COPPE/UFRJ

PROPOSTA DE UM MECANISMO DE SEGURANÇA BASEADO EM TROCA DE CHAVES ASSIMÉTRICAS PARA REDES TOLERANTES A ATRASOS E DESCONEXÕES

Rafael de Moraes Santos Fernandes

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador(es): Luís Felipe Magalhães de Moraes

Rio de Janeiro

Março de 2009

PROPOSTA DE UM MECANISMO DE SEGURANÇA BASEADO EM TROCA DE CHAVES ASSIMÉTRICAS PARA REDES TOLERANTES A ATRASOS E DESCONEXÕES

Rafael de Moraes Santos Fernandes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Luís Felipe Magalhães de Moraes, Ph. D.

Prof. Claudio Luis de Amorim, Ph. D.

Prof. Márcio Portes de Albuquerque, Dr.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2009

Fernandes, Rafael de Moraes Santos

Proposta de um Mecanismo de Segurança Baseado em Troca de Chaves Assimétricas para Redes Tolerantes a Atrasos e Desconexões/Rafael de Moraes Santos Fernandes. - Rio de Janeiro: UFRJ/COPPE, 2009.

XVIII, 120 p.: il.; 29,7 cm.

Orientador: Luís Felipe Magalhães de Moraes

Dissertação (mestrado) - UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2009.

Referências Bibliográficas: p. 95-100.

1. Redes Sem Fio. 2. Redes Ad Hoc. 3. DTNs. 4. Segurança de Redes. 5. Criptografia Assimétrica I. Moraes, Luís Felipe Magalhães de II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Dedicatória

Dedico este trabalho aos meus pais, Mario Luiz e Dayse, pela grande dedicação na minha formação como pessoa e pelos incontáveis incentivos dados ao longo da minha vida.

Agradecimentos

Primeiramente, gostaria de agradecer a Deus pela vida e por ter me dado saúde para poder realizar este trabalho.

Agradeço a minha família: Minha mãe e meu pai por tudo que fizeram por mim nesta vida. Aos irmãos Renato, Pedro e Matheus, que sempre proporcionam momentos de alegria. As minhas avós Maria e Zezé, por além de terem me criado, me deram muito apoio durante o desenvolvimento deste trabalho. Ao meu avô Armando, por todo apoio. A minha tia e madrinha Sandra, que sempre me deu apoio e incentivo e é uma pessoa muito especial na minha vida. A minha tia Valéria e ao tio Marinho, por todo incentivo, amor e carinho que me deram durante toda a vida. A Fatima por todo apoio, incentivo e carinho que me ofereceu durante este longo período que convivemos. Agradeço também a Fernanda, minha namorada, pelo apoio, carinho e atenção, principalmente na parte final deste trabalho.

Agradeço aos amigos do RAVEL e do LCP: Tiago, Rafael Bezerra, Júlio, Jorge, Paulo, Bruno, Eduardo, Airon, Cláudia, Diogo, Verissimo, Schiller, Michelini, Danielle, Gustavo, Beto, Luis, Rodrigo, Elenilson, Alexandre, Lauro, Arthur, Diego, Leonardo e todos os outros, que por ventura eu tenha esquecido. Agradecimento muito especial ao amigo Beto, pela amizade, pela ajuda desde o início deste trabalho, dando forças quanto eu achava que não ia conseguir fazer este trabalho.

Agradeço ao meu orientador, Prof. Luís Felipe, pelos grandes ensinamentos e pelo total apoio desde o início do meu trabalho e aos demais integrantes da banca, os Professores Claudio Amorim e Márcio Portes, pela valiosa ajuda nesta fase final.

As secretárias do PESC/COPPE: Solange, Sônia, Cláudia, Mercedes e Lúcia, pela ajuda na resolução dos problemas junto ao Programa.

Aos órgãos de fomento: CAPES e FAPERJ, pelo financiamento deste trabalho e ao PESC, pelo apoio operacional.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

PROPOSTA DE UM MECANISMO DE SEGURANÇA BASEADO EM TROCA DE CHAVES ASSIMÉTRICAS PARA REDES TOLERANTES A ATRASOS E DESCONEXÕES

Rafael de Moraes Santos Fernandes

Março/2009

Orientador: Luís Felipe Magalhães de Moraes

Programa: Engenharia de Sistemas e Computação

As redes tolerantes a atrasos e desconexões (*Delay Tolerant Networks* - DTN) provêem um ambiente de desafios, onde a viabilidade de comunicação entre os nós é intermitente. Um desses desafios é o oferecimento de segurança para essas redes. Porém, devido as constantes desconexões, os mecanismos de segurança existentes para redes tradicionais não funcionam adequadamente nas DTNs, pois os mesmos necessitam de comunicação da origem ao destino, o que não é garantido numa DTN. Os trabalhos da literatura relacionados à segurança para DTNs apresentam mecanismos de segurança incompletos; por exemplo, não informando a maneira pela qual os nós trocam as chaves, ou trabalhando com entidades centrais gerenciadoras de chaves. Isso pode ser um problema pois, há alguns pontos de falha na rede (as entidades gerenciadoras de chaves) e não se pode garantir que todos os nós irão obter a chave. Assim os nós podem não se conectar com uma entidade gerenciadora de chaves. Dentro deste contexto, este trabalho propõe um novo mecanismo de segurança para DTNs baseado em trocas de chaves assimétricas. Esta troca de chaves ocorre em cada contato realizado pelos nós da rede. Para garantir a autenticidade, confidencialidade e integridade, foram implementados um algoritmo de criptografia de chaves assimétricas e um mecanismo de assinatura digital em um simulador de DTNs. Além disso, uma avaliação de desempenho de alguns protocolos de roteamento em DTNs foi realizada com o objetivo de verificar o impacto do mecanismo proposto no funcionamento desses protocolos. Os resultados obtidos indicam que o mecanismo de segurança não insere grandes impactos no desempenho da rede e provêm um nível de segurança desejado para DTNs.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PROPOSAL OF A SECURITY MECHANISM BASED ON ASYMETRIC KEYS EXCHANGE FOR DELAY TOLERANT NETWORKS

Rafael de Moraes Santos Fernandes

March/2009

Advisor: Luís Felipe Magalhães de Moraes

Department: Systems Engineering and Computer Science

The Delay Tolerant Networks (DTN) provide an environment of challenges where the viability of communication between nodes is intermittent. One of this challenges is providing security in this networks. However, due to constant disconnections, the existent security mechanisms to traditional networks do not work with property in DTNs, because they need an end-to-end connection, and that is not guaranteed on a DTN. The related literature works on DTN security show incomplete security mechanisms, eg not informing the way the nodes change the security keys, or if they use a central key management. That can be a problem, because they are few failure points and it can not be guaranteed that all nodes will receive the keys. Therefore, a node may not connect with the central key management. Within this context, this work proposes a new security mechanism to DTNs based on asymmetric keys exchange. This exchange occurs in each contact made by two nodes. A cryptography algorithm of asymmetric keys and a digital signature mechanism in a DTN simulator were implemented to ensure its authenticity, confidentiality and integrity. Besides, a performance evaluation of some routing protocols of DTN was carried out to ascertain the impact of the proposed mechanism in this routing protocols. The results indicate that the security mechanism does not insert significant large impacts on the performance of the network and provides a desired security level for DTNs.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACK	:	*Acknowledgment*;
AES	:	*Advanced Encryption Standard*;
ARPANet	:	*Advanced Research Projects Agency Network*;
BAB	:	*Bundle Authentication Block*;
CB	:	*Confidentiality Block*;
DES	:	*Data Encryption Standart*;
DTN	:	*Delay and Disruption Tolerant Network*;
DTNRG	:	*Delay and Disruption Tolerant Network Research Group*;
ENIAC	:	*Electrical Numerical Integrator and Calculator*;
GPS	:	*Global Positioning System*;
HIBC	:	*Hierarchical Based Cryptography*;
IEEE	:	*Institute of Eletrical and Electronics Engineers*;
IRTF	:	*Internet Research Task Force*;
MAC	:	*Message Authentication Codes*;
MEED	:	*Minimum Estimated Expected Delay*;
MMIG	:	Modelo Markoviano de Mobilidade Genérico;
MSE	:	Mean Square Error;
OCB	:	*Offset CodeBook*;
PKG	:	*Public Key Generator*;
PKI	:	*Public Key Infrastructure*;
PROPHET	:	*Probabilistic Routing Protocol using History of Encounters and Transitivity*;
PSB	:	*Payload Security Block*;

RC6 : *Rivest Cipher 6*;

RFC : *Request-for-Comments*;

RSA : *Rivest-Shamir-Adleman*;

RWP : *Random Waypoint*;

SHA : *Secure Hash*;

SW : *Spray and Wait*;

TCP/IP : *Transmission Control Protocol/Internet Protocol*;

The ONE : *The Opportunistic Network Environment simulator*;

TRIPLEDES : *Triple Data Encryption Standart*;

Capítulo 1

Introdução

AS redes tolerantes a atrasos e desconexões (Delay and Disruption Tolerant Networks - DTN) provêem um ambiente de desafios, onde a comunicação entre os nós é intermitente. Isso significa que a conexão entre os nós nessas redes, pode não existir em um determinado intervalo de tempo. Com isso, mecanismos de segurança tradicionais, que necessitam de conexão fim-a-fim, não apresentam um bom funcionamento neste tipo de rede. Este capítulo inicia com um breve histórico sobre o surgimento das redes de computadores. Posteriormente é apresentada a motivação para a realização deste trabalho é apresentada. Em seguida, os objetivos e contribuições desta dissertação de mestrado também são apresentados. Por fim a estrutura de escrita deste trabalho é descrita de maneira resumida.

1.1 Considerações Iniciais

Durante o século XX, principalmente devido as duas grandes guerras mundiais, a busca do homem por soluções tecnológicas foi intensa. Um marco desta busca foi a criação do primeiro computador digital eletrônico de grande escala chamando ENIAC, com objetivo de computar trajetórias táticas durante a II Guerra Mundial, que exija muitos cálculos matemáticos. O mesmo começou a ser desenvolvido em 1943 e se tornou operacional em 1945 [4].

Com o avanço das pesquisas tecnológicas, notou-se rapidamente a necessidade de interação entre os computadores em pontos distintos. Com isso, em 1969 foi lançada a ARPANet que tinha por objetivo conectar as bases militares e os departamentos de pesquisa do governo americano. No início dos anos 1970 universidades americanas, que desenvolviam algum tipo de projeto militar, obtiveram permissão para se conectar a ARPANet. No final dos anos 1970 já haviam diversas instituições conectadas a ARPANet [5].

Pode-se dizer que a grande rede de computadores existente hoje, chamada *Internet*, surgiu da ARPANet e seu sucesso se deve a arquitetura em camadas baseada na pilha de protocolos denominados TCP/IP [6]. Esta arquitetura em camadas foi projetada para operar de forma independente de tecnologia de sub-rede existente, ou seja, deve operar em redes cabeadas, redes sem fio, redes de satélite, redes ópticas, entre outras.

Os atuais mecanismos da pilha de protocolos TCP/IP se baseiam em suposições de redes cabeadas, como existência de conectividade fim-a-fim, pequenos atrasos de comunicação, baixa taxa de erro e mecanismos de retransmissão efetivos para reparar os erros.

No entanto alguns ambientes não possuem tais características, como comunicação de dispositivos móveis sem fio, comunicação em grandes áreas (como ambientes rurais) e comunicações interplanetárias. Para esses ambientes, considerados desafiadores, o uso da pilha de protocolos TCP/IP é considerado inadequado, pois as suposições existentes em uma rede cabeada não são diretamente aplicáveis nestes

cenários desafiadores.

As redes que se adequam a esses cenários desafiadores, onde existe grande dificuldade de se manter uma comunicação fim-a-fim, com pouco atraso e baixa perda de pacotes, são chamadas de Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks* - DTN) [7, 8, 9]. Para contornar esses problemas, as DTNs utilizam as técnicas de comutação de mensagens e armazenamento persistente dos dados. Como a abreviatura "DTN" é um termo consagrado na área de redes de computadores, esta abreviatura será utilizada neste trabalho para se referir as Redes Tolerantes a Atrasos e Desconexões.

No encaminhamento das mensagens numa DTN, cada mensagem recebida é inicialmente armazenada e posteriormente repassada. Este repasse ocorre nó a nó, desde a origem até o destino. Por utilizar esta técnica, as DTNs são chamadas de armazena e encaminha (*store-and-forward*), ou seja, a mensagem primeiro é recebida integralmente, para posteriormente ser repassada para um próximo nó [1].

Como as DTNs não operam sobre enlaces sempre disponíveis, os nós devem armazenar as mensagens geradas por ele ou recebidas de outro nó por um tempo, até existir a oportunidade de repassar a mensagem. Para o armazenamento da mensagem é necessário que os nós possuam algum local de armazenamento, como memórias flash, disco rígido, etc.

1.2 Motivação

O sucesso da Internet foi tão grande, que atualmente milhões de computadores estão ligados por esta rede. Com isso, a Internet vem incorporando diversos serviços importantes para a sociedade, como estudos acadêmicos em escala mundial, comércio eletrônico, comunicação a baixo custo através de voz sobre IP (*Voice Over IP* - VOIP) e vídeo-conferências, que representam alguns dos serviços que fazem parte do cotidiano da sociedade nos dias atuais.

Com o aumento do uso da Internet e a quantidade de serviços utilizados através

dela, também aumenta o número de indivíduos mal intencionados na rede. Pessoas que se aproveitam de vulnerabilidades de segurança nos serviços da Internet para obter benefícios próprios de maneira ilícita, como por exemplo obter o número do cartão de crédito de uma pessoa para efetuar compras.

Um tipo de rede bastante visado por pessoas mal intencionadas são as redes sem fio, devido a facilidade em se obter informações dos usuários, já que o meio de transmissão utilizado neste tipo de rede é o ar atmosférico e não existe uma maneira de restringir a propagação do sinal no ar.

Com isso, a necessidade da utilização de mecanismos de segurança da informação nas redes de computadores, inclusive nas DTNs, é um fator essencial na atualidade. Porém, os mecanismos de segurança projetados para a Internet necessitam de conexão fim-a-fim para garantir a segurança da informação e esta característica nem sempre está presente nas DTNs [10, 11, 12].

Desta forma, torna-se necessário o estudo e a avaliação de novos mecanismos de segurança para DTNs, onde estes mecanismos devem se adequar aos mais diferentes cenários desafiadores na qual as DTNs estão sujeitas a operar.

1.3 Objetivos do Trabalho

Com base no exposto nas seções anteriores, esta dissertação de mestrado tem por objetivo o desenvolvimento e avaliação de um mecanismo de segurança para redes tolerantes a atrasos e desconexões que se adeque aos mais diferentes cenários, nas quais este tipo de rede está sujeito. Este mecanismo de segurança deverá seguir as três premissas básicas de segurança: autenticidade, confidencialidade e integridade.

Além disso, este trabalho visa responder as seguintes perguntas:

- Existe a possibilidade do uso de um mecanismo de segurança nas DTNs?
- Qual o impacto do uso do mecanismo no desempenho da rede?

- Existe variação do desempenho em função do protocolo de roteamento de DTN usado?
- O mecanismo de segurança funciona de forma adequada tanto para cenários de alta densidade como de baixa densidade?

1.4 Contribuições do Trabalho

Dentre as principais contribuições deste trabalho podem ser destacadas as seguintes:

- Elaboração de um mecanismo de segurança para DTNs que segue as três premissas básicas de segurança de redes (autenticidade, confidenciabilidade e integridade);
- Implementação deste mecanismo em um simulador de DTNs;
- Investigação do impacto do uso da segurança proposta em diferentes cenários;
- Utilização de diversas métricas para avaliação do mecanismo de segurança implementado;
- Avaliação do mecanismo de segurança sob a influência de diferentes protocolos de roteamento.

1.5 Organização do Trabalho

Com objetivo de fornecer uma melhor compreensão do restante deste trabalho e facilitar a sua leitura, segue abaixo a estrutura do texto, indicando como esta encontra-se organizada.

As características das DTNs, bem como sua arquitetura são apresentadas no Capítulo 2. Além disso, uma revisão dos trabalhos relacionados a segurança em DTNs,

que é o foco deste trabalho, também é descrita. Por fim, os trabalhos relacionados a segurança em geral também são apontados neste capítulo.

As DTNs foram concebidas para serem móveis. Por isso, o Capítulo 4 aborda os modelos de mobilidade sintéticos utilizados na avaliação deste trabalho. Além disso, nesta avaliação são utilizados rastros de mobilidade de pessoas em um cenário real e o meio de captura e tratamento destes rastros também é explicado no Capítulo 4. Porém, a maneira na qual os nós de uma DTN trocam mensagens também pode influenciar o mecanismo proposto neste trabalho. Então, os principais protocolos de roteamento para as DTNs também são apresentados neste capítulo.

Posteriormente, a técnica de segurança para DTNs proposta neste trabalho é apresentada no Capítulo 3.

No Capítulo 5, uma explicação das implementações feitas no simulador, utilizado para avaliação deste trabalho, é mostrada. Em seguida os cenários, parâmetros e métricas das simulações realizadas são mostrados e explicados. Por fim, os resultados obtidos através das simulações são apresentados e explicados.

Por conseguinte, as conclusões e trabalhos futuros são expostos no Capítulo 6. As referências bibliográficas utilizadas neste trabalho estão presentes após o Capítulo 6.

Este trabalho é finalizado com o Apêndice A, onde os mecanismos de segurança que promovem as três premissas básicas de segurança são apresentados e o Apêndice B que apresenta as características do simulador utilizado na avaliação deste trabalho.

Capítulo 2

As DTNs e Trabalhos Relacionados

ESTE trabalho é focado nas redes tolerantes a atrasos e desconexões (DTNs), portanto este capítulo apresenta uma contextualização das DTNs, apresentando suas características, técnicas e a representação da arquitetura em camadas deste tipo de rede. Em seguida, os trabalhos relacionados à segurança na troca de mensagens em redes sem fio, que na maioria das vezes possuem o foco na distribuição e manutenção das chaves na rede e na complexidade do uso de algoritmos criptográficos, também são apresentados e discutidos. Por fim, os trabalhos relacionados à segurança das DTNs e de outros tipos de redes são apresentados.

2.1 Redes Tolerantes a Atrasos e Desconexões

A arquitetura utilizada na Internet é comprovadamente uma solução de grande sucesso sendo utilizada em todo o mundo. Esta arquitetura se baseia em suposições existentes em redes cabeadas, como existência de conectividade fim-a-fim, pequenos atrasos na comunicação, baixa taxa de erro e mecanismos de retransmissão efetivos para reparar erros, conforme mencionado no capítulo anterior.

Porém, alguns ambientes não estão sujeitos a essas suposições da arquitetura utilizada na Internet. Por exemplo, em uma comunicação sem fio, onde os equipamentos (chamados nós) possuem mobilidade, um nó pode ficar sem comunicação com os outros nós da rede, ocasionando a não existência de conectividade fim-a-fim. As comunicações interplanetárias [13] são outro exemplo, onde os atrasos na comunicação são muito grandes e a taxa de erro é elevada. Outros exemplos são: comunicação rural, comunicação submarina, comunicação entre dispositivos com restrição de energia, etc.

As redes projetadas para funcionar nesses ambientes, considerados desafiadores, foram denominadas de Redes Tolerantes a Atrasos e Desconexões (*Delay Tolerant Networks - DTNs*), conforme descrito em [7, 8] e suas principais características são:

- **Atrasos longos e/ou variáveis** - os atrasos em uma DTN pode variar de alguns milisegundos até horas. O atraso desde o envio de uma mensagem até sua chegada no destino, chamado atraso fim-a-fim, é obtido através da soma dos atrasos obtidos salto a salto e é formado pelas seguintes componentes: tempo de espera de uma conexão, atraso nas filas, atraso de transmissão e atraso de propagação [14];

- **Frequentes desconexões** - As desconexões em uma DTN pode ocorrer pela mobilidade dos nós que provoca constantes mudanças na topologia da rede, por condições adversas de comunicação, economia de recursos dos elementos da rede (como em redes de sensores), por negação de serviço, etc. Devido a esses eventos a conectividade da rede pode ser intermitente, ou seja, pode não

existir um caminho do nó origem ao nó de destino em determinados instantes de tempo.

Para contornar esses problemas de atrasos e desconexões, as DTNs utilizam uma técnica de comutação de mensagens e armazenamento persistente. Na comutação de mensagens, nenhum circuito é estabelecido com antecedência entre a origem e o destino. Quando uma mensagem precisa ser enviada, inicialmente ela é armazenada e encaminhada nó a nó desde a origem até o destino. Pode-se observar na Figura 2.1 um exemplo da troca de mensagens nas DTNs. No tempo t_0 o nó origem iniciou o repasse de uma mensagem para um nó intermediário A. O nó A recebeu e armazenou a mensagem no tempo t_1. Posteriormente, A inicia o repasse da mensagem para o nó intermediário B no tempo t_2. No tempo t_3, B recebe a mensagem e a armazena. No tempo t_4 B inicia o envio da mensagem ao nó destino, que a recebe no tempo t_5. Por enviar as mensagens desta forma, as DTNs são redes do tipo armazena e encaminha (*store-and-forward*), ou seja, em cada repasse da mensagem o nó receptor armazena a mensagem integralmente para posteriormente repassá-la ao próximo nó, até o nó destino. Com isso, não há necessidade do nó destino estar ativo no momento do envio pelo nó origem, pois os nós intermediários armazenam a mensagem e podem entregar ao destino posteriormente.

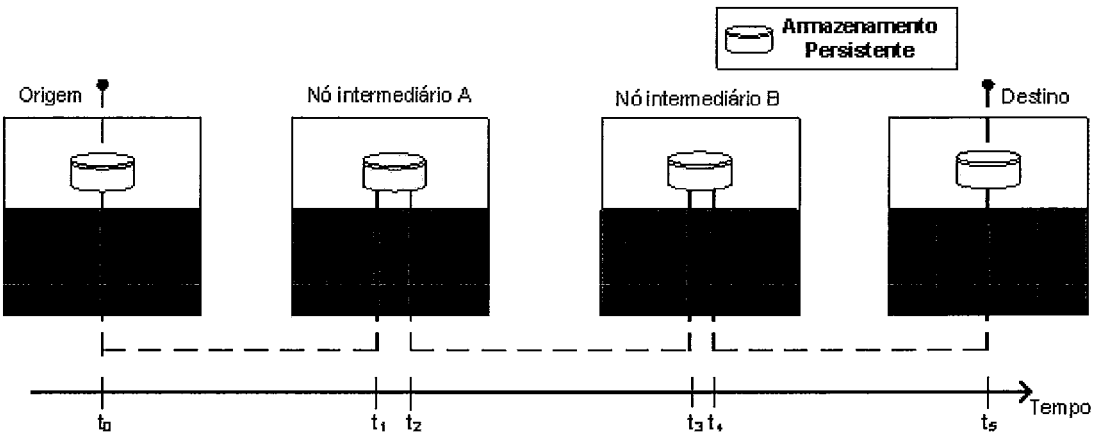

Figura 2.1: Exemplo de rede do tipo armazena e encaminha, com armazenamento persistente.

Como as técnicas de comutação de mensagens e armazenamento persistente são mandatórias em DTN, foi definida uma sobrecamanda (*overlay*) abaixo da camada

de aplicação, responsável pela comutação das mensagens e armazenamento persistente [13]. Esta camada é chamada Camada de Agregação (*Bundle Layer*) e as camadas de uma DTN podem ser observadas na Figura 2.2.

Figura 2.2: Camadas da arquitetura DTN.

As características da camada de agregação são [1]:

- Armazenamento persistente;
- Decisões de roteamento;
- Compartilhamento dos dados agregados entre as aplicações;
- Adaptar a camada de convergência a diferentes protocolos.

A Figura 2.3, retirada de [1] ilustra as características da camada de agregação.

Após a descrição das DTNs e de suas características, serão apresentados os principais trabalhos sobre segurança nessas redes.

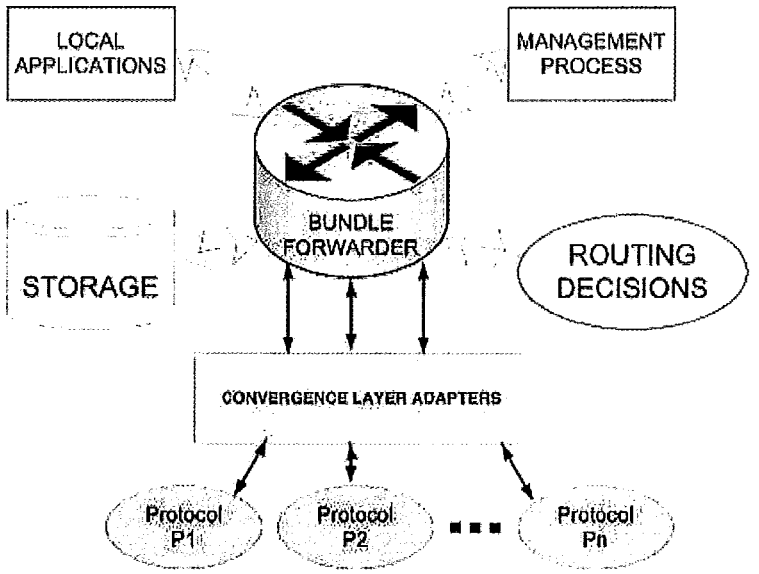

Figura 2.3: Características da camada de agregação das DTNs. Figura retirada de [1].

2.2 Segurança em Redes Tolerantes a Atrasos e Desconexões

Em [10, 11], uma arquitetura de segurança baseada em *Hierarchical Based Cryptography* (HIBC) para DTNs é proposta. A técnica de HIBC pode ser vista como uma árvore, onde a raiz dessa árvore é o provedor de chaves (*Private Key Genarator* - PKG) para seus nós filhos, ou seja, é o responsável por informar aos nós abaixo dele sua chave privada [15]. Ele também é responsável por informar a chave pública de qualquer nó abaixo dele, caso isso seja requisitado.

Para evitar sobrecarga na raiz, qualquer nó não folha pode ser um provedor de chaves para os nós abaixo dele. Um provedor de chaves sempre será a raiz de uma árvore e esta pode ser sub-árvore de uma árvore maior. As folhas das árvores são os usuários da rede, que necessitam das chaves para enviar suas mensagens criptografadas. Dado um nó provedor de chaves a, sua árvore A e seu nó filho a1, não folha, provedor de chaves da sub-árvore A.1, diz-se que a árvore A define uma região e a árvore A.1 define uma sub-região de a. A Figura 2.4 apresenta um exemplo

da estrutura do HIBC.

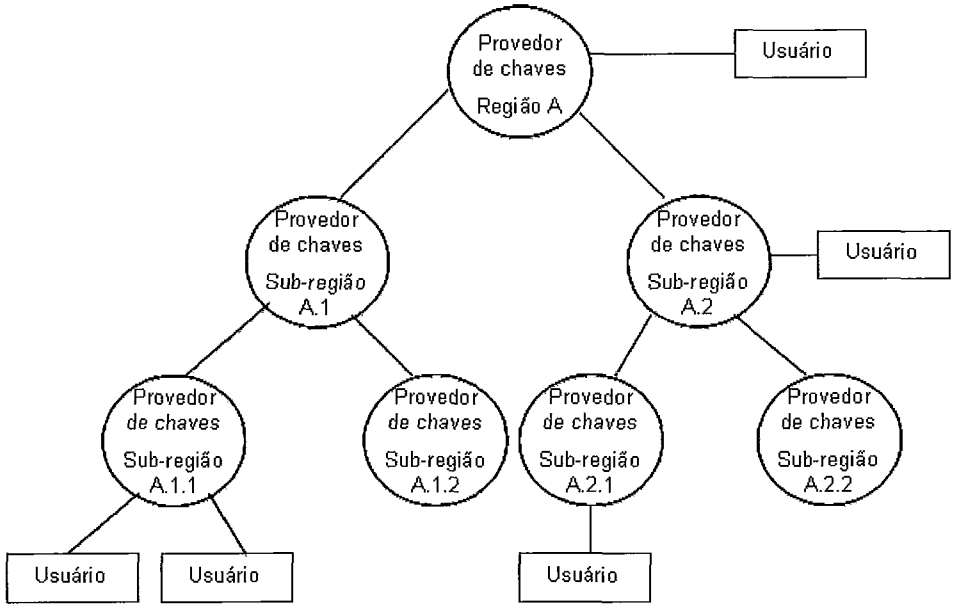

Figura 2.4: Exemplo de estrutura em árvore da técnica *Hierarchical Based Cryptography* (HIBC)

Se um usuário da sub-região A.1.1 necessita enviar uma mensagem para um usuário da sub-região A.2, ele envia a mensagem para o provedor de chaves da sua sub-região (A.1.1) cifrando a mensagem com a chave pública do provedor. O provedor de chaves irá decriptografar a mensagem e criptografá-la com a chave pública do provedor de chaves da sub-região A.1, e irá repassar a ele. O provedor de chaves da sub-região A.1, após decriptografar a mensagem, irá criptografá-la com a chave pública do provedor de chaves da região A, que irá criptografar a mensagem com a chave pública do provedor de chaves da sub-região A.2, após decriptografar a mensagem. Por fim, o provedor de chaves da sub-região A.2 decriptografa a mensagem e criptografa com a chave pública do destinatário, que após receber a mensagem poderá decriptografá-la com sua chave privada.

Nesta solução, se um provedor de chaves parar de funcionar, todos os usuários da sua região não conseguirão obter suas chaves e todos os nós da sua região, e sub-regiões abaixo dele, não poderão enviar mensagens cifradas para as regiões vizinhas. Outro problema ocorre se o provedor de chaves for comprometido, com isso, toda a segurança de sua região e sub-regiões abaixo dele ficará vulnerável. Por fim, o

fato de o provedor de chaves decriptografar uma mensagem para repassá-la à uma região vizinha, pode ocasionar mensagens não confiáveis na rede, pois o mesmo pode alterar o conteúdo da mensagem e repassá-la.

Uma comparação do uso de mecanismos de criptografia tradicionais, como o PKI (*Public Key Infrastructure*), e a técnica de HIBC é apresentada em [12]. Em se tratando de autenticação dos nós e integridade das mensagens, os resultados mostraram que o HIBC e as técnicas de criptografias tradicionais apresentam resultados semelhantes. Porém, em termos da confidencialidade da mensagem, a técnica de HIBC adota uma solução mais eficiente, em se tratando de carga nos servidores que provêem as chaves, pois o número de chaves geradas no HIBC é proporcional ao número de destinatários numa rede. Já na criptografia tradicional, o servidor tem que decriptografar cada mensagem recebida. Assim, o número de chaves geradas é proporcional ao número de emissores e do número de mensagens geradas.

Os protocolos de roteamento para DTNs foram desenvolvidos com a expectativa de que os nós estão sempre indisponíveis. Assim, ataques a DTN não são tão efetivos conforme apresentado em [16]. Ataques do tipo *dropping*, onde um nó malicioso apaga todos os pacotes que ele recebe; falsificação de tabelas de rotas, onde um nó malicioso pode enviar mensagens de controle falsas que fazem com que um nó repasse uma mensagem erroneamente para um outro nó; propagação de ACKs, onde um nó malicioso envia mensagens do tipo ACKs (*Acknowledgment*) confirmando a entrega de uma mensagem, na qual a mesma não foi efetivamente entregue; e ataques do tipo inundação de pacotes, onde um nó malicioso envia vários pacotes na rede, de maneira a congestionar a rede e no caso de DTN, ocasionar perda de pacotes por estouro de *buffer*.

Uma maneira de evitar esses ataques é efetuar a autenticação dos nós na rede. Porém, como a DTN não é muito suscetível a esses ataques (devido ao roteamento), o artigo [16] demonstra que não há necessidade de se efetuar autenticação, pois nem todos os nós que estão entrando na rede são atacantes e eles ajudam no roteamento. Os resultados apresentam que o pior ataque, em relação a entrega de pacotes, foi o tipo inundação de pacotes onde se 30% da rede for de nós atacantes a probabilidade

de entrega das mensagens é de 45% contra 70% se não houver ataques na rede. O trabalho ainda analisa a robustez dos protocolos de roteamento encaminhadores, que repassam as mensagens originais, e dos protocolos replicadores, que repassam cópias das mensagens. Os protocolos replicadores apresentaram uma robustez maior que os encaminhadores, para a maioria dos tipos de ataques.

Um grupo de pesquisa do *Internet Research Task Force* (IRTF) [17], chamado *Delay Tolerant Networking Research Group* (DTNRG) [18], tem o objetivo de estudar e criar padrões de interconexão de redes de cooperação altamente heterogêneas, mesmo que a comunicação fim-a-fim nunca possa ser disponibilizada, como no caso das DTNs. Uma das linhas de pesquisa deste grupo de trabalho é relacionada a segurança para DTNs. Porém, até o momento, a contribuição para segurança deste grupo de trabalho foi a especificação de três camadas independentes de segurança específicas para cada bloco de agregação (*bundle blocks*), que podem ser usados em conjunto para fornecer vários serviços de segurança ou independentemente um do outro [19]. O Bloco de Autenticação Agregado (*Bundle Authentication Block* (BAB)), assegura a autenticidade e integridade das mensagens salto a salto. O BAB permite que cada nó verifique a autenticidade da mensagem antes de repassá-la. Para promover segurança da mensagem da origem ao destino, o Bloco de Segurança da Carga Útil (*Payload Security Block* (PSB)) garante que a mensagem foi realmente enviada pelo emissor. Por fim, o Bloco de Confidencialidade (*Confidentiality Block* (CB)), assegura que somente o destino poderá obter a informação enviada na mensagem. Em [20], uma análise da segurança para essas redes, baseada nas decisões do DTNRG expostas anteriormente, é apresentada onde a principal crítica deste trabalho é a falta de definição de um mecanismo de distribuição de chaves nessas redes pelo DTNRG.

Foram apresentados nessa seção os trabalhos relacionados com segurança de DTNs. O principal problema destes trabalhos se deve à falta de um mecanismo de troca de chaves ou o uso de técnicas de troca de chaves que apresentam alguns problemas, como o HIBC. Na seção seguinte serão expostos os trabalhos referentes a segurança em outros tipos de rede que podem auxiliar em uma solução de segurança para DTNs.

2.3 Segurança em outros tipos de rede

Devido as dificuldades de se implementar mecanismos de segurança em redes de sensores e devido a características de persistência dos dados, mesmo que poucos dados e durante pouco tempo, e restrição de memória; o estudo de segurança para as redes de sensores foi de grande importância no desenvolvimento deste trabalho. As DTNs podem ser consideradas um tipo de rede ad hoc, por isso propostas de segurança em redes ad hoc também foram estudadas neste trabalho. Os trabalhos de redes de sensores e redes ad hoc que mais se adequam a este trabalho estão citados e comentadas a seguir.

Em [21], os autores desenvolveram uma camada de segurança para redes de sensores baseadas nas primitivas de segurança. São utilizadas técnicas de *Message Authentication Codes* (MACs) e de criptografia para proteger as mensagens trocadas. São usadas 2 técnicas: *Authenticaded Encrypton* - onde se faz a encriptação do *payload* da mensagem e posteriormente se computa o MAC do *payload* criptografado e *Authentication Only* - onde somente o MAC do *payload* é feito. O trabalho [22] segue a mesma lógica do artigo descrito em [21] porém, um algoritmo de *Offset Code Book* (OCB) é utilizado na encriptação dos dados. Com esta mudança o artigo [22] apresenta resultados melhores de segurança em relação ao [21], porém, nenhum dos dois trabalhos desenvolveram um mecanismo de distribuição de chaves. Eles partem da premissa de que todos os nós possuem as chaves de todos os outros nós.

O trabalho [23] também segue as técnicas de segurança de [21], porém determina que a troca de chaves é feita através de uma entidade central. Cada nó deve requisitar uma chave da unidade central, que enviará uma "chave mestra" e cada nó irá derivar suas chaves a partir da "chave mestra". Para se ter acesso a entidade central, é necessário uma comunicação fim-a-fim entre o nó requisitante e esta unidade, o que não é possível de se garantir em uma DTN e ainda, o comprometimento desta unidade central afeta toda a segurança da rede. Por fim, dependendo do tamanho da rede, esta unidade central pode ficar sobrecarregada de requisições de chaves, agregando atrasos na rede e possíveis perdas de pacotes.

Outra técnica para obtenção de chaves com segurança é a chamada Distribuição de Chaves Pré-estabelecida (*Predeployed Key Distribution*), onde os nós já possuem uma chave pré-definida no momento da sua fabricação. Assim, quando uma rede é estabelecida, os nós já possuem material criptográfico dos outros nós e com isso a comunicação segura entre eles é garantida. Uma variante disso é a escolha de diferentes chaves dependendo do instante de tempo, como proposto em [24], onde a chave escolhida por um nó dentro de um conjunto de chaves, depende diretamente do momento da escolha. O problema é a inviabilidade de haver uma chave comum a todos os dispositivos, pois apenas um dispositivo comprometido afetará toda a rede.

O uso de criptografia assimétrica em redes de sensores não é aconselhável, pois este tipo de criptografia necessita de mais computação na geração de chaves, na criptografia e decriptografia de mensagens, do que a criptografia simétrica. Além disso, a criptografia assimétrica necessita do uso de chaves longas, requerendo mais armazenamento e capacidade de transmissão dos nós sensores. Essa necessidade de uso de criptografia simétrica é apresentada em [25], onde a técnica de Distribuição de Chaves Pré-estabelecida, citada anteriormente, também é analisada. Porém, em uma DTN, o fato da possível diversificação de dispositivos usados nesta rede (mais processamento, memória e capacidade de transmissão) e da característica de armazenamento da mesma, fazem com que existam menos restrições ao uso de criptografia assimétrica nesse tipo de rede.

Uma classificação dos mecanismos de distribuição de chaves em redes ad hoc é proposta em [26]. São definidos 3 grupos principais: os **protocolos de gerenciamento centralizado de chaves**, onde uma única entidade é responsável por controlar um grupo de nós e gerenciar suas chaves; as **arquiteturas descentralizadas**, onde o gerenciamento é distribuído entre sub-grupos e cada sub-grupo possui uma entidade que é responsável pelo gerenciamento do seu sub-grupo; e os **protocolos de gerenciamento distribuído de chaves**, onde todos os nós do grupo são responsáveis por gerenciar a chave do grupo. Este trabalho ainda cita regras para um gerenciamento seguro de chaves como, **prover autenticação e identificação dos membros do grupo**, para prevenir que um invasor se passe por um nó legítimo do grupo ou ainda por um nó gerenciador de chaves de um grupo, **controle**

de acesso, usado para validar um nó antes do mesmo ter acesso a comunicação com outros nós do grupo e **geração, distribuição e instalação de chaves**, pois é necessário alterar as chaves de tempos em tempos para garantir a segurança da mesma.

O uso de uma entidade central para a distribuição de chaves em uma rede ad hoc pode causar atrasos. No trabalho [27], os autores demonstram que os nós recebem as chaves necessárias a criptografia mais rapidamente em um contexto distribuído em detrimento a um contexto centralizado. É mostrado que a mobilidade dos nós provê uma maior disseminação das informações criptográficas do que se os nós necessitassem de acessar uma entidade central para obter esta informação. É mostrado, através de simulações, que após 1000 segundos, 70% da segurança já havia sido estabelecida numa rede com 100 nós, 5 metros de raio de transmissão em cada nó, $1km^2$ de área e distribuição de chaves descentralizada, contra 10.000 segundos na distribuição centralizada para o mesmo cenário.

Nesta seção alguns trabalhos acadêmicos que propõem mecanismos de segurança para redes sem fio foram citados. Estes trabalhos foram importantes na definição do mecanismo de segurança proposto no Capítulo 3. No próximo capítulo as premissas de segurança e as técnicas utilizadas para garantir essas premissas serão apresentadas.

Capítulo 3

Mecanismo de Segurança Proposto para DTNs

AS DTNs possuem a característica de armazenamento persistente das mensagens. Esta característica permite que os nós da rede armazenem as mensagens quando o mesmo está desconectado dos outros nós da rede. Porém, esta característica pode ser usada também quando os nós necessitam de alguma outra informação para repassar a mensagem, como por exemplo a chave criptográfica do nó destino. Assim, este capítulo apresenta o mecanismo de segurança proposto neste trabalho. Inicialmente, o mecanismo de troca de chaves proposto é apresentado, seguido de como os nós procedem para enviar uma mensagem. Além disso, o procedimento de recepção de mensagens é apresentado e por fim, a apresentação da estrutura das mensagens utilizadas neste trabalho.

3.1 Mecanismo Proposto para a Troca de Chaves

Como descrito na Seção A.1, os algoritmos de criptografia necessitam de um par de chaves para efetuar a cifra, ou seja, um nó necessita saber a chave pública do outro nó destinatário, para poder cifrar sua mensagem e enviá-la de modo confidencial. Portanto, os nós em uma rede necessitam obter as chaves dos outros nós da rede para que possam enviar suas mensagens de forma confidencial. Nas redes móveis, existem duas maneiras de se obter as chaves: através de uma entidade central, onde esta entidade é um porto seguro e possui todas as chaves de todos os nós. Um exemplo dessa maneira é o *Private Key Generator* (PKG), onde geralmente, uma entidade gera as chaves de todos os nós da rede. Portanto, esta entidade sabe as chaves de todos os nós. Outro exemplo é citado no Capítulo 2, onde a técnica de *Hierarchical Based Cryptography* (HIBC), utiliza uma entidade central em cada sub-região para propagar aos seus nós as chaves que eles necessitam.

Outra maneira de efetuar a troca das chaves é cada nó gerenciar sua chave e repassá-la para seus vizinhos e o mesmo as repassa. Se a técnica de criptografia for assimétrica, as chaves podem ser repassadas pelo meio, sem necessidade de um canal seguro, pois se um atacante estiver "escutando" o canal, ele não poderá descobrir o segredo da criptografia apenas com a chave pública.

Porém, na criptografia simétrica existe a necessidade da troca de chaves ser efetuada de maneira segura pois, se um atacante descobrir a chave, ele pode decriptografar facilmente a mensagem. No Capítulo 2, foram descritas algumas soluções para a troca de chaves seguras em um esquema criptográfico simétrico, como exemplo, pode-se citar o uso de chaveiros USB para obtenção de chaves e chaves pré inseridas nos nós. Essas soluções tentam obter um canal inicialmente seguro, para que os nós possam trocar suas primeiras chaves, porém essas soluções não são consideradas boas, conforme comentado no Capítulo 2

Com isso, neste trabalho, resolveu-se usar o mecanismo de troca de chaves públicas e privadas (criptografia assimétrica), pois neste mecanismo não há a necessidade de se estabelecer um canal seguro para a troca de chaves.

O problema de se adotar técnicas de troca de chaves onde não existe uma entidade central de armazenamento é que no momento do envio de uma mensagem, o nó emissor pode não ter a chave pública do nó destino, portanto a mensagem não poderia ser enviada e conseqüentemente seria descartada.

Este trabalho propõe o uso de mecanismos de troca de chaves em DTNs que possuem por característica a persistência da informação. Assim, se uma mensagem não puder ser enviada, o nó a armazena até obter a chave para cifrá-la e poder enviar a mesma ou, se a memória de armazenamento do nó (*buffer*) ficar completa, a mensagem é descartada. Portanto, existe um tempo na qual a mensagem pode esperar pela chave para ser enviada, ao contrário de outros tipos de redes que removem a mensagem se não souber a chave do destino.

As DTNs também são caracterizadas pela não existência de conexão fim à fim. Conexão fim à fim pode ser definida como a habilidade de enviar uma mensagem para um destino e receber uma resposta imediatamente [12]. Portanto, não é possível obter a chave de um destinatário no momento do envio.

Outro desafio importante em DTNs é o repasse de mensagens. Como não existe comunicação fim à fim, não é possível saber a *priori* se a mensagem chegará ao destino. Para isso, diversos protocolos de roteamento foram criados, com o objetivo primário de obter um maior número de mensagens entregues e um menor número de mensagens descartadas.

Então, dado que nas DTNs se espera que existirá um caminho da origem ao destino (independente do tempo que leve para haver este caminho), para a distribuição de chaves proposta neste trabalho, pressupõe-se que existirá uma rota do destino a origem (independentemente do tempo que leve para haver esta rota). Portanto, a técnica de troca de mensagens apresentada neste trabalho pode ser vista como um roteamento do destino para origem, pois um nó irá propagar suas chaves com o intuito do nó que desejar enviar uma mensagem para ele, possuir sua chave pública.

No Capítulo 2, foram apresentados os resultados dos trabalhos [10] e [27]. Pode-se observar nos trabalhos sobre HIBC, que técnicas de troca de chaves centraliza-

das e descentralizadas possuem alguns pontos de falha, podendo comprometer a segurança da rede. Já em [27], mecanismos distribuídos conseguem disponibilizar material criptográfico, no caso as chaves, de forma mais rápida do que mecanismos centralizados. Em [12], os autores demonstram que os mecanismos tradicionais de segurança possuem bom desempenho em DTNs. Portanto, neste trabalho, será utilizada uma arquitetura distribuída, onde o material criptográfico será repassado no contato dos nós.

Define-se que um nó está em contato com outro nó se, sejam os nós $n1$ e $n2$, p_{n1}(t) e p_{n2}(t) suas respectivas posições no instante t e suas áreas de cobertura A_{n1} e A_{n2} . Se no instante t, $p_{n2}(t) \in A_{n1}$ e $p_{n1}(t) \in A_{n2}$ diz-se que $n1$ e $n2$ estão em contato no instante t.

Se dois nós estiverem em contato um com o outro, existe uma possibilidade de troca de informações entre eles, pois se o canal de comunicação estiver sendo usado (meio ocupado), a comunicação não ocorrerá.

O mecanismo de troca de chaves proposto, utiliza esses contatos entre os nós para efetuar a troca de material criptográfico e o armazenamento das chaves já recebidas de cada nó. O fluxograma da Figura 3.1 apresenta os passos da troca de chaves. Esse processo de troca de chaves ocorre da seguinte maneira: sempre que dois nós acabam de estabelecer uma conexão e o meio de comunicação está livre, eles trocam suas chaves e repassam as chaves que cada um já recebeu de contatos anteriores. Esse processo de troca se inicia com um dos dois nós requisitando o meio. Seja i o nó que conseguiu alocar o meio de comunicação. O nó i envia uma mensagem para o outro nó j, com uma lista contendo a identificação dos nós que ele possui a chave em sua memória. Essa mensagem é chamada de "Lista de Nós". O nó j, ao receber a "Lista de Nós", irá verificar em sua memória os nós na qual ele possui a chave pública. Então j irá enviar uma mensagem para i contendo a "Lista de Nós" dele, acrescida de uma lista contendo a identificação e a chave pública dos nós que j possui, mas i não possui. Essa segunda lista é chamada de "Lista de Chaves". Assim que i receber a mensagem de j com a "Lista de Chaves" e a "Lista de Nós" de j, irá atualizar a sua "Lista de Chaves" com a "Lista de Chaves" recebidas de j e

irá comparar a "Lista de Nós" recebida de j com os nós na qual ele possui a chave pública. Após esta comparação, i irá enviar a "Lista de Chaves" que ele possui, mas j não possui. Por fim, j, ao receber a "Lista de Chaves" de i, atualizará as chaves que ele possui em memória. No final da troca de chaves, i e j possuirão as chaves dos mesmos nós na memória.

Uma otimização desta troca de mensagens é a "Lista de Nós", pois ela serve para que a mensagem tipo "Lista de Chaves" seja reduzida, pois o nó só irá repassar as chaves que o outro nó não possui e ainda, se um nó que recebeu uma "Lista de Nós" verificar que o nó que enviou a "Lista de Nós" possui todas as chaves que ele possui, não será enviada uma mensagem tipo "Lista de Chaves", diminuindo o número de mensagens de segurança na rede. Esta otimização foi baseada na troca de mensagens do protocolo de roteamento epidêmico.

Se, no momento do estabelecimento do contato o meio estiver ocupado, os nós guardarão as mensagens contendo suas "Lista de Nós" no *buffer* de mensagens e esta será enviada como uma mensagem de dados (que não é tipo "Lista de Nós" nem "Lista de Chaves") quando o meio estiver livre, sendo que mensagens do tipo "Lista de Nós" e "Lista de Chaves" só são repassadas para o nó destino. O mesmo ocorre, se o meio estiver ocupado no envio de qualquer mensagem referente a troca de mensagens de segurança. Se um nó receber uma mensagem do tipo "Lista de Nós" ou "Lista de Chaves", ele irá proceder com a atualização ou envio das chaves recebidas.

3.2 Mecanismo Proposto para o Envio de Mensagens

No momento que um nó cria uma mensagem para ser enviada a um destinatário, o nó verifica se ele possui a chave pública do destino. Se possuir, ele criptografa o *payload* da mensagem, calcula o hash do *payload* criptografado, criptografa o hash com sua chave privada e coloca a mensagem no *buffer* de mensagens a serem

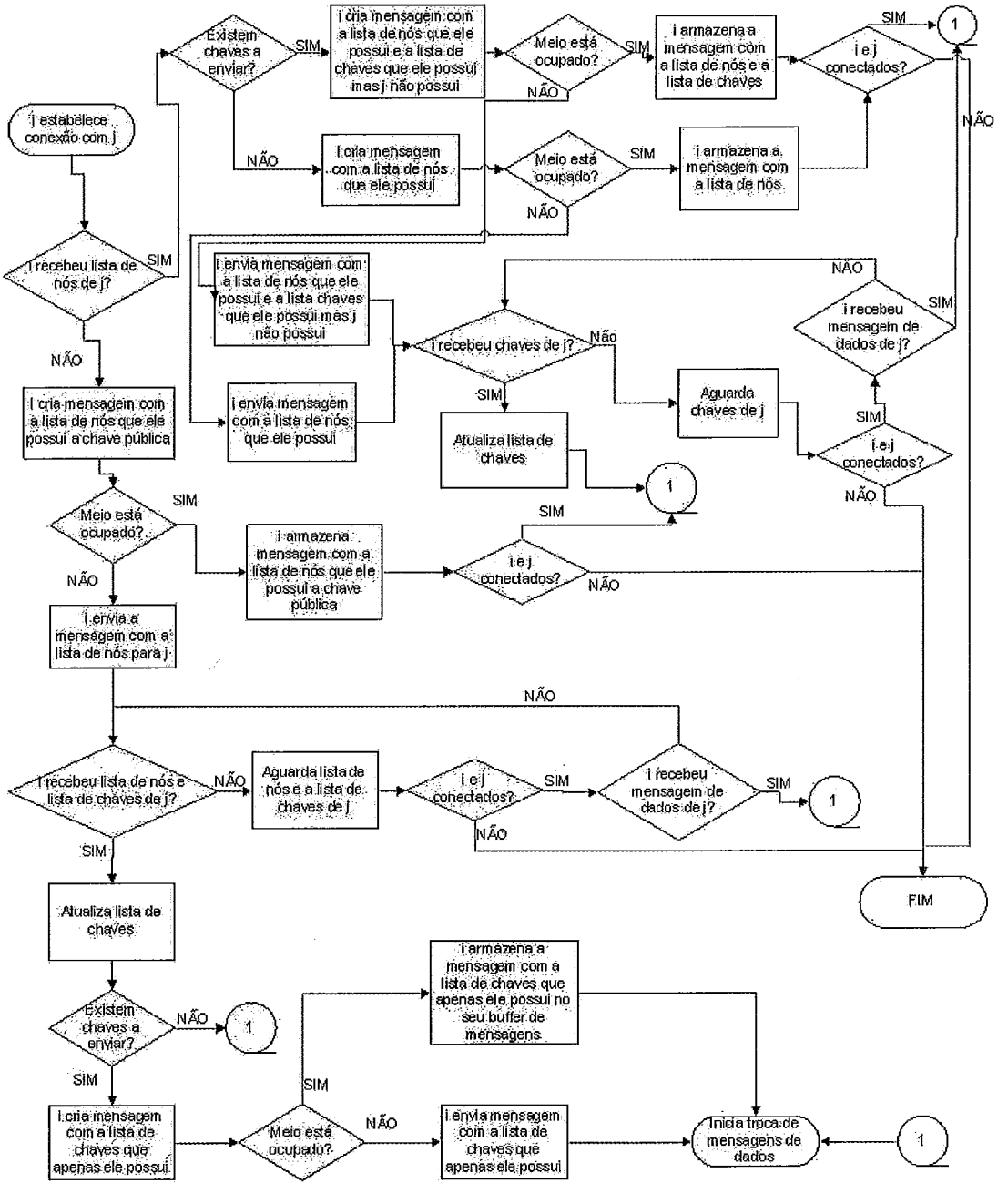

Figura 3.1: Fluxograma do mecanismo proposto para a troca de chaves

repassadas posteriormente. Caso o nó não possua a chave pública do nó destino, ele irá armazenar a mensagem no *buffer* de mensagens, a espera da chave pública do destino.

No momento do envio de uma mensagem, o nó inicialmente verifica se a mensagem é do tipo "Lista de Nós" ou "Lista de Chaves", se for o nó não verifica se a mensagem já foi cifrada, pois esses tipos de mensagens não são cifradas. Caso a mensagem não seja do tipo "Lista de Nós" ou "Lista de Chaves", o nó checa se ela já está criptografada, se não estiver, ele tenta criptografá-la; se conseguir, calcula o hash do *payload* criptografado, criptografa o hash com sua chave privada e envia a mensagem normalmente; caso contrário, deixa a mesma no *buffer* para tentar ser criptografada futuramente ou até ser eliminada do *buffer*.

O fluxograma da Figura 3.2 mostra os passos do envio de mensagens.

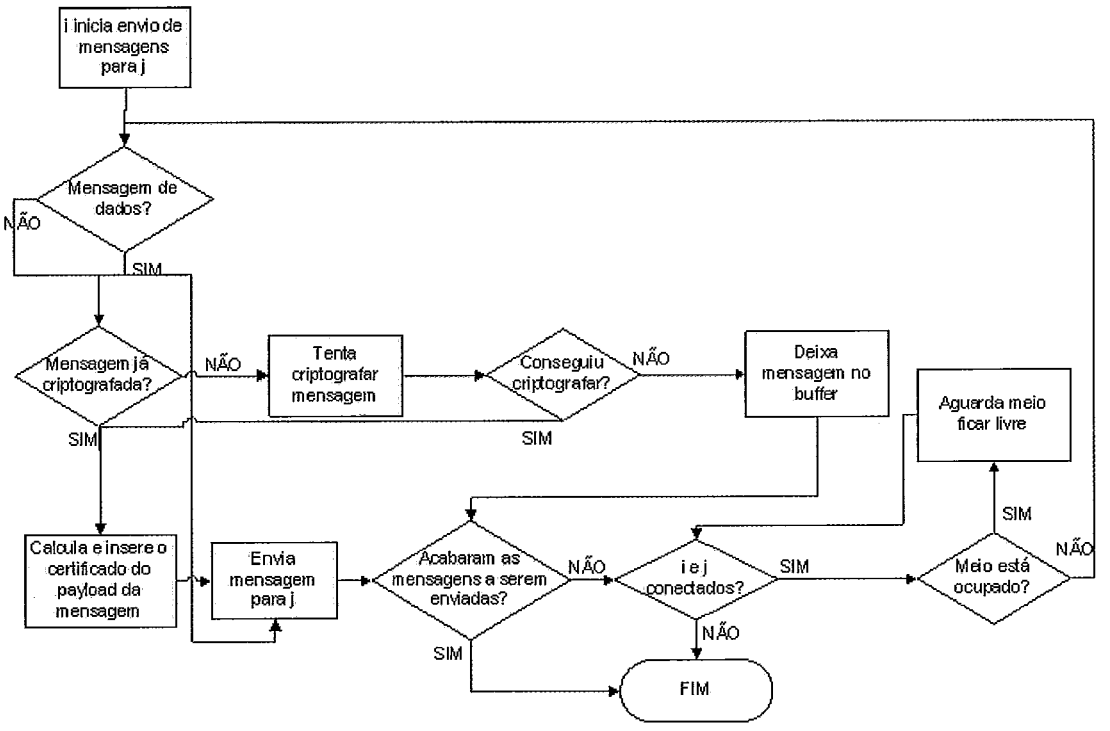

Figura 3.2: Fluxograma do envio de uma mensagem

3.3 Mecanismo Proposto para a Recepção de Mensagens

Após receber uma mensagem, o nó deve verificar o seu tipo. Se for do tipo "Lista de Nós" ou "Lista de Chaves", o nó irá proceder conforme o fluxograma da Figura 3.1 (troca de chaves). Caso contrário, o mesmo irá verificar se possui a chave pública do nó emissor, para verificação da autenticidade e integridade da mensagem. Se não possuir, ele verificará se é o destino. Se ele não for o destino, simplesmente coloca a mensagem no *buffer* de mensagens para ser repassada, sem checar a integridade e a autenticidade da mensagem. Se ele for o destino, irá colocar a mensagem em um *buffer* de mensagens recebidas e sempre que atualizar as chaves na memória, irá verificar se ele recebeu a chave pública do emissor, com intuito de verificar a autenticidade e integridade da mesma. É esperado que os nós sempre possuam a chave pública do emissor, já que as chaves são sempre trocadas antes das mensagens. Se o nó possuir a chave pública do emissor, ele irá chegar se a mensagem é autêntica e se o hash do pacote está correto. Se a mensagem não for autêntica ou o hash estiver errado, o nó irá descartar a mensagem. Caso contrário, o nó irá verificar se ele é o destinatário da mensagem. Se não for, ele irá armazenar a mensagem no seu *buffer* de mensagens. Porém, se ele for o destinatário da mensagem, irá decriptografar a mensagem com sua chave privada. Caso tenha sucesso, o nó irá armazenar a informação recebida, caso contrário irá descartar o pacote.

O fluxograma da recepção de uma mensagem está apresentado na Figura 3.3.

3.4 A Estrutura das Mensagens

A estrutura das mensagens utilizadas nesse trabalho é ilustrada na Figura 3.4. Os campos da mensagem foram criados tentando seguir a estrutura da mensagem proposta pelo DTNRG apresentado no Capítulo 2 e serão definidas a seguir.

Os campos **origem** e **destino** contêm a identificação dos nós de origem e destino,

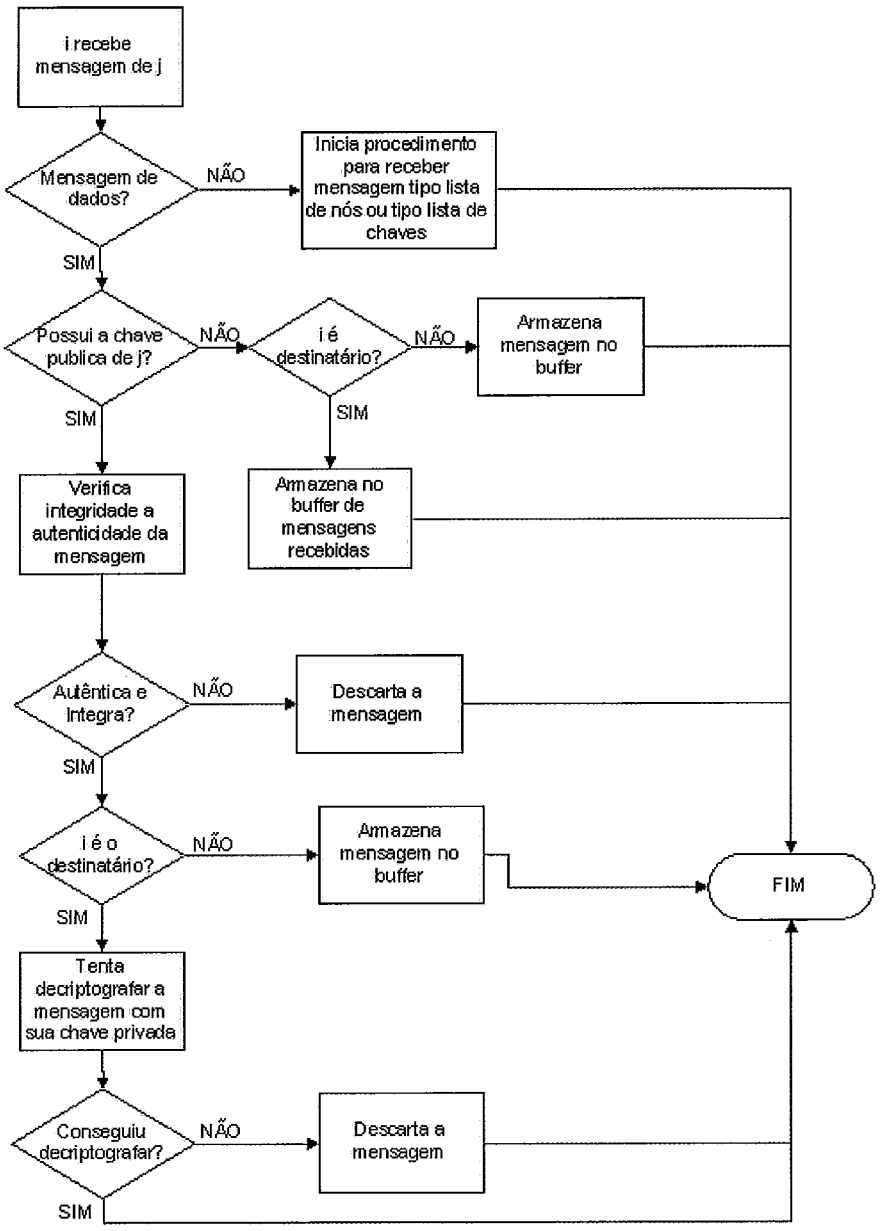

Figura 3.3: Fluxograma da recepção de uma mensagem

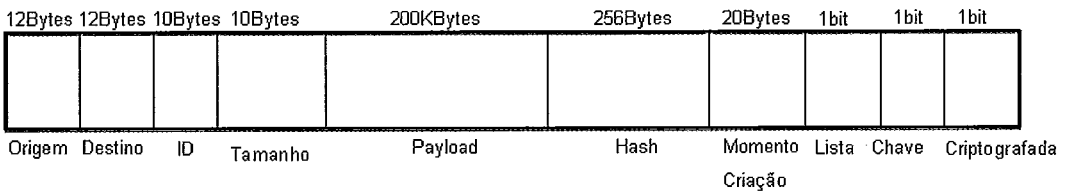

Figura 3.4: Estrutura das mensagens utilizadas neste trabalho

respectivamente. O campo **ID** é um identificador da mensagem. Cada mensagem na rede possui um único identificador. Em **tamanho**, o tamanho total da mensagem é informado, pois cada mensagem possui seu tamanho próprio. O campo ***payload*** possui a informação útil da mensagem. Este campo é criptografado quando a mensagem é do tipo normal ou possui a lista de nós ou lista de chaves, se a mensagem for utilizada para troca de chaves. O campo **hash** possui a criptografia do hash do campo *payload* criptografado com a chave privada do emissor, se a mensagem for do tipo normal. O instante em que a mensagem é criada é inserida no campo **momento criação**. Os campos **lista, chave e criptografada** são campos de controle para informar se a mensagem é do tipo "Lista de Nós", "Lista de Chaves" ou se a mensagem já foi criptografada, respectivamente.

Assim como proposto pelo DTNRG, neste trabalho a segurança está na camada de agregação da arquitetura DTN. No envio das mensagens agregadas, as mesmas são criptografadas e é calculada a assinatura digital antes de passar para a camada de transporte. No ato da recepção de uma mensagem, a assinatura digital da mesma é verificada e se a mensagem for autêntica e íntegra, será armazenada. Além disso, se o nó que está recebendo a mensagem for o destino, a mensagem será decriptografada e repassada para a camada de aplicação. No contexto de troca de chaves a camada de agregação, além de guardar as chaves; a criação e comparação das listas de nós e listas de chaves ocorrem também nesta camada. A Figura 3.5 apresenta os mecanismos de segurança utilizados nesta proposta e presentes na camada de agregação.

Na Figura 3.6 pode-se observar o percurso de uma mensagem pelas camadas da arquitetura de uma DTN desde a origem, passando pelos nós intermediários e chegando ao destino.

Quando uma informação for gerada pela camada de aplicação do nó origem, esta será repassada para a camada de agregação. A camada de agregação irá verificar se o nó origem possui a chave do nó destino para criptografar a informação. Se possuir, além de criptografar a mensagem, será calculada e armazenada a assinatura digital da mesma. Se não possuir, a informação será armazenada como texto plano até que o nó origem possua a chave do destino ou que a mensagem seja descartada. Se a

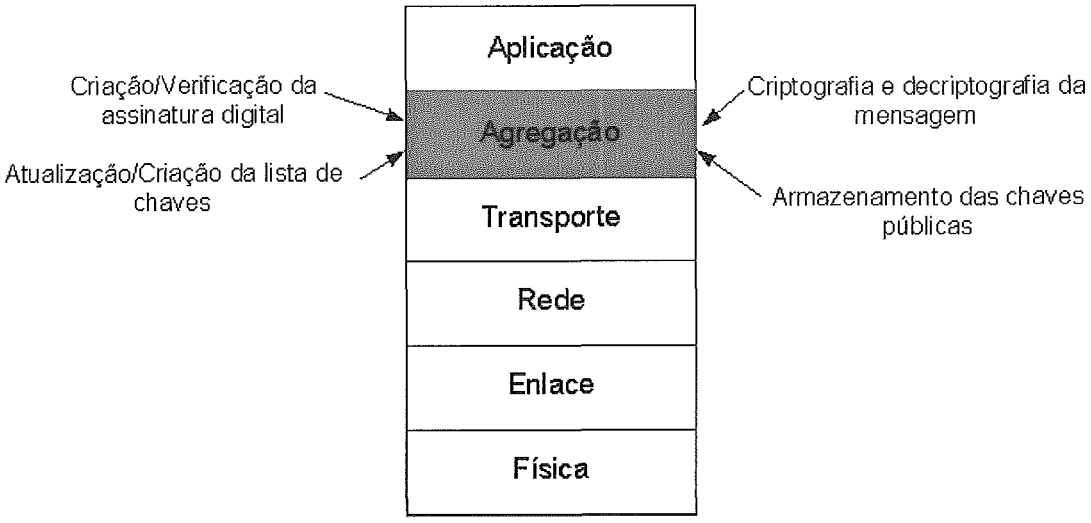

Figura 3.5: Os mecanismos da proposta de segurança presentes na camada de agregação.

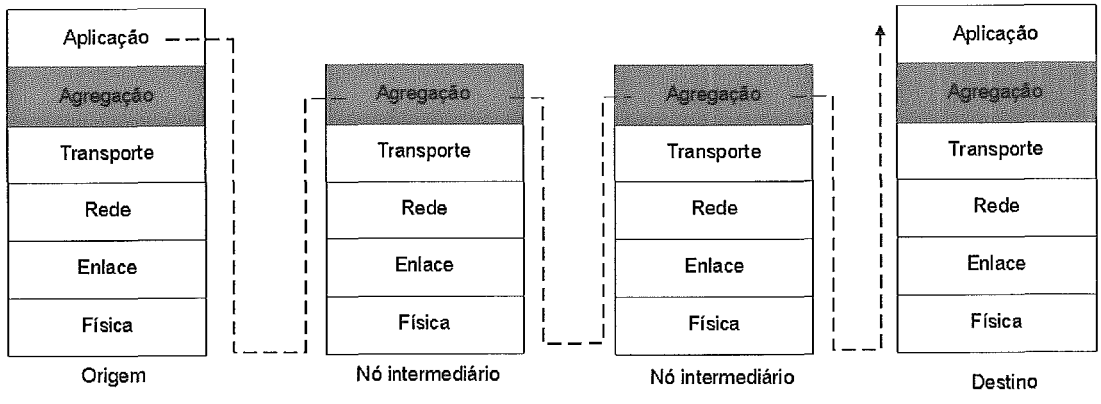

Figura 3.6: Percurso das mensagens entre as camadas da arquitetura DTN, desde a origem até o destino, passando pelos nós intermediários.

mensagem já estiver cifrada e houver uma oportunidade de repasse da mensagem as outras camadas da arquitetura, abaixo da camada de agregação, irão inserir seus campos e criar uma mensagem (pacote) para ser enviada para o próximo nó. Quando o nó intermediário receber a mensagem, irá retirar as informações das camadas física, enlace, rede e transporte e quando a informação chegar na camada de agregação, ele irá verificar se a assinatura digital é válida. Se for, o nó intermediário irá armazenar a informação na sua área de armazenamento. Se não for válida, ele irá descartar a mensagem. Este processo ocorre em todos os nós intermediários que a informação for repassada até a mesma chegar ao nó destino. No nó destino, após a retirada dos cabeçalhos das camadas abaixo da camada de agregação e a camada de agregação verificar a assinatura digital da mensagem; a camada de agregação irá decriptografar a mensagem e repassar o conteúdo útil decriptografado para a camada de aplicação do nó destino.

Este capítulo apresentou a teoria desenvolvida para prover segurança para as DTNs. Porém, como este mecanismo de segurança funciona na prática? Qual o impacto dele no desempenho da rede? O próximo capítulo apresenta as ferramentas e protocolos utilizados para a avaliação do impacto do mecanismo de segurança proposto no funcionamento das DTNs. No Capítulo 5 serão apresentados os resultados de simulações feitas com este mecanismo de segurança para verificar o impacto do mesmo na rede.

Capítulo 4

Mobilidade dos Nós e Protocolos de Roteamento em DTN

AS DTNs, que são o foco deste trabalho, possuem características de redes ad hoc, ou seja, os nós não necessitam de se conectarem numa entidade central para enviar e receber seus pacotes. Os pacotes são transmitidos entre os nós através do mecanismo de roteamento. Além disso, nas DTNs os nós são móveis, o que significa que cada nó pode estar se deslocando em uma direção, em um determinado intervalo de tempo. Então, a mobilidade dos nós e os protocolos de roteamento são componentes essenciais na avaliação de desempenho das DTN, com isso uma breve descrição desses componentes se torna necessária e serão apresentadas neste capítulo.

4.1 Mobilidade do Nós

A mobilidade dos nós se refere a maneira em que os nós se deslocam dentro de uma área definida. A mobilidade dos nós pode ser representada através de mobilidade sintética, onde modelos matemáticos determinam o movimento dos nós; ou através de dados reais, onde rastros (*traces*) de mobilidade real são utilizados para representar a mobilidade dos nós. Estes rastros de mobilidade real podem ser dados pela movimentação de pessoas em um determinado local, como um *shopping center*, deslocamento de carros em determinadas ruas de um centro urbano ou, ainda, movimentação de animais na natureza, por exemplo. Já na mobilidade sintética, o modelo matemático que define a direção, velocidade e aceleração dos nós, é o que difere um modelo do outro. Exemplos de modelos sintéticos são: *Random Walk* [28], *Random Waypoint* [28], MMIG [2], *Random Direction* [28], Gauss-Markov [28], entre outros.

Neste trabalho os modelos de mobilidade sintéticos *Random Waypoint* e MMIG foram utilizados na avaliação das técnicas propostas no Capítulo 3. Ainda, rastros de mobilidade humana, coletadas em uma área de lazer, serão utilizados com intuito de demonstrar o funcionamento da técnica sugerida em um ambiente de mobilidade real. Os modelos de mobilidade, a coleta e tratamento dos rastros reais são apresentados a seguir.

4.1.1 *Random Waypoint*

O modelo *Random Waypoint* (RWP), inicialmente foi proposto por [29] e atualmente existem diversas variações do mesmo. Na versão mais empregada do RWP, os nós são dispostos aleatoriamente na área de simulação e usualmente, esta distribuição dos nós, segue uma distribuição de probabilidade uniforme. Após esta distribuição de posição, os nós permanecem nesta posição por um período aleatório, chamando tempo de pausa. Após esta pausa, o nó escolhe aleatoriamente um novo destino dentro da área de simulação e uma velocidade, que é uniformemente distribuída entre $[v_{min}, v_{max}]$. Então, o nó se desloca para esse destino com a velocidade

escolhida e após chegar ao destino, espera outro tempo de pausa aleatório, para reiniciar o movimento em outra direção.

O deslocamento de um nó, sob o modelo de mobilidade RWP pode ser vista na Figura 4.1. Esta figura foi retirada do trabalho apresentado em [28].

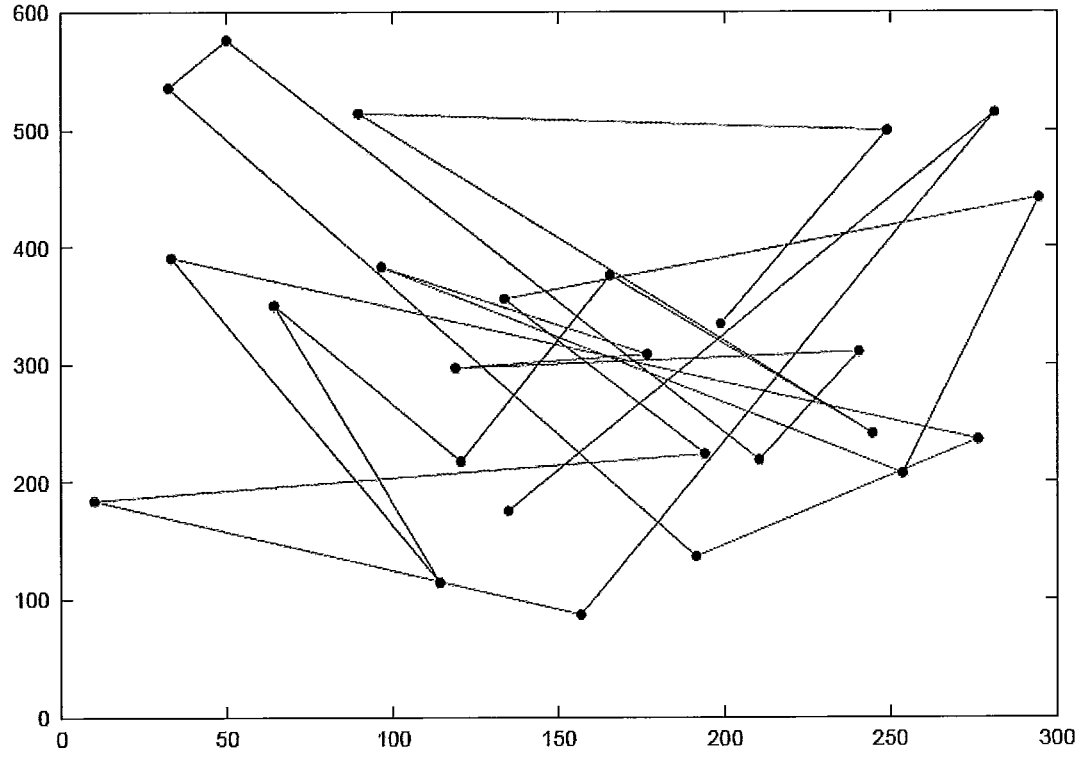

Figura 4.1: Exemplo da movimentação de um nó sob a influência do modelo de mobilidade RWP.

No RWP a escolha do próximo destino e da próxima velocidade não é influenciada pela posição e velocidade atuais. Com isso, o RWP é dito um modelo de mobilidade sem memória, ou seja, não leva em consideração as informações sobre o movimento em estágios de tempo anterior para gerar o movimento atual.

4.1.2 *MMIG*

O modelo de mobilidade MMIG, proposto em [2] usa uma cadeia de Markov de parâmetro discreto para simular a movimentação de um nó. A idéia é usar a

memória contida nos estados da cadeia de Markov para dar um senso de direção ao deslocamento de um nó e evitar mudanças bruscas na sua velocidade.

São utilizados duas cadeias de Markov: uma para deslocamentos na coordenada x e outra para deslocamentos na coordenada y. Em cada unidade de tempo, em função do estado da cadeia, é escolhido um deslocamento na direção x e outro na direção y. A cadeia de Markov possui probabilidade m de mudança para os estados à direita e probabilidade m de mudança para os estados à esquerda, com isso, a probabilidade de permanência no mesmo é (1 - $2m$). A Figura 4.2 ilustra a cadeia de Markov que representa a alteração de direção.

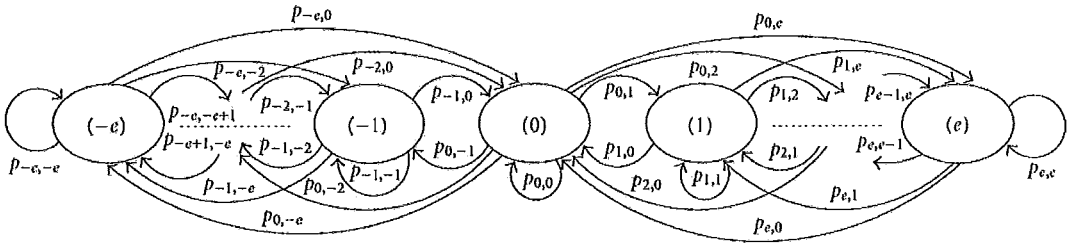

Figura 4.2: Cadeia de Markov de transição para o MMIG. Figura retirada de [2]

Nesse modelo é atribuído um conjunto de valores de incrementos na posição inicial do nó, que variará no intervalo [0, n]. Esse incremento representa a variação do valor da velocidade, em uma coordenada (x e y), e segue o comportamento de uma série geométrica onde o valor inicial é 1 e o valor máximo é n. Desta forma, o modelo permite um movimento suave com diversas velocidades resultantes.

Ajustando os valores de m, n e p é possível definir uma movimentação suave com pequenas variações de velocidade, representando por exemplo, o deslocamento de pessoas, bem como grandes acelerações, representando o deslocamento de veículos.

Na Figura 4.3, pode-se observar o rastro da movimentação de um nó utilizando o modelo de mobilidade MMIG.

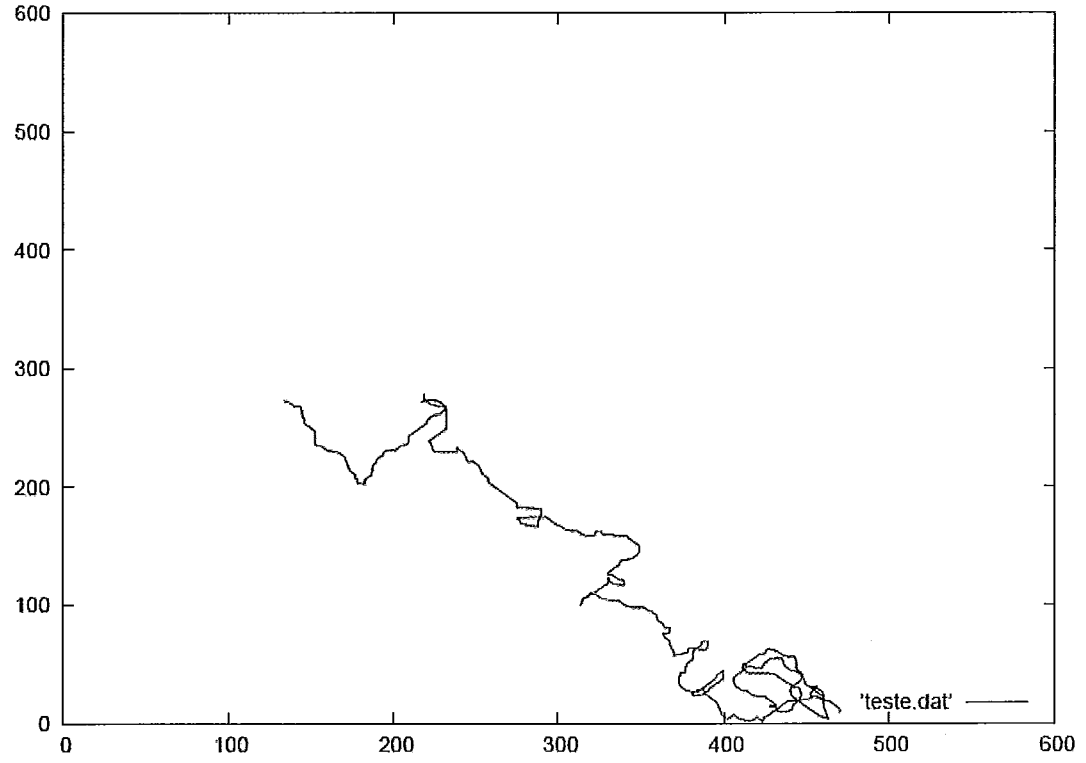

Figura 4.3: Exemplo da movimentação de um nó sob a influência do modelo de mobilidade MMIG. Figura retirada de [3]

4.1.3 Mobilidade Real

Nos últimos anos, várias características indesejáveis foram descobertas nos modelos aleatórios, como pode ser visto em [30, 31, 32]. Por isso, cuidados devem ser tomados quando usa-se, nas simulações, esses modelos. Assim, neste trabalho utilizou-se também um rastro de mobilidade real para validação do mesmo.

Esta mobilidade real foi coletada em uma área de lazer, chamada parque da Quinta da Boa Vista no Rio de Janeiro, Brasil. Este parque contém lagos, cavernas, muitas árvores e abriga dois importantes pontos turísticos: o jardim zoológico do Rio de Janeiro e o Museu Nacional. A coleta da mobilidade foi feita com o uso de um equipamento GPS da marca Trimble, modelo Geo XM 2, que possui uma alta precisão (com erro de localização submétrico) e permite correção diferencial, que é uma técnica de ajuste da posição geográfica coletada em relação a uma base de dados

geodésica. A coleta dos traces ocorreu durante vários dias da semana, das 9:00 às 16:00 horas. Esse horário foi escolhido com base em um software de planejamento da qualidade da constelação de satélites que cobrem uma determinada área geográfica em um determinado intervalo de tempo [1].

Para a coleta dos registros (*traces*) de movimento, 120 pessoas que caminhavam pelo parque, em dias e horários diferentes, foram escolhidas aleatoriamente e convidadas para participarem da pesquisa como voluntárias. Assim, o receptor de GPS foi entregue a cada um dos voluntários que se movimentaram, de maneira independente pelo parque. No final da caminhada de cada voluntário, o GPS era recolhido novamente. O tempo de coleta do movimento de cada voluntário variou de 300 à 1300 segundos aproximadamente, na qual o tempo entre amostras de cada posição coletada foi de um segundo.

Após a coleta dos experimentos, os dados foram submetidos à técnica de correção diferencial com o objetivo de melhorar a acurácia da medidas coletadas pelo GPS, ou seja, diminuir possíveis erros na captura da posição geográfica das pessoas através do GPS. Para isso, foi usada a ferramenta *Pathfinder Office*[2]. Além disso, valores discrepantes nos dados, não corrigidos pela técnica de correção diferencial, podem ocorrer devido a erros causados por obstáculos como as copas das árvores. Desta maneira, as posições geográficas que geraram deslocamentos (diferença entre a posição atual e a posição no segundo anterior) superiores a 2,5 metros foram descartadas[3]. Com o objetivo de ter-se uma avaliação com traces de durações iguais de tempo, foram escolhidos 100 experimentos que tiveram pelo menos 600 segundos de duração (tempo que foi usado nas simulações). O tempo remanescente (acima de 600 segundos) desses traces não foi considerado neste trabalho, juntamente com os 20 traces de duração menor que 600 segundos. Maiores detalhes desta coleta e tratamento dos dados coletados podem ser obtidos em [33].

[1]Para informações detalhadas sobre este software acesse http://www.trimble.com/planningsoftware.shtml

[2]Mais detalhes sobre esta ferramenta acesse http://www.trimble.com/pathfindertools.shtml

[3]Durante os experimentos, foi observado que nenhum voluntário chegou a velocidade de 2,5 m/s. Assim, deslocamentos superiores a 2,5 m, que podem ter sido gerados erroneamente, não foram considerados.

Na Figura 4.4, é mostrado o traçado do deslocamento de alguns dos experimentos usados na avaliação. Através desta figura, pode ser observado que o deslocamento dos voluntários pelo parque, na maioria das vezes, foi influenciado pelo formato das ruas e trilhas existentes no parque. Assim, pode-se dizer que esse deslocamento foi baseado em obstáculos ao contrário do deslocamento aleatório que é gerado pelos modelos de mobilidade, mais utilizados na literatura, *random walk* e *random waypoint.*

Figura 4.4: Traçado do deslocamento de alguns dos experimentos coletados no parque da Quinta da Boa Vista.

Com o uso das três mobilidades descritas acima, será avaliada a eficiência do mecanismo proposto neste trabalho no uso de mobilidade sintética, com e sem memória, e com o uso de mobilidade de pessoas, abrangendo a maioria dos casos de mobilidade possíveis.

4.2 Protocolos de Roteamento em DTN

O roteamento nas DTNs é caracterizado pela forma de trocar as mensagens a cada contato. Assim, os protocolos de roteamento são classificados em encaminhadores e replicadores. Os protocolos encaminhadores como, o PROPHET [34], o MEED [14] e o MAXPROP [35], escolhem quais mensagens serão transmitidas, e os replicadores, que têm como exemplo, o Epidêmico [36], Rapid [37], o *Spray and Wait* [38] e o *Spray and Focus* [39], repassam cópias de suas mensagens em cada contato. Os protocolos de roteamento Epidêmico, PROPHET e *Spray and Wait*, que são os mais usados na literatura, serão utilizados neste trabalho. Uma explicação do funcionamento dos mesmos é apresentada a seguir.

No trabalho [36], os autores propõem o protocolo Epidêmico, onde o roteamento é realizado pela própria mobilidade dos nós na DTN. Assim, quando um nó entra no alcance de transmissão de outro nó é estabelecida uma conexão. Em seguida, os nós trocam suas listas de mensagens armazenadas. Deste modo, as mensagens da lista recebida é comparada com as mensagens presentes no nó, para determinar quais mensagens o nó não possui. Feito isso, o nó solicita o envio de cópias destas mensagens. O processo de troca de mensagens se repete toda vez que um nó estabelece contato com um novo nó, o que permite que as mensagens sejam rapidamente distribuídas pela rede. Assim, quanto mais cópias de uma mesma mensagem forem encaminhadas na rede, maior será a probabilidade desta mensagem ser entregue e menor será o atraso. Este foi o primeiro protocolo de roteamento proposto para as DTNs.

O protocolo PROPHET (*Probabilistic ROuting Protocol using History of Encounters and Transitivity*) proposto por Lindgren e outros, em [34], utiliza o mesmo princípio de troca de mensagens utilizado no protocolo Epidêmico. Quando dois nós estabelecem uma conexão, eles trocam as suas listas de mensagens. A diferença é que nesta lista existe um parâmetro novo para cada mensagem da lista. Esse parâmetro corresponde à probabilidade de cada nó a entregar mensagens para um destino conhecido b ($P_{(a,b)} \in [0,1]$). O valor de $P_{(a,b)}$ aumenta sempre que a e b se encontram e diminui se a e b deixam de se encontrar freqüentemente. O tempo

é controlado por uma constante k, denominada constante de envelhecimento, que corresponde ao número de unidades de tempo transcorridas desde a última vez que a métrica foi atualizada. Quando um nó recebe a lista do vizinho, ele calcula a probabilidade de entrega para cada mensagem que ainda não possui. Em seguida, para cada mensagem, o nó compara a probabilidade indicada na sua lista com a probabilidade indicada na lista recebida do vizinho. Essa comparação é realizada para verificar qual dos dois nós possui a maior probabilidade de entrega. Após essa comparação, três procedimentos serão realizados: (i) o nó deve enviar um pedido das mensagens não armazenadas que possuem uma maior probabilidade de serem entregues através dele; (ii) recebe o pedido de mensagens do vizinho e as envia; e (iii) apaga todas as mensagens que o vizinho tem maior probabilidade de entregar. No final, cada nó possuirá somente mensagens cujas probabilidades de entrega sejam maiores através dele.

Já o protocolo *Spray and Wait* (SW) , apresentado por Spyropoulos e outros em [38], combina a velocidade do protocolo Epidêmico, com a simplicidade de um envio direto (*direct transmission*) para o nó destino. Este protocolo possui duas fases, na primeira, chamada de *Spray*, para cada mensagem gerada no nó origem, L cópias desta mensagem são repassadas para outros $L-1$ nós. Se o nó destino não foi alcançado nesta fase, o protocolo entra na fase de espera, chamada *Wait*, onde os L nós que contém cópias da mensagem, irão repassá-las somente para o nó destino. Uma otimização deste protocolo, bastante utilizada, é chamada de *binary Spray and Wait*. Nessa otimização, cada nó se possuir $n > 1$ cópias da mensagem (onde n é o número de cópias de uma mensagem contida em um nó), irá repassar $\lfloor n/2 \rfloor$ e manterá $\lceil n/2 \rceil$ cópias da mensagem consigo, até o nó possuir apenas uma cópia da mensagem ($n = 1$), quando entrará na fase *Wait*.

Com o uso destes três protocolos de roteamento, o mecanismo proposto será avaliado tanto para protocolos encaminhadores, quanto para replicadores, com e sem controle da quantidade de cópias das mensagens. Com isso, pode-se apresentar os resultados obtidos nos mais diferentes contextos de roteamento das DTNs.

Capítulo 5

Impacto do Mecanismo de Segurança Proposto em uma DTN

COM o intuito de avaliar o impacto do uso do mecanismo de segurança proposto neste trabalho, este capítulo apresenta os resultados de simulações de uma DTN, efetuadas com e sem o uso desse mecanismo, para efeitos de comparação. Inicialmente o simulador utilizado para obter os resultados descritos neste capítulo será apresentado. Após isso, uma descrição dos parâmetros das simulações e das métricas de desempenho avaliadas serão apresentadas. Cenários de alta densidade e baixa densidade serão utilizados, para que o funcionamento do mecanismo proposto, em diversas situações, seja investigado. Por fim, os resultados obtidos nessa avaliação serão apresentados e discutidos.

5.1 O Simulador

A avaliação do mecanismo proposto neste trabalho foi realizada através do uso do simulador *The Opportunistic Network Environment simulator* (*The ONE*) [40], desenvolvido pelo *Networking Laboratory* da *Helsinki University of Technology*, para estudo de mobilidade em DTNs.

O simulador *The ONE*, desenvolvido com o uso da linguagem de programação JAVA, possui código aberto e está disponível para uso na página web do mesmo. Alguns trabalhos publicados fizeram o uso deste simulador, como em [41, 42, 43]. Uma "fotografia" da execução do simulador *The ONE* pode ser vista na Figura 5.1.

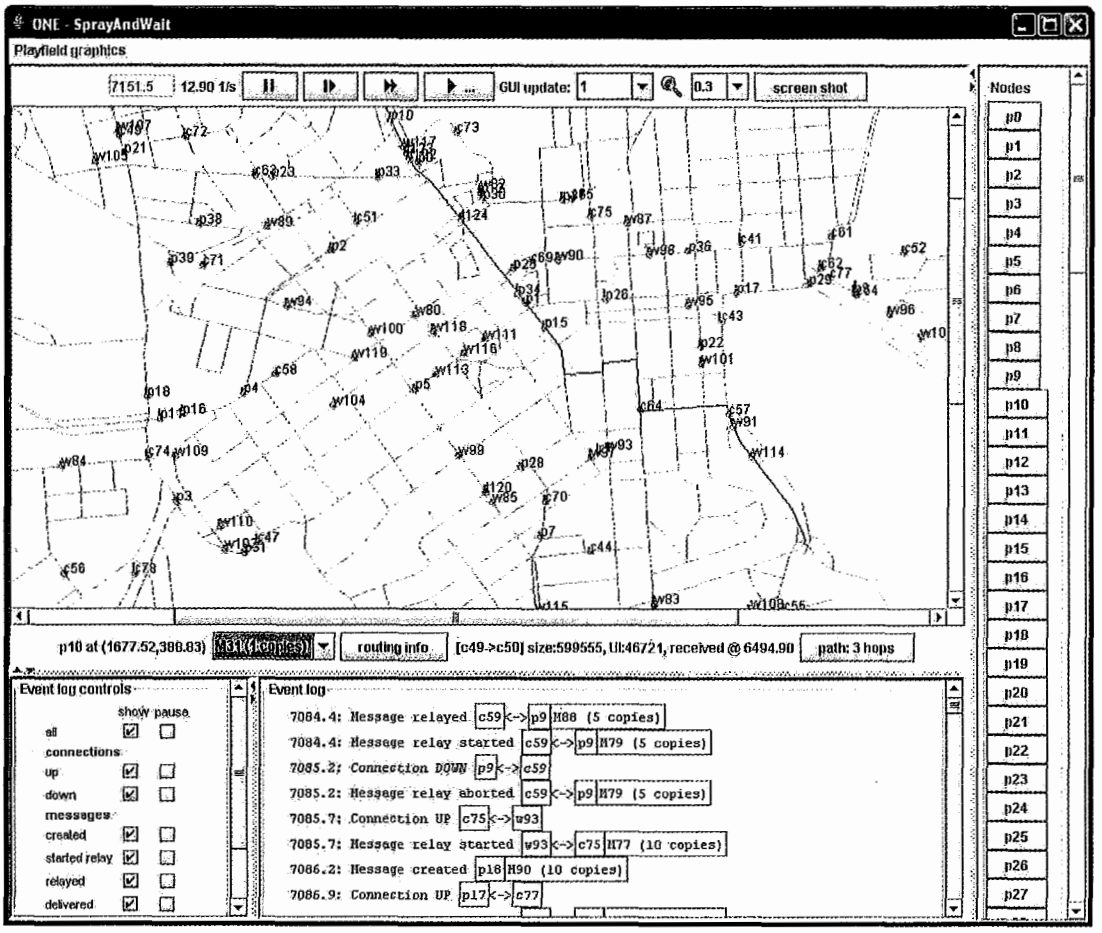

Figura 5.1: Exemplo de execução do simulador The One

A versão do simulador utilizada foi a 1.2 e suas características estão descritas no apêndice B. O simulador foi desenvolvido para ser utilizado na análise da mobi-

lidade dos nós nas DTNs assim alguns campos das mensagens de dados não estão implementados. Desta maneira houve a necessidade de se efetuar alterações no simulador nas partes referentes as mensagens do simulador e também foi implementado um módulo de segurança de acordo com o mecanismo proposto e apresentado no Capítulo 3.

5.2 Detalhes da Implementação do Mecanismo Proposto

O módulo de segurança desenvolvido, também na linguagem de programação JAVA, contém o algoritmo de criptografia RSA e o código hash SHA-256. Além disso, foi criado o código que gera a assinatura digital, utilizando os códigos do algoritmo de criptografia e o código hash. Com isso, este módulo contém o código para garantir as três premissas básicas de segurança.

O módulo que gerencia as mensagens na rede foi alterado para as mensagens possuírem a estrutura descrita na Seção 3.4. Para a criação da mensagem criptografada e da assinatura digital foi necessário alterar o módulo do simulador de criação das mensagens, para que no momento da geração de uma mensagem a mesma pudesse ser criptografada e sua assinatura digital fosse gerada ou, se o nó não possuísse a chave, que a mensagem fosse armazenada sem a criptografia, porém com o campo criptografada vazio (verificar estrutura da mensagem na Seção 3.4).

O módulo de roteamento das mensagens também foi alterado, pois o mesmo é que verifica se a mensagem chegou corretamente e se o nó que recebeu a mensagem é o nó destino. Neste módulo foi necessário inserir a verificação da assinatura digital sempre que uma nova mensagem é recebida por um nó e decriptografar a mensagem se o nó que recebeu a mensagem é o nó final. Ainda neste módulo é feita a verificação se a mensagem está criptografada antes da origem repassá-la.

No contexto da troca de chaves, o módulo que gerencia os nós da rede foi alterado para que os nós armazenem, atualizem e busquem as chaves na memória. O módulo

responsável pelo controle da conexão entre os nós e a troca de mensagens também foi alterado para permitir que ocorra a troca de chaves antes da troca de mensagens de dados. O módulo de roteamento das mensagens foi alterado para não permitir que cópias das mensagens do tipo "Lista de Nós" ou "Lista de Chaves" sejam repassadas para outros nós que não sejam o destino.

Por fim o módulo que controla os relatórios da execução do simulador foi alterado para apresentar relatórios referentes a segurança e para não contabilizar as mensagens de segurança em alguns resultados.

5.3 Métricas de Avaliação

Para avaliar o desempenho das DTNs utilizando o mecanismo de segurança proposto, algumas métricas de desempenho serão definidas nesta seção e utilizadas nas seções seguintes para apresentar e discutir a eficiência do mecanismo proposto.

Uma das métricas mais importantes na avaliação de redes ad hoc e especialmente em DTNs é a **probabilidade de entrega** das mensagens ao destino, definida como sendo a razão entre as mensagens que foram roteadas até o nó destino e o total de mensagens criadas na rede. Essa métrica é importante para avaliar se as trocas de mensagens de segurança, a demora no envio de uma mensagem devido a falta da chave pública do receptor e a perda de mensagens no *buffer* causam uma queda considerável na probabilidade de entrega de mensagens na rede. A *priori*, esta métrica tende a apresentar resultados piores quando aplicada a segurança em comparação aos resultados sem o uso da segurança.

O **atraso médio** é a soma dos atrasos de todas as mensagens trafegadas na rede sobre a quantidade dessas mensagens. Onde o atraso é definido como a diferença entre o instante de tempo que a mensagem foi entregue ao destino e o instante de tempo de criação da mesma. Esta métrica de grande importância pois se o roteamento da origem ao destino for muito longo, a mensagem pode se tornar obsoleta.

Já a **sobrecarga** (*overhead*) das mensagens na rede é dado pela quantidade de

repasses de mensagens (encaminhadas entre os nós intermediários) que não são para os nós finais dividido pelo número de mensagens entregues (msgRep/msgEntreg). As mensagens consideradas para o cálculo desta métrica são apenas as mensagens de dados, ou seja, as mensagens que não são do tipo "Lista de Nós" ou "Lista de Chaves". Esta métrica apresenta a quantidade adicional de mensagens geradas na rede necessárias para a entrega das mensagens ao seus destinos. O repasse de muitas cópias das mensagens numa DTN pode não ser uma característica interessante, pois o consumo de energia para se enviar uma mensagem é muito maior que o de processá-la e, além disso, os *buffers* dos nós são limitados. Esta métrica também mostra a influência do mecanismo de troca de chaves na rede, pois quanto maior o tempo necessário para troca de chaves, menor será o repasse das mensagens e, conseqüentemente, menor será a sobrecarga.

A métrica **sobrecarga de segurança** indica o custo em bytes para a implementação da segurança na DTN. Esta métrica é dada pelo número de bytes de segurança trafegados dividido pelo total de bytes trafegados na rede. Os bytes de segurança são compostos pelo tamanho das mensagens do tipo "Lista de Nós", tipo "Lista de Chaves" e do tamanho do campo Hash das mensagens de dados, repassadas na rede. O total de bytes trafegados na rede compreende os bytes de segurança e os bytes das mensagens de dados. A importância desta métrica se dá na verificação da quantidade de recursos alocados da rede, que poderiam ser usados para trafegar mensagens de dados e que são usados para troca de informações de segurança.

Por fim, a métrica representada por α é definida como a **porcentagem de mensagens não repassadas devido a falta de chave no nó origem**. Esta métrica ilustra a quantidade de mensagens que poderiam ser repassadas para outro nó mas, o nó origem não possuía a chave do destino, ou seja, a chave do destino não havia sido roteada para o nó origem no momento da oportunidade de transmissão. A importância desta métrica se dá na verificação do tempo necessário para que a convergência da troca de chaves na rede possibilite poucos atrasos no envio das mensagens, devido a falta de chave na origem, ou seja, o tempo necessário para que a grande parte das chaves já estejam repassadas aos nós da rede, com isso diminuindo a possibilidade de falta de chave no momento do envio de um mensagem.

5.4 Cenário de Alta Densidade

Diversos cenários podem ser vislumbrados em relação ao uso de uma DTN. Uma grande parte desses cenários pode possuir alta densidade de equipamentos em uma determinada área. Exemplos para cenários de alta densidade são shopping center, centros urbanos, áreas de lazer (como a área da Quinta da Boa Vista descrita na seção 4.1.3), entre outras.

Estes cenários se caracterizam pela possibilidade de existir muitas oportunidades de troca de mensagens, ou seja, muitos enlaces (*links*) oportunísticos, devido a grande quantidade de encontros nesses cenários. Um encontro é definido como o momento em que o nó A entrou na área de cobertura de B e vice-versa (neste trabalho, em cada rodada de simulação, todos os nós possuem o mesmo tamanho para o raio de transmissão).

Para uma maior avaliação deste trabalho, o raio de transmissão dos equipamentos foi variado, fazendo com que o número de encontros fosse diferente em cada situação. Com isso, este trabalho será avaliado para diferentes tipos de equipamentos, com o objetivo de representar desde os equipamentos de baixa capacidade de transmissão, por exemplo nós sensores, até equipamentos com alta capacidade de transmissão, por exemplo *notebooks*.

5.4.1 Parâmetros da Simulação

Para simular um cenário de alta densidade, 100 nós foram inseridos em uma área de 800x600 metros. Este tamanho de área foi configurado de acordo com o tamanho da área onde os dados de movimento foram coletados na Quinta da Boa Vista. O raio de transmissão dos nós foi variado de 10 em 10 metros, a partir de 10 metros até 100 metros de raio. Com isso, pode-se avaliar situações de baixa e alta conectividade neste cenário. Cabe ressaltar que com o aumento do raio de transmissão dos nós surgem questões como: maior interferência entre os nós, problema do terminal escondido e maior gasto de energia e estes problemas não são considerados nas simulações.

A mobilidade dos nós foi dada por rastros de mobilidade em um cenário real e pelos modelos de mobilidade Random Waypoint e MMIG. O cenário real de mobilidade foi descrito na seção 4.1.3 e consiste de rastros de 100 pessoas, escolhidas aleatoriamente, em um ambiente de lazer de área 800x600 metros durante 600 segundos. No modelo de mobilidade Random Waypoint, a velocidade dos nós foi escolhida aleatoriamente entre 1.52 e 1.58 m/s e não houve tempo de pausa. Já o MMIG, a velocidade máxima de um nó foi de 1.5 m/s, a probabilidade do nó mudar de estado na cadeia de Markov, para a esquerda ou direita, (parâmetro m) é 0.4 e a base do incremento do valor da velocidade do nó (parâmetro b) foi de 1.028. Os parâmetros das mobilidades sintéticas foram ajustados pela técnica de Erro Quadrático Médio (*Mean Square Error* - MSE) , que consiste em ajustar os parâmetros da mobilidade sintética conforme rastros de uma mobilidade real. Esta técnica está descrita em [44].

Os protocolos de roteamento utilizados foram o Epidêmico, o PROPHET e o *Spray and Wait*, com objetivo de avaliar os protocolos de roteamento com diferentes características. No protocolo PROPHET, a probabilidade a priori inicial de repassar uma mensagem (parâmetro b) é 0.75, a taxa de aumento (ou decréscimo) de b em função de encontros (ou desencontros) é 0.25 e a constante de envelhecimento é 0.98. Estes parâmetros do roteamento PROPHET foram escolhidos com intuito de haver maior quantidade de troca de mensagens entre os nós da rede. No *Spray and Wait*, o número cópias máximo de cada mensagem foi definido como 6. Este valor foi estabelecido conforme valor apresentado em tabela no artigo que define o protocolo Spray and Wait [38].

As mensagens foram geradas conforme uma distribuição de probabilidade uniforme, com taxa de 1 mensagem por segundo no sistema. O tamanho das mensagens foi escolhido aleatoriamente entre 1KB e 20KB, para representar desde pequenas até grandes mensagens. O *buffer* dos nós foi definido em 5MB e a taxa de transmissão das mensagens foi de 1Mbps. Estes valores de tamanho do *buffer* e de taxa de transmissão foram escolhidos com intuito de representar dispositivos com baixa capacidade de armazenamento e transmissão. O tempo de simulação foi de 600 segundos, com uma fase transiente de 1000 segundos para a mobilidade sintética,

por rodada. Foram realizadas 10 rodadas de simulação para cada combinação de parâmetros e os resultados foram obtidos com um nível de confiança de 95%.

O algoritmo de criptografia utilizado nos resultados com segurança foi o RSA e a assinatura digital foi criada com o uso do algoritmo de hash SHA-256 e criptografado com RSA, explicados no Apêndice A.

A Tabela 5.1 apresenta um resumo dos parâmetros da simulação.

Tabela 5.1: Parâmetros utilizados na Simulação do Cenário de Alta Densidade

Parâmetro	Valor
Número de nós	100
Área de Simulação	800x600m
Raio de Transmissão	10,20,30,40,50,60,70,80,90,100 m
Modelo de Movimentação dos nós	*Real, RWP e MMIG*
Tempo de Pausa do RWP	0 seg
Velocidade de Deslocamento do RWP	Entre 1.52 e 1.58 m/s
Velocidade Máxima de Deslocamento do MMIG	1.5 m/s
Probabilidade de Mudança do MMIG	0.4
Incremento da velocidade do MMIG	1.028
Algoritmos de Roteamento	*Epidemico, Spray and Wait e Prophet*
Probabilidade a Priori inicial de repassa a mensagem no PROPHET	0.75
Taxa de aumento (ou decréscimo) de b no PROPHET	0.25
Constante de envelhecimento no PROPHET	0.98
Número máximo de cópias da mensagem no Spray and Wait	6
Geração Mensagens	1 mensagem por segundo
Tamanho das mensagens	Entre 1KB e 20 KB
Tamanho do Buffer dos nós	5MB
Velocidade de Transmissão	1Mbps
Tempo de Simulação	600 seg
Tempo de simulação descartado da Mobilidade Sintética	1000 seg
Quantidade de Rodadas de Simulação	10
Nível de confiança dos resultados	95%
Assinatura Digital	SHA-256 + RSA
Algoritmo de Criptografia	RSA

Apesar do uso de diferentes modelos de mobilidades e diferentes protocolos de roteamento na avaliação realizada neste capítulo, não é objetivo do mesmo a comparação de resultados entre diferentes modelos de mobilidades ou protocolos de roteamento, e sim comparar a eficiência do mecanismo de segurança sob a influência

de cada tipo de mobilidade e protocolo de roteamento.

A seguir, os resultados do cenário descritos nessa seção serão apresentados para cada métrica descrita anteriormente.

5.4.2 Probabilidade de Entrega

Como definido na seção 5.3, a probabilidade de entrega é a razão entre as mensagens que foram roteadas até o nó destino e o total de mensagens criadas na rede. Para esta métrica, espera-se que a probabilidade de entrega seja menor quando se usa o mecanismo de segurança, pois haverá maior demora no repasse das mensagens, devido a falta de chaves, e haverá troca de dados de segurança, o que pode atrasar o envio de uma mensagem de dados. Além disso, é esperado que a diferença desta métrica sem o uso e com o uso do mecanismo de segurança diminua conforme se aumenta o raio (aumentando a conectividade), pois a troca de chaves será mais rápida e haverá mais oportunidades de troca.

A Figura 5.2 apresenta a probabilidade de entrega, variando os protocolos de roteamento para a mobilidade real. Pode-se observar que com o aumento do raio de transmissão, os valores da probabilidade de entrega sem e com o uso do mecanismo de segurança se aproximam, para cada protocolo de roteamento. Para exemplificar, considerando o raio de transmissão igual a 20 metros, o protocolo de roteamento que apresentou a maior diferença da probabilidade de entrega sem e com o uso do mecanismo de segurança foi o protocolo epidêmico. Neste caso a probabilidade de entrega para o protocolo epidêmico sem segurança foi 0.405 contra 0.277 com o uso da segurança, ou seja, uma redução de 31.6% na probabilidade de entrega.

Considerando agora o raio de transmissão igual a 50 metros, a probabilidade de entrega no protocolo epidêmico sem segurança foi 0.79 contra 0.775 com o uso da segurança, ou seja, uma redução de 1.9% na probabilidade de entrega. O protocolo que apresentou maior redução para o raio de 50 metros foi o Spray and Wait, que passou de uma probabilidade de 0.437 sem a segurança para 0.407 com a segurança, uma redução de 6.9%.

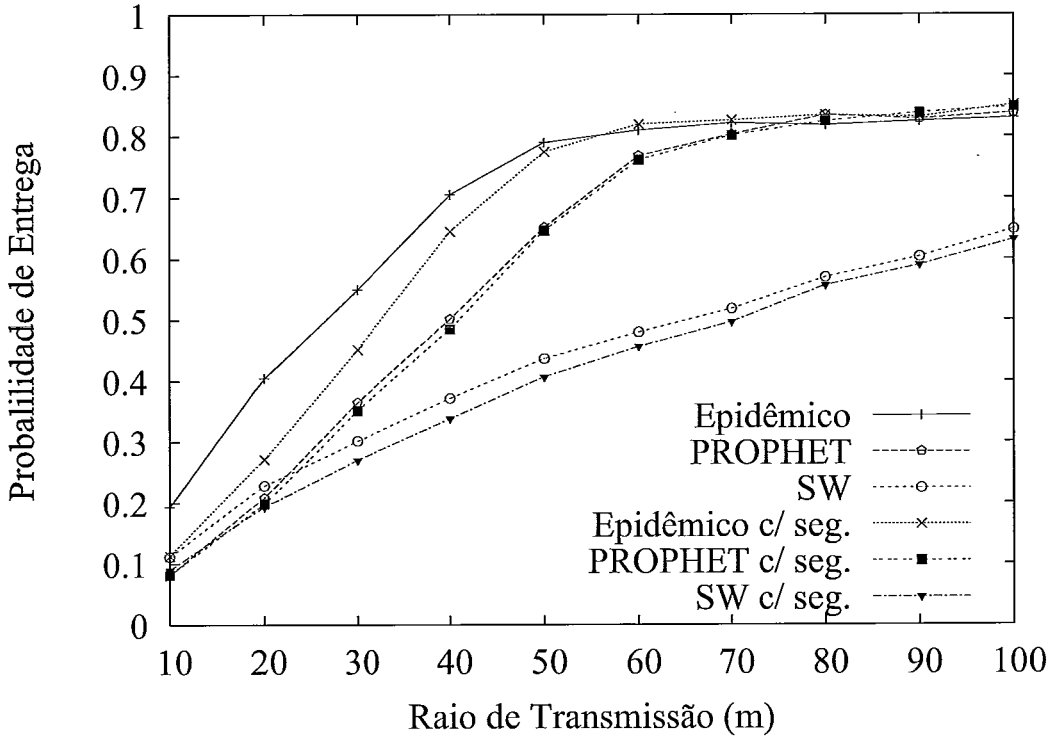

Figura 5.2: Probabilidade de entrega das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

A redução da diferença da probabilidade de entrega quando se compara o uso do mecanismo de segurança e não uso do mesmo aumentando o raio é ocasionado pela maior rapidez na troca das mensagens e na troca das chaves para os maiores raios.

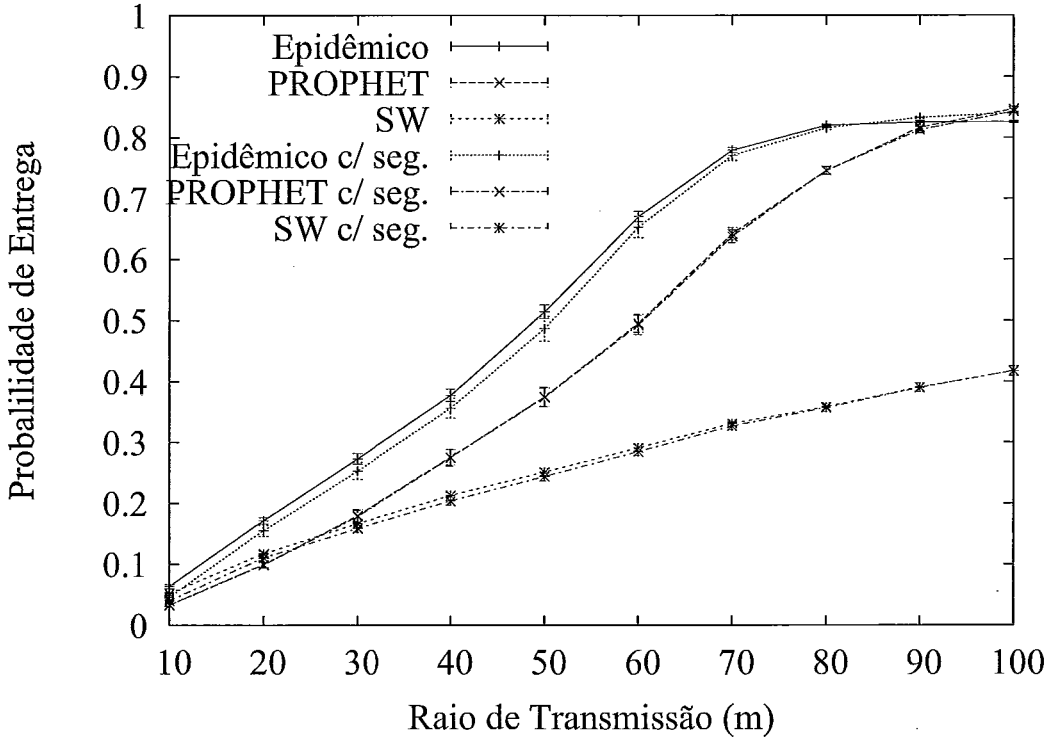

Figura 5.3: Probabilidade de entrega das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética usando o modelo MMIG.

O gráfico para a mobilidade sintética gerada pelo modelo MMIG pode ser observado na Figura 5.3, onde os resultados possuem o mesmo comportamento do resultado anterior, com a diferença entre as curvas diminuindo para um mesmo protocolo de roteamento, conforme se aumenta o raio de transmissão. Pode-se observar neste cenário que a diferença entre os resultados com o uso do mecanismo de segurança e sem o mecanismo, em raios pequenos foi menor se comparado com o resultado da mobilidade real. Isto reflete que o desempenho do mecanismo de segurança está associado diretamente a mobilidade dos nós.

Finalmente a Figura 5.4 apresenta os resultados para a mobilidade sintética gerada pelo modelo RWP, onde os resultados também apresentam um comportamento

bastante parecido com a mobilidade real. Novamente, para raios pequenos a diferença do não uso e do uso do mecanismo de segurança é considerável, porém não muito grande. Já a partir do raio de transmissão 60 metros esta diferença passa a ser muito pequena.

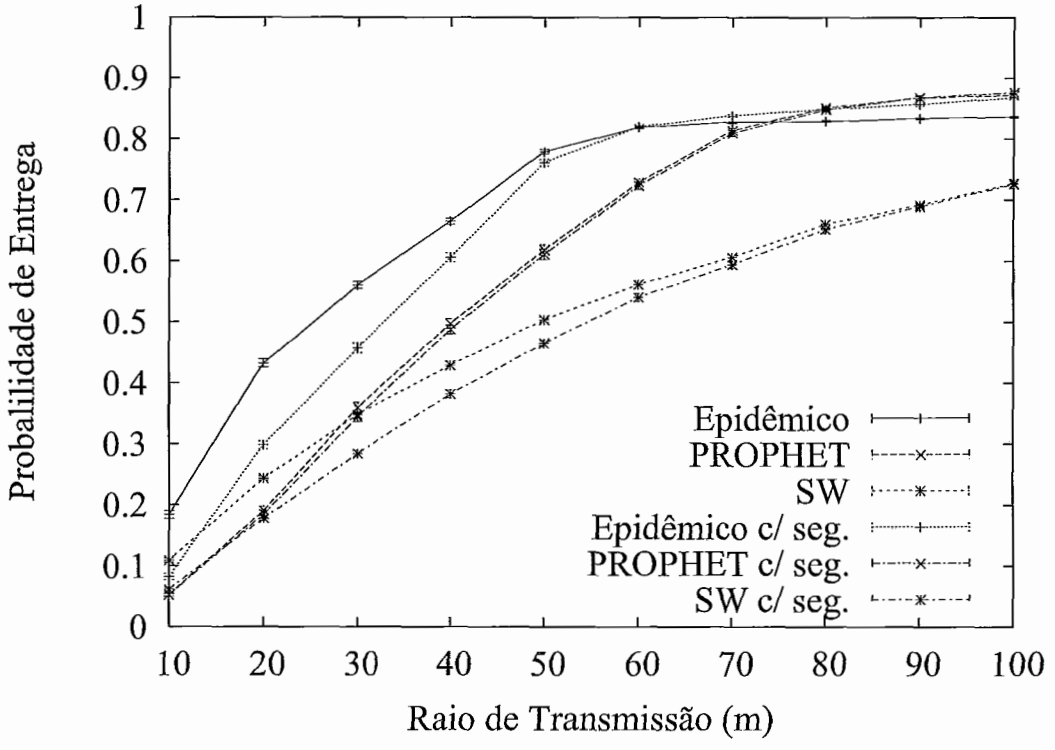

Figura 5.4: Probabilidade de entrega das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética usando o modelo RWP.

Analisando os resultados pode-se concluir que nos cenários avaliados para a métrica de probabilidade de entrega, o uso do mecanismo de segurança influenciou pouco na probabilidade de entrega das mensagens para cenários com média e alta conectividade (raios maiores que 40 metros). Por outro lado, nos cenários com baixa conectividade, houve uma diferença perceptível para o protocolo Epidêmico, porém não ultrapassando perda maior que 33%.

Porém algumas perguntas ainda estão em aberto, como: 1 - Os resultados ficarão próximos também para as outras métricas? 2- A quantidade de dados de segurança na rede apresenta grande impacto? As respostas para essas e outras perguntas serão

apresentadas nas próximas sub-seções.

5.4.3 Atraso Médio

A métrica do atraso médio apresenta o tempo decorrido desde a geração de uma mensagem até sua chegada ao destino. A princípio, o mecanismo de segurança deve ocasionar um maior atraso nas mensagens, novamente, devido a maior demora no repasse das mensagens, a falta de chaves e a troca de dados de segurança. Espera-se que a diferença entre o atraso sem segurança e com segurança diminua, conforme se aumenta o raio.

A Figura 5.5 mostra os valores de atraso médio para a mobilidade real, com diferentes protocolos de roteamento, variando o raio de transmissão. Observa-se que o atraso médio da entrega das mensagens é maior quando o raio é menor, ou seja, quando há menor conectividade. Este resultado é esperado, pois quanto menor a conectividade, menor a chance de se repassar uma mensagem, com isso a mensagem fica mais tempo no *buffer* de um nó até chegar ao destino.

Além disso, nota-se que a influência de uma maior conectividade da rede impacta o mecanismo de segurança para os protocolos Epidêmico e PROPHET, quando se analisa raios maiores que 50 metros, para a métrica do atraso médio. Este impacto é causado pela maior troca de material criptográfico no momento de uma conexão, o que aumenta o tempo de espera de uma mensagem no *buffer*.

A Figura 5.6 apresenta os valores de atraso médio para a mobilidade sintética gerada pelo modelo MMIG, para diferentes protocolos de roteamento, variando o raio de transmissão. O comportamento do gráfico é semelhante ao gráfico para o cenário da mobilidade real. O atraso médio diminui conforme se aumenta o raio, conforme esperado. A diferença do não uso do mecanismo de segurança em relação ao uso do mesmo é muito pequena, para raios maiores que 30 metros em todos os protocolos de roteamento.

A Figura 5.7 apresenta os valores de atraso médio para a mobilidade sintética gerada pelo modelo RWP, para diferentes protocolos de roteamento. Conforme

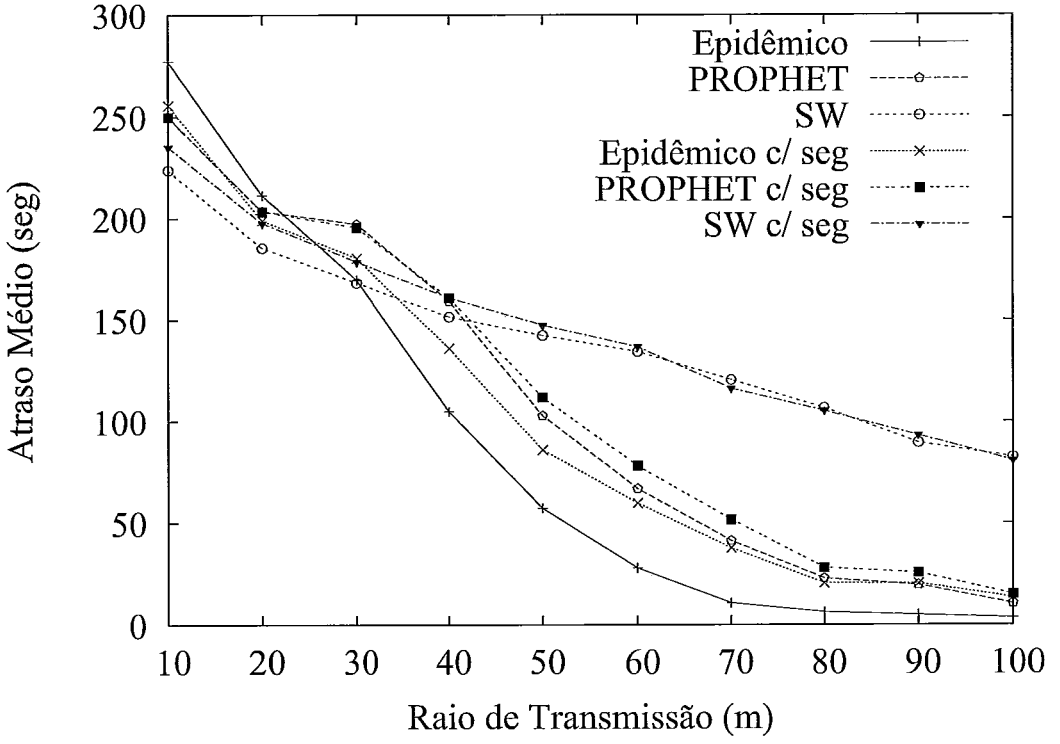

Figura 5.5: Atraso médio das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade Real.

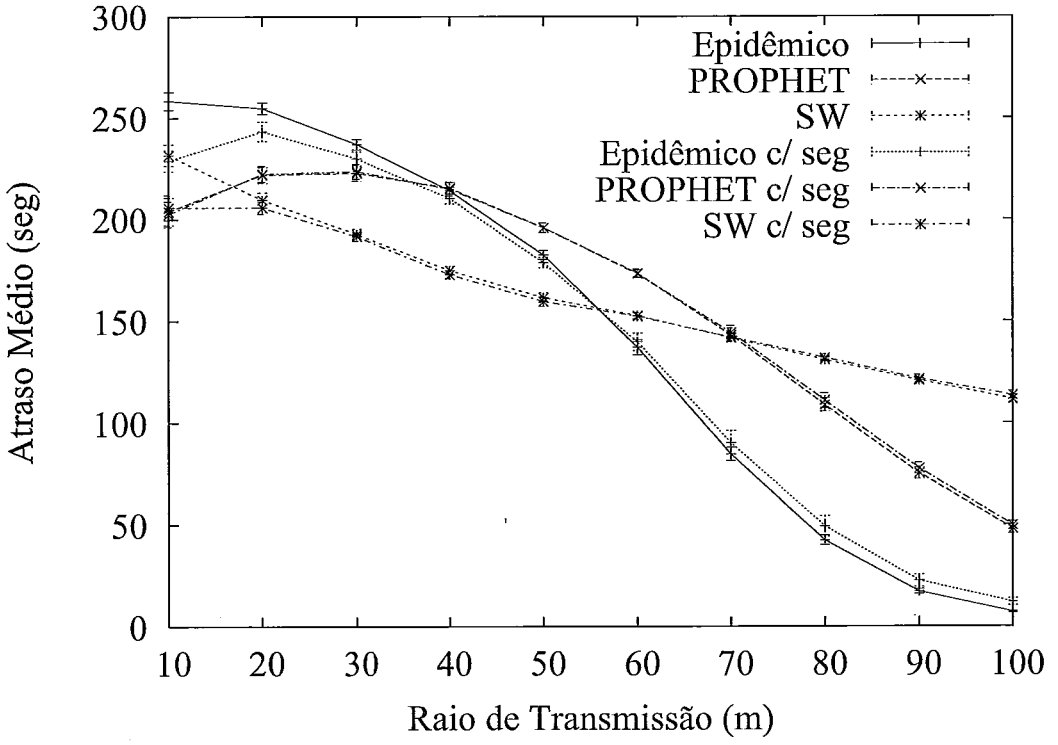

Figura 5.6: Atraso médio das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética usando o modelo MMIG.

esperado, o atraso diminui conforme se aumenta o raio e existe uma diferença, não muito alta, decorrente do uso do mecanismo de segurança em todos os protocolos.

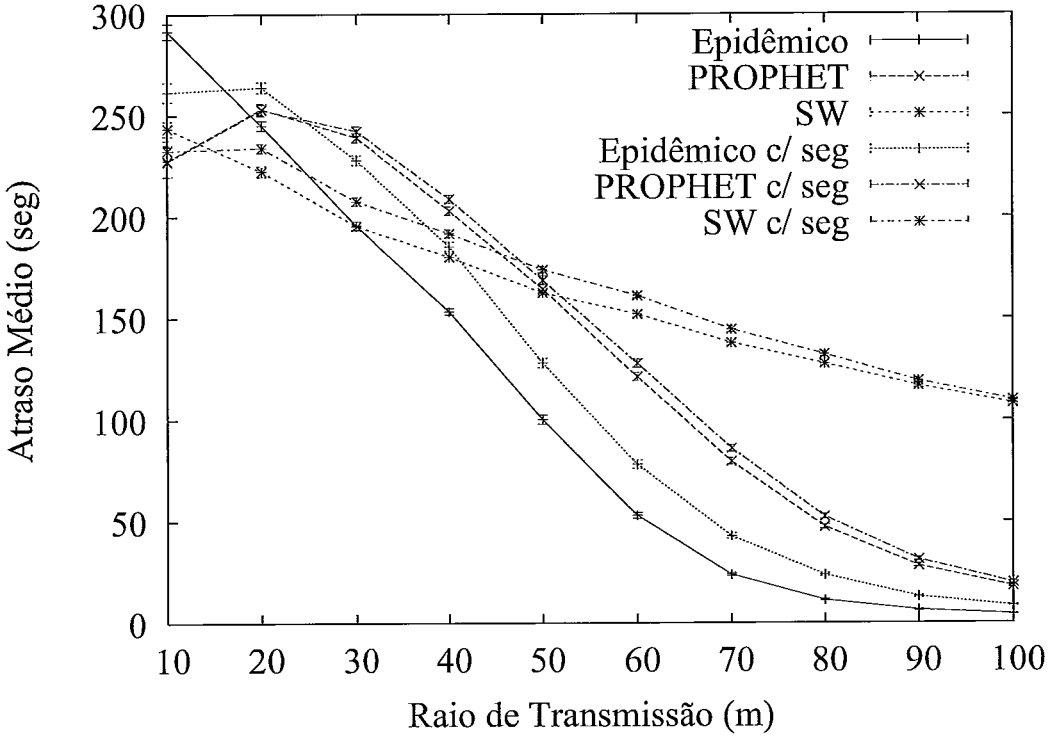

Figura 5.7: Atraso médio das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética usando o modelo RWP.

Nesta métrica, pode-se concluir que o atraso médio das mensagens entregues foi relativamente pequeno para os protocolos PROPHET e *Spray and Wait* para todos os raios e variou um pouco para o protocolo Epidêmico. O comportamento do atraso médio seguiu o que era esperado em todos os casos onde o atraso foi maior com o uso do mecanismo de segurança em relação ao não uso do mesmo.

5.4.4 Sobrecarga

A métrica sobrecarga indica a quantidade de mensagens de dados a mais enviadas na rede para que uma mensagem chegue ao destino. Esta métrica não contabiliza as mensagens de segurança e o objetivo é verificar se a segurança impede mensagens de

dados na rede, o que pode explicar o comportamento de alguns protocolos. Dada as características de cada protocolo de roteamento, espera-se que o Epidêmico apresente a maior sobrecarga, seguido do PROPHET e do *Spray and Wait*, pois o protocolo no Epidêmico repassa uma cópia das mensagens em cada contato sem algum controle de repasses, o PROPHET repassa as mensagens sem controle de repasses também, porém não faz cópias das mensagens. Já o protocolo *Spray and Wait*, apesar de repassar cópias das mensagens, limita essas cópias e portanto os repasses a um valor fixo. É esperado que sobrecarga com o uso da segurança seja menor para raios pequenos, comparado com a sobrecarga sem o uso da segurança, e que esta diferença diminua conforme se aumenta o raio de transmissão.

A Figura 5.8 ilustra os valores da sobrecarga para a mobilidade real, para diferentes protocolos de roteamento, variando o raio de transmissão. Observa-se pelo gráfico que a sobrecarga do protocolo Epidêmico é maior que de todos os outros protocolos de roteamento, devido a sua característica de inundação, onde cada nó repassa suas mensagens em cada encontro com outro nó. A sobrecarga do protocolo de roteamento PROPHET é menor que o do protocolo de roteamento Epidêmico devido ao repasse de suas mensagens que depende de uma métrica probabilística e ainda, só existe uma cópia de cada mensagem na rede, portanto o repasse de mensagens não é tão alto como o Epidêmico. Por fim, o protocolo *Spray and Wait* repassa suas mensagens no máximo 6 vezes, com isso a sobrecarga deste protocolo é baixo.

Pode-se observar ainda que a sobrecarga do *Spray and Wait* foi maior que 6 mensagens repassadas por mensagens entregues, isso se deve ao fato de que nem todas as mensagens repassadas chegaram ao destino. Com isso, a métrica sobrecarga pode ser maior que 6 mensagens repassadas por mensagens entregues. Resta comentar que para raios grandes, a métrica é próxima de 6 mensagens repassadas por mensagens entregues, devido ao maior número de mensagens entregues ao destino.

Em relação a segurança, o que era esperado para esta métrica se apresentou claramente no protocolo Epidêmico. Até o raio de transmissão de 60 metros, a sobrecarga do cenário com segurança foi maior que do cenário sem segurança. Nos protocolos PROPHET e *Spray and Wait*, esta característica também é verificada,

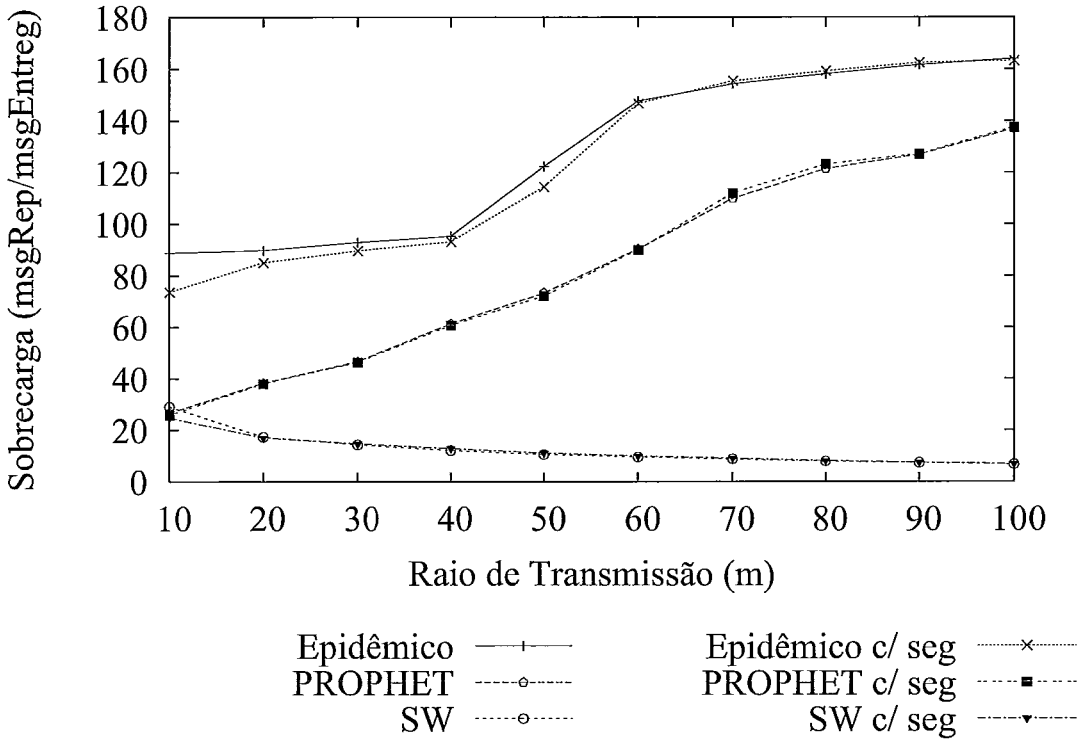

Figura 5.8: Sobrecarga das mensagens de dados variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

porém com menor diferença.

Estes mesmos comentários do cenário real podem ser observados na Figura 5.9, que ilustra a mesma métrica sob a influência do modelo de mobilidade MMIG.

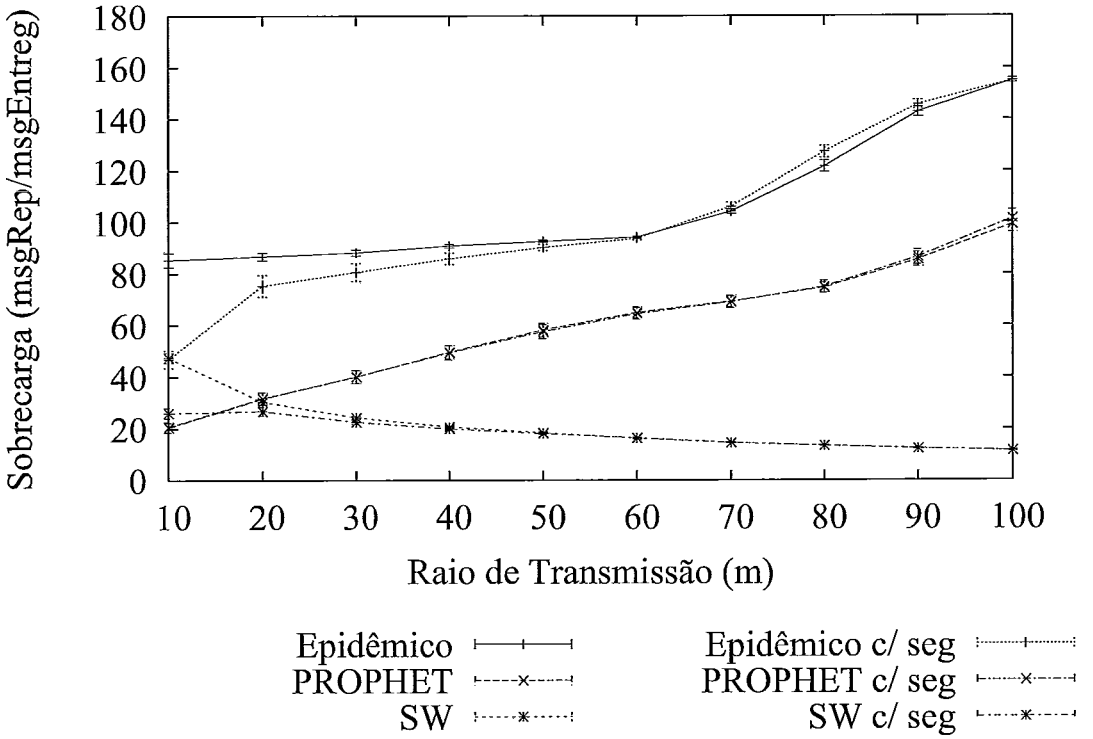

Figura 5.9: Sobrecarga das mensagens de dados variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética MMIG.

Novamente, o protocolo de roteamento Epidêmico apresenta a maior sobrecarga e a sua sobrecarga sem o uso da segurança é maior até o raio de transmissão de 60 metros. Os protocolos PROPHET e *Spray and Wait* apresentam o mesmo comportamento, sendo a diferença da sobrecarga sem e com o uso da segurança muito pequena.

Por fim, a Figura 5.10 ilustra a mesma métrica para a mobilidade sintética gerada pelo modelo RWP e pode-se observar que os comentários efetuados para os resultados obtidos nos cenários de mobilidade real e mobilidade sintética MMIG se adéquam ao cenário de mobilidade gerada pelo modelo RWP.

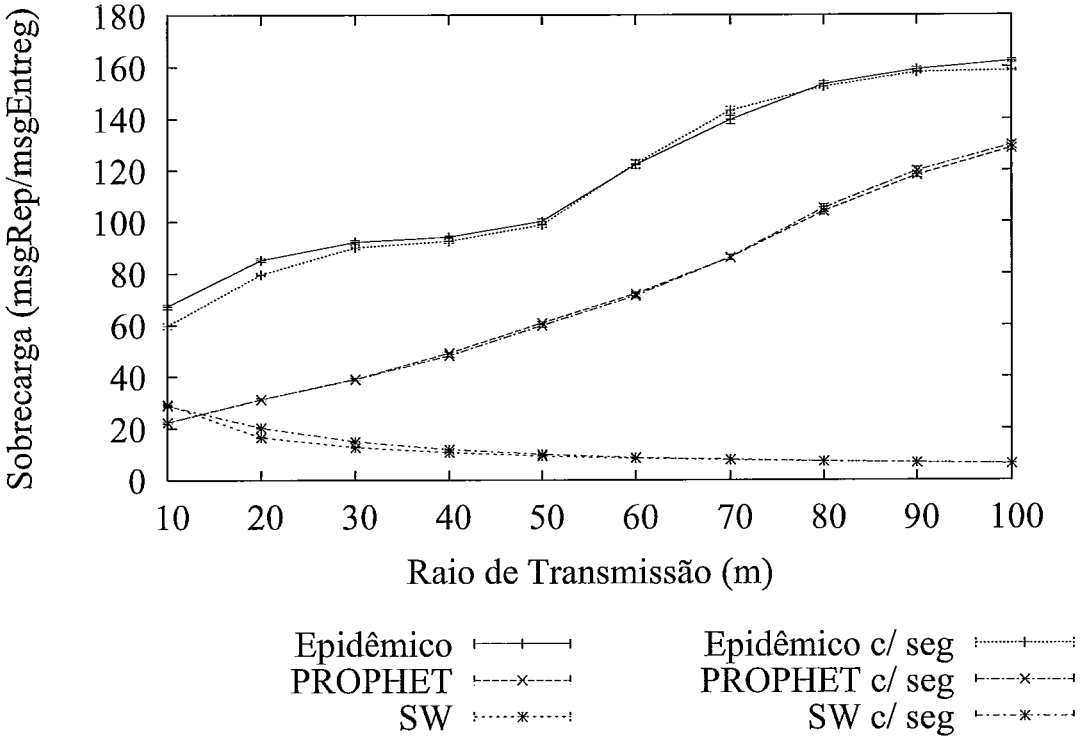

Figura 5.10: Sobrecarga das mensagens de dados variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética gerada pelo modelo RWP.

Pode-se concluir dos resultados apresentados que a inserção do mecanismo de segurança não aumentou a sobrecarga das mensagens de dados significativamente, para os protocolos de roteamento e modelos de mobilidade analisados, a partir do raio 20 metros.

Com isso, pode-se concluir que a troca de mensagens é pouco afetada pela troca de dados de segurança na rede. A próxima métrica ilustra a quantidade de dados de segurança inseridos na rede. A métrica sobrecarga deve estar de acordo com a próxima métrica (sobrecarga de segurança), ou seja, poucos dados de segurança são inseridos na rede, não afetando a troca de dados de mensagens normais.

5.4.5 Sobrecarga de Segurança

A sobrecarga de segurança ilustra a quantidade de bytes de segurança inseridos na rede necessária para prover segurança à troca de mensagens em uma DTN através do método proposto. Os bytes de segurança, essencialmente, são gerados em duas ocasiões: na troca de chaves e na troca de mensagens normais. Os bytes de segurança referentes a troca de chaves não dependem do protocolo de roteamento utilizado e sim da mobilidade, pois as trocas de chaves ocorrem no momento do contato dos nó. Com isso, o resultado desta métrica está diretamente ligado a quantidade de mensagens normais trocadas na rede, ou seja, relacionado as características de troca de mensagens de cada protocolo de roteamento. É esperado que a sobrecarga de segurança seja alta para raios pequenos, devido ocorrência de poucas oportunidades de conexão para a troca de chaves, e a sobrecarga de segurança diminua conforme aumenta o raio.

Na Figura 5.11, a sobrecarga de segurança é apresentado sob a mobilidade real, para diferentes protocolos de roteamento, variando o raio de transmissão. Pelo gráfico observa-se que o protocolo de roteamento Epidêmico apresenta uma sobrecarga de segurança menor que 10% para todos os raios. Esta sobrecarga de segurança é pequena devido a grande troca de mensagens de dados efetuadas pelo protocolo epidêmico.

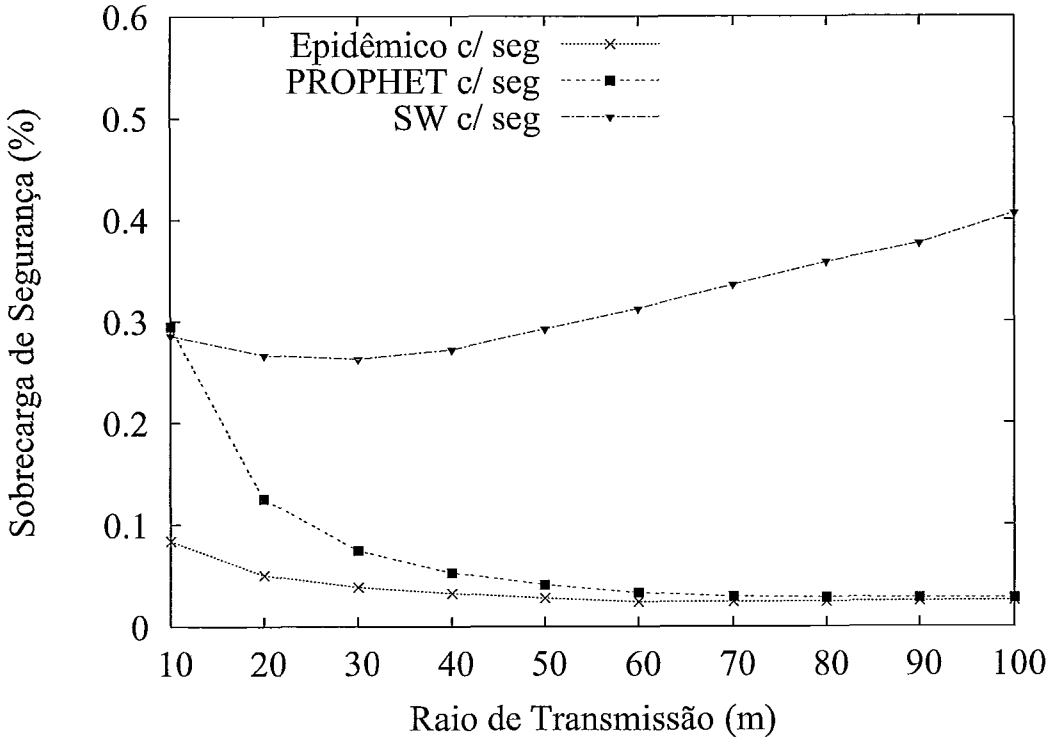

Figura 5.11: Sobrecarga de segurança das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

O protocolo PROPHET, apresenta uma sobrecarga de segurança considerável para raios até 30 metros, porém esta sobrecarga de segurança ficà abaixo de 10% para raios maiores que 30 metros. A maior sobrecarga de segurança deste protocolo, em relação ao protocolo de roteamento Epidêmico, se dá pela menor troca de mensagens de dados na rede, o que acarreta uma menor quantidade de bytes na rede e consequentemente, uma maior proporção de bytes de segurança devido a troca de chaves (que depende apenas da mobilidade).

Já o protocolo *Spray and Wait* apresenta um resultado, a princípio, inesperado. Pelo gráfico da Figura 5.11 nota-se que o protocolo *Spray and Wait* necessita de mais bytes de segurança do que os outros dois protocolos de roteamento. Porém, este resultado se deve a característica de poucos repasses das mensagens na rede deste protocolo, conforme observado na métrica anterior. Como a quantidade de bytes de segurança na troca de chaves depende apenas da mobilidade, o que faz a sobrecarga de segurança diminuir é a quantidade de bytes úteis existentes na troca de mensagens do tipo normal. Assim, como o *Spray and wait* repassa poucas mensagens normais, a quantidade de bytes de segurança na troca de chaves não é suavizado pela quantidade de bytes totais da rede.

Para exemplificar a explicação acima, se fixarmos o raio de transmissão em 50 metros, o protocolo Epidêmico gerou 576.081.324 bytes na rede, onde 15.483.592 bytes (2,5%) foram de segurança e 560.597.732 bytes (97,5%) foram de carga útil das mensagens normais. Dos bytes de segurança, 8.610.248 bytes foram de troca de chaves e 6.873.344 de segurança no campo hash das mensagens normais.

Já no protocolo *Spray and Wait*, para o mesmo raio de transmissão de 50 metros, 39.802.642 bytes foram gerados na rede, onde 9.003.092 (22,6%) foram bytes de segurança e 30.799.550 bytes (77,4%) de carga útil. Nos bytes de segurança, 8.623.956 (95,8%) bytes foi obtido na troca de chaves e 379.136 (4,2%) no campo hash das mensagens normais.

Através desses números, pode-se observar que, conforme descrito, os bytes de segurança relacionados à troca de chaves depende da mobilidade e os relacionados ao campo hash dependem do protocolo de roteamento e, ainda, o impacto da quantidade

de bytes na troca de chaves influencia, e muito, o resultado da métrica sobrecarga de segurança no protocolo *Spray and Wait.*

Espera-se que se o tempo de simulação fosse estendido, a sobrecarga de segurança deste protocolo diminua, como nos outros protocolos pois quando a rede se estabilizar, ou seja, não houver mais trocas de chaves, somente de "Lista de Nós", a quantidade de bytes na troca de chaves será suavizada também pela maior quantidade de mensagens trocadas na rede.

Nas Figuras 5.12 e 5.13, que representam a sobrecarga de segurança para a mobilidade sintética MMIG e para mobilidade sintética RWP, respectivamente, os mesmos comentários do cenário real podem ser observados.

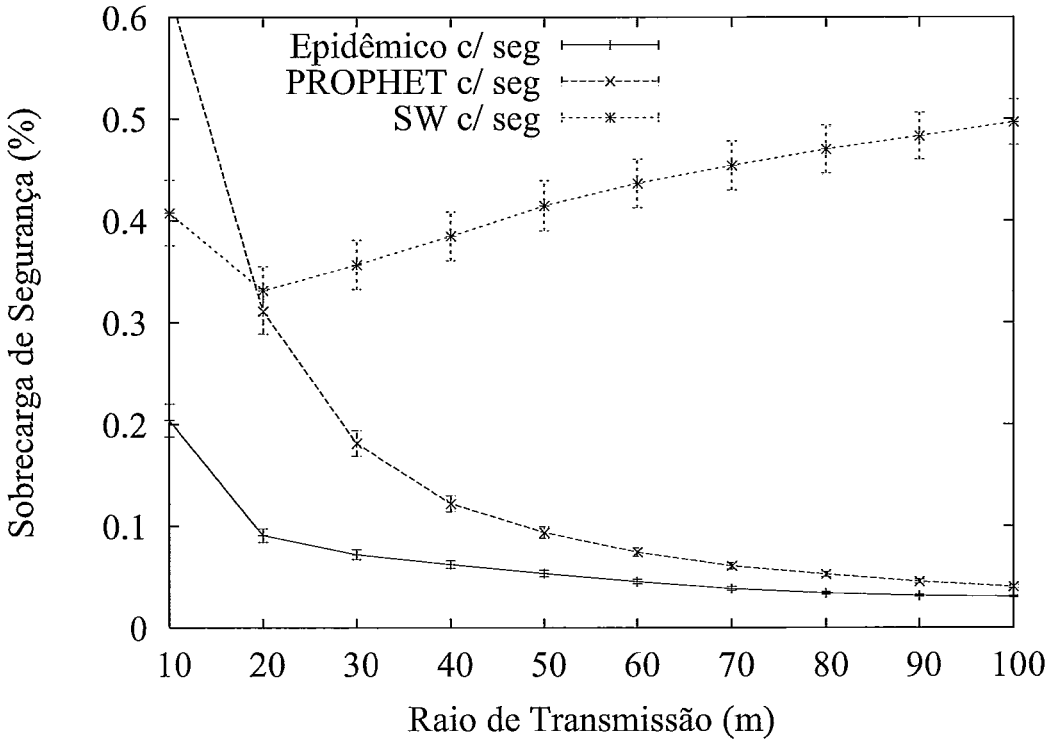

Figura 5.12: Sobrecarga de segurança das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética MMIG.

Pode-se concluir que a sobrecarga de segurança é pequeno, para protocolos de roteamento que repassam muitas mensagens, e é um valor não muito alto, mas con-

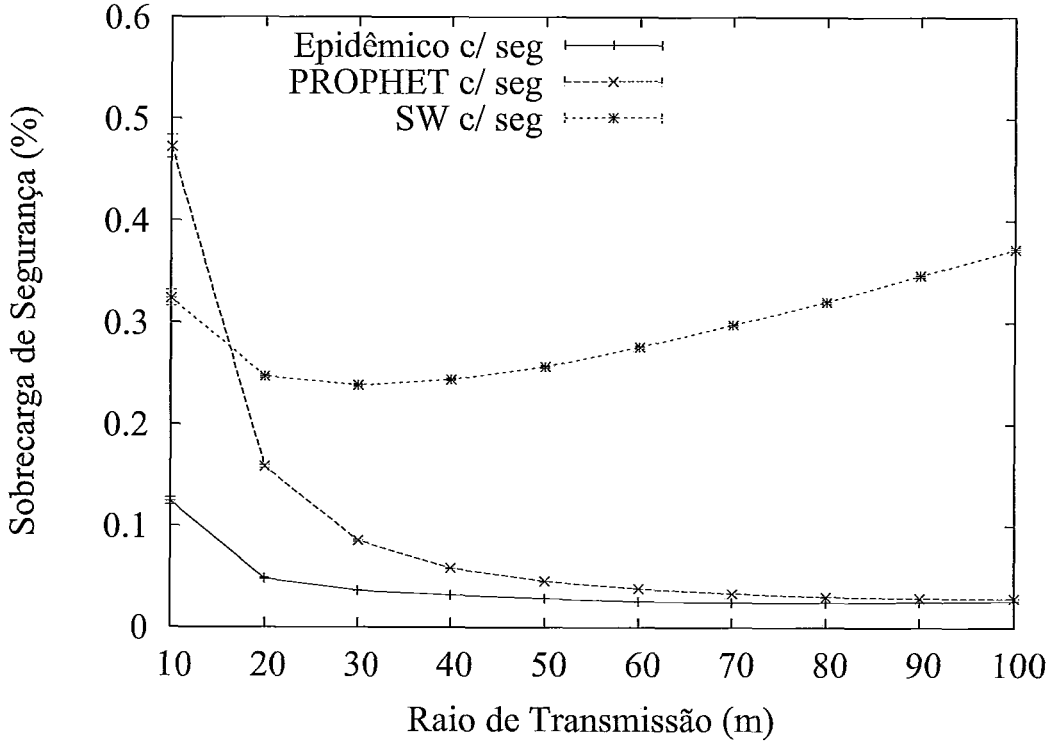

Figura 5.13: Sobrecarga de segurança das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética gerada pelo modelo RWP.

siderável, para protocolos que repassam poucas mensagens. Porém, conforme se aumenta o tempo de simulação, os protocolos de roteamento que repassam poucas mensagens possuem a tendência de diminuir a sobrecarga, até se ajustar aos protocolos que repassam muitas mensagens, devido a diminuição da quantidade de bytes gerados na rede por troca de chaves.

5.4.6 Porcentagem de Mensagens Não Repassadas Devido a Falta de Chave no Nó Origem (α)

A métrica α, que informa a porcentagem de mensagens não repassadas devido a falta de chave no nó origem, ilustra a perda de oportunidade de repasse de mensagens devido a falta de chave. Esta métrica apresenta o impacto do protocolo de segurança no início da troca de uma mensagem. Esta métrica, ao contrário das outras é apresentada em função do tempo de simulação. Espera-se que o valor de α diminua conforme passe o tempo de simulação, indicando que a troca de chaves está convergindo, ou seja, a troca de chaves está finalizando, e que esta convergência seja mais rápida para os maiores raios.

Em todos os cenários serão apresentados os valores de α para os raios 10, 50 e 100 metros, com o objetivo de facilitar a leitura do gráfico. Além disso, esta métrica está diretamente associada a mobilidade dos nós e não ao repasse de mensagens, que depende do protocolo de roteamento. Portanto, somente os resultados do protocolo de roteamento Epidêmico serão apresentados.

Na Figura 5.14 pode ser observado o valor de α para a mobilidade real e protocolo de roteamento Epidêmico. Pode-se observar que a troca de chaves não convergiu para o raio 10 metros nos 600 segundos de simulação pois as oportunidades de contato são poucas, como era esperado. Já no raio 50, a convergência ocorreu depois de 500 segundos de simulação e para o raio 100 metros aos 250 segundos de simulação. Estes resultados estão coerentes com o que era esperado para esta métrica. Pode-se destacar que para raios grandes, a convergência da troca de chaves ocorre em menos de 5 minutos.

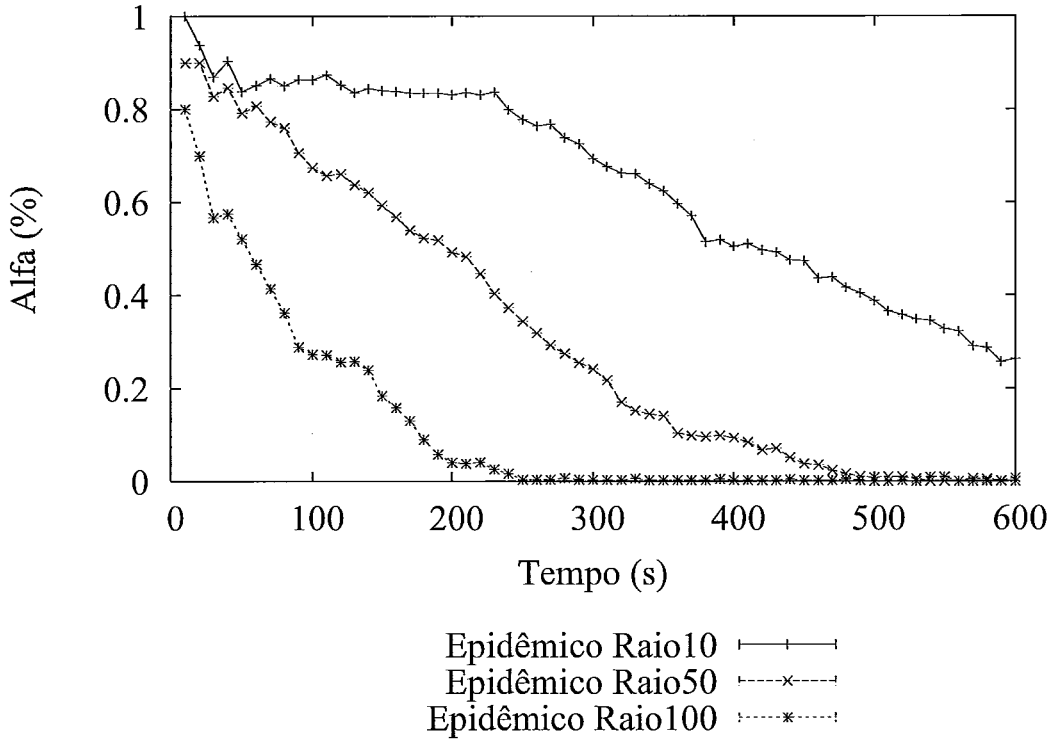

Figura 5.14: Métrica α para os raios de transmissão 10, 50 e 100 metros, para a mobilidade real e protocolo de roteamento Epidêmico.

O gráfico da Figura 5.15 apresenta o valor de α para a mobilidade sintética MMIG e o protocolo de roteamento Epidêmico. Pode-se observar que os resultados seguem a mesma tendência da mobilidade real. Porém, para os raios 50 e 100 metros apresentaram pouca perda de mensagens no início e no raio de 50 metros não havia convergido até os 600 segundos de simulação.

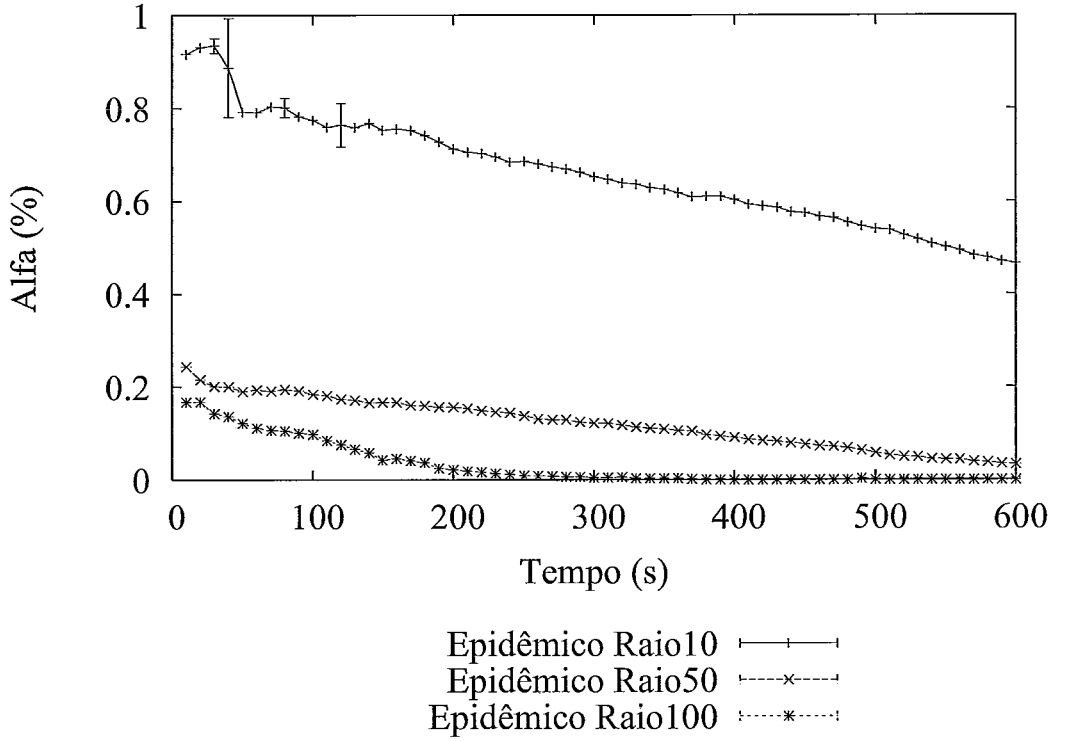

Figura 5.15: Métrica α para os raios de transmissão 10, 50 e 100 metros, para a mobilidade sintética MMIG e protocolo de roteamento Epidêmico.

Por fim, a Figura 5.16 apresenta a métrica α para a mobilidade sintética RWP e o protocolo de roteamento Epidêmico. Os resultados também são similares aos apresentados nos cenários de mobilidade real e sintética MMIG. Assim como na mobilidade real, a métrica α para o raio de 50 metros converge perto dos 500 segundos e para o raio de 100 metros perto dos 200 segundos. O raio 10 de metros novamente não converge nos 600 segundos de simulação.

Uma observação importante que pode ser feita nesta métrica é o tempo mínimo necessário para se implantar a substituição das chaves criptográficas em uma DTN. Por exemplo, para o raio de transmissão 100 metros, se a cada 10 minutos todos

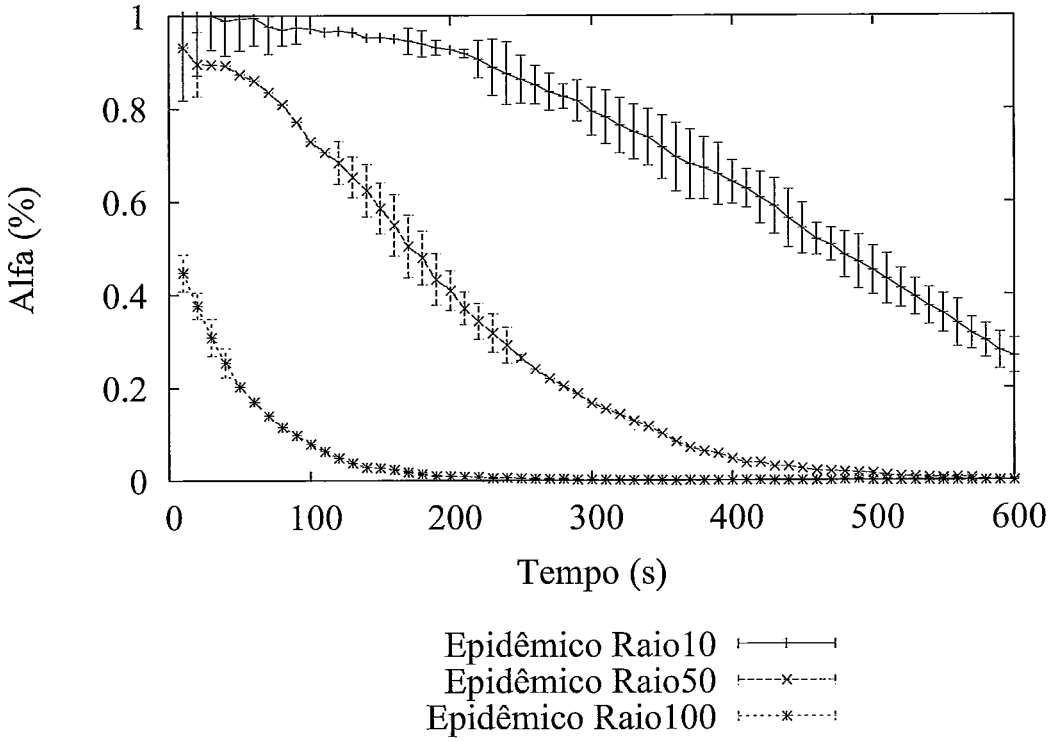

Figura 5.16: Métrica α para os raios de transmissão 10, 50 e 100 metros, para a mobilidade sintética RWP e protocolo de roteamento Epidêmico.

os nós trocassem suas chaves e mantivessem a chave anterior armazenada, a chance de chegar uma mensagem para um nó destinatário criptografada com uma chave pública já descartada é baixo, pois demora cerca de 5 minutos, em média, para a troca de chaves convergir.

A mesma análise pode ser feita para os outros raios. Para o raio de transmissão 50 metros, por volta de 15 minutos pode-se alterar a chave, pois na mobilidade sintética MMIG, a convergência para este raio é mais lenta. Porém, se considerar a mobilidade real e RWP, 10 minutos são suficientes para a convergência da troca de chaves para o raio de transmissão 50 metros.

Numa análise geral de todas as métricas apresentadas nessa seção, os resultados apresentados indicam que o protocolo de segurança implementado para DTNs não insere grande impacto no desempenho da DTN. Os resultados referentes ao desempenho das DTNs, quando se considera raios maiores que 40 metros, apresentaram resultados próximos quando se compara o uso do mecanismo de segurança e o não uso do mesmo. Para exemplificar, na métrica probabilidade de entrega a diferença do resultado sem o uso do mecanismo de segurança e com o uso do mesmo foi de apenas 1,9% para o raio de transmissão de 50 metros. Já os resultados referentes a métricas de segurança mostraram que o mecanismo de segurança insere poucos bytes de segurança na rede, por exemplo para o raio 50 metros do protocolo Epidêmico menos de 4% dos bytes transmitidos na rede foram de segurança.

Porém, neste cenário existe uma alta densidade de nós ocasionando uma grande probabilidade de encontro de nós e com isso, fazendo com que o desempenho da rede seja bom, tanto sem segurança quanto com segurança, pois há maiores oportunidades de troca de dados. Mas as DTNs foram desenvolvidas também para ambientes com poucos nós, com isso existindo poucas oportunidades de troca de mensagens. Para avaliar o desempenho do protocolo de segurança os resultados desta seção foram obtidos também para um cenário de baixa densidade e é apresentado na seção seguinte.

5.5 Cenário de Baixa Densidade

Cenários com poucos nós distribuídos em uma área grande são um desafio para as redes sem fio. Redes sem fio que não possuem a característica de persistência de dados, normalmente não funcionam satisfatoriamente em um ambiente de baixa densidade de nós. Porém, as DTNs foram desenvolvidas para operar em ambientes onde a oportunidade de repasse de mensagens é escassa.

Exemplos de cenários de redes terrestres de baixa densidade são áreas rurais, onde o encontro entre pessoas ocorre esporadicamente, cenários de catástrofes, onde grupos de sobreviventes podem ficar sem comunicação por algum tempo, populações nômades, que podem obter troca de informações ao se aproximar de alguma área povoada, entre outros.

Nas simulações a área de cobertura dos equipamentos foi variada, fazendo com que o número de encontros fosse diferente em cada situação (desde cenários de baixa conectividade até alta conectividade).

A modificação nos parâmetros de simulação neste cenário foi apenas a quantidade de nós presentes na área de simulação, que foi reduzido de 100 nós para 30 nós. Os outros parâmetros foram mantidos. As métricas também foram as mesmas e são apresentadas nas subseções a seguir.

5.5.1 Probabilidade de Entrega

A probabilidade de entrega no cenário de baixa densidade deve obter valores inferiores aos obtidos no cenário de alta densidade. Espera-se, novamente que os resultados desta métrica sejam menores com o uso da segurança do que sem o uso da mesma e que esta diferença diminua conforme se aumente o raio de transmissão. Porém, esta diferença deve ser mais acentuada do que no cenário de alta densidade, pois existem menos possibilidades de roteamento na rede, por haver menos nós na área de simulação.

As Figuras 5.17, 5.18 e 5.19 apresentam a probabilidade de entrega para os

protocolos de roteamento Epidêmico, PROPHET e *Spray and Wait*, variando-se o tamanho do raio de transmissão, para a mobilidade real, MMIG e RWP.

Nas mobilidades real e RWP, houve uma diferença considerável na comparação do não uso da segurança com o uso da mesma para um raio de transmissão até 80 metros. A maior diferença ocorreu no protocolo Epidêmico para o raio de transmissão 40 metros, com uma diferença de 32.7% na mobilidade sintética RWP.

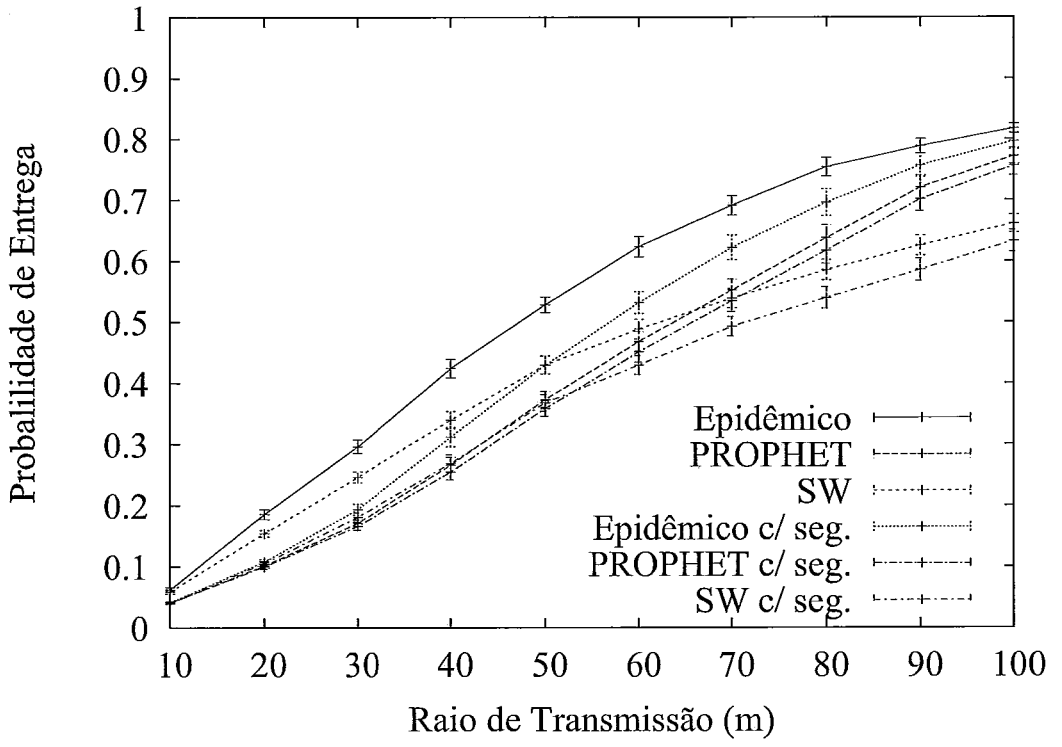

Figura 5.17: Probabilidade de entrega das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

Já na mobilidade MMIG, os resultados gerais foram relativamente piores, se comparado com os cenários da mobilidade real e random waypoint, porém a diferença do não uso e do uso da segurança foi muito pequena.

Apesar desta métrica ter apresentado diferenças de até 32.7% quando se usa a segurança, pode-se observar que os resultados do uso da segurança seguem o comportamento dos resultados sem segurança. É esperado que esta diferença dos

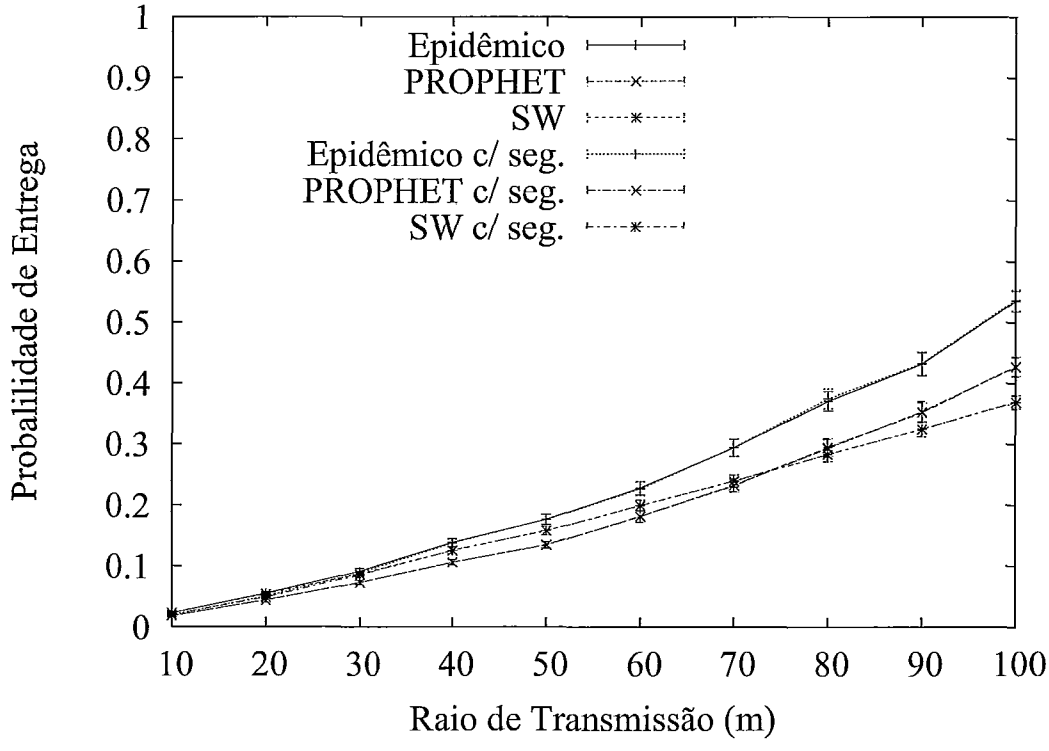

Figura 5.18: Probabilidade de entrega das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética MMIG.

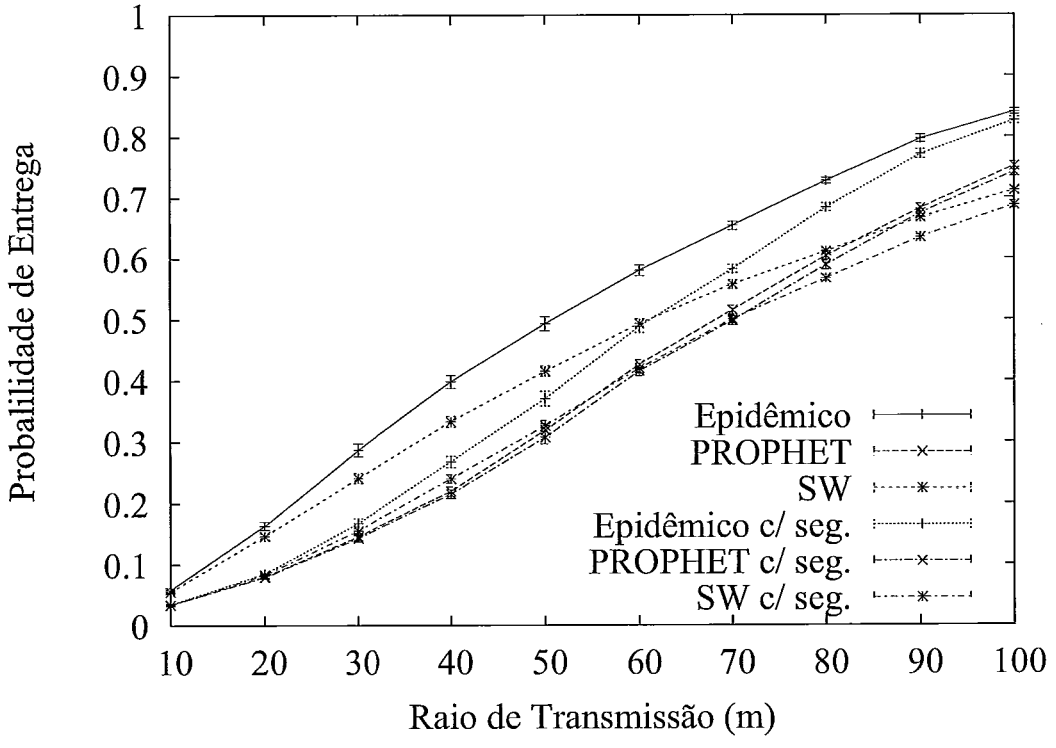

Figura 5.19: Probabilidade de entrega das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética RWP.

resultados com uso de segurança diminua conforme a troca de chaves estabilize, o que deve ser mais demorado no cenário de baixa densidade, do que no cenário de alta densidade.

5.5.2 Atraso Médio

O atraso médio no cenário de baixa densidade deve ser maior do que no cenário de alta conectividade, devido as menores oportunidades de repasse das mensagens. É esperado que o atraso médio das mensagens quando não se usa o mecanismo de segurança seja menor quando comparado com o uso do mesmo. Porém, como esta métrica depende da quantidade de mensagens entregues, pois esta métrica só contabiliza o atraso das mensagens entregues, espera-se uma variação do resultado para raios de transmissão pequenos pois para esses raios houve uma diferença na quantidade de mensagens entregues, conforme a métrica de probabilidade de entrega.

As Figuras 5.20, 5.21 e 5.22 apresentam o atraso médio para a mobilidade real, MMIG e RWP, respectivamente.

Pode-se observar que com o aumento do raio de transmissão o atraso médio na entrega das mensagens diminui, como esperado. Além disso, como existem menos caminhos para o destino neste cenário do que no cenário de alta densidade, a curva do atraso médio quando se aumenta o raio de transmissão possui uma inclinação menor, se comparada com o cenário de alta densidade nos protocolos de roteamento que repassam muitas mensagens.

Além disso, na comparação do atraso médio sem o uso do mecanismo de segurança e com o uso do mesmo, pode-se observar que a diferença entre as curvas do gráfico diminui conforme aumenta o raio de transmissão, como esperado. No protocolo Epidêmico e com os raios menores, o atraso médio foi maior sem o uso da segurança do que com o uso da mesma. Este fato pode ser atribuído a não entrega de mensagens que aguardaram muito tempo a chave mas não chegaram ao destino, com isso mensagens que possuiriam grande atraso não foram contabilizadas na métrica.

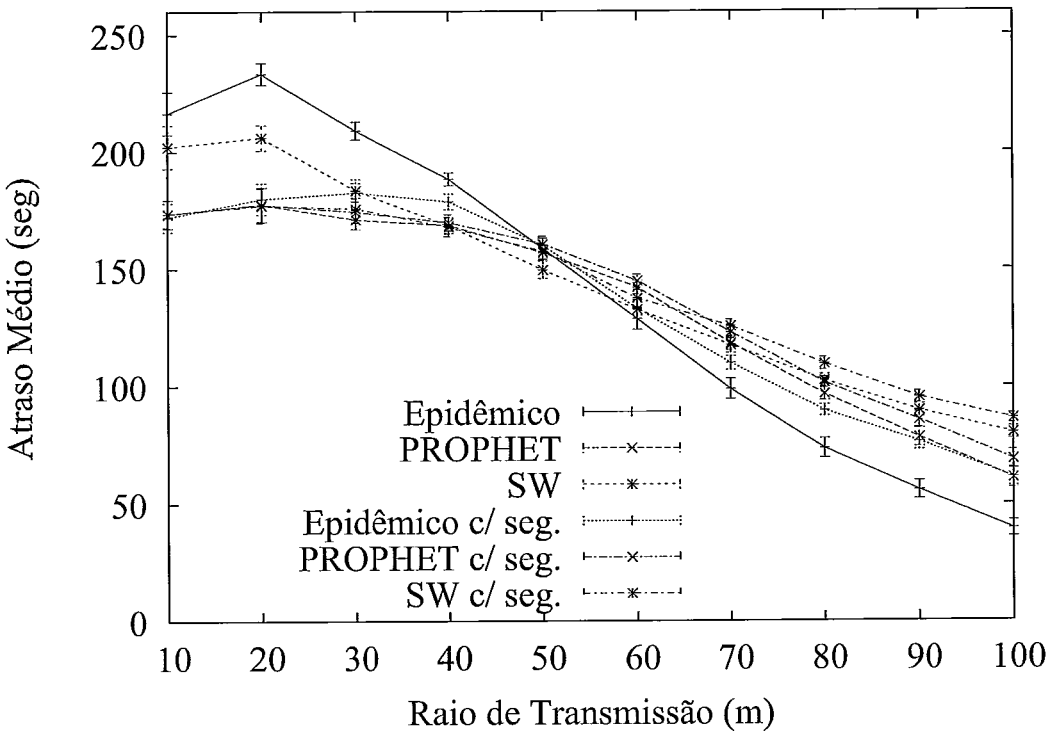

Figura 5.20: Atraso médio das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

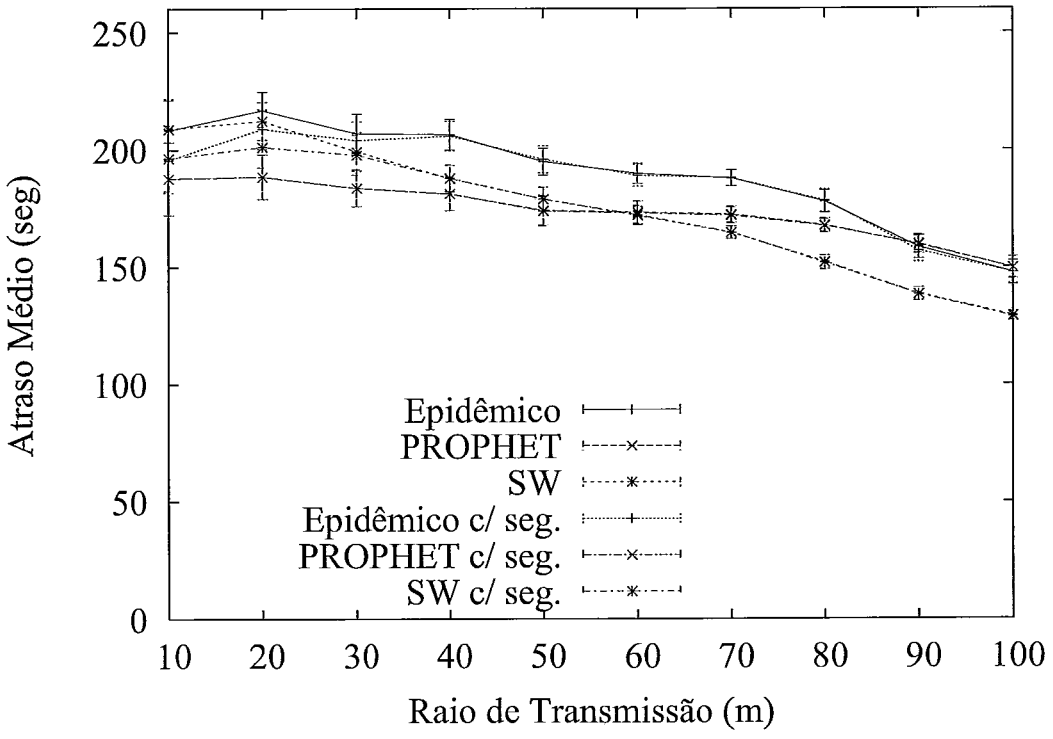

Figura 5.21: Atraso médio das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética MMIG.

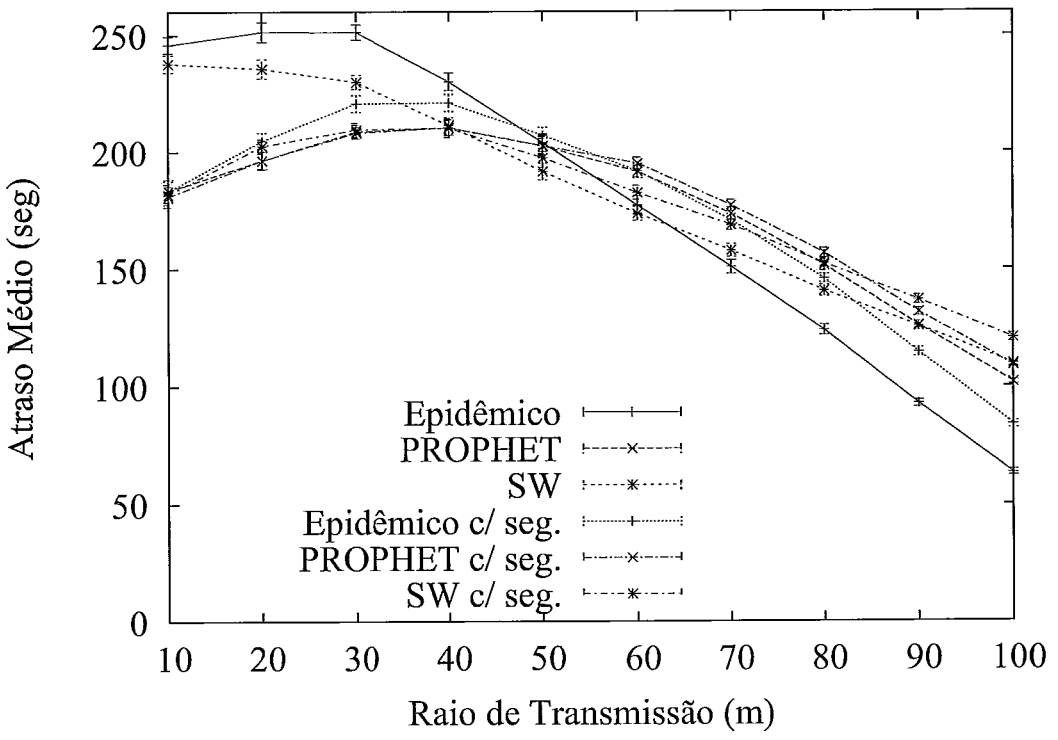

Figura 5.22: Atraso médio das mensagens variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética RWP.

5.5.3 Sobrecarga

O valor da sobrecarga das mensagens de dados no cenário de baixa densidade deve obter menores valores quando comparado com o cenário de alta densidade, devido a menor oportunidade de repasse das mensagens. Para exemplificar, no cenário de alta densidade o protocolo de roteamento Epidêmico poderia efetuar até 99 repasses de uma mesma mensagem na rede. Já no cenário de baixa densidade este valor é no máximo 29.

Quando a conectividade é pequena, ou seja, em cenários onde o raio de transmissão é de até 30 metros, a sobrecarga das mensagens de dados na rede sem o uso do mecanismo de segurança deverá ser bastante superior à sobrecarga das mensagens de dados da rede com o uso do mecanismo de segurança, devido ao menor repasse de mensagens de dados na rede e a demora na troca de chaves do protocolo de segurança.

As Figuras 5.23, 5.24 e 5.25 ilustram a sobrecarga das mensagens de dados na rede para as mobilidades real, MMIG e RWP, respectivamente. Conforme era esperado, a diferença da sobrecarga das mensagens de dados na rede sem o uso da segurança e com o uso da mesma diminui quando o raio de transmissão aumenta. Pode-se observar ainda que para o protocolo de roteamento *Spray and Wait* a sobrecarga das mensagens de dados para raios de transmissão pequenos é superior a 6 e a sobrecarga tende a este valor conforme se aumenta o raio de transmissão. Este comportamento está associado a baixa quantidade de mensagens que chegam ao destino quando o raio de transmissão é pequeno, como ocorreu no cenário de alta densidade.

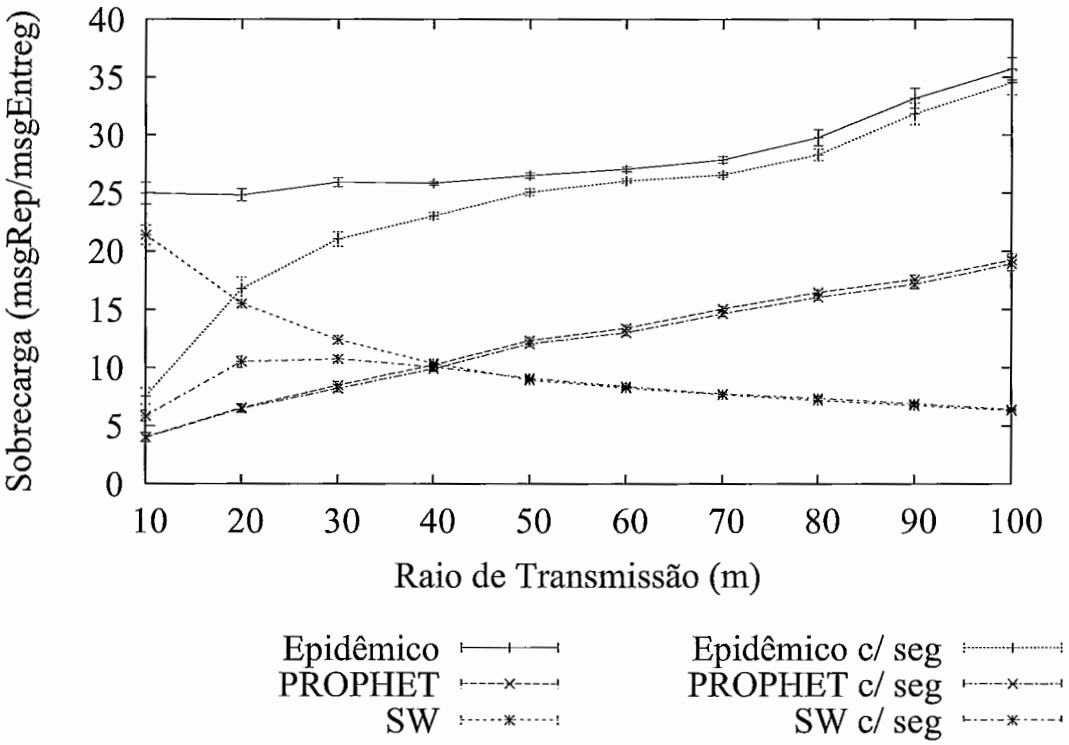

Figura 5.23: Sobrecarga das mensagens de dados variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

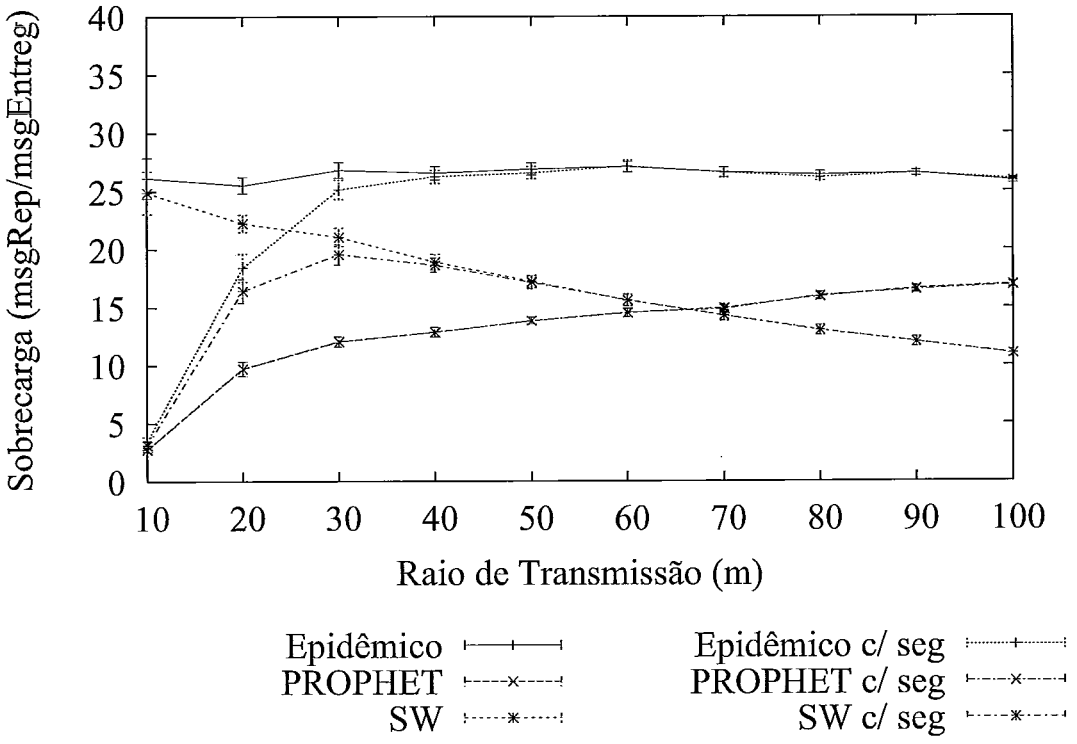

Figura 5.24: Sobrecarga das mensagens de dados variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética MMIG.

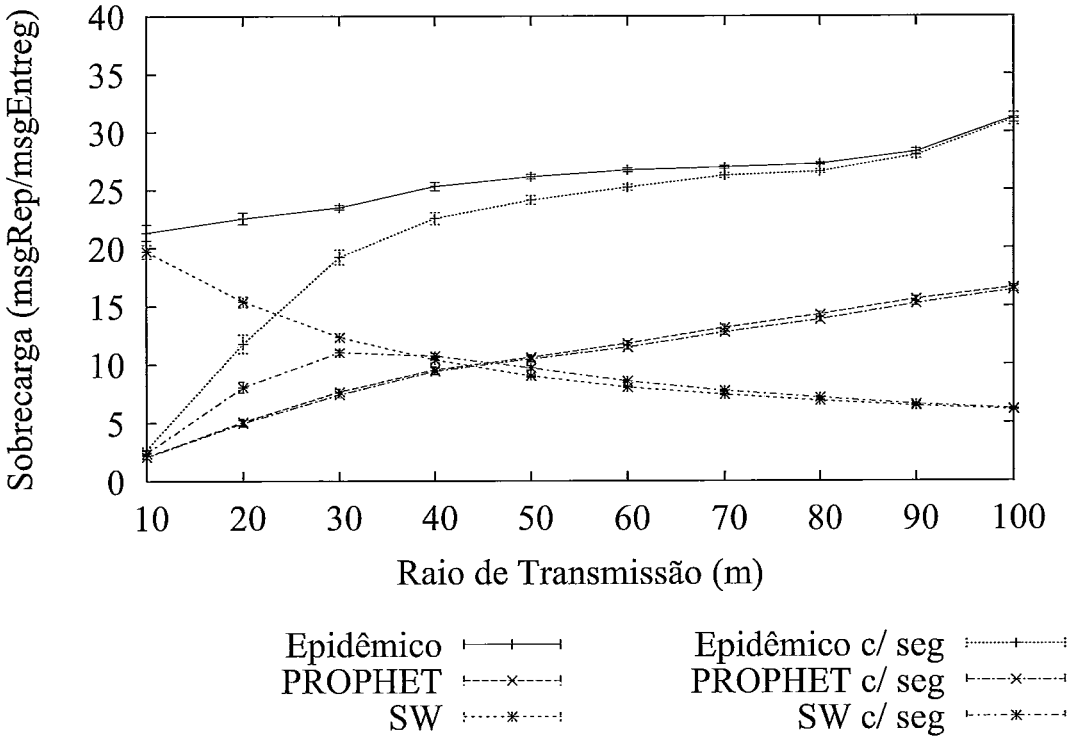

Figura 5.25: Sobrecarga das mensagens de dados variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética RWP.

5.5.4 Sobrecarga de Segurança

A Sobrecarga de segurança em um cenário de baixa densidade possui a tendência de ser menor do que no cenário de alta densidade. Como existem menos nós na área também existem menos chaves a serem trocadas, com isso o tamanho das mensagens de troca de chaves do tipo "Lista de Chaves" é menor, na média, que as mesmas mensagens no cenário de alta densidade. Além disso, a quantidade de troca de chaves é menor, novamente devido a quantidade de nós da rede.

Por esses dois fatores é esperado que a sobrecarga de segurança seja menor no cenário de baixa densidade. Além disso, conforme apresentado na Seção 5.4.5, o protocolo de roteamento *Spray and Wait* apresenta um resultado diferente do esperado, pois a quantidade de bytes na troca de chaves representa uma parcela significativa dos bytes totais trocados na rede. Porém, no cenário de baixa densidade, espera-se que o comportamento do *Spray and Wait* seja parecido dos outros protocolos de roteamento, devido a menor quantidade de bytes utilizados na troca de chaves. As Figuras 5.26, 5.27 e 5.28 apresentam a sobrecarga de segurança em um cenário de baixa densidade, seguindo a mesma ordem dos resultados anteriores.

Pode-se verificar nesses resultados que realmente a menor quantidade de bytes trocados durante a troca de chaves na rede, fez com que a sobrecarga de segurança fosse baixo. Além disso, o protocolo de roteamento *Spray and Wait* obteve um resultado similar aos outros protocolos de roteamento, confirmando a análise feita no cenário de alta densidade de que a influência da quantidade de bytes trocados durante a troca de chaves no protocolo em relação aos bytes totais da rede são os causadores do resultado inesperado para o protocolo de roteamento *Spray and Wait* no cenário de alta conectividade.

Portanto, a sobrecarga de segurança para um cenário de baixa densidade apresentou resultados bastante satisfatórios para todos os protocolos de roteamento e para todas as mobilidades analisadas.

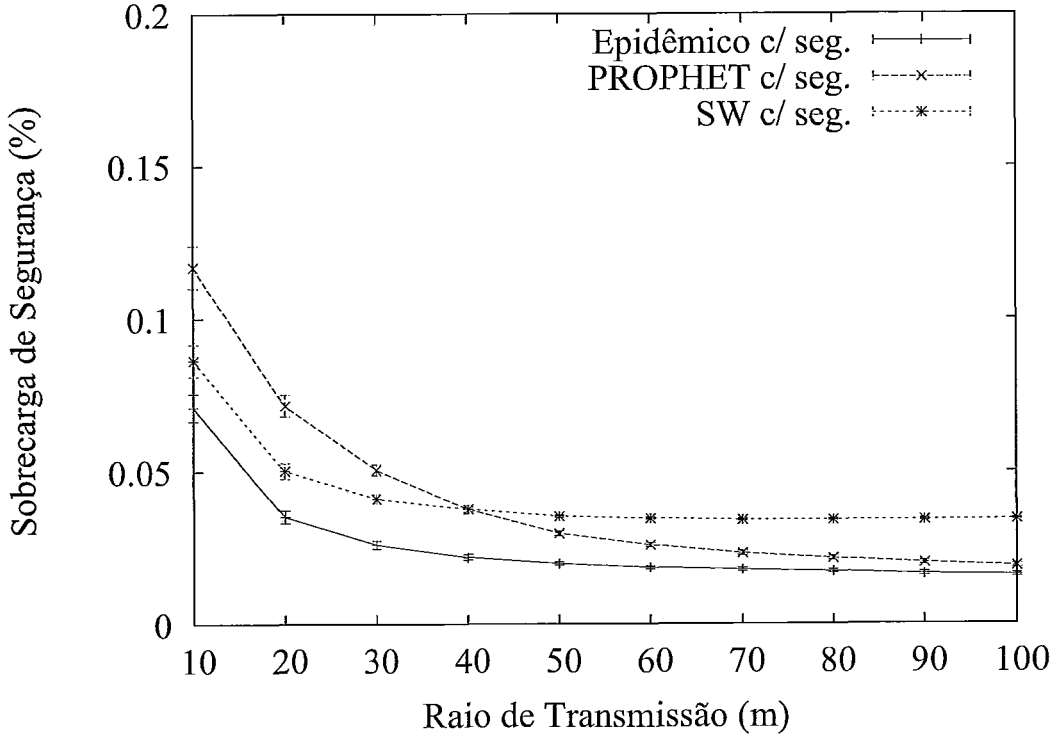

Figura 5.26: Sobrecarga de segurança variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade real.

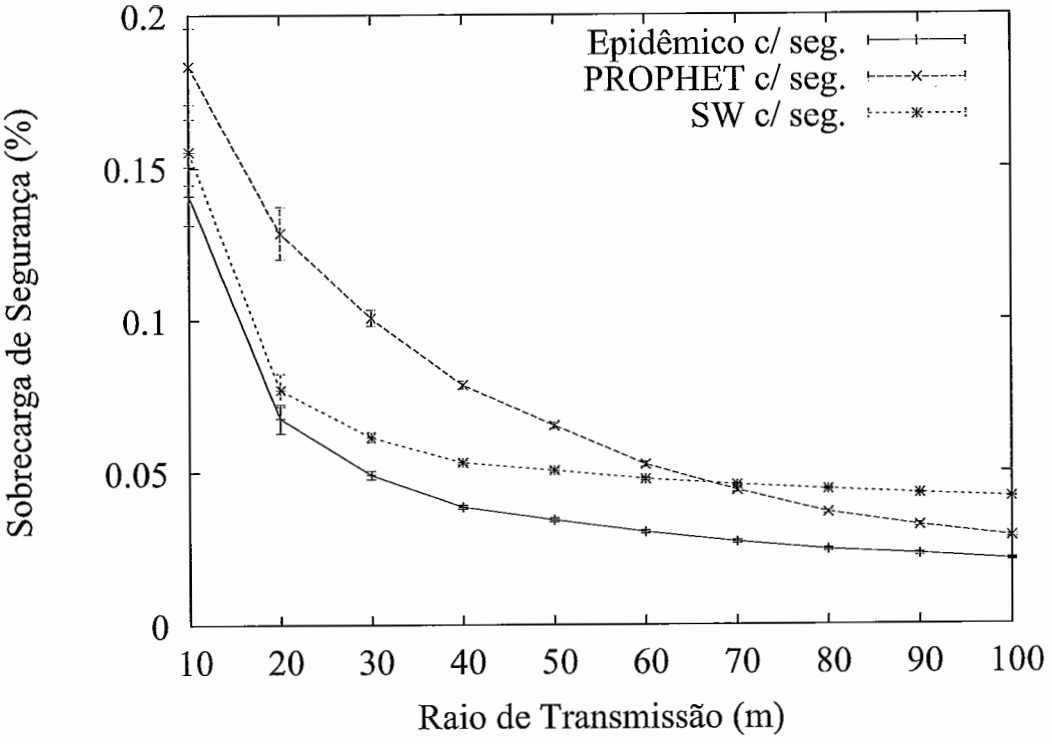

Figura 5.27: Sobrecarga de segurança variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética MMIG.

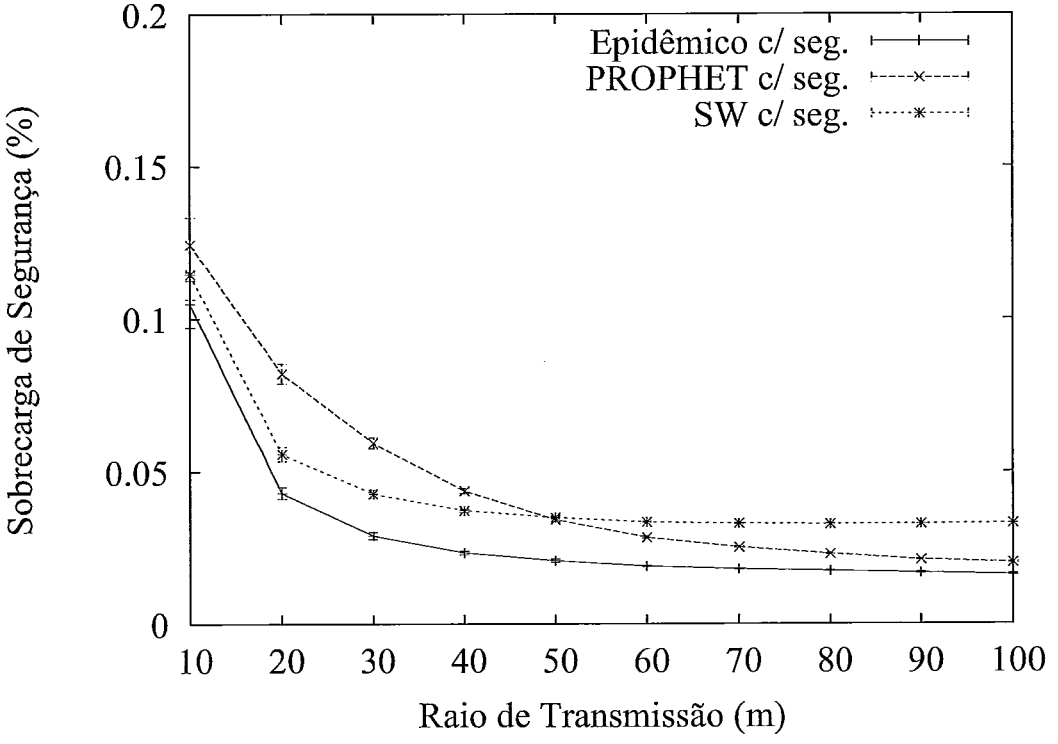

Figura 5.28: Sobrecarga de segurança variando o tamanho do raio de transmissão para diferentes protocolos de roteamento no cenário de mobilidade sintética RWP.

5.5.5 Porcentagem de Mensagens Não Repassadas Devido a Falta de Chave no Nó Origem (α)

Na métrica α é esperado que a convergência da troca de chaves seja mais demorada do que no cenário de alta densidade, devido as menores oportunidades de troca de chaves ocasionadas pela diminuição no número de nós na rede em comparação ao cenário de alta densidade. Porém, como existem menos nós na rede, a probabilidade de uma mensagem gerada na origem ser destinada a outro nó, na qual a origem já possua a chave é maior. Isto é explicado da seguinte maneira, seja X o número de nós no cenário de alta densidade e x o número de nós no cenário de baixa densidade, onde x « X, tem-se que o nó origem pode enviar uma mensagem para qualquer um dos X-1 nós no cenário de alta densidade. Já no cenário de baixa densidade, o nó origem pode enviar uma mensagem para qualquer um dos x-1 nós da rede. Como x « X, a probabilidade do nó origem gerar uma mensagem para um nó destino que ele já possua a chave é maior no cenário de baixa densidade do que no cenário de alta densidade.

Dadas as considerações anteriores, espera-se que para a métrica α, a convergência da troca de chaves seja mais demorada. Os gráficos das Figuras 5.29, 5.30 e 5.31 apresentam os resultados da métrica α. Assim como no cenário de alta densidade, somente os resultados para o protocolo de roteamento Epidêmico, para os raios de transmissão de 10, 50 e 100 metros foram apresentados.

Pode-se observar inicialmente que na mobilidade real e RWP, os resultados esperados foram confirmados. A convergência da troca de chaves foi mais demorada, porém o tempo de convergência não foi muito alto. Por exemplo, no raio de transmissão de 100 metros, o tempo de convergência foi de aproximadamente 550 segundos. No raio de transmissão de 50 metros, a troca de chaves não estabilizou após 600 segundos de simulação, porém após 450 segundos a chance de uma nova mensagem não ser enviada por falta de chave é menor que 20%.

Já no modelo de mobilidade MMIG, os resultados para raio de transmissão de 50 metros e 100 metros foram muito bons, a convergência da troca de chaves foi rápida

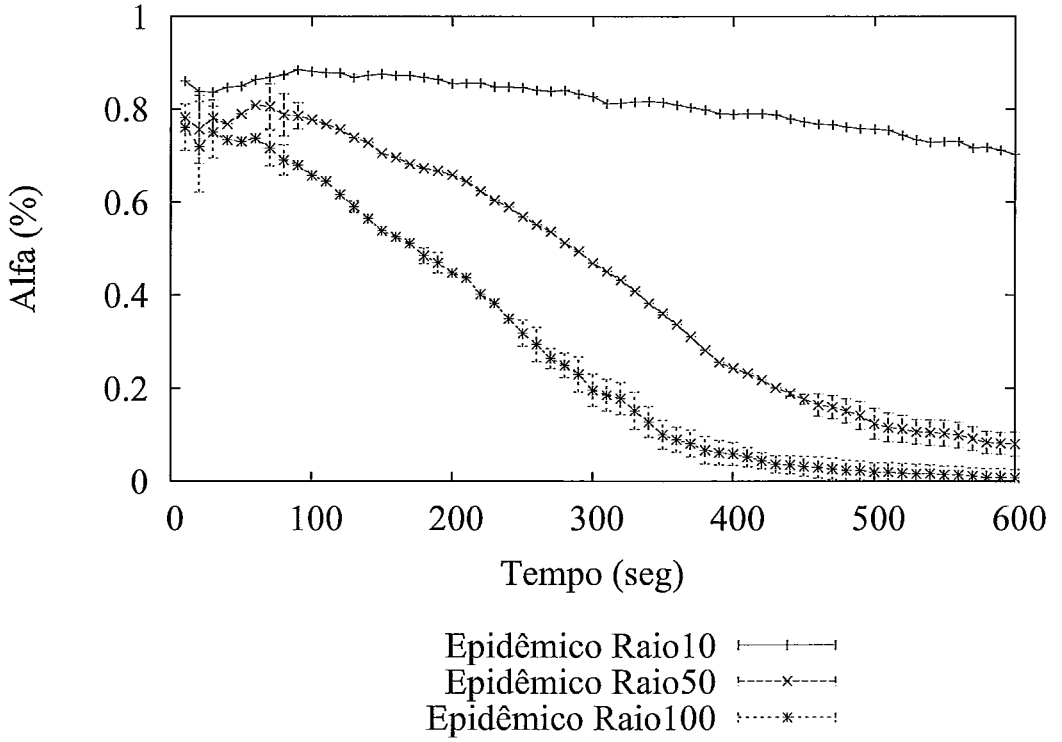

Figura 5.29: Métrica α para os raios de transmissão 10, 50 e 100 metros, para a mobilidade real e protocolo de roteamento Epidêmico.

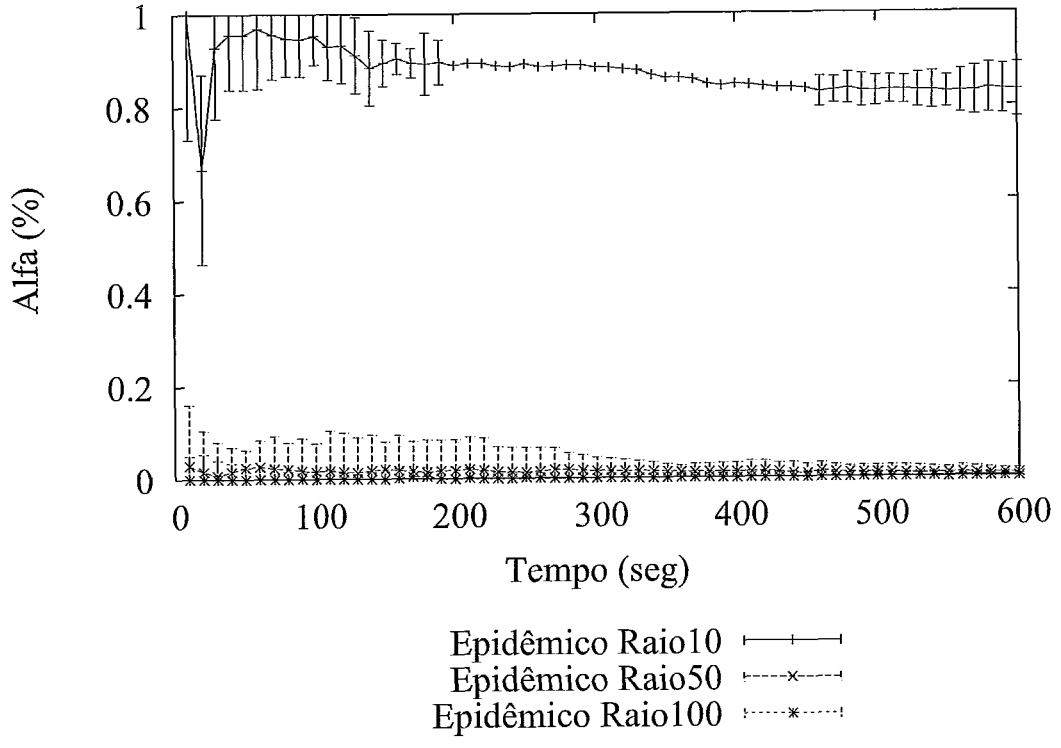

Figura 5.30: Métrica α para os raios de transmissão 10, 50 e 100 metros, para a mobilidade sintética MMIG e protocolo de roteamento Epidêmico.

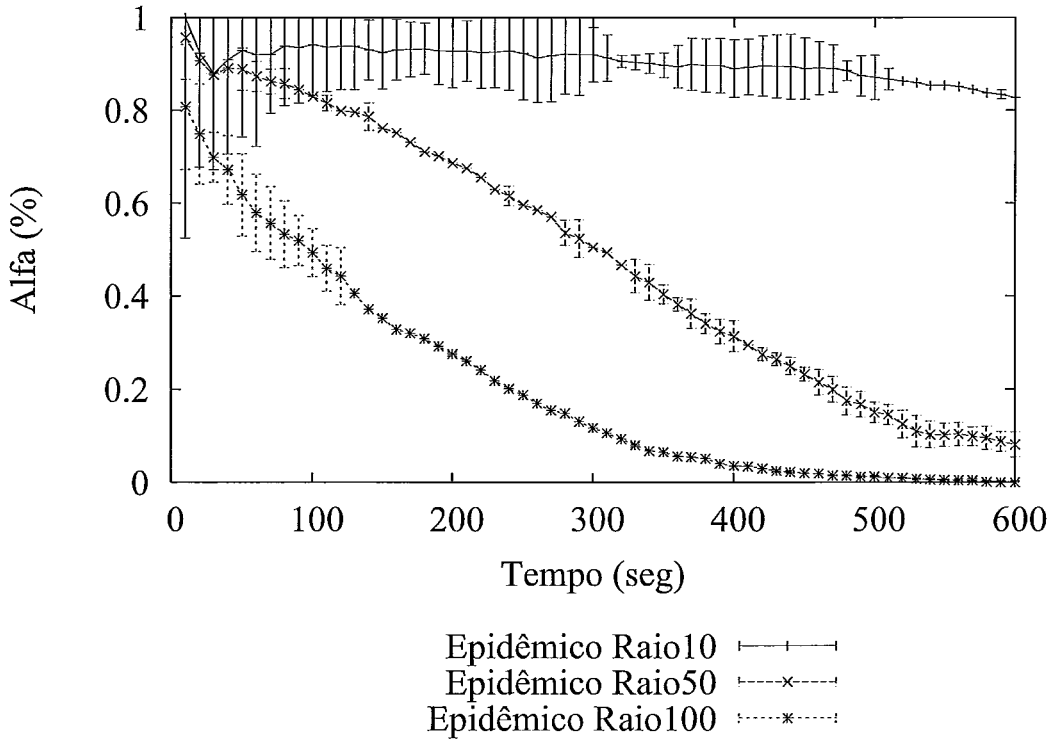

Figura 5.31: Métrica α para os raios de transmissão 10, 50 e 100 metros, para a mobilidade sintética RWP e protocolo de roteamento Epidêmico.

e poucos pacotes não foram enviados por falta de chave. Pode-se observar que para o raio de transmissão 50 metros, a barra de erro para um tempo de simulação de 300 segundos teve seu maior valor próximo de 0.2 (20% de mensanges não enviadas por falta de chave). Mesmo que o resultado fosse próximo a 0.2, o resultado nesta mobilidade foi muito bom.

Uma observação importante para os três gráficos é que a barra de erro é bastante nítida, principalmente para os resultados referentes ao raio de transmissão de 10 metros. Isto mostra que a convergência da troca de chaves é realmente dependente da mobilidade dos nós.

Neste capítulo foram apresentados os resultados da implementação da proposta do mecanismo de segurança para DTNs. Os resultados tanto para um cenário de alta densidade e para um cenário de baixa densidade se mostraram satisfatórios, apesar de alguns resultados um pouco discrepantes quando a conectividade de rede é baixa (raios de transmissão menor que 30 metros). É importante dizer que esses resultados são dependentes do tipo e do tamanho do cenário. Assim, avaliar o mecanismo proposto em outros cenários de uso de uma DTN é importante para sua validação. Esta avaliação pode ser considerada em um trabalho futuro. No próximo capítulo as conclusões referentes a este trabalho serão apresentadas e alguns trabalhos futuros serão sugeridos.

Capítulo 6

Conclusões

ESTE capítulo apresenta as conclusões do trabalho realizado, consolidando os resultados expostos anteriormente e apresentando algumas observações relevantes. Por fim, algumas perspectivas para trabalhos futuros são descritas.

6.1 Conclusão

A utilização de mecanismos de segurança em redes sem fio é essencial atualmente, para garantir a privacidade da comunicação nessas redes. Devido a características de comunicação intermitente das DTNs, os mecanismos tradicionais de segurança não podem ser utilizados na forma padrão.

Como foi descrito na sessão 2.2, alguns trabalhos na literatura propuseram mecanismos de segurança para as DTNs, porém esses mecanismos foram apresentados de maneira incompleta (não especificando como é feita a troca de chaves do mecanismo) ou indicam soluções com problemas (como as soluções baseadas na técnica de HIBC).

No Capítulo 3 o mecanismo de segurança baseado em chaves assimétricas para DTNs foi proposto. Este mecanismo tem como objetivo prover as três premissas básicas de segurança em um rede: confidencialidade, autenticidade e integridade. Estas premissas foram garantidas através do uso das técnicas de assinatura digital e criptografia dos dados. Porém essas técnicas necessitam de chaves criptográficas para funcionarem corretamente. Assim, foi apresentado, no Capítulo 3, um mecanismo de troca de chaves baseado nos contatos oportunísticos, realizados pelos nós da rede. Durante um contato o nó repassa todo seu histórico de conhecimento de chaves e assim, a propagação das chaves na rede ocorre de forma rápida. Além disso, um mecanismo de entrega de mensagens foi proposto e este deve funcionar em sintonia com os protocolos de roteamento existentes para as DTNs.

No Capítulo 5 foram apresentados os resultados de simulações de DTNs em um cenário de alta densidade, onde existiam muitos nós em uma determinada área; e um cenário de baixa densidade, com poucos nós na área de simulação. Para melhor avaliar o desempenho do mecanismo de segurança proposto foram utilizados dois modelos de mobilidade, um contendo a característica de ser sem memória e outro com memória; e rastros de mobilidade humana capturada em um cenário real. O desempenho do mecanismo também foi avaliado sob influência de três protocolos de roteamento diferentes, um protocolo repassador (PROPHET) e dois protocolos

replicadores (Epidêmico e *Spray and Wait*), sendo um sem controle do número de cópias das mensagens e outro com controle do número de cópias das mensagens.

Os resultados mostraram que o mecanismo de segurança não causou uma grande influência no desempenho da rede quando comparado com o desempenho da rede sem o mecanismo de segurança. Mostrou-se que em determinados cenários, a troca de chaves do protocolo de segurança finalizou após 200 segundos, ou seja, em menos de 5 minutos e em outros cenários o mecanismo de segurança ajudou o roteamento das mensagens, ocasionando maior entrega de mensagens quando se usa o mecanismo de segurança do que sem o mesmo.

Dadas essas considerações, as perguntas efetuadas na seção 1.3 podem ser respondidas:

- Existe a possibilidade do uso de um mecanismo de segurança nas DTNs? Sim, foi desenvolvido um mecanismo de segurança para DTNs baseado em chaves assimétricas, onde essas chaves são trocadas durante o tempo de conexão de dois nós.

- Qual é o impacto do uso do mecanismo de segurança proposto no desempenho da rede? No Capítulo 5 observou-se que os resultados indicaram que o mecanismo proposto interfere pouco no desempenho da DTN, principalmente se forem considerados cenários de alta conectividade.

- Existe alguma variação no desempenho das DTNs em função do protocolo de roteamento usado? Apesar dos protocolos de roteamento possuírem desempenhos diferentes, dependendo da métrica, os resultados obtidos com o uso do mecanismo de segurança apresentaram um comportamento similar aos resultados obtidos sem o uso do mesmo, para todos os protocolos de roteamento, em todas as métricas.

- O mecanismo de segurança funciona de forma adequada tanto para cenários de alta densidade como de baixa densidade? Sim, apesar dos resultados indicarem um melhor desempenho nos cenários de alta densidade, os cenários de baixa densidade também apresentaram um bom desempenho.

Com isso, pode-se concluir que o mecanismo de segurança, proposto neste trabalho, pode ser uma boa solução para o problema da falta de segurança na troca de mensagens nas redes tolerantes a atrasos e desconexões, para os cenários e parâmetros utilizados.

6.2 Trabalhos Futuros

Este trabalho apresentou um mecanismo de segurança baseado em chaves assimétricas para DTNs e este mecanismo foi avaliado através de simulações que mostraram que o impacto do mecanismo de segurança no desempenho da rede é bem pequeno. Porém, com objetivo de melhorar o mecanismo de segurança e a possibilidade de melhor adequação da solução proposta para DTNs, nesta Seção serão descritas algumas perspectivas de trabalhos futuros:

- Otimizar o armazenamento das chaves na memória dos nós para diminuir a utilização da memória pelo mecanismo de segurança, com isso permitindo mais mensagens armazenadas em cada nó;
- Inserir um mecanismo de alteração de chaves criptográficas após um determinado tempo para aumentar a segurança do mecanismo;
- Avaliar outros algoritmos de criptografia assimétricos e compará-los com o RSA com a intenção de verificar se o processamento necessário na criptografia e decriptografia pode influenciar o desempenho da rede;
- Variar o tamanho das chaves criptográficas geradas pelo mecanismo para avaliar se chaves de tamanho maior causam grande influência no desempenho da rede;
- Variar o tamanho das mensagens de dados e avaliar o impacto que as mensagens de dados, de tamanhos diferentes, pode causar no mecanismo de segurança;

- Inserir perda no meio de transmissão, para melhorar a representação do funcionamento do mecanismo em um cenário mais realista;
- Utilizar outros rastros de traces reais para verificar se o desempenho do mecanismo proposto se mantêm adequado;
- Implementar este mecanismo de segurança em equipamentos de comunicação em DTN e verificar se os resultados obtidos se mantêm em um ambiente real.
- Comparar o mecanismo de segurança proposto com outros mecanismos de segurança de DTNs que utilizam distribuição de chaves centralizada.

Bibliografia

[1] K. Fall and S. Farrell, "Dtn: an architectural retrospective," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 5, pp. 828–836, June 2008.

[2] C. A. V. Campos and L. F. M. de Moraes, "A markovian model representation of individual mobility scenarios in ad hoc networks and its evaluation," *EURASIP J. Wirel. Commun. Netw.*, vol. 2007, no. 1, pp. 35–35, 2007.

[3] E. Hargreaves, "Um novo modelo de mobilidade para redes sem fio: Distribuições exatas para velocidade e direção e aplicações em simulações," *Tese de Mestrado, Coppe/UFRJ*, set 2006.

[4] Wikipédia, "Eniac," http://pt.wikipedia.org/wiki/ENIAC.

[5] ——, "Arpanet," http://pt.wikipedia.org/wiki/ARPANET.

[6] T. Socolofsky and C. Kale, "Tcp/ip tutorial - rfc1180," United States, 1991.

[7] K. Fall, "A delay-tolerant network architecture for challenged internets," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications.* New York, NY, USA: ACM, 2003, pp. 27–34.

[8] M. Demmer, E. Brewer, K. Fall, M. Ho, R. Patra, M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing delay tolerant networking," IRB-TR-04-020, Intel Research Berkeley, Tech. Rep., 2004.

[9] F. Warthman, "Delay-tolerant networks (dtns): A tutorial," March 2003.

[10] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *Secure Network Protocols, 2005. (NPSec). 1st IEEE ICNP Workshop on*, Nov 2005.

[11] G. Z. Aniket Kate and U. Hengartner, "Anonymity and security in delay tolerant networks," *3rd International Conference on Security and Privacy in Communication Networks (SecureComm 2007)*, 2007.

[12] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo, "Applicability of identity-based cryptography for disruption-tolerant networking," in *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking.* New York, NY, USA: ACM, 2007, pp. 52–56.

[13] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[14] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking.* New York, NY, USA: ACM Press, 2005, pp. 237–243.

[15] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," in *ASIACRYPT-2002*, may 2002, pp. 548–566.

[16] J. Burgess, G. D. Bissias, M. D. Corner, and B. N. Levine, "Surviving attacks on disruption-tolerant networks without authentication," in *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing.* Montreal, Quebec, Canada: ACM Press, 2007, pp. 61–70.

[17] "Internet research task force," available at http://www.irtf.org/.

[18] "Delay tolerant networking research group," available at http://www.dtnrg.org/.

[19] H. W. S. Symington, S. Farell and P. Lovell, "Bundle security protocol specification," September 2009.

[20] S. Farrell and V. Cahill, "Security considerations in space and delay tolerant networks," in *SMC-IT '06: Proceedings of the 2nd IEEE International Conference on Space Mission Challenges for Information Technology.* Washington, DC, USA: IEEE Computer Society, 2006, pp. 29–38.

[21] D. W. Chris Karlof, Naveen Sastry, "Tinysec: A link layer security architecture for wireless sensor networks," *ACM Sensys*, pp. 162–175, November 2004.

[22] A. P. M. Luk, G. Mezzour and V. Gligor, "Minisec: A secure sensor network communication architecture," *International Conference on Information Processing in Sensor Networks-IPSN*, pp. 479–488, April 2007.

[23] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "SPINS: Security protocols for sensor networks," in *Seventh Annual International Conference on Mobile Computing and Networks (MobiCOM 2001)*, Rome, Italy, July 2001, pp. 521–534.

[24] J. Jeong and Z. J. Haas, "Predeployed secure key distribution mechanisms in sensor networks: current state-of-the-art and a new approach using time information," *Wireless Communications, IEEE*, vol. 15, pp. 42–51, 2008.

[25] S.-L. Wu and Y.-C. Tseng, *Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks.* CRC Press, 2007.

[26] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 309–329, September 2003.

[27] S. Capkun, J. Hubaux, and L. uttyán, "Mobility helps security in ad hoc networks," in *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, June 2003, pp. 46–56.

[28] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, pp. 483–502, 2002.

[29] J. Broch, D. A. Maltz, D. B. Johnson, Y. chun Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *MobiCom'98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998, pp. 85–97.

[30] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Comm. & Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[31] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: the incredibles," *SIGMOBILE Mobile Comput. and Comm. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[32] J. Yoon, M. Liu, and B. Noble, "A general framework to construct stationary mobility models for the simulation of mobile netwoks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, pp. 860–871, 2006.

[33] R. L. Bezerra, C. A. V. Campos, T. S. Azevedo, and L. F. M. de Moraes, "An analysis of human mobility using real traces." in *2009 IEEE Wireless Communications and Networking Conference (WCNC 2009)*, 2009.

[34] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," 2003. [Online]. Available: citeseer.ist.psu.edu/lindgren03probabilistic.html

[35] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *Proc. IEEE INFOCOM*, April 2006. [Online]. Available: http://prisms.cs.umass.edu/brian/pubs/burgess.infocom2006.pdf

[36] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," CS-200006. Duke University, Tech. Rep., Apr 2000.

[37] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 373–384, 2007.

[38] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking.* New York, NY, USA: ACM, 2005, pp. 252–259.

[39] ——, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 79–85.

[40] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *SIMUTools '09: Proceeding of the 2nd International Conference on Simulation Tools and Techniques.* New York, NY, USA: ACM, 2009.

[41] J. Karvo and J. Ott, "Time scales and delay-tolerant routing protocols," in *CHANTS '08: Proceedings of the third ACM workshop on Challenged networks.* New York, NY, USA: ACM, 2008, pp. 33–40.

[42] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models.* New York, NY, USA: ACM, 2008, pp. 33–40.

[43] J. G. Mikko Pitkänen, Teemu Kärkkäinen and J. Ott, "Searching for content in mobile dtns," in *7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE, 2009.

[44] R. L. Bezerra, C. A. V. Campos, and L. F. M. de Moraes, "Uma proposta de técnica para o ajuste de modelos de mobilidade em redes ad hoc e questionamentos sobre a adequação dos parâmetros envolvidos com base em dados reais." in *XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos(SBRC).* SBC, 2009.

[45] U. States., *Data encryption standard / U. S. Department of Commerce, National Bureau of Standards.* The Bureau ; for sale by the National Technical Information Service, Washington : Springfield, Va. :, 1977.

[46] A. N. S. Institute, "Triple data encryption algorithm modes of operation," 1998.

[47] W. Stallings, "The advanced encryption standard," *Cryptologia*, vol. XXVI, no. 3, pp. 165–188, 2002.

[48] J. luc Beuchat, "Fpga implementations of the rc6 block cipher," in *Field-Programmable Logic and Applications, number 2778 in Lecture*, 2003.

[49] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, February 1978.

[50] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[51] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[52] V. S. Miller, "Use of elliptic curves in cryptography," 1985, pp. 417+.

[53] J. C. Barbosa, "Uma proposta de utilização de curvas elípticas na criptografia baseada em identidades e sua aplicação na troca segura de mensagens," *Tese de Mestrado, Coppe/UFRJ*, Abril 2005.

[54] S. C. Coutinho, *Números Inteiros e Criptografia RSA*. IMPA, 2000.

[55] N. institute of standards and technology, "Fips 180-2, secure hash standard, federal information processing standard (fips), publication 180-2," DEPARTMENT OF COMMERCE, Tech. Rep., August 2002.

[56] N. Ferguson and B. Schneier, *Practical Cryptography*. Wiley, 2003.

[57] X. Wang, Y. L. Yin, and H. Yu, *Finding Collisions in the Full SHA-1*, November 2005, vol. 3621. [Online]. Available: http://dx.doi.org/10.1007/11535218_2

[58] A. Kerckhoffs, "La cryptographie militaire," *Journal des sciences militaires*, vol. IX, pp. 5–83, January 1883.

Apêndice A

Conceitos Sobre Premissas e Técnicas Usadas para Garantir Segurança

EM termos de segurança de comunicação em um rede de dados, três premissas são consideradas essenciais para garantir segurança da troca de mensagens: confidencialidade, integridade e autenticidade. Confidencialidade significa que apenas o destinatário da mensagem poderá ler o conteúdo útil da mesma (*payload*). Qualquer outro nó que receber a mensagem não será capaz de entender o *payload* da mesma. Já a integridade garante que a mensagem é a que foi enviada sem alterações. Alterações na mensagem podem ocorrer de maneira maliciosa, através de mudanças no seu *payload*, ou por perda de informação no momento de uma transmissão. Por fim, a autenticidade garante que a mensagem foi realmente enviada pelo emissor da mesma e não forjada por outro nó. Para cada premissa existe uma técnica para garanti-la. Essas técnicas serão apresentadas neste apêndice, de acordo com as premissas apresentadas.

A.1 Confidencialidade

Sejam Alice e Bob, duas pessoas necessitando se comunicar de modo seguro, e Eve uma outra pessoa no canal de comunicação de Alice e Bob. Alice deseja se comunicar com Bob de modo que Eve não intercepte a mensagem ou, caso Eve intercepte a mensagem, ela não possa entender o conteúdo da mesma. A Figura A.1 ilustra esta situação.

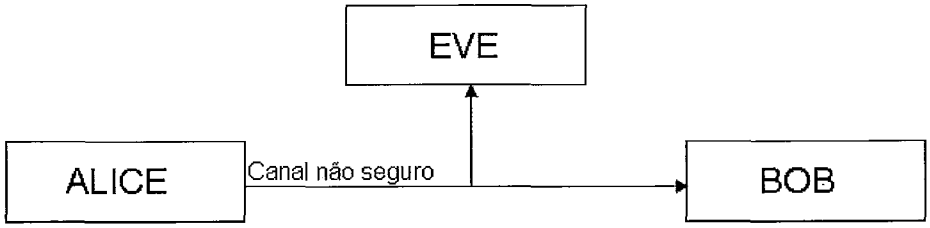

Figura A.1: Alice deseja se comunicar com Bob de maneira segura. Eve está escutando o canal inseguro

A premissa de segurança que permite que Alice e Bob se comuniquem de maneira segura, ou seja, nenhuma outra pessoa saberá o conteúdo da comunicação, é chamada de confidencialidade. Para prover a confidencialidade, o emissor deve encriptar a mensagem original, chamada de texto plano, obtendo uma mensagem cifrada e o destinatário deve ser capaz de, a partir da mensagem cifrada, retornar a mensagem original. Esta técnica é chamada criptografia.

Para se criptografar uma mensagem, uma função matemática transforma o texto plano em um texto cifrado de forma que se possa efetuar a função inversa, ou seja, a partir do texto cifrado obter o texto plano. Porém, esta função matemática não pode ser igual para criptografar todas as mensagens, pois o atacante (Eve) pode descobrir esta função matemática e decriptografar o pacote. Para tanto, essas funções matemáticas necessitam de um parâmetro que faça com que o uso da mesma função matemática resulte em resultados diferentes. Este parâmetro é a chave criptográfica. Os algoritmos de criptografia usam um par de chaves, uma chave para criptografar, chamada chave pública, e uma para decriptografar, chamada chave privada. Os algoritmos de criptografia podem ser classificados em:

- **Algoritmos Simétricos** - Utilizam a mesma chave para criptografar e decriptografar. Os algoritmos de chave-simétrica podem ser divididos em cifras de fluxo e em cifras por bloco. As cifras de fluxo cifram os bits da mensagem um a um, enquanto que as cifras por bloco pegam num número de bits e cifram como uma única unidade. Apesar deste tipo de algoritmo de criptografia ser computacionalmente mais rápido em relação aos algoritmos de criptografia assimétricos, existe a dificuldade de se compartilhar a chave secreta, entre a origem e destino, de maneira segura. Exemplos de algoritmos simétricos: DES [45], TRIPLEDES [46], AES [47], RC6 [48];

- **Algoritmos Assimétricos** - Possuem duas chaves: A chave pública, que pode ser distribuída livremente, e a chave privada, que deve ser apenas de conhecimento do dono. A chave pública é utilizada para criptografar uma mensagem e a chave privada, correspondente, utilizada para decriptografar a mensagem. Exemplos de algoritmos assimétricos: RSA [49], Diffie-Hellman [50], El Gamal [51] e Curvas Elípticas [52].

Apesar do algoritmo de curvas elípticas apresentar um nível de segurança maior do que o RSA para o mesmo tamanho de chave e ser computacionalmente mais rápido [53], o algoritmo de criptografia implementado neste trabalho foi o RSA, devido a sua ampla utilização em trabalhos acadêmicos e comercialmente, vasta literatura disponível e pela segurança oferecida por este algoritmo [54]. Uma explicação sobre o funcionamento desse algoritmo será apresentada a seguir.

A.1.1 RSA

O algoritmo RSA foi inventado por Ron Rivest, Adi Shamir e Len Adleman do Instituto MIT. O algoritmo leva o nome dos três inventores. Seu funcionamento é simples e é baseado na teoria dos números. Em [54], o autor explica toda teoria de números necessária para compreender o algoritmo RSA.

O algoritmo inicialmente escolhe as chaves assimétricas (pública e privada) para a encriptação e decriptação. O processo da escolha de chaves ocorre da seguinte

maneira:

1. Escolha de forma aleatória dois números primos grandes p e q;

2. Compute $n = pxq$;

3. Calcule $\phi(n) = (p-1)(q-1)$;

4. Escolha um inteiro e, de tal forma que mdc(e,$\phi(n)$) = 1 e $1 \leq e \leq \phi(n)$;

5. Compute d, de forma que $dxe \equiv 1 mod(\phi(n))$, ou seja, d seja o inverso multiplicativo de e em $mod(\phi(n))$.

Após a execução dos cinco passos anteriores temos que a chave privada é o par (n, d) e a chave pública é o par (n, e).

Seja M a mensagem a ser cifrada e $C(M)$ o resultado da cifra da mensagem M. Para efetuar a codificação é aplicada a seguinte fórmula:

$C(M) = M^e mod(n)$

Para decifrar a mensagem é computado:

$M = C(M)^d mod(n)$

Assim, usando técnicas matemáticas não complexas, é possível garantir a confidencialidade de uma mensagem de maneira altamente segura [54]. Na seção seguinte a integridade da mensagem é apresentada.

A.2 Integridade

As DTNs possuem a característica da mensagem poder passar por vários outros nós da rede antes de chegar ao destinatário. Devido a esta característica, nós maliciosos na rede podem alterar o conteúdo da mensagem, inserindo informações erradas ou até códigos maliciosos que prejudicarão o destinatário. Portanto, uma das premissas de segurança é a integridade da mensagem que tem por finalidade assegurar que o conteúdo da mensagem é o mesmo enviado pelo remetente. Além disso, a

integridade ajuda a evitar erros, pois como as redes sem fio possuem uma maior possibilidade da mensagem ser corrompida, em comparação com as redes cabeadas, devido a fatores como interferência, colisão e outros; este mecanismo ajuda a não propagação do erro de envio pela rede.

Uma das técnicas para garantir a integridade das mensagens é denominada *Message Authentication Codes* (MACs), onde uma função de uma via (*one way function*) é usada para garantir a integridade da mensagem [21]. Uma função de uma via é definida como uma função matemática que transforma um valor A em outro valor B mas a inversa não pode ser feita, ou seja, a partir de B não se consegue chegar em A. Uma das funções de uma via mais usadas em trabalhos acadêmicos é a função de hash, cujas características são:

- Ser simples de calcular;
- Assegurar que elementos distintos tenham índices distintos;
- Gerar uma distribuição equilibrada para os elementos dentro do conjunto;
- Deve ser aleatória ou pseudo-aleatória para prevenir adivinhações da mensagem original;
- Deve ser única, onde é praticamente impossível duas mensagens diferentes gerarem o mesmo resultado;
- Deve ter mão única, o que significa ser muito difícil a partir do resumo obter a mensagem original.

Podem ocorrer colisões no uso de funções de hash, onde colisões é definido da seguinte maneira. Dado duas mensagens, M1 e M2, se o resultado da aplicação da função de hash em M1 for igual a aplicação da mesma função de hash em M2, diz-se que houve uma colisão. Uma das técnicas para tratar colisões é chamada de re-hashing, onde é usada uma segunda função matemática para calcular os resultados da função de hash. Porém, para integridade não há problema de haver colisões, dado que as funções de hash não serão usadas para o armazenamento das mensagens e sim

para a verificação do conteúdo das mesmas. Portanto, duas mensagens diferentes gerando o mesmo resultado da função de hash não é um problema crítico para integridade, desde que estas colisões não sejam muito freqüentes.

Dentre as funções de hash existentes, foi utilizada neste trabalho a função de hash *Secure Hash* -256 (SHA256) [55], conforme recomendação de segurança descrita em [56]. Ao longo deste texto, as palavras "funções" ou "algoritmo" de hash serão usadas como sinônimas. Uma explicação do funcionamento da função de hash utilizada neste trabalho será apresentada a seguir.

A.2.1 *Secure Hash* 256 - SHA256

O algoritmo de hash SHA-256 faz parte da família de algoritmos de hash chamada de SHA-2. Dentre os membros desta família estão os algoritmos SHA-224, SHA-256, SHA-384 e SHA-512. A diferença entre os membros desta família é o tamanho do resultado do hash (em bits), o algoritmo para calcular o hash é o mesmo. As outras famílias são o SHA-0 e o SHA-1, porém eles são suscetíveis a ataques e não são recomendados para uso [57]. O algoritmo SHA-256 será explicado a seguir.

Antes de iniciar o cálculo do hash, um pré-processamento deve ser feito na mensagem M. Este pré-processamento consiste em três etapas.

- Na primeira é feita a inserção de bits de maneira a mensagem possuir o tamanho correto para o processamento. Esta inserção é feita da seguinte maneira: seja l o tamanho de M, insere-se o bit "1" no final da mensagem seguido de k bits zero, onde k é o menor valor, não negativo, que satisfaça $l+1+k \equiv 448 mod 512$. Por fim, se insere no final 64 bits dados pelo tamanho l da mensagem expressos em bits. Com isso, após esta fase, o tamanho da mensagem inserida de bits será múltiplo de 512 bits;

- A segunda etapa do pré-processamento é a divisão da mensagem gerada na etapa anterior. A mensagem é dividida em N blocos de 512 bits, chamados $M^{(1)}, M^{(2)}, ..., M^{(N)}$. Os 512 bits de cada bloco são divididos em dezesseis

palavras de 32 bits cada, chamadas $M_0^{(i)}, M_1^{(i)}, ..., M_{15}^{(i)}$, onde i é o i-ésimo bloco;

- A última etapa da inicialização é definir o valor inicial do hash, chamado de H^0. Este valor é fixo e é obtido através dos primeiros trinta e dois bits da parte fracional das raízes quadradas dos primeiros oito números primos. Portanto, oito palavras de 32 bits são definidas e apresentadas na Figura A.2.

$$
\begin{aligned}
H_0^{(0)} &= \texttt{6a09e667} \\
H_1^{(0)} &= \texttt{bb67ae85} \\
H_2^{(0)} &= \texttt{3c6ef372} \\
H_3^{(0)} &= \texttt{a54ff53a} \\
H_4^{(0)} &= \texttt{510e527f} \\
H_5^{(0)} &= \texttt{9b05688c} \\
H_6^{(0)} &= \texttt{1f83d9ab} \\
H_7^{(0)} &= \texttt{5be0cd19}.
\end{aligned}
$$

Figura A.2: As oito palavras de 32 bits do valor inicial do hash na computação do SHA-256.

Após esses três passos preliminares, a conversão da mensagem M, de tamanho l, onde $0 \leq l \leq 2^{64}$ em um código hash pode ser iniciada. O processamento ocorre em três etapas também. A primeira etapa efetua um escalonamento das 64 palavras de 32 bits, chamadas $W_0, W_1, ..., W_{63}$. A segunda etapa são calculadas oito variáveis de 32 bits, chamadas a, b, c, d, e, f, g e h. Na última etapa são calculados oito valores de hash de 32 bits cada, representadas por $H_0^{(i)}, H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}, H_5^{(i)}, H_6^{(i)}, H_7^{(i)}$. As operações de soma são sempre computadas em módulo 2^{32}, os valores T_1 e T_2 são variáveis temporárias e as funções $Ch(a,b,c)$, $Maj(a,b,c)$, $\sigma_0^{\{256\}}(x)$, $\sigma_1^{\{256\}}(x)$, $\sum_0^{\{256\}}(x)$ e $\sum_1^{\{256\}}(x)$ significam:

$$Ch(a,b,c) = (a \wedge b) \oplus (\neg a \wedge c)$$

$$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$$

$$\sum_0^{\{256\}}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$$

$$\sum_1^{\{256\}}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$$

$$\sigma_0^{\{256\}}(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$$

$$\sigma_1^{\{256\}}(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$$

$$ROTR^n(x) = (x >> n) \vee (x << w - n)$$

$$SHR^n(x) = x >> n,$$

onde $x >> n$ é o deslocamento à direita de x em n posições, onde os n bits mais a direita são descartados de x e no resultado são inseridos n "zeros" à esquerda e $x << n$ é o deslocamento a esquerda de x em n posições, em que os n bits mais a esquerda são descartados de x e são inseridos n "zeros" à direita do resultado.

Os passos para o cálculo do valor final do HASH é apresentado a seguir em forma de algoritmo.

Para i=1 até N faça {

1. Preparar o escalonamento da mensagem, W_t:

$$W_t = \begin{cases} M_t^{(i)} & 0 \leq t \leq 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \leq t \leq 63 \end{cases} \quad \text{(A.1)}$$

2. Inicializar as oito variáveis de 32 bits a, b, c, d, e, f, g e h, com o i-ésima valor de hash:

$$a = H_0^{(i-1)}$$

$$b = H_1^{(i-1)}$$

$$c = H_2^{(i-1)}$$

$$d = H_3^{(i-1)}$$

$$e = H_4^{(i-1)}$$

$$f = H_5^{(i-1)}$$

$$g = H_6^{(i-1)}$$

$$h = H_7^{(i-1)}$$

3. Para t=0 até 63 faça {

$$T_1 = h + \sum_1^{\{256\}}(e) + Ch(e, f, g) + K_t^{\{256\}} + W_t$$

$$T_2 = \sum_0^{\{256\}}(a) + Maj(a, b, c)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

}

4. Computar o i-ésimo valor intermediário do hash $H^{(i)}$:

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = a + H_1^{(i-1)}$$

$$H_2^{(i)} = a + H_2^{(i-1)}$$

$$H_3^{(i)} = a + H_3^{(i-1)}$$

$$H_4^{(i)} = a + H_4^{(i-1)}$$

$$H_5^{(i)} = a + H_5^{(i-1)}$$

$$H_6^{(i)} = a + H_6^{(i-1)}$$

$$H_7^{(i)} = a + H_7^{(i-1)}$$

}

Após N execuções, o hash da mensagem final é composto da concatenação de $H_0^{(N)}, H_1^{(N)}, H_2^{(N)}, H_3^{(N)}, H_4^{(N)}, H_5^{(N)}, H_6^{(N)}, H_7^{(N)}$, resultando em uma mensagem de 256 bits.

A.3 Autenticidade

O uso de algoritmos de hash e de criptografia do *payload* da mensagem garantem que a mesma estará integra e confidencial, mas não garantem que o emissor da mensagem seja o mesmo que está informado na mesma. É necessário garantir que o emissor da mensagem seja o mesmo que está informado no cabeçalho da mensagem.

Para isso, o nó emissor deve utilizar alguma informação que só ele possua ou somente ele e o nó destinatário possuam, evitando assim que outro nó se passe pelo emissor. Em um esquema criptográfico de chaves simétricas, essa informação pode ser a chave compartilhada pelos nós pois a mesma só é do conhecimento do emissor e do receptor, e geralmente existe uma chave simétrica para cada par de nós ou ainda, uma chave simétrica para cada troca de mensagens. Por exemplo, se o nó emissor criptografasse o hash de uma mensagem com a chave compartilhada, o destinatário só conseguiria verificar a integridade se o hash fosse criptografado com a mesma chave que ele possui. Portanto, como apenas a origem e o destino possuem a mesma chave, se o destino conseguir decriptografar o hash, ele garante que o emissor é realmente quem enviou a mensagem.

Entretanto, em um esquema criptográfico de chave assimétrica, se o nó emissor criptografar o hash com a chave pública do destino, uma atacante pode forjar uma mensagem, calcular o hash e criptografar com a chave publica do destino, pois a chave pública pode ser conhecida por qualquer nó na rede. Com isso a autenticidade do nó emissor não seria garantida.

Contudo, a chave privada do emissor só é de conhecimento dele. Com isso, é possível garantir a autenticidade da mesma. O emissor criptografa o hash da mensagem com sua chave privada e o receptor decriptografa com a chave pública do emissor. Como para cada chave privada só existe uma chave pública, se o destinatário conseguir decriptografar o hash com a chave pública do emissor, é porque o hash foi criptografado com a chave privada, de conhecimento apenas do emissor. Portanto, a mensagem foi realmente gerada pelo emissor, o que garante a autenticidade. Essa técnica de assinar o hash com a chave privada é chamada de assinatura digital. A Figura A.3 apresenta os passos da técnica de assinatura digital.

A autenticidade da mensagem é efetuada no hash do *payload* para inserir menos bits de segurança na mensagem pois, geralmente, o tamanho do hash é menor que o do *payload* e para garantir que o nó destino possa acessar o conteúdo do *payload*, mesmo não possuindo a chave pública do emissor.

Em [58], Auguste Kerckhoffs demonstra que a segurança de um sistema de criptografia depende apenas do segredo da chave e não do segredo do algoritmo, ou seja, se um atacante souber o algoritmo de criptografia usado em um esquema de criptografia, mas não souber a chave, este esquema está seguro.

Como o uso de chave é necessário, tanto para garantir confidencialidade quanto para garantir a autenticidade, conforme apresentado anteriormente, é de suma importância definir um mecanismo de troca de chaves seguro. Como o mecanismo de troca de chaves depende do tipo de rede para o qual ele foi desenvolvido, no Capítulo 3 o mecanismo de troca de chaves proposto para DTNs será apresentado e explicado em detalhes.

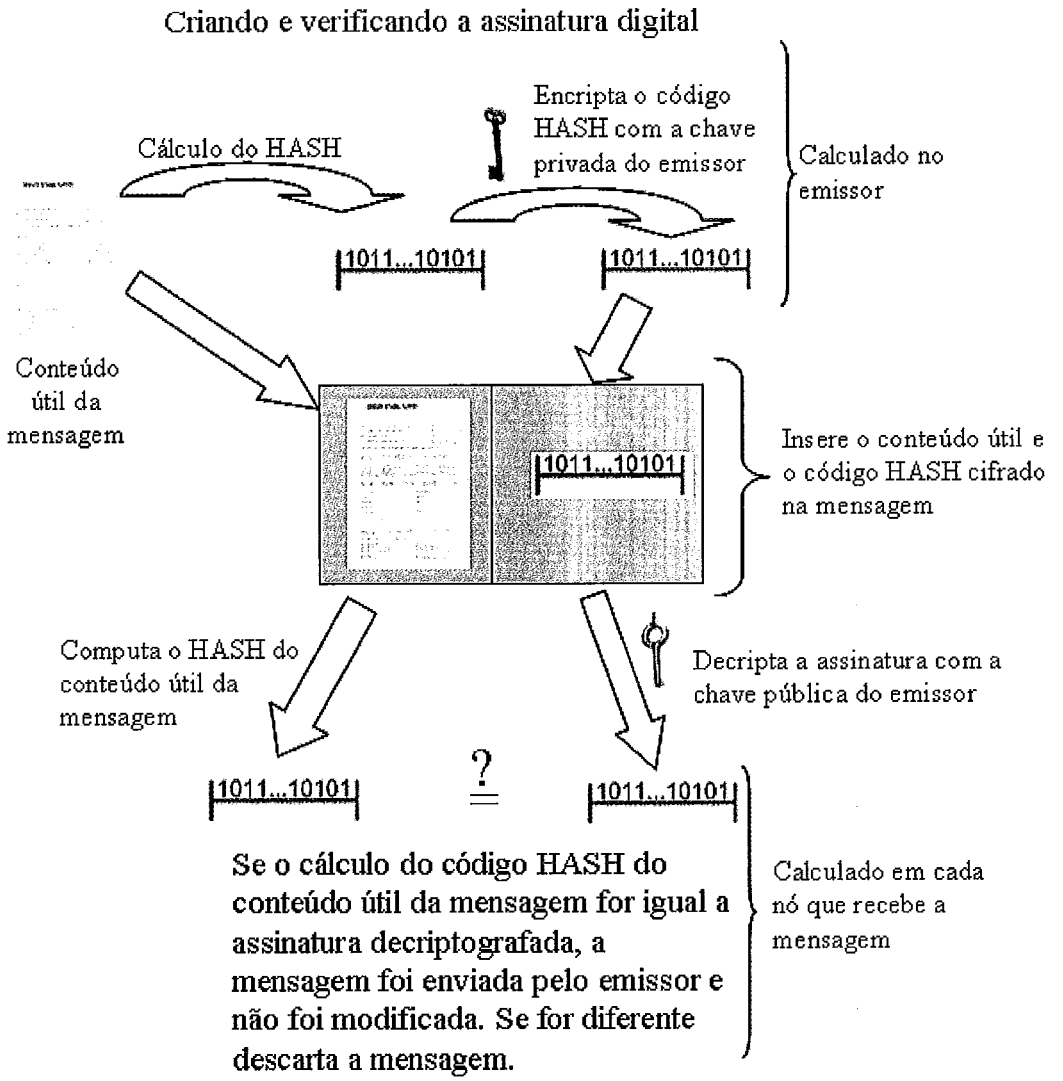

Figura A.3: Funcionamento da técnica de assinatura digital para garantir autenticidade da mensagem.

Apêndice B

O simulador The One

O simulador de DTNs denominado *The Opportunistic Network Environment simulator - The One* [40] simula tempos discretos utilizando divisão do tempo, com isso o avanço do tempo no simulador é feito em tamanhos de tempo fixos. O simulador *The One* foi originalmente desenvolvido com o propósito de analisar a mobilidade dos nós nas DTNs. Para tanto, o mesmo possui módulos de mobilidade para áreas sem obstáculos e com obstáculos, como áreas baseadas em mapas. O simulador foi todo implementado usando a linguagem de programação Java.

O *The One* é divido em módulos, o que facilita as alterações e manutenção do mesmo. Os módulos principais são:

- **Core** - Responsável pelo gerenciamento dos objetos principais do simulador. É este módulo que gerencia as informações dos nós, das mensagens, das conexões e do tempo de simulação.
- **Input** - Responsável pelo gerenciamento da entrada de dados para a simulação. Este módulo que repassa para a simulação, as informações de mobilidade externa, por exemplo os rastros de mobilidade real e a geração de eventos externos, como instantes de geração de mensagens e o par de nós origem e destino das mesmas.
- **Movement** - É o módulo que insere no simulador a mobilidade dos nós.

Neste módulo é que está implementado a mobilidade sintética, como o modelo Random Waypoint, utilizada neste trabalho, e outros modelos de mobilidade como o modelo Random Walk.

- **Report** - Os resultados, que são gerados na simulação, estão implementados neste módulo.
- **Routing** - É o módulo que gerencia as informações de roteamento das mensagens na rede. Além desta gerência, existe ainda a implementação de alguns protocolos de roteamento, como o Epidêmico, PROPHET e *Spray and Wait.*

O simulador possui dois modos de execução, em modo gráfico ou em modo *batch.* No modo gráfico é possível acompanhar a movimentação e a troca de mensagens de cada nó, durante a simulação. Este modo é útil para verificar possíveis anomalias no movimento dos nós, principalmente para implementações novas de mobilidade. Um exemplo deste modo pode ser visto na Figura 5.1. Porém, se não for necessária a visualização da simulação, o modo *batch* é o mais indicado, pois o mesmo utiliza todos recursos da máquina apenas na simulação, com isso a mesma é executada de maneira mais rápida.

É possível informar ao simulador alguns parâmetros de entrada, como o número de simulações a serem executadas, o modo de operação e um arquivo contendo as características da simulação. Neste arquivo pode ser informado o número de nós da simulação; a mobilidade dos nós; o protocolo de roteamento a ser usado; os eventos da rede, como o tempo de geração de mensagens na rede, os tipos de relatórios a serem gerados, entre outras informações. Um exemplo deste arquivo de configuração pode ser visto abaixo.

```
#
# Default settings for the simulation
#

## Scenario settings
```

```
Scenario.name = MMIG_%%Group.router%%_98_600seg_600x800m_1Mbps_range_
%%Group.transmitRange%%_SEME_1
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
# 43k ~= 12h
Scenario.endTime = 600

Scenario.nrofHostGroups = 1

## Group-specific settings:
# groupID : Group's identifier. Used as the prefix of host names
# nrofHosts: number of hosts in the group
# transmitRange: range of the hosts' radio devices (meters)
# transmitSpeed: transmit speed of the radio devices (bytes per second)
# movementModel: movement model of the hosts (valid class name from movement
#package)
# waitTime: minimum and maximum wait times (seconds) after reaching destination
# speed: minimum and maximum speeds (m/s) when moving on a path
# bufferSize: size of the message buffer (bytes)
# router: router used to route messages (valid class name from routing package)
# activeTimes: Time intervals when the nodes in the group are active
#(start1, end1, start2, end2, ...)
# msgTtl : TTL (minutes) of the messages created by this host group,
#default=infinite

## Group and movement model specific settings
# pois: Points Of Interest indexes and probabilities (poiIndex1, poiProb1,
#poiIndex2, poiProb2, ... ) - for ShortestPathMapBasedMovement
# okMaps : which map nodes are OK for the group (map file indexes), default=all
# - for all MapBasedMovent models
# routeFile: route's file path - for MapRouteMovement
```

```
# routeType: route's type - for MapRouteMovement

# common settings for all groups
Group.movementModel = RandomWaypoint
Group.router = [EpidemicRouter;ProphetRouter;SprayandWaitRouter]
Group.bufferSize = 5M
Group.transmitRange = [20]
# transmit speed of 2 Mbps = 250kBps  - Dado em BYTES por seg
Group.transmitSpeed = 125k
Group.waitTime = 0, 0
# walking speeds
Group.speed = 1.52, 1.58
#Group.msgTtl = 60
#Para o ProphetRouter
Group.pois = 1,0.3, 2,0.1, 3,0.1, 4, 0.1

Group.nrofHosts = 100

# group1 (pedestrians) specific settings
Group1.groupID = p

## Message creation parameters
# How many event generators
Events.nrof = 1
# Class of the first event generator
#Poisson
#Events1.class = MessagePoissonEventGenerator
#Uniforme
Events1.class = MessageEventGenerator
# (following settings are specific for the MessageEventGenerator class)
# Creation interval in seconds (one new message every 25 to 35 seconds)
```

```
#Poisson
#Events1.interval = 2.5
#Uniforme
Events1.interval = 1
# Message sizes (200B)
Events1.size = 1000,20000
# range of message source/destination addresses
Events1.hosts = 0,100
# Message ID prefix
Events1.prefix = M


## Movement model settings
# seed for movement models' pseudo random number generator (default = 0)
MovementModel.rngSeed = [1]
# World's size for Movement Models without implicit size
# (width, height; meters)
MovementModel.worldSize = 600, 800
#MovementModel.worldSize = 3000, 3000
#MovementModel.worldSize = 100, 100
# How long time to move hosts in the world before real simulation
MovementModel.warmup = 1000

## Map based movement -movement model specific settings
#MapBasedMovement.nrofMapFiles = 4

#MapBasedMovement.mapFile1 = data/roads.wkt
#MapBasedMovement.mapFile2 = data/main_roads.wkt
#MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
#MapBasedMovement.mapFile4 = data/shops.wkt
```

```
## Points Of Interest -specific settings
#PointsOfInterest.poiFile1 = data/ParkPOIs.wkt
#PointsOfInterest.poiFile2 = data/CentralPOIs.wkt
#PointsOfInterest.poiFile3 = data/WestPOIs.wkt
#PointsOfInterest.poiFile4 = data/shops.wkt

## Reports - all report names have to be valid report classes

# how many reports to load
Report.nrofReports = 22
# length of the warm up period (simulated seconds)
Report.warmup = 0
# default directory of reports (can be overridden per Report with
#output setting)
Report.reportDir = reports/
# Report classes to load
Report.report1 = MessageStatsReport
Report.report2 = InterContactTimesReport
Report.report3 = ContactTimesReport
Report.report4 = DeliveredMessagesReport
Report.report5 = DistanceDelayReport
Report.report6 = MessageDelayReport
Report.report7 = MessageDeliveryReport
Report.report8 = TotalContactTimeReport
Report.report9 = DeliveredMessagesReport2
Report.report10 = MessageDropReport
Report.report11 = AdjacencyGraphvizReport
Report.report12 = ConnectivityDtnsim2Report
Report.report13 = ContactsDuringAnICTReport
Report.report14 = ContactsPerHourReport
Report.report15 = EncountersVSUniqueEncountersReport
```

```
Report.report16 = MessageGraphvizReport
Report.report17 = TotalEncountersReport
Report.report18 = UniqueEncountersReport
Report.report19 = MessageFaltaChave
Report.report20 = MessageKeyStartReceive
Report.report21 = DeliveredMessagesReport3
Report.report22 = DistrFaltaChave

## Default settings for some routers settings
ProphetRouter.secondsInTimeUnit = 30
SprayAndWaitRouter.nrofCopies = 6
SprayAndWaitRouter.binaryMode = true


## Optimization settings -- these affect the speed of the simulation
## see World class for details.
Optimization.connectionAlg = 2
Optimization.cellSizeMult = 5
Optimization.randomizeUpdateOrder = true


## GUI settings

# GUI underlay image settings
GUI.UnderlayImage.fileName = data/helsinki_underlay.png
# Image offset in pixels (x, y)
GUI.UnderlayImage.offset = 64, 20
# Scaling factor for the image
GUI.UnderlayImage.scale = 4.75
# Image rotation (radians)
GUI.UnderlayImage.rotate = -0.015
```

```
# how many events to show in the log panel (default = 30)
GUI.EventLogPanel.nrofEvents = 30
# Regular Expression log filter (see Pattern-class from the
#Java API for RE-matching details)
#GUI.EventLogPanel.REfilter = .*p[1-9]<->p[1-9]$
```

Como todo simulador, este possui algumas limitações. Como dito anteriormente o tempo de simulação é dividido em tempos finitos, onde todos os nós se movem e suas mensagens são repassadas no mesmo tempo. Esse tempo de avanço da simulação é definido no arquivo de configuração. No exemplo acima, o parâmetro Scenario.updateInterval foi definido que o avanço ocorrerá a cada 0,1 segundos. Outra limitação do simulador é a falta de maiores características das camadas Física e de Enlace como, por exemplo, na comunicação entre dois nós a velocidade de transmissão é definida no arquivo de configuração e não leva em conta obstáculos, interferência e distância. Outra restrição é que os rádios dos nós estão sempre ligados, o que não ocorre na realidade, onde os rádios dos nós são colocados em modo de espera, principalmente para poupar energia.

Apesar das limitações descritas acima, que podem ser encontradas em diversos simuladores, o simulador *The One* apresentou um bom funcionamento e de fácil utilização. Como dito na Seção 5.1, alguns trabalhos publicados utilizaram este simulador, como descritos em [41, 42, 43]. Apesar de recente, onde a primeira versão foi lançada em 6 de maio de 2008, este simulador deverá ser uma referência para simulações em DTNs.