UNIVERSIDADE FEDERAL DO AMAZONAS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

"GAAP: UM PROTOCOLO DE ALOCAÇÃO DE ENDEREÇOS PARA REDES AD HOC MÓVEIS EM CENÁRIOS DE EMERGÊNCIA"

LAÉRCIO PÉRICLES BACELAR JÚNIOR

MANAUS

2011

UNIVERSIDADE FEDERAL DO AMAZONAS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LAÉRCIO PÉRICLES BACELAR JÚNIOR

"GAAP: UM PROTOCOLO DE ALOCAÇÃO DE ENDEREÇOS PARA REDES AD HOC MÓVEIS EM CENÁRIOS DE EMERGÊNCIA"

Dissertação submetida ao Departamento de Ciência da Computação da Universidade Federal do Amazonas, como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. Eduardo Souto

MANAUS

2011

FOLHA DE APROVAÇÃO

FICHA CATALOGRÁFICA

CODIGO	Junior, Laércio Péricles Bacelar GAAP: um protocolo de alocação de endereços para redes ad hoc móveis em cenários de emergência, Orientador Dissertação (Mestrado)

AGRADECIMENTOS

Gostaria, em primeiro lugar, agradecer a Deus que tem até aqui me conduzido e me levará a salvo ao seu Reino Celestial. Muitos foram os desafios e dificuldades nesses últimos anos, e Deus por sua graça tem me sustentado. A Ele toda glória, honra e louvor eternamente!

Agradeço aos meus pais Laércio e Emília por seu cuidado, apoio e compreensão durante a minha caminhada escolar, e em especial nesse mestrado.

Agradeço aos meus irmãos Andreyson e Bruno pelo incentivo.

Agradeço a todos os irmãos em Cristo da Igreja Presbiteriana de Manaus pelas orações e incentivo.

Agradeço a minha namorada Luciléia pelo imenso amor, compreensão e carinho nesses últimos anos.

Agradeço a Kamila Mendes pela ajuda na correção ortográfica da dissertação.

Agradeço ao meu orientador Eduardo Souto pelas sugestões, incentivo, compreensão e atenção.

Agradeço ao meu gerente Wladymir Carvalho pela cordialidade e compreensão na liberação do meu trabalho nas últimas semanas do mestrado.

Agradeço a Rafael Aschoff e Fernando Rodrigues pelo auxílio na implementação dos códigos-fonte dos protocolos e experimentos.

Agradeço a todos que direta e indiretamente contribuíram para a conclusão deste trabalho!

Muito obrigado!

LISTA DE FIGURAS

FIGURA 2.1: MODELO DE UMA MANET 21
FIGURA 2.2: EXEMPLO DE MODELO DE UMA REDE DE EMERGÊNCIA (ADAPTADO DE [MOTA09]) 24
FIGURA 2.3: EXEMPLO DE MOVIMENTAÇÃO DE EQUIPE DE RESGATE EM CENÁRIO DE EMERGÊNCIA 24
FIGURA 2.4: ALOCAÇÃO DE ENDEREÇO PARA UM NOVO NÓ EM UMA MANET..... 26
FIGURA 2.5: CENÁRIOS CRÍTICOS REFERENTES À MANUTENÇÃO DE ENDEREÇOS EM UMA MANET 27
FIGURA 2.6: MECANISMO DE ATRIBUIÇÃO DE ENDEREÇOS NO DCDP..... 32
FIGURA 2.7: ALGORITMO DE ALOCAÇÃO DE ENDEREÇOS DO PROPHET..... 33
FIGURA 2.8: ÁRVORE DE ALOCAÇÃO DE ENDEREÇO E ATUALIZAÇÃO DE ESTADO EM $\boldsymbol{f(n)}$ *[WEHBI05]*..... 34
FIGURA 2.9: MÁQUINA DE ESTADOS DO ALGORITMO DE SOLUÇÃO DE JUNÇÃO DE REDES..... 36
FIGURA 2.10: ÁRVORE DE ALOCAÇÃO DE ENDEREÇOS DO PRIME..... 38
FIGURA 3.1: FLUXO BÁSICO DE ALOCAÇÃO DE ENDEREÇO NO GAAP..... 47
FIGURA 3.2: EXEMPLO DE ALOCAÇÃO DE ENDEREÇOS LOCAL PARA TRÊS NÓS NO GAAP..... 49
FIGURA 3.3: EXEMPLO DE ALOCAÇÃO DE ENDEREÇOS REMOTA NO GAAP 50
FIGURA 3.4: RECUPERAÇÃO DE ENDEREÇOS NO GAAP..... 52
FIGURA 4.1: ESTRUTURA DE GRADE ESPIRAL 66
FIGURA 4.2: VIZINHANÇA NA ESTRUTURA DE GRADE ESPIRAL 66
FIGURA 4.3: LATÊNCIA DE CONFIGURAÇÃO NO CENÁRIO ESTÁTICO 70
FIGURA 4.4: ERROS DE CONFIGURAÇÃO NO CENÁRIO ESTÁTICO 72
FIGURA 4.5: SOBRECARGA DE MENSAGENS PERIÓDICAS EM KB NO CENÁRIO ESTÁTICO..... 73
FIGURA 4.6: SOBRECARGA TOTAL EM KB NO CENÁRIO ESTÁTICO 74
FIGURA 4.7: NÚMERO DE MENSAGENS BROADCAST NO CENÁRIO ESTÁTICO..... 75
FIGURA 4.8: NÚMERO DE MENSAGENS UNICAST NO CENÁRIO ESTÁTICO 75
FIGURA 4.9: MUDANÇAS DE ENDEREÇO NO CENÁRIO ESTÁTICO..... 76
FIGURA 4.10: LATÊNCIA DE CONFIGURAÇÃO NO CENÁRIO A 78
FIGURA 4.11: NÚMERO DE NETIDS GERADOS NO CENÁRIO A 79
FIGURA 4.12: NÚMERO DE NETIDS REMANESCENTES NO CENÁRIO A 80
FIGURA 4.13: MUDANÇAS DE ENDEREÇO NO CENÁRIO A..... 80
FIGURA 4.14: SOBRECARGA DE MENSAGENS PERIÓDICAS EM KB NO CENÁRIO A..... 82
FIGURA 4.15: SOBRECARGA TOTAL EM KB NO CENÁRIO A 82
FIGURA 4.16: NÚMERO DE MENSAGENS BROADCAST NO CENÁRIO A..... 83
FIGURA 4.17: NÚMERO DE MENSAGENS UNICAST NO CENÁRIO A 83
FIGURA 4.18: LATÊNCIA DE CONFIGURAÇÃO NO CENÁRIO B 84
FIGURA 4.19: NÚMERO DE NETIDS GERADOS NO CENÁRIO B..... 85
FIGURA 4.20: NÚMERO DE NETIDS RESTANTES NO CENÁRIO B..... 85
FIGURA 4.21: MUDANÇAS DE ENDEREÇO NO CENÁRIO B..... 87
FIGURA 4.22: SOBRECARGA DE MENSAGENS PERIÓDICAS EM KB NO CENÁRIO B..... 88
FIGURA 4.23: SOBRECARGA TOTAL EM KB NO CENÁRIO B 89

FIGURA 4.24: NÚMERO DE MENSAGENS BROADCAST NO CENÁRIO B 90
FIGURA 4.25: NÚMERO DE MENSAGENS UNICAST NO CENÁRIO B 91

ABREVIATURAS E ACRÔNIMOS

DAD	Duplicate Address Detection
DCDP	Dynamic Configuration and Distribution Protocol
DHCP	Dynamic Host Configurarion
FEMA	Federal Emergency Management Emergency
GAAP	Greedy Address Allocation Protocol
HCQA	Hybrid Centralized Query-based Autoconfiguration
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
MANET	Mobile Ad Hoc Network
NRC	National Research Council
NetID	Network Identifier
NS-3	Network Simulator version 3
NS-2	Network Simulator version 2
PACMAN	Passive Autoconfiguration for Mobile Ad Hoc Networks
PDAD	Passive Duplicate Address Detection
SDAD	Strong Duplicate Address Detection
TCP	Transmission Control Protocol
WDAD	Weak Duplicate Address Detection
WI-FI	Wireless Fidelity

ABSTRACT

In emergency scenarios, such as in natural disasters, technological or manmade, search and rescue teams in the affected region may use solutions for mobile ad hoc networks (Mobile Ad hoc Networks - MANETs) to address any lack of network infrastructure communication.

Emergency networks are those built on disaster scenarios and have properties such as robust and resilient communication, not necessarily have some communication infrastructure and mainly offer not only data communication but also voice. Thus, in case of lack of communication infrastructure, a MANET can act as an emergency network.

For communication between nodes in a MANET can be performed, it is necessary that each node is configured with unique address. Due to the ability of nodes self-organize, by creating temporary and arbitrary topologies, address allocation should be done automatically.

Many papers have appeared dealing with the allocation of addresses in MANETs, but most solutions presents a number of limitations related mainly to the limited applicability in scenarios and introduction of high control traffic on the network.

In this context, this paper presents a new solution for address allocation that assigns unique addresses to nodes in emergency networks. Results showed that the proposed solution is efficient in critical scenarios in comparison with other protocols examined, with low latency to obtain an address and reduced control traffic on the network reduced.

Keywords: MANETs, emergency networks, address allocation.

RESUMO

Em cenários de emergência, tais como em desastres naturais, tecnológicos ou causados pelo homem, equipes de busca e resgate da região afetada podem utilizar soluções de redes *ad hoc* móveis (*Mobile Ad hoc Networks* - MANETs) para suprir eventuais carências de infraestrutura da rede de comunicação.

Redes de emergência são aquelas construídas sobre cenários de desastres e têm propriedades tais como comunicação robusta e resiliente, não são necessariamente infraestruturadas e principalmente oferecem comunicação de dados e não somente voz. Assim, em caso de carência de infraestrutura de comunicação, uma MANET pode atuar como uma rede de emergência.

Para que a comunicação entre os nós de uma MANET possa ser realizada, é necessário que cada nó esteja configurado com endereço único. Devido à capacidade dos nós de se auto-organizarem criando topologias temporárias e arbitrárias, a alocação de endereços deve seja feita de forma automática.

Muitos trabalhos têm surgido tratando da alocação de endereços em MANETs, porém a maioria das soluções apresenta uma série de limitações relacionadas principalmente à aplicabilidade em cenários restritos e introdução de elevado tráfego de controle na rede.

Neste contexto, este trabalho apresenta uma nova solução de alocação de endereços que atribui endereços únicos aos nós em redes de emergência. Resultados mostraram que a solução proposta se mostrou eficiente em cenários críticos em comparação com outros protocolos analisados, apresentando baixa latência na obtenção de endereço e tráfego de controle reduzido na rede.

Palavras-chave: MANETs, redes de emergência, alocação de endereços

SUMÁRIO

1 INTRODUÇÃO 14

1.1 MOTIVAÇÃO 15
1.2 OBJETIVOS 18
1.3 ESTRUTURA DO TRABALHO 19

2 ESTADO DA ARTE 20

2.1 REDES AD HOC MÓVEIS 20
2.1.1 CENÁRIOS DE EMERGÊNCIA 21
2.1.2 REDES DE EMERGÊNCIA 23
2.2 ALOCAÇÃO DE ENDEREÇOS EM REDES AD HOC MÓVEIS 25
2.3 ABORDAGENS *STATELESS* 28
2.3.1 SDAD 28
2.3.2 WDAD 29
2.3.3 PDAD 30
2.4 ABORDAGENS *STATEFUL* 30
2.4.1 DCDP 32
2.4.2 PROPHET 33
2.4.3 PRIME DHCP 37
2.4.4 MANETCONF 40
2.5 ABORDAGENS HÍBRIDAS 41
2.5.1 PACMAN 41
2.5.2 HCQA 42
2.6 DISCUSSÃO 43

3 GAAP 44

3.1 ALOCAÇÃO DE ENDEREÇOS 46
3.2 MANUTENÇÃO DE ENDEREÇOS 51
3.2.1 RECUPERAÇÃO DE ENDEREÇOS 51
3.2.2 TRATAMENTO DE PARTIÇÕES E JUNÇÕES DE REDES 54
3.3 DESCRIÇÃO DAS MENSAGENS 55

3.3.1 MENSAGENS DE ALOCAÇÃO DE ENDEREÇO 56
3.3.1.1 Server Discovery 56
3.3.1.2 Pool Offer 56
3.3.1.3 Pool Request 56
3.3.1.4 Pool Reply 57
3.3.1.5 Remote Server Discovery 57
3.3.1.6 Remote Pool Offer 58
3.3.1.7 Remote Pool Request 58
3.3.1.8 Remote Pool Reply 59
3.3.1.9 Relay Server Discovery 59
3.3.2 MENSAGENS DE RECUPERAÇÃO DE ENDEREÇOS 60
3.3.2.1 Reclamation Request 60
3.3.2.2 Relay Reclamation Request 60
3.3.2.3 Reclamation Reply 61
3.3.2.4 Reclamation Inform 61
3.3.3 MENSAGENS PERIÓDICAS 62
3.3.3.1 Hello 62
4 EXPERIMENTOS E RESULTADOS 63
4.1 SIMULAÇÃO 63
4.2 CENÁRIOS 64
4.2.1 CENÁRIO ESTÁTICO 65
4.2.2 CENÁRIO DINÂMICO 67
4.3 MÉTRICAS 69
4.4 RESULTADOS DO CENÁRIO ESTÁTICO 70
4.4.1 LATÊNCIA DE CONFIGURAÇÃO 70
4.4.2 ERROS DE CONFIGURAÇÃO 71
4.4.3 SOBRECARGA DE MENSAGENS 73
4.5 RESULTADOS DO CENÁRIO DINÂMICO 77
4.5.1 CENÁRIO A 77

4.5.1.1 Latência de configuração *77*
4.5.1.2 Criação de redes *78*
4.5.1.3 Mudanças de endereço *80*
4.5.1.4 Sobrecarga de mensagens *81*
4.5.2 CENÁRIO B 84
4.5.2.1 Latência de configuração *84*
4.5.2.2 Criação de redes *85*
4.5.2.3 Mudanças de endereço *87*
4.5.2.4 Sobrecarga de mensagens *88*
4.6 DISCUSSÃO 91
5 CONCLUSÕES E TRABALHOS FUTUROS 94
5.1 CONCLUSÕES 94
5.2 TRABALHOS FUTUROS 95
REFERÊNCIAS BIBLIOGRÁFICAS 97

1 INTRODUÇÃO

Em cenários de emergência, tais como em desastres naturais, tecnológicos ou causados pelo homem, equipes de busca e resgate da região afetada podem utilizar soluções de redes *ad hoc* móveis[1] (*Mobile Ad hoc Networks* - MANETs) para suprir eventuais carências de infraestrutura de rede de comunicação [MOTA09]. MANETs são redes autônomas e auto-organizáveis, cujos dispositivos integrantes[2] podem se mover aleatoriamente e auto-organizarem suas tabelas de roteamento [MANET08].

Redes de emergência são aquelas construídas sobre cenários de desastres e têm propriedades tais como comunicação robusta e resiliente, não são necessariamente infraestruturadas e principalmente oferecem comunicação de dados e não somente voz [RAO07]. Dessa forma, em caso de carência de infraestrutura de comunicação, uma MANET pode atuar como uma rede de emergência.

Nas MANETs, a informação é transmitida diretamente entre os nós (comunicação através de um único salto) ou através de nós intermediários (comunicação com múltiplos saltos). Assim, cada nó integrante da rede pode, a qualquer momento, atuar como um roteador para assegurar a comunicação entre dois outros nós distantes.

Para que a comunicação entre os nós de uma MANET possa ser realizada, é necessário que cada nó tenha alguma forma de identificação. Os protocolos de roteamento assumem *a priori* que os nós móveis são configurados com um endereço IP válido (livre de conflitos) [SAK06]. Na camada de enlace, cada nó é identificado por um endereço MAC formado por 48 bits. Contudo, é necessário que cada nó final tenha um endereço de rede para que a comunicação entre dois nós finais seja realizada. Este endereço de rede será usado para identificar um nó de maneira única.

Usar soluções tradicionais como o DHCP [DROMS97] para atribuição de endereços aos nós em uma MANET não é uma solução, visto que a alta mobilidade dos nós inviabiliza a existência um servidor fixo de endereços. Empregar a técnica

[1] Em todo o documento, tanto o acrônimo MANET quanto “redes *ad hoc* móveis” são utilizados.
[2] Os dispositivos móveis integrantes de uma MANET também são chamados de “nós”.

de IP móvel [PERK96] também não é uma solução, visto que os nós geralmente são móveis e não ficam conectados a um serviço central o tempo inteiro. Portanto, a ausência de um coordenador centralizado torna a questão de endereçamento em MANET um problema interessante e desafiador. Neste contexto, em MANETs os nós necessitam cooperar uns com os outros para configurar a si mesmos com endereços únicos. Um protocolo de alocação de endereços é necessário para possibilitar a atribuição de endereços únicos de forma dinâmica a todos os nós na rede.

1.1 MOTIVAÇÃO

O processo de alocação de endereços para MANETs em cenários de emergência apresenta especiais desafios em comparação com redes infraestruturadas. Como previamente mencionado, a topologia da rede é altamente dinâmica e o emprego de uma autoridade central é inviável. A configuração manual é impraticável e, geralmente, não existe o papel do administrador da rede. O mecanismo de alocação de endereços da rede deve configurar cada nó com um endereço IP único antes que o roteamento possa iniciar. Assim, o processo de configuração de endereços deve ser rápido, visto que um nó não pode participar da comunicação até que tenha seu endereço configurado.

A saída repentina de um nó, aliada a eventos de partições e junções, representa um dos maiores desafios em MANETs. Se um nó fica desconectado permanentemente da rede, devido, por exemplo, à falha física ou carga baixa de bateria, seu endereço se torna inutilizável. Se isto acontece frequentemente, a rede rapidamente ficará sem endereços. Por outro lado, se o nó se desconecta temporariamente, a recuperação de seu endereço irá forçá-lo a requisitar um novo endereço quando se juntar à rede novamente.

Se a movimentação de nós acontece em grupo, eventos de partições e junções podem acontecer. Quando dois ou mais nós saem juntos do raio de alcance da rede, uma partição na MANET ocorre. Por outro lado, se um grupo de nós se junta a uma rede existente, uma junção acontece. Como eventos de partições e junções levam a um número maior de desconexões de nós, mais recuperações de endereço são necessárias na configuração de novos nós em caso de partições, e mais liberações de endereços em caso de junções. Além disso, a junção de redes

completamente distintas muito provavelmente leva a conflitos de endereços, se não for tratada adequadamente. Para o tratamento de partições e junções, tráfego de controle é gerado pelo mecanismo de alocação de endereço. Manter o tráfego de controle requerido para tratar esses eventos em um nível mínimo é outro desafio em MANETs.

Muitos trabalhos têm surgido tratando da alocação de endereços em MANETs [YUAN05, WEN05, VAID02 e ZHOU03], porém a maioria das técnicas foca em cenários específicos, na inundação de mensagens para tratamento de partições e junções de redes e não cobre de maneira eficiente todos os desafios apresentados.

A seguir são listadas as principais questões a serem consideradas na especificação de um protocolo de alocação de endereços [SAK06]:

- Espaço de endereços: a forma como o espaço de endereços é aproveitado pode variar de protocolo para protocolo. Deve-se evitar o desperdício de endereços durante a alocação, levando ao esgotamento rápido do espaço de endereços. Além disso, é importante que o protocolo consiga atribuir endereços a todos os nós da rede mesmo quando o espaço de endereços é reduzido.
- Processo de iniciação da rede (*bootstrapping*): na formação de uma nova rede, é necessário estabelecer o comportamento do primeiro nó da rede e o seu papel no atendimento a requisições de endereço de outros nós. Por exemplo, uma das questões a serem pensadas é se o primeiro nó da rede fará uso de alguma função de alocação e se a mesma será propagada aos próximos nós da rede.
- Processo de entrada de um novo nó: deve-se definir a forma como um novo nó poderá requisitar sua entrada na rede e as etapas de atendimento à solicitação. Por exemplo, precisaria ser avaliado se o novo nó fará sua requisição em *broadcast* pela rede e se realizará novas tentativas caso não receba nenhuma resposta.
- Distinção entre requisições concorrentes para atribuição de endereço: todo mecanismo de endereçamento precisa diferenciar requisições provenientes de diferentes nós, de forma a garantir a alocação de endereços para os nós corretos.
- Mobilidade: durante a concepção de qualquer mecanismo de endereçamento, a questão mobilidade deve ser levada em consideração.

O protocolo deve ser capaz de alocar endereços a novos nós mesmo em cenários móveis, onde perdas de mensagens trocadas durante o processo de alocação são mais comuns.

- Particionamento de rede: a mobilidade dos nós em MANETs pode levar a um particionamento na rede, que ocorre quando dois ou mais nós saem do raio de alcance dos demais. Para tratar partições, o mecanismo de endereçamento deve primeiramente ter uma identificação única para sua rede e ser capaz de lidar com os possíveis eventos a seguir:
  - Particionamento de rede e junção subsequente: quando um grupo de nós sai temporariamente da rede e retorna logo em seguida, o mecanismo de endereçamento deve aceitar o retorno dos nós visto que os mesmos ainda mantêm a identificação única da rede.
  - Partições se juntam novamente à rede: quando um grupo de nós sai da sua rede e depois de certo tempo as partições não apresentam a mesma identificação, o mecanismo de endereçamento deve iniciar o processo de junção.
  - Partições se movem independentemente ou se juntam a uma nova rede: no caso de partições originárias de redes distintas se juntarem a uma nova rede, o processo de junção de redes deve ser iniciado pelo mecanismo de endereçamento.
- Recuperação de endereços: a saída ou desligamento de nós da rede pode levar à perda de endereços e mais tarde ao esgotamento do espaço de endereçamento. Por isso, há a necessidade de um processo de recuperação que identifique os nós que deixaram a rede e retome os endereços perdidos.
- Detecção/distinção entre múltiplas MANETs: o mecanismo de endereçamento deve ser capaz de detectar a presença de outras redes e iniciar o processo de junção se necessário.
- Junção de redes independentes/MANETs: o mecanismo de endereçamento deve ser capaz de identificar a presença de redes distintas e disparar o processo de junção de redes, onde ao final do processo uma rede só exista com identificação única.
- Detecção de endereço duplicado: caso o mecanismo de endereçamento identifique nós com o mesmo endereço na rede, deve ser capaz de

resolver tais conflitos, solicitando a um dos nós que realize uma nova requisição de endereço.

- Tráfego de comunicação, energia limitada e baixa largura de banda: em todo projeto de um mecanismo de endereçamento, é necessário levar em conta restrições das MANETs como energia limitada dos dispositivos móveis e baixa largura de banda. Assim, o tráfego de controle propagado pelo mecanismo de endereçamento deve ter níveis aceitáveis para não consumir rapidamente os recursos da rede.

1.2 OBJETIVOS

Este trabalho propõe um novo protocolo de alocação de endereços para redes *ad hoc* móveis utilizadas por equipes de resgate em cenários de emergência como catástrofes e desastres. A solução proposta busca alocar endereços únicos aos nós de forma a reduzir o tráfego de controle introduzido pela detecção de duplicação.

O protocolo visa alocar os endereços de forma distribuída, mantendo a consistência do espaço de endereçamento durante o tempo de vida da rede. Assim, saídas individuais de nós bem como partições e junções devem ser levadas em consideração.

A solução proposta não é dependente de protocolo de roteamento e pode ser usada com protocolos proativos, reativos e híbridos. Além disso, a solução foi desenvolvida para redes IPv4, mas pode ser facilmente modificada para redes IPv6.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma: o Capítulo 2 caracteriza as MANETs e sua aplicabilidade em cenários de emergência. Além disso, é discutida a alocação de endereços e apresentadas as principais abordagens de endereçamento no contexto das MANETs; o Capítulo 3 discute em detalhes a solução de alocação de endereços proposta; o Capítulo 4 discorre sobre a metodologia de avaliação de desempenho da solução proposta, em comparação com outras soluções, e apresenta os resultados dos experimentos realizados; e, por fim, o Capítulo 5 apresenta as conclusões deste trabalho e algumas sugestões de trabalhos futuros.

2 ESTADO DA ARTE

Neste capítulo, é apresentado na seção 2.1 o conceito de redes *ad hoc* móveis e sua aplicabilidade em cenários de emergência; na seção 2.2 é discutida a alocação de endereços dentro do contexto das redes *ad hoc* móveis; nas seções 2.3 a 2.5 é apresentada uma classificação das principais soluções de alocação de endereços, bem como trabalhos relevantes existentes na literatura referentes a cada solução; por fim, na seção 2.6 é apresentada uma discussão das principais soluções de alocação de endereços apresentadas neste capítulo.

2.1 REDES AD HOC MÓVEIS

As redes *ad hoc* móveis, do inglês *Mobile Ad Hoc Networks* – MANETs, surgiram da necessidade de comunicar dispositivos móveis sem uma infraestrutura previamente instalada. As MANETs surgiram no começo dos anos 70, a partir de um projeto do departamento de defesa americano chamado DARPA *packet radio network* [JUBIN87], tornando-se mais tarde um interessante campo de pesquisa da indústria de computação.

Nas MANETs, os nós são livres para se mover de forma independente e aleatória tornando a topologia da rede altamente dinâmica. O tamanho da rede é constantemente alterado quando os nós entram e saem da área de cobertura da rede. Um nó não faz parte da MANET até que o mesmo esteja dentro do raio de transmissão de algum nó já configurado na MANET.

As MANETs têm uma característica intrínseca de os nós se comunicarem entre si sem a necessidade de uma entidade central que coordene a rede e exerça o papel de reencaminhar e rotear os pacotes da rede [AGG05]. Cada nó integrante da rede pode, a qualquer momento, atuar como um roteador para assegurar a comunicação entre dois outros nós distantes. Por exemplo, a Figura 2.1 ilustra um exemplo de MANET em que os nós A e D não podem se comunicar diretamente, por não estarem no raio de alcance um do outro. Desta forma, o nó B ou nó C deve atuar como roteador para que a comunicação entre os nós A e D seja realizada.

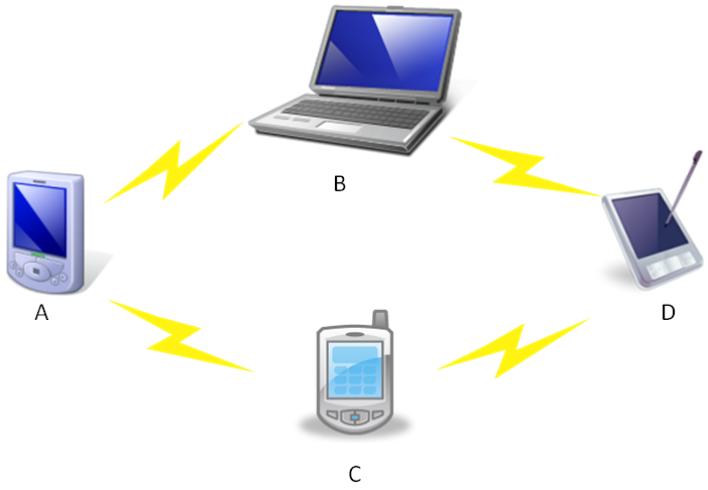

Figura 2.1: Modelo de uma MANET

Em cenários de emergência, tais como em desastres naturais, tecnológicos ou causados pelo homem, equipes de busca e resgate da região afetada podem utilizar as MANETs para suprir eventuais carências de infraestrutura de rede de comunicação. A subseção 2.1.1 descreve com mais detalhes o conceito de cenário de emergência.

2.1.1 Cenários de Emergência

Cenários de emergência são situações ocasionadas por desastres naturais, tecnológicos ou causados pelo homem, nos quais é interrompido o funcionamento normal da economia e da sociedade em grande escala [MOTA09].

Para minimizar o impacto de desastres em áreas afetadas, é necessária a ação coordenada dos agentes humanos envolvidos. Esses agentes podem ser bombeiros, policiais, médicos, entre outros. A comunicação entre as partes envolvidas deve ser realizada de maneira eficiente e o uso de tecnologia da informação pode auxiliar nesse trabalho.

No início de 2005, atendendo a um pedido da Agência Federal de Gerenciamento de Emergências (*Federal Emergency Management Agency, FEMA*) [FEMA11], o Conselho Nacional de Pesquisa Americano (*National Research Council, NRC)* [NRC11] criou o Comitê do Uso de Tecnologia da Informação no Gerenciamento de Desastres. Esse comitê gerou como resultado um relatório com oportunidades do uso de tecnologia da informação no gerenciamento de desastres

[RAO07]. No Brasil, a Defesa Civil Nacional é o órgão responsável pela gerência e definição do grau de um desastre ocorrido.

A Defesa Civil Brasileira caracteriza cenários de emergência levando em consideração alguns critérios preponderantes associados à intensidade dos danos (humanos, materiais e ambientais) e a ponderação dos prejuízos (sociais e econômicos). Com essa forma de avaliação, busca-se valorizar o contexto social no qual o desastre ocorreu para que seja possível relacionar os recursos disponíveis com as necessidades desejadas [SNDC09]. Os critérios preponderantes na avaliação de uma situação de emergência são:

1. Intensidade dos danos
    a. Danos Humanos
    b. Danos Materiais Destruídos ou Danificados
2. Danos Ambientais
3. Ponderação dos Prejuízos
    a. Prejuízos Econômicos
4. Prejuízos Sociais
    a. Alguns Critérios agravantes
    b. Ocorrência de desastres secundários
    c. Despreparo da administração local (geral e defesa civil)
    d. Grau de vulnerabilidade do cenário e da comunidade
    e. Padrão evolutivo do desastre

Com base nos critérios citados, não se pode caracterizar todo cenário de desastre como de emergência. Por exemplo, cenários de acidentes de trânsito e acidentes da construção civil, mesmo gerando prejuízos econômicos e sociais, não caracterizam situação de emergência.

A subseção 2.1.2 descreve uma solução eficiente em tecnologia da informação para comunicação entre os agentes humanos envolvidos no atendimento a situações de emergência, chamada *redes de emergência*.

2.1.2 REDES DE EMERGÊNCIA

Redes de emergência são aquelas construídas sobre cenários de desastres e têm propriedades tais como comunicação robusta e resiliente, não são necessariamente infraestruturadas e principalmente oferecem comunicação de dados e não somente voz.

Uma rede de emergência pode ser composta por um conjunto de nós heterogêneos. Estes nós podem ser representados pelas diferentes equipes de resgate, nós sensores para o monitoramento, pontos de acesso fixos que proveem conexão externa ao local do acidente e computadores gerenciadores [MOTA09]. A Figura 2.2 ilustra os componentes da infraestrutura *ad hoc* de uma rede de emergência. Os nós se comunicam de modo *ad hoc* entre si e podem trocar dados com nós gerenciadores. O acesso externo à Internet pode ser provido por pontos de acesso utilizando tecnologias comerciais como Wi-Fi [WIFI07], Bluetooth [BLUE05] ou WiMAX [WIMAX07].

As equipes podem ser formadas por participantes de instituições diferentes, governamentais ou não, como bombeiros, policiais, médicos, dentre outros. Assim, é necessária a interoperabilidade entre os integrantes das redes que podem estar com equipamentos de diversos modelos e tecnologias. Na Figura 2.2, por exemplo, equipes de bombeiros e policiais pertencem a redes distintas identificadas por 10.98.0.0 e 192.168.0.0. Para que essas redes distintas possam se comunicar, é necessário que cada nó seja identificado com um endereço único. A alocação de endereços únicos é uma das funções da solução proposta neste trabalho.

Os nós móveis podem ser dispositivos portáteis como *tablets* ou *laptops*, carregados pelos agentes das equipes de resgate. Esses dispositivos permitem a comunicação de dados e voz.

Os nós gerenciadores são dispositivos de maior capacidade de processamento e memória, responsáveis por receber e processar solicitações dos usuários. Essas solicitações podem ser, por exemplo, informações sobre localização de outras equipes e imagens de mapas da região afetada.

Os nós sensores podem ser utilizados para o monitoramento e sensoriamento de regiões em risco como áreas próximas de um incêndio e no envio dessas informações a um nó gerenciador.

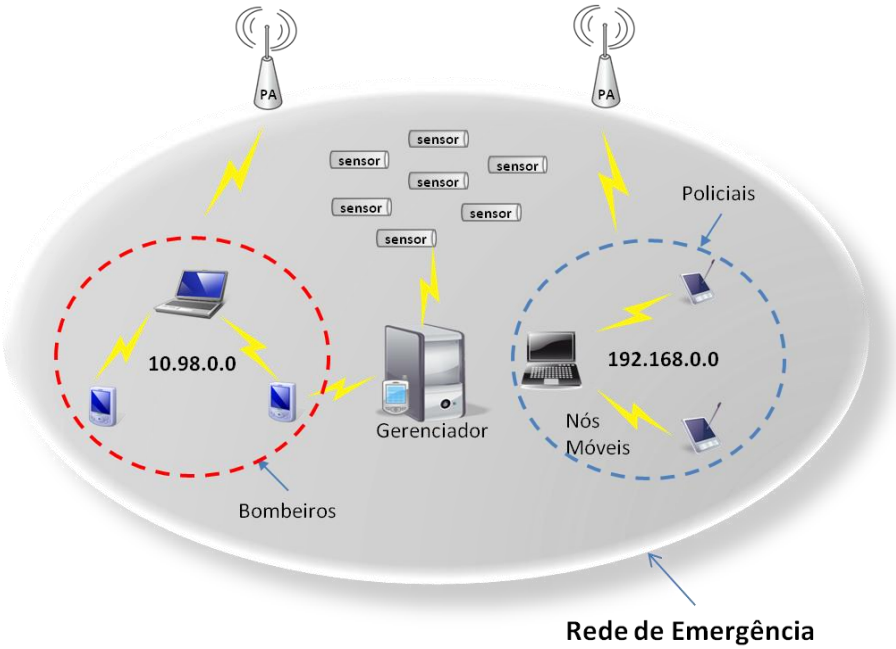

Figura 2.2: Exemplo de Modelo de uma Rede de Emergência (adaptado de [MOTA09])

A mobilidade em redes de emergência é resultado de eventos ocorridos e está associada aos locais para os quais os agentes devem se mover em cada atendimento. Por exemplo, no atendimento a um terremoto, diversas equipes atuam no resgate aos atingidos. Os locais para onde as vítimas são levadas (hospitais, abrigos etc) são considerados regiões de interesse. A Figura 2.3 ilustra a movimentação de uma equipe de resgate, partindo de uma base de apoio em direção a uma área afetada por um terremoto. Depois de resgatadas, as vítimas são levadas para um hospital e/ou para um abrigo se necessário.

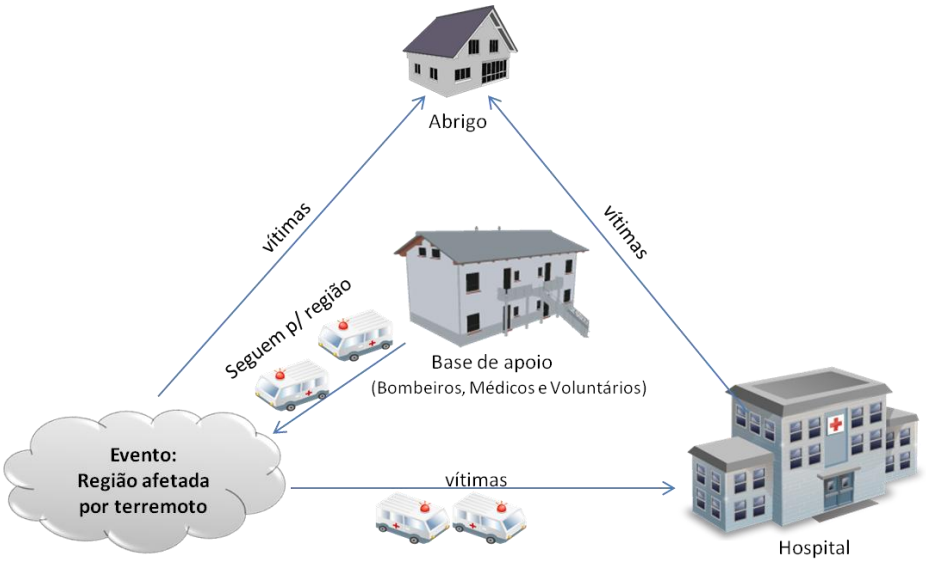

Figura 2.3: Exemplo de movimentação de equipe de resgate em cenário de emergência

2.2 ALOCAÇÃO DE ENDEREÇOS EM REDES AD HOC MÓVEIS

Como descrito anteriormente, as redes *ad hoc* móveis não possuem um nó central que coordene a rede e exerça o papel de reencaminhar e rotear os pacotes de dados. Assim, é necessário que todo nó da rede atue como roteador para que a comunicação seja realizada.

Para que o roteamento dos pacotes seja possível, é necessário que cada nó tenha alguma forma identificação na rede. Os protocolos de roteamento assumem *a priori* que os nós móveis são configurados com um endereço IP válido (livre de conflitos) [SAK06]. Na camada de enlace, cada nó é identificado por um endereço MAC formado por 48 bits. Contudo, é necessário que cada nó final tenha um endereço de rede para que a comunicação entre dois nós finais seja realizada. Este endereço de rede será usado para identificar um nó de maneira única.

Devido à capacidade dos nós de se auto-organizarem criando topologias temporárias e arbitrárias, faz-se necessário que a distribuição de endereços seja feita de forma automática.

Em redes cabeadas, a distribuição de endereços é geralmente realizada de forma centralizada. Nesse caso, um nó da rede é responsável por atender às requisições de endereço. Um exemplo de mecanismo de alocação de endereços bastante utilizado em redes cabeadas é o DHCP (*Dynamic Host Configuration Protocol*) [DROMS97].

No contexto das MANETs, há predominância do uso de mecanismos de alocação descentralizada de endereços, devido principalmente ao caráter dinâmico dessas redes. A mobilidade dos nós torna pouco recomendada a centralização da tarefa de atribuição de endereços em um único nó. Este nó pode eventualmente sair da rede, comprometendo a atribuição de endereços até que outro nó da rede exerça o papel do nó que saiu ou este retorne à rede.

A Figura 2.4 ilustra um típico processo de alocação de endereço em uma MANET. Em (a), o nó Z que ainda não dispõe de um endereço é incapaz de estabelecer uma conexão direta com os nós já configurados X, Y, V e W. Já em (b), o mecanismo de endereçamento da rede deve garantir a alocação de um endereço único para Z, o qual pode ter sido oferecido por qualquer dos outros nós já configurados. Depois da obtenção do novo endereço, o nó Z consegue então se comunicar com os demais nós da rede.

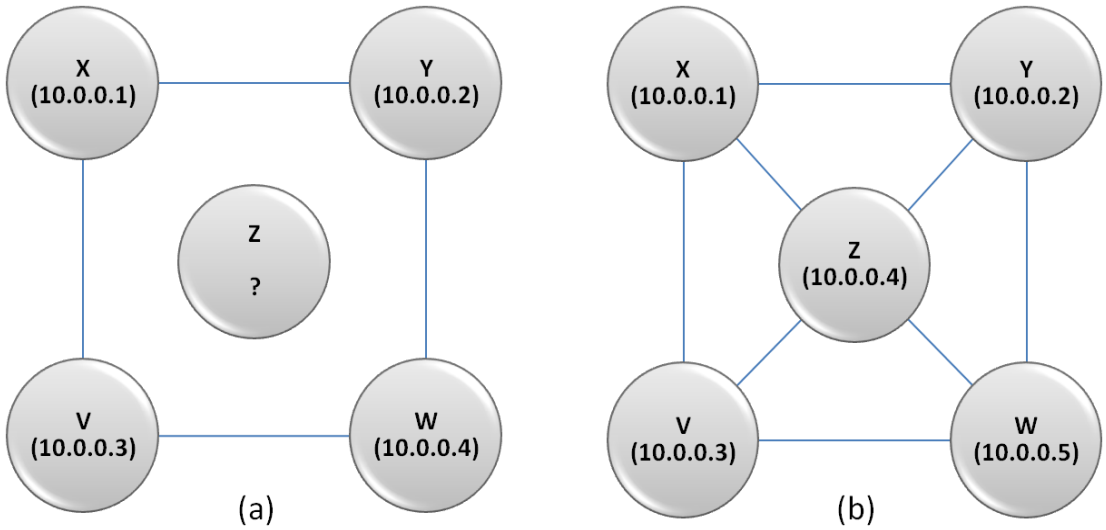

Figura 2.4: Alocação de endereço para um novo nó em uma MANET.

Além de atribuir identificadores aos dispositivos de forma dinâmica, é necessário manter a unicidade dos endereços. Para isso, a rede deve ser capaz de realizar as seguintes tarefas [NETO10]:

(a) Detectar endereços repetidos: pode ser realizada se duas redes distintas se juntam ou um mesmo endereço é atribuído equivocadamente para nós distintos;

(b) Recuperar endereços perdidos: pode ser necessária quando um endereço alocado não está mais em uso. Por exemplo, a recuperação pode ser realizada em caso de partição na rede, falta de bateria ou defeito físico no nó.

A Figura 2.5 ilustra três cenários críticos referentes ao processo de manutenção de endereços. Em (a), o nó Z deixa a rede e sua saída pode ser causada por diversos fatores, tais como falta de bateria ou defeito no próprio dispositivo. Se a saída do nó acontecer repentinamente, é muito provável que o nó não consiga indicar sua saída a tempo, ficando seu endereço inutilizado. Em caso de ausência de endereços para alocação, um processo de recuperação do endereço recém-perdido será necessário. Em (b), um grupo formado pelos nós X e Y movem-se para fora do alcance da rede principal, dividindo-a em duas partições distintas. Em (c), o grupo formado pelos nós X e Y retorna à rede de origem, depois de se ausentarem temporariamente. Isso pode causar inconsistência, caso os endereços sejam recuperados antes de os nós X e Y retornarem à rede. Ao contrário de (b) onde é ilustrada a junção de redes particionadas, em (c) é apresentada a junção de

duas redes completamente distintas. Nesse caso, há uma grande probabilidade de algum endereço de uma rede esteja presente também na outra, já que os processos de alocação de endereço das redes são completamente independentes.

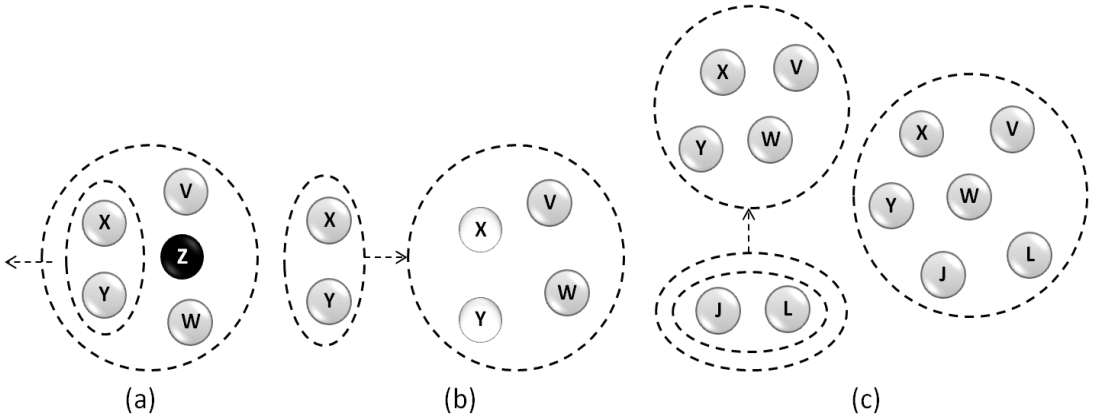

Figura 2.5: Cenários críticos referentes à manutenção de endereços em uma MANET

Um dos maiores desafios no gerenciamento do espaço de endereços é saber identificar um endereço perdido e também o melhor momento de recuperá-lo. Depois de detectar uma perda no espaço de endereços, se a recuperação for realizada imediatamente, um tráfego de controle maior pode ser introduzido, já que há o risco do antigo dono do endereço perdido retornar à rede e ter que iniciar uma nova requisição de endereço. Por outro lado, se a detecção ou o processo de recuperação atrasar, um novo nó pode não obter endereço se a rede não possuir endereços disponíveis.

Portanto, além de distribuir endereços únicos aos nós de forma automática, a rede deve manter a unicidade dos endereços alocados. O processo de manutenção envolve o tratamento de partições e junções, realizando a detecção de endereços repetidos e recuperação de endereços perdidos quando necessário. Nas seções 2.3 a 2.5, são apresentadas as principais soluções de alocação de endereços para MANETs presentes na literatura, podendo ser classificadas em abordagens *stateless, stateful* e híbridas [SAK06].

2.3 ABORDAGENS *STATELESS*

Nas abordagens *stateless*, os nós geram seus próprios endereços e utilizam o mecanismo DAD (*Duplicate Address Detection)* para detectar conflitos de endereço e tentar garantir que apenas endereços únicos sejam alocados. Estas abordagens têm a vantagem de não precisarem de mecanismos de recuperação de endereços, pois os nós podem gerar endereços novamente. No entanto, o mecanismo de detecção de conflito de endereço inunda toda a rede com mensagens de controle, de forma a verificar a unicidade do endereço gerado junto a todos os nós. Há pelo menos três mecanismos de detecção de duplicação de endereço: SDAD [PERK01], WDAD [VAIDYA02] e PDAD [WEN03].

2.3.1 SDAD

No SDAD (do inglês *Strong Duplicate Address Detection*) [PERK01], cada nó atribui seu próprio endereço na rede e verifica se este endereço já está em uso ou não. Um nó em sua inicialização obtém dois endereços: um “endereço temporário” e um “endereço de tentativa”. O nó utiliza o endereço temporário como um endereço fonte de requisições propagadas na rede para verificar se o endereço de tentativa está em uso ou não.

O novo nó propaga um pacote de requisição de endereço destinada ao endereço de tentativa e espera por uma resposta. Se uma resposta é recebida pelo nó, o mesmo conclui que o endereço está em uso. Se nenhuma resposta é recebida, a requisição é repetida certo número de vezes ao endereço de tentativa, de forma a assegurar que o endereço não está em uso, antes de liberar o endereço temporário e passar a usar o endereço de tentativa.

Apesar de sua simplicidade, o SDAD apresenta algumas desvantagens. Mesmo enviando repetidamente mensagens de requisição de endereço para assegurar que o endereço de tentativa não está em uso, a detecção de endereço duplicado se restringe à fase de inicialização da rede. Se um nó configurado se desconectou temporariamente da rede, seu endereço pode erroneamente ser considerado como não usado. Além disso, partições e junções de rede não são

consideradas, o que pode levar a ocorrências de duplicação de endereço não detectadas.

Como pode ser verificado, o SDAD não garante a unicidade de endereço. Além disso, a probabilidade de duplicação cresce com o tamanho da rede em caso de espaço de endereçamento limitado. Finalmente, um alto tráfego de controle é gerado cada vez que um nó se junta à rede, devido a inúmeras mensagens propagadas por toda a rede.

2.3.2 WDAD

Diferentemente do SDAD, o WDAD (do inglês *Weak Duplicate Address Detection*) [VAID02] estende a detecção de duplicação de endereço para o tempo de vida da rede, em vez de aplicar apenas durante a fase de inicialização de cada nó. No WDAD, cada nó mantém em sua camada de roteamento um endereço virtual que consiste de uma chave de identificação e seu endereço IP. Em uma comunicação, um nó é capaz de enviar pacotes para o “nó destino” mesmo se o endereço deste destino já estiver sendo usado, pois cada nó usa sua chave de identificação para diferenciar entre possíveis endereços duplicados.

Todo nó da rede gera sua chave de identificação na fase de inicialização e propaga-a nas mensagens de roteamento, de forma que chaves e endereços IP são mantidos nas tabelas de roteamento locais dos nós. Se um nó recebe uma mensagem de roteamento com endereço de destino presente em sua tabela de roteamento, as chaves de identificação são comparadas. Se forem diferentes, a duplicação de endereço é detectada. Então, o nó marca a entrada correspondente na tabela de roteamento como inválida e informa os demais nós da rede.

O WDAD tem a desvantagem de depender do protocolo de roteamento, pois requer algumas modificações na camada de roteamento para adição da chave de identificação. O protocolo não gera qualquer tráfego de controle adicional para autoconfiguração da rede, mas o uso da chave de identificação em pacotes de roteamento aumenta a sobrecarga de controle.

2.3.3 PDAD

Semelhantemente ao WDAD, o PDAD (do inglês *Passive Duplicate Address Detection*) [WEN03] utiliza mensagens de controle do protocolo de roteamento para detectar duplicação de endereço.

O PDAD é aplicável somente a roteamento proativo, onde cada nó propaga mensagens a respeito de sua vizinhança através da rede. Tais mensagens contêm números de sequência para distinguir entre pacotes antigos e novos. Se um nó recebe uma mensagem com seu endereço IP como o endereço fonte e um número de sequência maior que seu próprio valor, um conflito de endereço é detectado.

A vantagem do protocolo é que nenhum tráfego de controle adicional é gerado, sendo o PDAD aplicável somente para roteamento proativo. Assim, o método requer análise complexa da informação de roteamento para detectar duplicação de endereço.

2.4 ABORDAGENS *STATEFUL*

Abordagens *stateful* mantêm informações tais como tabelas de alocação de endereço de forma que se tenha registro de quais endereços estão disponíveis ou não para serem usados pelos nós. Como apenas endereços livres são alocados, soluções *stateful* não precisam de nenhum mecanismo de detecção de endereço duplicado. A unicidade de endereço é a maior vantagem das abordagens *stateful* em relação às abordagens *stateless*, já que a geração de sobrecarga na rede para detecção de endereço duplicado é evitada.

De acordo como usam o espaço de endereçamento para alocar os endereços e gerenciá-lo, as abordagens *stateful* podem adotar algum dos mecanismos a seguir [ASCH10]: *tabela de alocação centralizada*, *tabela de alocação distribuída*, *múltiplas tabelas de alocação* ou *função de alocação distribuída*.

Nas abordagens *stateful* que utilizam a estratégia centralizada, uma tabela de endereços única é mantida por um nó central responsável pela alocação dos endereços. Na estratégia distribuída com tabela de alocação única, uma cópia da tabela de endereços é mantida em cada nó. Por sua vez, a estratégia distribuída com múltiplas tabelas de alocação mantém em cada nó porções disjuntas da tabela

de endereços. Por fim, a estratégia com função de alocação distribuída armazena em cada nó uma função de alocação baseada em valores de estado e semente, utilizados na geração e alocação de endereços.

Abordagens *stateful* com tabela de alocação centralizada não precisam gerar endereços e dispensam o uso de DAD para resolução de conflitos de endereço. No entanto, a concentração da tabela de endereços em um único nó é uma grande desvantagem, pois o nó central pode sair da rede acarretando indisponibilidade no processo de alocação até que outro nó assuma o papel do nó central ou este retorne à rede. Um exemplo de método com tabela de alocação centralizada é o *Agent Based Addressing* [GÜN02].

As abordagens *stateful* com tabela de alocação distribuída têm a vantagem de manterem em cada nó uma cópia da tabela de endereços, o que diminui o atraso na alocação de endereços, pois os nós podem requisitar endereços a seus vizinhos e não precisam depender de uma entidade central. No entanto, há um elevado custo de sincronização das tabelas de endereços dos nós, de forma a evitar que um endereço existente na rede seja realocado a outros nós, gerando conflitos de endereços. Exemplo desse tipo de abordagem é o MANETconf [NES02].

Nas abordagens *stateful* com múltiplas tabelas de alocação, os nós não precisam manter a tabela de endereços completa, mas apenas porções disjuntas do espaço de endereços da rede, garantindo a unicidade dos endereços alocados. A principal desvantagem dessa técnica é que a rede pode apresentar um elevado número de redirecionamento de requisições quando não há mais endereços disponíveis em alguns nós da rede, contribuindo para o aumento do tráfego de controle e atraso na alocação de endereços. Exemplos desse tipo de abordagem são DCDP [MISRA01] e *Buddy* [MOH02].

Por fim, as abordagens *stateful* com função de alocação distribuída geralmente dispensam mecanismos de detecção de conflitos de endereços por gerarem endereços únicos ou com baixa probabilidade de conflito. Em geral utilizam mecanismos complexos de recuperação de endereços e tratamento de partições e junções de rede. Exemplos desse tipo de abordagem são *Prophet* [ZHOU03] e *Prime* DHCP [YUAN05].

A seguir são descritos alguns métodos baseados na abordagem *stateful*, que estão dentre os mais conhecidos na literatura.

2.4.1 DCDP

No DCDP (do inglês *Dynamic Configuration and Distribution Protocol*) [MISRA01], o primeiro nó na rede atua como um servidor DCDP. Cada nó na MANET desempenha um papel integrante no processo de alocação de endereço. Quando um nó recebe uma requisição de endereço de outro nó que deseja se juntar à rede, o nó ofertante divide seu bloco de endereços (ou *pool* de endereços) ao meio e aloca uma metade ao nó requisitante.

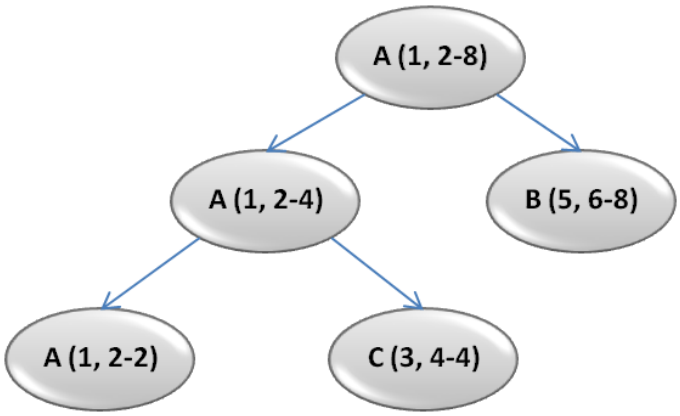

Figura 2.6: Mecanismo de atribuição de endereços no DCDP

Por exemplo, na Figura 2.6, o nó A com endereço 1 representa o primeiro nó na rede. Tal nó tem um *pool* de endereços com endereços de 2 a 8. Quando B se junta à rede, o mesmo recebe o endereço 5 e A divide seu *pool* em metades e dá uma metade a B. Então o *pool* de endereços de A torna-se de 2 a 4, enquanto B fica com *pool* de endereços de 6 a 8. Quando C se junta à rede como um vizinho de A, o mesmo recebe de A o endereço 3 e A divide seu *pool* de endereços em metades novamente e dá a C um *pool* que tem apenas o endereço 4 disponível para alocação. Depois disso, o *pool* de endereços de A tem apenas o endereço 2.

Uma vantagem desta técnica é o suporte ao tratamento de partições e junções de redes. Se um particionamento ocorre na rede, as partições formadas mantêm consigo *pools* de endereços distintos e caso se reconectem não é necessário usar qualquer mecanismo para tratar conflitos de endereços.

Uma desvantagem do DCDP é o reuso de endereço IP alocado. Quando um nó na rede é desligado com prévio aviso, o mesmo pode liberar seu endereço IP bem como seu *pool* de endereços. Mas se o nó deixa a rede sem avisar ou desliga subitamente, o mesmo irá levar consigo seu endereço IP e seu *pool* de endereços

pode tornar-se não usável. Nesse caso, é necessário utilizar algum mecanismo para recuperação de endereços. Tal mecanismo deve ser acionado apenas se necessário, devido ao tráfego de controle adicional gerado na rede.

2.4.2 PROPHET

O *Prophet Allocation* [ZHOU03] é um mecanismo de autoconfiguração de endereço onde o primeiro nó da rede, chamado profeta, é capaz de descobrir previamente o endereço IP de cada futuro integrante da rede. O nó profeta inicialmente escolhe aleatoriamente um número como seu endereço IP e um estado inicial (semente) para função $f(n)$ que gera o endereço IP de cada nó da rede. A função $f(n)$ tem as seguintes propriedades [ZHOU03]:

(a) Gera uma sequência de inteiros distintos a partir de uma semente escolhida aleatoriamente;

(b) A probabilidade de um número se repetir em sequências distintas geradas a partir de sementes distintas é muito pequena.

Ao receber a requisição de endereço de um nó que deseja entrar na rede, o nó profeta, utilizando a função $f(n)$, gera um valor inteiro e um estado e os repassa ao nó requisitante juntamente com a função $f(n)$. O nó requisitante considera o valor inteiro como seu endereço IP e o estado como a semente de sua função $f(n)$. Para melhor entendimento, a Figura 2.7 descreve o algoritmo de alocação de endereços do *Prophet*.

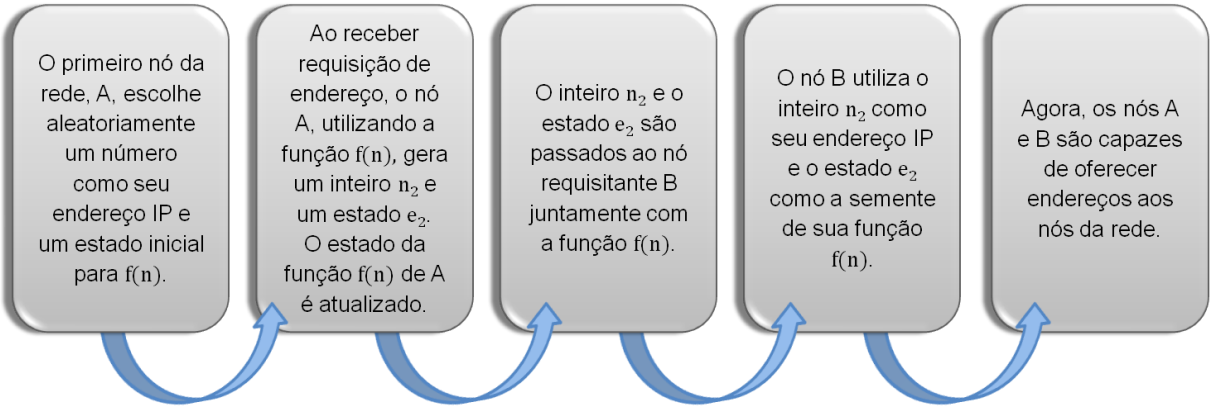

Figura 2.7: Algoritmo de alocação de endereços do *Prophet*

Como existe uma probabilidade mesmo que muito pequena de conflitos de endereços, o nó profeta pode previamente detectar as ocorrências e avisar os nós da rede antecipadamente. Se o nó profeta identificar vários conflitos, uma nova semente pode ser escolhida para gerar outras sequências até que haja poucos conflitos. Para gerar sequências distintas dentro de um determinado intervalo, a função $f(n)$ utiliza a propriedade dos números inteiros positivos descrita pela equação em (2.1), onde: p_i são primos e $p_1 < p_2 < \cdots < p_k$.

$$n = \prod_{i=1}^{k} p_i^{e_i} \tag{2.1}$$

Os expoentes são números naturais e todo inteiro positivo n pode ser representado por uma *k*-tupla $(e_1, e_2, e_3, \ldots, e_k)$. A ideia do algoritmo de alocação de endereços é gerar diferentes *k*-tuplas para cada nó.

Como exemplo, suponha $k = 4$. Seja x o endereço do primeiro nó da rede e $(0,0,0,0)$ o estado inicial da função $f(n)$ do nó. Os demais nós da rede são representados por $\big(endereço, (e_1, e_2, e_3, e_4)\big)$, onde $endereço$ é representado pela equação em (2.2). Nessa equação, r é o tamanho do espaço de endereçamento.

$$endereço = ((x + 2^{e_1}3^{e_2}5^{e_3}7^{e_4}) \bmod r) + 1 \tag{2.2}$$

A Figura 2.8 mostra a árvore de geração de endereço e atualização de estado em $f(n)$ para uma rede com 4 nós e espaço de endereçamento $r = 4$.

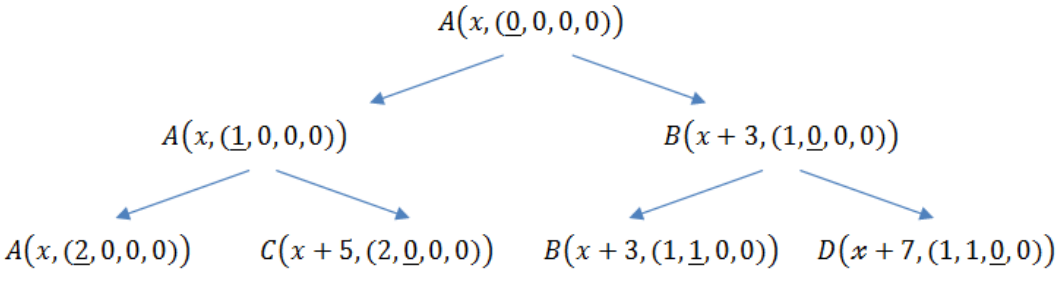

Figura 2.8: Árvore de alocação de endereço e atualização de estado em $f(n)$ [WEHBI05]

O primeiro nó da rede, chamado A, têm endereço x e estado inicial igual a $(0,0,0,0)$. Quando o nó B entra na rede, o mesmo recebe o par $(x + 3, (1, 0, 0, 0))$,

onde endereço $x+3$ é calculado a partir do seu estado $(1,\underline{0},0,0)$, segundo a equação em (2.2) da seguinte forma:

$$endereço = \left((x + 2^1 3^0 5^0 7^0)\ mod\ 4\right) + 1 = x + 3$$

O estado $(1,\underline{0},0,0)$ do nó requisitante B é obtido a partir do estado $(\underline{0},0,0,0)$ do nó ofertante A, seguindo as regras abaixo:

(a) O elemento sublinhado na tupla do nó ofertante é acrescido de 1;

(b) O estado do nó ofertante é copiado para o estado do novo nó e a sublinha é deslocada em 1 para direita.

Seguindo as regras de atualização de estado, o elemento sublinhado do estado $(\underline{0},0,0,0)$ do nó A é primeiramente incrementado em 1, resultando em $(\underline{1},0,0,0)$. Em seguida, o estado $(\underline{1},0,0,0)$ de A é copiado para o estado do novo nó B e a sublinha é deslocada em 1 para direita, ficando B com o estado $(1,\underline{0},0,0)$.

Na Figura 2.8, o comportamento de alocação de endereço e atualização de estado dos nós A e B é o mesmo para os nós C e D.

Para aplicações reais, o valor de *r* deve ser suficientemente grande, e, por conseguinte também o valor de k, produzindo tuplas maiores. Para acelerar o processamento do algoritmo, uma coleção (*array*) de números primos pré-calculados pode ser usada na alocação.

O *Prophet* dispensa a recuperação de endereços visto que os endereços perdidos podem ser novamente gerados pela função $f(n)$. Como já mencionado, uma sequência de inteiros positivos é gerada de forma cíclica dentro de um intervalo suficientemente grande. Além disso, se acontecer de ser gerada uma sequência com vários conflitos, uma nova semente pode ser escolhida e o processo de geração é reiniciado.

Partições de rede não precisam ser tratadas pelo *Prophet*, pois ao saírem alguns nós da rede, os demais nós continuam alocando endereços através de suas funções $f(n)$.

Junções de redes, por sua vez, devem ser tratadas pelo protocolo. O nó profeta, durante sua configuração, gera também o identificador de sua rede, conhecido como *Network ID* (NetID), o qual é repassado para os demais nós ainda no processo de alocação. Ao detectar uma rede com NetID diferente, um nó verifica

se o NetID dentro da mensagem recebida é menor que o seu. Em caso afirmativo, o nó inicia uma requisição de endereço para outra rede e comunica aos demais nós que façam o mesmo.

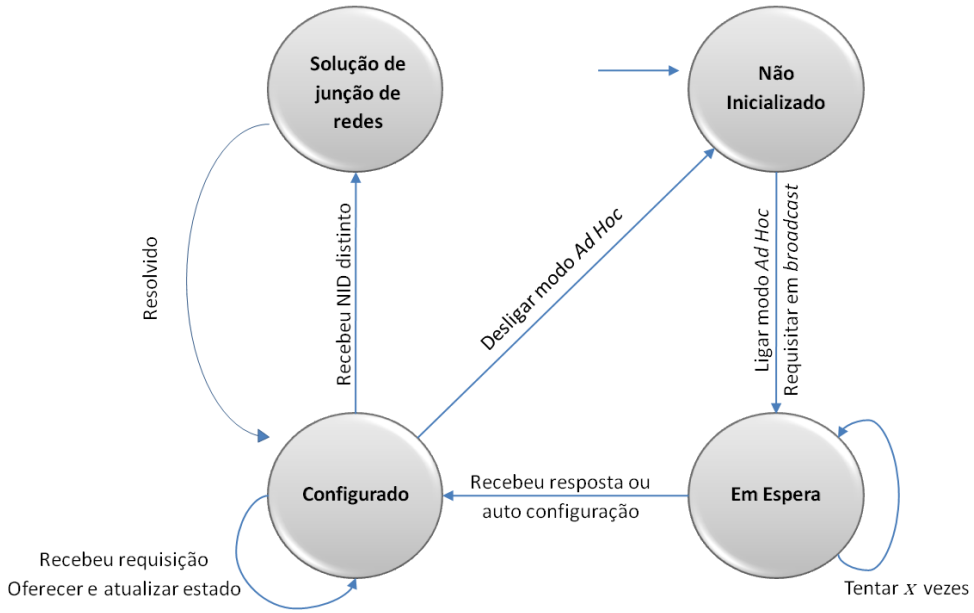

Figura 2.9: Máquina de estados do algoritmo de solução de junção de redes

De forma conclusiva, o protocolo *Prophet* pode ser descrito em seis passos conforme ilustra a Figura 2.9. Quando um nó não está integrado a nenhuma rede sem fio, o mesmo está no estado inicial *Não inicializado*. Se há interesse em se integrar a alguma rede sem fio, o nó liga o modo A*d Hoc* e começa a enviar mensagens de requisição de endereço em *broadcast* informando seu endereço MAC para que a resposta seja feita em *unicast*. Depois do envio das mensagens de requisição, o nó passa do estado *Não inicializado* para *Em espera*. Caso não receba nenhuma resposta, o nó permanece no estado *Em Espera* e repete o processo por até x vezes. Em caso de resposta, o nó recebe um valor inteiro que utilizará como endereço IP, o estado inicial de sua função $f(n)$ e o NetID, presentes na mensagem. Caso não obtenha resposta, o nó cria uma nova rede e se torna profeta. Em ambos os casos, o nó passa para o estado *Configurado*. No estado *Configurado*, o nó envia periodicamente mensagens *Hello* que podem ser encapsuladas nas mensagens de roteamento. Além disso, o nó responde a requisições de endereços e atualiza o estado de sua função $f(n)$. Ao receber uma mensagem *Hello* contendo um NetID distinto do seu, o nó passa ao estado *Solução de junção de redes*, onde executa os passos descritos anteriormente para tratamento de junção de redes.

Depois de concluída a etapa de solução de junção de redes, o nó retorna para o estado *Configurado*. Por fim, caso desligue o modo *Ad Hoc,* o nó retorna para o estado *Não inicializado*.

2.4.3 PRIME DHCP

O *Prime* DHCP [YUAN05] é um mecanismo de endereçamento que se baseia, assim como o *Prophet,* no uso de uma função geradora de números inteiros positivos durante o processo de alocação de endereços. A alocação de endereços no *Prime* DHCP se dá basicamente da seguinte forma:

(a) O primeiro nó da rede, chamado raiz, tem endereço igual a 1;

(b) Os endereços dos demais nós são gerados multiplicando o endereço do nó ofertante pelo número primo não usado dentro do nó, começando pelo maior fator primo resultante da fatoração do seu próprio endereço.

O processo de alocação gera números primos distintos, sem possibilidade de duplicação. Cada nó deve armazenar basicamente um conjunto de número primos e o seu estado, isto é, o último primo utilizado. Como exemplo, suponha que o nó raiz tem armazenado um conjunto X de primos $\{1, 3, 5, \cdots, N\}$ e seu estado inicial é igual ao seu endereço (1). O nó inicialmente oferece o endereço 3, resultante da multiplicação do seu endereço (1) pelo próximo primo não usado dentro do nó (3), excluindo o fator primo 1 já usado em sua alocação. O nó raiz então atualiza o seu estado para 3, o último número primo utilizado. Quando recebe uma nova requisição, o nó raiz gera o endereço 15, resultante da multiplicação do seu estado atual (3) pelo próximo não usado (5) de X, e atualiza o seu estado para 5, e assim por diante. A Figura 2.10 ilustra a árvore de alocação de endereços do *Prime*.

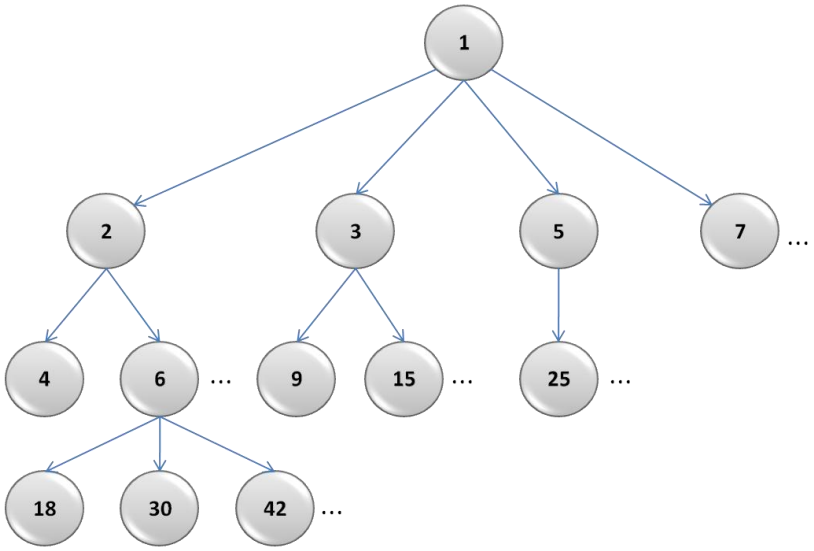

Figura 2.10: Árvore de alocação de endereços do *Prime*

Quando um nó deseja entrar na rede, o mesmo envia uma mensagem de requisição de endereço em *broadcast* chamada *DHCP_Discover* e aguarda resposta. Se não houver resposta, o nó repete o processo até certo número de vezes. Não havendo resposta, o nó se autoconfigura com endereço 1, torna-se o nó raiz da rede e também capaz de atender requisições de endereço de outros nós.

Quando um nó previamente configurado recebe uma mensagem de requisição de endereço, é gerado um endereço utilizando a função de geração descrita anteriormente e enviado em *unicast* ao nó requisitante dentro de uma mensagem *DHCP_Offer*. Se não puder oferecer endereço, o nó passa a atuar como um *proxy*, encaminhando a mensagem *DHCP_Discover* ao nó pai que lhe ofereceu endereço. Ao receber várias ofertas, o nó requisitante escolhe a oferta de menor endereço, evitando assim que a árvore de alocação de endereço cresça rapidamente. Para comunicar o nó ofertante de sua escolha, o nó requisitante envia uma mensagem *DHCP_Request* em *broadcast*. Tal mensagem só será propagada pela rede caso a oferta tenha sido gerada por um ancestral de um nó vizinho ao nó requisitante.

Ao receber a mensagem *DHCP_Request,* o nó ofertante atualiza o seu estado e envia uma mensagem *DHCP_Ack* em *unicast* ao nó requisitante, concluindo o processo de alocação de endereço. Segundo a especificação do protocolo, caso o nó queira sair da rede “educadamente”, o mesmo envia uma mensagem *DHCP_Release* em *unicast* para o seu pai, de forma a acelerar o processo de recuperação de endereços e evitar a perda de endereços. Depois de liberado o

endereço, o nó pai pode oferecê-lo na próxima alocação. Se o nó que está saindo for a raiz da rede, então o mesmo avisa o último filho alocado para que se torne a nova raiz. É importante destacar que mensagens *DHCP_Release* não são esperadas ou obrigatórias, visto que dificilmente os nós saem “educadamente” da rede.

Partições na rede não causam problema, pois os nós não podem gerar endereços iguais mesmo estando separados, não havendo necessidade de tratamento pelo protocolo. Porém, para evitar perda de endereços devido à saída de nós sem aviso, o nó raiz envia periodicamente mensagens *Recycle* que devem ser propagadas pela rede e respondidas por cada nó com a informação do seu estado atual. Depois de receber todas as respostas dos nós, o nó raiz pode reconstruir a árvore de alocação de endereço e identificar eventuais saídas não reportadas. Havendo saídas não reportadas, o nó raiz comunica a ocorrência aos pais de cada endereço recuperado.

Junções de redes podem ser também detectadas pelo processo de reciclagem periódica. Ao receber mensagens *Recycle* originárias de nós de uma rede distinta, a raiz verifica os estados de alocação dos nós na outra rede, detecta conflitos de endereços e solicita a um dos nós conflitantes que inicie uma nova requisição de endereço. Embora partições de rede não gerem conflitos de endereços como citado anteriormente, a rede particionada precisará de uma nova raiz. Nesse caso, cada nó que não receber várias mensagens *Recycle*, pode se considerar a nova raiz após um tempo inversamente proporcional ao seu endereço, para que nós com maior endereço (possivelmente nenhum filho) tenham mais chances de se tornarem raiz.

2.4.4 MANETCONF

No MANETconf [NES02], todos os nós são capazes de atribuir endereços. Cada um dos nós mantém uma tabela de alocação que contém endereços já alocados ou livres. Ao entrar na rede, o nó envia uma mensagem de anúncio à sua vizinhança. O primeiro nó que responder à mensagem é escolhido pelo novo nó para gerar seu possível endereço.

O novo nó A então requisita um endereço ao nó B. Este escolhe um endereço livre em sua tabela de alocação e o propaga através da rede inteira para ter a permissão de usá-lo. Se todos os nós da rede respondem positivamente, o nó B conclui que o endereço está livre e informa ao nó A. Se um ou mais nós respondem negativamente, o nó B conclui que o endereço está em uso e repete o procedimento com outro endereço. Deixando um ou mais nós de responderem, o nó B realiza um novo contato requisitando permissão. Não havendo resposta depois de algumas tentativas, o nó B informa a rede inteira da saída do(s) nó(s) não respondente(s). Se porventura o(s) nó(s) responde(m), o processo de configuração continua.

Cada nó tem um identificador que consiste de dois elementos: o menor endereço IP e um número único gerado pelo nó com o menor endereço IP. Quando uma partição ocorre na rede, um das duas redes mantém o identificador original e a outra terá que gerar seu novo identificador. Neste caso, um novo nó com o menor endereço IP terá que ser escolhido para gerar o identificador da rede e propagá-lo dentro da mesma.

Quando dois ou mais nós estão dentro do raio de comunicação um do outro, os mesmos trocam entre si seus identificadores de rede. Se um nó recebe um identificador que difere da identificação da rede à qual pertence, uma junção de redes é detectada. Quando isto ocorre, estas redes trocam suas diferentes tabelas de alocação permitindo a todos os nós atualizarem suas respectivas tabelas de alocação e detectarem localmente duplicação de endereço. Para cada duplicação de endereço, um dos nós conflitantes, no caso o nó com menor número de conexões TCP, deve gerar seu novo endereço.

2.5 ABORDAGENS HÍBRIDAS

As abordagens híbridas combinam características das abordagens *stateful* e *stateless* a fim de melhorar o desempenho total da rede. Em geral, usam uma ou mais tabelas de endereços para manter registro do estado da rede e também fazem uso de alguma técnica de detecção de conflito de endereço. Porém, a busca pela maior eficiência torna as abordagens híbridas mais complexas. Exemplos de abordagens híbridas são o PACMAN [WEN05] e HCQA [SUN03].

2.5.1 PACMAN

O PACMAN (do inglês *Passive Autoconfiguration for Mobile Ad Hoc Networks*) [WEN05] é uma abordagem para autoconfiguração de endereço distribuída que gera um tráfego de controle muito baixo, pois usa informação derivada de tráfego do protocolo de roteamento. São usados elementos de ambas as abordagens *stateful* e *stateless*. Como em abordagens *stateful*, os nós utilizam a informação do protocolo de roteamento para saber quais os endereços atribuídos na rede. Conflitos de endereço são detectados passivamente, baseando-se em anomalias no tráfego do protocolo de roteamento. Nesse caso, o PACMAN opera independentemente do tipo de protocolo de roteamento. Como em abordagens *stateless*, cada nó atribui a si mesmo um endereço IP, habilitando a compressão de endereço, o que contribui para a redução do tráfego do protocolo de roteamento.

PACMAN dá suporte ao tratamento de partições e junções de redes. Depois que acontece a junção de duas redes, conflitos de endereço podem acontecer. PACMAN usa PDAD para detectar endereços duplicados. Usando esta técnica, um nó observa o tráfego de controle proveniente do protocolo de roteamento e verifica se há conflito de endereço, não consumindo para isso largura de banda adicional.

2.5.2 HCQA

O HCQA (do inglês *Hybrid Centralized Query-based Autoconfiguration*) [SUN03] combina características do mecanismo *stateless* SDAD juntamente com uma abordagem de autoconfiguração de endereço centralizada. Um nó que deseja se juntar à rede escolhe dois endereços, endereço de tentativa e endereço temporário, e executa o processo SDAD descrito na subseção 2.3.1. Se a autoconfiguração foi bem-sucedida, o nó terá que registrar seu endereço de tentativa com um nó chamado *Address Authority (AA)*. Então, o nó requisitante espera por um anúncio de AA durante certo período de tempo e, ao recebê-lo, envia um pedido de registro de seu endereço de tentativa ao AA e espera por uma confirmação. Se a mensagem de confirmação é recebida, o nó pode começar a usar o endereço de tentativa. Depois de realizar seu registro com o AA, o novo nó executa um temporizador e reinicia o processo quando o temporizador expira.

Na inicialização da rede, o primeiro nó que obtém um endereço IP torna-se o AA na rede. O AA escolhe um identificador único (ex. seu endereço MAC) e propaga-o periodicamente para identificar a rede. Se não receber qualquer anúncio do AA por certo tempo, um nó constata que uma partição de rede aconteceu, torna-se o novo AA e gera um novo identificador único. Ao receber um novo identificador de rede, um nó deve registrá-lo com o novo AA, não sendo necessária nenhuma mudança de endereço. Quando a presença de dois identificadores de rede é verificada, uma junção de redes é detectada. Já que AA's armazenam o estado de todos os endereços IP atribuídos, os AA's envolvidos na junção detectada trocam suas tabelas de alocação para detectar possíveis conflitos de endereço.

O protocolo realiza *backup* da tabela de endereço do AA para reduzir a centralização neste nó. O AA considera o primeiro nó que registrou seu endereço como seu nó *backup* ou “*Address Authority Backup*”. Quando um nó registra seu endereço com o AA, este também envia uma atualização contendo a nova informação para o seu nó backup.

O mecanismo de detecção de endereço duplicado e o envio de mensagens periódicas pelo AA geram um alto tráfego de controle na rede. Além disso, a autoconfiguração de endereço é dependente de uma entidade central que requer o registro de todos os nós por meio de mensagens *unicast*, o que pode elevar a latência de configuração e a sensibilidade à perda de mensagens.

2.6 DISCUSSÃO

Este capítulo apresentou o conceito de alocação de endereços e sua aplicabilidade em cenários de emergência, com uso de redes *ad hoc* móveis para comunicação entre as equipes de resgate. Além disso, foram apresentadas as principais soluções de alocação de endereços para redes ad hoc móveis existentes na literatura, classificadas em abordagens *stateful*, *stateless* e híbridas. Estas soluções, contudo, não abrangem bem todos os aspectos relacionados à dinamicidade e imprevisibilidade das MANETs. Por exemplo, soluções baseadas no mecanismo SDAD não asseguram a unicidade dos endereços e introduzem tráfego de controle significativo na rede. As soluções PDAD e WDAD, por sua vez, requerem mudanças na camada de roteamento. O protocolo *Prophet* não garante a unicidade dos endereços alocados e somente é recomendado para redes de grande escala, ao passo que no *Prime* há uma grande dependência do nó raiz e os mecanismos de tratamento de partições e junções não são eficientes.

A Tabela 2.1 apresenta uma visão geral dos protocolos apresentados neste capítulo, destacando o tipo de abordagem ao qual pertencem, se garantem a unicidade dos endereços alocados ou utilizam algum mecanismo de detecção de endereço duplicado, bem como a latência de obtenção de endereço. Na Tabela 2.1, o período de sincronização, propagação de mensagens na rede ou qualquer procedimento repetitivo se existir é representado por T. d é o raio médio da rede em número de saltos. t é o tempo de ida e volta para uma comunicação de um salto. k é o número de iterações se existirem. l, D e s são a latência média de um salto, o diâmetro da rede em número de nós e o tempo de recuperação, respectivamente.

Protocolo	Abordagem	Unicidade de endereço	DAD	Latência
SDAD	*Stateless*	*Não garantida*	*Sim*	$k * T$
WDAD	*Stateless*	*Garantida com alta probabilidade*	*Sim*	0
PDAD	*Stateless*	*Garantida com alta probabilidade*	*Sim*	0
DCDP	*Stateful*	*Garantida*	*Não*	$O(2 * l)$
Prophet	*Stateful*	*Não garantida*	*Não*	$2 * t$
Prime DHCP	*Stateful*	*Garantida*	*Não*	$O(2 * l)$
MANETconf	*Stateful*	*Garantida*	*Não*	$(2 + d) * t$
PACMAN	*Híbrida*	*Garantida com alta probabilidade*	*Sim*	$O(2 * l * D * s)$
HCQA	*Híbrida*	*Garantida*	*Sim*	$k * T$

Tabela 2.1: Visão geral dos protocolos de alocação de endereço (adaptado de [WEHBI05])

3 GAAP

Como observado no Capítulo 2, as abordagens existentes têm vantagens e desvantagens. Das abordagens apresentadas, algumas se destacam por serem mais eficientes que outras. Tratam-se das abordagens *stateful* que não precisam manter a tabela de endereços completa em um ou mais nós, onde umas utilizam múltiplas tabelas de alocação disjuntas e outras fazem uso de função de alocação distribuída.

As abordagens *stateful* que usam função de alocação distribuída levam vantagem por garantirem menor latência na alocação de endereços e não precisarem inundar a rede com mensagens de sincronização para manter a consistência das tabelas de endereços. No entanto, técnicas com função de alocação distribuída em geral não utilizam mecanismos eficientes de recuperação de endereços e tratamento de partições e junções de forma a lidar com a chegada e saída de nós da rede, o que acaba gerando um tráfego de controle significativo. Por exemplo, o método *Prophet*, apesar de utilizar um mecanismo de tratamento de partições e junções eficiente, não garante a unicidade dos endereços alocados e é adequado apenas para redes de larga escala.

As abordagens *stateful* com múltiplas tabelas de alocação disjuntas, por sua vez, podem apresentar um elevado tráfego de controle e um atraso significativo na alocação de endereços, devido ao aumento do número de redirecionamento de requisições quando não há mais endereços disponíveis em alguns nós da rede. Na tentativa de solucionar esse problema, o método *Buddy* [MOH02] implementa um mecanismo onde o nó que não dispõe de endereços a oferecer busca em sua tabela de endereços pelo nó que possui o maior bloco de endereços livres na rede. Uma requisição é enviada ao nó com o maior bloco de endereços que responde com uma fatia de seu bloco de endereços. Apesar de reduzir o tráfego de controle gerado pelo número de redirecionamento de requisições, há um elevado custo de sincronização da tabela de endereços, visto que cada nó periodicamente propaga em *broadcast* sua tabela de endereços.

Este trabalho propõe um novo protocolo de alocação de endereços para redes *ad hoc* móveis utilizadas por equipes de resgate em cenários de emergência como catástrofes e desastres. A solução proposta, denominada GAAP (*Greedy*

Address Allocation Protocol), é dividida em duas partes: alocação e manutenção de endereços. Essas partes juntas visam atribuir endereços únicos aos nós da rede e manter a consistência do espaço de endereços durante o tempo de vida da rede. O protocolo proposto tem esse nome, pois adota uma estratégia "gulosa" (*greedy* em inglês) durante a alocação de endereço, onde o *nó requisitante*[3] escolhe o maior bloco dentre os blocos de endereços oferecidos pelos nós servidores da rede.

O GAAP busca alocar endereços únicos aos nós de forma a reduzir o tráfego de controle introduzido pela detecção de duplicação. Para manter a consistência do espaço de endereçamento durante o tempo de vida da rede, o protocolo realiza a recuperação de endereços perdidos e tratamento de partições e junções de redes quando necessários.

A solução proposta não é dependente de protocolo de roteamento e pode ser usada com protocolos proativos, reativos e híbridos. Além disso, a solução foi desenvolvida para redes IPv4, mas pode ser facilmente modificada para redes IPv6.

No processo de alocação de endereços do GAAP, quando não encontra nenhum vizinho, um nó inicia uma nova rede atribuindo a si mesmo um endereço e alocando um *bloco de endereços*[4] a serem usados pelos nós que se juntarem à rede. Este nó, ao receber uma requisição de endereço de um novo nó, entrega a este a metade do seu bloco de endereços e fica com a outra metade. O nó requisitante atribui a si o primeiro endereço do bloco recebido e os demais endereços ficam disponíveis para alocação. Assim, cada nó depois de configurado passa a atuar como servidor de endereços, alocando sempre metade do seu bloco de endereços atual aos nós requisitantes. Além disso, o custo de geração dos endereços é praticamente zero, pois o espaço total de endereços da rede constitui apenas um bloco de endereços consecutivos formado na inicialização da rede.

Além de atuar como servidor de endereços, um nó também participa do processo de manutenção de endereços. Este processo visa lidar com a saída abrupta de nós, desconexões temporárias, partições e junções de redes. Tais eventos podem levar a perdas ou conflitos de endereços. O GAAP usa um mecanismo para lidar com perdas no espaço de endereçamento chamado

[3] Utilizado para designar um nó que deseja se integrar à rede, sendo o responsável por disparar um processo de alocação ou recuperação de endereços.
[4] Também chamado de *pool* de endereços no restante do documento.

recuperação de endereços e outro mecanismo para tratar partições e junções a fim de evitar conflitos de endereços.

Nas próximas seções, os mecanismos de alocação e manutenção de endereços que compõem o GAAP são descritos em detalhes.

3.1 ALOCAÇÃO DE ENDEREÇOS

O processo de alocação de endereços do GAAP é relativamente simples. Como mencionado anteriormente, um endereço é composto de duas partes: prefixo de rede e identificador de *host*. O prefixo de rede é um valor definido previamente de forma estática e o identificador de *host* constitui o primeiro valor do bloco de endereços alocado a um nó em sua inicialização. Por conveniência, o GAAP adota o prefixo de rede 10.0.0.0/8 como valor padrão, apesar de nos experimentos realizados neste trabalho o espaço de endereçamento não ser superior a 256 endereços.

Quando um nó deseja se juntar a uma rede, o mesmo envia uma mensagem de descoberta de servidores chamada *Server Discovery* e aguarda resposta. Se não houver resposta, o nó repete o processo até certo número vezes. Não havendo resposta, o nó aloca um bloco de endereços de tamanho igual ao espaço de endereçamento da rede e atribui a si o primeiro valor do bloco como seu endereço. Durante sua configuração, o nó gera um número inteiro aleatório entre zero (0) e o maior número inteiro de 32 bits (2^{32}), que servirá como identificador único da rede, denominado *Network Identifier* (NetID). O nó permanece com o restante do bloco de endereços, passando a atuar como servidor de endereços.

Depois de configurado, o nó passa a enviar periodicamente mensagens em *broadcast* a um salto avisando de sua presença na rede. Tais mensagens são semelhantes às utilizadas em mecanismos de roteamento, conhecidas na literatura como mensagens *Hello* ou *keepalive*. Uma mensagem *Hello* armazena o endereço do nó que a originou bem como o NetID da rede. Quando um nó recebe uma mensagem *Hello* originária de sua própria rede, o mesmo atualiza a sua lista de endereços de vizinhos a um salto.

Eventualmente, mensagens *Hello* podem alcançar nós que não pertencem à rede. Se isso acontecer, o nó que recebeu a mensagem *Hello* pode requisitar a sua

entrada na rede enviando uma mensagem *Server Discovery* em *broadcast*. A Figura 3.1 descreve a troca de mensagens durante o processo de alocação de endereço. Depois de enviar uma mensagem *Server Discovery* (Msg. 1), o nó requisitante aguarda resposta. A mensagem enviada contém dentre outros parâmetros o identificador de rede recebido na mensagem *Hello*. Se a requisição encontra um servidor de endereços, este então encaminha ao nó requisitante uma mensagem *Pool Offer* (Msg. 2) contendo a primeira metade do seu bloco de endereços disponíveis para alocação. Se mais de um nó responde à requisição de um novo nó, então este pode receber mais de uma oferta de endereços e deve optar pela oferta de maior bloco de endereços e, em caso de empate, a oferta que contém o menor endereço. Assim como acontece com o primeiro nó da rede, os demais nós que recebem blocos de endereços tornam-se servidores.

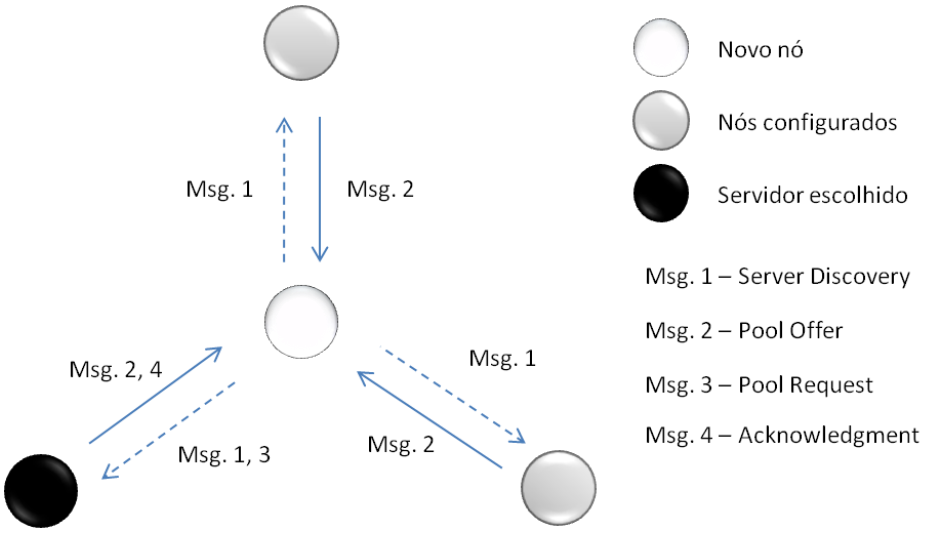

Figura 3.1: Fluxo básico de alocação de endereço no GAAP

Depois de escolher a oferta, o nó requisitante envia uma mensagem *Pool Request* (Msg. 3) ao nó ofertante, cujo endereço está presente na mensagem *Pool Offer* recebida. O nó escolhido compara a oferta presente na mensagem *Pool Request* com a última oferta por ele enviada. Se houver correspondência, o nó servidor encaminha uma mensagem *Acknowledgment* (Msg. 4) ao nó requisitante e em seguida atualiza o seu bloco de endereços. Ao receber a confirmação do nó servidor, o nó requisitante se configura utilizando o primeiro valor do bloco de endereços recebido como seu endereço e pode agora atuar como servidor dos endereços restantes do bloco. Se o nó requisitante não receber a mensagem de confirmação do nó servidor, o processo de alocação de endereço é reiniciado.

O processo de alocação de endereços descrito onde um nó requisitante obtém seu endereço a partir de vizinhos a um salto é chamado de *alocação de endereços local*.

Como exemplo, considere agora o esquema da Figura 3.2, onde o primeiro nó (A) de uma rede possui inicialmente um bloco de oito endereços, representado pelo intervalo [1-8]. O nó atribui a si mesmo o primeiro valor do intervalo (1) e o restante ([2-8]) fica disponível para alocação. Em seguida, um nó C que deseja se juntar à rede envia uma mensagem *Server Discovery* em *broadcast* de forma a alcançar os nós vizinhos a um salto. O nó A ao receber a mensagem originada pelo nó C, divide seu bloco de endereços ([2-8]) ao meio e encaminha uma metade ([2-5]) ao nó C em uma mensagem *Pool Offer* e então fica com a outra metade ([6-8]). Depois de escolher a oferta, o nó C envia uma mensagem *Pool Request* ao nó A. Este então encaminha uma mensagem *Acknowledgment* ao nó C, que atribui a si mesmo o primeiro valor (2) da oferta recebida ([2-5]) e passa a atuar como um servidor de endereços.

Semelhantemente, um nó B que deseja também se juntar à rede envia uma mensagem *Server Discovery* que busca alcançar os vizinhos a um salto. Ao receber a requisição de endereço do nó B, o nó A encaminha uma mensagem *Pool Offer* contendo o bloco de endereço ([6-7]), restando o endereço 8 para alocação. O nó C ao receber a requisição do nó B também envia uma oferta, que corresponde ao bloco de endereço ([3-4]), sobrando ainda o endereço 5. Como as ofertas recebidas têm o mesmo tamanho de bloco, o nó B opta pela oferta ([3-4]) que contém o menor endereço (3). O nó B então envia uma mensagem *Pool Request* ao nó A, responsável pelo envio da oferta escolhida. Este nó então encaminha uma mensagem *Acknowledgment* ao nó B, que atribui a si mesmo o primeiro valor (3) da oferta recebida ([3-4]), restando o endereço 4. A técnica de divisão de blocos de endereço ao meio, empregada pelo GAAP, é semelhante à usada em [MOH02].

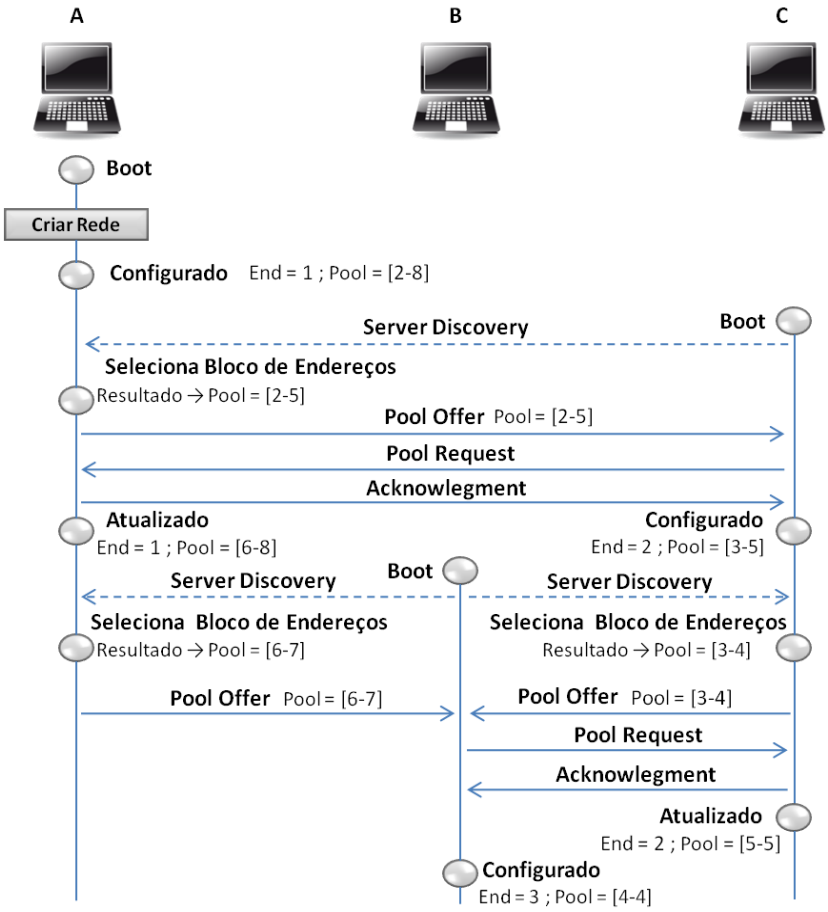

Figura 3.2: Exemplo de alocação de endereços local para três nós no GAAP

No GAAP, se um nó após três tentativas não consegue obter endereço devido ao fato de seus vizinhos a um salto não terem endereços a oferecer, o mesmo inicia um mecanismo de descoberta remota buscando servidores de endereços que estejam a mais de um salto. Nós que não tem endereços a oferecer passam a atuar como um *proxy*, encaminhando as mensagens do novo nó para o restante da rede e vice-versa. Esse processo é chamado de *alocação de endereços remota*, ilustrado na Figura 3.3.

Mais especificamente, após enviar uma mensagem de *broadcast Server Discovery* e não obter resposta, o nó requisitante (nó A) procede com a seguinte sequência de mensagens, conforme mostrado na Figura 3.3:

1. Envia uma mensagem de *broadcast Proxy Request* (Msg. 1) para todos os seus vizinhos (nós B, C, D e E);
2. O vizinho ao receber a mensagem e tiver adquirido novos endereços, responde com uma mensagem *Pool Offer* endereçada ao nó requisitante, caso contrário atua como *proxy* (nós D e E) encaminhando uma

mensagem de *broadcast Remote Server Discovery* (Msg. 2) para o restante da rede;

3. Ao receber a mensagem *Remote Server Discovery*, cada nó reencaminha a mensagem aos seus vizinhos ou responde ao *proxy* com uma mensagem *Remote Pool Offer* (Msg. 3) se possuir uma oferta (nós G e I);
4. O *proxy* encaminha as ofertas diretamente ao novo nó em mensagens *Proxy Pool Offer* (Msg. 4), que escolhe a oferta com o maior bloco de endereços;
5. O nó requisitante então encaminha uma mensagem *Proxy Pool Request* (Msg. 5) ao *proxy* do qual recebeu a oferta escolhida (nó D). Este, por sua vez, encaminha a requisição ao servidor que teve sua oferta escolhida em uma mensagem *Remote Pool Request* (Msg. 6);
6. O nó servidor escolhido responde ao *proxy* com uma mensagem *Remote Pool Reply* (Msg. 7);
7. Por fim, o *proxy* encaminha em uma mensagem *Proxy Pool Reply* (Msg. 8) a oferta ao novo nó que pode então se configurar.

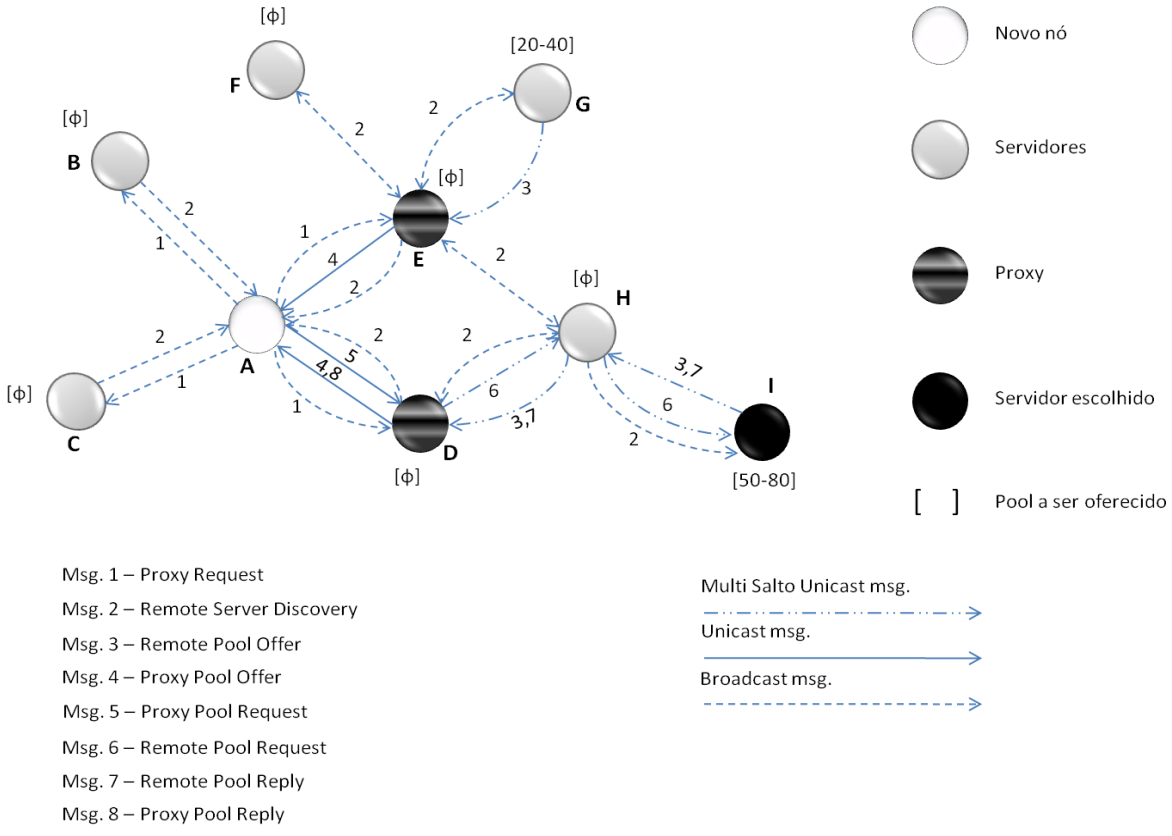

Figura 3.3: Exemplo de alocação de endereços remota no GAAP

3.2 MANUTENÇÃO DE ENDEREÇOS

O processo de manutenção de endereços permite aos nós lidarem com desconexões de rede e seus efeitos. Esse processo engloba três procedimentos: recuperação de endereços, tratamento de partições e junções de redes. O primeiro procedimento ocorre quando novos nós detectam a existência da rede, mas, mesmo depois do mecanismo de descoberta remota, não conseguem obter endereços. A busca mal-sucedida pode ser resultante da rede ter atingido o seu número máximo de dispositivos ou endereços terem se perdido devido a qualquer outro evento que impeça os nós de enviarem ou receberem mensagens (saída dos nós do alcance da rede, falha de hardware/software em nós, dentre outros). Nessa etapa, endereços que não estiverem em uso devem ser recuperados. O segundo e terceiro procedimentos visam tratar possíveis conflitos que surgem quando partições ou junções de redes ocorrem. A seguir, esses procedimentos são descritos em detalhes.

3.2.1 RECUPERAÇÃO DE ENDEREÇOS

Como comentado no Capítulo 2, um dos maiores problemas de gerenciamento do espaço de endereços é saber identificar um endereço perdido e também o melhor momento de recuperá-lo. Depois de detectar uma perda no espaço de endereços, se a recuperação for realizada imediatamente, há o risco do antigo dono do endereço perdido retornar à rede e ter que iniciar uma nova requisição de endereço gerando tráfego de controle na rede. Por outro lado, se a detecção ou o processo de recuperação atrasar, um novo nó pode não obter endereço se a rede não possuir endereços disponíveis. No GAAP os endereços são recuperados somente quando não houver mais endereços disponíveis na rede e um novo nó que está se juntando à rede precisar de endereço.

No melhor caso, um nó pode propagar uma mensagem pela rede informando sua saída e então seu bloco de endereços é recuperado. Tal mensagem, chamada *Release Pool*, contém o endereço do nó que ofertou o bloco de endereços. Quando o nó servidor do bloco de endereços recebe a mensagem, o referido bloco é recuperado. Caso o nó servidor tenha saído da rede, nenhuma recuperação de

endereços é realizada. No GAAP, a mensagem *Release Pool* não é esperada ou mesmo obrigatória, mas acelera o processo de recuperação e quase não introduz tráfego de controle na rede.

Para cobrir os casos mais comuns em que os nós deixam a rede sem qualquer aviso, a rede tenta detectar os blocos de endereços não ocupados e redistribuí-los. O processo inicia quando um novo nó detecta a existência de uma rede, mas mesmo depois de executar o mecanismo de descoberta remota não consegue obter endereço. A existência da rede é detectada através do uso de mensagens *Hello*, que são enviadas periodicamente por todos os nós configurados.

Conforme a Figura 3.4, quando um nó deseja se juntar à rede e não obtém nenhuma oferta depois de executado o processo de alocação de endereço, uma nova rede é criada se o nó requisitante não detectou a existência de outra rede. Por outro lado, se um nó requisitou endereço para uma rede já existente e não obteve nenhuma oferta, o mesmo inicia a recuperação de endereços com o envio de uma mensagem *Pool Reclamation Request*. Esta mensagem contém um número de sequência, gerado pelo nó requisitante, e o identificador da rede recebido dentro de mensagens *Hello*. Ao receber a requisição, cada nó que tem exatamente o mesmo identificador da rede cria uma mensagem *Pool Reclamation*, contendo os mesmos campos da mensagem *Pool Reclamation Request* mais seu próprio endereço, e propaga a mensagem na rede. O nó também envia ao nó requisitante uma mensagem *Pool Reclamation Reply* contendo seu endereço e o número de sequência da mensagem *Pool Reclamation Request* recebida.

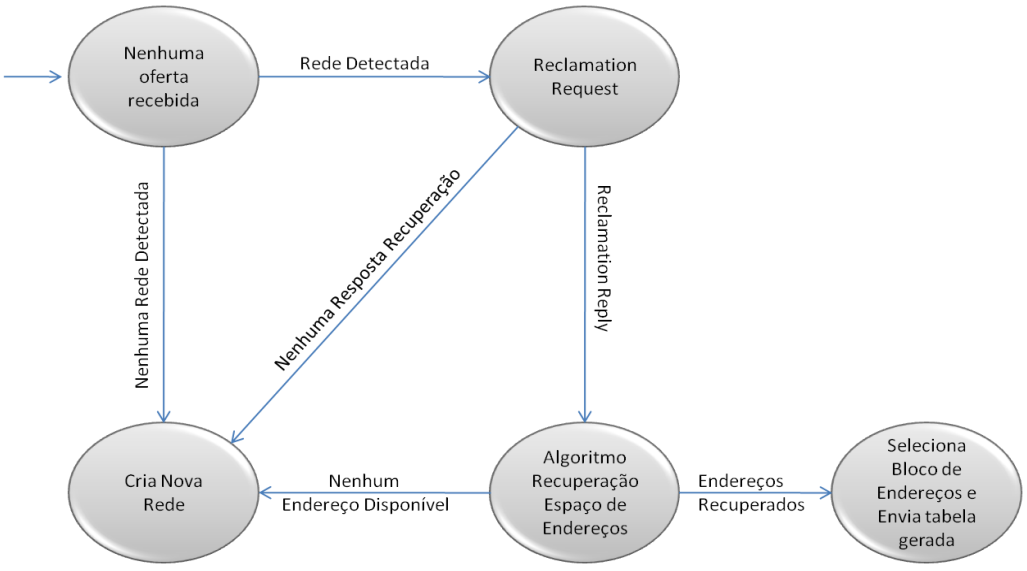

Figura 3.4: Recuperação de endereços no GAAP

Ao receber a mensagem *Pool Reclamation*, cada nó verifica se já a processou, observando o número de sequência. Se não foi processada ainda, uma mensagem *Pool Reclamation Reply* é enviada para o endereço presente na mensagem *Pool Reclamation* e esta é então propagada pela rede. Enquanto as mensagens *Pool Reclamation Reply* são recebidas, as mesmas são encaminhadas ao novo nó.

Depois de receber todas as mensagens *Pool Reclamation Reply* ou um tempo máximo de espera se esgotar, o nó requisitante passa a executar um algoritmo de recuperação de endereços descrito abaixo:

1. Uma tabela é criada contendo uma entrada para o nó requisitante e cada nó que respondeu à requisição de recuperação de endereços com uma mensagem *Pool Reclamation Reply*;
2. Para cada nó na tabela são definidos novos blocos de endereços. Os blocos têm tamanhos iguais obtidos a partir da divisão do espaço total de endereços pelo número total de nós. Por exemplo, se o espaço de endereços utilizáveis é igual a 256 e o número total de nós é 8, cada nó da tabela terá blocos de endereços de tamanho 32.
3. A tabela é preenchida com os blocos de endereços de cada nó, começando pelo nó requisitante. Se o tamanho do bloco de endereços é 8 e o endereço inicial do espaço de endereços é 1, o nó requisitante ficará com o bloco [1-8], o segundo nó terá o bloco [9-16] e assim por diante.

Após gerar a tabela com novos blocos de endereços de cada nó, o nó requisitante seleciona o seu bloco de endereços correspondente. Para completar a sua configuração, o nó usa o prefixo e máscara de rede disponíveis nas mensagens enviadas pelos primeiros nós que responderam à requisição de recuperação. Após se autoconfigurar, o nó requisitante envia a tabela gerada à MANET juntamente com o novo identificador da rede também gerado pelo nó.

A etapa final do processo ocorre quando os nós da rede recebem a tabela de endereços criada. Os nós verificam se estão presentes na tabela e, em caso negativo, que pode ser causado por mensagens perdidas, liberam seus endereços e iniciam o processo de alocação novamente. Caso contrário, o nó seleciona o seu bloco de endereços correspondente na tabela e se reconfigura com o primeiro endereço do bloco. Após o término do processo, cada nó da tabela deve estar

reconfigurado e com seu bloco de endereços, sendo capaz de atender a requisições de novos nós.

Apesar do processo de recuperação de endereços gerar um tráfego de controle elevado e alta latência de configuração do nó requisitante e requerer que todos os nós da rede se reconfigurem, é esperado que isso ocorra apenas quando um nó requisitante não obtiver nenhuma oferta dos nós servidores.

3.2.2 Tratamento de Partições e Junções de redes

Partições e junções de rede são comuns em MANETs, tendo como principal causa a entrada e saída repentina de nós da rede, devido principalmente à mobilidade dos nós. Como mencionado no Capítulo 2, a junção de duas redes pode levar a conflitos de endereços. Para evitar esse problema, o GAAP inclui um mecanismo que detecta quando uma junção ocorre e determina o que fazer para manter a unicidade dos endereços.

Partições de rede por si só não levam a qualquer conflito de endereço. No melhor caso, redes podem particionar e se unirem novamente sem qualquer problema. No entanto, redes que passaram pelo processo de recuperação de endereços e se unem novamente podem apresentar conflitos de endereços. No GAAP, o processo de recuperação de endereços atribui um novo identificador à rede, de maneira que partições que passaram pela recuperação de endereços são tratadas como redes distintas.

No tratamento de partições e junções de rede, além do identificador da rede, é utilizado o tamanho estimado da MANET. Essa estimativa é feita com base na informação do número de vizinhos diretos armazenada em cada nó. Tal informação é propagada pelo nó em mensagens *Hello*. Quando um nó recebe uma mensagem *Hello*, é armazenado o endereço do nó que originou a mensagem bem como a informação do número de vizinhos diretos do mesmo. De forma a manter atualizada a “foto” da rede, cada vez que uma mensagem *Hello* é enviada, o nó emissor calcula o tamanho aproximado da MANET. Este tamanho consiste na soma do número de vizinhos diretos com o número de vizinhos informado por cada um dos vizinhos diretos detectados. Apesar de o valor aproximado em geral ser maior do que o valor real, o tamanho obtido ainda é uma boa estimativa, principalmente no cenário mais comum onde poucos nós se juntam para formar uma rede maior.

Quando uma mensagem *Hello* é recebida, o nó verifica se o identificador de rede reportado é igual ao seu. Em caso afirmativo, o nó atualiza a tabela de vizinhos para estimar o tamanho da rede e também ajudar no processo de recuperação de endereços descrito na subseção 3.2.1. Se os identificadores de rede são diferentes, o nó compara o tamanho de sua rede com o tamanho reportado. O nó liberará seu endereço e requisitará um novo à rede descoberta, se o tamanho de sua rede for menor ou igual ao tamanho reportado.

Na estratégia utilizada, redes com maior número de nós são privilegiadas, de forma que a rede que detectou a junção precisa realizar menos mudanças de endereço por ser menor que a outra rede, gerando assim menor tráfego de controle. Além disso, as junções são tratadas uma a uma em vez de liberar ao mesmo tempo todos os endereços de uma das redes, o que pode gerar um elevado tráfego de controle, aumentando a probabilidade de perda ou corrompimento de mensagens e afetando o desempenho e estabilidade da rede.

Há um caso especial em que ambas as redes têm tamanho estimado igual a zero. Nessa situação, em vez de enviar uma mensagem *Server Discovery* para uma rede específica, usando o identificador da rede, o nó libera seu endereço e inicia nova requisição sem nenhuma restrição. Se um nó está sozinho, o mesmo poderá se integrar a qualquer rede que responder à requisição de endereço. Apesar da não restrição para este caso em específico, é importante, se possível, que os mecanismos de alocação local e remota sejam disparados para redes específicas, evitando que um nó ao deixar a rede e entrar em outra acabe retornando à rede anterior.

3.3 DESCRIÇÃO DAS MENSAGENS

A seguir é descrito o conjunto de mensagens que compõem o GAAP, com detalhamento dos campos e suas funções. As mensagens estão organizadas em três grupos:

(a) Mensagens de alocação de endereço;

(b) Mensagens de recuperação de endereços;

(c) Mensagens periódicas.

3.3.1 MENSAGENS DE ALOCAÇÃO DE ENDEREÇO

São mensagens propagadas pela rede durante o processo de alocação de endereço.

3.3.1.1 Server Discovery

É originada por um nó que deseja se integrar à rede e propagada em *broadcast* para os nós vizinhos a um salto. Caso não receba oferta de endereço de algum nó servidor, o nó requisitante repete o processo até três vezes. A mensagem é composta apenas pelo campo *netID* (32 bits), o qual identifica unicamente a rede de destino da requisição de endereço. O campo pode assumir dois valores possíveis:

- Valor zero: se o nó requisitante não detecta nenhuma rede antes de propagar a mensagem, o campo assume o valor zero (padrão);
- *netID* da rede detectada: o campo assume o valor do *netID* da rede detectada.

3.3.1.2 Pool Offer

É originada por um nó servidor em resposta a uma mensagem de requisição de endereço *Server Discovery.* A mensagem carrega uma oferta local e se destina ao nó originador da mensagem *Server Discovery* Os seguintes campos compõem a mensagem:

- *linf* (32 bits): limite inferior do *pool* de endereços ofertado;
- *lsup* (32 bits): limite superior do *pool* de endereços ofertado.

Por exemplo, quando um nó servidor oferta o *pool* de endereços [1-8], os valores dos campos da mensagem são *linf*=1 e *lsup*=8.

3.3.1.3 Pool Request

É originada pelo nó requisitante depois de enviar uma mensagem *Server Discovery* e receber pelo menos uma mensagem *Pool Offer* dentro de certo intervalo de tempo. Cada mensagem *Pool Offer* contém uma oferta de *pool* de endereços. A mensagem *Pool Request* é destinada ao nó originador da mensagem *Pool Offer* que contém o *pool* de endereços escolhido. A mensagem *Pool Request* contém os seguintes campos:

- *linf* (32 bits): limite inferior do *pool* de endereços escolhido;
- *lsup* (32 bits): limite superior do *pool* de endereços escolhido;
- *netID* (32 bits): identificação da rede à que pertence o nó servidor do *pool* de endereços escolhido.

3.3.1.4 Pool Reply

É uma mensagem originada pelo nó servidor do *pool* de endereços escolhido em uma requisição de endereço, como resposta a uma mensagem *Pool Request* proveniente de um nó que deseja se juntar à rede. Os seguintes campos compõem a mensagem:

- *netPrefix* (32 bits): é o prefixo da rede ao que o nó servidor pertence;
- *freeBits* (8 bits): representa a quantidade de bits disponíveis na formação do endereço a ser ofertado a um nó que deseja se integrar à rede. No GAAP, *freeBits* está configurado para o valor 16 por padrão. Por exemplo, para uma rede com endereço 10.0.0.0, os dois últimos octetos podem assumir valores de 0 a 254. Portanto, o espaço de endereços válidos é 10.0.0.1 a 10.0.254.254;
- *netSize* (32 bits): corresponde ao tamanho da rede à qual o nó servidor pertence;
- *netID* (32 bits): representa a identificação única da rede à qual o nó servidor pertence.

3.3.1.5 Remote Server Discovery

É originada por um nó que deseja se integrar à rede, depois de se esgotarem as três tentativas locais de obter endereço por meio da propagação de mensagens *Server Discovery*. A mensagem *Remote Server Discovery* é propagada em *broadcast* por toda a rede e não somente aos vizinhos a um salto como a mensagem *Server Discovery*. Caso não receba oferta de endereço de algum nó servidor, o nó requisitante repete o processo até três vezes.

A mensagem é composta pelos seguintes campos:

- *netID* (32 bits): identifica unicamente a rede de destino da requisição de endereço. O campo assume o valor zero se o nó requisitante não detecta nenhuma rede antes de propagar a mensagem. Caso contrário, é atribuído ao campo o valor do *netID* da rede detectada;

- *seqNumber (32 bits):* identifica unicamente a mensagem originada no nó requisitante. A informação do campo é útil para controle dos encaminhamentos na rede, evitando que os nós intermediários reencaminhem a mensagem mais de uma vez, o que geraria tráfego de controle desnecessário e possíveis *loops* de roteamento.

3.3.1.6 Remote Pool Offer

É originada por um nó servidor ao receber uma mensagem de requisição de endereço *Remote Server Discovery.* A mensagem carrega uma oferta remota e se destina ao nó originador da mensagem *Remote Server Discovery*. Os seguintes campos compõem a mensagem:

- *linf* (32 bits): limite inferior do *pool* de endereços ofertado;
- *lsup* (32 bits): limite superior do *pool* de endereços ofertado;
- *server* (32 bits): endereço IP do nó servidor;
- *req* (32 bits): endereço IP do nó requisitante.

3.3.1.7 Remote Pool Request

É originada pelo nó requisitante depois de enviar uma mensagem *Remote Server Discovery* e receber pelo menos uma mensagem *Remote Pool Offer* dentro de certo intervalo de tempo. Cada mensagem *Remote Pool Offer* contém uma oferta de *pool* de endereços. A mensagem *Remote Pool Request* é destinada ao nó originador da mensagem *Remote Pool Offer* que contém o *pool* de endereços escolhido. A mensagem *Remote Pool Request* contém os seguintes campos:

- *linf* (32 bits): limite inferior do *pool* de endereços ofertado;
- *lsup* (32 bits): limite superior do *pool* de endereços ofertado;
- *server* (32 bits): endereço IP do nó servidor;
- *req* (32 bits): endereço IP do nó requisitante;
- *netID* (32 bits): identifica unicamente a rede à que o nó servidor pertence.

3.3.1.8 Remote Pool Reply

É uma mensagem originada pelo nó servidor do *pool* de endereços escolhido em uma requisição de endereço, como resposta a uma mensagem *Remote Pool Request* proveniente de um nó que deseja se juntar à rede. Os seguintes campos compõem a mensagem:

- *netPrefix* (32 bits): é o prefixo da rede à que o nó servidor pertence;
- *freeBits* (8 bits): representa a quantidade de bits disponíveis na formação do endereço a ser ofertado a um nó que deseja se integrar à rede;
- *netSize* (32 bits): corresponde ao tamanho da rede à que o nó servidor pertence;
- *netID* (32 bits): representa a identificação única da rede à que o nó servidor pertence;
- *req* (32 bits): endereço IP do nó originador da mensagem *Remote Pool Request* recebida.

3.3.1.9 Relay Server Discovery

É originada por nós intermediários quando recebem uma mensagem *Remote Server Discovery* que contém o mesmo *netID* da rede à qual pertencem ou *netID* igual a zero. A mensagem é propagada por nós intermediários apenas uma vez, buscando alcançar algum nó servidor de endereços.

- *seqNumber* (32 bits): tem o mesmo valor do campo correspondente na mensagem *Remote Server Discovery* recebida. A informação do campo é útil para controle dos encaminhamentos na rede, evitando que os nós intermediários reencaminhem a mensagem mais de uma vez, o que geraria tráfego de controle desnecessário e possíveis *loops* de roteamento;
- *requestAddr* (32 bits): endereço IP do nó originador;
- *relayAddr* (32 bits): endereço IP do ultimo nó intermediário que encaminhou a mensagem. Neste caso, o campo assume o endereço IP do nó originador.

3.3.2 MENSAGENS DE RECUPERAÇÃO DE ENDEREÇOS

3.3.2.1 Reclamation Request

Trata-se de uma requisição de recuperação de endereços originada por um nó que deseja se integrar à rede, depois de esgotadas as tentativas locais e remotas para obtenção de endereço. A mensagem é propagada em *broadcast* pela rede e contém os seguintes campos:

- *netID* (32 bits): identificação da rede à qual o nó originador pertence;
- *seqNumber* (32 bits): identifica unicamente a mensagem gerada pelo nó. A informação do campo é útil para controle dos encaminhamentos na rede, evitando que os nós intermediários reencaminhem a mensagem mais de uma vez, o que geraria tráfego de controle desnecessário e possíveis *loops* de roteamento;
- *requestAddr* (32 bits): endereço IP do nó originador.

3.3.2.2 Relay Reclamation Request

É originada por nós intermediários quando recebem uma mensagem de requisição de recuperação *Reclamation Request* que contém o mesmo *netID* da rede à qual pertencem. A mensagem é propagada por nós intermediários apenas uma vez, buscando alcançar todos os nós da rede. Os seguintes campos compõem a mensagem:

- *netID* (32 bits): identificação da rede à qual o nó originador pertence;
- *seqNumber* (32 bits): tem o mesmo valor do campo equivalente da mensagem *Reclamation Request* recebida. A informação do campo é útil para controle dos encaminhamentos na rede, evitando que os nós intermediários reencaminhem a mensagem mais de uma vez, o que geraria tráfego de controle desnecessário e possíveis *loops* de roteamento;
- *relayAddr* (32 bits): endereço IP do ultimo nó intermediário que encaminhou a mensagem. Neste caso, o campo assume o endereço IP do nó originador;

- *requestAddr* (32 bits): endereço IP do nó requisitante do processo de requisição de recuperação de endereços, presente no campo equivalente da mensagem *Reclamation Request* recebida.

3.3.2.3 Reclamation Reply

É originada por um nó intermediário ao receber uma mensagem de requisição de recuperação de endereços *Reclamation Request.* A mensagem é destinada ao originador da mensagem *Reclamation Request* recebida. Os seguintes campos compõem a mensagem:

- *seqNumber*: tem o mesmo valor do campo equivalente da mensagem *Reclamation Request* recebida. A informação do campo é útil para controle dos encaminhamentos na rede, evitando que os nós intermediários reencaminhem a mensagem mais de uma vez, o que geraria tráfego de controle desnecessário e possíveis *loops* de roteamento;
- *addr*: endereço IP do nó originador.

3.3.2.4 Reclamation Inform

É originada pelo nó requisitante do processo de recuperação de endereços após esperar certo intervalo de tempo pelo recebimento de mensagens *Reclamation Reply*. Antes de propagar a mensagem pela rede, o nó requisitante gera uma tabela com os novos *pools* de endereços de cada um dos nós cujas mensagens *Reclamation Reply* chegaram. Os seguintes campos compõem a mensagem:

- *src* (32 bits): endereço IP do nó originador;
- *newNetID* (32 bits): nova identificação da rede gerada pelo processo de recuperação de endereços;
- *netPrefix* (32 bits): prefixo da rede;
- *freeBits* (8 bits): quantidade de bits disponíveis para formação dos endereços dos nós da rede;
- *currentNodesNumber* (8 bits): número de nós cujas respostas à requisição de recuperação de endereços chegaram ao nó requisitante;
- *poolSize* (32 bits): tamanho médio dos *pools* de endereços dos nós cujas respostas à requisição de recuperação de endereços chegaram ao nó requisitante;

- *currentNodes* ((32 * currenNodesNumber) bits): lista de nós cujas respostas à requisição de recuperação de endereços chegaram ao nó requisitante;
- *newAddrs* ((64 * currentNodesNumber) bits): lista de nós cujas respostas à requisição de recuperação de endereços chegaram ao nó requisitante, juntamente com o novo *pool* de endereços de cada nó.

3.3.3 MENSAGENS PERIÓDICAS

3.3.3.1 Hello

É enviada periodicamente por cada nó configurado na rede. A mensagem carrega informação de vizinhança do nó originador e é composta dos seguintes campos:

- *netID* (32 bits): identificação da rede à qual o nó originador pertence;
- *netSize* (32 bits): armazena informação do tamanho da rede à qual o nó originador pertence;
- *neighNumber* (32 bits): número de vizinhos a um salto do nó originador.

4 EXPERIMENTOS E RESULTADOS

O GAAP foi implementado em um ambiente de simulação para verificar suas funcionalidades e avaliar o seu desempenho. Este capítulo apresenta a metodologia usada para avaliar o desempenho do protocolo GAAP. Primeiramente, são apresentadas algumas características-chave do ambiente de simulação escolhido. Em seguida, os cenários e métricas de avaliação usados para avaliar o comportamento e o desempenho do protocolo são apresentados e discutidos. Por fim, são discutidos os resultados dos experimentos com o GAAP em comparação com outras abordagens *stateful,* a saber, *Prime* e *Prophet*, bastante conhecidas na literatura.

4.1 SIMULAÇÃO

Simulações têm sido largamente utilizadas pela comunidade científica na realização de experimentos. Montar uma infraestrutura real para execução de experimentos demanda mais recursos e tempo em comparação com o uso de simulação. Desta forma, simuladores possibilitam a configuração e execução de diversos cenários de maneira rápida e eficiente.

O simulador adotado neste trabalho foi o *Network Simulator* 3 [NSNAM06], mais conhecido como NS-3. Trata-se de um simulador de rede de eventos discretos utilizado como foco primário para fins de pesquisa e ensino. Pode se tornar eventualmente o substituto para o popular simulador NS-2 [FAL09], e tem grande foco no realismo, produzindo modelos próximos do mundo real e fáceis de validar. O NS-3 foi desenvolvido na linguagem de programação C++ usando boas práticas de engenharia de software, e está sob a licença GNU de software livre, o que permite os usuários contribuírem com o desenvolvimento e solução de problemas no código-fonte do simulador.

Neste trabalho, as simulações foram feitas em um notebook Dell *Inspiron* 1525 com sua configuração padrão, utilizando a distribuição Ubuntu 10.04 do sistema operacional Linux, com versão 2.6.32-24 do *kernel*. Neste trabalho, os protocolos avaliados tiveram sua implementação realizada no simulador NS-3.

4.2 CENÁRIOS

O desempenho dos protocolos foi avaliado em dois tipos de cenário de simulação. O primeiro deles é o cenário estático, onde os nós estão parados. A principal utilidade do cenário estático é permitir avaliar a funcionalidade dos protocolos sem grandes variações de comportamento da rede. Já o cenário dinâmico representa uma rede de emergência onde os nós estão em constante movimento. Esse cenário permite uma avaliação sob condições adversas da rede, sendo de comportamento mais próximo da realidade.

Em ambos os cenários, cada nó possui uma interface de rede sem fio Wi-Fi com alcance de 50 metros e taxa de transmissão de 1 Mbps, podendo representar qualquer dispositivo móvel e portátil, como *laptop*, *netbook, tablet,* dentre outros. O atraso e perda de propagação são simulados usando dois modelos existentes no NS-3, a saber, *Constant Speed Propagation e Friis Propagation Loss*. O protocolo de roteamento OLSR [JAC03], já implementado no NS-3, foi adotado com seus parâmetros padrão.

Nas simulações, foram definidos parâmetros para os protocolos analisados. A Tabela 4.1 mostra os valores dos parâmetros comuns aos três protocolos.

Parâmetro	Valor
Número de tentativas	3
Tempo de espera	0.5s
Mensagens periódicas	5s
Espaço de endereçamento	254

Tabela 4.1: Tabela de parâmetros comuns aos protocolos avaliados

O parâmetro *Número de tentativas* representa o número de tentativas de um nó em busca de um servidor de endereços. Mais precisamente, no protocolo proposto as tentativas locais e remotas foram configuradas para o valor 3, significando que um nó busca três vezes por servidores locais antes de realizar outras três tentativas por servidores remotos. O *Tempo de espera* corresponde ao tempo que um nó espera por uma resposta após o envio de uma mensagem. No GAAP, o tempo de espera por respostas locais foi configurado para 0.5 segundo, ao

passo que o aguardo por respostas remotas foi configurado inicialmente para 2 * *Tempo de espera* e é dobrado para tentativas consecutivas. Quanto ao parâmetro *Mensagens periódicas*, representa o intervalo de tempo entre envio de mensagens *Hello*, denominadas de *Recycle* no protocolo *Prime*. Por fim, o espaço de endereçamento escolhido permite que sejam atribuídos aos nós até 254 endereços, distribuídos na faixa de 10.0.0.1 a 10.0.0.254, excluindo-se o endereço da rede (10.0.0.0).

4.2.1 Cenário Estático

Este cenário permite avaliar o comportamento dos protocolos sob baixa variação das condições da rede. É o cenário utilizado para observar a eficiência, corretude e escalabilidade dos protocolos em um ambiente onde os nós estão parados.

A topologia da rede segue uma estrutura chamada grade espiral, ilustrada na Figura 4.1. O primeiro nó da rede inicia-se no centro e os demais entram na rede formando uma espiral. Optou-se por tal topologia, pois permite a um nó alcançar vários vizinhos a um salto de distância.

O alcance dos nós foi configurado para 50 metros, que é a mesma distância entre nós diagonalmente separados. O primeiro nó inicia-se no instante zero e os nós subsequentes iniciam de forma ordenada respeitando uma variável aleatória uniforme distribuída entre 0.5 e 4.5 segundos. Tal atraso é calculado somente depois do primeiro nó ter iniciado, o que evita dois nós de iniciarem ao mesmo tempo. Além disso, o tamanho do intervalo da variável uniforme foi escolhido baseando-se no tempo médio de configuração dos nós. No GAAP, como o primeiro nó leva mais tempo para criar a rede em comparação com os protocolos *Prime* e *Prophet*, o valor da variável aleatória uniforme recebida pelo segundo nó é acrescido de sete segundos em vez de cinco segundos. Conforme a Figura 4.1, a distância entre os nós 7 e 21 é 50 metros, o que nos permite concluir que o nó 7 alcança os nós 20, 21, 22, 23, 8, 1, 6 e 19. A Figura 4.2 mostra as quantidades possíveis de vizinhos de um nó dependendo de sua localização dentro da estrutura de grade espiral, podendo ser três (nó 48), cinco (nó 39) ou oito vizinhos (nó 12).

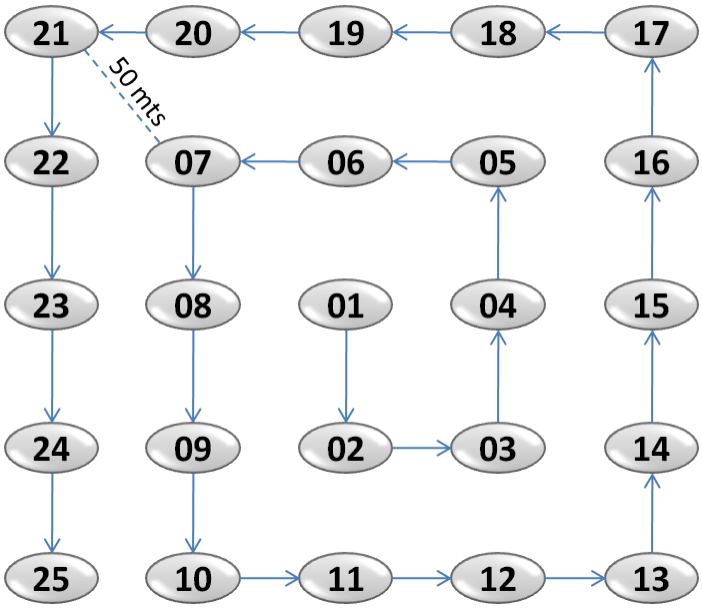

Figura 4.1: Estrutura de grade espiral

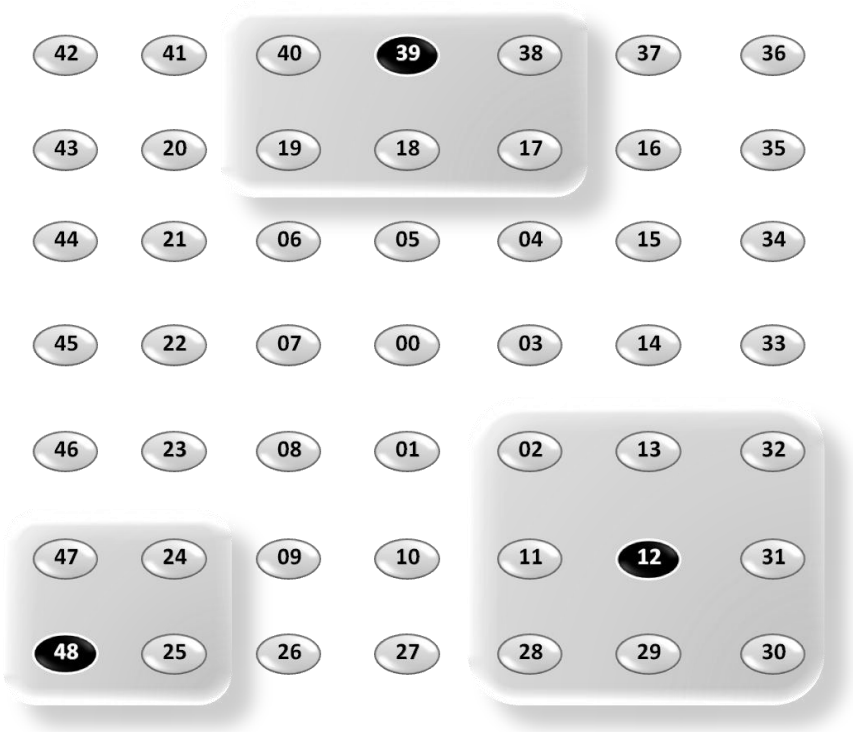

Figura 4.2: Vizinhança na estrutura de grade espiral

No cenário estático não há tratamento de partições e junções de redes, pois os nós não se movimentam. O foco principal do cenário estático é permitir a avaliação das funcionalidades básicas dos protocolos. Todos os nós não podem criar uma nova rede, com exceção do primeiro nó. Assim, quando os nós não

encontram um servidor de endereço, em vez criarem uma nova rede ou iniciarem o processo de recuperação de endereços, um erro de configuração é computado.

Foram coletadas amostras para redes com 1, 4, 9, 16, 25, 36, 49, 64, 81 e 100 nós. Os tamanhos de rede foram escolhidos buscando a correta formação da grade espiral. Tais valores são quadrados perfeitos, resultando em uma forma geométrica com número igual de nós nos quatro lados. Nas simulações o tamanho do espaço de endereçamento foi configurado para ser o menor possível, sendo igual ao número de nós, de forma a avaliar o impacto do espaço de endereçamento no desempenho dos protocolos. Considerando que um novo nó leva em média dois segundos para se configurar, são necessários aproximadamente 200 segundos para que uma rede com 100 nós esteja completamente configurada. Assim, a duração das simulações foi fixada em 300 segundos, o que representa um tempo suficiente para todos os nós obterem um endereço ou falharem no processo.

4.2.2 CENÁRIO DINÂMICO

Este cenário foi definido para permitir a avaliação dos protocolos sob uma perspectiva mais realista, onde os nós se movimentam livremente. No cenário dinâmico, ao contrário do cenário estático, partições e junções de redes são possíveis.

No cenário dinâmico, não foi adotada a estrutura de grade espiral do cenário estático, já que os nós se movimentam livremente. Em vez disso, todos os nós iniciam no centro de uma região quadrada, de forma a simular um cenário de emergência onde equipes de resgate partem de uma base de apoio em direção a regiões afetadas por desastres.

O cenário dinâmico é representado por dois ambientes móveis. O cenário A, de maior densidade de nós, é usado para representar um cenário de emergência onde as equipes de resgate se deslocam com velocidade máxima menor para atender uma área afetada de menor tamanho, por exemplo, a área alagada de um bairro residencial atingido por uma enchente. O cenário B, de menor densidade de nós, representa um cenário de emergência onde as equipes de resgate se deslocam com velocidade máxima maior para atender uma área afetada de maior tamanho, por exemplo, uma cidade assolada por um tsunami.

Para simular o movimento dos nós em uma rede de emergência no modo *ad hoc*, foi adotado o modelo de mobilidade *Random Walk 2D Mobility*, já implementado no NS-3, que funciona da seguinte forma:

(a) Cada nó se movimenta em direção e velocidade definidas aleatoriamente até alcançar os limites de uma região quadrada pré-estabelecida;

(b) Ao atingir o limite da região quadrada, o nó para por um determinado tempo e então o processo é reiniciado.

Em relação aos parâmetros do modelo de mobilidade, a área e velocidade foram definidas baseando-se em trabalhos sobre análise de mobilidade na literatura [CAI08 e SPY05]. Foram definidos os seguintes valores para o cenário dinâmico:

(a) *Cenário A*: os nós se movimentam dentro de uma região quadrada de 300 metros de lado com velocidade uniformemente distribuída entre 0 e 5 metros por segundo e tempo de parada dos nós igual a 8 segundos;

(b) *Cenário B*: os nós se movimentam dentro de uma região quadrada de 800 metros de lado com velocidade uniformemente distribuída entre 0 e 8 metros e tempo de parada dos nós igual a 10 segundos.

O primeiro nó se configura no instante zero e os demais nós respeitam uma variável aleatória uniformemente distribuída entre 5.5 e 2*número de nós em segundos, o que significa que os nós são iniciados em média a cada dois segundos. Semelhantemente ao cenário estático, a duração das simulações foi fixada em 300 segundos, o que representa um tempo suficiente para todos os nós obterem um endereço ou falharem no processo.

Diferentemente do cenário estático, no cenário dinâmico não ocorrem erros de configuração. Em vez disso, são contabilizados o número de redes criadas, o número de redes restantes no final da simulação e o quantitativo de mudanças de endereço dos nós. Devido o cenário dinâmico ser mais adverso que o cenário estático, onde perdas de endereços podem ocorrer com maior facilidade, o tamanho do espaço de endereçamento é maior do que no cenário estático. No cenário dinâmico A, o tamanho do espaço de endereçamento é igual ao dobro do número de nós e as redes podem ter 25, 50, 75 ou 100 nós. Devido à natureza menos densa do cenário dinâmico B, o tamanho do espaço de endereçamento é igual a 254 e as redes podem ter 20, 40, 60, 80 ou 100 nós.

4.3 MÉTRICAS

De forma a auxiliar na avaliação de desempenho dos protocolos deste trabalho, foram escolhidas algumas métricas levando-se em consideração as características dos cenários de simulação descritos anteriormente. As métricas são as seguintes:

- **Latência de configuração:** representa o tempo que um nó leva para obter um endereço a partir do momento que entra na rede;
- **Sobrecarga de controle:** corresponde à quantidade total de informação de controle trocada pelos nós na alocação e manutenção do espaço de endereçamento da rede;
- **Erros de configuração:** refere-se à quantidade de vezes que um nó não consegue obter um endereço, mesmo depois das tentativas locais e remotas de servidores. Essa métrica só é aplicável ao cenário estático, onde não é iniciado o mecanismo de recuperação de endereços quando um nó não consegue obter um endereço;
- **Mudanças de endereço:** representa o número de mudanças de endereço dos nós decorrentes de junções de redes, endereços duplicados ou não recebimento de mensagens de confirmação da presença de nós na rede;
- **Criação de redes:** representa o número de identificadores distintos de rede (ou NetIDs) gerados pelos nós que estão criando uma nova rede, por não detectarem uma rede existente ou quando um processo de recuperação de endereços é disparado, gerando um novo NetID. Além disso, quando ocorre junção de redes, os nós convergem para uma rede maior com um único NetID.

Foram executadas 30 rodadas de simulações para cada métrica dos cenários analisados. Foi realizado um tratamento estatístico dos resultados obtidos, obtendo média das amostras de cada métrica com nível de confiança de 95%.

4.4 RESULTADOS DO CENÁRIO ESTÁTICO

Nesta seção, são apresentados os resultados dos experimentos para o cenário estático e a discussão sobre os mesmos. O cenário estático é utilizado para observar a eficiência, corretude e escalabilidade dos protocolos escolhidos em um ambiente onde os nós estão parados. Primeiramente são discutidos os resultados para a métrica de latência de configuração. Em seguida, é discutida a sobrecarga de mensagens de controle adicionada à rede e, por último, os erros de configuração de cada protocolo.

4.4.1 LATÊNCIA DE CONFIGURAÇÃO

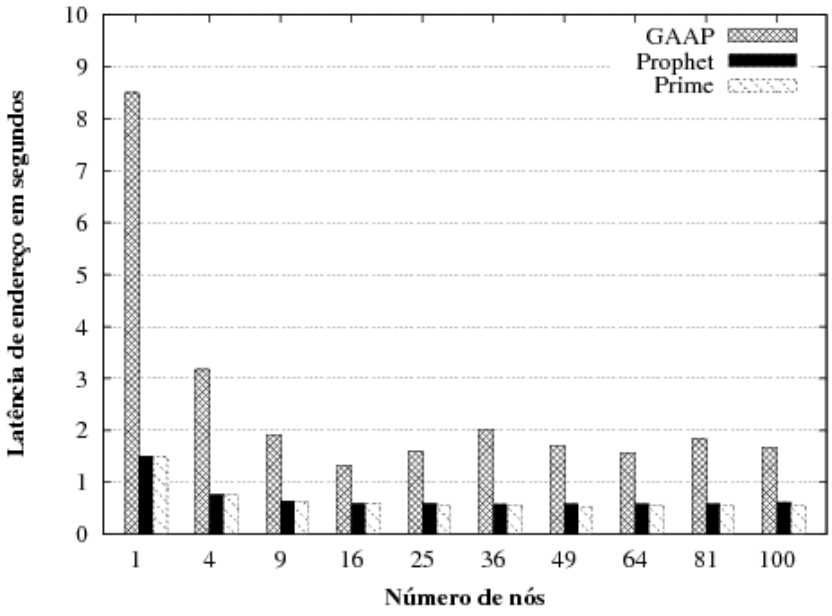

Figura 4.3: Latência de configuração no cenário estático

Observando a Figura 4.3, nota-se que o protocolo GAAP tem uma grande latência em comparação com os demais protocolos em uma rede com pequeno número de nós. Isso se deve à alta latência do primeiro nó da rede para obter um endereço após tentativas locais e remotas, o que acaba afetando a média geral do GAAP quando o número de nós é reduzido. Nesse caso, o nó envia três mensagens *Server Discovery* com tempo de espera de 0.5 segundo e então três mensagens *Proxy Discovery* com tempo de espera de 1, 2 e 4 segundos respectivamente, resultando em 8.5 segundos no total. O protocolo tem uma queda nos valores de

latência até uma rede com 16 nós, quando então começa a apresentar valores mais altos devido à topologia da rede. A latência atinge a marca de dois segundos para 36 nós, quando então os valores têm pouca variação e praticamente se estabilizam em um pouco abaixo de dois segundos. Ao verificar essa estabilidade, conclui-se que a topologia da rede passa a não ter grande influência nos resultados a partir de um determinado número de nós, pois apesar de solicitações remotas em redes maiores levarem mais tempo para serem respondidas, a quantidade maior de nós na rede proporciona maiores chances do nó requisitante encontrar um servidor de endereço em um tempo satisfatório.

Os protocolos *Prophet* e *Prime* apresentaram um bom desempenho para todas as populações, ficando abaixo da marca de dois segundos. O bom desempenho no caso do *Prophet* deve-se ao fato de todos os vizinhos diretos poderem oferecer endereço, não precisando o nó requisitante fazer qualquer tipo de solicitação remota, diminuindo significativamente o tempo de busca por endereço. Assim, o nó só irá se autoconfigurar se não tiver vizinhos dentro do seu alcance de comunicação. Quanto ao *Prime*, o tempo de obtenção de endereço reduzido é explicado pelo protocolo não diferenciar requisições locais e remotas, mas aproveitar uma mesma requisição acionando o ancestral de cada nó que não tiver endereço a oferecer. O *Prophet* apesar de ter a vantagem em relação ao *Prime* de não precisar requisitar endereço a nós a mais de um salto, perde em desempenho devido a conflitos de endereços gerados pela sua função de alocação, o que obriga os nós conflitantes a reiniciarem o processo de requisição de endereço, gerando um aumento da latência média da rede.

Em todos os protocolos, nota-se que o aumento do tamanho da rede não teve grande influência na latência, reflexo do comportamento estático dos nós e o tamanho do espaço de endereçamento acompanhar o crescimento da rede.

4.4.2 Erros de configuração

Para essa métrica, analisou-se o número de erros de configuração de cada protocolo. Para verificar o comportamento dos protocolos quanto à alocação de endereços em um cenário adverso com espaço de endereçamento igual ao número de nós, foi desabilitada a criação de novas redes quando os nós não conseguem obter um endereço a partir de tentativas locais e remotas. A Figura 4.4 mostra o

número de erros de configuração com o aumento do tamanho da rede para cada protocolo.

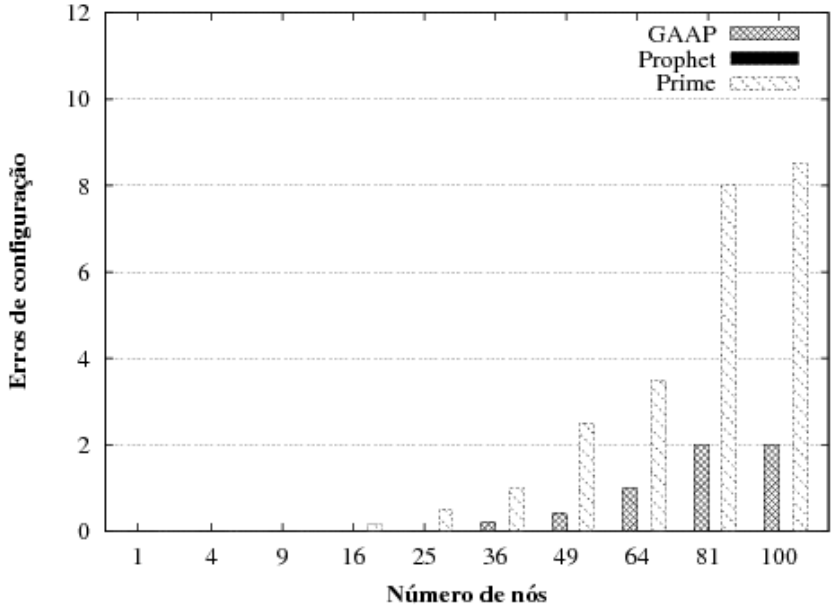

Figura 4.4: Erros de configuração no cenário estático

Verifica-se que o protocolo *Prophet* não apresentou erro de configuração nas populações analisadas. A topologia da rede e a ordem de aparição dos nós no cenário estático proporcionam ao nó requisitante ao menos um nó servidor no seu alcance de comunicação. Além disso, como o espaço de endereçamento é reaproveitado devido ao uso da função de alocação cíclica pelos nós servidores, todos os nós podem se configurar em qualquer momento.

O pior desempenho do *Prime* em relação ao GAAP deve-se ao mecanismo de alocação que não diferencia requisições locais e remotas, mas propaga mensagens de busca de servidores em uma única requisição. No *Prime*, como as requisições são propagadas em uma única direção, o esgotamento de endereços nos nós finais acaba gerando um número maior de erros de configuração. Apesar de resultar em maior latência na obtenção de endereço em um cenário estático como observado anteriormente, o mecanismo de tentativas locais e remotas utilizado pelo GAAP aumenta as chances do nó requisitante obter um endereço. As tentativas locais e remotas geram um atraso que pode se tornar benéfico, pois alguns nós acabam se tornando servidores durante o mecanismo de busca. Além disso, se mensagens contendo ofertas não chegarem ao nó requisitante, novas ofertas podem ser obtidas nas tentativas restantes de requisição de endereço.

Embora o número de erros de configuração do *Prime* possa parecer muito alto à primeira vista, os nós podem deixar de se configurar também pela perda de mensagens de alocação de endereço, motivada pela baixa capacidade e instabilidade dos canais sem fio. Além disso, deve-se levar em consideração que o tamanho do espaço de endereçamento é o mínimo possível (pior caso), sendo igual ao número de nós da rede.

Os resultados em geral podem melhorar se for aumentado o número de tentativas na alocação de endereços e/ou tempo de espera por respostas de endereço pelos nós requisitantes. No entanto, isso pode comprometer seriamente o tempo médio do nó para obtenção de endereço.

4.4.3 SOBRECARGA DE MENSAGENS

O gráfico da Figura 4.5 mostra o número total de bytes relacionado a mensagens periódicas enviadas pelos nós, sendo mensagens *Hello* para os protocolos GAAP e *Prophet* e mensagens *Recycle* para o protocolo *Prime*. Para cada nó, a contabilidade das mensagens periódicas é iniciada no momento de sua configuração e perdura enquanto o nó estiver configurado, até o término da simulação. A sobrecarga total de mensagens periódicas representa o somatório dos tráfegos individuais de todos os nós da rede.

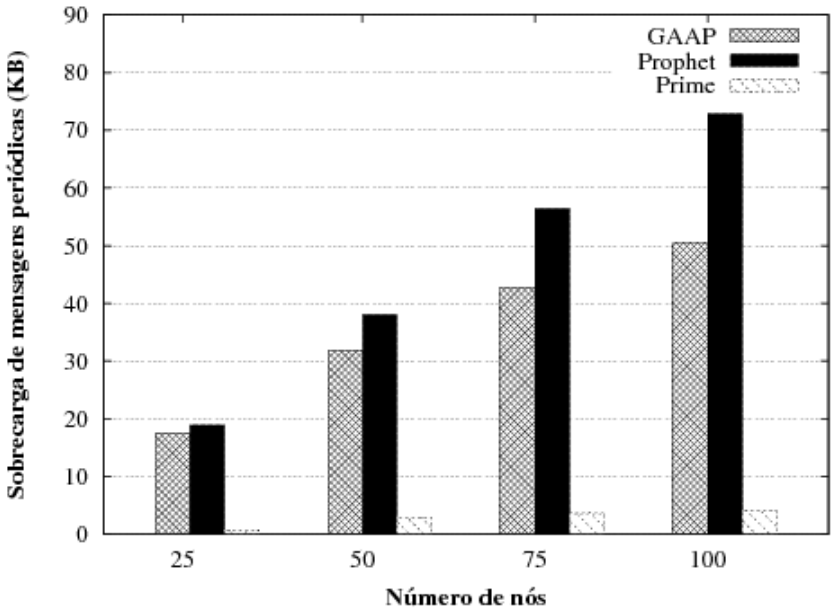

Figura 4.5: Sobrecarga de mensagens periódicas em KB no cenário estático

Na Figura 4.5, verifica-se que os protocolos GAAP e *Prophet* apresentaram valores significativamente mais altos que o *Prime* em todas as populações. Isso porque, em redes com GAAP e *Prophet,* todos os nós periodicamente enviam mensagens *Hello*, ao passo que no *Prime* a responsabilidade pelo envio de mensagens *Recycle* é apenas do nó raiz e mensagens de resposta dos nós a *Recycle* não são contabilizadas. Como no cenário estático apenas uma rede é criada, o único nó raiz no *Prime* gera um tráfego que não apresenta variação mesmo com o aumento das populações de nós. A sobrecarga um pouco maior do *Prophet* em relação ao GAAP, para redes a partir de 64 nós, deve-se ao fato de mais nós no *Prophet* poderem propagar mensagens periódicas, já que alguns nós no GAAP ficaram impossibilitados devido a erros de configuração, conforme descrito na subseção 4.4.2.

O gráfico da Figura 4.6 mostra o número total de bytes trafegados na rede, representado pelo tráfego gerado por mensagens periódicas e mensagens trocadas pelos nós durante o processo de alocação de endereço.

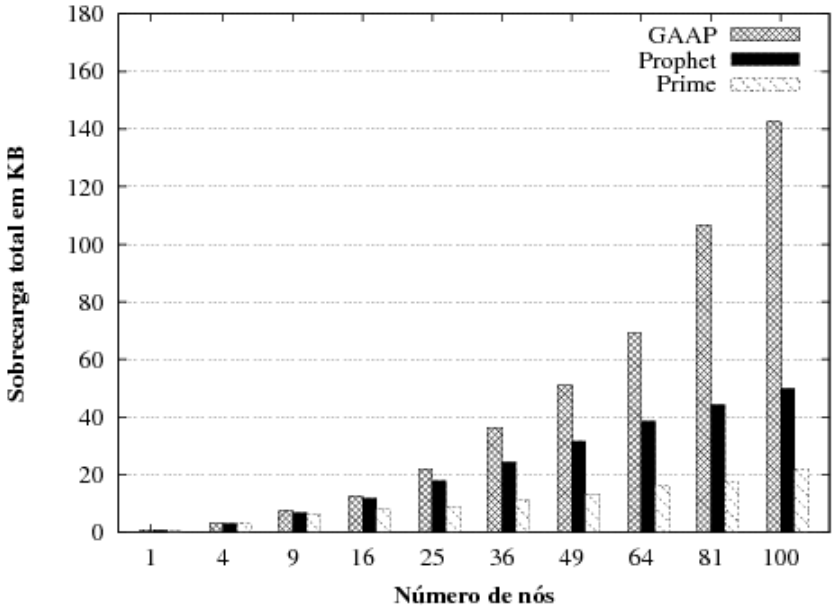

Figura 4.6: Sobrecarga total em KB no cenário estático

O protocolo GAAP teve o pior desempenho dentre os protocolos analisados, devido ao grande número de mensagens trocadas pelos nós durante o processo de alocação de endereço. O mecanismo de alocação local e remota utilizado pelo GAAP leva ao envio de várias mensagens em *broadcast*, cujo número cresce com o tamanho da rede, conforme ilustra a Figura 4.7. Ao receberem as mensagens em

broadcast provenientes do nó requisitante, cada nó servidor que tiver endereços disponíveis para alocação, envia sua oferta em mensagem *unicast*. Desta forma, como várias ofertas podem ser enviadas para uma mesma requisição de endereço, o número de mensagens *unicast* também se torna elevado, conforme mostra a Figura 4.8.

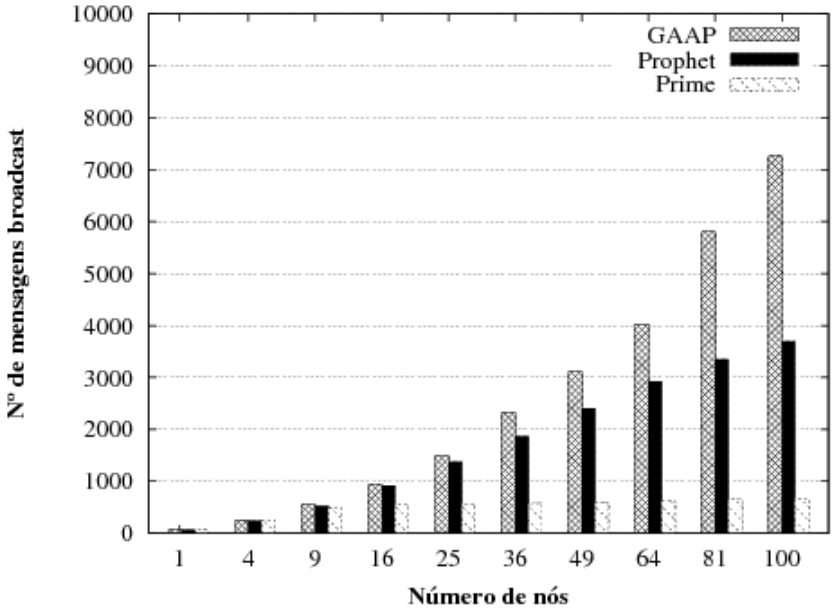

Figura 4.7: Número de mensagens *broadcast* no cenário estático

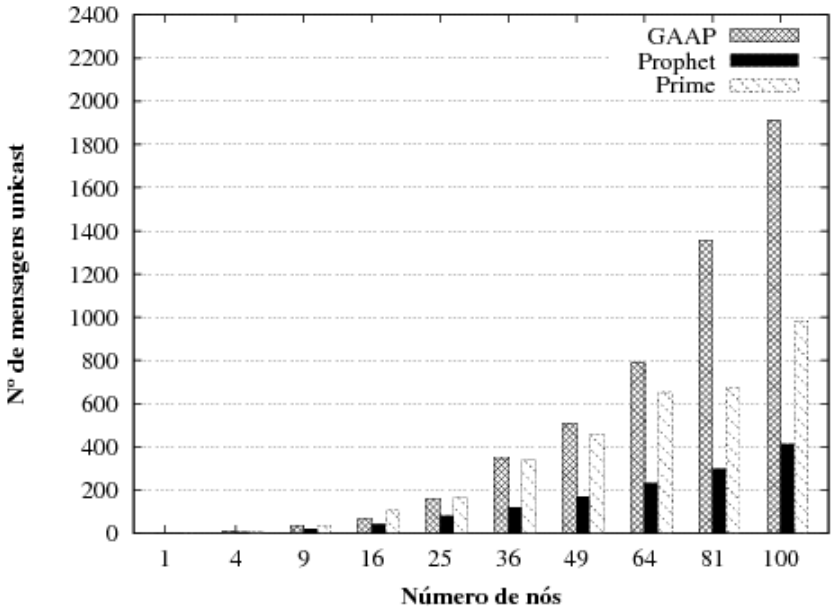

Figura 4.8: Número de mensagens *unicast* no cenário estático

Apesar de utilizar um mecanismo de alocação de endereço que dispensa o envio de mensagens a mais de um salto, o protocolo *Prophet* adiciona um tráfego de controle significativo à rede com o envio periódico de mensagens *Hello* realizado por todos os nós, bem superior ao tráfego gerado por mudanças de endereço efetuadas por nós que apresentaram endereços duplicados. O número máximo de mudanças de endereço chega a 4 para uma rede com 100 nós, conforme ilustra a Figura 4.9. Para populações abaixo de 49 nós, não foram registradas mudanças de endereço nos protocolos. Os protocolos GAAP e *Prime* não apresentaram mudanças de endereço em todas as populações, devido à característica dessas soluções de garantirem a unicidade na alocação. Além disso, no caso do *Prime*, não foi necessário que nós iniciassem novas requisições de endereço geradas por perdas no envio de respostas às mensagens *Recycle* propagadas pelo nó raiz.

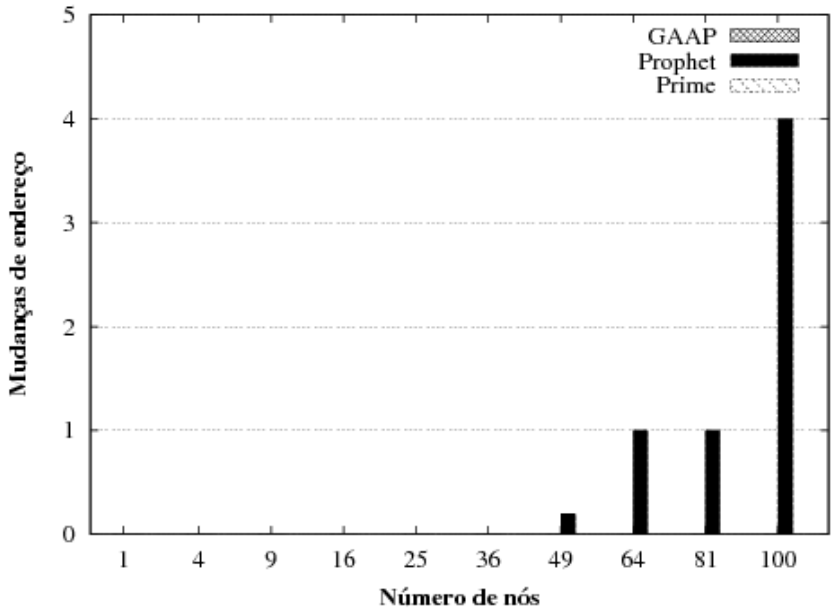

Figura 4.9: Mudanças de endereço no cenário estático

O desempenho melhor do *Prime* dentre os protocolos analisados deve-se principalmente ao tráfego significativamente menor gerado por mensagens periódicas como já exposto e por utilizar um mecanismo de alocação de endereço que não propaga tráfego significativo de mensagens *broadcast* e *unicast*, conforme ilustram as Figuras 4.7 e 4.8.

4.5 RESULTADOS DO CENÁRIO DINÂMICO

Nesta seção apresentamos os resultados dos experimentos para o cenário dinâmico, utilizado para representar situações de desastre onde a região afetada não possui infraestrutura de comunicação e as redes *ad hoc* móveis surgem como uma solução para interação entre as equipes de resgate. Conforme descrito anteriormente, o cenário dinâmico é representado por dois ambientes móveis. O cenário A, de maior densidade de nós, é usado para representar um cenário de emergência onde as equipes de resgate se deslocam com velocidade máxima menor (5 m/s) para atender uma área afetada menor (300m x 300m), por exemplo, a área alagada de um bairro residencial atingido por uma enchente. O cenário B, de menor densidade de nós, representa um cenário de emergência onde as equipes de resgate se deslocam com velocidade máxima maior (8 m/s) para atender uma área afetada de maior tamanho (800m x 800m), por exemplo, uma cidade assolada por um tsunami.

A seguir são apresentados os resultados dos experimentos com os protocolos analisados levando em consideração as métricas de latência de configuração, criação de redes, mudanças de endereço e sobrecarga de mensagens.

4.5.1 CENÁRIO A

4.5.1.1 Latência de configuração

A Figura 4.10 mostra o tempo médio de obtenção de endereço dos protocolos analisados em um cenário dinâmico.

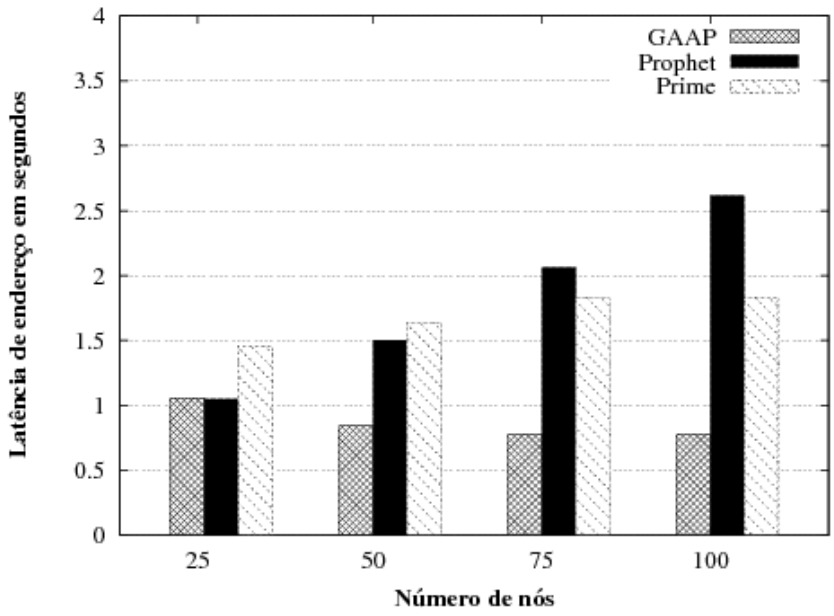

Figura 4.10: Latência de configuração no cenário A

Nota-se, ao contrário dos resultados apresentados no cenário estático, diminuição da latência no GAAP com o aumento do tamanho da rede. A latência média do GAAP para uma rede com 25 nós foi ligeiramente superior em comparação com o *Prophet*, devido à influência da latência de 8.5 segundos do primeiro nó da rede na média geral. O decréscimo da latência no GAAP com o aumento do número de nós deve-se à disponibilidade maior de servidores de endereço nas proximidades do nó requisitante, evitando que mensagens de busca de servidor remoto sejam propagadas pela rede.

Os protocolos *Prophet* e *Prime* apresentaram um crescimento da latência devido ao número de mudanças de endereço. A partir de 75 nós, os maiores valores de latência do *Prophet* em relação ao *Prime* coincidem com o salto no número de mudanças de endereço, conforme será descrito na subseção 4.5.1.3.

4.5.1.2 Criação de redes

As Figuras 4.11 e 4.12 mostram respectivamente o número de NetIDs gerados durante toda a simulação e o número de NetIDs remanescentes após o término da simulação.

Observando a Figura 4.11, verifica-se que os protocolos *Prophet* e GAAP obtiveram um ótimo desempenho, sendo criada apenas uma rede durante toda a simulação. No caso do *Prophet*, o desempenho se deve principalmente à

capacidade de todos os nós oferecerem endereço em qualquer momento, pois utilizam funções cíclicas onde o espaço de endereçamento é reaproveitado. Além disso, a alta densidade e baixa mobilidade dos nós no cenário A favorecem a disponibilidade de pelo menos um nó servidor no alcance de comunicação do nó requisitante. No GAAP, de forma semelhante, a alta densidade e baixa mobilidade dos nós favorecem a disponibilidade de nós servidores no raio de alcance do nó requisitante, diminuindo a possibilidade de criação de novas redes depois da busca de servidor remoto ou recuperação de endereços. O protocolo *Prime* apresentou desempenho ruim devido ao não recebimento de ofertas de endereços pelos nós requisitantes, resultando na criação de novas redes. Por não utilizar um método de alocação de endereço exaustivo que diferencie requisições locais e remotas, o recebimento de ofertas de endereço tende a oscilar mais devido à mobilidade dos nós.

Conforme mostra a Figura 4.12, verifica-se que o número de NetIDs finais é menor do que o número de NetIDs gerados, como esperado. No GAAP e *Prophet*, o número de NetIDs finais é o mesmo que o número de NetIDs gerados, já que apenas uma rede foi criada. O mecanismo de junção de redes do *Prime* não se mostrou eficiente, apresentando uma redução pequena no número de NetIDs em todas as populações de nós.

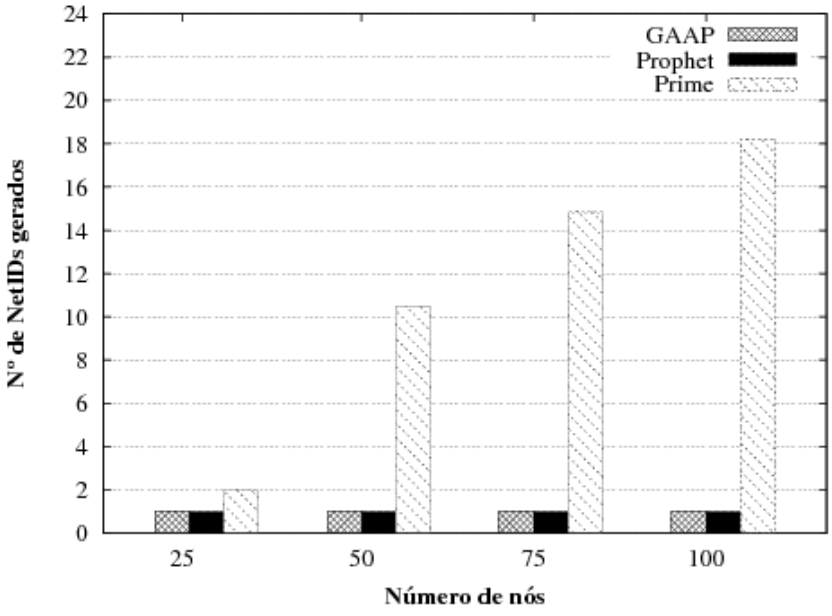

Figura 4.11: Número de NetIDs gerados no cenário A

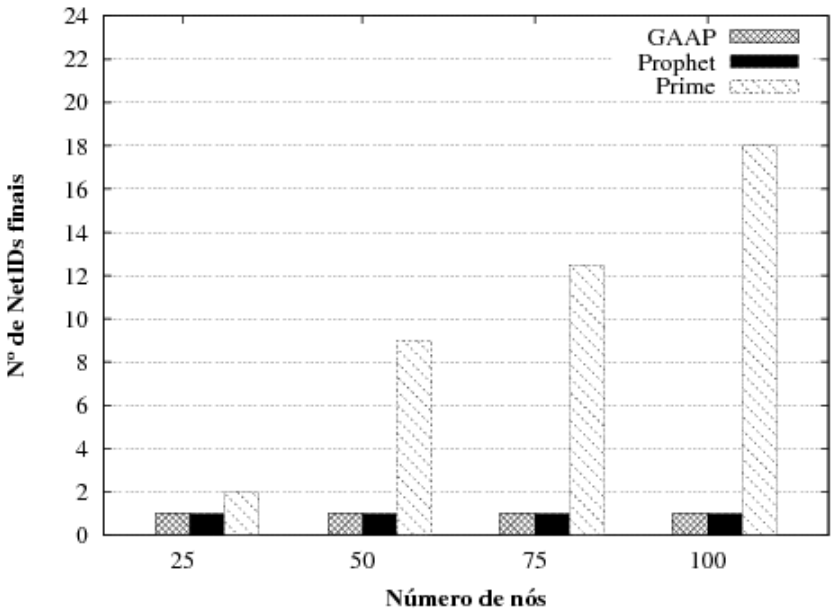

Figura 4.12: Número de NetIDs remanescentes no cenário A

4.5.1.3 Mudanças de endereço

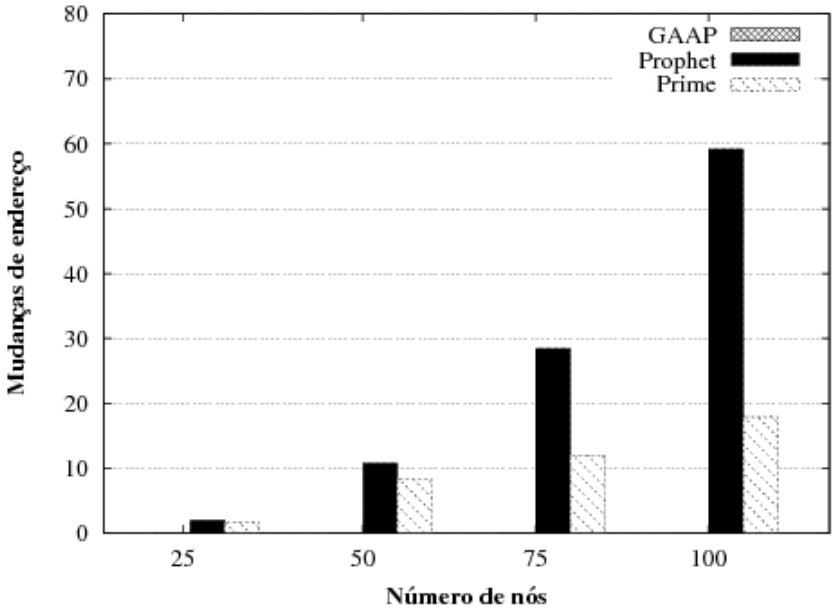

Figura 4.13: Mudanças de endereço no cenário A

Observando a Figura 4.13, verifica-se que o protocolo GAAP obteve os melhores resultados, não apresentando nenhuma mudança de endereço. Isso está relacionado ao fato de apenas uma rede ter sido criada em todas as populações de nós conforme descrito na Figura 4.11. Assim, não há possibilidade de ocorrência de junções de redes, as quais resultariam em mudanças de endereço.

No *Prophet*, as mudanças de endereço têm um crescimento significativo com o aumento do número de nós. A presença de nós na vizinhança do nó requisitante tem efeito prejudicial, diferentemente do GAAP, no que se refere a mudanças de endereço. Por não possuir nenhum mecanismo de confirmação de recebimento de oferta, várias ofertas podem chegar ao nó requisitante que escolherá uma delas e não avisará os outros nós de sua escolha. Desta forma, o espaço de endereçamento se esgota mais rapidamente e conflitos de endereços começam a surgir devido ao reinício do ciclo de ofertas. Por conseguinte, os conflitos levam a mudanças de endereço dos nós envolvidos.

As mudanças de endereço apresentadas pelo *Prime* são decorrentes da perda ou não envio de respostas às mensagens *Recycle* enviadas pelo nó raiz. Tais ocorrências estão associadas à mobilidade dos nós. Nós cujas respostas não chegaram, ao receberem uma mensagem enviada pelo nó raiz notificando a rede sobre o estado atual da alocação depois do processo de *Recycle*, disparam uma nova requisição de endereço para seus pais, gerando assim mudanças de endereço.

4.5.1.4 Sobrecarga de mensagens

As Figuras 4.14 e 4.15 mostram respectivamente a sobrecarga de mensagens periódicas na rede, sendo mensagens *Hello* nos protocolos GAAP e *Prophet* e mensagens *Recycle* no *Prime,* e a sobrecarga total de mensagens, incluindo mensagens periódicas e mensagens trocadas pelos nós durante o processo de alocação de endereço.

No que se refere à sobrecarga de mensagens periódicas, os protocolos analisados tiveram comportamento semelhante ao apresentado no cenário estático. Observando a Figura 4.14, verifica-se que o *Prime* apresentou novamente o menor número de bytes trafegados em mensagens periódicas para todas as populações, o que se deve ao fato de apenas o nó raiz ser responsável por propagar esse tipo de mensagem. Como há um crescimento no número de redes criadas com o aumento do número de nós conforme descrito na subseção 4.5.1.2, o número de nós raiz também cresce e consequentemente a quantidade de mensagens periódicas enviadas. Os protocolos *Prophet* e GAAP, por sua vez, apresentaram valores significativamente mais altos de bytes enviados, pois todos os nós periodicamente realizam o envio dessas mensagens.

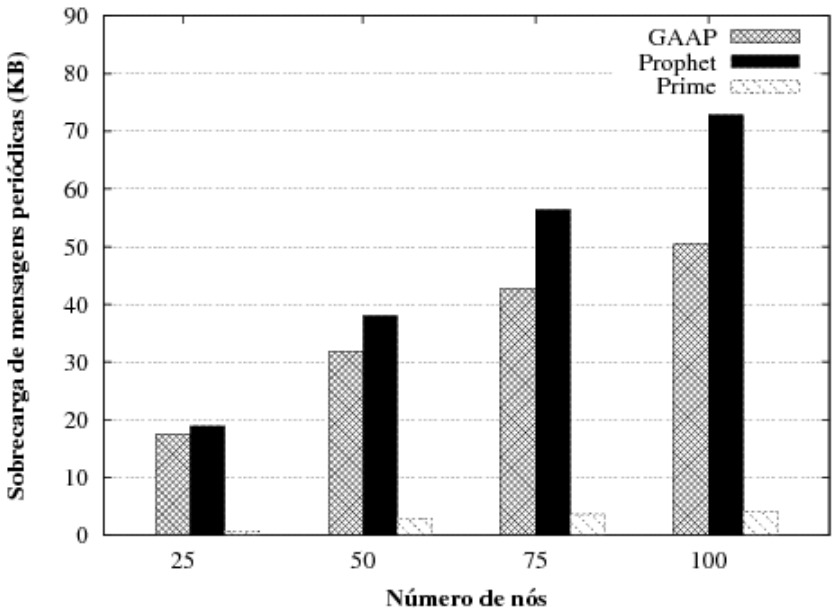

Figura 4.14: Sobrecarga de mensagens periódicas em KB no cenário A

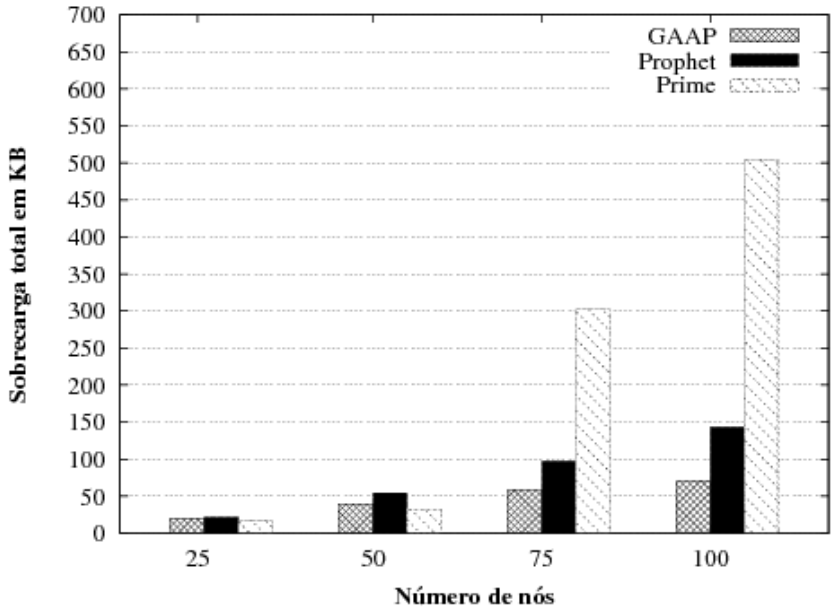

Figura 4.15: Sobrecarga total em KB no cenário A

Observando a Figura 4.15, o *Prime* apresentou o pior desempenho em termos de valores de sobrecarga total da rede. Apesar de não adicionar um tráfego de controle elevado em mensagens periódicas, uma quantidade significativa de mensagens de alocação de endereço é gerada por novas redes criadas. A independência dessas redes gera tráfegos de controle desassociados, que de forma separada contribuem de maneira significativa para o aumento da sobrecarga total da rede. O *Prophet* apresentou uma sobrecarga significativa nas populações

analisadas, o que se deve em grande parte ao tráfego de controle gerado pelo envio de mensagens periódicas. Além disso, tráfego significativo é gerado por novas buscas de servidores, resultantes de conflitos de endereços. O bom desempenho do GAAP deve-se em grande parte ao fato do protocolo não ter apresentado nenhuma mudança de endereço. Além disso, devido à disponibilidade de servidores de endereços no alcance de comunicação do nó requisitante, houve uma propagação menor no número de mensagens de *broadcast* e *unicast*, conforme ilustram as Figuras 4.16 e 4.17.

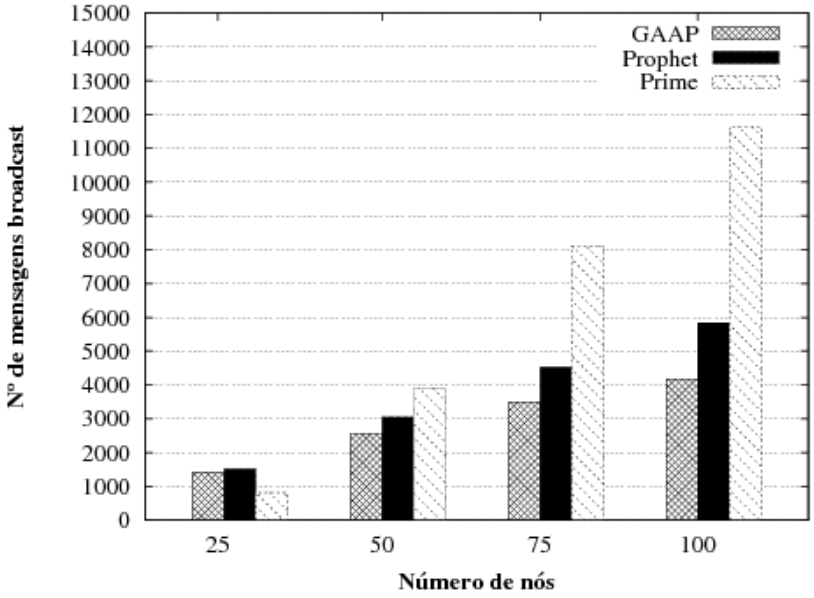

Figura 4.16: Número de mensagens *broadcast* no cenário A

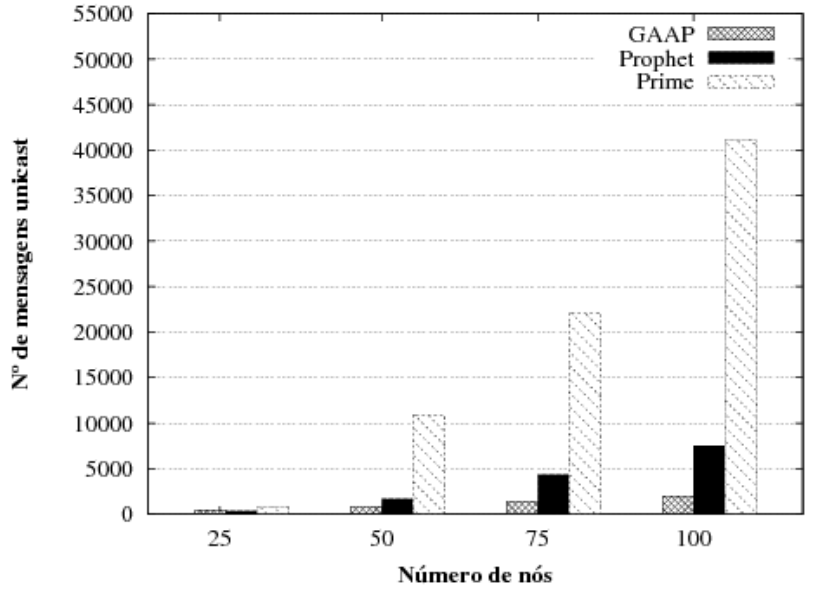

Figura 4.17: Número de mensagens *unicast* no cenário A

4.5.2 CENÁRIO B

4.5.2.1 Latência de configuração

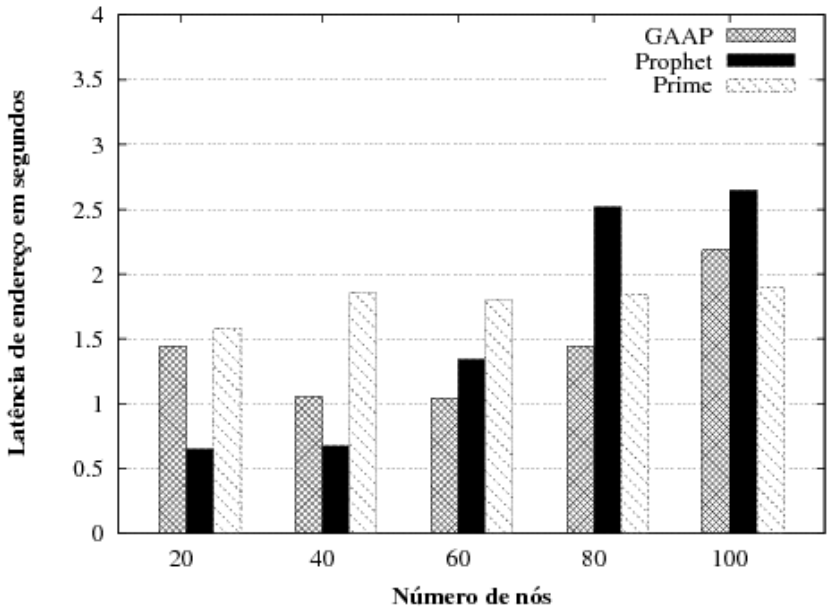

Figura 4.18: Latência de configuração no cenário B

Nota-se, ao contrário dos resultados apresentados no cenário A, crescimento da latência no GAAP com o aumento do número de nós. Observando a Figura 4.18, uma rede com 20 nós apresentou uma latência maior em comparação com redes de 40 e 60 nós. Devido à densidade menor de uma rede com 20 nós, há uma probabilidade maior de um nó estar em uma região pouco povoada e escassa de endereços, ou sem nenhum nó próximo. Além disso, a diminuição da latência até 60 nós deve-se à maior disponibilidade de nós servidores com o aumento do número de nós. A partir de 80 nós, um nó requisitante passar a não encontrar com frequência nós servidores na sua vizinhança direta devido ao esgotamento do *pool* de endereços. Essa indisponibilidade de servidores gera um aumento do número de reencaminhamentos de mensagens de busca de servidores e, consequentemente, o tempo de espera por um endereço também aumenta. Em alguns casos, o nó requisitante não consegue obter um endereço depois de realizar as tentativas locais e remotas, obrigando-o a disparar o processo de recuperação de endereços.

Assim como no cenário A, os protocolos *Prophet* e *Prime* apresentaram um crescimento da latência devido ao número de mudanças de endereço. A partir de 60 nós, os maiores valores de latência do *Prophet* em relação ao *Prime* coincidem com

o salto no número de mudanças de endereço, conforme será descrito na subseção 4.5.2.3.

4.5.2.2 Criação de redes

As Figuras 4.19 e 4.20 mostram respectivamente o número de NetIDs gerados durante toda a simulação e o número de NetIDs remanescentes após o término da simulação. Assim como no cenário A, os protocolos *Prophet* e GAAP obtiveram os melhores resultados.

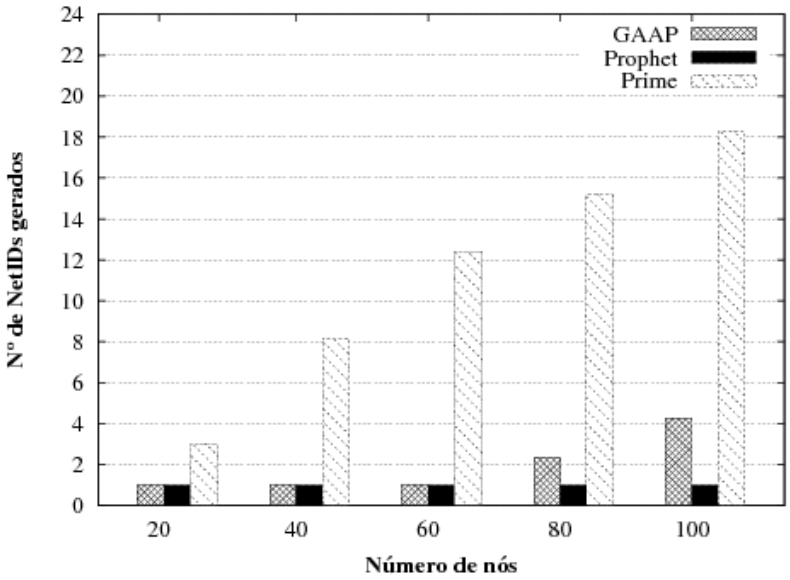

Figura 4.19: Número de NetIDs gerados no cenário B

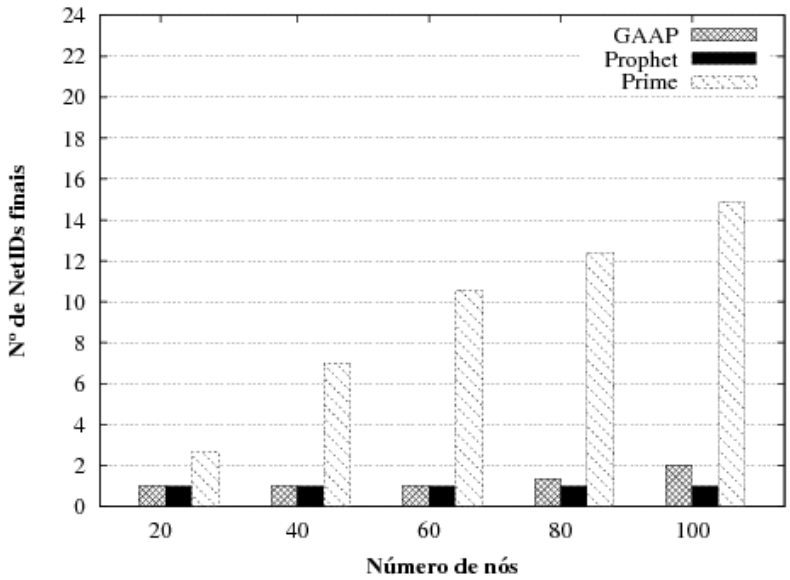

Figura 4.20: Número de NetIDs restantes no cenário B

De acordo com a Figura 4.19, no *Prophet* é gerado apenas um NetID em todas as populações de nós, devendo-se a dois fatores. O primeiro deles é a capacidade de todos os nós oferecerem endereço em qualquer momento devido ao uso de funções cíclicas de alocação, permitindo o reaproveitamento do espaço de endereçamento. O segundo fator é de existir pelo menos um nó servidor no alcance de comunicação do nó requisitante, o que se explica pela intensa movimentação dos nós. No GAAP, em vez de ser gerado apenas um NetID em todas as populações de nós como no cenário A, mais de um NetID foi gerado para 80 e 100 nós. Apesar do aumento da densidade dos nós contribuir para o esgotamento mais rápido dos *pools* de endereços, isso não é suficiente para explicar o crescimento no número de NetIDs gerados. Devido à área de abrangência da rede e movimentação dos nós serem maiores no cenário B, há uma disponibilidade menor de nós servidores nas proximidades do nó requisitante, o que acabou resultando em alguns casos no disparo do processo de recuperação de endereços quando as tentativas locais e remotas de obtenção de endereço não foram bem-sucedidas. O *Prime*, por sua vez, apresentou desempenho ruim devido ao não recebimento de ofertas de endereço pelos nós requisitantes, resultando na criação de novas redes. Por não utilizar um método de alocação de endereço exaustivo que diferencie requisições locais e remotas, o recebimento de ofertas de endereço tende a oscilar mais devido à mobilidade dos nós.

Observando a Figura 4.20, verifica-se que o número de NetIDs finais é menor do que o número de NetIDs gerados, como esperado. No *Prophet*, o número de NetIDs finais é o mesmo que o número de NetIDs gerados, já que apenas uma rede foi criada. O mecanismo de junção de redes do GAAP se mostrou eficiente, reduzindo aproximadamente pela metade o número de NetIDs após o término da simulação. O *Prime,* por sua vez, teve um desempenho ruim, apresentando uma redução pequena no número de NetIDs em todas as populações de nós.

4.5.2.3 Mudanças de endereço

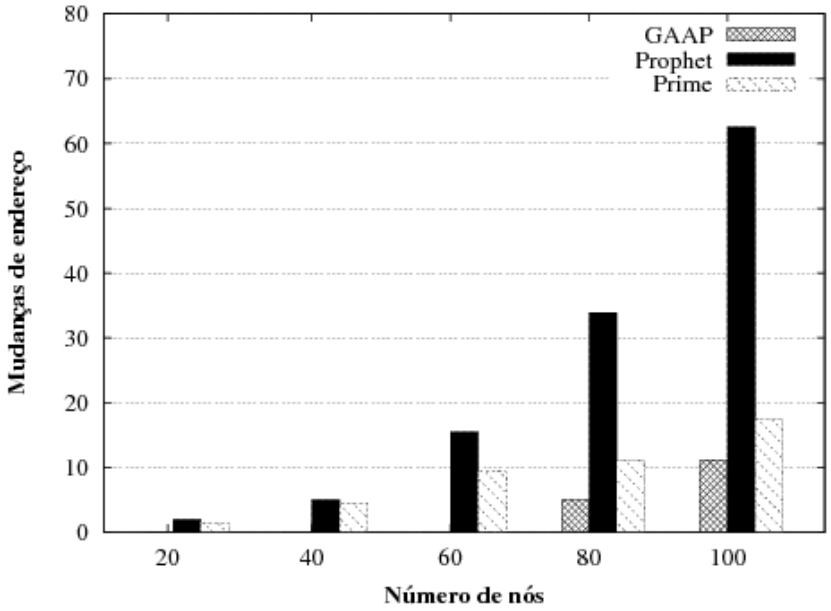

Figura 4.21: Mudanças de endereço no cenário B

Observando a Figura 4.21, verifica-se que o protocolo GAAP obteve os melhores resultados dentre os protocolos avaliados. No GAAP, os valores para 80 e 100 nós estão relacionados a junções de redes, as quais resultam dos processos de recuperação de endereços e consequente criação de novos NetIDs. A não ocorrência de mudanças de endereço para as demais populações deve-se ao fato de apenas uma rede ter sido gerada conforme ilustra a Figura 4.19.

Os protocolos *Prime* e *Prophet* tiveram comportamento semelhante ao apresentado no cenário A. No *Prophet*, as mudanças de endereço têm um crescimento significativo com o aumento do número de nós. A presença de nós na vizinhança do nó requisitante tem efeito prejudicial, diferentemente do GAAP, no que se refere a mudanças de endereço. Por não possuir nenhum mecanismo de confirmação de recebimento de oferta, várias ofertas podem chegar ao nó requisitante que escolherá uma delas e não avisará os outros nós de sua escolha. Dessa forma, o espaço de endereçamento se esgota mais rapidamente e conflitos de endereços começam a surgir devido ao reinício do ciclo de ofertas. Por conseguinte, os conflitos levam a mudanças de endereço dos nós envolvidos. As mudanças de endereço apresentadas pelo *Prime* são decorrentes da perda ou não envio de respostas às mensagens *Recycle* enviadas pelo nó raiz. Tais ocorrências estão associadas à mobilidade dos nós. Nós cujas respostas não chegaram, ao

receberem uma mensagem enviada pelo nó raiz notificando a rede sobre o estado atual da alocação depois do processo de *Recycle*, disparam uma nova requisição de endereço para seus pais, gerando assim mudanças de endereço.

4.5.2.4 Sobrecarga de mensagens

As Figuras 4.22 e 4.23 mostram respectivamente a sobrecarga de mensagens periódicas na rede, sendo mensagens *Hello* nos protocolos GAAP e *Prophet* e mensagens *Recycle* no *Prime,* e a sobrecarga total de mensagens, incluindo mensagens periódicas e mensagens trocadas pelos nós durante o processo de alocação de endereço.

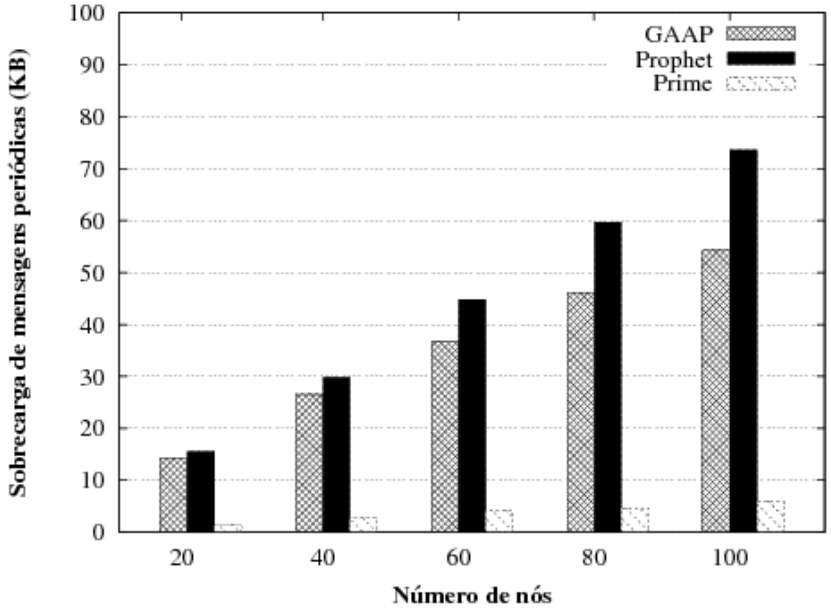

Figura 4.22: Sobrecarga de mensagens periódicas em KB no cenário B

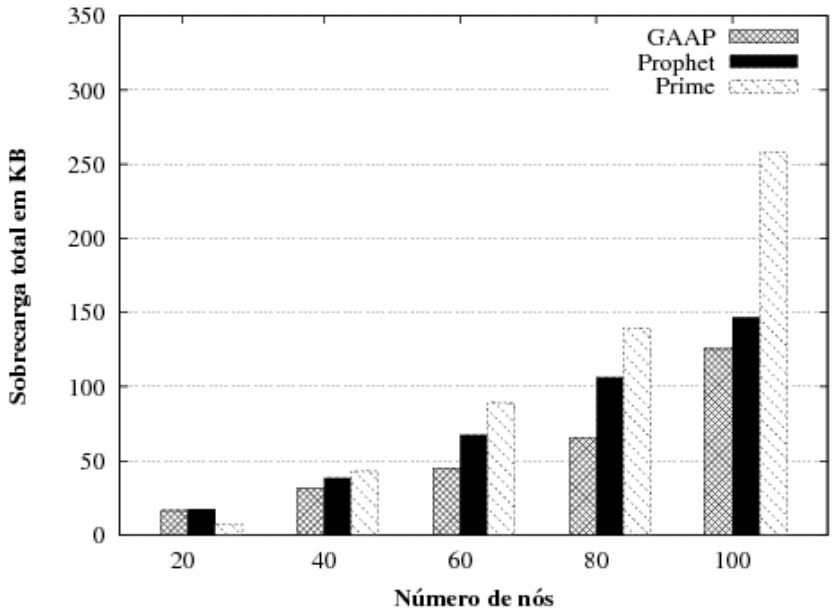

Figura 4.23: Sobrecarga total em KB no cenário B

No que se refere à sobrecarga de mensagens periódicas, os protocolos analisados tiveram comportamento semelhante ao apresentado no cenário A. De acordo com a Figura 4.22, o *Prime* apresenta novamente o menor número de bytes trafegados em mensagens periódicas para todas as populações, já que apenas o nó raiz é responsável por propagar esse tipo de mensagem. Como há um crescimento no número de redes criadas com o aumento do número de nós conforme descrito na subseção 4.5.2.2, o número de nós raiz também cresce e consequentemente a quantidade de mensagens periódicas enviadas. Os protocolos *Prophet* e GAAP, por sua vez, apresentaram valores significativamente mais altos de bytes enviados, pois todos os nós periodicamente realizam o envio dessas mensagens.

Com relação à sobrecarga total de mensagens na rede, os resultados são semelhantes aos apresentados no cenário A. Observando a Figura 4.23, verifica-se que o *Prime* apresentou os piores resultados. Apesar de não adicionar um tráfego de controle elevado em mensagens periódicas, uma quantidade significativa de mensagens de alocação de endereço é gerada por novas redes criadas. A independência dessas redes gera tráfegos de controle desassociados, que de forma separada contribuem de maneira significativa para o aumento da sobrecarga total da rede. O *Prophet* apresentou uma sobrecarga significativa nas populações analisadas, o que se deve em grande parte ao tráfego de controle gerado pelo envio de mensagens periódicas. Além disso, tráfego significativo é gerado por novas

buscas de servidores, resultantes de conflitos de endereço. O melhor desempenho do GAAP deve-se em grande parte à maior disponibilidade de nós servidores nas proximidades do nó requisitante, resultando em uma propagação menor no número de mensagens *broadcast* e *unicast*, conforme ilustram as Figuras 4.24 e 4.25. Na Figura 4.24, a partir de 80 nós, há um crescimento maior no número de mensagens *broadcast* devido à escassez de servidores nas proximidades do nó requisitante. Essas mensagens estão relacionadas à busca de servidores remotos e aos processos de recuperação de endereços disparados quando o nó requisitante não consegue obter endereço depois de tentativas locais e remotas. Apesar de esse crescimento culminar em um desempenho pior que o *Prophet* para 100 nós, o número de mensagens *unicast* propagadas pelo GAAP para a mesma população é significativamente menor, conforme mostra a Figura 4.25.

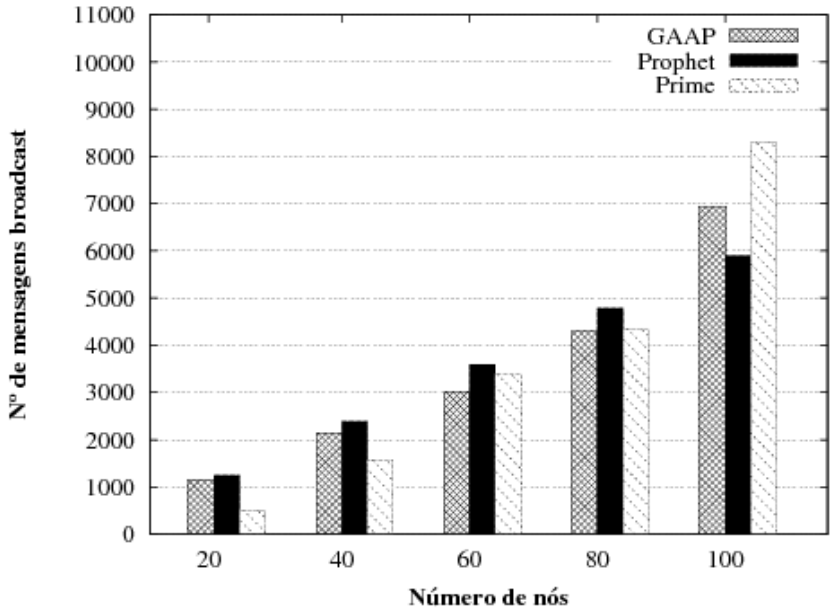

Figura 4.24: Número de mensagens broadcast no cenário B

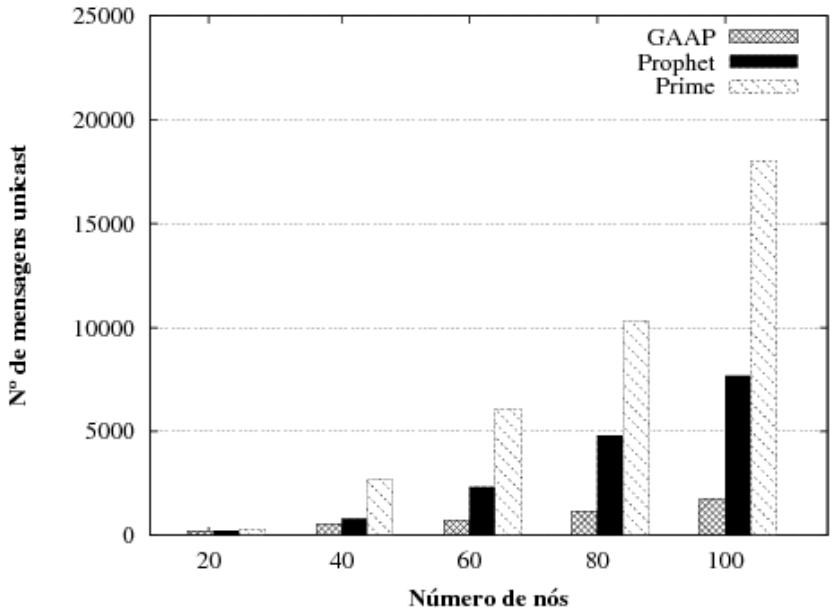

Figura 4.25: Número de mensagens unicast no cenário B

4.6 DISCUSSÃO

Nesse capítulo foram apresentados os resultados dos experimentos com os protocolos analisados para os cenários estático e dinâmico. No cenário estático, a ideia é avaliar o funcionamento básico dos protocolos sob baixa variação das condições da rede, onde os nós estão parados. No cenário dinâmico, é avaliado o desempenho dos protocolos sob uma perspectiva mais realista, onde os nós se movem livremente. Dentro do cenário dinâmico, foram simulados dois ambientes móveis. O cenário A é usado para representar um cenário de emergência cuja área afetada e velocidade dos nós são menores em relação ao cenário de emergência do cenário B. No cenário estático, a solução proposta não se mostrou eficiente, apresentando valores de latência de configuração e sobrecarga de mensagens superiores em comparação com os demais protocolos. Tal desempenho deve-se principalmente ao mecanismo de alocação de endereço utilizado, que apesar de aumentar as chances de encontrar um servidor de endereço a partir de requisições locais e remotas, gera uma sobrecarga maior de mensagens na rede e um tempo médio de espera maior para configuração de um nó.

Apesar do seu desempenho geral não ter sido satisfatório no cenário estático, o GAAP apresentou bons resultados para a métrica erros de configuração. Os

resultados mostraram um número reduzido de erros de configuração mesmo para uma rede com 100 nós e espaço de endereçamento restrito, devido ao seu mecanismo de alocação de endereço que aumenta as chances de configuração dos nós.

Os protocolos *Prophet e Prime* apresentaram resultados semelhantes no cenário estático, mas o *Prophet* se destaca por apresentar bons resultados nas métricas latência e erros de configuração. Tais resultados devem-se principalmente ao seu mecanismo de alocação de endereços local que dispensa a propagação de requisições remotas, bastando que haja pelo menos um nó servidor no raio de alcance do nó requisitante.

No que diz respeito ao cenário dinâmico, os protocolos apresentaram comportamentos diferentes do cenário estático. A solução proposta teve o melhor desempenho dentre os mecanismos de endereçamento analisados. Ao contrário do que aconteceu no cenário estático, os valores de latência de configuração e sobrecarga de mensagens se mostraram satisfatórios. No cenário dinâmico A, a alta densidade de nós na rede favorece a disponibilidade de servidores de endereço no raio de alcance do nó requisitante, diminuindo o tempo de obtenção de um endereço e evitando a inundação da rede com mensagens de busca de servidores remotos. No cenário dinâmico B, a solução proposta também apresentou os melhores resultados. Esse desempenho se mostrou satisfatório mesmo para redes com maior número de nós, onde o esgotamento do espaço de endereçamento acontece mais rapidamente. Isso se deve ao mecanismo de recuperação de endereços, que apesar de inundar a rede com mensagens de controle, é disparado apenas quando o nó requisitante não obtém um endereço após realizar buscas locais e remotas por um nó servidor. Além disso, o método exaustivo de alocação do GAAP contribuiu de forma significativa para a criação de poucas redes, diminuindo o tráfego de controle gerado por junções de redes.

Os protocolos *Prophet e Prime* apresentaram valores de latência mais altos devido principalmente ao número significativo de mudanças de endereço dos nós. O desempenho pior do *Prophet* deve-se ao uso de uma função cíclica no processo de alocação onde o espaço de endereçamento é reaproveitado, resultando em conflitos de endereços depois de um determinado intervalo.

O protocolo *Prime* apesar do tráfego de controle reduzido de mensagens periódicas enviadas à rede, apresentou a maior sobrecarga total de mensagens, o

que se deve a um grande número de perda de mensagens no processo de alocação de endereço decorrente da mobilidade dos nós. Tais perdas acarretaram em muitos casos na criação de novas redes. Como o mecanismo de junção de redes não se mostrou eficiente, o tráfego de controle gerado por essas redes elevou de forma significativa a sobrecarga total de mensagens no enlace de dados. Quanto ao protocolo *Prophet*, apesar do número elevado de mudanças de endereço e do envio de mensagens periódicas por todos os nós, a presença de apenas uma rede durante toda a simulação contribuiu para uma menor sobrecarga total de mensagens em relação ao *Prime*.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as considerações finais do trabalho, descrevendo as conclusões e sugestões de trabalhos futuros.

5.1 CONCLUSÕES

Este trabalho propôs e avaliou um novo protocolo de alocação de endereços para redes *ad hoc* móveis utilizadas por equipes de resgate em cenários de emergência como catástrofes e desastres. A solução proposta, denominada GAAP (*Greedy Address Allocation Protocol*), busca alocar endereços únicos aos nós de forma a reduzir o tráfego de controle introduzido pela detecção de duplicação. Para manter a consistência do espaço de endereçamento durante o tempo de vida da rede, o protocolo realiza a recuperação de endereços perdidos e tratamento de partições e junções de redes quando necessários.

Um dos principais desafios deste trabalho foi implementar a solução proposta no simulador de redes NS-3, pois este ainda não está ajustado por padrão para trabalhar com mecanismo de endereçamento. Assim, uma nova camada de endereçamento foi desenvolvida, incluindo funções para inicialização dos nós da rede em diferentes instantes. Além disso, grande esforço foi empregado na comparação da solução com outros mecanismos de endereçamento também baseados na abordagem *stateful*. A solução proposta foi comparada com os protocolos *Prime* e *Prophet*, os quais também foram implementados no NS-3.

Foram usados dois tipos de cenário de simulação, sendo um deles de comportamento mais previsível onde os nós estão parados (cenário estático), e outro onde os nós se movimentam aleatoriamente (cenário dinâmico). No cenário dinâmico, mais próximo da realidade, foram definidos dois ambientes móveis para representar cenários de emergência como catástrofes e desastres. O cenário A é usado para representar um cenário de emergência cuja área afetada e velocidade dos nós são menores em relação ao cenário de emergência representado pelo cenário B.

Os resultados dos experimentos de simulação mostraram que a solução proposta se mostrou eficiente no cenário dinâmico. O protocolo foi capaz de alocar

endereços para todos os nós com latência média satisfatória, gerando uma sobrecarga total moderada na rede. No cenário A, a latência apresentou valores próximos de um segundo e a sobrecarga total chegou no máximo a 75KB, para uma rede com 100 nós. No cenário B, para o mesmo número de nós, a latência média ficou próxima de dois segundos e sobrecarga total em torno de 130KB. No cenário estático, apesar da solução proposta ter apresentado um elevado tráfego de controle na rede, a latência de configuração apresentou níveis aceitáveis, ficando abaixo de dois segundos para maioria das populações de nós. Ainda no cenário estático, o GAAP apresentou um número reduzido de erros de configuração, mesmo com espaço de endereçamento restrito. As demais soluções apesar de apresentarem latência e tráfego de controle moderados no cenário estático, não obtiveram bom desempenho no cenário dinâmico. Enquanto o *Prophet* obteve valores altos de latência devido ao elevado número de mudanças de endereço, o *Prime* apresentou uma sobrecarga elevada de tráfego de controle devido ao aparecimento de novas redes.

5.2 TRABALHOS FUTUROS

As sugestões de trabalhos futuros seguem a linha de melhorar o desempenho da solução atual e comparar com outros tipos de abordagens de endereçamento.

Uma ideia é modificar o GAAP para tornar-se eficiente em um cenário estático, pois os resultados mostraram que as tentativas locais e remotas durante o processo de alocação de endereço elevam de forma significativa o tempo médio de configuração dos nós e a sobrecarga total da rede. Uma possibilidade é acrescentar nas mensagens *Hello* originárias de cada nó a informação do tamanho do seu bloco de endereços disponíveis para alocação. Assim, cada nó da rede possuiria uma tabela de vizinhos com seus respectivos tamanhos de blocos de endereços. De posse de sua tabela de vizinhos, ao receber uma requisição de endereço, o nó busca em sua tabela o nó com maior tamanho de bloco de endereço e redireciona a requisição para o mesmo. Com isso evita-se que várias mensagens de uma mesma requisição sejam propagadas para toda a rede.

Outra sugestão de trabalho diz-se respeito à melhoria da estimativa do tamanho da rede utilizada no tratamento de junções de redes. O GAAP pode obter

uma estimativa mais precisa ao se utilizar da informação de conectividade dos nós da rede provida por protocolos de roteamento proativo.

Por fim, pretende-se comparar o desempenho do GAAP com outros tipos de abordagens, principalmente *stateless*, avaliando o comportamento dos protocolos em cenários variados e com número maior de métricas. Tais avaliações podem levantar questões ainda não consideradas na solução proposta, apresentando-se como oportunidades de melhoria.

REFERÊNCIAS BIBLIOGRÁFICAS

[AGG05] Aggelou, G., "Mobile ad hoc networks: from wireless LANs to 4G networks", New York, NY: McGraw-Hill Professional Engineering, pp. xiii, (2005).

[ASCH10] Aschoff, R. R. (2010). "DNCP: Dynamic Node Configuration Protocol". Dissertação de Mestrado. Universidade Federal de Pernanbuco, Centro de Informática, Recife. Maio 2010.

[BLUE04] Bluetooth SIG. Bluetooth Specification Version 2.0+ EDR. November 2004.

[CAI08] Cai, H. e Eun, D. Y. (2008). "Toward stochastic anatomy of inter-meeting time distribution under general mobility models", In MobiHoc '08: Proceedings of the 9th ACM international Symposium on Mobile Ad Hoc Networking and Computing, pp. 273_282. ACM.

[DROMS97] Droms, R. (March de 1997). "Dynamic Host Configuration Protocol", IETF Network Working group - RFC 2131.

[FALL09] Fall, K., & Varadhan, K. (2009). “The Ns Manual (Formely ns notes and Documentation)”, The VINT Project, 2009.

[FEMA11] FEMA (2011). Federal Emergency Management Agency. http://www.fema.gov.

[GÜN02] Günes, M., & Reibel, J. (2002). “An IP Address Configuration Algorithm for Zeroconf Mobile Multihop Ad Hoc Networks”, Proc. Int'l. Wksp. Broadband Wireless Ad Hoc Networks and Services. Sophia Antipolis, France.

[JAC03] P.Jacquet, & Clausen, T. (October de 2003). “Optimized Link State Routing Protocol (OLSR)”, IETF Network Working group - RFC3623.

[JUBIN87] Jubin, J. e Tornow, J. (1987). "The DARPA packet radio network protocols", Proceedings of the IEEE, 75(1):21-32.

[MANET08] Manet (2008). IETF working group on mobile ad-hoc networks. Disponível em http://www.ietf.org/html.charters/manet-charter.html.

[MISRA01] Misra, A., Das, S., McAuley, A., & Das, S. K. (2001). "Autoconfiguration, Registration and Mobility Management for Pervasive Computing", IEEE Personal Communications , 8 (04).

[MOH02] Mohsin, M., & Prakash, R. (2002). “IP Address Assignment in a Mobile Ad Hoc Network”, Proc. IEEE MILCOM 2002. Anaheim, CA.

[MOTA09] Mota, V. "Um protocolo de roteamento tolerante a interrupções de comunicação para redes sem fio móveis em cenários de emergência". Dissertação de mestrado. Universidade Federal de Minas Gerais, Belo Horizonte. Julho 2009.

[NES02] Nesargi, S., & Prakash, R. (2002). “MANETconf: Configuration of hosts in a mobile ad hoc network”, Proc. IEEE INFOCOM 2002. New York, NY.

[NETO10] Neto, F. R. (2010). “Análise comparativa de mecanismos de endereçamento para MANETs”. Trabalho Final de Graduação. Universidade Federal de Pernambuco, Centro de Informática, Recife. Dezembro 2010.

[NRC11] NRC (2011). National Research Council. http://www.nationalacademies.org/nrc.

[NSNAM06] NS-3. (2006). (XORP Organization) Acesso em 2010, disponível em NS-3 official website: http://www.nsnam.org.

[PERK01] Perkins, C., Wakikawa, R., Belding-Royer, J. M., & Sun, Y. (November de 2001). "IP address autoconfiguration for ad hoc networks", IETF Network Working group.

[PERK96] Perkins, C. (October de 1996). "IP Mobility Support", IETF Network Working group, RFC 2002.

[RAO07] Rao, R.; Eisenberg, J. e Schmitt, T. (2007). "Improving Disaster Management: The Role of IT in Mitigation, Preparedness, Response, and Recovery". National Research Council, National Academy of Sciences Washington DC, ISBN: 0-309-66744-5.

[SAK06] Sakander, Z. "Address Allocation to Mobile Ad Hoc Networks". Master’s Thesis. Simon Fraser University, 2006.

[SNDC09] SNDC (2009). Defesa Civil. Secretaria Nacional de Defesa Civil. http://www.defesacivil.gov.br.

[SPY05] Spyropoulos, T.; Psounis, K. e Raghavendra, C. (2005). "Spray and wait: an efficient routing scheme for intermittently connected mobile networks". In ACM SIGCOMM workshop on Delay-tolerant networking, pp. 252_259. ACM.

[SUN03] Sun, Y., & Belding-Royer, E. M. (2003). “Dynamic Address Configuration in Mobile Ad Hoc Networks”. UCSB tech. rep. 2003-11. Santa Barbara, CA.

[VAID02] Vaidya, N. (2002). “Weak Duplicate Address Detection in Mobile Ad Hoc Networks”, Proc. ACM MobiHoc. Lausanne, Switzerland.

[WEHBI05] WEHBI, B. “Address Autoconfiguration in Ad Hoc Networks”. Internal Report. Institut National des Telecommunications, 2005.

[WEN03] Weniger, K. (2003). “Passive Duplicate Address Detection in Mobile Ad Hoc Networks”. Proc.IEEE WCNC. New Orleans, L.A.

[WEN05] Weniger, K. (2005). "PACMAN: Passive Autoconfiguration for Mobile Ad Hoc Networks". IEEE Journal on Selected Areas in Communications , 23 (3), 507-519.

[WIFI07] IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. Technical report, 2007.

[WIMAX07] Jeffrey G. Andrews, Arunabha Ghosh, Rias Muhamed, “Fundamentals of WiMAX:Understanding Broadband Wireless Networking”, Prentice Hall, 2007

[YUAN05] Yuan-Ying, H., & Chien-Chao, T. (2005). "Prime DHCP: A Prime Numbering Address Allocation Mechanism for MANETs". IEEE Communications Letters.

[ZHOU03] Zhou, H., M., N. L., & Mutka, M. W. (2003). “Prophet Address Allocation for Large Scale MANETs". Proc. of IEEE INFOCOM 2003, 2, pp. 1304-1311. San Franciso, CA.