uma mensagem para se candidatar a agente passivo somente após ter efetuado com sucesso o seu registro na rede. Em vista dessa observação, modificamos o código novamente de modo que o *passive agent request* fosse enviado somente depois de verificado o recebimento de um *registration reply* com *code* 0 (zero). Dentro do arquivo mn_reg.c existe uma função denominada *handle_registration* que é responsável por tratar a chegada de mensagens do tipo *registration reply*. Se o campo *code* da mensagem for igual a 0 (zero), é chamada a função *handle_reg_accept* (declarada no mesmo arquivo), dentro da qual se inseriu a chamada à função *send_passive_agent_request*.

Inseriu-se no arquivo *mn_reg.c* a nova função *fill_pass_req_header* (baseada na original *fill_req_header*), que é responsável por preencher os valores dos campos da mensagem *passive agent request*. Adicionou-se também a função *send_passive_agent_request* (baseada na original *send_registration*), a qual lida com o envio efetivo da mensagem para o *socket* e endereço de destino apropriados,

fazendo mais uma vez uso da função *sendto* para esta finalidade. A porta de destino é definida como aquela presente no arquivo *dynmnd.conf*, sob o parâmetro *UDPPort*, cujo valor padrão atribuído pela *RFC-3222* é 434.

Nesse ponto, convém ressaltar que foi declarado em *mn.h* (arquivo de biblioteca para o *mn.c*) a variável inteira *passive_election_socket*, que identifica um *socket* exclusivo para o fluxo de mensagens dos tipos *passive agent request* e *MNnodelect* (similarmente à existência prévia da variável *registration_socket*, pertinente às mensagens *registration request* e *reply*). A variável *passive_election_socket* foi propriamente declarada dentro da estrutura *mn_data* do arquivo *mn.h*, a qual permite armazenar todos os parâmetros necessários para o correto funcionamento do *daemon* do MN. Inicialmente, estava sendo utilizado o mesmo *socket* das mensagens de registro para as mensagens de eleição do agente secundário e isto acarretou em problemas no envio das mesmas, devido a conflitos inesperados. Em vista disso, o procedimento detalhado anteriormente foi tomado como medida corretiva.

Dentro da função *send_passive_agent_request*, no arquivo *mn_reg.c*, é chamada uma função especial denominada *parse_msg*, declarada no arquivo *msgparser.c*, do diretório */other*. Essa trata do particionamento do conjunto de bits que compõe as mensagens do tipo *registration request* e *registration reply*, cuidando automaticamente do reconhecimento da mensagem através do campo *TYPE* (primeiro octeto do conjunto de bits). A mesma salva a mensagem já particionada em um ponteiro do tipo *msg_extensions*, que é uma estrutura definida na biblioteca *msgparser.h*. De modo a implementar o código do RDAIPM, aprimorou-se a função *parse_msg* para que reconhecesse também as mensagens do tipo *passive agent request* (mais à frente veremos que também foi incluído o *TYPE* das mensagens *MNnodelect*). Para que isso se tornasse possível, foi preciso adicionar à *msg_extensions* as estruturas das mensagens RDAIPM declaradas em *message.h*. Nesta última biblioteca, foram definidos novos valores para o campo *TYPE* para cada uma das mensagens implementadas para o RDAIPM (o valor atribuído para o *passive agent request* foi o decimal 81).

Em se tratando do recebimento da mensagem *passive agent request* por parte do HA, destaca-se uma função principal denominada *handle_reg_msg* chamada na *main* do

ha.c, dentro do *loop* infinito. Originalmente, essa função era usada apenas para processar mensagens do tipo *registration request*, sendo que foram feitas as devidas alterações para que ela se tornasse apta a tratar o recebimento das mensagens do tipo *passive agent request*.

Dentro da função *handle_reg_msg* acontece o processo de recebimento do conjunto de bits do *socket* aonde chegou a mensagem através da função *recvmsg*, que é semelhante à função *recvfrom*. Após o recebimento dos bits, é usada a função *parse_msg* descrita acima para identificar e particionar a mensagem, armazenando-a em um ponteiro específico. Inseriu-se um módulo de controle para tratar a mensagem recebida caso ela seja um *passive agent request*. Foi ainda criado um *array* para armazenamento dos dados de endereço IP e porta dos nós móveis que se candidatam a agente secundário. Esse *array* corresponde à tabela de cadastro de nós RDAIPM. A idéia original era escolher o agente passivo examinando-se a rota AODV de menor número de seqüência DSN e menor número de saltos até o nó em questão.

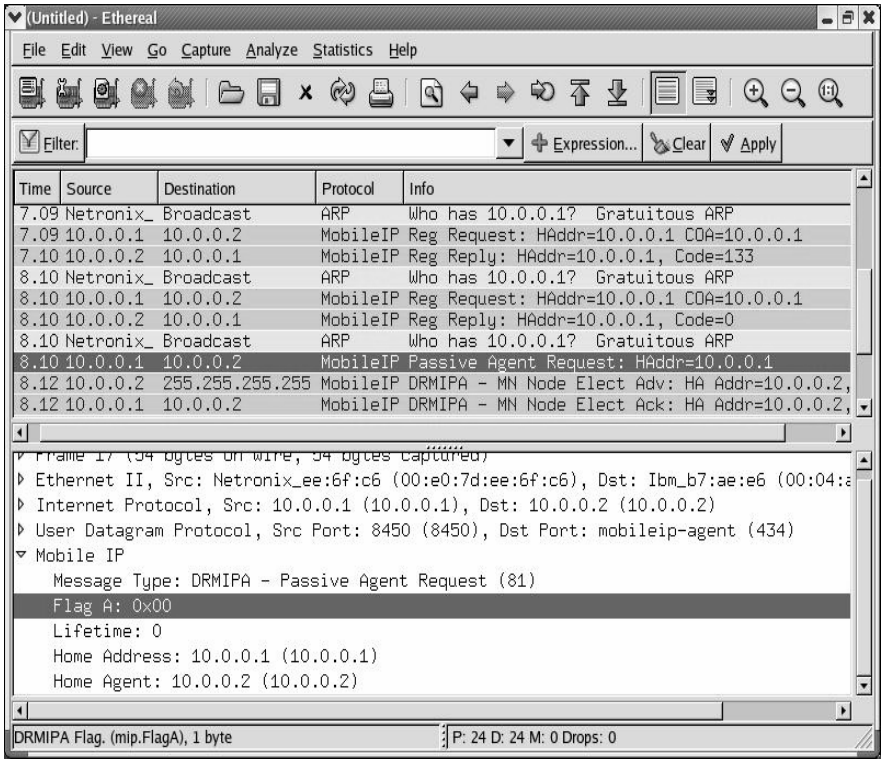


Figura A.5 - Visualização do Passive Agent Request no Ethereal.

Entretanto, a efeito de se obter uma simplicidade inicial na implementação, decidiu-se utilizar a regra substitutiva comentada anteriormente (a primeira mensagem *passive*

agent request que chega ao HA é utilizada para eleger o agente secundário). Em vista disso foi declarada uma variável global de controle dentro de ha.c para verificar se o agente passivo já foi escolhido. Abaixo é dada a visualização da mensagem *passive agent request* no software *Ethereal* propriamente adaptado, Figura A.5 anterior.

Ao receber com sucesso as mensagens do tipo *passive agent request*, o HA se encarrega de eleger o agente secundário e de enviar (via *Flooding*) uma mensagem *MNnodelect advertisement*, como será descrito no próximo tópico.

MNNODELECT ADVERTISEMENT

Uma vez terminada a implementação da mensagem *passive agent request*, prosseguiu-se para a próxima mensagem da Figura 5.10, A.7: *MNnodelect advertisement*. Essa mensagem tem o intuito de informar todos os nós da rede a respeito do nó escolhido pelo agente primário para ser o agente secundário; assim sendo, ela é enviada por *Flooding*, fazendo com que os MNs cessem de enviar mensagens do tipo *passive agent request*. Os nós, ao receberem o *MNnodelect advertisement*, verificam o endereço IP do nó escolhido anunciado na mensagem a fim de comparar com o próprio endereço IP. Caso o nó tenha sido eleito, terá de enviar uma mensagem do tipo *MNnodelect acknowledgement*, como será mais bem explicado à frente.

Do mesmo modo feito para a mensagem *passive agent request*, foi declarada uma estrutura para a mensagem *MNnodelect advertisement* no arquivo *message.h*, assim como definido o valor a ser assumido para o seu campo *TYPE* (decimal 82). Como se pode ver na Figura A.6 abaixo, a mensagem possui o mesmo formato do *passive agent request*, sendo que agora são armazenados os endereços IP do agente móvel ativo e do nó eleito para ser agente passivo.

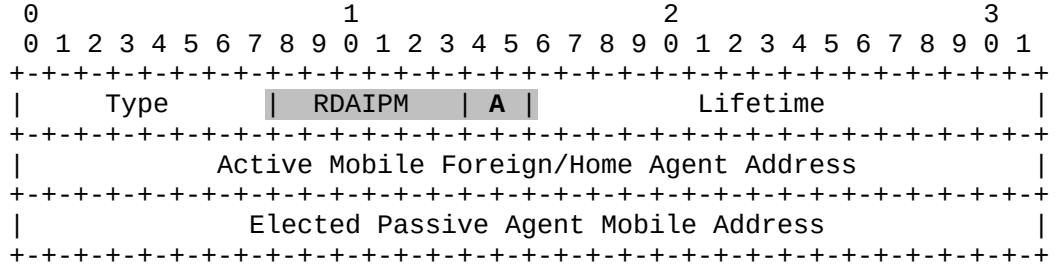


Figura A.6 - Estrutura do MNnodelect Advertisement.

Note que a *flag* A assume o valor 01, o que representa uma mensagem do agente ativo para o agente passivo, como estabelecido na Tabela A.1. Foi criada uma função para envio da mensagem *MNnodelect advertisement* dentro do arquivo *ha.c*, cujo nome é *send_mn_nodelect_adv*. Essa função foi espelhada no procedimento de envio da mensagem *registration reply* pelo agente. Ela é chamada uma única vez dentro da função *handle_reg_msg* (presente no mesmo arquivo) após a chegada das mensagens *passive registration request* enviadas pelos nós da rede e havendo sido realizada a eleição do agente secundário.

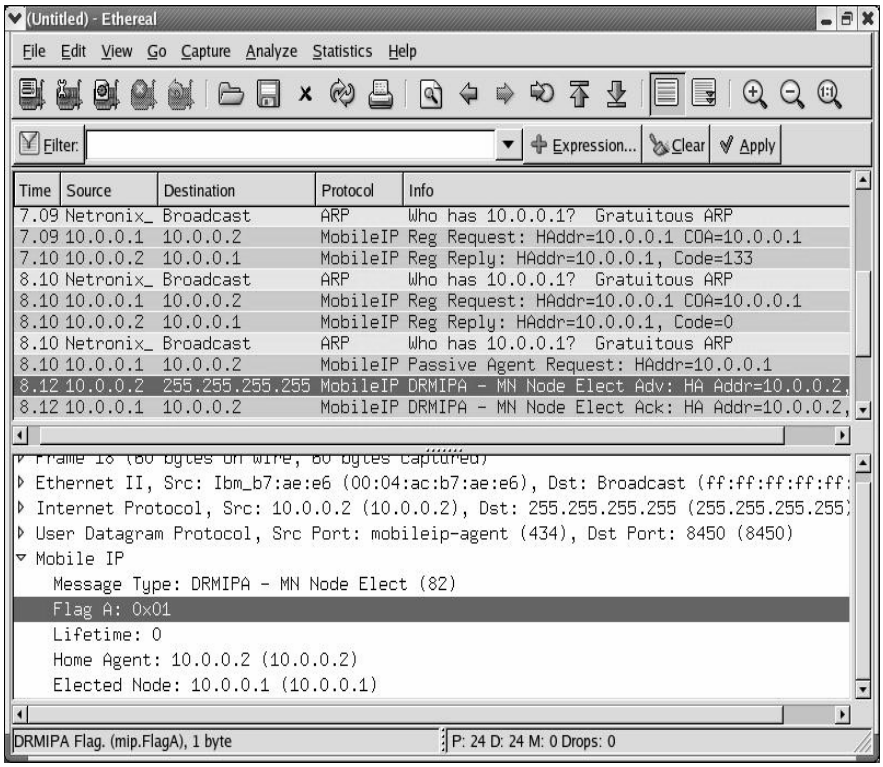


Figura A.7 - Visualização do MNnodelect Advertisement no Ethereal.

Dentro de *send_mn_nodelect_adv*, são atribuídos os valores para os campos da estrutura da mensagem *MNnodelect advertisement*. Como as mensagens são enviadas através de uma conexão por *socket*, foi necessária a utilização da função *setsockopt* com o parâmetro *optname* setado como SO_FLOODING de modo a habilitar o *Flooding*. Mais uma vez foi utilizada a função *sendto* para enviar a mensagem pelo *socket* específico *passive_election_socket*.

Foram modificados os arquivos *msgparser.h* e *msgparser.c* do diretório */other*. No primeiro, incluiu-se a estrutura da mensagem *MNnodelect advertisement* em

msg_extensions; no último, foi adicionado o reconhecimento da nova mensagem dentro da função *parse_msg*, através da identificação pelo valor do campo *TYPE*.
Em *mn_util.c* , um arquivo de suporte para o funcionamento do MN, foi criada uma nova função *create_mnodelect_socket* (a espelho da *create_registration_socket*) para criar o *socket* exclusivo ao envio e recebimento das mensagens de eleição de agente secundário já mencionado anteriormente (o *passive_election_socket*). Esse *socket* foi associado com um valor definido para o número da porta (escolheu-se o valor 8450), que pode ser observado na visualização das mensagens nas Figuras A.3, A.5 e A.7. A função *create_mnodelect_socket* é chamada dentro de *mn_init*, uma rotina presente no mesmo arquivo que se encarrega de inicializar os *sockets* e variáveis necessários ao funcionamento do MN.

Sabe-se que no *loop* principal da função *main* de mn.c verifica-se a chegada de novas mensagens da rede. O recebimento de uma nova mensagem *registration reply* é constatado através da ativação de um *file descriptor* associado ao *socket* em questão. Procedeu-se da mesma maneira para identificar a chegada de uma nova mensagem *MNnodelect advertisement*, ao se criar um novo *file descriptor* associado ao *passive_election_socket*. De fato, ao se detectar a ativação do mesmo, é chamada uma função para reconhecimento da mensagem e posterior envio do *Mnnodelect acknowledgement*, procedimento que será mais detalhado no tópico a seguir. A mensagem *MNnodelect advertisement* pode ser vista na figura abaixo.

MNNODELECT ACKNOWLEDGEMENT

A última mensagem implementada até o momento foi a *MNnodelect acknowledgement*, representada pela mensagem número 4 da figura 3.6. Assim que os nós móveis recebem o *MNnodelect advertisement*, como dito anteriormente, verificam o endereço IP do nó eleito a fim de saber se são eles próprios o agente secundário escolhido. Em caso positivo, deve ser enviada uma mensagem do tipo *MNnodelect acknowledgement* para o HA com o objetivo de confirmar sua eleição.

Uma vez confirmado como agente secundário na rede, o MN deve levantar o processo do HA com a finalidade de exercer efetivamente a sua função. Daí em diante, o agente primário irá repassar via *unicast* a lista de *bindings* para o agente secundário e,

à medida que acontecerem novos registros, enviar uma mensagem com o novo *binding* para que a lista possa ser atualizada. Essas mensagens devem ser confirmadas pelo agente passivo através de mensagens do tipo *Binding acknowledgement*. Os procedimentos relacionados à operação do agente secundário ainda não foram implementados.

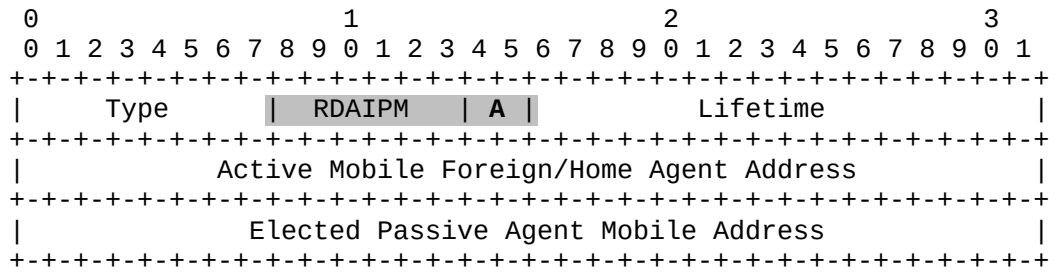


Figura A.8 - Estrutura do MNnodelect Acknowledgement.

De maneira similar às mensagens anteriores, foi declarada a estrutura da mensagem no arquivo *message.h*. O *TYPE* definido foi o decimal 83. O valor agora assumido pela *flag* A é 10, representando uma mensagem enviada de um agente passivo para um agente ativo, como visto anteriormente. Na Figura A.9 está representada a estrutura do *MNnodelect acknowledgement*, idêntica à do *advertisement*, exceto pelo valor da *flag*.

Assim como feito para as mensagens *passive agent request* e *MNnodelect advertsiment*, modificou-se mais uma vez os arquivos *msgparser.h* e *msgparser.c* do diretório */other*. No primeiro, incluiu-se a estrutura da mensagem *MNnodelect acknowledgement* em *msg_extensions*; no último, foi adicionado o reconhecimento da nova mensagem dentro da função *parse_msg*, através da identificação pelo valor do campo *TYPE*.

No arquivo mn_reg.c do diretório */mn* criou-se a função *handle_mnodelect*, que possui a finalidade de receber um *MNnodelect advertisement*, verificar se o endereço IP divulgado como eleito é o mesmo endereço local do MN em questão e, caso positivo, montar e enviar a mensagem *MNnodelect acknowledgement*. O recebimento e envio das mensagens em questão se dá com as funções *recvfrom* e *sendto*, já citadas anteriormente. Mais uma vez foi utilizada a função *parse_msg* para reconhecer e salvar a mensagem recebida pelo *socket* (no caso, o *MNnodelect advertisement*).

A função *handle_mnodelect* é chamada dentro do *loop* principal da função *main* do arquivo *mn.c*, caso o *file descriptor* associado ao *passive_election_socket* seja ativado (o que significa que chegou uma nova mensagem *MNnodelect advertisement*). A visualização da mensagem *MNnodelect acknowledgement* está presente na Figura A.9.

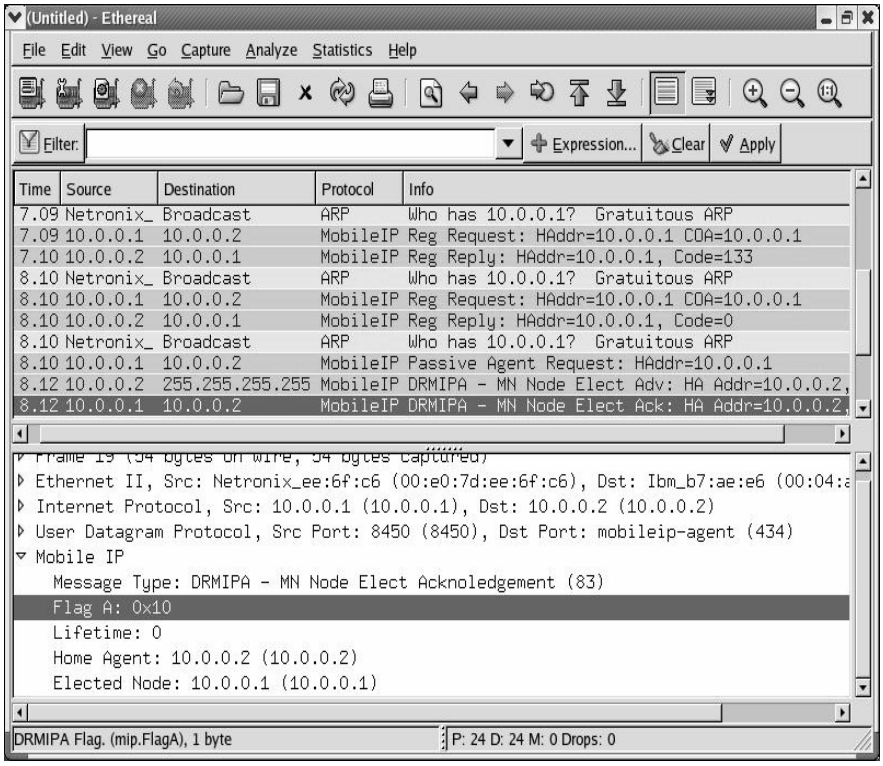


Figura A.9 - Visualização do MNnodelect Acknowledgement no Ethereal

B - PROPOSTA DRAFT-IETF-DRMIPA-00

Internet Draft G. Amvame
UnB
C. Barenco
USB
Expiration Date: December 2006 August 2006

Dynamic Reconfiguration of Mobile IP Agents (DRMIPA) for MANET
draft-georges-DRMIPA-00

Status of this Memo

Copyright Notice

Abstract

The Dynamic Reconfiguration of Mobile IP Agents (DRMIPA) protocol is intended for use by Mobile IPv4 nodes in wireless multihop networks such as MANET. It is an extension of the Mobile IPv4 protocol and offers session continuity for distinct MANET networks using a Dynamic Reconfiguration of its primary MIPv4 Home and Foreign Agents. The algorithm elects Passive agents from active agents to provide a fault tolerance network environment.

Table of Contents

1. Overview 3

2. Terminology 4

3. Data Structures 6
3.1. DRMIPA Message Formats 6
3.1.1. Packet and Message Structure 6
3.1.2. Agent Advertisement 6
3.1.3. Passive Agent Request 6
3.1.4. Mobile Node Elect Advertisement 7
3.1.5. Mobile Node Elect Acknowledgment 9
3.1.6. Mobile Node Binding Update between Active and Passive Agents 10

4. Detailed Operation 11
4.1. DRMIPA Passive Agent Election Operations 11
4.1.1. Maintaining a Passive Node Election Table 11
4.1.2. Maintaining a Passive Node Request State 11
4.1.3. Maintaining a Passive Node Election State 11
4.2. DRMIPA Active Agent Election Operations 12
4.2.1. Maintaining an Active Node State 12
4.2.2. Maintaining a Passive to Active Node State 12
4.3. DRMIPA Mobile Nodes Flag Operations 12
4.3.1. Unicast Message from Mobile Node to Active Agent . . . 13
4.3.2. Flooding Message from Active Agent to Passive Agent 13
4.3.3. Unicast Message from Passive Agent to Active Agent 14
4.3.4. Flooding Message from Active Agent to Mobile Node 14
4.4. Internet Attachment and Gateway Support 14
4.5. Mobile MHA and MFA Multiple Interfaces. 14
4.6. Support for Routing Protocol. 14

5. Configuration Parameters 15

6. IANA Considerations 15

7. Security Considerations 16

8. Acknowledgments 16

9. References 16
9.1. Normative References 16
9.2. Informative References 16

Authors' Addresses 17

1. Overview

The Dynamic Reconfiguration of Mobile IP Agents (DRMIPA) primary goal is to support IP mobility in different multihop environments. DRMIPA is an extension of the Mobile IPv4 protocol. All mobility agents are now integrated into MANET and are no more needed in an infrastructure environment. The MIPv4 agents are now called Mobile Home Agent (MHA) and Mobile Foreign Agent (MFA).The basic MIPv4 mechanisms are kept as in [1] so that data session continuity are maintained.

A DRMIPA node SHOULD operate as a MIPv4 node. DRMIPA ensures session continuity fore all nodes in MANET even if the Mobile Agent shuts down or move to another network. The protocol proposes a bidirectional tunnel between Mobile nodes and their MHA, using the a reactive or proactive routing protocol (Tests have been done using the AODV Routing protocol as routes are made active upon Mobile Node request). So, all traffic between DRMIPA mobile nodes passes thru the MHA.

The basic operations of DRMIPA are agent election and fault tolerance management. The DRMIPA Mobile Home and Foreign agents when turned on SHOULD listen to any agent solicitation coming from any MIPv4 mobile nodes with or without a DRMIPA protocol.

The new protocol floods a modified agent advertisement to mobile nodes in the network. It then receives a passive agent request from nodes that WOULD implement DRMIPA. Only nodes that have this protocol SHOULD respond to this agent advertisement sending a unicast message to the active agent.

ALL new control messages SHOULD be transparent for actual MIPv4 mobile nodes that do not implement DRMIPA. The MANET routing protocol to be used SHOULD permit the exchange of all DRMIPA messages between nodes no matter their locations throughout the network and, find the destination target node.

A tunnel has to be created between ACTIVE Home and Foreign agents as in an infrastructure network [2]. Therefore, the routing protocol to be used SHOULD allow the creation of this tunnel. Each node SHOULD listen to their ACTIVE and PASSIVE agents, in order to decide if they SHOULD switch their internal node STATE to: LISTEN, PASSIVE or ACTIVE. If the ACTIVE and PASSIVE agents cease their activity in the network,an internal IP sequence number enable the correct node to change its actual state to ACTIVE and this node SHOULD begin the passive node election procedure.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. In [6].

Mobile Home Agent (MHA)

A mobile router and host on a Mobile Adhoc NETwork (MANET) mobile node?s home network which elects another mobile node from his network to be a Passive Agent. This election would guaranty the availability of a MHA in case the active one ceases its own functionalities or moves to another network. This node implements all basic MIPv4 routines as specified in [1] and [4]. It is the gateway for any MIPv4/DRMIPA mobile node that wishes to communicate with a node in another network.

Mobile Foreign Agent (MFA)

A mobile router and host on a MANET mobile nodes home network which elects another mobile node from his network to be a Passive Agent. This election would guaranty the availability of a MFA in case the active one ceases its own functionalities or moves to another network. This node implements all basic MIPv4 routines as specified in [1]. It is the gateway for any MIPv4/DRMIPA mobile node that wishes to communicate with a node in another network.

Mobile Node (MN)

A mobile host and router that moves freely inside a MANET network or subnetwork to another. The MN should change its IP Address while changing networks. This mobile node must be ready to take MHA or MFA functionalities if necessary to help the network in being fault tolerant. This node must implement all basic MIPv4 routines as specified in [1].

Agent Advertisement

An advertisement message constructed as specified in [1], but with the addition of a flag A. This message has to be flooded in MANET, forwarded from hop to hop to the most reachable node.

Authentication

A process used to verify the identity of a mobile node host or router in the network. The cryptographic techniques follow the ones stated in [1].

Target

The Target is the last destination seen in the IPv4 Destination Address of any message.

Active Agent

A mobile DRMIPA router that assumes the MHA or MFA functionalities in the MANET environment.

Passive Agent

A mobile DRMIPA host that is prepared to behave as a MHA or MFA in the MANET environment.

ACTIVE

The state of a working mobile DRMIPA router. It means that the router is a MHA (Flag H set) or MFA (Flag F set).

PASSIVE

The state of a working mobile DRMIPA host. It means that the host can behave as a DRMIPA router at any moment. The mobile host is stated as PASSIVE for his MHA (Flag H set) or MFA (Flag F set).

Prefix

Indicates a correspondence between the actual IP network address and host or router one. This Prefix check is important for the MHA or MFA when Passive Agents are chosen.

3. Data Structure

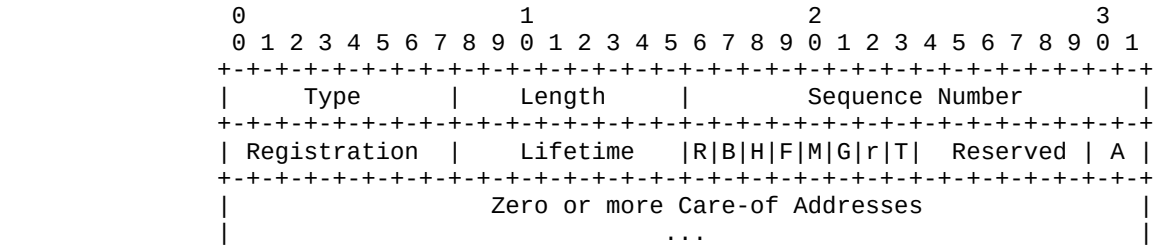
3.1. DRMIPA Message Formats

3.1.1. Packet and Message Structure

All DRMIPA messages conform to the MIPv4 packet and message format as described in [1], except for the Type field numbering which SHOULD comply with IANA.

3.1.2. Agent Advertisement

As the agent advertisement is disseminated in the network and all DRMIPA nodes SHOULD understand the current message, a new flag (A) is included in the former reserved field. Only SHOULD the ACTIVE Home or Foreign agent floods the agent advertisement with the flag (A) set.



DRMIPA Flag (A)

The DRMIPA flag is set to indicate, to all DRMIPA MIPv4 mobile node, that the advertised message is from an ACTIVE MHA (flag H set) or MFA (flag F set). This message is a 2 bits field included in the actual reserved field [1].

3.1.3. Passive Agent Request

If the DRMIPA mobile node receives an agent advertisement message with flag A set, it SHOULD send a passive agent request message to notify the MHA/MFA his willingness to be a future Passive MHA/MFA respectively.

IP field

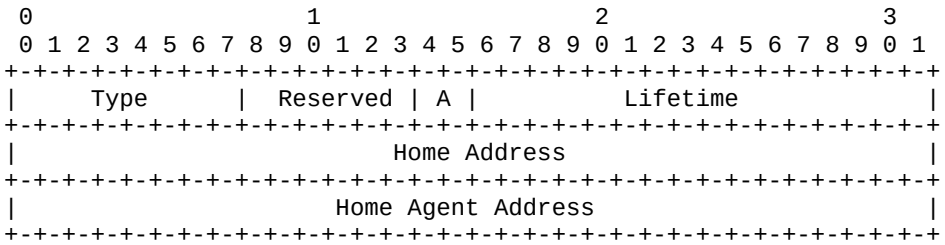
Source Address: Typically the interface address from which the message is sent. This is the MN 32 bits IP address.

Destination Address: Typically that of the MHA or MFA. This is the MHA or MFA 32 bits IP address copied from the 32 bits IP Source Address of the Agent Advertisement message.

UDP field

Source Port: <variable>

Destination Port: 434



Type 81 (Pass_Agent_Req).

Reserved

Sent as zero and ignored on reception.

DRMIPA Flag (A)

The DRMIPA flag is set to indicate, to the MHA/MFA that the mobile node WOULD participate of the PASSIVE agent election procedure. This message is sent from a DRMIPA node to an ACTIVE MHA or MFA. An internal random check SHOULD decide if the sends the message to the MHA or MFA. This message is a 2 bits field.

Lifetime

The number of seconds remaining before the agent request is considered expired.

Home Address

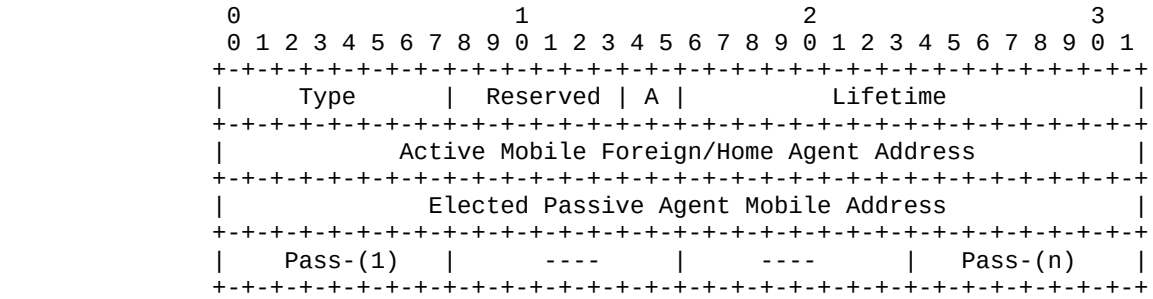
The IP address of the mobile node.

Home Agent Address

The IP address of the mobile node?s MHA or MFA.

3.1.4. Mobile Node Elect Advertisement

The MHA/MFA has to floods a Mobile Node Elect Advertisement throughout the network for DRMIPA nodes to be aware of the elected PASSIVE agent IP address. This message SHOULD make other DRMIPA nodes which IP address is not in the Elected Passive Agent Mobile Address field, to cease sending a Passive Agent Request message.



Type 82 (MNnodelect_Adv).

Reserved

Sent as zero and ignored on reception.

DRMIPA Flag (A)

The DRMIPA flag is set to indicate all DRMIPA nodes that the mobile node WOULD participate of the PASSIVE agent election procedure. This message is sent from a DRMIPA node to an ACTIVE MHA or MFA. An internal random check SHOULD decide if the sends the message to the MHA or MFA. This flag is a 2 bits field.

Lifetime

The number of seconds remaining before the election advertisement is considered expired.

Active Mobile Foreign/Home Agent Address

The IP address of the mobile nodes MHA or MFA.

Elected Passive Agent Mobile Address

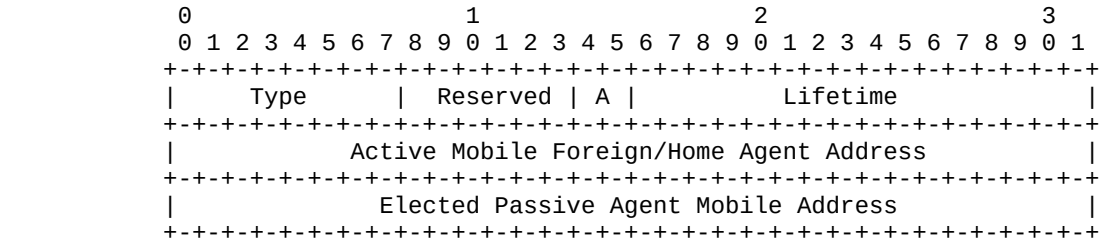
The IP address of the elected mobile node.

Pass-(n)

The IP Host Prefix address of the n-th elected mobile node.

3.1.5. Mobile Node Elect Acknowledgment

The Elected mobile node SHOULD unicast a Mobile Node Elect Acknowledgement back to its MHA or MFA. The Mobile Agent destination address should be the one specified at the Active Mobile Foreign/Home Agent Address field. This message SHOULD be sent ONLY to the appropriate ACTIVE MHA or MFA.



Type 83 (MNnodelect_Ack).

Reserved

Sent as zero and ignored on reception.

DRMIPA Flag (A)

The DRMIPA flag is set to indicate the MHA/MFA that the mobile node did ACCEPT the passive election mechanism. The PASSIVE state SHOULD be set as to prevent the node from being an ACTIVE agent without notice agent election procedure. This flag is a 2 bits field.

Lifetime

The number of seconds remaining before the election advertisement is considered expired.

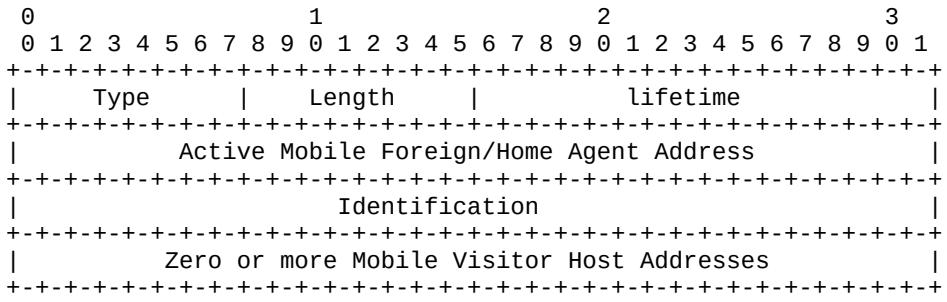
Active Mobile Foreign/Home Agent Address

The IP address of the mobile nodes MHA or MFA.

Elected Passive Agent Mobile Address

The IP address of the elected mobile node.

3.1.6. Mobile Node Binding Update between Active and Passive Agents



Length

(10 + 4*V), where 10 accounts for the number o bytes in the lifetime, Active Mobile Foreign/Home Agent Address and identification fields. And 4 the number of bytes representing the IP address field with V being the number of mobile visitor host addresses advertised.

Active Mobile Foreign/Home Agent Address

The MHA/MFA address provided by the actual active mobile foreign or home agent. This new binding update is sent by the current active MHA/MFA to the newly elected passive agent such that it would have the current mobile visitor list before take over.

Zero or more Mobile Visitor Host Addresses

The IP Address(es) of the mobile(es) visitor host(s) provided by this active MFA. The number of mobile visitor host addresses present is determined by the Length field in the Extension.

4. Detailed Operation

4.1. DRMIPA Passive Agent Election Operations

4.1.1. Maintaining a Passive Node Election Table

ACTIVE DRMIPA Mobile Agents Nodes (MHA and MFA) SHOULD maintain an array of future PASSIVE nodes. This WOULD facilitate the choice of another elected node in case actual PASSIVE agent leaves the network for any reason. Therefore, the network WOULD not be flooded unnecessary for another election. If the MHA/MFA does not receive a MNnodelect_Ack from previous MNnodelect_Adv sent after another Passive_Agent_Table_Lookup, the process SHOULD be repeated until all nodes in Passive_Agent_Table have been searched. If the current listed nodes do not respond any MNnodelect_Adv, the next agent advertisement SHOULD trigger a Pass_Agent_Req message from nodes in current network. Only nodes present and from the current network MUST be allowed for all passive agent requests. The Passive_Agent_Table array WOULD have a maximum node entry specified administratively.

4.1.2. Maintaining a Passive Node Request State

All DRMIPA mobile nodes SHOULD not set their state to PASSIVE agent unless having been already chosen by the ACTIVE agents. If a DRMIPA node receives a MHA/MFA agent advertisement message with flag A present, it SHOULD reply with a unicast Passive Agent Request message to the correct ACTIVE agent. The mobile node SHOULD have a simple mechanism to choose which ACTIVE agent it SHOULD send the Passive Agent Request message. This message SHOULD not be sent to both ACTIVE agents at anytime. The chosen ACTIVE agent SHOULD have its address in the IP Home Agent Address field of the Passive Agent Request message.

Upon receipt of a Passive Agent Request message, the ACTIVE MHA/MFA agents SHOULD record all mobile nodes request (as described in Section 4.1.1). The ACTIVE MHA/MFA SHOULD only record nodes having its own network prefix and same security identification. Security measures are not the scope of this procedure.

4.1.3. Maintaining a Passive Node Election State

All non DRMIPA elected mobile nodes SHOULD not set their state to PASSIVE agent. Only mobile nodes receiving the correct MNnodelect_Adv with their IP address corresponding to the IP Elected Passive Agent Mobile Address field, in the MNnodelect_Adv message SHOULD set their state to PASSIVE agent. The new PASSIVE agent SHOULD not change its PASSIVE state to ACTIVE unless it does not receive another MIPv4 DRMIPA_Agent_Adv message after,

```
actual.time_sec > (ha.current_adv->last.time_sec+3*agent_adv_value)
```

The ACTIVE agent SHOULD record its actual PASSIVE agent IP address and set it as ELECTED. The ACTIVE MHA/MFA SHOULD floods regularly a MNnodelect_Adv message throughout the network so that all DRMIPA nodes WOULD know that a PASSIVE agent has been elected and is still in activity in the network.

4.2. DRMIPA Active Agent Election Operations

4.2.1. Maintaining an Active Node State

All ACTIVE DRMIPA Mobile Agents Nodes (MHA and MFA) SHOULD be configured manually by an administrative authority. The MHA and MFA SHOULD have their state set as ACTIVE. As the system is set up, the DRMIPA mechanism SHOULD take place as described in section 4.1.1.

The dissemination of the DRMIPA Agent Advertisement MUST be received by all DRMIPA mobile nodes in the network. An IP address prefix and security check SHOULD be done in the DRMIPA mobile nodes as to verify if the message is coming from its own network.

If the ACTIVE MHA or MFA shutdown or leaves the network it MUST have its ACTIVE state unset. The Active MHA/MFA SHOULD be deactivated and the node WOULD be in a LISTEN state awaiting further DRMIPA Agent Advertisement in the network.

4.2.2. Maintaining a Passive to Active Node State

If the actual ACTIVE MHA/MFA ceases activity in the network, the PASSIVE mobile node SHOULD change his agent state to ACTIVE. At this point, the DRMIPA mobile node MUST have the Active MHA or MFA agent algorithm running in the system. The MN to MHA/MFA algorithm switch SHOULD be done transparently. The mobile node SHOULD maintain its IP address for data and session continuity with other node in the network. All procedures described in Section 4.1.1 SHOULD take place in this new ACTIVE agent.

4.3. DRMIPA Mobile Nodes Flag Operations

Flag A	From	To
00	MN	ACTIVE
01	ACTIVE	PASSIVE
10	PASSIVE	ACTIVE
11	ACTIVE	MN

4.3.1. Unicast Message from Mobile Node to Active Agent

A unicast message has to be built by each DRMIPA mobile node and need to have the Active Agent as the TARGET destination. This message is called Passive Agent Request and flag A SHOULD be set to 00.

The receiving Active MHA or MFA MUST process this message verifying the MIPv4 Home Address and flag A fields. After message process, the Active MHA or MFA SHOULD keeps the received Home Address in its Passive_Agent_Table array entry. The first Passive Agent in that table entry WOULD be the first IP Address used for further MNnodelect_Adv message.

It should be noticed that, only Mobile Nodes from current network not listed in the MFA visitor list entry and MHA binding list will be eligible for the passive agent election process.

4.3.2. Flooding Message from Active Agent to Passive Agent

After the election process, the Active MHA or MFA SHOULD flood a MNnodelect_Adv message to all nodes in the network. The new elected passive agent IP Address SHOULD be copied into the Elected Passive Agent Mobile Address field, located in the Mobile Node Elect Advertisement message (as described in Section 3.1.4). If non DRMIPA nodes receive this message, they MUST drop it. All other DRMIPA nodes SHOULD keep the first Elected Passive Agent IP address in cache. The MNnodelect_Adv message SHOULD have at least 4 more DRMIPA IP addresses taken from the Passive_Agent_Table entry (as described in Section 3.1.4). In case both Active and Passive Agents SHOULD cease their operations or move to another network, the next elected Passive Agent (Pass-(1)) SHOULD change its state to ACTIVE and SHOULD elect a new Passive Agent or use Pass-(2) as Passive Agent:

* Pass-(n): Node change state to ACTIVE and starts acting as an Active Agent (MHA or MFA). Where Pass-(n) is the IP Host Prefix octet field of the next passive agent (as described in Section 3.1.4), preceeding the 32 bits Elected Passive Agent Mobile Address field.

* Pass-(n+1): Node change state to PASSIVE and starts acting as a Passive Agent for Pass-(n). Where Pass-(n+1) is the IP Host Prefix octet field of the next passive agent (as described in Section 3.1.4), preceeding the Pass-(n) 8 bits field.

These addresses SHOULD be included after the 32 bits Elected Passive Agent Mobile Address field. Each mobile node WOULD then be aware that they are next to be Active or Passive Agents.

This message is called Mobile Node Elect Advertisement and flag A SHOULD to be set to 01.

4.3.3. Unicast Message from Passive Agent to Active Agent

The newly elected Passive Agent MUST build and send a Mobile Node Elect Acknowledge message to the correct MHA or MFA. This message SHOULD be unicast after previous process of the MNnodelect_Adv message. Only nodes with their states set to PASSIVE should send this message.

This message is called Mobile Node Elect Acknowledge and flag A SHOULD to be set to 10.

4.3.4. Flooding Message from Active Agent to Mobile Node

The MHA and MFA first task is to disseminate Agent Advertisements throughout the network to announce their presence. This message SHOULD be transparent for any Mobile Node implementing MIPv4 or DRMIPA. The included flag A SHOULD only be analyzed by DRMIPA nodes. In order to maintain correct integration with actual MIPv4 protocol, all MIPv4 Mobile Node SHOULD be registered normally at the MHA (when being part the Home Network) or MFA (when from seen as a Foreign Network).

This message is called Mobile Agent Advertisement and flag A SHOULD to be set to 11.

4.4. Internet Attachment and Gateway Support

In Global Connectivity for IPv4 Mobile Ad hoc Networks [5] a solution is presented where AODV cooperates with the Mobile IP protocol. MIPv4 is used for mobile node registrations with a mobility agent, while AODV is used for routing mechanisms within the mobile Ad Hoc network and obtaining routes to the Internet using a fixed gateway.

As a tunnel has to be created between ACTIVE MHA and MFA agents [2] and [3], the routing protocol to be used SHOULD allow the creation of another tunnel between the MHA/MFA and an Internet Gateway [5]. So, if MIPv4 and DRMIPA nodes need access to any Correspondent Node in the Internet, the MHA/MFA SHOULD have at least two (2) wireless interfaces as described in Section 4.5.

4.5. Mobile MHA and MFA Multiple Interfaces

It is important to notice that as a DRMIPA MN can be a future Passive to Active MHA/MFA Agent in MANET, all DRMIPA nodes SHOULD implement at least two (2) wireless interfaces. One interface SHOULD have an Ad Hoc IP address and the second an IP address for Internet access.

If both MHA and MFA are using a WAN access, one of its interfaces SHOULD have a valid public IP address for DRMIPA and MIPv4 nodes in MANET to access any internet Correspondent Node.

4.5. Support for Routing Protocol

To achieve the best performance in MANET, the protocol proposes a bidirectional tunnel between Mobile nodes and their Active Agents, using a reactive or proactive routing protocol (Tests have been done using the AODV Routing protocol as routes are made active upon Mobile Node request [7] and less flooding is introduced in the network).

This situation WOULD also allow the Active Agent to keep track of all DRMIPA nodes in its network.

All traffic between DRMIPA and MIPv4 nodes in MANET SHOULD NOT be directed to the MHA when communicating in the same network. This SHOULD avoid unnecessary traffic going to the MHA. Normal routing mechanisms SHOULD take place for data exchange between nodes as stated in [7] for example.

A tunnel has to be created between ACTIVE Home and Foreign agents as in an infrastructure network [1], with or without a fixed Gateway between them [5]. Therefore, the routing protocol to be used SHOULD allow the creation of this tunnel between MHA and MFA. If Internet access is required for any MANET implementation, procedures describe in sections 4.4 and 4.5 SHOULD be applied.

5. Configuration Parameters

The default parameter values for DRMIPA are:

- Two (2) wireless cards: one (1) on-board and one (1) PCMCIA
- Agent_adv_value 6 seconds

6. IANA Considerations

A new flag value MUST be assigned from the reserved Agent Advertisement field as described in section 3.1.2:

This document also defines new Messages for the Dynamic Reconfiguration to take place as described in section 4.

Flag-type	value
A (FADV)	11
A (FPREQ)	00
A (FMNADV)	01
A (FMNACK)	10

Msg-type value		
Pass_Agent_Req	(PREQ)	81
MNnodelect_Adv	(MNADV)	82
MNnodelect_Ack	(MNACK)	83

7. Security Considerations

Security considerations are not addressed in this document, but are generally similar to those outlined in [1] and [5].

8. Acknowledgments

The authors wish to thank CNPQ and UNB for their financial support.

9. References

9.1. Normative References

[1] C. Perkins, "IP Mobility Support for IPv4", IETF RFC 3344, August 2002.

[2] C. Perkins, "IP Encapsulation within IP," IETF RFC 2003, May 1996.

[3] C. Perkins, "Minimal Encapsulation within IP," IETF RFC 2004, May 1996.

[6] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, March 1997.

[7] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003.

9.2. Informative References

[4] C. Perkins, Mobile IP: Design Principles and Practice, Addison-Wesley Longman, Reading, Mass., 1998.

[5] Belding-Royer E.M.; Sun Y.; Perkins C., Global Connectivity For IPv4 Mobile Ad Hoc Networks, IETF Internet Draft, Nov. 2001. Work in progress.

Authors' Addresses

Georges Amvame Nze
Dept. Electrical Engineering - ENE
University of Brasilia - UNB, Campus Darcy Ribeiro
Bras?lia, Brazil CEP.70910-900
+55 61 33072708
georges@labcom.unb.br

Claudia Jacy Barenco Abbas
Departamento de Computacion y Tecnologia de la Informacion,
Universidad Simon Bolivar,
Oficina MYS 213-B, Apartado Postal 89.000
Caracas, 1080 Venezuela
barenco@ieee.org

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

C - TEMPOS DE RECONFIGURAÇÃO DO AGENTE PASSIVO PARA ATIVO

Neste apêndice são apresentados alguns tempos, em segundos, refletindo a reconfiguração de um *agente passivo* previamente eleito para atuar como *agente ativo*. Todos os experimentos foram executados durante 60 s, sendo tempo suficiente para se perceber a reconfiguração do *agente passivo*. Para ativar o mecanismo de reconfiguração, foi desativado o *agente ativo* deliberadamente para provocar uma falha no cenário contendo apenas dois nós [ver item 7.2.1]. Os dados das Figuras C.1/2/3 foram filtrados no ethereal para melhor leitura dos dados gerados pelos agentes ativo e passivo [ver apêndice A]. A título de exemplo, tem-se na figura C.1, na linha 67, em ~14,9650 s, um agente ativo (`172.1.16.5`) que faz Flooding do seu último agent advertisement (*realçado em negrito*). Na linha 68, o agente passivo (`172.1.16.42`) responde em ~14,9651s com uma mensagem *Mnnodelect_Ack* para o agente ativo. Nesse espaço de tempo, tem-se a troca de mensagens de sinalização do protocolo de roteamento AODV para garantir a manutenção da rota ativa entre os dois agentes. Em [3], define-se que um nó MIPv4 tem de esperar por pelo menos 3 *agent advertisements* consecutivos para decidir se perdeu a sinalização com o agente de mobilidade (neste caso deveria ter acontecido em ~14,9650+3 = ~17, 9650 s). Assim sendo, a reconfiguração ocorre como previsto na linha 77, com o agente (`172.1.16.42`) sendo o novo ativo na rede.

```
64 13.852884   172.1.16.42          172.1.16.5           MobileIP DRMIPA - MN Node Elect Ack: HA Addr=172.1.16.5, Elected Node=172.1.16.42
65 14.395494   172.1.16.5           255.255.255.255      AODV     Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
66 14.433107   172.1.16.42          255.255.255.255      AODV     Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
67 14.965042   172.1.16.5           255.255.255.255      ICMP     Mobile IP Advertisement
68 14.965191   172.1.16.42          172.1.16.5           MobileIP DRMIPA - MN Node Elect Ack: HA Addr=172.1.16.5, Elected Node=172.1.16.42
69 15.429446   172.1.16.5           255.255.255.255      AODV     Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
70 15.457904   172.1.16.42          255.255.255.255      AODV     Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
71 16.456022   172.1.16.5           255.255.255.255      AODV     Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
72 16.506674   172.1.16.42          255.255.255.255      AODV     Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
73 17.484375   172.1.16.5           255.255.255.255      AODV     Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
74 17.553435   172.1.16.42          255.255.255.255      AODV     Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
75 18.502794   172.1.16.5           255.255.255.255      AODV     Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
76 18.602201   172.1.16.42          255.255.255.255      AODV     Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
77 19.105479   172.1.16.42          255.255.255.255      ICMP     Mobile IP Advertisement
78 19.548809   172.1.16.5           255.255.255.255      AODV     Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
79 19.642965   172.1.16.42          255.255.255.255      AODV     Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
80 20.134042   172.1.16.42          255.255.255.255      ICMP     Mobile IP Advertisement
```

Figura C.1 – Reconfiguração dinâmica do agente passivo após desativação do agente ativo, com ***agent advertisements*** enviados a cada **1s**.

```
148 46.194030   172.1.16.5          255.255.255.255     ICMP    Mobile IP Advertisement
149 46.194214   172.1.16.42         172.1.16.5          MobileIP DRMIPA - MN Node Elect Ack: HA Addr=172.1.16.5, Elected Node=172.1.16.42
150 46.236642   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
151 46.310744   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
152 46.853538   drmipaGeorges.local localhost.local      ARP     Who has 172.1.16.5?  Tell 172.1.16.42
153 46.855418   localhost.local     drmipaGeorges.local ARP     172.1.16.5 is at 00:01:f4:96:60:08
154 47.286902   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
155 47.314527   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
156 48.319187   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
157 48.360294   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
158 49.336423   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
159 49.400063   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
160 50.360687   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
161 50.417838   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
162 51.361952   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
163 51.419638   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
164 52.404219   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
165 52.434398   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
166 53.431471   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
167 53.466170   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
168 54.478828   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
169 54.486922   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
170 54.512448   172.1.16.42         255255.255.255      ICMP    Mobile IP Advertisement
171 55.521716   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
172 55.527049   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
173 56.557486   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
174 56.575596   172.1.16.42         255.255.255.255     ICMP    Mobile IP Advertisement
```

Figura C.2 – Reconfiguração dinâmica do agente passivo após desativação do agente ativo, com ***agent advertisements*** enviados a cada **2s**.

```
 71 22.615301   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 72 22.728748   172.1.16.5          255255.255.255      ICMP    Mobile IP Advertisement
 73 22.728948   172.1.16.42         172.1.16.5          MobileIP DRMIPA - MN Node Elect Ack: HA Addr=172.1.16.5, Elected Node=172.1.16.42
 74 23.509585   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 75 23.626555   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 76 24.555357   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 77 24.654850   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 78 25.583128   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 79 25.704093   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 80 26.589915   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 81 26.719366   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 82 27.607675   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 83 27.766616   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 84 28.616453   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 85 28.768890   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 86 29.629245   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 87 29.781145   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 88 30.665016   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 89 30.818396   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 90 31.670782   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 91 31.868675   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 92 32.705566   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 93 32.871930   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 94 33.732333   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 95 33.889183   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 96 34.745108   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 97 34.906451   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
 98 35.787881   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
 99 35.945833   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
100 36.796663   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
101 36.948975   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
102 37.834432   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
103 37.981233   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
104 38.857207   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
105 38.994486   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
106 39.894973   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
107 40.004758   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
108 40.943745   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
109 41.030021   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
110 41.987518   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
111 42.050256   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
112 42.994289   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
113 43.091537   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
114 44.016059   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
115 44.130799   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
116 44.999405   172.1.16.42         255.255.255.255     ICMP    Mobile IP Advertisement
117 45.051862   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
118 45.182077   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
119 46.099611   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
120 46.216332   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
121 47.144382   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
122 47.226605   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
123 48.146161   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
124 48.271870   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
125 49.149932   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
126 49.286100   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
127 50.169772   172.1.16.42         255.255.255.255     AODV    Route Reply, D: 172.1.16.42, O: 172.1.16.42 Hcnt=0 DSN=3 Lifetime=2000
128 50.327405   172.1.16.5          255.255.255.255     AODV    Route Reply, D: 172.1.16.5, O: 172.1.16.5 Hcnt=0 DSN=3 Lifetime=2000
129 51.082641   172.1.16.42         255.255.255.255     ICMP    Mobile IP Advertisement
```

Figura C.3 – Reconfiguração dinâmica do agente passivo após desativação do agente ativo, usando ***agent advertisements*** enviados a cada **6s**.

D - CONTRIBUIÇÕES DA TESE PARA NS2

Neste apêndice são apresentadas algumas contribuições da tese para o NS2, implementadas na versão 2.27. Foram utilizados os arquivos do MIP e AODV da distribuição na versão 2.27. O arquivo *cmu-trace.cc*, até o presente momento, não tem o suporte para rastreio das mensagens MIP na versão 4, então fez-se uma modificação do mesmo, para filtrar as mensagens do MIPv4 e RDAIPM (*DRMIPA*). As mensagens em negrito representam as modificações agregadas ao arquivo *cmu-trace.cc*.

```
/* Georges Amvame Nze <georges@labcom.unb.br> 13/07/2005
 *Initial modification for UDP packets encapsulating MIP and RDAIPM
 */
void
CMUTrace::format_msg(Packet *p, int offset)
{
      struct hdr_mip *miph = HDR_RDAIPM(p);

        switch(miph->type_) {

      case MIPT_REG_REQUEST:

            if (pt_->tagged()) {
                sprintf(pt_->buffer() + offset,
                      "-MIP_REG (%d) ",
                      miph->haddr_);
            } else if (newtrace_) {
                  sprintf(pt_->buffer() + offset,
                      "-MIP_REG (%d) ",
                      miph->haddr_);
            } else {

                sprintf(pt_->buffer() + offset,
                      "-MIP_REG (%d) ",
                      miph->haddr_);
            }
      break;

      case MIPT_REG_REPLY:

            if (pt_->tagged()) {
                sprintf(pt_->buffer() + offset,
                      "-MIP_REPLY (%d) ",
                      miph->haddr_);
            } else if (newtrace_) {
                  sprintf(pt_->buffer() + offset,
                      "-MIP_REPLY (%d) ",
                      miph->haddr_);
            } else {

                sprintf(pt_->buffer() + offset,
                  "-MIP_REPLY (%d) ",
                      miph->haddr_);
            }
      break;
```

```
case MIPT_ADS:

      if (pt_->tagged()) {
          sprintf(pt_->buffer() + offset,
                "-RDAIPM_ADS (%d) ",
                miph->A);
      } else if (newtrace_) {
            sprintf(pt_->buffer() + offset,
                "-RDAIPM_ADS (%d) ",
                miph->A);
      } else {

          sprintf(pt_->buffer() + offset,
            "-RDAIPM_ADS (%d) ",
                miph->A);
      }
break;

case RDAIPM_PASS_REQ:

      if (pt_->tagged()) {
          sprintf(pt_->buffer() + offset,
                "-RDAIPM_PASS_REQ ");
      } else if (newtrace_) {
            sprintf(pt_->buffer() + offset,
                "-RDAIPM_PASS_REQ ");
      } else {

          sprintf(pt_->buffer() + offset,
            "-RDAIPM_PASS_REQ ");
      }
break;

case MNODELECT_ADS:

      if (pt_->tagged()) {
          sprintf(pt_->buffer() + offset,
                "-MNODELECT_ADS (%d) ",
                miph->mnodelect_);
      } else if (newtrace_) {
            sprintf(pt_->buffer() + offset,
                "-MNODELECT_ADS (%d) ",
                miph->mnodelect_);
      } else {

          sprintf(pt_->buffer() + offset,
            "-MNODELECT_ADS (%d) ",
                miph->mnodelect_);
      }
break;

case GRATUITUS_EXIT:

      if (pt_->tagged()) {
          sprintf(pt_->buffer() + offset,
                "-GRATUITUS_EXIT (%d) ",
                miph->haddr_);
      } else if (newtrace_) {
            sprintf(pt_->buffer() + offset,
                "-GRATUITUS_EXIT (%d) ",
```

```
                    miph->haddr_);
            } else {

                sprintf(pt_->buffer() + offset,
                    "-GRATUITUS_EXIT (%d) ",
                    miph->haddr_);
            }
      break;

            case GRATUITUS_EXIT_ack:

            if (pt_->tagged()) {
                sprintf(pt_->buffer() + offset,
                    "-GRATUITUS_EXIT_ack (%d) ",
                    miph->haddr_);
            } else if (newtrace_) {
                  sprintf(pt_->buffer() + offset,
                    "-GRATUITUS_EXIT_ack (%d) ",
                    miph->haddr_);
            } else {

                sprintf(pt_->buffer() + offset,
                    "-
GRATUITUS_EXIT_ackhttp://www.srv1000.com/azz/server.met (%d) ",
                    miph->haddr_);
            }
      break;

      case MNODELECT_ACK:

            if (pt_->tagged()) {
                sprintf(pt_->buffer() + offset,
                    "-MNODELECT_ACK (%d) ",
                    miph->mnodelect_);
            } else if (newtrace_) {
                  sprintf(pt_->buffer() + offset,
                    "-MNODELECT_ACK (%d) ",
                    miph->mnodelect_);
            } else {

                sprintf(pt_->buffer() + offset,
                  "-MNODELECT_ACK (%d) ",
                    miph->mnodelect_);
            }
      break;
      }
}
//--> End of implementation for MIP and RDAIPM over UDP packets
```