UNIVERSIDADE FEDERAL FLUMINENSE

JULIANO FISCHER NAVES

CONTRAMEDIDAS AO ATAQUE DE FALSIFICAÇÃO DE RECONHECIMENTOS POSITIVOS EM REDES TOLERANTES A ATRASOS E DESCONEXÕES

NITERÓI

2017

UNIVERSIDADE FEDERAL FLUMINENSE

JULIANO FISCHER NAVES

CONTRAMEDIDAS AO ATAQUE DE FALSIFICAÇÃO DE RECONHECIMENTOS POSITIVOS EM REDES TOLERANTES A ATRASOS E DESCONEXÕES

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor em Computação. Área de concentração: Sistemas de Computação.

Orientador:
IGOR MONTEIRO MORAES

NITERÓI

2017

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

N323 Naves, Juliano Fischer
Contramedidas ao ataque de falsificação de reconhecimentos positivos em redes tolerantes a atrasos e desconexões / Juliano Fischer Naves. – Niterói, RJ : [s.n.], 2017.
194 f.

Tese (Doutorado em Computação) - Universidade Federal Fluminense, 2017.
Orientador: Igor Monteiro Moraes.

1. Segurança de dados on-line. 2. Redes tolerantes a atrasos e desconexões. 3. Reconhecimento de padrão. I. Título.

CDD 005.8

JULIANO FISCHER NAVES

CONTRAMEDIDAS AO ATAQUE DE FALSIFICAÇÃO DE RECONHECIMENTOS POSITIVOS EM REDES TOLERANTES A ATRASOS E DESCONEXÕES

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor em Computação. Área de concentração: Sistemas de Computação.

Aprovada em setembro de 2017.

BANCA EXAMINADORA

Prof. Igor Monteiro Moraes, D.Sc., UFF (Orientador)

Prof. Célio Vinicius Neves de Albuquerque, Ph.D., UFF

Profª. Débora Christina Muchaluat Saade, D.Sc., UFF

Prof. Miguel Elias Mitre Campista, D.Sc., UFRJ

Prof. Pedro Braconnot Velloso, Dr., UFRJ

Niterói

2017

Dedico este trabalho à memória de minha amada mãe, Delci Lima Fischer.

E tudo o que vocês fizerem ou disserem, façam em nome do Senhor Jesus e por meio dele agradeçam a Deus, o Pai (Cl. 3:17).

Agradecimentos

Agradeço a Deus, pela retomada, pois "O Senhor sustenta a todos os que caem, e levanta a todos os abatidos" (Salmos 145:14). Meus mais profundos e sinceros agradecimentos à minha amada mãe, Delci Lima Fischer, que não mediu esforços para me fazer o que sou e pelo amor incondicional e sem limites. Ao meu pai, Daguioberto Naves, pelo apoio e sacrifícios. Às minhas filhas, Juliane Naves e Olívia Delci Naves, por trazerem sentido novo a minha vida. À minha esposa, Jhaneffer, pelo apoio e paciência. Aos meus irmãos, Diego e Débora, e ao meu sobrinho, Breno Diego, pelos momentos de valor incalculável. Aos meus avôs, Breno, Irena, Humberto e Dagmar. A todas as minhas tias e primos, em especial Leusa, Rose, Vone e Gleice. Aos meus sogros, Cleide e Jaime, pela ajuda e confiança. Ao meu orientador Igor Monteiro Moraes, pela confiança, paciência e auxílio inestimável. À banca examinadora pelo tempo dedicado ao julgamento deste trabalho e pelas importantes contribuições. Aos amigos e professores da UFMT e UFF, em especial Cleyton, Digão, Jivago e Kabeça. Aos amigos e professores do MídiaCom, em especial Célio, Débora, Diego, Edelberto e Joacir. Aos amigos de São José do Rio Claro, MT, G18 em peso. Aos amigos de Ji-Paraná e Vilhena, em especial Lourival, Cleide, Adalberto, Davi, Deilton, Heldo e Régis. Aos antigos professores da Escola Luterana, em especial Selma, Socorro, Yuri e Alberto. À instituição IFRO, pelo apoio. Finalmente, a todos aqueles que de alguma forma apoiaram o transcorrimento deste trabalho, muito obrigado!

Resumo

Os protocolos de roteamento para Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks* - DTNs) são suscetíveis a comportamentos maliciosos dos nós. Particularmente, no ataque de falsificação de reconhecimentos positivos, nós maliciosos forjam reconhecimentos positivos, ou ACKs, com o objetivo de causar impacto negativo no desempenho da rede removendo mensagens que ainda não chegaram ao destino dos *buffers* dos nós. Este trabalho propõe novas contramedidas contra o ataque de falsificação de reconhecimentos positivos em DTNs. As contramedidas são denominadas DRAC (*Drop Acknowledged Messages First*) e DRAC-SF (*Drop Acknowledged Messages First and Stop Forwarding*) e funcionam como se segue. Quando ACKs são recebidos, as propostas não descartam imediatamente as mensagens para as quais estes reconhecimentos foram gerados. Ao invés disso, elas alteram a prioridade destas mensagens na fila de descarte de modo que estas mensagens terão prioridade para serem descartadas em caso de estouro do *buffer*. Adicionalmente, a contramedida DRAC-SF para de encaminhar e replicar mensagens para as quais ACKs foram recebidos. Ressalta-se que DRAC e DRAC-SF não se baseiam em nenhum método de autenticação, visto que não tentam identificar quais são os nós maliciosos. A análise considera quatro cenários reais de mobilidade, sete protocolos de roteamento e dois modelos de ataque distintos. Os resultados mostram que as propostas reduzem a eficiência do ataque de falsificação de reconhecimentos positivos. Adicionalmente, cabe destacar o desempenho da proposta DRAC-SF, que supera o desempenho da principal contramedida existente na literatura em 88% dos cenários avaliados, alcançando taxas de entrega superiores em até 135%.

Palavras-chave: Redes tolerantes a atrasos e desconexões; reconhecimentos positivos; segurança.

Abstract

Routing protocols for Delay and Disruption Tolerant Networks (DTNs) are prone to suffer with malicious behavior of nodes. Particularly, in the acknowledgment counterfeiting attack, malicious nodes forge positive acknowledgments, also known as ACKs, in order to negatively impact network performance by removing from nodes' buffers messages which were not yet delivered to their destinations. This thesis proposes new countermeasures against the acknowledgement counterfeiting attack in DTNs. The proposed countermeasures are named DRAC (DRop ACknowledged messages first) and DRAC-SF (DRop ACknowledged messages first and Stop Forwarding) and work as follows. When an ACK is received, DRAC and DRAC-SF do not immediately drop the message for which this ACK was generated. Instead, in case of buffer overflow drop priority is given to this message. Additionally, DRAC-SF also stops forwarding and replicating messages for which ACKs were received. We emphasize that DRAC and DRAC-SF do not rely in any authentication method because both countermeasures do not try to identify malicious nodes. The analysis considers traces of four real networks, seven routing protocols and two distinct attacker models. Results show our proposals decrease the efficiency of the acknowledgement counterfeiting attack. In addition, we highlight that DRAC-SF outperforms the main countermeasure against this kind of attack found in literature in 88% of the evaluated scenarios, providing higher delivery rates up to 135%.

Keywords: Delay and Disruption Tolerant Networks; positive acknowledgements; security.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

3GPP	:	*3rd Generation Partnership Project*;
BAB	:	*Bundle Authentication Block*;
BIB	:	*Block Integrity Block*;
BEK	:	*Bundle Encryption Key*;
BP	:	*Bundle Protocol*;
BPSec	:	*Bundle Protocol Security*;
BSP	:	*Bundle Security Protocol*;
CA	:	*Certification Authority*;
CL-ACK	:	*Congestion Level based end-to-end ACKnowledgment*;
CRL	:	*Certificate Revocation List*;
DoS	:	*Denial of Service*;
DRAC	:	*DRop ACknowledged Messages First*;
DRAC-SF	:	*DRop ACknowledged Messages First and Stop Forwarding*;
DTN	:	*Delay And Disruption Tolerant Network*;
DTNRG	:	*Delay And Disruption Tolerant Network Research Group*;
EMV	:	*Europay, Mastercard, and Visa*;
ESB	:	*Extension Security Block*;
FIFO	:	*First in First out*;
GAA	:	*Generic Authentication Architecture*;
HMAC	:	*Keyed-Hash Message Authentication Code*;
IBE	:	*Identity Based Encryption*;
IETF	:	*Internet Engineering Task Force*;
IPMEIR	:	*Internet Protocol Security Minimum Essential Interoperability*;
ISS	:	*International Space Station*;
KEK	:	*Key Encryption Key*;
LTP	:	*Licklider Transmission Protocol*;
MAC	:	*Message Authentication Code*;
MANET	:	*Mobile Ad Hoc Networks*;
ONE	:	*Opportunistic Network Environment Simulator*;

PAN : *Primary Account Number*;
PCB : *Payload Confidentiality Block*;
PIB : *Payload Integrity Block*;
PKG : *Private Key Generator*;
PKI : *Public Key Infrastructure*;
PRoPHET : *Probabilistic Routing Protocol using History of Encounters and Transitivity*;
PSN : *Pocket Switched Networks*;
RTT : *Round-Trip Time*;
SBSP : *Streamlined Bundle Security Protocol*;
SnW : *Spray and Wait*;
TFT : *Tit for Tat*;
VDTN : *Vehicular Delay and Disruption Tolerant Networks*;

Sumário

Capítulo 1

Introdução

Na arquitetura TCP/IP, para que haja comunicação entre um par de nós, assume-se que sempre existe um caminho fim-a-fim entre a origem e o destino de uma mensagem. No entanto, tal suposição pode não ser apropriada para modelos existentes de redes sem-fio, que são caracterizados pela grande variação das condições do meio de transmissão e pela mobilidade dos nós. Desconexões frequentes da rede são resultantes destas características, ou seja, um caminho fim-a-fim pode nem sempre estar disponível ou até mesmo não existir entre os nós que desejam se comunicar. Nestes cenários, a utilização da arquitetura TCP/IP é pouco eficiente e, por isso, faz-se necessário o desenvolvimento de uma nova arquitetura, específica para essas redes, que são chamadas de Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks* - DTN) [82]. As redes DTN, são baseadas no paradigma armazena-carrega-e-encaminha. Neste paradigma, os nós da rede são dotados de *buffers* e podem armazenar persistentemente uma mensagem, caso não haja um caminho fim-a-fim entre origem e destino, até que uma oportunidade de encaminhamento apropriada surja. Estas oportunidades de encaminhamento são chamadas de contatos. Em razão da duração limitada dos contatos, cabe ao protocolo de roteamento decidir a ordem de encaminhamento das mensagens, com o objetivo de aumentar o desempenho da rede.

Vários protocolos de roteamento, com diferentes características, foram propostos para operarem em redes DTN [19]. É comum que estes protocolos façam uso de replicação de mensagens durante um contato como uma forma de aumentar a probabilidade de entrega de mensagens. No entanto, tal medida pode esgotar rapidamente os recursos de armazenamento dos nós, levando ao congestionamento da rede [78, 79]. Portanto, mecanismos de controle de congestionamento são empregados para evitar ou reduzir os efeitos negativos causados pelo congestionamento em DTNs. Não por acaso, esforços

recentes foram empregados na análise, caracterização [107, 104] e também na avaliação de mecanismos de controle de congestionamento [106, 103, 77].

No contexto dos mecanismos de controle de congestionamento, surgiu como alternativa a utilização de reconhecimentos positivos [1]. Em DTNs, um reconhecimento positivo $ack(M_n)$ é uma mensagem que é utilizada com o objetivo de sinalizar que uma determinada mensagem M_n chegou ao destinatário. Sendo assim, os nós da rede que ainda estiverem armazenando uma réplica de M_n em seus *buffers* podem remover esta réplica, consequentemente liberando espaço para outras mensagens e diminuindo o congestionamento na rede.

No entanto, os contratempos no projeto e implementação de DTNs não restringem-se ao potencial congestionamento. A segurança é um conceito desafiador nesta arquitetura de rede desde sua idealização até o momento atual [35]. As técnicas criptográficas tradicionais, baseadas em uma infraestrutura de chaves públicas (*Public Key Infrastructure* - PKI), por exemplo, assumem acesso contínuo à rede, tornando-as impraticáveis em DTNs, dada a essência desconectada deste tipo de rede. Em DTNs não é possível assumir acesso contínuo a uma autoridade certificadora (*Certification Authority* - CA) e os métodos tradicionais de revogação de chaves através da atualização *online* de listas de revogação de certificados (*Certificate Revocation List* - CRL) são inapropriados ante as características desconexas e de atrasos indeterminados [57]. A utilização de criptografia simétrica também não é uma opção viável. Isto porque em alguns cenários de comunicação oportunista não é possível estabelecer chaves previamente entre todos os nós da rede, pois a princípio, eles não são sequer conhecidos. Além disso, este sistema não é escalável, pois exige que cada nó possua diferentes chaves para um dos nós da rede. Adicionalmente, a distribuição segura de chaves neste sistema é um outro problema visto que a segurança é quebrada caso um nó malicioso intercepte a troca de chaves entre qualquer par de nós.

A ausência de uma infraestrutura de segurança apropriada torna estas redes suscetíveis a inúmeros ataques. Os nós maliciosos podem utilizar diversas informações a respeito do funcionamento da rede e inclusive tirar proveito dos mecanismos de controle de congestionamento para realizarem ataques. Este é o caso do ataque de falsificação de reconhecimentos positivos, foco deste trabalho. Neste ataque, os nós maliciosos falsificam reconhecimentos positivos com o objetivo de remover completamente da rede mensagens que ainda não foram entregues ao destino, impactando negativamente no desempenho da rede. Portanto, é necessário empregar esforços de pesquisa no desenvolvimento de contra-

[1] Os termos reconhecimentos positivos, ACKs ou acks serão utilizados de modo intercambiável doravante neste trabalho.

medidas eficientes contra este ataque e que funcionem em cenários nos quais a autenticação não pode ser garantida.

1.1 Objetivos

Este trabalho tem como principais objetivos o desenvolvimento e avaliação de contramedidas ao ataque de falsificação de reconhecimentos positivos em Redes Tolerantes a Atrasos e Desconexões. É necessário que estas contramedidas sejam eficientes na redução dos efeitos negativos deste ataque e que além disso, não se baseiem em infraestruturas de autenticação, visto que a implantação de uma infraestrutura de segurança em DTNs pode ser impossível ou inviável em vários cenários, como argumentado anteriormente.

1.2 Contribuições

Entre as contribuições deste trabalho, destaca-se a proposta e avaliação de duas contramedidas ao ataque de falsificação de reconhecimentos positivos em DTNs: DRAC (Drop Acknowledged Messages First) e DRAC-SF (*Drop Acknowledged First and Stop Forwarding*). Estas contramedidas funcionam como se segue. Quando recebem ACKs para uma mensagem, não removem a mensagem do *buffer* do nó, visto que este reconhecimento positivo pode ser falsificado. Em vez disso, prioridade de descarte é dada às mensagens para as quais ACKs foram recebidos. Ademais, a última delas evita continuar replicando mensagens para as quais ACKs foram recebidos. Dentre as propostas, destaca-se o desempenho da contramedida DRAC-SF, que superou o desempenho da principal contramedida existente na literatura em termos de taxa de entrega, em 88% dos 336 cenários avaliados.

Complementarmente à avaliação das contramedidas propostas, duas outras avaliações são realizadas neste trabalho. A primeira avalia os efeitos da utilização de reconhecimentos positivos sob três importantes métricas da rede: taxa de entrega, sobrecarga e atraso de entrega. A segunda avalia os efeitos da realização do ataque de falsificação de reconhecimentos positivos em DTNs. Ambas as avaliações são conduzidas em quatro cenários reais de mobilidade distintos e com sete protocolos de roteamento diferentes. Cabe ressaltar que não foram encontradas na bibliografia avaliações tão abrangentes quanto as desenvolvidas neste trabalho, a respeito da utilização de reconhecimentos positivos e do ataque de falsificação de reconhecimentos positivos em DTNs.

1.3 Organização do Trabalho

O restante deste trabalho está organizado como a seguir. O Capítulo 2 apresenta desafios e propostas de segurança em DTNs. O Capítulo 3 discorre sobre o roteamento e problemas de segurança do roteamento em DTNs. Por sua vez, o Capítulo 4 apresenta uma avaliação da utilização de reconhecimentos positivos em DTNs. Uma avaliação dos efeitos do ataque de falsificação de reconhecimentos positivos em DTNs é feita no Capítulo 5. As propostas deste trabalho, bem como a avaliação do desempenho destas propostas, são apresentadas no Capítulo 6. Finalmente, o Capítulo 7 apresenta as conclusões decorrentes deste trabalho, assim como suas considerações finais.

Capítulo 2

Segurança em Redes Tolerantes a Atrasos e Desconexões

Desde a idealização das Redes Tolerantes a Atrasos e Desconexões até o momento atual, a segurança é um conceito desafiador nesta arquitetura [35].

Mais de uma década de pesquisas levou a uma grande quantidade de propostas de segurança. O esforço empregado pela *Internet Engineering Task Force* (IETF) levou a padronização de uma arquitetura que atua como uma rede de sobreposição, através da especificação do protocolo *Bundle* (*Bundle Protocol* - BP) [97]. Em seguida, a especificação de um protocolo que objetiva prover serviços de integridade de dados e confidencialidade para o protocolo *Bundle*, denominado *Bundle Security Protocol* (BSP), foi publicada [111]. Posteriormente, a especificação de uma versão simplificada do BSP, denominada *Streamlined Bundle Security Protocol* (SBSP) foi proposta [7]. Por sua vez, o protocolo *Bundle Protocol Security* (BPSec) [9] é definido como uma continuação e um refinamento dos protocolos BSP e SBSP. Atualmente, uma nova padronização do protocolo *Bundle* está sendo desenvolvida [14] na forma de um *Internet Draft*. O objetivo desta nova especificação é tornar o protocolo mais simples e fácil de usar, baseando-se na experiência adquirida através da implementação e implantação do protocolo em sua primeira especificação [97].

Adicionalmente, outras propostas de segurança foram feitas pela comunidade científica. A diversidade destas propostas varia desde o aproveitamento da infraestrutura de segurança pré-existente, como a infraestrutura de telefonia celular [60] ou das redes de pagamento de cartão de crédito [22], até a proposta de mecanismos de segurança específicos para determinadas vulnerabilidades [12], como por exemplo, defesas contra ataques específicos ao roteamento, tais quais as que são apresentadas no Capítulo 3. Estas propos-

tas diferem-se com relação às premissas assumidas para o projeto e cenários de avaliação, à complexidade e ao objetivo, apesar de todas visarem aumentar de alguma forma a segurança em redes DTN.

A despeito dos esforços da comunidade científica, problemas de segurança em DTNs persistem, graças a inexistência de um mecanismo de gerenciamento de chaves tolerante a atrasos, assim como a heterogeneidade das características destes tipos de rede. Além disso, os protocolos BSP e SBSP, além de outras propostas da literatura, assumem a existência de uma infraestrutura de chaves públicas na rede, o que pode ser impraticável em alguns cenários atendidos pela arquitetura DTN [116, 118]. Tal impraticabilidade advém de algumas circunstâncias que serão descritas a seguir. Van Besien [118] afirma que o acesso às autoridades de certificação ou de distribuição de chaves dentro de um prazo fixo não pode ser assumida, além disso, a negociação de chaves envolvendo tempos de ida-e-volta (*Round-Trip Time* (RTT)) pode não ser viável. Outro desafio advindo dos períodos indeterminados de interrupção na comunicação é a revogação de credenciais através de *Certificate Revocation Lists*. Este desafio também é abordado por Seth *et al.* [100]. Ainda, de acordo com Templin [112], os mecanismos de segurança propostos para o BP podem ser utilizados somente com chaves previamente distribuídas e irrevogáveis, visto que não existe um mecanismo publicado para o gerenciamento de chaves. No entanto, segundo Uddin *et al.* [116], a distribuição de chaves *a priori* para todos os nós pode ser impossível. Os autores citam um cenário pós-desastre, onde a coordenação global entre diversas entidades, exigida para o estabelecimento de chaves em todos os nós, pode não ser possível devido à natureza imediata da implantação. Portanto, a utilização de PKI em DTNs pode ser considerada inviável.

Não obstante, as características desafiadoras tornam a arquitetura DTN suscetível a um número diverso de ataques ao roteamento. Segundo Choo *et al.* [25], apesar da segurança do roteamento ser extensivamente estudada em redes *ad hoc* tradicionais, estes trabalhos não podem ser facilmente estendidos para DTNs devido às características distintas. Como por exemplo, em redes *ad hoc* tradicionais, geralmente rotas são estabelecidas antes da transferência de dados. Por sua vez, os ataques ao roteamento nestas redes visam prejudicar o estabelecimento de rotas. Em DTNs tipicamente não há o estabelecimento de rotas antes do envio de dados.

Portanto, expostos os desafios confrontados pelas DTNs no que diz respeito ao modelo de segurança tradicional, considera-se importante o estudo das principais propostas de segurança em DTN, além da proposta de novos mecanismos de segurança, com menor

complexidade e/ou que atinjam um melhor desempenho com relação aos mecanismos pré-existentes.

2.1 Desafios de Segurança em DTNs

A segurança é comprovadamente um desafio em DTNs desde a sua concepção [35] e mesmo após anos de pesquisas, este desafio permanece com um número grande de problemas em aberto [36]. Em alguns casos, a recuperação de nós após ataques em DTNs é dita possivelmente mais difícil, principalmente no que diz respeito a nós remotos, como por exemplo em satélites, devido às características dos enlaces, como longos atrasos, assimetria e largura de banda limitada [17].

Farrel *et al.* [39] citam alguns aspectos que diferenciam a segurança em DTNs do modelo de segurança tradicional. Com a ascensão das DTNs, surgiu a necessidade da utilização de serviços de segurança em nós presentes no meio de uma rota e que não são nem fonte nem destino da comunicação. Um exemplo fornecido pelos autores é a de uma rede de sensores sem fio, onde é necessária a proteção da comunicação entre um nó sensor e um *gateway*. Este é o caso do *Bundle Security Protocol*.

De acordo com Farrel *et al.* [39], um outro exemplo do modo como os serviços de segurança de DTNs são diferenciados dos serviços de segurança tradicionais está relacionado com a autenticação das mensagens. Quando uma mensagem é autenticada com uma assinatura digital, normalmente todos os nós no caminho podem ao menos verificar a exatidão da assinatura criptográfica. No entanto, é necessário verificar se o signatário está autorizado a utilizar este serviço. Todavia, devido aos períodos de latência indeterminados e a possível restrição de recursos dos nós, os modos práticos de conseguir este tipo de verificação de autorização são problemáticos em DTNs.

A despeito das dificuldades encontradas para se estabelecer um modelo de segurança apropriado em DTNs, algumas premissas que necessitam de um modelo consistente de segurança são frequentemente adotadas. Como exemplo, o acesso e uso não autorizado dos recursos em DTNs é considerado um sério problema, dada a escassez de recursos que caracteriza várias destas redes. Isto motiva a utilização de um mecanismo de autenticação de origem de uma maneira salto-a-salto, de modo que os nós intermediários possam verificar a validade dos *bundles*[1], com o objetivo de evitar armazenar e encaminhar *bundles* não autorizados [67]. No entanto, isto não leva em consideração que em DTNs o ge-

[1]Neste trabalho, os termos *bundle*, mensagem e pacote serão utilizados de modo intercambiável.

renciamento de chaves é um problema em aberto e que as operações necessárias para se estabelecer segurança de maneira salto-a-salto podem ser impraticáveis para nós com recursos limitados.

A seguir, possíveis ameaças de segurança em DTNs são apresentadas, bem como especificações de segurança existentes para esta arquitetura.

2.2 Ameaças de Segurança

O DTNRG (*DTN Research Group*), um grupo que faz parte do IETF, listou algumas das ameaças que foram consideradas durante o desenvolvimento dos mecanismos de segurança para DTNs, como o *Bundle Security Protocol*. Foram consideradas ameaças de nós que não fazem parte da DTN, consumo de recursos, negação de serviço, confidencialidade e integridade, tempestade de tráfego (*traffic storm*) e a proteção parcial da rede [39].

Com relação a dinâmica das ameaças, Bhutta *et al.* [5] as dividem em ameaças passivas e ameaças ativas. Ameaças passivas incluem um intruso monitorando a difusão de *bundles*. Isto inclui a interceptação de mensagens com o objetivo de obter informações particulares. Ameaças ativas, também chamadas de ataques, abrangem o mascaramento para tentar obter acesso aos recursos da rede sem autorização, modificação de mensagens transmitidas, ataques de reprodução de mensagens e ataques de negação de serviço.

Algumas ameaças e problemas que devem ser levados em consideração durante a concepção de mecanismos de segurança para DTNs são listados a seguir.

2.2.1 Consumo de Recursos

Um aspecto preocupante, dada a natureza das DTNs que entre outras coisas envolve a escassez de recursos, é o acesso não autorizado aos recursos da rede. Entre as ameaças que podem advir da carência de recursos, lista-se o acesso à rede por entidades não autorizadas, o controle da rede por aplicações não autorizadas, aplicações autorizadas enviando *bundles* em taxas não permitidas e modificação não autorizada de conteúdo do *bundle*. Além disso, os nós da rede podem contribuir ajudando ou amplificando o consumo inapropriado de recursos ao encaminhar *bundles* que não foram gerados por nós autorizados ou simplesmente não detectando o comportamento inadequado de outros nós da rede [39].

2.2.2 Negação de Serviço

Em adição as ameaças de consumo de recursos citadas anteriormente, dada a natureza da escassez de recursos, DTNs são suscetíveis a uma ampla variedade de ataques de negação de serviço (*Denial of Service* - DoS). Neste tipo de ameaça os atacantes comportam-se de maneira maliciosa, coletivamente ou individualmente, com o objetivo de prejudicar ou até mesmo interromper o funcionamento da rede. Um ataque DoS pode ser realizado através da injeção de uma quantidade anormal de dados na rede, de forma que a mesma não consiga entregar as mensagens consideradas regulares [75]. Este tipo de ataque pode ser realizado através do descarte de pacotes legítimos pelos nós maliciosos, impedindo que estes pacotes alcancem o destino [80]. Além disso, a latência de comunicação relativamente alta pode contribuir para a efetividade deste tipo de ataque, visto que esta característica pode obrigar os protocolos a manter estados por longos períodos [91]. Ademais, como mencionado por Farrel e Cahil [39], os próprios mecanismos de segurança criam novas oportunidades de ataques de negação de serviço. Como exemplo, um mecanismo que exige a verificação do estado de alguma chave em um servidor de chaves cria oportunidades de ataques tanto contra os nós da rede, quanto contra o servidor, visto que esta verificação consome recursos.

2.2.3 Confidencialidade e Integridade

Aplicações DTNs também podem ser vulneráveis a uma variedade de ataques contra a confidencialidade e a integridade, como por exemplo: a falsificação da origem e destino de um *bundle*, alteração de outros campos do *bundle* e a cópia ou revelação dos dados do *bundle* [39, 111, 7]. Logo, estes são aspectos importantes no projeto de mecanismos de segurança para DTNs.

2.2.4 Tempestade de Tráfego

A especificação do *Bundle Protocol* permite o envio de mensagens que consistem em registros administrativos, como relatórios do estado do *bundle* (*bundle status report*), que são utilizados para indicar o progresso do *bundle* na rede. Isto inclui informações de encaminhamento, recebimento, transferência de custódia [37, 69], entrega final e descarte [97]. Com o objetivo de prevenir tempestades de tráfego, esta especificação proíbe que *bundles* contendo registros administrativos gerem outros *bundles* contendo registros administrativos incluindo a restrição de que as *flags* de relatórios de estados nestes *bundles*

sejam configuradas como zero. No entanto, se um nó da rede DTN ou da rede subjacente conseguir alterar estas *flags*, uma tempestade de tráfego pode ser gerada.

2.2.5 Proteção Parcial da Rede

A carência de poder de processamento ou a proteção da informação por camadas inferiores pode resultar na condição em que nem todos os nós desejam proteger todas as partes de todos os *bundles*. Além das ocasiões mencionadas anteriormente, um nó pode desejar não proteger um *bundle* com o objetivo de permitir a fragmentação [42].

2.2.6 Ameaças à Privacidade

Quando um dispositivo sem fio é associado a um usuário, rastrear este dispositivo revela a localização e o padrão de movimentação desta pessoa [67]. Este é o caso particular das *Pocket Switched Networks* (PSNs), um cenário mais específico de DTNs onde os dispositivos que formam a rede são sempre carregados pelos usuários, como *smartphones* e *notebooks* [49], alguns tipos específicos de redes de sensores sem fio (*Wireless Sensor Networks* - *WSNs*) e também redes veiculares tolerantes a atrasos e desconexões (*Vehicular Delay and Disruption Tolerant Networks* - VDTNs) [88].

2.2.7 Ameaças Físicas

Enquanto alguns dispositivos são pessoais e carregados pelos usuários, outros são autônomos e podem ser colocados em localizações remotas [67]. O posicionamento remoto, longe de acompanhamento, torna estes dispositivos suscetíveis a ataques físicos, seja vandalismo ou até mesmo, como observado por Farrell [38], o desaparecimento ou furto de dispositivos.

2.2.8 Ameaças de Nós Que Não Fazem Parte da DTN

Visto que a arquitetura DTN define a rede como uma rede de sobreposição [20, 97], *bundles* geralmente atravessam vários elementos da rede subjacente a cada salto na rede DTN. Desta forma, quaisquer vulnerabilidades dos protocolos da arquitetura DTN poderão ser exploradas por estes elementos. Além disso, as vulnerabilidades dos protocolos subjacentes podem ser exploradas com o objetivo de impactar negativamente na rede DTN.

2.3 Problemas em Aberto

Apesar do extenso interesse da comunidade científica na segurança de redes DTN, ainda existem problemas que permanecem em aberto e que ainda demandam muita pesquisa. A seguir, alguns destes problemas são apresentados.

2.3.1 Gerenciamento de Chaves

Segundo Caini *et al.* [17], o maior problema em aberto em DTNs é a segurança, em particular, a ausência de um mecanismo de gerenciamento de chaves que seja tolerante a atrasos. Geralmente, os esquemas existentes exigem serviços de verificação de estado ou de distribuição de chaves que sejam conectados com os demais nós da rede, o que não é viável em ambientes com grandes latências ou altamente desconectados. Apesar dos esforços empregados pela comunidade científica no desenvolvimento de especificações suplementares ao *Bundle Protocol* [97], através da especificação do *Bundle Security Protocol* [111] e do *Streamlined Bundle Security Protocol*, além das extensões de segurança propostas para o *Licklider Transmission Protocol* (LTP), estas especificações deixam em aberto o problema do gerenciamento de chaves em DTNs. Este é reconhecido por vários autores como um problema em aberto em DTNs [119, 38, 67, 17, 112].

Templin e Burleigh [112] listam vários requisitos desejáveis para um sistema de gerenciamento de chaves que seja eficiente para DTNs. A seguir, estes requisitos são brevemente descritos:

- chaves devem ser fornecidas quando necessário, sem depender de respostas rápidas;
- o sistema deve ser confiável, baseado em uma entidade da qual a confiança é assumida ao invés de derivada, conhecida como âncora de segurança;
- o sistema não deve conter um ponto único de falha, sendo resistente a uma ou mais falhas;
- o sistema deve conter múltiplos pontos de autoridade e não deve obrigar o usuário a confiar em informações de uma única autoridade sem a corroboração de outras autoridades;
- o sistema não deve permitir que uma única autoridade comporte-se inadequadamente, recusando-se a corroborar com informações providas por outras autoridades, degradando a rede;

- o sistema deve vincular a chave pública com a identidade do nó DTN de maneira confiável;
- o sistema deve suportar a inicialização segura da identidade de um nó e sua chave pública correspondente;
- o sistema deve suportar revogação de chaves;
- a revogação de chaves deve ser tolerante a atrasos.

Além dos requisitos anteriormente apresentados, Caini *et al.* citam que é desejável que protocolos de gerenciamento de chaves possam suportar heterogeneidade, para lidar com nós com diferentes capacidades. Observa-se que a despeito do que já foi desenvolvido em seguranças em DTNs, o desenvolvimento de um sistema de gerenciamento de chaves é um desafio que exigirá o emprego de esforços da comunidade científica, dada a complexidade do objetivo de atingir um sistema tolerante a atrasos.

2.3.2 Reprodução de Mensagens

A reprodução de mensagens também é uma ameaça em DTNs [5]. Um nó malicioso pode reproduzir várias cópias de uma mesma mensagem na rede, utilizando inadvertidamente os recursos, como largura de banda e espaço em *buffer*. No entanto, mesmo com o risco de consumir os escassos recursos da rede, existem ocasiões onde a reprodução de mensagens (*message replays*) é desejada [52]. Como exemplo, suponha que o nó B deseje atuar como uma mula de dados em favor do nó A, carregando o *bundle* X até o nó C. Suponha que após receber o *bundle*, o espaço em *buffer* do nó B se esgote. B então descarta o *bundle* X. No entanto, em posterior contato B entra em contato com A e recebe novamente o *bundle* X. Após isso, B locomove-se até o alcance de C e envia X para C. Fica claro que, neste caso, a reprodução (*replay*) do *bundle* X foi essencial para que este alcançasse o destino. Adicionalmente, a maioria dos protocolos de roteamento propostos para operarem em DTNs utiliza múltiplas cópias [18].

Com o objetivo de evitar a reprodução de mensagens, os nós podem manter uma lista de *bundles* recentemente vistos. No entanto, dada a possível alta latência da rede, pode ser necessário manter esta lista por vários dias. Isto pode ser dispendioso, dadas as restrições de recursos dos nós da rede. Claramente, maiores esforços devem ser empregados para a resolução deste problema.

2.3.3 Comportamento Egoísta

Os protocolos de roteamento desenvolvidos para operarem em DTNs geralmente assumem a premissa de que cada nó individual está disposto a encaminhar os *bundles* para os demais. No entanto, a existência de nós egoístas é plausível e viola esta premissa. Com base na arquitetura de referência [20], os nós DTN podem decidir de modo autônomo se devem ou não aceitar a custódia de *bundles*. Consequentemente, a cooperação não é plenamente garantida e não deve ser assumida. Estes nós denominados egoístas tendem a não agirem como retransmissores de *bundles* com o objetivo de não desperdiçarem seus recursos próprios.

Segundo Zhu, é comum que os esquemas de incentivo existentes, utilizados em redes *ad hoc* tradicionais, assumam que exista um caminho fim-a-fim entre a fonte e o destino da comunicação. Além disso, de maneira geral, os esquemas existentes foram projetados para a utilização em sistemas cuja comunicação assume a existência de uma única cópia de cada mensagem. Ambas estas premissas podem não vigorar em DTNs. Isto torna o desenvolvimento de mecanismos de incentivo que impeçam o comportamento egoísta um desafio de pesquisa [124].

Segundo Khabbaz *et al.* [59], a pesquisa de técnicas cooperativas, explorada na comunicação par-a-par e nas redes *ad hoc*, é voltada para a investigação de três problemas principais:

1. A averiguação de como a cooperação afeta o desempenho da rede.
2. Detecção de nós não-cooperativos.
3. Projeto de protocolos que impõem a cooperação dos nós.

Em DTNs, deve-se levar em consideração a heterogeneidade dos nós com relação aos recursos. Desta forma, um nó pode não ser egoísta, mas ainda assim ser levado a não cooperar devido a restrição de limitação de recursos, como espaço em *buffer* e energia.

Sermpezis e Spyropoulos [99] avaliam os efeitos do egoísmo social em comunicações oportunistas. Segundo os autores, o egoísmo se dá de três maneiras distintas de acordo com o comportamento do usuário. São elas:

1. Um nó não encaminhará tráfego algum (egoísmo individual - *individual selfishness*);
2. Um nó escolherá encaminhar pacotes com alguma probabilidade p (egoísmo uniforme - *uniform selfishness*);

3. Um nó encaminhará pacotes preferencialmente para outros nós com o quais ele mantém algum relacionamento social (egoísmo social - *social selfishness*).

Atentando-se aos fatos de que o comportamento egoísta é mutável e possui diferentes vertentes em adição aos inúmeros e heterogêneos cenários que compreendem a arquitetura DTN, a detecção eficiente de nós cooperativos e a imposição de cooperação para os nós participantes da rede é um desafio em DTNs.

2.3.4 Problema da Fragmentação

Embora a utilização de fragmentação seja adequada para aumentar a taxa de entrega de mensagens [90], na ocorrência de fragmentação de *bundles*, esquemas tradicionais de autenticação, onde o emissor gera uma assinatura para a mensagem inteira, podem não funcionar adequadamente. Isto se deve ao fato de que na fragmentação reativa, o tamanho do fragmento não é conhecido antecipadamente. Adicionar assinaturas em cada fragmento pode ser dispendioso do ponto de vista que mais recursos computacionais e de transmissão serão necessários. Logo, a segurança dos *bundles* na presença de fragmentação é um desafio.

2.3.5 Problemas de Desempenho

A criptografia de chave pública pode resultar em uma significativa sobrecarga de desempenho e isto pode inviabilizar a utilização da mesma. A utilização de chaves, resulta em uma sobrecarga de transmissão e armazenamento. Além disso, a verificação de chaves públicas geralmente requer o uso de operações de computação intensivas. Esta sobrecarga pode ser agravada através da utilização de protocolos de múltiplas cópias, visto que a transmissão, armazenamento e verificação das chaves será feita para cada cópia [124].

Ginzboorg [46] argumenta que o gerenciamento de recursos está estritamente relacionado com a segurança em DTNs. Desta forma, o gerenciamento de recursos pode influenciar na segurança e vice-versa. Segundo ele, a segurança tem impacto no gerenciamento de recursos toda vez que recursos escassos são alocados para mitigar ataques potenciais no sistema. Por outro lado, o acesso a recursos escassos deve ser controlado, o que implica a autorização dos dispositivos que acessam estes recursos. Desta forma Ginzboorg argumenta que existe uma compensação entre segurança e gerenciamento de recursos: deve haver recursos suficientes para suportar mecanismos de segurança, ao mesmo tempo em que mecanismos de segurança são introduzidos para proteger os recursos escassos. A

conclusão é de que devido a este estreito relacionamento, esquemas de gerenciamento de recursos não devem ser introduzidos sem examinar seu impacto na segurança da rede ao mesmo tempo que recursos de segurança não devem ser fornecidos sem primeiro considerar os recursos que eles consomem.

2.4 Propostas de Segurança

2.4.1 Protocolo *Bundle*

Com o objetivo de auxiliar na compreensão do protocolo *Bundle*, esta seção será dedicada a fornecer uma breve introdução sobre o funcionamento deste protocolo.

O protocolo *Bundle* [97] foi proposto para prover os serviços necessários em uma arquitetura tolerante a atrasos e desconexões [20]. Para fornecer estes serviços, o protocolo *Bundle* forma uma camada de sobreposição sobre um número de inter-redes que constituem a DTN, assim como ilustrado na Figura 2.1. Esta camada de sobreposição é chamada de camada *bundle*.

O projeto do protocolo *Bundle* é destinado a atender a necessidade de uma variedade de redes existentes, tais como redes espaciais, redes subaquáticas, redes de sensores, redes *ad hoc* e tantas outras chamadas de redes desafiadoras [119, 35]. Na Figura 2.1, o protocolo *Bundle* está sendo utilizado para interconectar duas inter-redes distintas, através da utilização de 3 diferentes pilhas de protocolos. A interface entre a pilha de protocolos da inter-rede e o protocolo *Bundle* é chamada de adaptador de camada de convergência (*convergence layer adapter*). Na Figura 2.1, existem três camadas de convergência: T1, T2 e T3.

A unidade de dados do protocolo *Bundle* é chamada de *bundle*. Múltiplas instâncias do mesmo *bundle* podem existir concorrentemente em várias partes da rede. Um *bundle* pode conter uma ou mais unidades de dados geradas pela aplicação. Uma transferência de arquivos, por exemplo, pode conter informações de autenticação e informação de localização do arquivo, além da informação sobre a operação em si [20].

Dentre os principais recursos deste protocolo, lista-se:

- retransmissão baseada em transferência de custódia;
- habilidade de lidar com conectividade intermitente;
- habilidade de tomar vantagem de contatos agendados, previstos ou oportunistas.

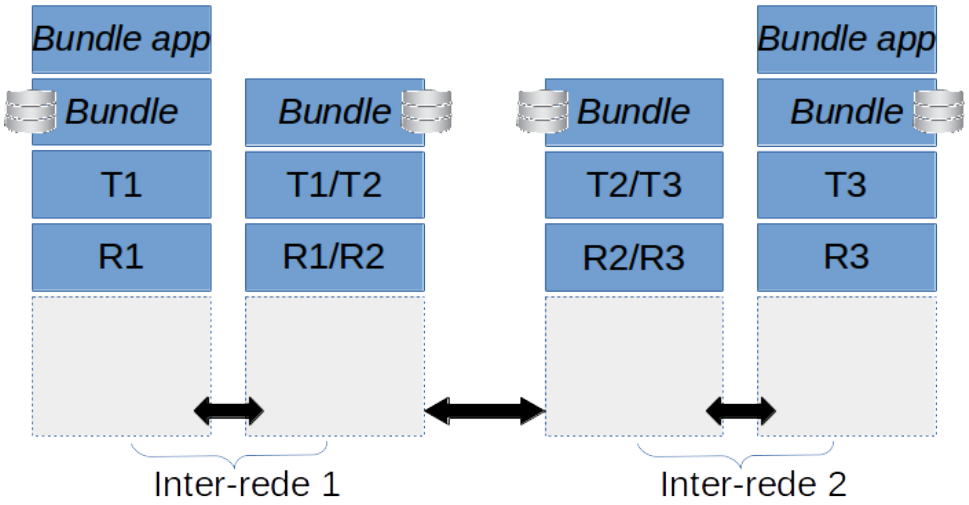


Figura 2.1: O *bundle* protocol situa-se na camada de aplicação do modelo TCP/IP

Quando um nó aceita a custódia de um *bundle* ele se compromete a manter uma cópia deste *bundle*, possivelmente encaminhando cópias a outros nós, até que a custódia deste *bundle* seja liberada. Entre as características para suportar a conectividade intermitente estão a possibilidade de fragmentação proativa e reativa de um *bundle*, além da existência de *buffers* nos nós. Na fragmentação proativa, o nó fragmenta antecipadamente um *bundle* em partes menores, para que o mesmo possa ser enviado em contatos de curta duração, que não seriam suficientes para enviar o *bundle* inteiro. Na fragmentação reativa, os nós têm a capacidade de identificar que a conexão foi interrompida durante a transmissão de um *bundle*, armazenando a porção que conseguiu efetivamente ser transmitida de forma a aproveitar melhor as conexões. Com relação aos *buffers*, eles possibilitam que os nós armazenem os *bundles* por tempo indeterminado, até que uma oportunidade de encaminhamento apropriada surja. Esta característica é frequentemente chamada de paradigma armazena-carrega-e-encaminha (*store-carry-and-forward*).

Com relação ao formato do *bundle*, cada um deve ser formado pela concatenação de dois ou mais blocos. Blocos contêm informações tipicamente encontradas no cabeçalho ou carga útil de unidades de dados de outros protocolos. Devido à possibilidade de um bloco não aparecer no início de um *bundle*, o termo bloco é utilizado em preterimento do termo cabeçalho. O primeiro bloco da sequência deve ser um bloco primário, que contém informações básicas para rotear o *bundle* até o destino. Um *bundle* nunca poderá ter mais de um bloco primário. Somente um dos blocos seguintes pode conter a carga útil dos dados. Além destes dois blocos, cada *bundle* pode conter outros blocos, denominados blocos de extensão, com o objetivo de suportar extensões ao protocolo *Bundle*.

2.4.2 *Bundle Security Protocol*

O *Bundle Security Protocol* [111] define recursos de segurança para o *Bundle Protocol* [97]. Nós que implementam o BSP são chamados de nós *cientes de segurança*. Podem existir outros nós na DTN que não implementam o BSP.

O BSP diferencia a fonte de um *bundle* da *fonte de segurança* de um *bundle*. Esta última pode ser definida como o nó que aplica os recursos de segurança ao *bundle*. De maneira análoga, também são diferenciados o destino e o *destino de segurança* de um *bundle*, sendo este último, o nó que decifra ou verifica a integridade do *bundle*.

De modo a prover recursos de segurança para o *Bundle Protocol*, quatro tipos de blocos de segurança que podem ser incluídos em um *bundle* são definidos. São eles, *Bundle Authentication Block* (BAB), *Payload Integrity Block* (PIB), *Payload Confidentiality Block* (PCB) e *Extension Security Block* (ESB). Para cada bloco, o BSP define um único conjunto de cifras (*cipher suite*) obrigatório. Uma visão geral sobre estes blocos é apresentada a seguir.

2.4.2.1 *Bundle Authentication Block*

O BAB é utilizado para garantir a autenticidade e integridade do *bundle* ao longo de um único salto. Dada a possibilidade da existência de nós que não implementam o BSP e que, portanto, não são cientes de segurança, um único salto de um nó ciente de segurança para o próximo nó ciente de segurança pode na realidade ser mais de um salto.

Visto que o BAB visa proteger o *bundle* de uma forma "salto-a-salto", toda vez que algum outro bloco estiver presente, o BAB deve formar uma camada de proteção exterior. Ou seja, ele deve ser calculado e adicionado após o cálculo dos outros blocos de segurança.

Para atingir os objetivos, ou seja, garantir a autenticidade e integridade do *bundle*, um BAB pode utilizar tanto um código de autenticação de mensagens (*Message Authentication Code* - MAC) computado através da utilização de uma chave simétrica, quanto uma assinatura digital, computada com o uso de chaves assimétricas. A *ciphersuite* obrigatória definida pela especificação para o BAB utiliza o algoritmo HMAC (*Keyed-Hash Message Authentication Code*), que por sua vez utiliza uma chave secreta compartilhada.

2.4.2.2 *Payload Integrity Block*

O PIB é utilizado para garantir a integridade e autenticidade da carga útil. Ao contrário do BAB, que visa garantir a integridade e a autenticidade salto-a-salto, o PIB é utilizado para garantir a integridade da fonte de segurança, o nó que criou o PIB, até o destino de segurança do PIB, o nó que processará o PIB. Segundo a definição, se permitido pelo conjunto de cifras (*cipher suite*), a informação de autenticação no PIB pode ser verificada por todos os nós entre a fonte de segurança e o destino de segurança do PIB, desde que tenham acesso as chaves criptográficas e a informação de revogação de chaves.

Assim como no caso dos blocos BAB, um bloco PIB pode utilizar tanto um código MAC quanto uma assinatura digital para atingir seus objetivos de integridade e autenticidade. A *ciphersuite* obrigatória definida pela especificação para o PIB utiliza o algoritmo de criptografia RSA em conjunto com o algoritmo de funções de dispersão SHA-256.

2.4.2.3 *Payload Confidentiality Block*

A adição de um PCB indica que a carga útil do *bundle* foi encriptada integralmente ou parcialmente, na fonte segura do PCB, com o objetivo de garantir a confidencialidade da carga útil até que o *bundle* seja entregue ao destino seguro do PCB. Segundo a definição, uma *ciphersuite* típica para o PCB utiliza uma chave de encriptação de *bundle* (*Bundle Encryption Key* - BEK) gerada aleatoriamente que é encriptada utilizando-se uma chave de criptografia de chave (*Key Encryption Key* - KEK) de longo prazo e carregada no cabeçalho do bloco ou uma chave pública. Além disso, os autores da especificação sugerem que um mecanismo de integridade deve ser utilizado em conjunto com a confidencialidade e que *ciphersuites* que fazem somente a encriptação dos dados não devem ser utilizadas.

2.4.2.4 *Extension Security Block*

Caso seja necessária a proteção de porções do *bundle* que não são relacionadas a carga útil, o ESB deve ser utilizado. Este bloco não deve ser utilizado com porções relacionadas a carga útil. Logo, não é adequado para proteger PIBs, PCBs, a própria carga útil ou bloco primário do *bundle*. Uma instância da utilização de ESBs é para a proteção de metadados do *bundle* [111, 110]. Segundo a definição, ESBs são tipicamente utilizados para aplicar confidencialidade, mas podem ser utilizados para aplicar proteção da integridade. Segundo a definição, uma *ciphersuite* típica para um bloco ESB possui

os mesmos pré-requisitos da *ciphersuite* descrita anteriormente para os PCBs.

2.4.3 *Streamlined Bundle Security Protocol*

Outro protocolo que define recursos de segurança para o protocolo *Bundle* é o *Streamlined Bundle Security Protocol* [7]. O objetivo da proposta deste protocolo é separar a função e roteamento das funções de segurança. Isto se deve ao fato de que no protocolo BSP uma origem de segurança de um *bundle* pode definir um destino de segurança que não é necessariamente o destino da mensagem. Além disso, um *bundle* pode ter múltiplos blocos de segurança e o destino de segurança de um bloco de segurança deste *bundle* pode ser diferente do destino de segurança de um outro bloco de segurança. Logo, para que um *bundle* chegue ao destino, ele deve passar por todos os destinos de segurança necessários, o que resulta no acoplamento das funções de segurança com o roteamento [6, 8].

Similar ao BSP, o SBSP define algumas terminologias, cuja compreensão é crucial para a interpretação do protocolo. Por definição, o SBSP é aplicado somente aos nós que o implementam e estes nós são chamados de *cientes de segurança* (*security-aware*). O SBSP define os *serviços de segurança* (*security-services*) como os recursos suportados pela especificação, são eles: autenticação, integridade e confidencialidade. Similar ao BSP, o SBSP define uma *fonte de segurança* (*security-source*) como um nó que adiciona um bloco de segurança ao *bundle*. De maneira análoga, o *destino de segurança* (*security-destination*) é o nó que valida a informação de segurança de um bloco. Desta forma, quando um serviço de segurança é aplicado salto-a-salto, o destino de segurança é o próximo recebedor do *bundle*. Caso contrário, o destino de segurança é o mesmo que o destinatário do *bundle*. Esta definição previne que o roteamento da mensagem seja impactado pela escolha do destino de segurança, como pode acontecer no BSP. Além disso, o SBSP define um *alvo de segurança* (*security-target*) como a porção de um *bundle* que receberá o serviço de segurança e uma *operação de segurança security-operation* como a aplicação de um serviço de segurança a um alvo de segurança específico.

Neste protocolo, três blocos de segurança que podem ser incluídos em um *bundle* são definidos. Estes três blocos definidos pelo SBSP são descritos a seguir.

2.4.3.1 *Bundle Authentication Block*

O *Bundle Authentication Block* (BAB) é utilizado para garantir a autenticidade e a integridade de um *bundle* ao longo de um salto. Desta forma, os BABs operam entre nós

topologicamente adjacentes. Nós cientes de segurança podem escolher solicitar BABs de um determinado vizinho na rede. A autenticação e a integridade de um *bundle* através de um BAB é feita através da utilização e um código MAC ou de uma assinatura digital.

2.4.3.2 *Block Integrity Block*

O *Block Integrity Block* (BIB) é utilizado para garantir a autenticidade e integridade de seus alvos-seguros, a partir da fonte-de-segurança do BIB, a qual cria o BIB, para o destino do *bundle*, o qual verifica o autenticador do BIB. Quando possível, a informação de autenticação do BIB pode ser verificada por quaisquer nós entre a fonte-de-segurança do BIB e o destino do *bundle*. Assim como no BAB, a autenticação de um bloco utilizando um BIB é garantida através da utilização de um código MAC ou de uma assinatura digital.

2.4.3.3 *Block Confidentiality Block*

O *Block Confidentiality Block* (BCB) indica que o alvo-de-segurança foi encriptado, integralmente ou parcialmente, na fonte-de-segurança do BCB com o objetivo de proteger seu conteúdo enquanto em trânsito até o destino do *bundle*. É interessante ressaltar a semelhança entre o BCB e o PCB definido pelo BSP: ambos visam garantir a confidencialidade de parte do *bundle*. No entanto, enquanto o PCB é aplicado somente a carga útil do *bundle*, o BCB é aplicado a blocos. Este pode ser aplicado a carga útil, bem como ao BIB e qualquer outro bloco não pertencente ao SBSP.

2.4.4 Segurança do protocolo LTP

Com o objetivo de suprir a necessidade de tolerância a longos atrasos em uma camada de rede inferior à camada *bundle*, o DTNRG desenvolveu o *Licklider Transmission Protocol* [15, 91]. O LTP aborda tolerância a atrasos e desconexões em um ambiente ponto-a-ponto, com ênfase na operação em um único enlace. Segundo Farrel *et al.* [40], dado que o LTP é um protocolo ponto-a-ponto, não existem considerações a fazer sobre congestionamento e roteamento. Além disso, segundo os autores, o LTP é projetado para ser uma potencial camada de convergência para o BP.

Extensões de segurança que compreendem um serviço de autenticação e a utilização de um mecanismo de *cookies* foram propostas para o LTP [41]. Ambas as extensões são descritas a seguir.

A extensão de autenticação do LTP deve estar presente no primeiro segmento de

cada sessão LTP que utiliza autenticação. Para autenticação, é utilizado ou um código MAC ou uma assinatura digital. Além disso, uma terceira alternativa que também utiliza um MAC, mas com uma chave pré-distribuída incorporada (*hardcoded*) com o objetivo de fornecer um código de verificação de erros. É interessante observar que a última alternativa fornece uma forma de verificar a integridade da mensagem sem autenticação e, portanto, está sujeita a ataques ativos.

Com relação ao mecanismo de *cookies*, este é proposto para dificultar a realização de ataques de negação de serviço. Os autores sugerem que esse mecanismo deve ser utilizado em ambientes onde uma implementação do LTP é suscetível a este tipo de ataque [41]. Um *cookie* pode ser visto como um número aleatório de tamanho satisfatório. Este valor pode ser incrementado a qualquer momento através da concatenação de mais bits aleatórios. O LTP pode incluir um *cookie* em um segmento a qualquer momento. Após isso, todos os segmentos correspondentes a mesma sessão LTP devem conter um valor válido de *cookie*, isto é, um valor que começa com o valor atualmente armazenado. Uma vez que um valor de *cookie* é visto e tratado como válido, o valor anterior passa a ser inválido.

Em face da retransmissão de um segmento, é necessário que o mesmo contenha um valor correto de *cookie*, que pode ser diferente do valor do *cookie* no momento da transmissão. Ainda no contexto da utilização de *cookies*, um segmento com um valor inválido de *cookie* ou que deveria conter um *cookie* mas não contém, deve ser silenciosamente descartado.

2.4.5 Criptografia Baseada na Identidade

A criptografia baseada em identidade (*Identity Based Encryption* - IBE) é um método de criptografia onde a chave pública de um usuário é alguma informação pública única a respeito do usuário, como endereço de e-mail ou número de telefone. Apesar de proposto por Shamir [101] em 1984, este problema foi recentemente solucionado através da proposta de um sistema de criptografia baseada em identidade inteiramente funcional [26, 11].

Um sistema de criptografia baseado em identidade é composto pelos *principals*, isto é, as origens e destinos de mensagens, além de uma entidade confiável, denominada gerador de chave privada (*Private Key Generator* - PKG). Durante a inicialização do sistema, o PKG gera parâmetros públicos denominados PP. Além disso, uma chave secreta mestre é criada, denominada S_{PKG}. Para gerar uma chave privada para um *principal* P, o PKG utiliza S_{PKG} e id_P. Para encriptar uma mensagem para P, um nó Q utiliza id_P e PP.

Asokan *et al.* [3], chegam a conclusão que apesar de fornecer melhores meios para alcançar confidencialidade fim-a-fim, a utilização de sistemas IBC em DTNs não tem vantagem significativa para autenticação e integridade comparados a criptografia tradicional (i.e., criptografia assimétrica). Além disso, Ginzboorg [46] argumenta que não é claro se os benefícios são suficientes para preferir IBC em DTNs, visto que a criptografia tradicional possui mais suporte no que diz respeito às implementações.

2.4.6 Aproveitamento de Infraestrutura Pré-existente

Uma alternativa ao estabelecimento de associações de segurança a partir do zero é o aproveitamento de associações de segurança previamente estabelecidas. Estas associações podem ser aquelas estabelecidas com redes de telefonia celular, cartões EMV (Europay, Mastercard, and Visa), entre outras. A principal vantagem desta abordagem é que ao explorar a utilização desta infraestrutura pré-existente, evita-se a necessidade de implantação, possivelmente cara, de uma nova infraestrutura de segurança.

2.4.6.1 Inicialização Através da Rede de Telefonia Celular

A *3rd Generation Partnership Project* (3GPP) é a organização que padroniza redes de telefonia celular. Entre suas publicações está o desenvolvimento de uma arquitetura que possibilita que operadoras de celular estendam a autenticação celular, oferecendo-a como um serviço. A esse padrão foi atribuído o nome de *Generic Authentication Architecture* (GAA) [60].

A inicialização da segurança em DTNs através do aproveitamento das associações seguras de redes de telefonia celular é interessante, especialmente no caso das PSNs, onde a rede é formada por dispositivos móveis portáteis. Desta forma, de acordo com Asokan *et al.* [2], através da utilização do padrão GAA pode-se implantar servidores CA e PKG, mantidos ou pela operadora de telefonia celular ou por terceiros que tenham contratado este serviço da operadora.

Por último, segundo Asokan *et al.* [3], embora a maioria das pessoas tenha telefones celulares, nem todos os clientes de DTNs possuem o cartão SIM necessário para o estabelecimento da segurança. De acordo com os autores, este problema pode ser resolvido através da utilização de uma conexão sem fio de curto alcance, tal como o *Bluetooth*. Desta forma, é proposto que quando um dispositivo que não possua acesso a rede de telefonia celular precise estabelecer uma associação de segurança, o mesmo pode enviar

uma solicitação ao telefone móvel, que procede com a autenticação e envia a chave obtida para o dispositivo que solicitou a autenticação.

2.4.6.2 Inicialização Através de Cartões EMV

Europay, Mastercard, and Visa (EMV) é um padrão global para autenticação de transações de débito e crédito que envolvam cartões compatíveis com *chips* e os terminais presentes nos pontos de venda.

Chen *et al.* [22] propõem a utilização da infraestrutura de segurança dos cartões EMV através da arquitetura GAA. A esta arquitetura de segurança é dado o nome de EMV-GAA. Em suma, o funcionamento da arquitetura EMV-GAA é similar ao funcionamento da arquitetura descrita na Seção 2.4.6.1. No entanto, a inicialização da segurança é feita aproveitando-se a relação de segurança estabelecida entre o portador do cartão EMV e o banco emissor deste cartão.

Duas ameaças de segurança são apontadas pelos autores. A revelação ou descoberta do número do cartão (*Primary Account Number* - PAN) e a realização de transações não autorizadas pelo leitor do cartão. É sugerido que estas ameaças podem ser mitigadas através da utilização de um leitor especial, desenvolvido exclusivamente para ser utilizado com EMV-GAA.

2.4.7 Segurança de Fragmentos

A fragmentação dos *bundles* foi introduzida na arquitetura DTN com o objetivo de aproveitar da melhor maneira possível os contatos, evitando a retransmissão por completo de *bundles* parcialmente encaminhados [20]. A fragmentação pode acontecer de duas maneiras distintas. Na fragmentação proativa, nós DTN podem dividir um *bundle* em tamanhos menores e transmiti-los como *bundles* independentes. A fragmentação reativa acontece quando um *bundle* é parcialmente recebido pelo nó, como no caso, por exemplo, da interrupção prematura do contato. Neste caso, o camada *bundle* que recebeu o fragmento é responsável por modificá-lo, de modo a indicar que se trata de um fragmento.

Neste contexto, proteger o *bundle* com o *hash* e a assinatura da origem não é uma boa abordagem em caso de fragmentação. Isto se deve ao fato de que o recebedor da mensagem não consegue autenticar o *bundle* enquanto não recebê-lo por completo [3].

Partridge propôs algumas soluções para o problema da autenticação de fragmen-

tos [87]. Na primeira solução, denominada autenticação cumulativa (*cumulative authentication*), cada fragmento f_i é autenticado através da aplicação da função de autenticação em todos os fragmentos anteriores até o fragmento atual. Esta alternativa tem um alto custo computacional e de sobrecarga, visto que uma assinatura precisa ser gerada para cada fragmento.

Uma segunda alternativa é através da definição de funções (*function definitions*). A ideia é que ao invés de proteger os dados com um *hash* de uma função de autenticação, eles são protegidos por uma função gerada dinamicamente que retorna um resultado conhecido. Considerando uma função de autenticação A, dados os fragmentos $f_1, f_2, ...f_m$ de um *bundle* e um vetor de inicialização iv para este *bundle*, de tal forma que os *hashes* obtidos serão:

$$A(iv, f_1) = 1, A(iv, f_2) = 2, ..., A(iv, f_m) = m$$

A ideia é encontrar um iv para cada *bundle* que seja relativamente pequeno e produza menos sobrecarga que m *hashes* assinados. No entanto, encontrar uma função de autenticação eficiente para este esquema ainda é um problema em aberto.

Asokan *et al.* [3] e Zhu *et al.* [126] propõem a utilização de uma árvore *hash* binária para a autenticação de fragmentos. Neste esquema, o emissor seguro prepara n fragmentos $f_i...f_n$ para envio e calcula os *hashes* de cada fragmento. Após isso, ele calcula um próximo conjunto de *hashes* combinando os *hashes* anteriores em pares: $h(h_i, h_{i+1})i = 1, 3, 5...n$. O *hash* final é assinado pelo emissor seguro do *bundle*. Como observado na Figura 2.2, extraída de Asokan *et al.* [3], para autenticar o fragmento f_3, um nó precisa dos *hashes* h_4, h_{1-2}, h_{5-8}, além da assinatura do *hash* final.

Solis *et al.* [108] propõem a utilização de um esquema de autenticação de fragmentos por melhor-esforço. Desta forma, este esquema assume que em alguns casos é admissível uma taxa relativamente alta de falsos positivos, enquanto que falsos negativos não são aceitáveis. Os autores enfatizam que a proposta não visa substituir os esquemas de integridade fim-a-fim tradicionais e que deve ser utilizada somente em nós intermediários. Uma possibilidade de implementação deste esquema abordada pelos autores é a utilização de filtros de Bloom.

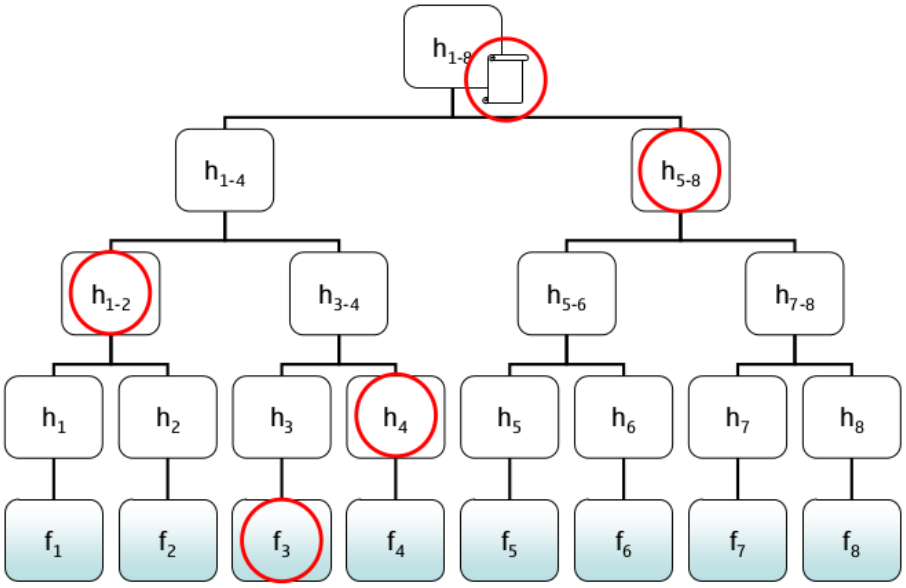


Figura 2.2: Árvore Hash Binária - [3]

2.5 Considerações Adicionais

Ao finalizar este capítulo, fica claro que a segurança em DTNs é uma área rica, tanto em diversidade de pesquisas, quanto de problemas em aberto.

Requisitos de segurança, ameaças, mecanismos e políticas, são geralmente conceitos específicos de aplicação. Farrel [38] demonstrou que a implantação de redes experimentais, mesmo em ambientes aparentemente não ameaçadores, exigiu a consideração de aspectos de segurança e privacidade com antecedência. Além disso, diferentes aplicações podem culminar no surgimento de diferentes ataques, previamente não considerados.

Com relação aos mecanismos de segurança, como observado na Seção 2.3.5, exige-se que a troca inerente entre a adição de mecanismos de segurança e o consumo de recursos seja considerada antecipadamente, para cada aplicação.

Informações recentes sinalizam que a NASA e a Innoflight Inc. estão testando o protocolo SBSP na plataforma de testes SCaN pertencente à NASA. Esta é uma plataforma avançada de comunicações instalada na Estação Espacial Internacional (*International Space Station* - ISS) [43]. No entanto, neste teste é utilizado o IPMEIR (*Internet Protocol Security Minimum Essential Interoperability*) para proteção de enlaces. Isto pode ser um indicativo de que o SBSP por si só não é uma solução completamente autônoma. Esta conjectura é corroborada com as propostas de mecanismos de segurança para o protocolo LTP e a especificação de uma versão simplificada do protocolo BSP, o SBSP. Ou seja,

os mecanismos de segurança propostos para o protocolo *Bundle* não são completamente suficientes para garantir a segurança necessária em todos os cenários.

Uma característica é comum a todas as propostas de segurança apresentadas neste capítulo: elas utilizam algum tipo de infraestrutura de chaves. No entanto, como previamente discutido, isto pode ser inviável em alguns cenários de DTNs, o que tornar necessário o desenvolvimento de propostas de segurança que não baseiem-se nestas infraestruturas tradicionais de criptografia.

Finalmente, foi apresentada neste capítulo uma variedade de problemas e propostas de segurança para DTNs, entretanto, nenhuma delas pode ser considerada um consenso. Mais que isso, considera-se que algumas vezes as propostas são complementares entre si. Neste contexto, o desenvolvimento de propostas de segurança para aspectos particulares de DTNs ganha força. Este é o caso de propostas de segurança que visam tornar a rede robusta ou reduzir os efeitos de ataques contra o roteamento. Considerando-se a importância do roteamento como função essencial de uma DTN, no Capítulo 3, ataques contra o roteamento serão revisados e propostas que visam assegurar a segurança em DTNs na iminência destes ataques serão apresentadas.

Capítulo 3

Segurança do Roteamento em DTNs

Apesar dos esforços empregados pela comunidade científica no desenvolvimento de especificações suplementares ao BP [97], através da especificação do BSP [111] e do SBSP [7], estas especificações deixam em aberto o problema do gerenciamento de chaves em DTNs. Além disso, algumas redes podem compreender nós com características extremamente desafiadoras em termos de processamento de forma que a utilização de uma infraestrutura de chave pública compreendendo os processos de encriptar/descriptografar, assinatura e verificação de *bundles* tornam-se dispendiosos ou, de certa forma, impossíveis [36, 85, 118]. Adicionalmente, devido à conectividade intermitente, atrasos inerentes que podem variar de poucos segundos a vários dias e ausência de um caminho fim-a-fim, propostas de distribuição de chaves para redes *ad hoc* móveis (*Mobile Ad Hoc Networks* - MANETs) não são apropriadas para DTNs [84, 12].

A carência de uma solução de segurança apropriada para DTNs favorece um número de ataques que podem ser realizados contra o roteamento, objetivando diminuir o desempenho da rede ou interromper seu funcionamento. Mesmo que seja considerada a existência de um modelo de segurança apropriado para DTNs e que a rede só seja acessada por dispositivos autorizados, alguns ataques ao roteamento ainda podem ser realizados. Algumas destas possibilidades são descritas a seguir. Nós autorizados podem ser infectados com algum tipo de *software* malicioso. Pessoas não autorizadas podem obter controle de nós autorizados através do furto ou "sequestro" destes nós. Além disso, através da apreensão temporária de um nó, um indivíduo poderia acessar e roubar as credenciais deste nó. Sendo assim, propostas pontuais que visem aumentar a robustez da rede na ocorrência de ataques ao roteamento são necessárias.

A seguir, alguns protocolos de roteamento em DTNs são apresentados. Posterior-

mente, alguns ataques que podem ser realizados ao roteamento, conhecidos e debatidos na literatura, são revisados. Além disso, algumas das contramedidas propostas e avaliadas para lidar com estes ataques também são apresentadas.

3.1 Roteamento em DTNs

O roteamento em DTNs é um grande desafio, principalmente quando o comportamento da rede é aleatório ou desconhecido [53, 54, 123, 55, 18]. Neste contexto, diversas métricas e protocolos de roteamento foram propostos para estas redes. Estes protocolos utilizam entre outras métricas, o histórico de contatos [65, 30, 13] e métricas sociais [51, 27, 28, 10]. Quanto à quantidade de cópias, esta pode ser limitada como no trabalho de Spyropoulos *et al.* [109], controlada como no trabalho de Lindgren *et al.* [65] ou não controlada como no trabalho de Vahdat e Becker [117].

A seguir, são descritos os sete protocolos utilizados nas avaliações desta tese, a saber: Epidêmico, *Life*, MaxProp, Prophet, ProphetV2, *Spray and Wait* e *Wave*. As avaliações deste trabalho utilizam exatamente estes sete protocolos, a menos que seja estabelecido de maneira diferente na seção que precede cada avaliação.

3.1.1 Protocolo Epidêmico

O roteamento epidêmico [117] é em essência um protocolo baseado na inundação de mensagens e uma das primeiras propostas de protocolos para DTNs. Nesse protocolo, dois nós trocam o maior número de mensagens que eles não têm em comum toda vez em que se encontram. O número de mensagens trocadas é dado pelo tempo de contato entre os nós. Desse modo, as mensagens são espalhadas entre os nós, o que aumenta a probabilidade dessas mensagens chegarem ao nó de destino. Com tamanho de *buffer* e capacidade de transmissão limitada, o protocolo epidêmico pode exaurir os recursos da rede, o que acarreta descarte de mensagens e retransmissões. Isso resulta em baixa eficiência da rede, em termos da taxa de entrega e sobrecarga de mensagens.

A Figura 3.1 ilustra um exemplo do roteamento epidêmico de mensagens. Nela, os círculos da cor cinza representam nós que possuem uma réplica da mensagem, assim como os círculos na cor branca ilustram nós que não possuem uma réplica da mensagem em questão. As setas que partem de um nó representam a movimentação desse nó. Observa-se que a mensagem é gerada no nó F no instante 07:00 e entregue ao nó D no instante 10:00 através da replicação feita pelo nó F e pelos nós intermediários.

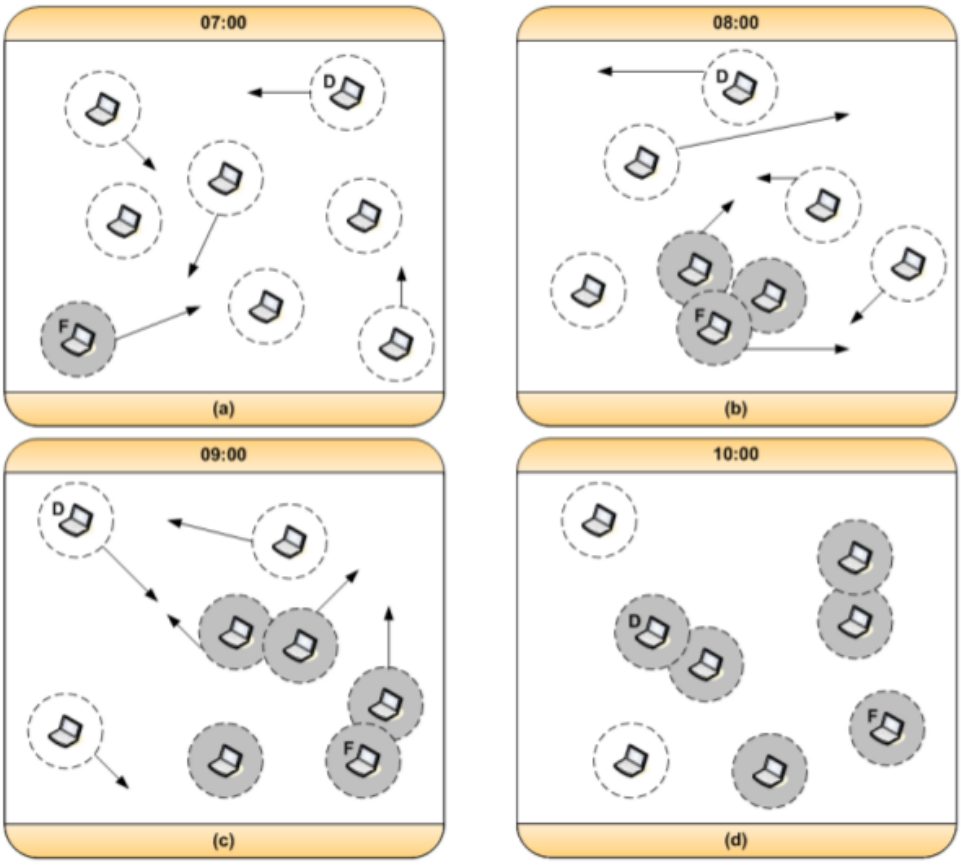


Figura 3.1: Exemplo de roteamento epidêmico [82].

3.1.2 Protocolo PRoPHET

O protocolo PRoPHET [65] (*Probabilistic Routing Protocol using History of Encounters and Transitivity*) baseia-se no histórico dos contatos para calcular a probabilidade de um novo encontro entre dois nós. Para tanto, uma métrica chamada previsibilidade de entrega (*delivery predictability*), $P_{(a,b)} \in [0,1]$, é estabelecida em cada nó, em cada nó a, para cada nó destino conhecido b. Essa métrica indica quão provável um nó poderá entregar uma mensagem para o destino. Assim, quando dois nós se encontram, eles trocam um vetor de sumário, que contém as mensagens que cada nó possui, junto com a métrica de previsibilidade de entrega para cada destino. Então, essa informação é utilizada para decidir quais mensagens requisitar para o outro nó.

O cálculo da previsibilidade no protocolo PRoPHET é composto por três etapas. A primeira etapa consiste em atualizar a métrica toda vez que um nó é encontrado. Esse cálculo é mostrado na Equação 3.1, onde $P_{init} \in [0,1]$ é uma constante de inicialização.

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{init} \tag{3.1}$$

Se dois nós ficam um grande período de tempo sem se encontrar, então, é menos provável que eles entreguem mensagens um para o outro. Logo, a métrica de previsibilidade de entrega deve ter algum tipo de envelhecimento. Para isso, a Equação 3.2, denominada Equação de envelhecimento, é aplicada. Nesta Equação, $\gamma \in [0, 1)$ é uma constante de envelhecimento e k é a quantidade de unidades de tempo que passou desde a última vez que a Equação de envelhecimento foi aplicada a métrica.

$$P_{(a,b)} = P_{(a,b)old} \times \gamma^k \tag{3.2}$$

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \tag{3.3}$$

A métrica de previsibilidade de entrega tem também uma propriedade de transitividade. A propriedade de transitividade baseia-se no fato que se um determinado nó A encontra frequentemente um nó B, que também encontra frequentemente um outro nó C, então é provável que o nó A seja uma boa alternativa para encaminhar mensagens destinadas ao nó C. Na Figura 3.1, ainda é possível observar que o nó A pode entregar uma mensagem transitivamente para o nó D através dos nós B e D. A Equação 3.3 mostra como a transitividade afeta a métrica de previsibilidade de entrega. Nela, a constante $\beta \in [0, 1]$ determina quão grande deve ser o impacto da transitividade na métrica de previsibilidade de entrega.

3.1.2.1 Protocolo ProphetV2

O protocolo ProphetV2 [47] é definido pelos autores como a evolução do protocolo Prophet. Os autores identificaram a necessidade de modificações através da validação do protocolo Prophet por meio de simulações e implantações, como o projeto *Network for Communication Challenged Communities* (N4C) [71]. Entre os problemas descobertos no protocolo Prophet, é citada a superestimação do número de encontros entre nós devido a conexões sem-fio instáveis. Além disso, observou-se que um número de estudos mostrou que o melhor desempenho do protocolo Prophet aconteceu quando o parâmetro β foi configurado para 0, efetivamente desativando a propriedade de transitividade.

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{enc} \tag{3.4}$$

P_{enc} é calculado como na Equação 3.5. I_{typ} é um parâmetro configurado como o tempo

estimado entre as conexões no cenário em questão. $Intvl_b$ é o tempo transcorrido desde o último contato com o nó b.

$$P_{enc} = \begin{cases} P_{max} \times \dfrac{Intvl_b}{I_{typ}}, & \text{se } 0 \leq Intvl_b \leq I_{typ} \\ P_{max} & \text{caso contrário} \end{cases} \tag{3.5}$$

O problema encontrado pelos autores com a equação de transitividade original é que contanto que $\beta > 0$, o valor calculado para a métrica previsibilidade de entrega aumentará para todo nó n, independente se algum nó da rede tenha encontrado n recentemente ou não. Para solucionar este problema, os autores propõem uma nova equação para o cálculo da métrica de previsibilidade de entrega, transcrita para a Equação 3.6. Esta equação não possui a propriedade aditiva, como a anterior. Na realidade, ela compara o valor da métrica de previsibilidade de entrega antiga com o valor do produto de $P_{(b,i)}$, $P_{(a,b)}$ e β, selecionando o maior valor para ser o valor da previsibilidade de entrega.

$$P_{(a,i)} = max(P_{(a,i)_{old}}, P_{(b,i)} \times P_{(a,b)} \times \beta) \tag{3.6}$$

3.1.3 Protocolo MaxProp

O protocolo MaxProp [13] requer que cada nó da rede mantenha um vetor de probabilidades de entrega. Cada entrada deste vetor armazena o custo que representa a probabilidade de um contato futuro entre o nó local e seus vizinhos. A soma dos valores de probabilidade para todos os vizinhos é normalizada para ser sempre 1. Inicialmente, visto que não existem informações disponíveis, todas as entradas são configuradas para $1/(n-1)$, onde n é o número de nós da rede. Sempre que um contato entre dois nós A e B acontece, o nó A procura a entrada referente ao nó B em seu vetor de probabilidade de entrega e adiciona 1 ao seu valor. Logo após, o nó atualiza todas as entradas, dividindo-as por 2 com o objetivo de normalizar os valores para que voltem a somar 1. Analogamente, o nó B realiza o mesmo procedimento. Como resultado deste procedimento, os nós que são encontrados com mais frequência recebem maiores valores de probabilidade. É importante ressaltar que sempre que um contato acontece, os nós trocam seus vetores de probabilidade de entrega. Desta forma, um nó A conhece o vetor de probabilidade de entrega no momento do contato para todos os nós que ele encontrou.

De posse das probabilidades de entrega, o protocolo MaxProp utiliza-as para ordenar a ordem de prioridade de cada pacote armazenado nos *buffers* dos nós. Esta ordenação é

feita como se segue. Cada nó cria um grafo dirigido representando todos os nós da rede. Para cada par de nós A e B, à aresta $A \rightarrow B$ é atribuído um peso igual ao complemento do valor de probabilidade para B no vetor de probabilidade de entrega de A. Baseado neste grafo, o nó local seleciona para cada outro nó na rede o caminho com menor custo, definido como a soma dos pesos de suas arestas. A todos os pacotes destinados ao mesmo nó é atribuída a mesma prioridade, que é igual ao custo do caminho selecionado. Sendo assim, sempre que um contato acontece, pacotes com maior prioridade são transmitidos primeiro. De modo análogo, nós com menor prioridade são os primeiros a serem descartados em caso de estouro do *buffer*.

Na realidade, a transmissão de mensagens no protocolo MaxProp segue a seguinte ordem:

1. Primeiramente, as mensagens destinadas ao nó vizinho são enviadas.
2. Posteriormente, os nós trocam os vetores de probabilidade de entrega.
3. Reconhecimentos de mensagens entregues são trocados pelos nós.
4. Mensagens cujo número de saltos é menor que um valor t são ordenadas por ordem de número de saltos e são enviadas. É importante ressaltar que t é dinamicamente calculado, tendo como base o número de bytes x transferido a cada contato.
5. Os pacotes restantes que ainda não foram transmitidos são transmitidos de acordo com a prioridade descrita anteriormente.

3.1.4 Protocolo *Spray and Wait*

O protocolo de roteamento *Spray and Wait* [109] é um protocolo de múltiplas cópias, no entanto, impõe um limite estrito de replicação e de encaminhamentos através da utilização de um parâmetro L. Essencialmente, o protocolo *Spray and Wait* pode ser divido em duas fases como a seguir:

1. Fase de **espalhamento**: para toda mensagem originada em um nó fonte, L cópias são disseminadas na rede.
2. Fase de **espera**: finalizada a fase de espalhamento, se o nó destino não foi encontrado, todos os nós carregando uma das L cópias da mensagem para atuar como transmissão direta, isto é, encaminharão a mensagem somente para o nó destinatário.

Com relação ao modo como as mensagens são espalhadas na rede, os autores definem basicamente duas estratégias. Na primeira delas, o nó fonte encaminha todas as L cópias disponíveis para os L primeiros nós que encontra. No entanto, os autores definem uma nova forma, que chamam de modo binário e que argumentam ser melhor do que a estratégia anterior. No modo binário, o nó fonte inicia com L tíquetes de uma mensagem. Qualquer nó A que tenha $n > 1$ tíquetes de uma mensagem e encontre um outro nó B, que não possui uma cópia da mensagem em questão, pode replicar esta mensagem para o nó B. Neste caso A envia a réplica com $\lfloor n/2 \rfloor$ tíquetes da mensagem para o nó B e mantém $\lceil n/2 \rceil$ tíquetes para si. Consequentemente, ambos os nós A e B podem replicar a mensagem na rede. Este processo se repete até que a quantidade de tíquetes seja igual a 1, quando a mensagem poderá ser entregue somente para o nó destinatário.

3.1.5 Protocolo *Life*

Ott *et al.* [83] propõem um protocolo de roteamento inspirado no "Jogo da Vida", um autômato celular desenvolvido pelo matemático John Conway [45]. A ideia básica do "Jogo da Vida" é iniciar com uma configuração simples de contadores ou "organismos", cada um em uma célula, e observar como este conjunto se modifica de acordo com algumas regras estabelecidas por Conway. As regras foram escolhidas com o objetivo de satisfazer 3 aspirações:

1. não deve existir um padrão inicial para o qual exista uma prova simples que a população possa crescer sem limites;
2. devem existir padrões onde ***aparentemente*** a população crescerá sem limites;
3. devem existir padrões que cresçam e mudem por um período considerável de tempo antes de chegar em um de três possíveis resultados:
   - Desaparecer completamente, seja por uma superpopulação ou por uma população muito esparsa;
   - Chegar a um estado estável, que permanece inalterado depois disso;
   - Entrar em uma fase de oscilação, na qual um ciclo sem fim é repetido.

A regras de modificação dos contadores são listadas a seguir:

- **Sobreviventes**: cada contador com 2 ou 3 vizinhos sobrevive para a próxima geração;

- **Mortes**: cada contador com quatro ou mais vizinhos morre de "superpopulação". Cada contador com um ou nenhum vizinho morre de "isolamento".
- **Nascimentos**: cada célula vazia com exatamente 3 vizinhos é uma célula de nascimento. Um contador é posicionado nesta célula no próximo movimento.

No protocolo de Ott *et al.*, cada contador é um nó da rede. Um nó é um contador em estado de "sobrevivente" se possui uma cópia da mensagem M_k em *buffer*, caso contrário, este nó não é um contador ativo. A cada movimento, um nó X verifica quantos de seus vizinhos possuem uma réplica da mensagem M_k e então aplica uma série de regras de modificação similar às regras do "Jogo da Vida". Estas regras definem se a mensagem M_k será replicada para X (nascimento), se será mantida em X (sobrevivência) ou se será descartada por X (morte).

3.1.6 Protocolo *Wave*

Outro protocolo proposto por Ott *et al.* [83] é o protocolo *Wave*. Neste protocolo, cada nó mantém uma lista das mensagens que recebeu recentemente. As mensagens são mantidas nesta lista por um certo período de tempo, que os autores chamam de "tempo de imunidade". Como regra, um nó não aceita uma mensagem se ela está contida nesta lista. Além disso, ao receber uma mensagem, um nó aceita tomar a custódia desta mensagem. Isto significa que este nó é impedido de descartar esta mensagem até encaminhá-la para outro nó, junto com a custódia desta mensagem.

Segundo os autores, neste protocolo a decisão de roteamento é uma decisão local que garante que as mensagens visitarão áreas que não foram recentemente visitadas. É interessante ressaltar que tanto o protocolo *Wave* quanto o protocolo *Life*, descrito na Seção 3.1.5, são definidos para lidar com um cenário específico de comunicação, denominado pelos autores de *BeachNet*. Segundo os autores, estes cenários são caracterizados por longos períodos estáticos, com pouca mobilidade, população de nós variável e uma vasta população de nós, esparsas e densas.

3.2 Ataque do Buraco Negro

No ataque do buraco negro (*blackhole attack*), um atacante cria um *buraco negro de roteamento*, forjando informações de roteamento. Neste ataque, o atacante alega estar no caminho mais curto entre a origem e o destino das mensagens. Desta forma, o atacante

atrai as mensagens para si e após isso, as descarta. Um tipo especial de ataque do buraco negro é o ataque do buraco cinza (*grayhole attack*), onde os pacotes são seletivamente descartados, isto é, alguns são descartados, outros não são [94]. Além disso, um ataque simples que um nó malicioso pode realizar contra a rede é simplesmente descartar os pacotes que ele recebe [12]. Embora este ataque possa não ser propriamente dito um ataque de buraco negro, visto que não forja informações da tabela de roteamento, considera-se que possui similaridade o suficiente para que seja tratado nesta seção.

De acordo com Li *et al.* [62], ao forjar métricas e distribuí-las aos demais nós da rede com os quais um nó malicioso faz contato, com o objetivo de atrair um maior número de pacotes para ele, este nó malicioso, pode descartar estes pacotes ou utilizá-los para iniciar outros ataques mais sofisticados. Como exemplo é mencionado que após atrair os pacotes um nó poderia disseminar reconhecimentos positivos falsificados, realizando um ataque de falsificação de reconhecimentos positivos.

Segundo Chen *et al.* [23], o ataque do buraco negro pode ser bastante nocivo ao desempenho de redes oportunistas. De acordo com Lindgren e Hui [66], os protocolos de roteamento devem garantir que não seja fácil a criação de ataques de buraco negro na rede.

As contramedidas da literatura lidam de diferentes maneiras com o ataque do buraco negro em DTNs. Existem contramedidas que baseiam-se no histórico de encontros [80, 62, 120, 125, 32], na utilização de um nó especial responsável por monitorar a rede [95], na utilização do histórico de entrega de mensagens [93, 63, 125], na utilização de mecanismos de reputação [33, 72, 125, 24], na utilização do histórico de mensagens encaminhadas [48] e na utilização de mecanismos de incentivo [70].

3.3 Ataques de Inundação

No ataque de inundação (*Flooding Attack*), um atacante maliciosamente ou egoisticamente, inunda a rede com pacotes, visando interromper o funcionamento normal da rede ou, no caso egoísta, aumentar a probabilidade de sua informação chegar ao destino, enviando mais réplicas do que o permitido. Burgess *et al.* [12] demonstram que protocolos de replicação são resistentes a ataques de inundação. No entanto, não é claro o modelo exato de ataque que foi utilizado, fazendo-se necessárias novas avaliações utilizando outros protocolos e diferentes cenários.

Devido às características das DTNs, com limitações de recursos como espaço de ar-

mazenamento em *buffer* e largura de banda, a inundação de pacotes pode rapidamente exaurir os recursos limitados, levando a uma redução do desempenho. Nagrath *et al.* [73] afirmam que, dada a influência do tamanho do *buffer* no desempenho da rede, este recurso deve ser protegido contra ataques de inundação. Silva *et al.* [105] citam que o controle de congestionamento em DTNs é desafiador principalmente devido a duas razões: (1) conectividade intermitente sem garantia de caminho fim-a-fim e (2) latência de comunicação pode ser arbitrariamente longa. Dadas estas características desafiadoras do controle de congestionamento e levando-se em consideração que o ataque de inundação tem como objetivo congestionar a rede, é prudente evitar ou mitigar os efeitos dos ataques de inundação de alguma maneira.

Li *et al.* [64] categorizam os ataques de inundação em dois diferentes tipos: inundação de pacotes e inundação de réplicas. No ataque de inundação de pacotes os atacantes injetam tantos pacotes quanto possível na rede, gerando mais pacotes do que o permitido. O segundo tipo, isto é, o ataque de inundação de réplicas, distingui-se do primeiro na medida que o atacante encaminha réplicas de um pacote já existente para tantos nós quanto possível, replicando mais do que o permitido por quaisquer taxas limites.

Uma outra classificação é feita por Lee *et al.* [61]. Nela, o ataque de inundação é dividido em 4 categorias: (1) inundação aleatória, (2) inundação através da seleção de destinatários, (3) inundação com destinatários inexistentes e (4) inundação através de *spoofing*. Na primeira categoria, a inundação é feita de uma maneira aleatória, ou seja, o atacante cria mensagens para destinatários selecionados aleatoriamente. A segunda categoria visa inundar alvos selecionados. Segundo os autores, um alvo com alto grau de participação na rede (centralidade) seria um alvo favorito para os atacantes. A inundação com destinatários inexistentes visa, entre outros efeitos, que as mensagens permaneçam a maior parte de tempo possível na rede. Sendo assim, os nós maliciosos criam mensagens com destinatários inexistentes, de forma que as mesmas jamais chegarão ao destino. Por último, na inundação através de *spoofing*, os nós maliciosos falsificam as origens das mensagens criadas. O objetivo é enganar a vítima, para que esta pense que as mensagens são de distintas origens.

Independente das classificações, em face do exposto argumenta-se que o ataque de inundação tem efeitos potencialmente danosos ao desempenho da rede, portanto, faz-se necessária a criação de medidas para limitar ou evitar os efeitos negativos deste ataque. As contramedidas tentam detectar ou mitigar o ataque de inundação de diversas maneiras, pode-se citar a utilização de um mecanismo de gerenciamento de *buffer* [61], a utilização de

um *gateway* responsável por detectar o comportamento malicioso [75], o estabelecimento de uma taxa limite de replicação [64] e a utilização de mecanismos de reputação [74, 86].

3.4 Falsificação de Identidade

A carência de métodos apropriados para fornecer autenticação em DTNs torna este tipo de rede propensa a ataques de falsificação de identidade (*spoofing*). Aproveitando desta vulnerabilidade, um nó malicioso M pode induzir um nó legítimo L a pensar que é na realidade o destino D de uma mensagem. Assim, M reivindica mensagens destinadas a D. Após isso M pode descartar estas mensagens. Além disso, D se convencerá de que entregou as mensagens ao destino e também realizará o descarte destas mensagens.

Segundo Uddin *et al.* [116], o ataque de falsificação de identidade é possivelmente severo, visto que além de descartar as mensagens de outros nós, o atacante também induz nós possivelmente legítimos a descartarem estas mensagens. Um nó malicioso pode perseguir um outro nó móvel na rede, anunciando de tempo em tempo uma nova identidade diferente. No caso da existência de um nó estacionário, como um *gateway*, o nó malicioso pode ficar em seu raio de alcance, anunciando diferentes identidades. Além disso, os autores citam que a falsificação de identidades pode acontecer de diversas formas que são mencionadas a seguir.

- Fixa: um conjunto de nós sempre reivindica o mesmo endereço, interceptando todos as mensagens destinadas ao nó vitimado.
- Entre pares: atacantes sempre escolhem endereços diferentes quando encontram diferentes nós, no entanto, sempre reivindicam o mesmo endereço quando encontram o mesmo nó.
- Aleatória: atacantes podem reivindicar quaisquer endereços que ele conheça aleatoriamente.

Choo *et al.* [25] demonstraram que um ataque de falsificação de identidade pode reduzir consideravelmente o desempenho da rede. Além disso, este ataque torna-se mais prejudicial quando combinado com o ataque de inundação. Segundo os autores, com um tempo de contato suficientemente longo, um nó malicioso falsificando identidades pode levar a remoção de todas as mensagens do *buffer* de um nó legítimo, passando-se como o destinatário de todas as mensagens deste nó.

Ademais, mencionou-se na Seção 3.3, a possibilidade de utilização do ataque de falsificação de identidades em conjunto com o ataque de inundação, falsificando a origem de mensagens durante um ataque de inundação com o objetivo de ludibriar as vítimas, para que pensem que as mensagens são de fontes distintas, dificultando a detecção do ataque e identificação do nó malicioso.

Uma contramedida da literatura contra o ataque de falsificação de identidade é denominada SPREAD (*countermeasure against SPoofing by REplica ADjustment*) [116]. Nela, protocolos de quota podem decidir por aumentar a quantidade de réplicas na rede caso detectem indícios de nós maliciosos realizando o ataque de falsificação de identidades. Esta detecção é realizada através da utilização de *tokens*. Quando um nó A encontra um nó B pela primeira vez, ele gera um *token* $T_A(B)$. Esse *token* é calculado utilizando-se uma função de dispersão sobre o identificador do nó B e uma *string* privada. Nos contatos subsequentes, os nós trocam os *tokens* recebidos no primeiro contato. Se um *token* diferente é retornado, diz-se que uma incompatibilidade de *tokens* aconteceu e conclui-se que dois nós afirmaram a mesma identidade e portanto, ao menos um deles é malicioso. Ademais, cada nó mantém um contador de incompatibilidades de *tokens* $cn(Y)$ para cada outro nó Y, de forma que possa estimar a quantidade de potenciais atacantes na rede.

3.5 Ataque *Sybil*

Em DTNs, dada a possível ausência de uma entidade centralizada responsável por atestar a veracidade de uma identidade, é possível que um nó crie várias identidades e use-as para prejudicar o funcionamento da rede. Este tipo de ataque é conhecido na literatura como ataque *Sybil* [34]. O ataque *sybil* pode ser realizado com o objetivo de burlar sistemas de reputação, recomendação, ou votação. Além disso, as várias identidades podem ser utilizadas com o objetivo de obter uma maior quantidade de recursos da rede.

Segundo Chen e Chan [21], mecanismos de reputação e esquemas de pagamento são vulneráveis ao ataque *sybil*. Nestes sistemas, um nó malicioso com múltiplas identidades pode aumentar a reputação de um nó indevidamente ou punir um nó alvo espalhando acusações falsas. Além disso, um ataque citado por Chen e Chan que é relacionado com o ataque *sybil* é o ataque *whitewashing*. Neste ataque, um nó malicioso deixa a rede e reingressa com uma nova identidade sempre que necessário, de modo a evitar sofrer as consequências de um sistema de reputação.

Trifunovic e Hossman-Picu [114, 115] estudam os tipos e efeitos de ataques *sybil* em

redes oportunistas. Os autores observam que um ataque deste tipo é constituído de duas ações diferentes: a criação de identidades e a criação de enlaces com nós honestos. Um fato interessante sobre as identidades é que elas podem ser divididas em dois diferentes grupos: identidades reais e identidades virtuais. As identidades reais, são aquelas que criam enlaces com nós honestos. As identidades virtuais são identidades que interagem somente com as identidades reais de um nó malicioso.

3.6 Ataque de Falsificação de Reconhecimentos Positivos

No ataque de falsificação de reconhecimentos positivos (ACK *Counterfeiting Attack*), nós maliciosos enviam reconhecimentos positivos falsificados com o objetivo de expurgar da rede mensagens que ainda não chegaram ao destino, impactando negativamente na taxa de entrega média da rede. Um reconhecimento positivo pode ser implementado como um *hash* criptográfico do conteúdo, fonte e destino de cada mensagem, mas nós maliciosos podem continuar enviando reconhecimentos falsificados para as mensagens que eles veem na rede. Além disso, nós maliciosos podem agir em conluio forjando reconhecimentos positivos para mensagens cujo destino ou a fonte não é nenhum dos nós coniventes com o conluio. Isto pode resultar em um aumento da taxa de entrega para os nós em conluio, porque recursos como espaço em *buffer* e contatos são liberados através do descarte impróprio de mensagens ocasionado pela falsificação de reconhecimentos positivos.

Burgess *et al.* [12] avaliaram o ataque de falsificação de reconhecimentos positivos em Redes Tolerantes a Atrasos e Desconexões sem autenticação. Os autores propuseram uma contramedida, daqui em diante chamada de BRG, baseada em uma premissa simples. Segundo eles, as mensagens são encaminhadas de nós próximos a fonte até nós próximos ao destino. Por sua vez, os reconhecimentos positivos percorrem o caminho contrário e são encaminhados de nós próximo ao destino até nós próximos a fonte. A contramedida BRG firma-se nesta premissa. Logo, nós rodando esta contramedida somente aceitarão um reconhecimento positivo como legítimo se já tiverem recebido a mensagem para qual este reconhecimento positivo foi gerado. Através de simulações, os autores demonstram que a BRG é efetiva na mitigação do ataque de falsificação de reconhecimentos positivos.

Capítulo 4

Avaliação do Uso de Reconhecimentos Positivos em DTNs

Um número de protocolos usa reconhecimentos positivos em DTNs. Por exemplo, os protocolos CARTOON [31], MaxProp [13], RAPID [4], ORWAR [96], Fuzzy-Spray [68], APRP-Ack [81] e *Storage Routing* [98] fazem uso de ACKs.

O protocolo NECTAR [30, 29] utiliza um tipo de reconhecimento que os autores chamam de passivo. Isto é, suponha que um determinado nó A tenha entregado uma mensagem M para outro nó B. Ao entrar em contato com um terceiro nó C que também tem a cópia da mensagem M, A avisará que a mensagem já foi entregue ao destino. Desta forma, o nó C pode remover a mensagem M de seu *buffer*. A principal diferença entre este tipo de reconhecimento passivo e os reconhecimentos anteriormente mencionados é que este mecanismo não propaga ativamente os ACKs, avisando somente os nós que possuem as mensagens de que elas foram entregues ao destino. A despeito da diferença, este mecanismo também é suscetível ao ataque de falsificação de reconhecimentos positivos.

No trabalho de An *et al.* [1], um mecanismo de disseminação de ACKs é proposto. Para tanto, dois modos para o envio de ACKs são identificados. O primeiro dos modos é o envio passivo de ACKs, assim como no protocolo NECTAR, comentado anteriormente. Além disso, o envio ativo de ACKs consiste na disseminação de ACKs para todos os nós da rede, independente se possuem ou não a mensagem para qual o ACK foi gerado, assim como acontece no protocolo MaxProp. A proposta consiste em alternar o modo de envio de ACKs na rede entre estes dois modos. Basicamente, prefere-se o envio passivo de ACKs quando indícios de congestionamento são detectados na rede. O objetivo é evitar agravar o congestionamento, visto que os ACKs ocupam espaço em *buffer*. A propósito, os autores consideram que o tamanho dos ACKs é de 5 KB e que o tamanho das mensagens é de

100 KB. Os resultados demonstraram que o mecanismo proposto, denominado *Congestion Level based end-to-end ACKnowledgment* (CL-ACK), possui desempenho superior aos modos de envio passivo e ativo, em termos de taxa de entrega.

Shevade *et al.* [102] propõem um mecanismo de incentivo *Tit-for-Tat* (TFT) para DTNs que utiliza reconhecimentos positivos como forma de prova de encaminhamentos realizados por outros nós. O objetivo é evitar que os nós da rede se comportem de maneira egoísta, o que prejudica o desempenho da rede. É interessante destacar que neste cenário em específico, nós maliciosos podem forjar ACKs com o objetivo de aumentar seus créditos no sistema de incentivo.

Raveneau *et al.* [92] avaliam a utilização de ACKs em um cenário de redes de sensores sem fio tolerante a atrasos e desconexões. O objetivo é verificar se a implementação do uso de ACKs é necessária em cenários com restrições rigorosas de memória. Os autores concluem que é melhor prover mecanismos que utilizem a memória de modo mais eficiente, como é o caso dos ACKs, ao invés de simplesmente aumentar a memória disponível nos nós. Além disso, conclui-se que o uso de ACKs reduz o número de mensagens que não mais contribuirão com a melhoria da taxa de entrega na rede, induzindo a um decréscimo da sobrecarga e, ao mesmo tempo, um aumento na taxa de entrega.

Com os exemplos supracitados, evidencia-se a difusão da utilização de reconhecimentos positivos em DTNs, para diversos fins, mas principalmente para o auxílio no controle de congestionamento. Neste capítulo, uma avaliação da utilização de reconhecimentos positivos em DTNs é conduzida. O objetivo desta avaliação é identificar os efeitos positivos que justificam a utilização dos ACKs na rede. Para tanto, são utilizadas três métricas de desempenho, quatro cenários de mobilidade reais e sete protocolos de roteamento em DTNs. A seguinte avaliação difere das avaliações existentes na literatura em razão da sua abrangência com relação à quantidade de protocolos e cenários avaliados.

4.1 Protocolos de Roteamento Utilizados

Sete protocolos da literatura foram utilizados na avaliação desta seção, são eles, Epidêmico, Life, Prophet, ProphetV2, MaxProp, *Spray and Wait* e Wave. Estes sete protocolos foram apresentados na Seção 3.1.

Destaca-se que estes protocolos lidam com a replicação de mensagens de maneiras distintas e portanto, atingem diferentes níveis de congestionamento. De modo breve, os controles de replicação desempenhados por cada um destes protocolos são especificados a

seguir.

O protocolo Epidêmico não impõe qualquer controle na replicação de mensagens. Pelo contrário, este protocolo replica as mensagens a maior quantidade possível, objetivando aumentar a confiabilidade da rede e por consequência, a taxa de entrega de mensagens. Logo, este protocolo é o mais suscetível ao congestionamento, pois estressa os recursos da rede através da replicação.

O protocolo Life estabelece um limite mínimo e máximo de réplicas de uma mensagem que podem existir nos vizinhos de um determinado nó. O objetivo é permitir que um número suficiente de réplicas seja mantido na rede para que a mensagem continue a ser transmitida na rede ao mesmo tempo que não inunde a rede com um número desnecessário de réplicas de uma mensagem. Visto que o protocolo Life trabalha com um limite de réplicas, este deve ser menos suscetível ao congestionamento do que o protocolo Epidêmico. Para o protocolo Life, a menos que estabelecido o contrário, os parâmetros da Tabela 4.1 foram utilizados na simulações deste trabalho. Ressalta-se que estes são os parâmetros escolhidos para avaliação pelos autores deste protocolo [83].

Tabela 4.1: Parâmetros utilizados para o protocolo Life.

Variável	l_b	l_s	u_b	u_s
Valor	0	0	3	3

Os protocolos Prophet e ProphetV2 utilizam a probabilidade de entrega calculada pelo próprio protocolo para decidir se uma mensagem deve ser replicada para outro nó. Mais especificamente, uma mensagem M_1 é replicada pelo nó A para o nó B somente se a probabilidade de B enviar esta mensagem ao destino D for maior do que a probabilidade de A enviar esta mensagem para o destino D. Logo, o protocolos Prophet e ProphetV2 são menos suscetíveis ao congestionamento do que o protocolo Epidêmico. Comparar protocolos Prophet e ProphetV2 com o protocolo Life, com relação ao congestionamento, é uma tarefa complexa neste momento, visto que o funcionamento destes protocolos depende das características de conectividade de cada rede. Caso não seja definido o contrário, os parâmetros listados nas Tabelas 4.2 e 4.3 são utilizados durante as simulações deste trabalho para os protocolos Prophet e ProphetV2, respectivamente.

Tabela 4.2: Parâmetros utilizados para o protocolo Prophet.

Variável	γ	β	***Time Unit (s)***
Valor	0,98	0,25	30

O protocolo MaxProp utiliza mecanismos que impõem um limite estrito de réplicas e

Tabela 4.3: Parâmetros utilizados para o protocolo ProphetV2.

Variável	**Gama**	**Beta**	***Time Unit (s)***	I_{typ}	P_{EncMax}
Valor	0,98	0,25	30	1800	0,5

encaminhamentos. Um destes mecanismos, chamado daqui em diante de *Hop List* (HL), mantém uma lista de saltos em cada mensagem, que ilustra o caminho percorrido por esta mensagem. Toda vez que uma mensagem M_x é replicada para um determinado nó A, A é inserido na lista de saltos da mensagem M_x. Uma mensagem não é replicada para um determinado nó A se este nó constar na lista de saltos desta mensagem. O mecanismo HL impõe um limite estrito de replicações de uma mensagem na rede, logo, o protocolo MaxProp tem um baixo nível de congestionamento, quando comparado com os protocolos anteriores. A menos que seja estabelecido o contrário, um parâmetro α utilizado para *incremental averaging* [56] é fixado em 1,0.

O protocolo *Spray and Wait* estabelece na rede um limite máximo de cópias e encaminhamentos por cada mensagem. Este limite é alcançado através da utilização de tíquetes de mensagens, como explicado na Seção 3.1.4. A depender da quantidade de tíquetes configurada, o protocolo *Spray and Wait* é o protocolo menos suscetível a congestionamento, dentre os protocolos avaliados.

O protocolo *Wave* mantém uma lista das mensagens que recebeu previamente e não receberá uma mensagem se esta está contida nesta lista de mensagens. As mensagens são mantidas na lista por um período predeterminado, denominado tempo de imunidade (*immunity time*). É interessante observar que dependendo do tempo de imunidade configurado, a rede pode tornar-se mais ou menos congestionada. Os parâmetros utilizados neste trabalho para o protocolo *Wave* estão de acordo com os parâmetros utilizados pelos autores e são listados na Tabela 4.4.

Tabela 4.4: Parâmetros utilizados para o protocolo Wave.

Variável	***Immunity Time***	***Custody Fraction***
Valor	300	2

4.2 Registros de Mobilidade Utilizados

Quatro registros reais de mobilidade são utilizados para avaliar a utilização de reconhecimentos positivos em DTNs, são eles: Dieselnet, Infocom, Rollernet e Shopping.

O conjunto de registros Rollernet [113] é resultado de um experimento no qual fo-

ram distribuídos 62 iMotes para voluntários. Este conjunto tem duração aproximada de 3 horas. No conjunto de registros Infocom [50], 41 iMotes foram distribuídos entre os participantes da conferência IEEE Infocom do ano de 2005. Este conjunto tem duração de aproximadamente 3 dias. Por sua vez, o conjunto de registros DieselNet possui 31 nós e é resultado de um experimento realizado pela Universidade de Massachusetts, que implantou uma plataforma de testes para DTNs usando seus ônibus [122, 13]. Por último, o conjunto Shopping [44] é resultado de um experimento no qual 25 dispositivos foram distribuídos em um *shopping center*, seja em pontos fixos ou carregados por vendedores.

É interessante ressaltar que estes registros são distintos em termos de duração, número de nós e conectividade, como pode ser observado na Tabela 4.2, que apresenta as seguintes características destes cenários:

- **Número de nós**: a quantidade de nós no cenário;
- **Tempo de duração**: o tempo de duração aproximado de cada cenário;
- **Número de conexões**: a quantidade total de conexões que são criadas durante toda a duração do cenário;
- **Conexões por minuto**: número médio de conexões por minuto em consideração toda a duração do cenário;
- **Média de conexões**: número médio de conexões para cada nó do cenário;
- **Média de Conexões (/nó/dia)**: número médio de conexões para cada nó para cada dia do tempo de simulação;
- **Tempo total de conexão**: soma do tempo de duração de todas as conexões que são criadas em cada cenário;
- **Tempo médio de conexão**: a média do tempo de duração de cada conexão;
- **Grau máximo dos nós**: número máximo de arestas (vizinhos) que um determinado nó tem durante toda a duração do cenário;
- **Grau médio dos nós**: representa a média dos graus de todos os nós a cada segundo, isto é, a cada segundo é calculado o grau médio, ao final, o grau médio do cenário é calculado através da razão entre a soma dos graus médios a cada segundo e a quantidade de segundos de simulação;

- **Maior componente conexa**: grau máximo de um subgrafo conectado durante toda a simulação.
- **Maior componente conexa (%)**: grau máximo de um subgrafo conectado durante toda a simulação dividido pelo número de nós da simulação.

Tabela 4.5: Características de conectividade dos cenários de mobilidade.

Características	**Cenário**			
	Dieselnet	**Infocom**	**Rollernet**	**Shoppping**
Dispositivo	802.11	*iMote*	*iMote*	*Bluetooth*
Tempo de Duração (≈)	7,8 dias	3 dias	3 horas	5,5 dias
Número de Nós	31	41	62	25
Número de Conexões	1.052	22.459	15.803	26.651
Conexões por Minuto	0,09	5,30	94,81	3,37
Média de Conexões	33,93	547,78	254,89	1.066
Média de Conexões (/nó/dia)	4,35	182,66	2.040	193,81
Tempo Total de Conexão (h)	2.382	1.446	95,47	6.931
Tempo Médio de Conexão (m)	135	3,9	0,36	15,6
Grau Máximo dos Nós	8	14	10	19
Grau Médio dos Nós	0,34	0,52	1,11	4,21
Maior Componente Conexa	18	31	41	25
Maior Componente Conexa (%)	0,58	0,75	0,66	1

Como forma de medir a conectividade de cada cenário, para cada métrica de conectividade os cenários foram ordenados de maneira descendente e um sistema de pontuação foi utilizado. Foram escolhidas as métricas que estão de alguma forma normalizadas, visto que números absolutos podem não refletir a conectividade real dos cenários. Para cada métrica escolhida, são atribuídos 3 pontos ao cenário com maior conectividade, 2 pontos ao cenário com segunda maior conectividade e assim por diante. O cenário com menor valor para a métrica não pontua. O resultado desta classificação é exibido na Tabela 4.6.

Tabela 4.6: Classificação dos cenários referente as métricas.

Características	***Cenários/Pontuação***			
	3	**2**	**1**	**0**
Conexões por Minuto	Rollernet	Infocom	Shopping	Dieselnet
Média de Conexões (/nó/dia)	Rollernet	Shopping	Infocom	Dieselnet
Tempo Médio de Conexão (m)	Dieselnet	Shopping	Infocom	Rollernet
Grau Médio dos Nós	Shopping	Rollernet	Infocom	Dieselnet
Maior Componente Conexa (%)	Shopping	Infocom	Rollernet	Dieselnet

A pontuação final obtida por cada um dos cenários é exibida na Tabela 4.7. Nela, é possível observar que o cenário que obteve a maior pontuação foi o cenário Shopping, com 11 pontos, seguido pelos cenários Rollernet e Infocom, com 9 e 7 pontos, respectivamente. Por último, o cenário Dieselnet alcançou 3 pontos.

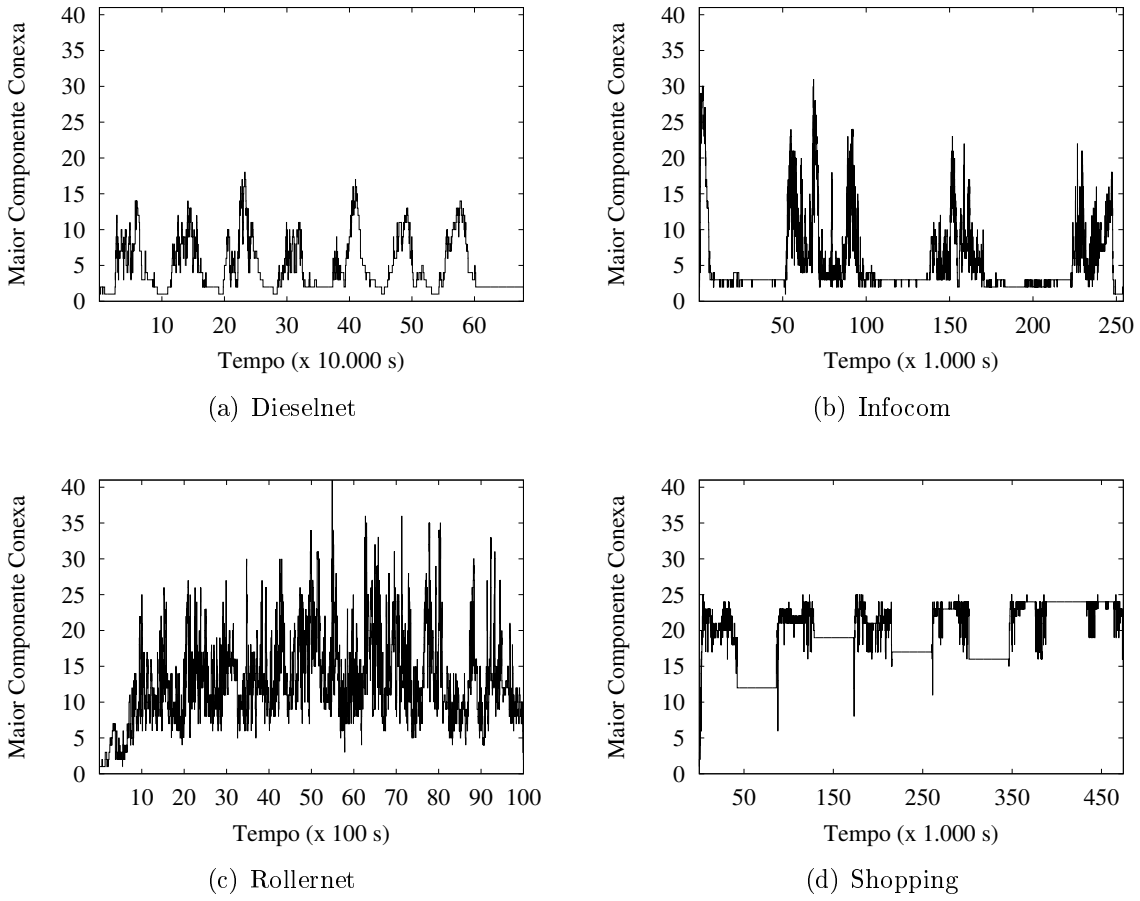
Embora provavelmente não retrate precisamente a conectividade dos cenários, visto que um maior número de características deve ser levado em consideração para isso, o método juntamente com as métricas oferecem fundamentos razoáveis sobre a conectividade de cada cenário. Desta forma, estas noções serão utilizadas na avaliação deste trabalho, em conjunto com informações obtidas em outro trabalhos [89, 89, 50].

Tabela 4.7: Pontuação final obtida por cada cenário na classificação de conectividade utilizada.

Cenário	**Dieselnet**	**Infocom**	**Rollernet**	**Shopping**
Pontuação total	3	7	9	11

A Figura 4.1 ilustra a evolução segundo a segundo da maior componente conexa nos 4 cenários. Observa-se em todos os cenários uma certa oscilação no tamanho da maior componente conexa, onde a rede alterna entre períodos de alta e baixa conectividade. Para o cenário Dieselnet, esta oscilação pode ser explicada pelos períodos de atividade e inatividades dos veículos utilizados no experimento. No cenário Infocom, os períodos de baixa conectividade podem ser explicados pela mobilidade dos participantes da conferência, que recolhem-se para seus quartos e outras atividades ao fim da programação diária da conferência, diminuindo a conectividade. Esta oscilação foi estudada para o cenário Rollernet no que os autores nomearam de "Efeito Acordeão" [113]. Segundo os autores, a oscilação é advinda da contração e expansão dos patinadores, adaptando seus movimentos

ao trânsito. Por último, no cenário Shopping, os períodos de menor conectividade podem ser explicados pela mobilidade dos vendedores, que retornam para seus lares ao final de cada dia de trabalho. Em todos os casos, observa-se que a conectividade é resultado da mobilidade adjacente dos agentes que portam os dispositivos coletores, sejam veículos ou pessoas.



(a) Dieselnet (b) Infocom

(c) Rollernet (d) Shopping

Figura 4.1: Evolução da maior componente conexa a cada segundo para os 4 cenários.

De posse de todas as informações apresentadas nesta seção, é razoável classificar os cenários Shopping e Rollernet como os cenários de maior conectividade e os cenários Dieselnet e Infocom como os cenários de menor conectividade. Estas informações serão utilizadas posteriormente como ferramentas no auxílio para interpretação dos resultados obtidos neste trabalho.

4.3 Ambiente de Simulação

Para esta avaliação, versões modificadas dos protocolos escolhidos para a avaliação, que suportam a utilização de reconhecimentos positivos, foram implementadas no simu-

Tabela 4.8: Parâmetros das simulações e características dos cenários.

Parâmetros/Conjuntos	Dieselnet	Infocom	Rollernet	Shopping
Dispositivo	802.11	iMote	iMote	*Bluetooth*
Duração (≈)	7,8 dias	3 dias	3 horas	5,5 dias
Número de nós	31	41	62	25
Número de mensagens	1.000	1.000	1.000	1.000
Tamanho das mensagens (MB)	1	1	1	1
Taxa de Transmissão (Mbps)	1	1	1	1
TTL	∞	∞	∞	∞
Tamanho do *Buffer* (MB)	20	20	20	20

lador *Opportunistic Network Environment* (ONE) [58]. Para o protocolo MaxProp foi implementada uma versão que não faz uso de reconhecimentos positivos, visto que sua versão original já suporta este mecanismo.

Os parâmetros de configuração das simulações também são resumidos na Tabela 4.8. O padrão de tráfego de mensagens para cada conjunto de registros é o seguinte. Para os conjuntos Rollernet, Dieselnet, Infocom e Shopping, foram geradas 1.000 mensagens nas primeiras, respectivamente, 2, 143, 52 e 125 horas, aproximadamente, de um tempo de simulação total de aproximadamente 3, 187, 72 e 132 horas de simulação. O período de inatividade na geração de mensagens tem como objetivo diminuir a quantidade de mensagens que não chegam ao destino devido ao encerramento abrupto da simulação. O tamanho configurado para as mensagens é de 1,0 MB, visto que redes DTN trabalham com agregados que podem conter várias mensagens. Ainda sobre o padrão de tráfego, em todos os cenários os nós maliciosos não são fonte e nem destino de nenhuma mensagem. Para evitar efeitos negativos da má configuração do TTL, nenhuma mensagem expira durante todo o período de simulação. O tamanho do *buffer* foi variado de 20 M a 100 M.

4.4 Resultados Obtidos

Para avaliar os efeitos da utilização de reconhecimentos positivos três métricas de desempenho são utilizadas: a taxa de entrega de mensagens, a sobrecarga de transmissão de mensagens e o atraso de entrega de mensagens[1]. Os resultados apresentados foram obtidos através da média de 10 rodadas de simulação distintas. Para todas as médias apresentadas nas figuras, calcula-se um intervalo de confiança para um nível de confiabilidade de 95% através da distribuição *t-Student*. Este intervalo de confiança é representado por barras verticais.

[1]Doravante os termos atraso de entrega de mensagens e latência são utilizados de modo intercambiável.

Os resultados para protocolos Epidêmico, MaxProp, Prophet e *Spray and Wait* foram escolhidos para serem apresentados nesta seção em razão dos diferentes modos de operação destes protocolos, que podem representar uma grande quantidade de protocolos existentes na literatura. Os demais resultados foram omitidos com o objetivo de evitar a repetição excessiva, visto que são similares aos resultados exibidos nesta seção. Estes resultados podem ser conferidos no Apêndice A.

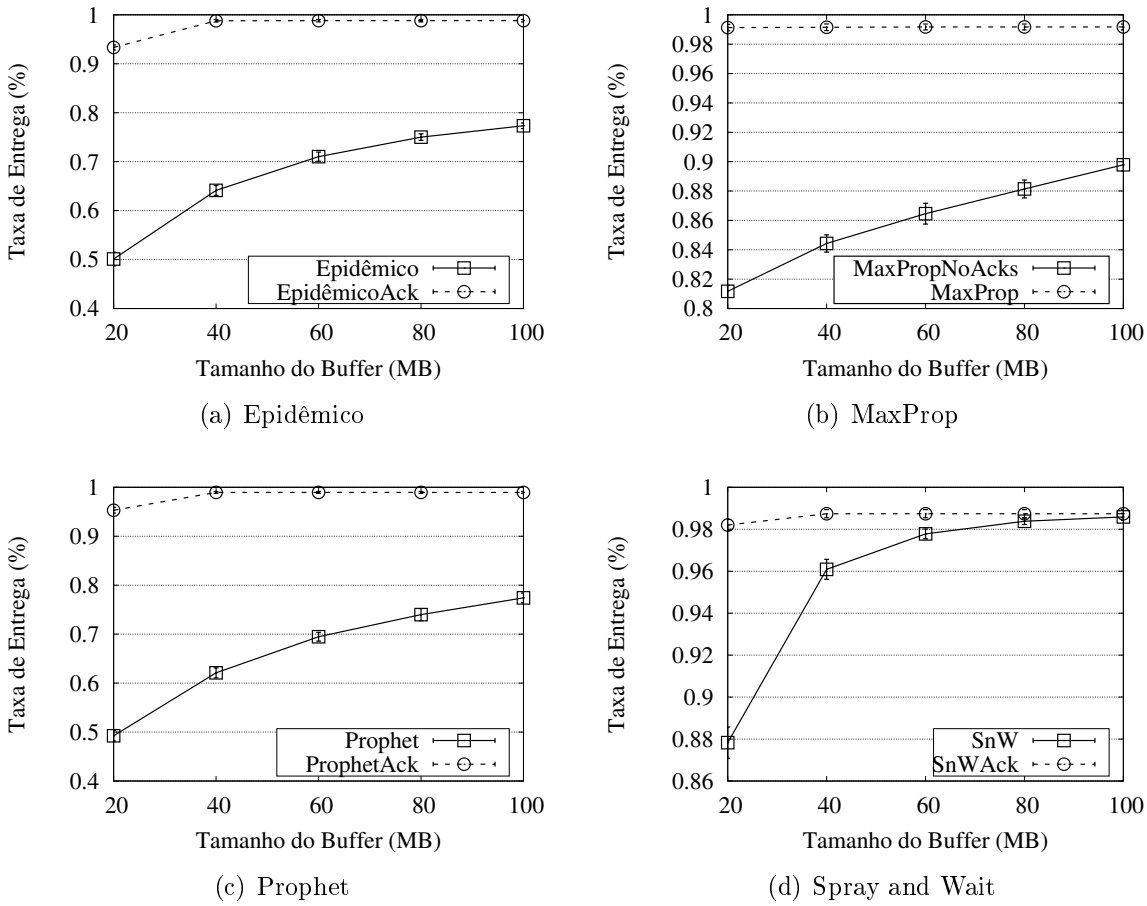
Todas as imagens desta avaliação comparam a versão do protocolo que não utiliza reconhecimentos positivos com sua versão que comporta a utilização de reconhecimentos positivos. Nesta avaliação de resultados, assumiu-se a seguinte nomenclatura, os protocolos Epidêmico, MaxPropNoAcks, Prophet e *Spray and Wait* são as versões dos protocolos que não utilizam reconhecimentos. As curvas dos protocolos que utilizam reconhecimentos positivos foram nomeadas como EpidêmicoAck, MaxProp, ProphetAck e SnWAck, sendo esta última a nomenclatura adotada para protocolo *Spray and Wait*. A mesma regra de nomenclatura foi adotada para os protocolos cujos resultados são exibidos no Apêndice A, ou seja, LifeAck, ProphetV2Ack e WaveAck são os protocolos modificados para utilizarem reconhecimentos positivos. Por sua vez, Life, ProphetV2 e Wave são as versões padrão destes protocolos que não utilizam o mecanismo de reconhecimento de mensagens.

É válido informar novamente que as versões padrão dos protocolos Epidêmico, Prophet e *Spray and Wait* não utilizam reconhecimentos positivos, enquanto que a versão padrão do protocolo MaxProp utiliza este mecanismo. Logo, os protocolos Epidêmico, Prophet e *Spray and Wait* foram modificados para comportar a utilização de reconhecimentos positivos, enquanto que o protocolo MaxProp foi modificado para não utilizar reconhecimentos positivos.

4.4.1 Taxa de Entrega

A taxa de entrega é definida como o percentual de mensagens criadas que chegaram ao destino durante a simulação. Seja a taxa de entrega representada por T_E , M a quantidade de mensagens criadas e M_E a quantidade de mensagens que chegaram ao destino, a taxa de entrega é calculada da seguinte maneira: $T_E = \frac{M_E}{M}$.

Para as figuras exibidas nesta seção, o eixo Y ilustra a taxa de entrega de mensagens enquanto que o eixo X ilustra o tamanho dos *buffers* dos nós de 20 M a 100 M. As Figuras 4.2 e 4.3 ilustram as taxas de entregas obtidas para os cenários Rollernet e Dieselnet, respectivamente. Para todos os cenários, algumas tendências que podem ser observadas nestes resultados são discutidas a seguir.



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 4.2: Taxa de entrega para o cenário Rollernet.

Em primeiro lugar, nota-se que quanto maior o tamanho do *buffer*, maior a taxa de entrega. Isto pode ser explicado pelo fato de que quanto menor o tamanho do *buffer*, maior a probabilidade de descarte de mensagens devido à sobrecarga dos *buffers* dos nós. Além disso, o aumento do tamanho dos *buffers* dos nós possibilita que um número maior de mensagens permaneça disponível na rede por um período maior de tempo, levando a uma melhoria na taxa de entrega.

Também é possível observar que a utilização de reconhecimentos positivos leva a um melhor desempenho com relação a taxa de entrega. As Tabelas 4.9 e 4.10 quantificam o percentual de melhoria relativa e absoluta de todos os protocolos para os cenários Rollernet e Dieselnet, respectivamente. A melhoria absoluta M_A é calculada através da diferença entre a taxa de entrega $Delivery_{ack}$ com a utilização de ACKs e a taxa de entrega sem a utilização de ACKs $Delivery$, como apresentado na Equação 4.1.

$$M_A = (Delivery_{ack} - Delivery) \times 100 \tag{4.1}$$

Por sua vez, a melhoria relativa representa a proporção de ganho de desempenho alcançada pela utilização de ACKs na rede e é calculada como na Equação 4.2.

$$M_R = \left(\frac{Delivery_{ack}}{Delivery} - 1 \right) \times 100 \tag{4.2}$$

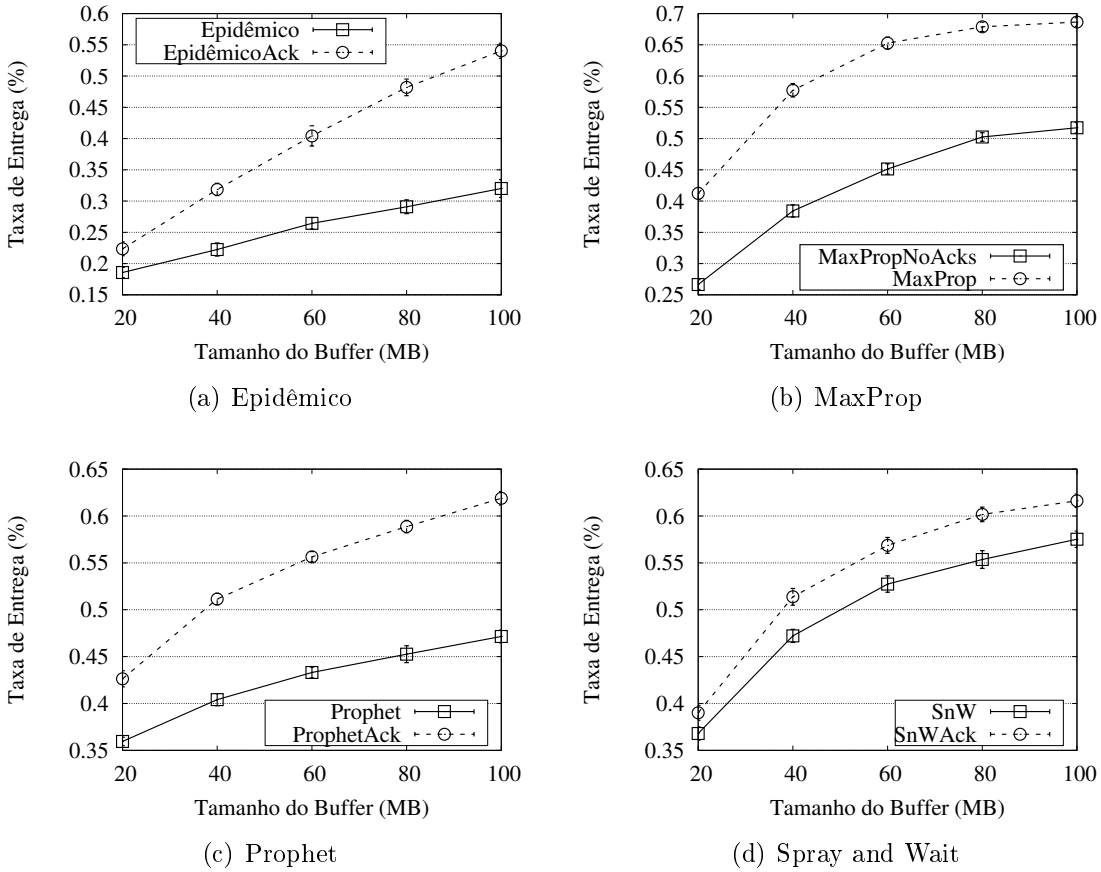
Tabela 4.9: Melhoria na taxa de entrega para o cenário Rollernet.

	Melhoria Absoluta (%)					**Melhoria Relativa (%)**				
Protocolo/Buffer (MB)	**20**	**40**	**60**	**80**	**100**	**20**	**40**	**60**	**80**	**100**
Epidêmico	43	34	28	24	21	86	54	39	32	28
Life	43	34	27	23	21	86	53	38	30	27
MaxProp	18	15	13	11	9	22	17	15	12	10
Prophet	46	37	30	25	22	93	59	42	34	28
ProphetV2	31	21	16	14	12	46	26	20	16	14
SnW	10	3	1	0	0	12	3	1	0	0
Wave	58	36	28	24	21	189	58	39	32	28

Observas-se que a utilização de ACKs contribui para uma melhoria absoluta na taxa de entrega para o protocolo Wave no cenário Rollernet de até 58%. Relativamente, esta melhoria chega a alcançar 189%. Mesmo nos cenários com maior tamanho de *buffer*, o uso de ACKs implica em um aumento absoluto de até 22% na taxa de entrega. No cenário Dieselnet, 22% é a melhoria absoluta máxima em decorrência da utilização de ACKs. Em termos relativos, esta melhoria alcança 70%.

A melhoria de desempenho com a utilização de ACKs é justificada pela diminuição do congestionamento da rede. Isto acontece porque os ACKs são propagados com o objetivo de remover dos *buffers* dos nós as mensagens que já chegaram ao destino, liberando espaço de armazenamento para mensagens que ainda não foram entregues com consequente aumento da disponibilidade destas mensagens na rede. Por fim, ao empregar seus recursos no encaminhamento e replicação de mensagens que ainda não chegaram ao destino, diminuindo a quantidade de recursos utilizados com mensagens que já foram entregues, a rede alcança uma melhoria de desempenho com relação a taxa de entrega.

Quando comparados, os resultados das Tabelas 4.9 e 4.10 mostram que enquanto a melhoria de desempenho para os cenário Rollernet diminui com o aumento do *buffer*, o contrário ocorre para o cenário Dieselnet, isto é, a melhoria de desempenho aumenta conforme o tamanho dos *buffers* dos nós também aumenta. Este comportamento decorre do fato de que o cenário Dieselnet é menos conectado, logo, o aumento do *buffer* neste cenário tem maior impacto na diminuição do congestionamento, o que leva a melhores resultados com relação a taxa de entrega. Por outro lado, o cenário Rollernet é mais



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 4.3: Taxa de entrega para o cenário Dieselnet.

conexo, resultando em uma melhor taxa de entrega mesmo com tamanhos de *buffer* mais restritos. Neste caso, o incremento do *buffer* apesar de contribuir para uma melhoria de desempenho, tem menor impacto pois o cenário é menos impactado pelo congestionamento.

Finalmente, observa-se que os protocolos que menos controlam a replicação são os mais beneficiados pelo uso de reconhecimentos positivos. Dentre os protocolos avaliados, aqueles que fazem uso de replicação limitada, isto é *Spray and Wait* e MaxProp, são os que alcançaram menor melhora de desempenho através do uso de ACKs com relação a taxa de entrega. Este é um resultado esperado, visto que ao valerem-se de estratégias de controle de congestionamento mais restritivas, estes protocolos estressam menos os recursos da rede, diminuindo o congestionamento. Além disso, estes protocolos alcançam os melhores desempenhos para os cenários avaliados, deixando uma menor margem para a melhoria do desempenho.

Tabela 4.10: Melhoria na taxa de entrega para o cenário Dieselnet.

	Melhoria Absoluta (%)					**Melhoria Relativa (%)**				
Protocolo/Buffer (MB)	**20**	**40**	**60**	**80**	**100**	**20**	**40**	**60**	**80**	**100**
Epidêmico	4	10	14	19	22	20	43	53	66	69
Life	4	10	14	18	19	20	35	48	53	53
MaxProp	15	19	20	18	17	55	50	45	35	33
Prophet	7	11	12	14	15	19	26	28	30	31
ProphetV2	7	12	14	14	14	18	28	30	30	28
SnW	2	4	4	5	4	6	9	8	9	7
Wave	6	12	15	20	22	44	54	59	67	70

4.4.2 Sobrecarga

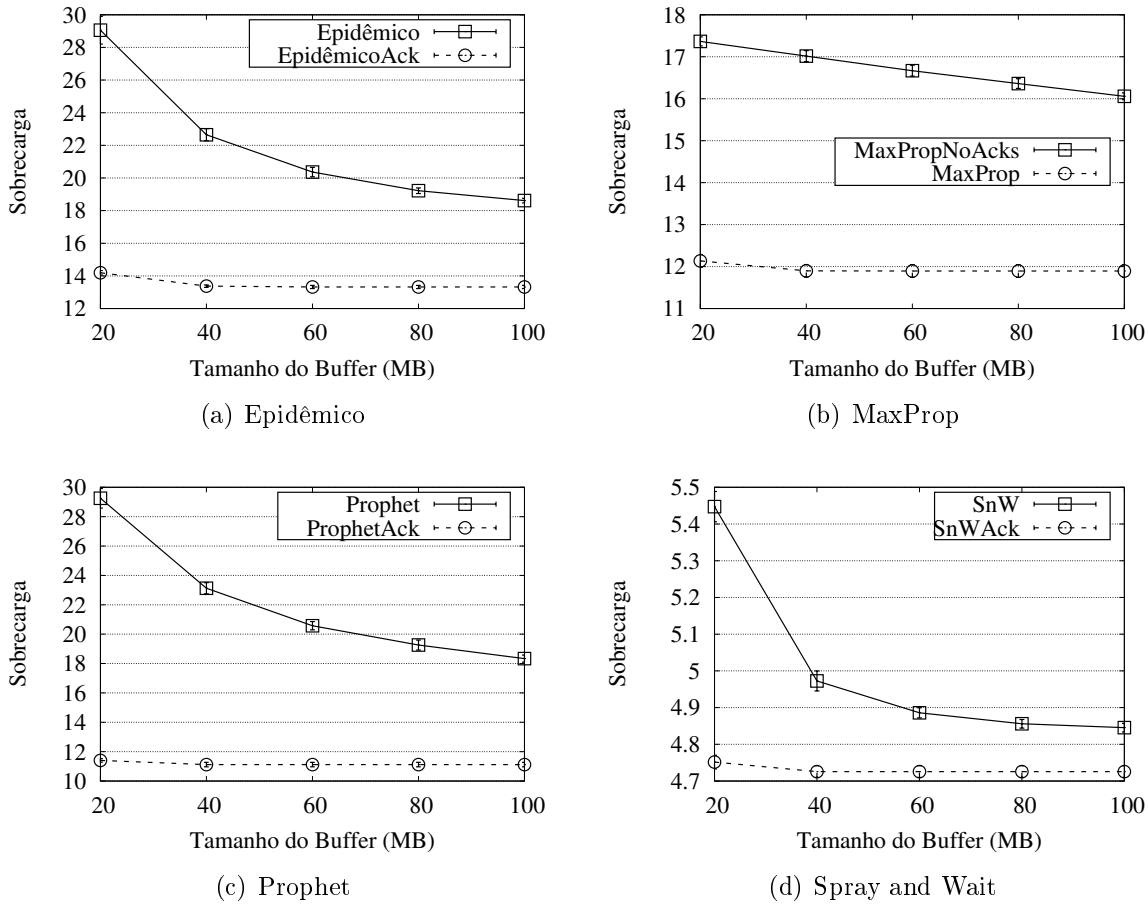
A sobrecarga S é calculada pela razão entre a diferença do número de mensagens transmitidas M_T e mensagens entregues M_E e o número de mensagens entregues M_E, como apresentado na Equação 4.3.

$$S = \frac{M_T - M_E}{M_E} \tag{4.3}$$

Essa relação expressa a quantidade de mensagens adicionais que foram transmitidas para cada mensagem que chegou ao destino e pode ser compreendida como uma medida de eficiência do protocolo. Com relação aos gráficos desta seção, o eixo Y ilustra a sobrecarga como calculada na Equação 4.3 e o eixo X representa o tamanho dos *buffers* dos nós, variando de 20 M a 100 M.

As Figuras 4.4 e 4.5 apresentam os resultados obtidos para sobrecarga nos cenários Rollernet e Dieselnet, respectivamente. A mesma tendência pode ser observada em todos os cenários: a utilização de reconhecimentos positivos aumenta a eficiência da rede através da diminuição da sobrecarga. Para o cenário Rollernet, a utilização de ACKs resulta em uma sobrecarga razoavelmente estabilizada para todos os tamanhos de *buffer* em todos os protocolos. No entanto, para o cenário Dieselnet, a sobrecarga continua a diminuir com o aumento do tamanho do *buffer*. Isto acontece em razão da baixa conectividade deste cenário. Neste caso, o aumento do tamanho do *buffer* contribui significativamente para o aumento da taxa de entrega, como visto na Seção 4.4.1, levando a uma diminuição da sobrecarga, visto que a sobrecarga é expressada pela quantidade de mensagens encaminhadas pela quantidade de mensagens entregues.

Ainda no cenário Dieselnet, para o protocolo Epidêmico, como exibido na Figura 4.5(a), observa-se um aumento na sobrecarga quando o tamanho do *buffer* é alterado de 20 M



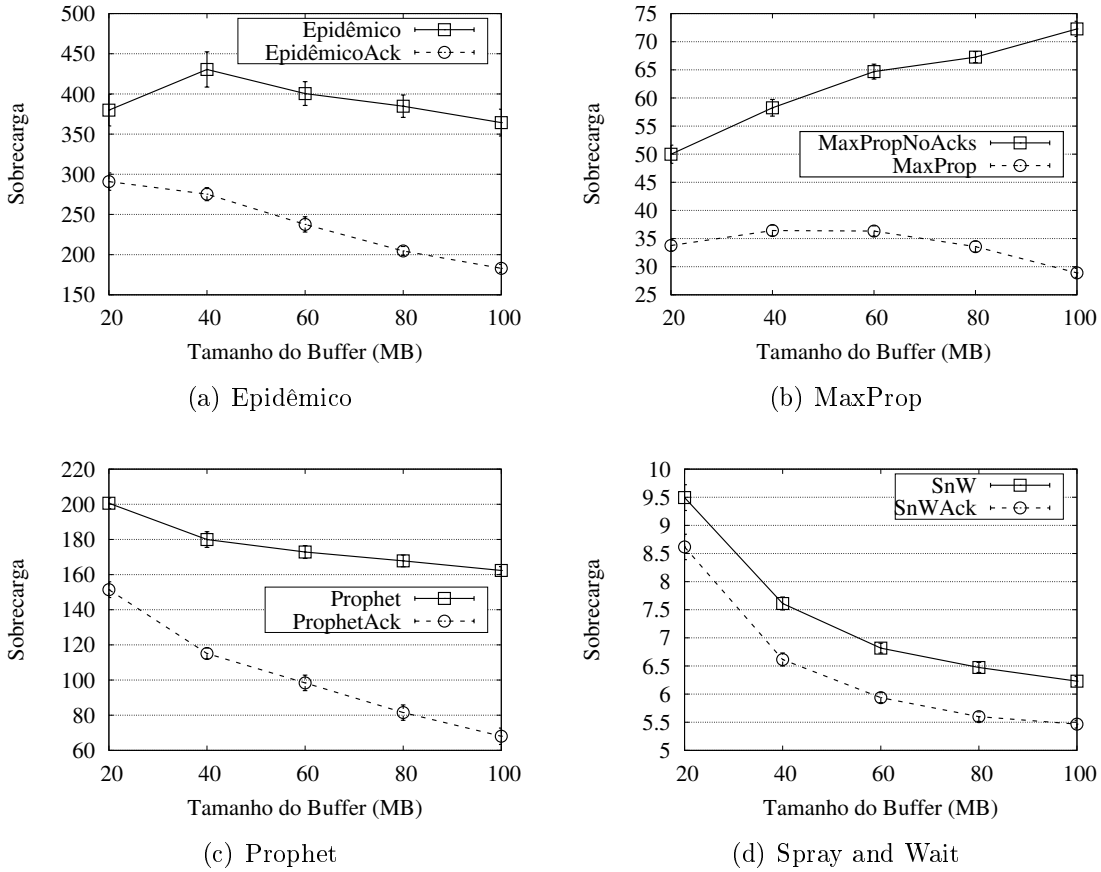
(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 4.4: Sobrecarga de entrega para o cenário Rollernet.

para 40 M. Isto ocorre porque o protocolo Epidêmico replica ao máximo possível suas mensagens, sem fazer nenhum controle. Logo, quanto o *buffer* é aumentado, aumenta também a quantidade de mensagens que os nós podem replicar a cada conexão, visto que o Dieselnet possui conexões relativamente duradouras, como pode ser observado através da métrica "Tempo Médio de Conexão", na Tabela 4.2. Este aumento na replicação não é totalmente compensado pela melhoria do desempenho com relação a taxa de entrega, levando a um aumento da sobrecarga. De maneira similar, o protocolo MaxPropNoAcks também leva a um aumento da sobrecarga, conforme aumenta o tamanho do *buffer*. Assim como no protocolo Epidêmico, isto pode ocorrer devido ao fato de que aumento na taxa de entrega não acompanha o aumento na quantidade de replicações proporcionado pelo acréscimo dos *buffers* dos nós, resultando em um aumento na sobrecarga para o protocolo MaxPropNoAcks.

Finalmente, assim como observado para a taxa de entrega, os protocolos que menos controlam a replicação na rede são mais beneficiados em termos absolutos pela utilização de reconhecimentos positivos.



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

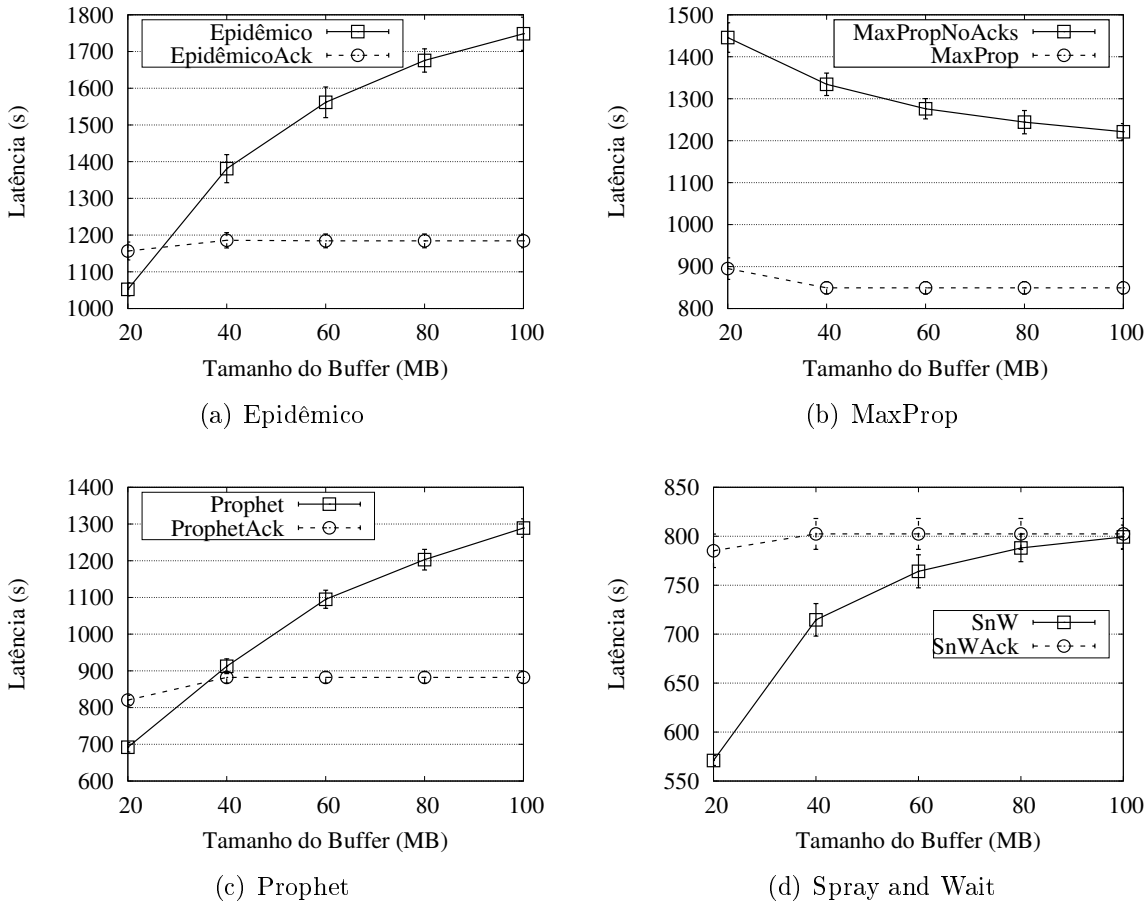
Figura 4.5: Sobrecarga para o cenário Dieselnet.

4.4.3 Atraso de Entrega

A métrica de atraso de entrega é compreendida como a média de tempo entre a criação e a entrega das mensagens, dentre as mensagens entregues. Cabe ressaltar que mensagens que não são entregues durante a operação da rede não são utilizadas no cálculo de atraso de entrega. Com relação aos gráficos de atraso de entrega, o eixo Y ilustra o atraso de entrega médio em segundos e o eixo X ilustra o tamanho dos *buffers* dos nós, variando de 20 M a 100 M.

As Figuras 4.7 e 4.6 apresentam os resultados obtidos para a métrica de atraso de entrega nos cenários Dieselnet e Rollernet, respectivamente. Para o cenário Rollernet, é possível observar que a utilização de reconhecimentos positivos mantém um atraso de entrega relativamente estável, enquanto que os protocolos que não utilizam ACKs têm um aumento neste atraso conforme o tamanho dos *buffers* dos nós aumenta.

O aumento do atraso tem relação com o aumento do tamanho do *buffer* e pode ser explicado pelo fato de que o aumento deste leva consequentemente a maiores atrasos



(a) Epidêmico (b) MaxProp

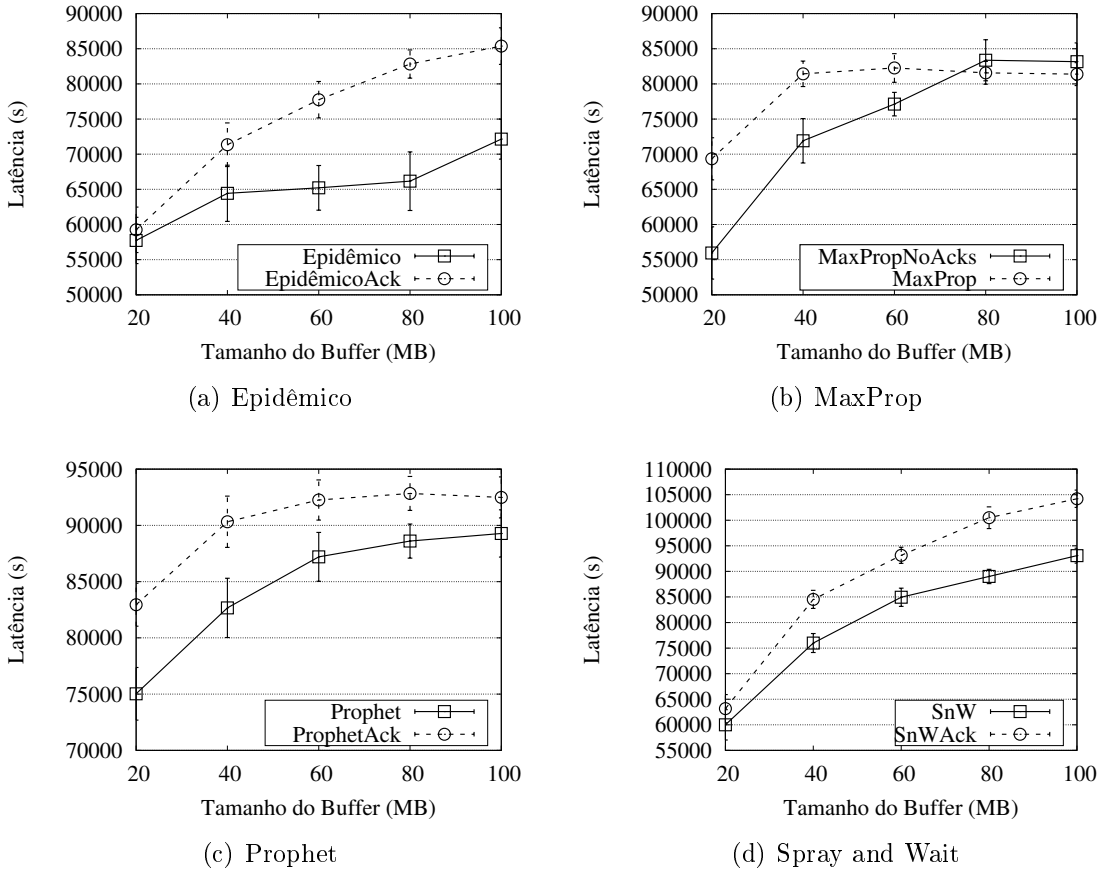
(c) Prophet (d) Spray and Wait

Figura 4.6: Atraso de entrega para o cenário Rollernet.

de enfileiramento, visto que os nós passam a manter por mais tempo mensagens que antes descartariam. Além disso, aumentar o tamanho dos *buffers* permite a entrega de mensagens que outrora não chegariam ao destino, pois necessitam trafegar por caminhos mais longos. Isto é, a consequente entrega de mensagens com um maior atraso de entrega eleva a média de atraso de entrega na rede.

Uma exceção ao aumento no atraso de entrega no cenário Rollernet é o protocolo MaxProp, como exibido na Figura 4.6(b). Isto decorre da maior aptidão deste protocolo em disseminar eficientemente as mensagens na rede. O MaxProp alcança um melhor desempenho através de vários mecanismos, desde a previsão de probabilidade de entrega através de contatos, a priorização de mensagens que foram pouco replicadas na rede, a utilização do mecanismo *Hop List*, entre outros mecanismos.

Para o cenário Dieselnet, o aumento do tamanho dos *buffers* dos nós aumenta o atraso de entrega para todos os protocolos avaliados. É relevante ressaltar que este cenário possui baixa conectividade e que esta característica, junto com o aumento dos *buffers* dos nós, resulta em um maior tempo de enfileiramento nos *buffers* dos nós. Entregar um



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 4.7: Atraso de entrega para o cenário Dieselnet.

maior número de mensagens que passam mais tempo armazenadas nos *buffers* dos nós até chegarem ao destino resulta no aumento do atraso de entrega médio. Isto também explica o fato de a utilização de ACKs resultar em maiores médias de atraso de entrega neste cenário. Este comportamento é resultante da característica pouco conexa do cenário, que possui o menor número de conexões.

4.5 Conclusões

Como resultado da utilização de reconhecimentos positivos, além de melhorar em até 189% a taxa de entrega da rede, sua eficiência também é melhorada, visto que os ACKs levam a uma redução na sobrecarga. A melhoria na taxa de entrega pode ser explicada pela diminuição do congestionamento da rede, visto que a propagação dos ACKs libera espaço nos *buffers* dos nós. A melhoria na sobrecarga é um efeito colateral da remoção de mensagens que já foram entregues dos *buffers* dos nós. Nestes casos, os dispositivos da rede param de encaminhar e replicar mensagens que já foram entregues, com considerável aumento na eficiência da rede.

Observou-se também que os protocolos mais beneficiados pelo uso de reconhecimentos positivos são aqueles que menos controlam a quantidade de réplicas na rede e portanto, sofrem um maior congestionamento.

Com relação a latência, observou-se que esta pode diminuir em razão da diminuição do congestionamento ou aumentar, em razão da entrega de um maior número de mensagens. Estes resultados dependem das características de conectividade do cenário em questão.

Em face do exposto, é factível argumentar que a utilização de reconhecimentos positivos é uma alternativa viável para atenuar os efeitos do congestionamento.

Capítulo 5

Avaliação do Ataque de Falsificação de Reconhecimentos Positivos em DTNs

Como demonstrado no Capítulo 4, nós de uma DTN podem lançar mão da utilização de reconhecimentos positivos para aumentar o desempenho e a eficiência da rede. No entanto, a dificuldade de estabelecimento de uma infraestrutura para autenticação nestes tipos de redes propicia que nós maliciosos realizem ataques. Neste caso, um dos ataques que pode ser realizado é o ataque de falsificação de reconhecimentos positivos, como explicado na Seção 3.6.

Em trabalhos anteriores [12, 25] os autores classificaram as DTNs robustas contras certos ataques, entre eles o ataque de falsificação de reconhecimentos positivos. No entanto, a avaliação de Burgess *et al.* [12] considera somente dois cenários de mobilidade, uma métrica de desempenho e além disso, os resultados apresentados são sucintos.

Neste capítulo, pretende-se avaliar os efeitos negativos do ataque de falsificação de reconhecimentos positivos em DTNs. Para esta finalidade, três métricas de desempenho, quatro cenários e sete protocolos de roteamento são utilizados. O objetivo é avaliar os riscos introduzidos em uma rede sem autenticação ao optar-se pela utilização de reconhecimentos positivos.

5.1 Avaliação do Ataque de Falsificação de Reconhecimentos Positivos

Esta seção apresenta a avaliação do ataque de falsificação de reconhecimentos positivos em DTNs. Os protocolos de roteamento utilizados para a avaliação, bem como os registro de mobilidade reais, são apresentados na Seção 5.2. Os modelos de atacantes im-

plementados são apresentados na Seção 5.2.1 e o ambiente de simulação é apresentado na Seção 5.3. Os resultados obtidos são apresentados na Seção 5.4. Finalmente, as conclusões alcançadas são apresentadas na Seção 5.5.

5.2 Protocolos de Roteamento e Registros de Mobilidade

Sete protocolos de roteamento foram utilizados para a avaliação deste capítulo. São eles: Epidêmico, *Life*, MaxProp, Prophet, ProphetV2, *Spray and Wait* e *Wave*. Estes são os mesmos protocolos utilizados para a avaliação do uso de reconhecimentos positivos realizada no Capítulo 4. Para evitar redundância, as descrições dos protocolos foram omitidas e para mais informações podem ser consultadas nas Seções 3.1 e 4.1.

De modo similar, optou-se pela utilização dos mesmos registros de mobilidade utilizados no Capítulo 4, isto é, Dieselnet, Infocom, Rollernet e Shopping. As informações sobre estes registros de mobilidade podem ser consultadas na Seção 4.2.

5.2.1 Modelos de Ataque

Dois modelos de atacantes são definidos. No primeiro, daqui em diante chamado simplesmente de ataque de falsificação de reconhecimentos positivos, nós maliciosos enviam ACKs para cada mensagem que eles recebem na rede. Neste ataque, nós maliciosos não removem mensagens de seus *buffers* para as quais eles geraram ACKs falsificados. Com relação ao comportamento legítimo dos nós, eles também enviam ACKs legítimos que recebem e atuam como retransmissores de mensagens. O objetivo é tornar o comportamento do nó malicioso tão próximo do legítimo quanto possível.

No segundo modelo, os nós maliciosos também descartam de seus *buffers* mensagens para os quais ACKs falsificados foram gerados. Visto que os nós maliciosos geram reconhecimentos positivos falsificados para todas as mensagens que recebem, eles acabam descartando todas as mensagens que recebem, logo após a geração dos reconhecimentos positivos falsificados. Em razão disso, este modelo de ataque é daqui em diante chamado de ataque de falsificação de reconhecimentos positivos com buraco negro. Na realidade, este pode ser considerado um ataque de descarte, visto que os nós maliciosos não enviam informação de roteamento adulterada com o objetivo de atrair tráfego, como mencionado na Seção 3.2.

5.3 Ambiente de Simulação

Os modelos de atacantes foram implementados no simulador ONE, para os sete protocolos de roteamento utilizados. A Tabela 5.1 sumariza as configurações das simulações utilizadas. Ressalta-se que as mesmas configurações foram utilizadas em todos os cenários avaliados.

Para todos os cenários, 1.000 mensagens são geradas durante cada rodada de simulação. A geração destas mensagens é finalizada cerca de 6 horas antes do final da rodada de simulação para os cenários Dieselnet, Infocom e Shopping. Para o cenário Rollernet, a geração termina cerca de 1 hora antes do final da duração do cenário. O período final de cerca de 1 hora para o cenário Rollernet e 6 horas para os cenários Dieselnet, Infocom e Shopping visa evitar que mensagens sejam geradas a pouco tempo do final da simulação, não tendo tempo suficiente para que sejam entregues ao destinatário. Nós maliciosos não são nem fonte nem destinatário das mensagens. O tamanho das mensagens é de 1 MB. O tempo de vida das mensagens nunca expira durante as simulações. Com relação ao tamanho do *buffer*, este foi configurado para 20 MB. O número de nós maliciosos varia de 0 a 5 com o objetivo de mensurar o efeito da adição gradual de nós maliciosos na rede.

Tabela 5.1: Parâmetros de simulação.

Parâmetro de Simulação	**Valor Utilizado**
Número de Mensagens	1.000
Tamanho das Mensagens (MB)	1
Taxa de Transmissão (Mbps)	1
TTL	∞
Tamanho do *Buffer* (MB)	20
Número de Nós Maliciosos	0$\sim$5

5.4 Resultados Obtidos

Para a avaliar os efeitos que o ataque de falsificação de reconhecimentos positivos causa em uma DTN, são utilizadas as métricas taxa de entrega de mensagens, sobrecarga de transmissão de mensagens e o atraso de entrega de mensagens. Para os resultados são apresentadas as médias de 10 rodadas distintas de simulação. Um intervalo de confiança de 95% é exibido por barras verticais.

Os resultados para os protocolos Epidêmico, MaxProp, Prophet e *Spray and Wait* foram escolhidos para serem apresentados nesta seção em razão dos diferentes modos de

operação destes protocolos, que podem representar uma grande quantidade de protocolos existentes na literatura. Além disso, somente são exibidos neste capítulo os resultados para os cenários Dieselnet e Rollernet. Os demais resultados foram omitidos com o objetivo de evitar a repetição excessiva, visto que são similares aos resultados exibidos neste capítulo. Estes resultados omitidos podem ser consultados no Apêndice B.

As imagens desta avaliação, apresentam, ao menos que seja estabelecido o contrário, a seguinte estrutura. O eixo Y ilustra o resultado alcançado para a métrica em questão. Ou seja, a taxa de entrega de mensagens, o atraso médio em segundos ou a sobrecarga da rede. Uma linha horizontal sólida representa o desempenho do protocolo em que está sendo avaliado quando utiliza reconhecimentos positivos no cenário ilustrado pela figura em questão. Uma linha horizontal pontilhada representa o desempenho do protocolo que está sendo avaliado quando não utiliza reconhecimentos positivos no cenário ilustrado pela figura em questão. Cabe dizer que ambas as linhas são rotuladas "Com Acks" e "Sem Acks", respectivamente. Em ambos os casos, os resultados foram extraídos do Capítulo 4 e portanto, não consideram a ação de nós maliciosos. O objetivo é avaliar se o ataque de falsificação de reconhecimentos positivos leva o desempenho da rede a patamares inferiores de quando não utiliza reconhecimentos positivos. Desta forma, é possível concluir se em ambientes sem autenticação é mais apropriado não utilizar ACKs a usá-los.

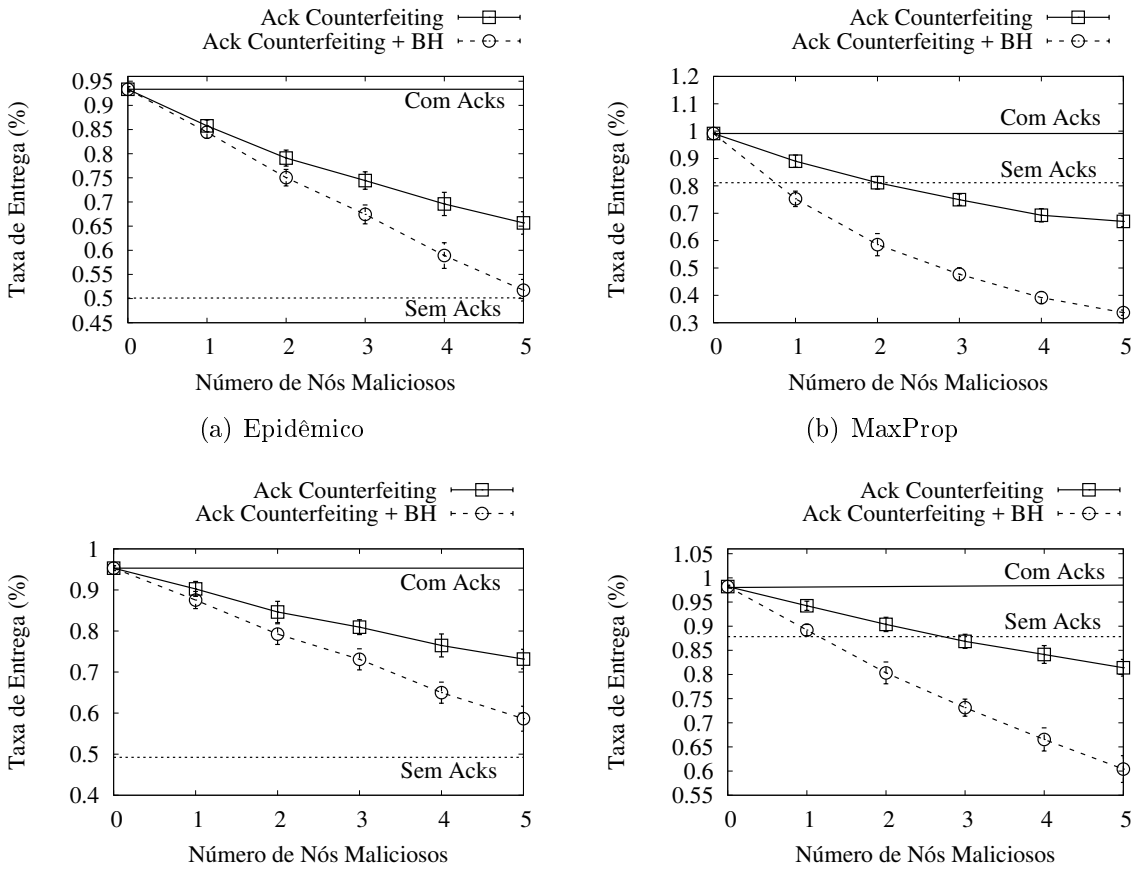
Com relação aos ataques avaliados, as figuras estão padronizadas como a seguir. O desempenho do protocolo que sofre a ação do ataque de falsificação de reconhecimentos positivos é ilustrado por uma curva de linha sólida cujos pontos são quadrados. Por sua vez, o desempenho do protocolo quando sofre a ação do ataque de falsificação de reconhecimentos positivos com buraco negro é ilustrado por uma curva de linha pontilhada cujos pontos são círculos. O eixo X ilustra a quantidade de nós maliciosos no cenário, variando de 0 a 5.

Finalmente, é importante salientar que para a exibição destes resultados, adotou-se como nomenclatura os seguintes termos. ***Ack Counterfeiting*** para o modelo de ataque de falsificação de reconhecimentos positivos e ***Ack Counterfeiting + BH*** para o modelo de falsificação de reconhecimentos positivos com buraco negro.

5.4.1 Taxa de Entrega

A Figura 5.1 ilustra a taxa de entrega obtida no cenário Rollernet para os protocolos Epidêmico (5.1(a)), MaxProp (5.1(b)), Prophet (5.1(c)) e *Spray and Wait* (5.1(d)).

Para o cenário Rollernet, é possível observar que o ataque de falsificação de reconhecimentos positivos e o ataque de falsificação de reconhecimentos positivos com buraco negro diminuem a taxa de entrega para todos os protocolos avaliados. Isto acontece porque os nós maliciosos enganam os nós legítimos com reconhecimentos positivos falsificados. Logo, mensagens que não foram entregues ao destinatário são removidas dos *buffers* dos nós, o que diminui a disponibilidade destas mensagens na rede. Em um caso extremo, mensagens que não chegaram do destinatário podem ser removidas completamente da rede. Evidentemente, isto prejudica a taxa de entrega da rede.



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 5.1: Taxa de entrega para o cenário Rollernet.

Outra observação que deve ser feita é que o ataque de falsificação de reconhecimentos positivos com buraco negro é mais prejudicial à rede do que o ataque sem buraco negro, principalmente a partir de dois nós maliciosos na rede. Este é um resultado intuitivo, visto que a utilização de buraco negro torna o ataque mais agressivo, expurgando de modo mais rápido as mensagens da rede, com consequente deterioração do desempenho da rede.

Como forma de avaliar o impacto dos ataques na rede, duas métricas são introduzidas. A Degradação Absoluta (D_A) representa a quantidade percentual absoluto de degradação

Tabela 5.2: Impacto do ataque de falsificação de reconhecimentos positivos na taxa de entrega do cenário Rollernet.

	Degradação Absoluta (%)					Degradação Relativa (%)				
Protocolos/Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	-7,5	-14,25	-18,90	-23,75	-27,68	-8,12	-15,26	-20,24	-25,44	-29,65
Life	-7,19	-14,13	-19,11	-23,63	-28,22	-7,69	-15,12	-20,45	-25,29	-30,20
MaxProp	-10,11	-17,96	-24,18	-29,86	-32,11	-10,19	-18,11	-24,39	-30,12	-32,39
Prophet	-5,30	-10,68	-14,38	-18,83	-22,15	-5,30	-11,20	-15,08	-19,75	-23,23
ProphetV2	-5,91	-10,11	-13,20	-16,16	-19,07	-5,91	-10,33	-13,49	-16,52	-19,49
Spray and Wait	-3,98	-7,84	-11,34	-14,06	-16,81	-4,05	-7,98	-11,54	-14,31	-17,11
Wave	-5,08	-10,51	-14,50	-20,04	-23,04	-5,69	-11,77	-16,24	-22,45	-25,81

na taxa de entrega, ocasionada pelos ataques avaliados nesta seção. Por outro lado, a Degradação Relativa (D_R) representa a proporção da degradação do desempenho. Ambas são calculadas similarmente às melhorias absolutas e relativas apresentadas na Seção 4.4.1. Por questões de explicitude o cálculo de D_A e D_R são apresentados nas Fórmulas 5.1 e 5.2, respectivamente. Nestas equações, $Delivery_{malicious}$ representa o desempenho do protocolo quando um dos ataques avaliados é executado. Por sua vez, $Delivery_{ack}$ representa o desempenho do protocolo com a utilização de reconhecimentos positivos, mas quando não existem nós maliciosos na rede.

$$D_A = (Delivery_{malicious} - Delivery_{ack}) \times 100 \tag{5.1}$$

$$D_R = \left(\frac{Delivery_{malicious} \times 100}{Delivery_{ack}} \right) - 100 \tag{5.2}$$

A Tabela 5.2 mostra a degradação absoluta e a degradação relativa ocasionada pelo ataque de falsificação de reconhecimentos positivos para o cenário Rollernet.

Destaca-se que as degradações absolutas e relativas podem chegar a 30%, quando 5 nós maliciosos estão na rede. Além disso, destaca-se que o protocolo MaxProp é o protocolo mais afetado pelo ataque de falsificação de reconhecimentos positivos, sofrendo uma degradação de até 32% na taxa de entrega. Isto se deve ao fato de o protocolo MaxProp fazer um maior controle da quantidade de réplicas através de seus mecanismos de controle de congestionamento. Logo, como mantém um menor número de réplicas na rede, a remoção de mensagens maliciosamente da rede e resulta em um maior prejuízo ao desempenho deste protocolo.

Contraditoriamente ao argumento supracitado, o protocolo *Spray and Wait* é o protocolo menos prejudicado neste cenário, apesar de manter um controle rígido do número de réplicas na rede. Isto se deve ao fato de que os nós maliciosos só criam reconhecimentos

Tabela 5.3: Impacto do ataque de falsificação de reconhecimentos positivos com buraco negro na taxa de entrega do cenário Rollernet.

	Degradação Absoluta (%)					Degradação Relativa (%)				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	-8,90	-18,31	-25,91	-34,43	-41,61	-9,53	-19,61	-27,75	-36,88	-44,57
Life	-8,84	-19,03	-26,57	-35,73	-42,16	-9,46	-20,36	-28,43	-38,24	-45,12
MaxProp	-23,87	-40,60	-51,41	-59,94	-65,45	-24,07	-40,95	-51,86	-60,46	-66,02
Prophet	-7,77	-16,07	-22,21	-30,33	-36,67	-8,15	-16,86	-23,30	-31,82	-38,47
ProphetV2	-10,33	-19,99	-27,99	-36,02	-42,49	-10,56	-20,43	-28,61	-36,83	-43,44
Spray and Wait	-9,01	-17,87	-25,07	-31,65	-37,78	-9,17	-18,19	-25,52	-32,23	-38,47
Wave	-15,65	-30,74	-40,54	-49,49	-54,92	-17,53	-34,44	-45,42	-55,45	-61,54

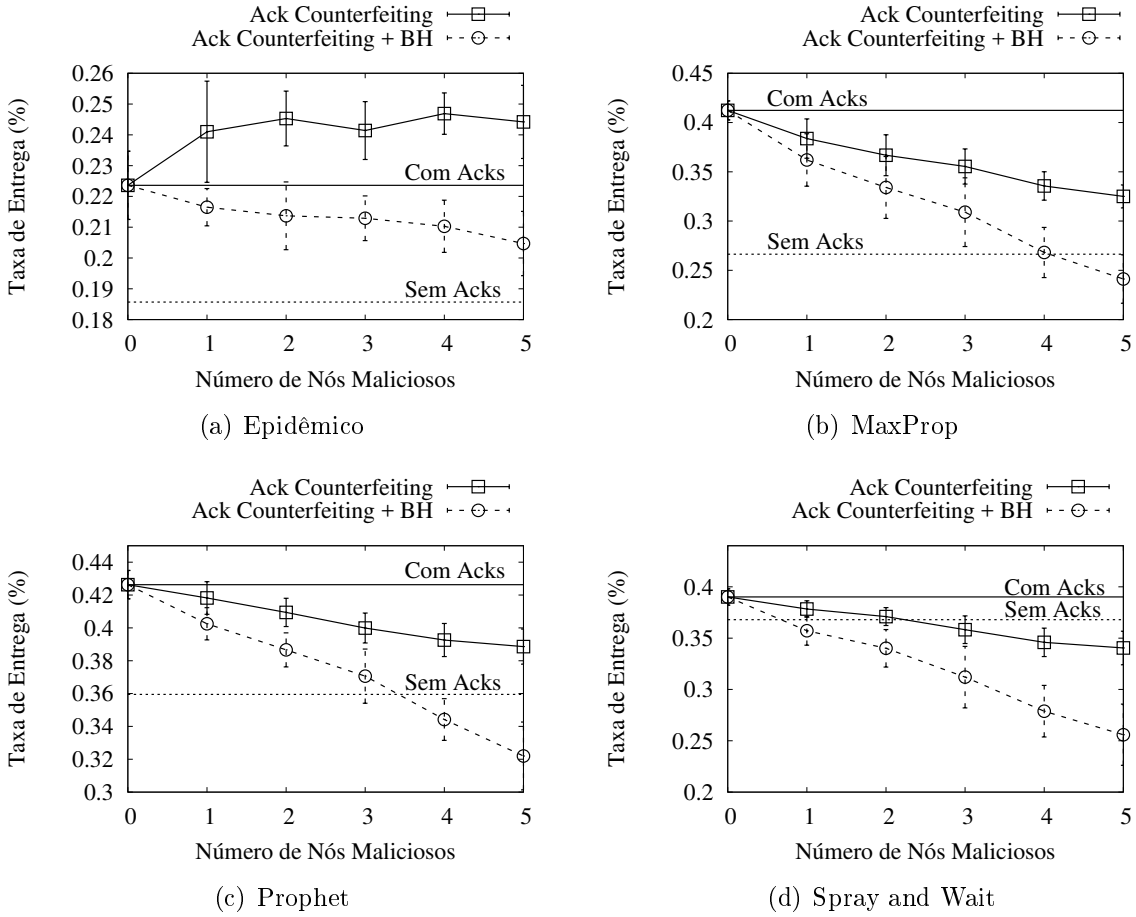
positivos falsificados para mensagens que recebem. A limitação do número de réplicas do protocolo *Spray and Wait* resulta em algumas mensagens nunca serem enviadas aos nós maliciosos. Além disso, cabe destacar que algumas mensagens podem entrar na fase de espera do protocolo (*wait*) antes mesmo de alcançarem qualquer nó malicioso. Logo, reconhecimentos positivos nunca são forjados para estas mensagens, tornando o protocolo mais resiliente que os demais.

Por último, observamos que os ataques levam os protocolos MaxProp e *Spray and Wait* a níveis de taxa de entrega que são alcançados sem a utilização de reconhecimentos positivos. Isto pode ser um argumento contra a utilização de reconhecimentos positivos na ausência de mecanismos que mitiguem eficientemente este ataque.

A Figura 5.2 ilustra a taxa de entrega obtida no cenário Dieselnet. É importante ressaltar o resultado contraintuitivo alcançado pelo protocolo Epidêmico neste conjunto de mobilidade, como ilustrado na Figura 5.2(a). Observa-se que o ataque de falsificação de reconhecimentos positivos leva a uma melhoria no desempenho do protocolo Epidêmico neste cenário. Isto acontece porque este protocolo realiza a replicação sem nenhum controle, inundando a rede de mensagens e esgotando seus recursos rapidamente. Com a adição de nós maliciosos na rede, estes passam a enviar reconhecimentos positivos falsificados com o objetivo de prejudicar o desempenho da rede. No entanto, visto que a rede está saturada devido à replicação do protocolo Epidêmico, estes reconhecimentos positivos falsificados resultam na diminuição do congestionamento, ou seja, os nós maliciosos mitigam involuntariamente o congestionamento na rede.

É importante destacar que dos protocolos avaliados neste cenário, além do protocolo Epidêmico, somente os protocolos *Life* e *Wave* apresentaram este resultado preliminarmente contraintuitivo. Ressalta-se que estes protocolos, Epidêmico, *Life* e *Wave* são os únicos que alcançam desempenho relativo a taxa de entrega inferior a 30%. Isto corrobora com a conclusão que os nós maliciosos atuam involuntariamente reduzindo o congestio-

namento da rede, nestes cenários. Além disso, este comportamento foi verificado somente para os cenários Dieselnet e Infocom, classificados como os conjuntos de mobilidade menos conectados na avaliação feita na Seção 4.2, fortalecendo ainda mais a ideia anterior.



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 5.2: Taxa de entrega para o cenário Dieselnet.

Em razão destes resultados existe a possibilidade de argumentação a favor da replicação como forma de tornar a rede mais robusta contra o ataque de falsificação de reconhecimentos positivos. No entanto, o desempenho da rede na ausência de nós maliciosos seria prejudicado, visto que os protocolos que controlam menos o congestionamento resultam em pior desempenho, como pode ser observado neste capítulo e também nos resultados do Capítulo 4.

As Tabelas 5.4 e 5.5 listam a degradação absoluta e relativa para o cenário Dieselnet, respectivamente. É possível observar novamente que o protocolo mais afetado pelos ataques é o protocolo MaxProp. Destaca-se novamente que o ataque é menos eficiente neste cenário em razão do congestionamento, onde algumas vezes os nós maliciosos acabam contribuindo para melhorar o desempenho da rede, além da baixa conectividade, que resulta em uma menor quantidade de reconhecimentos positivos falsificados na rede.

Tabela 5.4: Impacto do ataque de falsificação de reconhecimentos positivos na taxa de entrega do cenário Dieselnet.

	Degradação Absoluta (%)					Degradação Relativa (%)				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	1.74	2.17	1.78	2.33	2.06	7.78	9.70	7.96	10.42	9.21
Life	0.60	0.77	0.99	1.37	1.37	2.34	3.00	3.86	5.34	5.34
MaxProp	-2.86	-4.55	-5.69	-7.67	-8.73	-6.94	-11.04	-13.80	-18.60	-21.17
Prophet	-0.81	-1.69	-2.64	-3.37	-3.77	-1.90	-3.96	-6.19	-7.91	-8.84
ProphetV2	-0.38	-0.97	-1.65	-2.41	-3.20	-0.90	-2.29	-3.90	-5.69	-7.56
Spray and Wait	-1.17	-1.91	-3.19	-4.42	-4.97	-3.00	-4.90	-8.18	-11.33	-12.74
Wave	3.17	3.72	3.87	3.74	4.06	14.59	17.12	17.81	17.21	18.68

Tabela 5.5: Impacto do ataque de falsificação de reconhecimentos positivos com buraco negro na taxa de entrega do cenário Dieselnet.

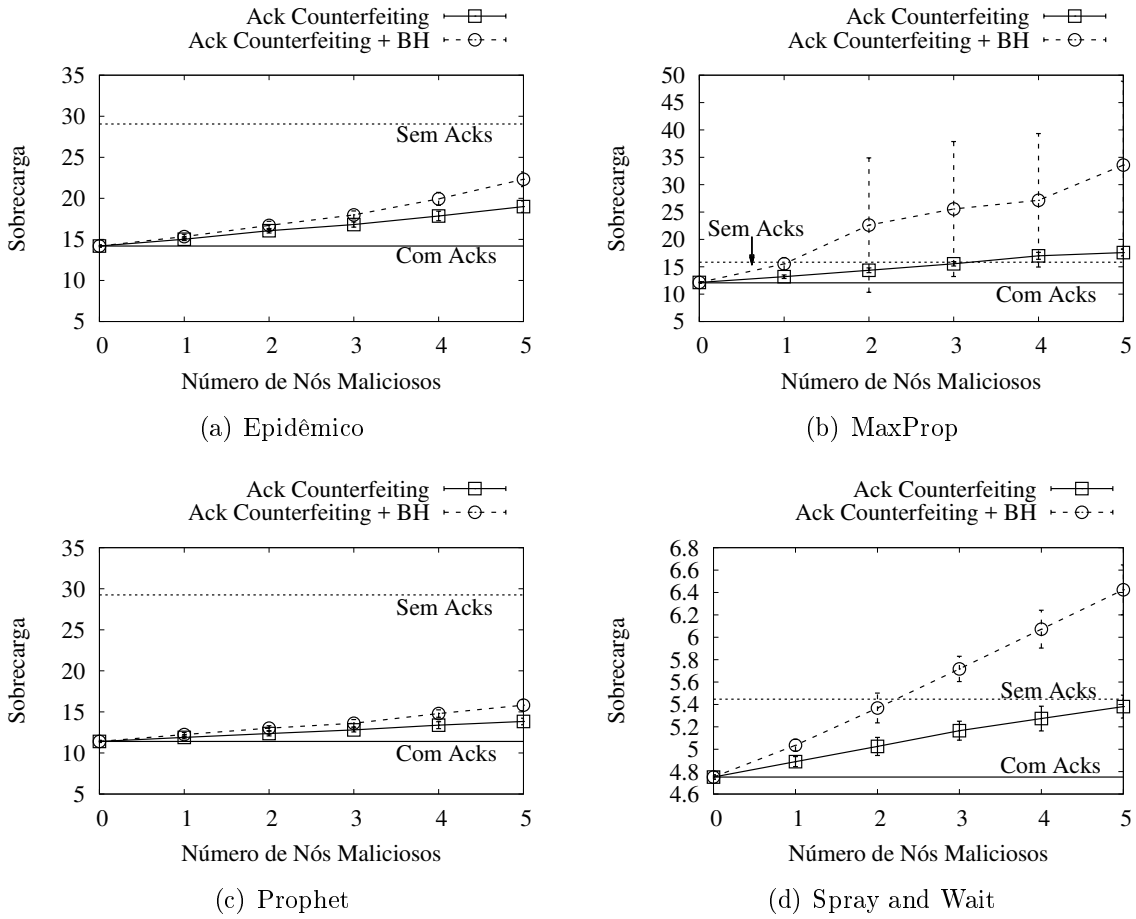
	Degradação Absoluta (%)					Degradação Relativa (%)				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	-0.71	-0.99	-1.07	-1.33	-1.89	-3.18	-4.43	-4.79	-5.95	-8.45
Life	-1.55	-1.44	-2.68	-2.76	-3.64	-6.04	-5.61	-10.44	-10.75	-14.18
MaxProp	-5.03	-7.82	-10.33	-14.42	-17.11	-12.20	-18.97	-25.05	-34.97	-41.50
Prophet	-2.38	-3.97	-5.57	-8.21	-10.43	-5.58	-9.31	-13.07	-19.26	-24.47
ProphetV2	-2.33	-3.57	-5.29	-7.89	-10.18	-5.50	-8.43	-12.49	-18.63	-24.04
Spray and Wait	-3.29	-5.00	-7.81	-11.13	-13.42	-8.43	-12.82	-20.02	-28.53	-34.40
Wave	1.35	1.92	1.46	0.47	0.08	6.21	8.84	6.72	2.16	0.37

5.4.2 Sobrecarga

Esta seção trata da avaliação da sobrecarga resultante da execução dos ataques de falsificação de reconhecimentos positivos e falsificação de reconhecimentos positivos com buraco negro, para os cenários avaliados.

A Figura 5.3 ilustra a sobrecarga para o cenário Rollernet. É possível observar que os ataques resultam em uma maior sobrecarga para todos os protocolos no cenário Rollernet. Outra observação é que o ataque com buraco negro tem mais impacto negativo do que o ataque sem buraco negro. Novamente o ataque com buraco negro causa mais danos ao desempenho com relação a esta métrica devido a seu modo de operação mais agressivo, removendo uma maior quantidade de mensagens da rede.

A Figura 5.4 ilustra a sobrecarga para o cenário Dieselnet. Para os protocolos Epidêmico e Prophet, menos restritivos com relação à replicação de mensagens, a sobrecarga da rede quando um ataque de falsificação de reconhecimentos positivos está sendo executado mantém-se similar a sobrecarga do protocolo com a utilização de reconhecimentos positivos mas sem nós maliciosos na rede. Associa-se este fato ao alto congestionamento resultante destes protocolos. Por outro lado, o ataque de falsificação de reconhecimentos positivos com buraco negro levou a um aumento na sobrecarga para estes dois protocolos. Como este ataque remove um maior número de mensagens da rede, em alguns casos os nós acabam encaminhando e replicando as mesmas mensagens uma maior quantidade de



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

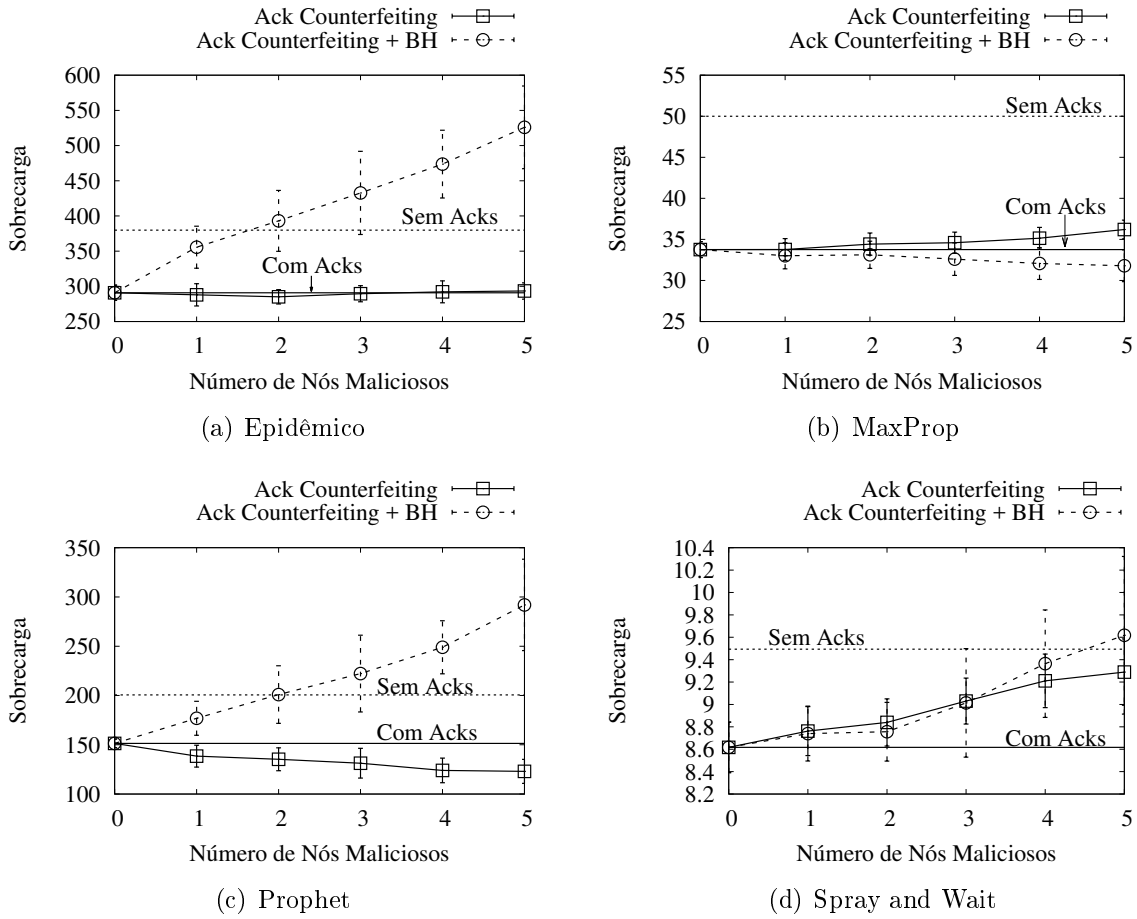
Figura 5.3: Sobrecarga para o cenário Rollernet.

vezes, levando a um aumento na sobrecarga da rede.

Para o protocolo *Spray and Wait*, ambos os ataques resultaram em um aumento da sobrecarga, levando-a a níveis similares a não utilização de reconhecimentos positivos. No entanto, ressalta-se que este aumento é muito menor que os demais, em termos absolutos, sendo de aproximadamente 1 mensagem.

O aumento na sobrecarga consequente da realização de ataques na rede é reflexo da queda no desempenho com relação a taxa de entrega. A rede entrega um menor número de mensagens e como a sobrecarga é uma razão da quantidade de mensagens entregues, ela tende a aumentar, se for considerado que a quantidade de encaminhamentos/replicações continua a mesma.

Embora exista a tendência do aumento da sobrecarga em razão da diminuição da taxa de entrega, isto não é observado para o protocolo MaxProp, como pode ser observado na Figura 5.4(b). Para este protocolo, no cenário Dieselnet, a sobrecarga mantém-se similar a quando reconhecimentos positivos são utilizados mas não existem nós maliciosos na rede.



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 5.4: Sobrecarga para o cenário Dieselnet.

Isto ocorre porque embora a taxa de entrega diminua, a quantidade de encaminhamentos também o faz. Este comportamento é decorrente do modo como o protocolo MaxProp controla o congestionamento na rede. Nele, um nó A não envia uma mensagem M novamente para um nó B, se B já recebeu M e já a descartou. Sendo assim, se B for um nó malicioso, ele não continua a receber a mensagem de A, o que diminui o impacto do ataque. Para efeito de comparação, se estivessem utilizando o protocolo Epidêmico, A enviaria a mensagem M para B quantas vezes fosse possível, durante a operação da rede.

Portanto, ao evitar enviar excessivamente uma mesma mensagem M para um mesmo nó N, o protocolo MaxProp possui um grau de robustez a mais contra o ataque de buraco negro em DTNs.

5.4.3 Atraso de Entrega

A Figura 5.5 ilustra os resultados de atraso de entrega para o cenário Rollernet. Nela, é possível observar que, de maneira geral, os ataques diminuem o atraso de entrega

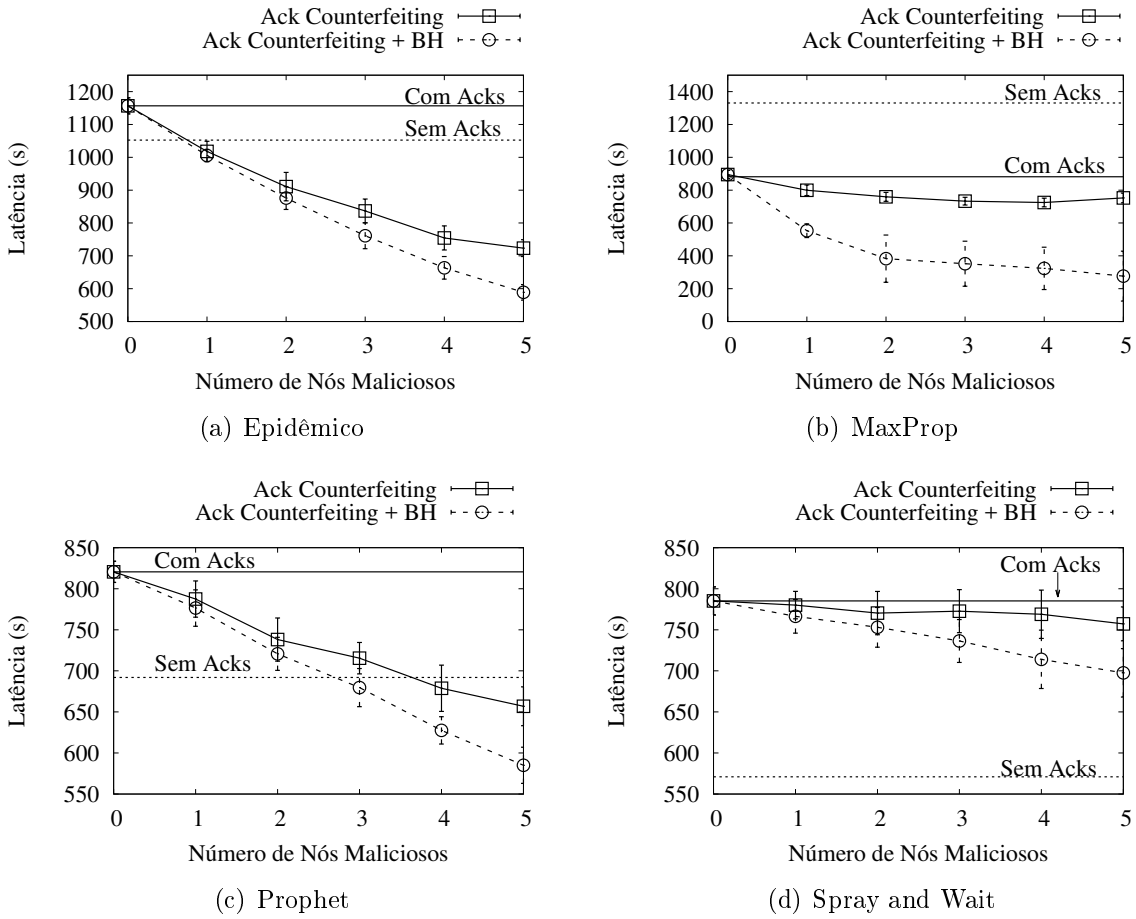
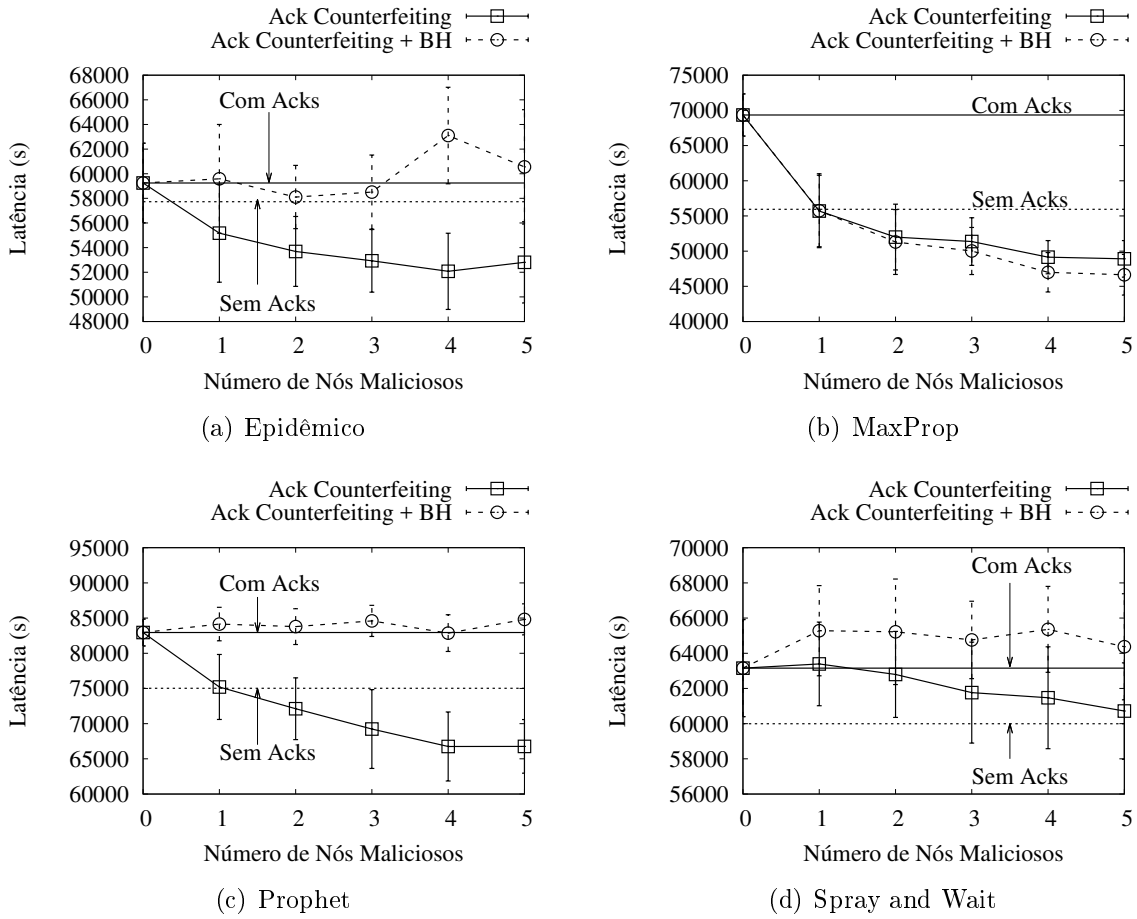


Figura 5.5: Atraso de entrega para o cenário Rollernet.

de mensagens a valores inferiores de quando os protocolos não utilizam reconhecimentos positivos e não estão sujeitos aos ataques. Isto ocorre porque mensagens são removidas dos *buffers* dos nós, resultando em um menor congestionamento. Além disso, as mensagens que alcançam o destino mais rápido são menos suscetíveis aos ataques. Ao entregar um número menor de mensagens, que não por acaso são as que chegam mais rápido ao destino, o atraso de entrega médio cai para estes cenários.

Por sua vez, os resultados para o cenário Dieselnet são apresentados na Figura 5.6. Observa-se que para os protocolos Epidêmico, Prophet e *Spray and Wait*, o ataque de falsificação de reconhecimentos positivos com buraco negro mantém o atraso de entrega oscilando próximo ou acima do atraso de entrega alcançado quando os protocolos utilizam reconhecimentos positivos mas não são alvo de ataques. Por outro lado, quando alvo do ataque de falsificação de reconhecimentos positivos, o atraso de entrega diminui para todos os protocolos.

Portanto, temos neste cenário que: (1) enquanto o ataque de falsificação de reconhecimentos positivos com buraco negro atrasa a entrega das mensagens pois os nós maliciosos



(a) Epidêmico (b) MaxProp

(c) Prophet (d) Spray and Wait

Figura 5.6: Atraso de entrega para o cenário Dieselnet.

removem estas mensagens imediatamente de seus *buffers*; (2) no ataque de falsificação de reconhecimentos positivos os nós maliciosos mantêm a mensagem em *buffer*, contribuindo eventualmente para a entrega desta mensagem ao destino. Ou seja, apesar da redução da taxa de entrega estar algumas vezes associada à diminuição do atraso de entrega, como anteriormente citado, o ataque de falsificação de reconhecimentos positivos com buraco negro aumenta o tempo que as mensagens levam para serem entregues, visto que agem reduzindo a disponibilidade destas mensagens na rede. Isto contribui para o aumento ou manutenção da média de atraso.

5.5 Conclusões

Os resultados demonstraram que o desempenho dos protocolos de roteamento com relação às métricas de taxa de entrega e atraso de entrega é comprometido pelo ataque de falsificação de reconhecimentos positivos. Além disso, o ataque de falsificação de reconhecimentos positivos com buraco negro é potencialmente mais prejudicial, pois atua

mais agressivamente reduzindo a disponibilidade de cópias de mensagens na rede.

Especificamente tratando-se da taxa de entrega, observou-se que os cenários mais conectados, como é o caso dos cenários Shopping, Rollernet e Infocom, são mais prejudicados pelo ataque. Isto decorre do fato de que os nós maliciosos precisam receber as mensagens para que possam forjar os reconhecimentos positivos. Logo, em cenários mais conectados, eles receberão um maior número de mensagens, o que expande os limites do ataque. No entanto, mesmo nos cenários menos conectados, como é o caso do cenário Dieselnet, o ataque de falsificação de reconhecimentos positivos com buraco negro reduziu a taxa de entrega em até 21%.

Com relação aos protocolos, observou-se que o protocolo MaxProp é o protocolo que mais sofre degradação de desempenho. Isto acontece porque este protocolo faz um controle de replicação mais restrito que os demais. Logo, com menos réplicas na rede, este protocolo sofre maiores prejuízos quando réplicas de mensagens são removidas, seja por ACKs falsificados ou através de buracos negros. Excepcionalmente, a taxa de entrega alcançada pelo protocolo *Spray and Wait* é a menos prejudicada pelos ataques. Isto acontece porque com este protocolo a probabilidade de que réplicas das mensagens cheguem aos nós maliciosos é menor, diminuindo a quantidade de ACKs forjados. Como consequência, a efetividade do ataque também diminui.

Tratando-se do atraso de entrega, observou-se que o ataque de falsificação de reconhecimentos positivos com buraco negro pode levar um aumento do atraso de entrega médio, visto que reduzem imediatamente a disponibilidade das mensagens e não contribuem com o encaminhamento das mensagens que recebem. Por outro lado, o ataque de falsificação de reconhecimentos positivos pode reduzir o atraso de entrega, visto que as mensagens que permanecem mais tempo na rede sem alcançarem o destinatário estão mais suscetíveis a este ataque.

Por fim, é importante ressaltar que em alguns casos os ataques levam o desempenho da rede a níveis inferiores a quando reconhecimentos positivos não são utilizados. Isto pode ser um argumento para a rede não utilizar reconhecimentos positivos na ausência de contramedidas eficientes. Destaca-se então a necessidade de projeto e implementação de contramedidas eficientes contra o ataque de falsificação de reconhecimentos positivos.

Este trabalho propõe e avalia duas contramedidas contra o ataque de falsificação de reconhecimentos positivos em DTNs. No próximo capítulo, estas propostas são apresentadas a avaliadas.

Capítulo 6

Proposta e Avaliação de Contramedidas

A avaliação do ataque de falsificação de reconhecimentos positivos realizada no Capítulo 5 mostra que este ataque é potencialmente prejudicial para rede. Neste sentido, é importante o projeto e desenvolvimento de contramedidas que mitiguem os efeitos negativos deste ataque.

Apesar dos resultados apresentados por Burgess *et al.* [12] para a contramedida BRG, apresentada na Seção 3.6, a premissa feita por Burgess *et al.* pode não ser verdade em vários cenários, quando as mensagens e seus ACKs trafegam por caminhos diferentes em razão da mobilidade dos nós. A Figura 6.1 ilustra o problema do nó malicioso intermediário. Neste cenário, existem dois nós legítimos A e B, rodando a contramedida BRG. Além disso, existe um nó malicioso C. Quatro contatos ocorrem. Estes contatos são representados por setas. A ordem cronológica dos contatos é dada pelos números posicionados acima das setas. O primeiro contato acontece entre os nós A e B. Durante este contato, A encaminha a mensagem m_1 para B. O destinatário da mensagem m_1 é um nó que não está ilustrado na Figura 6.1. Após o primeiro contato, outro contato acontece, agora entre os nós B e C. Durante este contato, B envia a mesma mensagem m_1 recebida de A para o nó C. O terceiro contato é novamente entre o nó legítimo B e C. Neste contato, C envia um ACK falsificado $ack(m_1)$ para B, sugerindo que m_1 foi entregue ao destinatário. Assim que B recebe $ack(m_1)$ a mensagem m_1 é então removida de seu *buffer*, visto que a regra proposta pela contramedida BRG é satisfeita: B recebeu m_1 previamente e então deve considerar o ACK $ack(m_1)$ como legítimo. O último contato acontece entre os nós legítimos A e B. Durante este contato, o nó B encaminha o ACK falso $ack(m_1)$ para A. Por sua vez, A remove m_1 de seu *buffer*. Logo, a mensagem m_1 foi completamente removida da rede por um nó malicioso utilizando o ataque de falsificação de reconhecimentos positivos, mesmo quando os nós legítimos estão utilizando a contramedida BRG.

É importante destacar que o contrário também é válido. Isto é, a contramedida BRG pode considerar que um ACK legítimo é falsificado, se este ACK estiver trafegando um caminho diferente ao trafegado pela mensagem. Dada a mobilidade dos nós, este é um cenário possível. Esta característica torna a disseminação de ACKs legítimos mais lenta em redes que utilizam a contramedida BRG.

O objetivo deste capítulo é propor e avaliar contramedidas que mitiguem os efeitos negativos do ataque de falsificação de reconhecimentos positivos em DTNs, além de estender a avaliação da contramedida existente na literatura, isto é, BRG.

As seções subsequentes estão dividias da seguinte forma. A Seção 6.1 descreve a contramedida DRAC, posteriormente, a Seção 6.1.1 apresenta uma variante da contramedida DRAC, denominada DRAC-SF. Todos os detalhes refentes a validação destas duas contramedidas são abordados na Seção 6.2. Os resultados obtidos são apresentados na Seção 6.3. Finalmente, as conclusões são apresentadas na Seção 6.4.

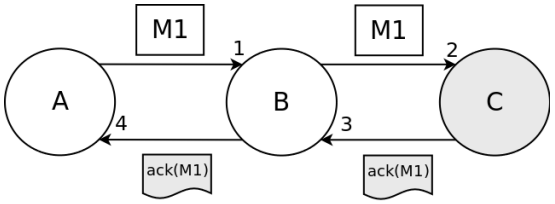


Figura 6.1: O problema do nó malicioso intermediário: após quatro contatos a mensagem m_1 é removida da rede sem ser entregue ao destinatário. A e B são nós legítimos rodando a contramedida BRG.

6.1 A Contramedida DRAC

Nesta seção, uma nova contramedida para lidar com o ataque de falsificação de reconhecimentos positivos em DTNs é proposta. Nomeada *Drop Acknowledged Messages First* (DRAC) esta contramedida consiste em não descartar imediatamente as mensagens para as quais ACKs tenham sido recebidos [76]. Ao invés disso, a proposta reordena a fila de descarte do nó, isto é, maior prioridade de descarte é dada as mensagens para as quais ACKs tenham sido recebidos. Desta forma, a contramedida DRAC evita descartar mensagens que ainda não chegaram ao destino, fornecendo a elas outras oportunidade de replicação com o objetivo de manter a taxa de entrega de mensagens dentro de uma taxa satisfatória.

Ressalta-se que a DRAC não é suscetível ao problema do nó malicioso intermediário, ilustrado na Figura 6.1. A proposta funciona como se segue. Quando um ACK é recebido

por um nó, ao invés de subitamente descartar a mensagem para a qual o ACK foi recebido, o nó somente realoca esta mensagem em sua fila de descarte. Deste modo, sempre que não existe espaço suficiente para receber uma nova mensagem, a mensagem a ser descartada é escolhida entre aquelas para as quais reconhecimentos já foram recebidos. De fato, a mensagem a ser descartada é selecionada pela política de gerenciamento de *buffer* em uso pelo protocolo de roteamento. A proposta neste trabalho utiliza a política de descarte *Least-Recently Forwarded* (LRF), que demonstrou-se eficiente na avaliação feita por Naves *et al.* [78]. Entre os benefícios da LRF estão o uso de somente informações locais, possibilitando a implementação desta política de gerenciamento de *buffer* por qualquer protocolo de roteamento.

Ao esperar pelo estouro do *buffer* ao invés de imediatamente descartar as mensagens para as quais reconhecimentos foram recebidos, a DRAC visa evitar remover dos *buffers* dos nós mensagens que não foram entregues ao destinatário, mas que para as quais ACKs foram forjados por nós maliciosos. Isto permite que esta mensagem continue a ser encaminhada e, principalmente, permite que seja entregue ao destino final. Se o nó em questão não possui reconhecimentos para as mensagens que ele possui armazenadas em *buffer*, uma mensagem é escolhida entre todas as que estão armazenadas de acordo com a política de gerenciamento de *buffer* utilizada.

A Figura 6.2 apresenta o funcionamento da contramedida DRAC. Nesta figura, as subfiguras são rotuladas por números que representam a ordem cronológica dos eventos. Na figura, os círculos representam nós na rede. A e B são nós legítimos e o nó C é um nó malicioso. As regiões cinzas abaixo de cada círculo representam os *buffers* dos nós e as setas acima dos mesmos círculos representam a prioridade de descarte atribuída para as mensagens naquele nó, seguindo a política de descarte *First In First Out* (FIFO). No início, como apresentado na Figura 6.2(a), o nó A possui as mensagens $M1$, $M2$ e $M4$ em seu *buffer*, por sua vez, o nó B possui as mensagens $M1$ e $M3$. Na Figura 6.2(b), um contato entre A e B ocorre. Neste contato A envia para B a mensagem $M2$. Neste caso, como $M2$ é a última mensagem recebida por B, ela encontra-se no final da fila de descarte, como pode ser observado na Figura 6.2(c). Um contato entre B e C é mostrado na Figura 6.2(d). Neste contato, B envia a mensagem $M2$ para o nó C. Como o nó C é malicioso, ele cria um reconhecimento positivo falsificado para a mensagem $M2$ e envia para B. Este passo é ilustrado na Figura 6.2(e). Como B está rodando a contramedida DRAC, ao receber o reconhecimento positivo falsificado, ao invés de descartar a mensagem $M2$ imediatamente, o nó B apenas reordena sua prioridade de descarte. B posiciona a mensagem $M2$ na cabeça da fila de descarte. Estes passos são ilustrados nas

Figuras 6.2(f) e 6.2(g). Ao optar por não descartar a mensagem $M2$, quando recebe um reconhecimento positivo, B pode enviar esta mensagem para outros nós em contatos futuros, aumentando a robustez da rede contra o ataque de falsificação de reconhecimentos positivos.

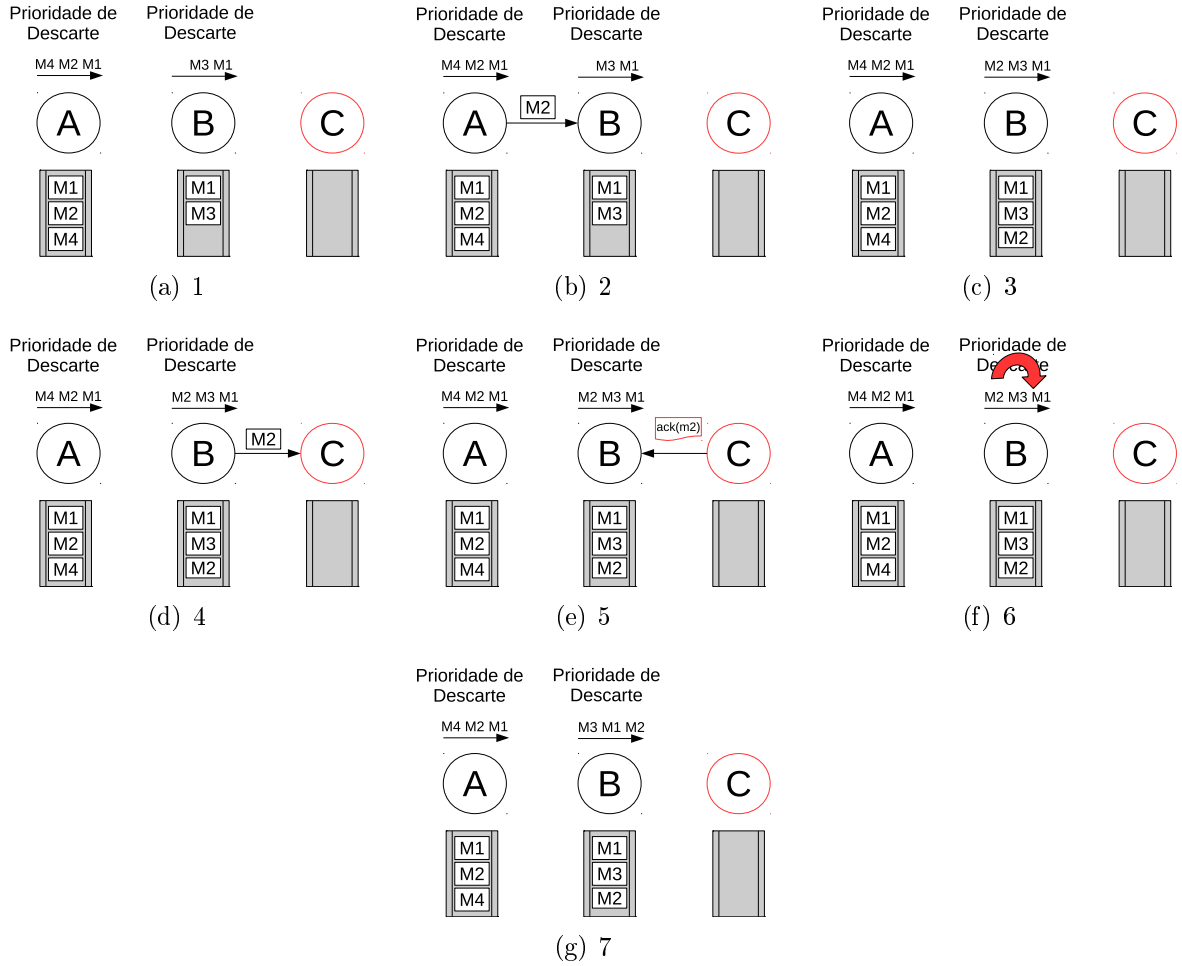


Figura 6.2: Representação conceitual do funcionamento da contramedida DRAC.

É importante ressaltar que a proposta deste trabalho não tenta identificar nós maliciosos que estão forjando e enviando ACKs. Então, DRAC não se baseia em nenhum mecanismo de autenticação, similar a contramedida BRG.

6.1.1 A Variante DRAC-SF

Uma variante da contramedida DRAC, denominada *Drop Acknowledged Messages First and Stop Forwarding* (DRAC-SF), também é proposta. Nesta variante, além de realocar na fila de descarte as mensagens para as quais reconhecimentos positivos foram recebidos, dando prioridade de descarte a estas mensagens, elas também deixam de ser

encaminhadas/replicadas pelos nós legítimos. No entanto, em caso de contato com o destinatário da mensagem, a mesma será encaminhada para o destino final.

O objetivo da DRAC-SF é evitar replicar mensagens que foram legitimamente reconhecidas, visto que replicar mensagens que já foram entregues ao destino diminui a eficiência da rede. No entanto, esta medida tem um custo para a rede, que também para de replicar mensagens que não chegaram ao destino, mas para as quais ACKs foram forjados.

6.2 Cenários de Avaliação

6.2.1 Protocolos de Roteamento e Registros de Mobilidade

Para a avaliação das contramedidas DRAC e DRAC-SF, são utilizados os mesmos protocolos de roteamento utilizados nos Capítulos 4 e 5, a saber, Epidêmico, *Life*, MaxProp, Prophet, ProphetV2, *Spray and Wait* e *Wave*. Estes protocolos são descritos em detalhes na Seção 3.1.

Com relação aos registros de mobilidade, são utilizados os conjuntos Dieselnet, Infocom, Rollernet e Shopping. Estes registros de mobilidade são descritos com maiores detalhes na Seção 4.2.

6.2.2 Ambiente de Simulação

As contramedidas DRAC e DRAC-SF, além da contramedida BRG, foram implementadas no simulador ONE, nos 7 protocolos apresentados previamente. As configurações dos cenários são as mesmas utilizadas na Seção 5.3 e foram sumarizadas na Tabela 5.1. Para dispensar a necessidade de referir-se sucessivamente às seções anteriores, este ambiente de simulação é novamente descrito a seguir.

Para todos os cenários, 1.000 mensagens são geradas durante cada rodada de simulação. A geração destas mensagens é finalizada cerca de 6 horas antes do final da rodada de simulação para os cenários Dieselnet, Infocom e Shopping. Para o cenário Rollernet, a geração termina cerca de 1 hora antes do final da duração do cenário. O período final de cerca de 1 hora para o cenário Rollernet e 6 horas para os cenários Dieselnet, Infocom e Shopping visa evitar que mensagens sejam geradas a pouco tempo do final da simulação, não tendo tempo suficiente para que sejam entregues ao destinatário. Nós maliciosos não são nem fonte nem destinatário das mensagens. O tamanho das mensagens é de 1 MB.

O tempo de vida das mensagens nunca expira durante as simulações. Com relação ao tamanho do *buffer*, este foi configurado para 20 MB. O número de nós maliciosos varia de 0 a 5 com o objetivo de mensurar o efeito da adição incremental de nós maliciosos na rede.

6.2.3 Modelos de Ataque

Nesta avaliação, são utilizados os dois modelos de ataque descritos na Seção 5.2.1, isto é, o ataque de falsificação de reconhecimentos positivos no qual o nó malicioso forja reconhecimentos para cada mensagem que recebe na rede e o ataque de falsificação de reconhecimentos positivos com buraco negro, no qual o nó maliciosos além de criar reconhecimentos falsificados para cada mensagem que recebe, também descarta estas mensagens de seu *buffer*.

6.3 Resultados

Para avaliação do desempenho das contramedidas DRAC e DRAC-SF, três métricas são utilizadas, isto é, taxa de entrega, atraso de entrega e sobrecarga. Os resultados apresentados foram obtidos através da média de 10 rodadas de simulação. O intervalo de confiança para um nível de 95% é calculado e representado por barras verticais para todas as médias apresentadas.

6.3.1 Taxa de Entrega

Esta seção apresenta os resultados obtidos para a métrica Taxa de Entrega. Reitera-se que a taxa de entrega é o percentual de mensagens criadas que chegaram ao destino durante a rodada de simulação. Para os gráficos desta seção, o eixo Y exibe a taxa de entrega e o eixo X apresenta o número de nós maliciosos na rede, variando de 0 a 5. Para todas as imagens, quatro curvas são exibidas. A curva com linha contínua e cujos pontos são apresentados por um quadrado é referente ao protocolo utilizado, isto é, Epidêmico, *Life*, MaxProp, Prophet, ProphetV2, *Spray and Wait* e *Wave*, sem a inclusão de uma contramedida. A curva com linha tracejada e cujos pontos são apresentados por um círculo é referente ao protocolo que está sendo utilizado com a inclusão da contramedida BRG. Por sua vez, a contramedida DRAC é apresentada por uma linha pontilhada com triângulos brancos como pontos. Finalmente, a contramedida DRAC-SF é ilustrada por

uma linhada tracejada-pontilhada com triângulos pretos como pontos.

Para cada um dos conjuntos de mobilidade são apresentadas 4 figuras, uma para cada um dos seguintes protocolos: Epidêmico, MaxProp, Prophet e *Spray and Wait*. Esta organização segue a concepção utilizada nas avaliações feitas nos Capítulos 4 e 5, isto é, estes protocolos representam uma pluralidade de protocolos existentes na literatura.

Para cada uma das figuras apresentadas, a sub-figura (a), posicionada ao lado esquerdo, exibe os resultados obtidos para o ataque de falsificação de reconhecimentos positivos. Por sua vez, a sub-figura (b), posicionada ao lado direito, exibe os resultados obtidos para o ataque de falsificação de reconhecimentos positivos com buraco negro. Além disso, com o objetivo de auxiliar na interpretação dos dados, tabelas contendo as diferenças entre o desempenho da proposta DRAC-SF e da contramedida BRG são apresentadas.

Ressalta-se que alguns resultados foram omitidos desta seção com o propósito de tornar esta avaliação mais concisa. No entanto, estes resultados são similares aos que são aqui analisados e podem ser observados no Apêndice C.

6.3.1.1 Cenário Rollernet

As Figuras 6.3 a 6.6 apresentam os resultados referentes a taxa de entrega para o cenário Rollernet. A Tabela 6.1 apresenta a diferença de desempenho entre DRAC-SF e BRG para o ataque de falsificação de reconhecimentos positivos e a Tabela 6.2 apresenta esta diferença para o ataque de falsificação de reconhecimentos positivos com buraco negro.

Tabela 6.1: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Rollernet e o ataque de falsificação de reconhecimentos positivos.

	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	7	6	6	6	9	10	8	7	8	8	11	14
Life	7	6	7	7	9	10	8	7	8	8	12	15
MaxProp	-1	4	8	11	15	18	-1	4	9	13	20	24
Prophet	4	4	4	5	6	7	4	4	5	6	8	9
ProphetV2	1	2	3	5	6	7	1	2	3	6	7	9
SnW	-7	-5	-2	1	3	5	-7	-5	-2	1	3	6
Wave	10	11	12	12	13	15	13	14	15	16	17	21

Para o protocolo Epidêmico, como é possível observar na Figura 6.3, a contramedida DRAC-SF obtém o melhor desempenho para ambos os modelos de ataque. Constata-se

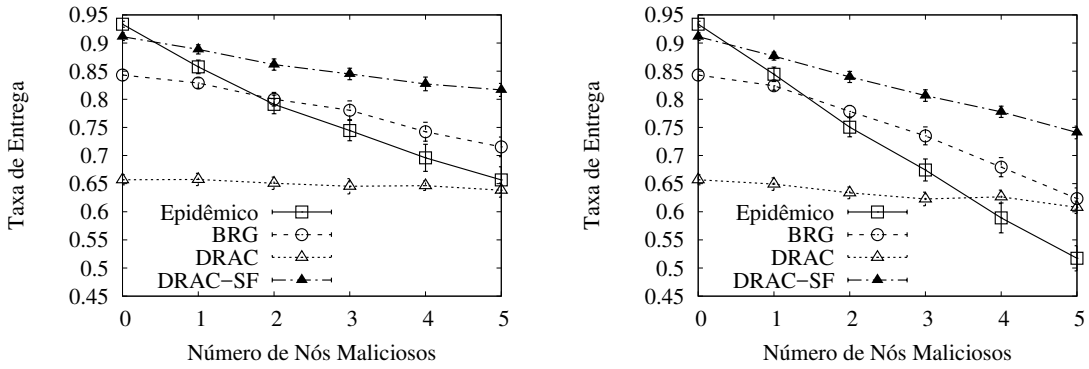
Tabela 6.2: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Rollernet e o ataque de falsificação de reconhecimentos positivos com buraco negro.

	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	7	5	6	7	10	12	8	6	8	10	14	19
Life	7	6	6	8	10	12	8	8	8	10	15	19
MaxProp	-1	9	18	24	32	36	-1	11	26	39	61	76
Prophet	4	5	6	7	9	12	4	6	7	10	12	17
ProphetV2	1	3	4	8	11	14	1	3	5	10	16	21
SnW	-7	-2	3	7	12	16	-7	-3	4	9	16	24
Wave	10	14	19	23	29	33	13	19	29	40	56	71

nas Tabelas 6.1 e 6.2 que para o protocolo Epidêmico a contramedida DRAC-SF supera o desempenho da contramedida BRG em até 19%, em termos relativos. No entanto, a contramedida DRAC obtém um resultado inferior até quando comparado com o protocolo Epidêmico sem contramedidas. Isto ocorre porque o protocolo Epidêmico envia tantas réplicas quanto possível, congestionando a rede. Quando não utiliza contramedidas, o protocolo remove mensagens da rede através de ACKs, sejam legítimos ou não, aliviando o congestionamento. Por outro lado, a contramedida DRAC não remove mensagens da rede através de ACKs, sejam eles legítimos ou não. Isto permite que o protocolo continue disseminando estas mensagens, inclusive aquelas que foram entregues e para as quais reconhecimentos legítimos foram gerados, intensificando o congestionamento e causando prejuízos ao desempenho. Por sua vez, a contramedida DRAC-SF apesar de não remover mensagens através de ACKs, interrompe a disseminação das mensagens para as quais ACKs, novamente legítimos ou não, foram gerados. Esta característica da proposta DRAC-SF resulta em um menor congestionamento, além de proteger a rede do ataque de falsificação de reconhecimentos positivos. Com relação a contramedida BRG, observa-se que ela melhora o desempenho do protocolo Epidêmico conforme o número de nós maliciosos aumenta, no entanto, seu desempenho continua inferior ao desempenho da DRAC-SF.

Outra observação que deve ser feita é a de que existe uma penalização para todas as contramedidas quando não existem nós maliciosos na rede. Paras as contramedidas DRAC e DRAC-SF, isto ocorre porque ambas atuam evitando descartar mensagens para as quais ACKs foram reconhecidos, mesmo quando não existem nós maliciosos na rede, o que prejudica o controle de congestionamento na rede. Para a contramedida BRG, isto pode ser explicado pela ocorrência de falsos positivos, isto é, ACKs são considerados falsi-

ficados mas na realidade não são, além de falsos positivos, quando ACKs são considerados legítimos, mas na verdade são falsificados.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

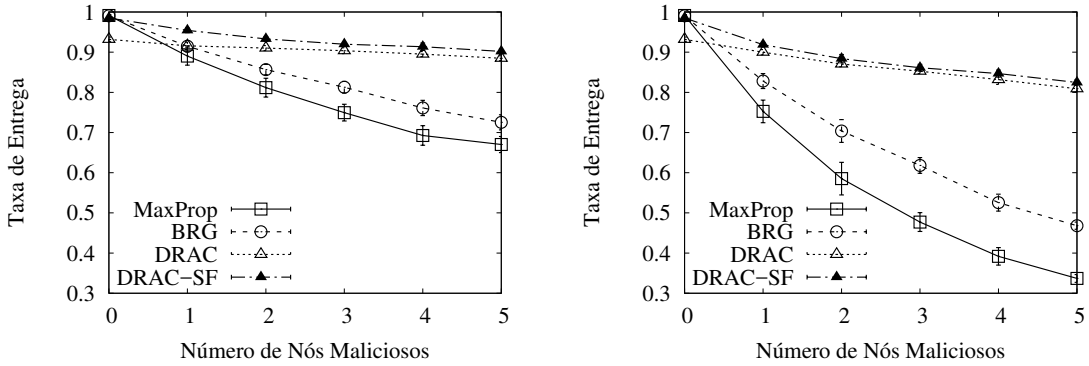
Figura 6.3: Taxa de entrega para o protocolo Epidêmico no cenário Rollernet.

Os resultados para o cenário Rollernet e o protocolo MaxProp são apresentados na Figura 6.4. Observa-se que as contramedidas DRAC e DRAC-SF obtêm sucesso na mitigação do ataque de falsificação de reconhecimentos positivos, em ambos os modelos de ataque. Observa-se também que a contramedida DRAC-SF possui desempenho ligeiramente superior ao desempenho da contramedida DRAC. Quando comparado com a BRG, o desempenho da DRAC-SF alcança melhorias absolutas e relativas de até 36% e 76%, respectivamente. Por sua vez, a contramedida BRG, apesar de alcançar um melhor desempenho com relação ao protocolo MaxProp sem contramedidas não é tão eficiente quanto as propostas deste trabalho.

É importante ressaltar que o protocolo MaxProp realiza um controle de replicação mais rigoroso, quando comparado ao protocolo Epidêmico. Portanto, o MaxProp não sofre tanto os efeitos do congestionamento da rede. Por este motivo, a contramedida DRAC apresenta com o protocolo MaxProp um desempenho superior ao apresentado com o protocolo Epidêmico. Neste caso, como o MaxProp não é tão suscetível ao congestionamento quanto o Epidêmico, a contramedida DRAC mitiga o efeito dos ataques mas não tem um efeito negativo no congestionamento da rede, que implicaria um desempenho inferior, como é observado no protocolo Epidêmico.

Observa-se ainda que as contramedidas propostas são mais resilientes ao ataque de falsificação de reconhecimentos positivos com buraco negro. Isto se deve ao fato de que as contramedidas mantêm as mensagens em *buffer* quando ACKs são recebidos ao invés de descartá-las imediatamente. Ao tomar esta decisão, um maior número de réplicas é

mantido na rede, tornando a rede mais robusta contra o ataque conjunto de falsificação de reconhecimentos positivos com buraco negro.



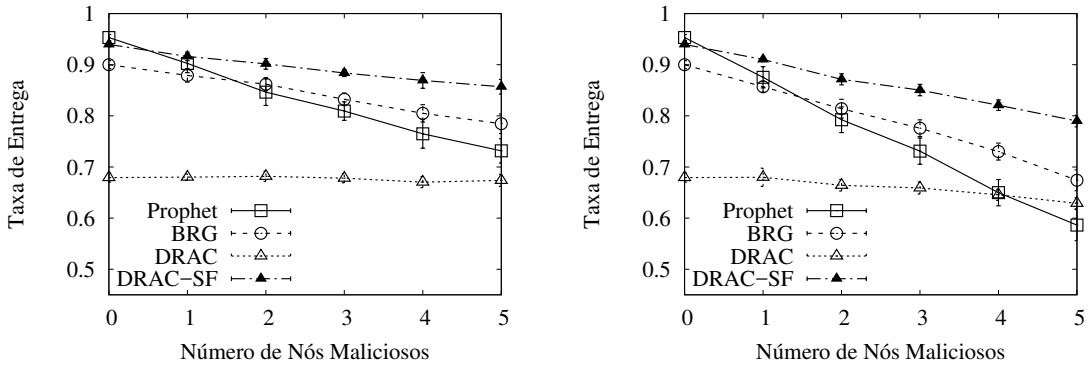
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.4: Taxa de entrega para o protocolo MaxProp no cenário Rollernet.

Os resultados para o protocolo Prophet no cenário Rollernet são apresentados na Figura 6.5. Os resultados são similares àqueles alcançados pelo protocolo Epidêmico. Cabe ressaltar novamente que apesar de fazer um controle de replicação que não é feito pelo protocolo Epidêmico, o protocolo Prophet pode replicar indefinidamente as mensagens, levando a rede ao congestionamento. De fato, a sobrecarga incorrida no protocolo Prophet pode ser similar à sobrecarga resultante do protocolo Epidêmico, como é possível observar na Seção 4.4.2. Passando aos resultados alcançados por este protocolo, a contramedida DRAC-SF obtém o melhor desempenho dentre as contramedidas avaliadas. Cabe ressaltar que a DRAC-SF é a única contramedida que atenua os efeitos negativos dos ataques, mesmo quando há somente um nó malicioso atuando na rede. Numericamente, o desempenho da contramedida DRAC-SF alcança uma melhoria relativa na taxa de entrega de até 17%, quando comparado com a BRG. Por sua vez, a contramedida DRAC alcança um desempenho inferior, visto que não é adaptada para cenários com alto congestionamento, como previamente exposto.

Os resultados para o protocolo *Spray and Wait* no cenário Rollernet são apresentados na Figura 6.6. É possível observar que as contramedidas propostas neste trabalho alcançam os melhores desempenhos com resultados similares. Os desempenho similar pode ser explicado pelo fato do protocolo *Spray and Wait* executar o controle mais estrito de replicação na rede, estabelecendo um limite máximo de réplicas para cada mensagem. Logo, o mecanismo *Stop Forwarding* adicionado à contramedida DRAC-SF não contribui para diminuição do congestionamento. Além disso, este mecanismo é menos acionado neste protocolo, visto que as mensagens podem estar na fase de espera (*wait*). Como



(a) Ataque de falsificação de reconhecimentos positivos.

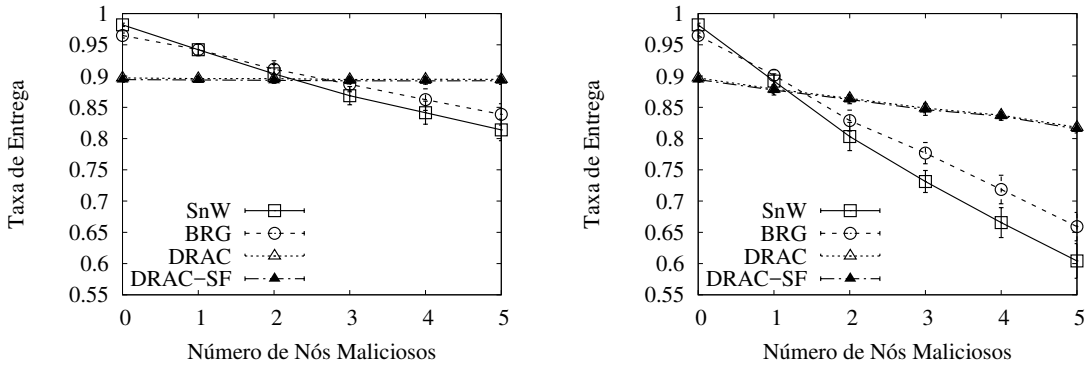
(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.5: Taxa de entrega para o protocolo Prophet no cenário Rollernet.

resultado, as contramedidas DRAC e DRAC-SF operam de modo mais semelhante neste protocolo do que nos demais.

Embora resultem no melhor desempenho, a atenuação dos efeitos negativos dos ataques alcançada pelas contramedidas propostas só é observada a partir de 3 nós maliciosos para o ataque de falsificação de reconhecimentos positivos e a partir de 2 nós maliciosos para o ataque de falsificação de reconhecimentos positivos com buraco negro. Como observado na Seção 5.4.1, o protocolo *Spray and Wait* é o protocolo mais resiliente ao ataque de falsificação de reconhecimentos positivos, dado o fato de que como este protocolo cria menos réplicas, existe uma menor probabilidade destas réplicas alcançarem nós maliciosos, o que resulta em uma menor probabilidade de criação de reconhecimentos positivos falsificados. Evidentemente, esta probabilidade decresce conforme decresce o número de nós maliciosos na rede. Portanto, com poucos nós maliciosos na rede, as contramedidas DRAC e DRAC-SF sofrem a penalização de desempenho na rede, como discutido anteriormente, visto que não tiram proveito da utilização de ACKs. No entanto, conforme aumenta o número de nós maliciosos e portanto, a eficiência dos ataques, as propostas atenuam os efeitos negativos dos ataques. Desta forma, quando existem 5 nós maliciosos atuando na rede, o desempenho da DRAC-SF obtém uma melhora relativa de até 24%, quando comparado com a BRG.

Finalmente, ao observar as Tabelas 6.1 e 6.2, é possível inferir que a contramedida DRAC-SF, proposta neste trabalho, supera o desempenho da contramedida BRG, a principal contramedida existente na literatura em 77 dos 84 dos cenários avaliados, ou seja, em 91,6% dos cenários avaliados, para o conjunto de mobilidade Rollernet. Cabe ressaltar que se forem ignorados os resultados para o protocolo *Spray and Wait* e também quando



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.6: Resultados para o protocolo *Spray and Wait* no cenário Rollernet.

não existem nós maliciosos na rede, a proposta DRAC-SF supera o desempenho da BRG em 100% dos cenários para o cenário Rollernet.

6.3.1.2 Cenário Dieselnet

As Figuras 6.7 a 6.10 apresentam os resultados obtidos pelos protocolos avaliados no cenário Dieselnet. As Tabelas 6.3 e 6.4 apresentam as diferenças de desempenho entre a contramedida DRAC-SF e BRG, para os ataques de falsificação de reconhecimentos positivos e falsificação de reconhecimentos positivos com buraco negro, respectivamente. Estas tabelas mostram que a contramedida DRAC-SF supera o desempenho da contramedida BRG em 72 dos 84 cenários avaliados para o conjunto de mobilidade Dieselnet, o que representa aproximadamente 86% dos cenários.

Tabela 6.3: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Dieselnet e o ataque de falsificação de reconhecimentos positivos.

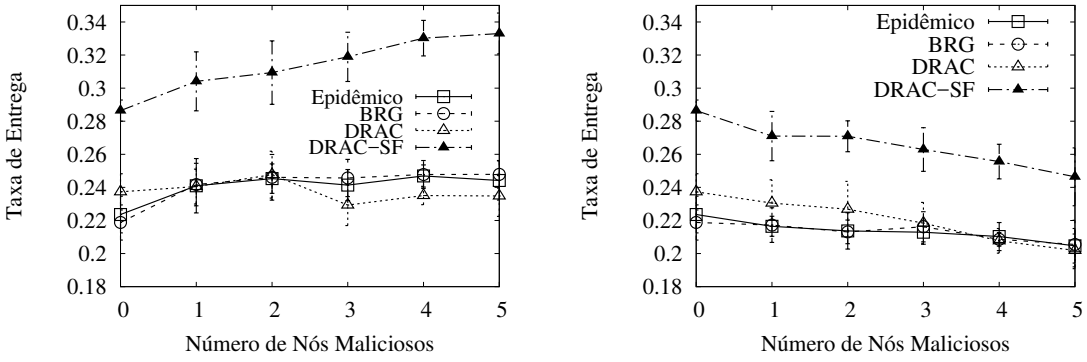
	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	7	6	6	7	8	9	31	26	26	30	33	34
Life	7	6	7	7	7	7	27	25	24	26	27	26
MaxProp	2	3	3	3	4	4	5	7	9	9	11	13
Prophet	2	1	2	2	2	2	5	4	4	5	5	4
ProphetV2	2	2	2	3	3	3	4	5	6	7	7	7
SnW	-7	-7	-6	-6	-5	-4	-19	-18	-17	-16	-14	-12
Wave	15	12	11	10	9	9	71	48	43	41	35	34

Os resultados para o protocolo Epidêmico são apresentados na Figura 6.7. Observa-se

que a contramedida DRAC-SF mitiga os ataques avaliados. Para o ataque de falsificação de reconhecimentos positivos, a contramedida DRAC-SF resulta em um aumento de desempenho mesmo quando o número de nós maliciosos na rede aumenta. Presume-se que este aumento está relacionado ao congestionamento da rede. Desta forma, um maior número de nós maliciosos resulta em um maior número de reconhecimentos positivos forjados. Como resultado, os nós legítimos param de replicar uma maior quantidade de mensagens, logo, o congestionamento na rede é reduzido.

Tabela 6.4: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Dieselnet e o ataque de falsificação de reconhecimentos positivos com buraco negro.

	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	7	5	6	5	5	4	31	25	27	22	23	20
Life	7	5	5	5	5	5	27	22	23	20	21	22
MaxProp	2	2	3	2	3	4	5	6	7	7	11	15
Prophet	2	2	2	1	2	1	5	5	4	3	4	3
ProphetV2	2	2	2	2	2	2	4	4	5	4	5	5
SnW	-7	-6	-6	-5	-3	-3	-19	-18	-17	-15	-11	-10
Wave	15	10	10	8	7	6	71	40	41	34	30	25



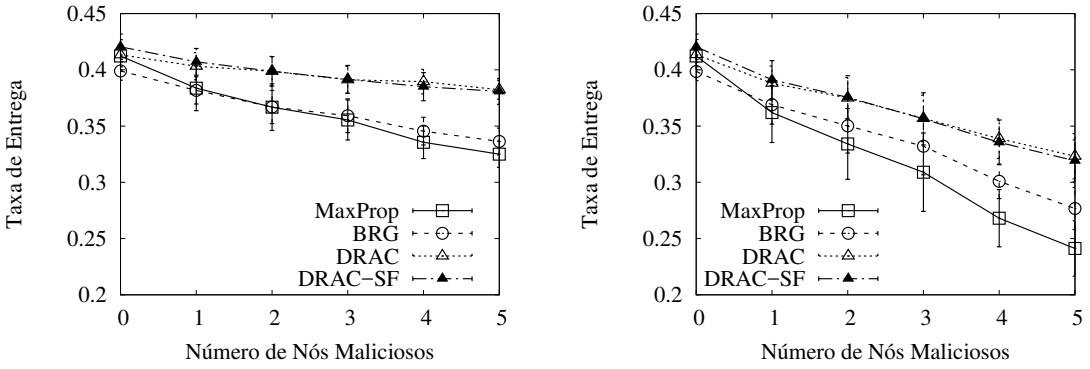
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.7: Taxa de entrega para o protocolo Epidêmico no cenário Dieselnet.

A Figura 6.8 apresenta os resultados para o protocolo MaxProp no cenário Dieselnet. Observa-se que as propostas alcançam o melhor desempenho, superando a contramedida BRG. A taxa de entrega alcançada pela DRAC-SF supera aquela obtida pela BRG em até 15%, em termos relativos.

Para protocolo Prophet, como mostra a Figura 6.9, a DRAC-SF mitiga o efeito negativo do ataque de falsificação de reconhecimentos positivos, embora a diferença no

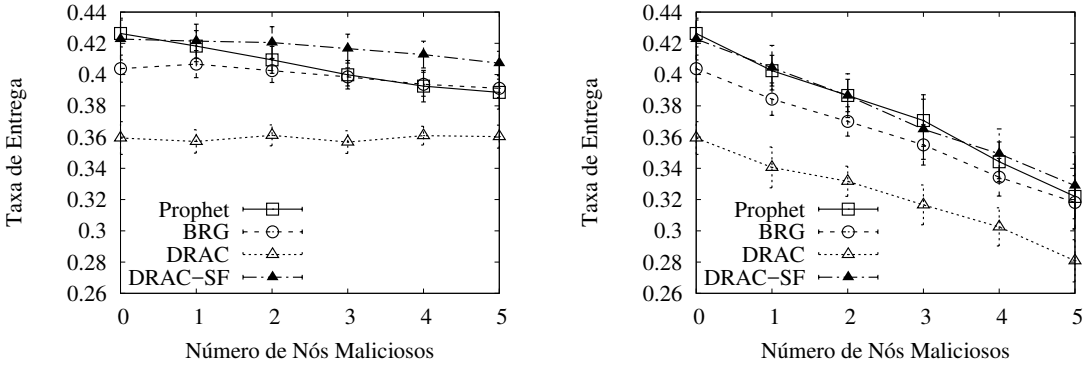


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.8: Taxa de entrega para o protocolo MaxProp no cenário Dieselnet.

desempenho seja pequena. Esta diferença pequena pode ser explicada pelo fato de que o próprio ataque de falsificação de reconhecimentos positivos não é eficiente em prejudicar o desempenho da rede, levando a uma queda absoluta máxima de aproximadamente 3% na taxa de entrega do protocolo Prophet. Por outro lado, no ataque de falsificação de reconhecimentos positivos com buraco negro, nenhuma das contramedidas mostrou-se eficiente para este cenário. Esta resultado pode ser compreendido da seguinte maneira. O ataque reduz o número de réplicas na rede através dos reconhecimentos positivos falsificados e dos buracos negros. Por sua vez, o ataque de buraco negro tem maior eficiência contra protocolos que replicam menos mensagens na rede, como pode ser observado na avaliação dos ataques feita na Seção 5.4. Além disso, nenhuma destas contramedidas foi projetada com o objetivo específico de mitigar o ataque de buraco negro. Adicionando-se o congestionamento na rede devido a tamanhos de *buffers* restritos, as contramedidas não são eficientes em mitigar o ataque neste cenário.

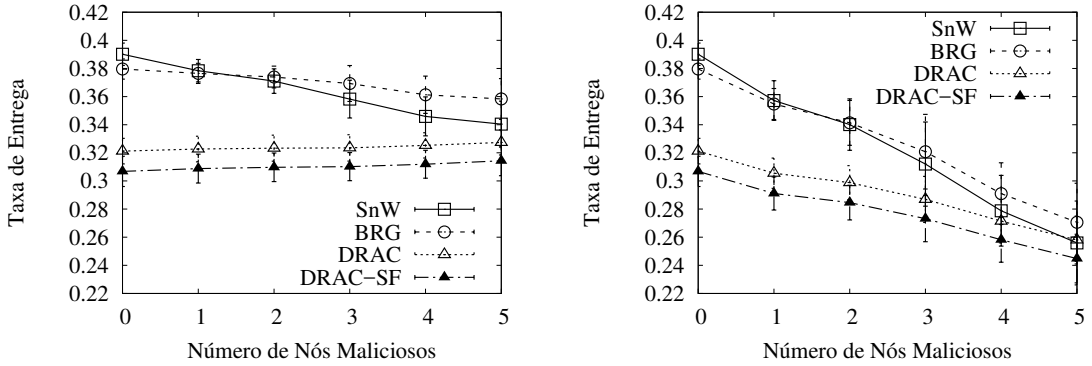


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.9: Taxa de entrega para o protocolo Prophet no cenário Dieselnet.

Os resultados para o protocolo *Spray and Wait* no cenário Dieselnet são apresentados na Figura 6.10. Este é o cenário onde as contramedidas propostas alcançaram o pior desempenho. Além disso, neste cenário a DRAC-SF obteve um desempenho inferior a DRAC. Atribui-se este resultado ao fato de a DRAC-SF parar a replicação das mensagens, visto que o protocolo *Spray and Wait* controla estritamente o número de réplicas na rede, o resultado é que um número menor ainda de réplicas é gerado, diminuindo a robustez da rede. Quando comparadas com a contramedida BRG, o desempenho das propostas pode ser compreendido da seguinte maneira. Neste cenário em específico, a ação do ataque de falsificação de reconhecimentos positivos é limitada, como discutido anteriormente. Logo, quando um nó recebe um reconhecimento positivo, a probabilidade deste reconhecimento positivo ser legítimo é maior do que nos cenários onde o ataque é mais efetivo. Portanto, ao manter todas as mensagens reconhecidas em *buffer*, a contramedida perde a oportunidade de remover dos *buffers* dos nós mensagens que já foram entregues ao destino.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.10: Taxa de entrega para o protocolo *Spray and Wait* no cenário Dieselnet.

Assim como no cenário Rollernet, ignorando os resultados obtidos para o protocolo *Spray and Wait* e para os cenários com nenhum nó malicioso, a contramedida DRAC-SF supera o desempenho da contramedida BRG em 100% dos cenários para o conjunto Dieselnet.

6.3.1.3 Cenários Infocom e Shopping

Embora omitidos nesta seção por questão de espaço e para evitar redundância, cabe destacar os resultados para os cenários de mobilidade Infocom e Shopping, que são apresentados no Apêndice C. Para o Infocom, a proposta DRAC-SF superou o desempenho da contramedida BRG em 74 dos 84 cenários avaliados, o que corresponde a 88% dos

cenários. Para o conjunto de mobilidade Shopping, a taxa de entrega alcançada pela DRAC-SF foi maior que a taxa de entrega alcançada pela BRG em aproximadamente 87% dos cenários avaliados.

Com relação a diferença relativa entre o desempenhos das contramedidas DRAC-SF e BRG, ela chega a 65% no cenário Infocom contra o ataque de falsificação de reconhecimentos positivos. No mesmo cenário, contra o ataque de falsificação de reconhecimentos positivos, esta melhoria chega a 135%.

Finalmente, assim como nos cenários Rollernet e Dieselnet, se forem ignorados os resultados para o protocolo *Spray and Wait* e para os demais protocolos quando não existem nós maliciosos na rede, a contramedida DRAC-SF supera o desempenho da contramedida BRG em todos os cenários para os conjuntos de mobilidade Infocom e Shopping.

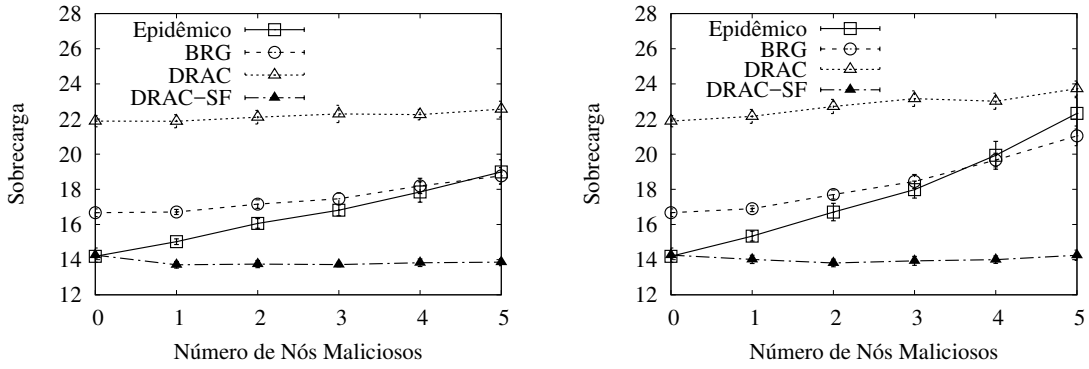
6.3.2 Sobrecarga

Nesta seção, a sobrecarga alcançada pelas contramedidas nos cenários Rollernet e Dieselnet é apresentada.

6.3.2.1 Cenário Rollernet

As Figuras 6.11 a 6.14 ilustram as sobrecargas obtidas no cenário Rollernet. Observa-se que a contramedida DRAC-SF resulta na menor sobrecarga em todos os protocolos avaliados, com exceção do protocolo *Spray and Wait* sofrendo ataque de falsificação de reconhecimentos positivos, o qual é ilustrado na Figura 6.14(a). Neste cenário, a DRAC-SF alcança o melhor resultado para sobrecarga somente quando existem 5 nós maliciosos na rede. Para os demais cenários, enquanto as demais contramedidas apresentam uma tendência de aumento da sobrecarga com o aumento do número de nós maliciosos na rede, a contramedida DRAC-SF mantém a sobrecarga estabilizada.

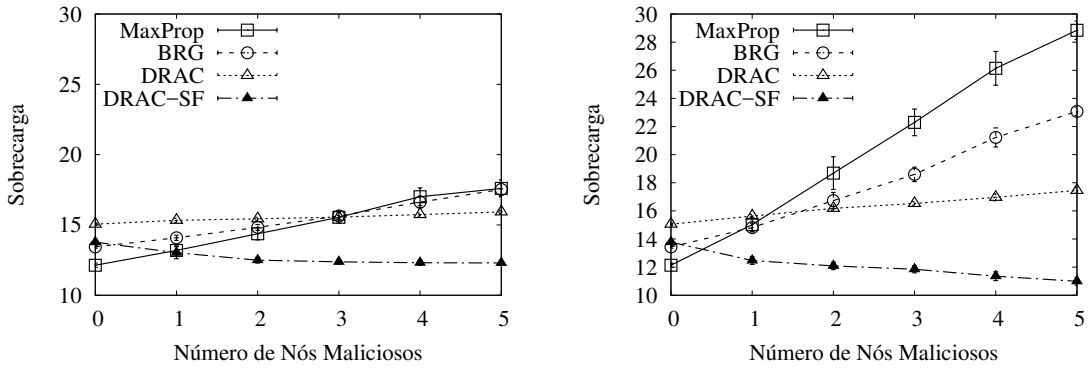
A menor sobrecarga alcançada pela contramedida DRAC-SF decorre da maior eficiência desta contramedida. Por sua vez, esta eficiência decorre da mitigação do ataque de falsificação de reconhecimentos positivos, pois evita descartar mensagens que não foram entregues ao destinatário, o que diminuiria a taxa de entrega e resultaria em uma menor eficiência com relação a sobrecarga. Além disso, a sobrecarga também é reduzida pelo fato da contramedida DRAC-SF não replicar mensagens para as quais reconhecimentos positivos foram recebidos.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

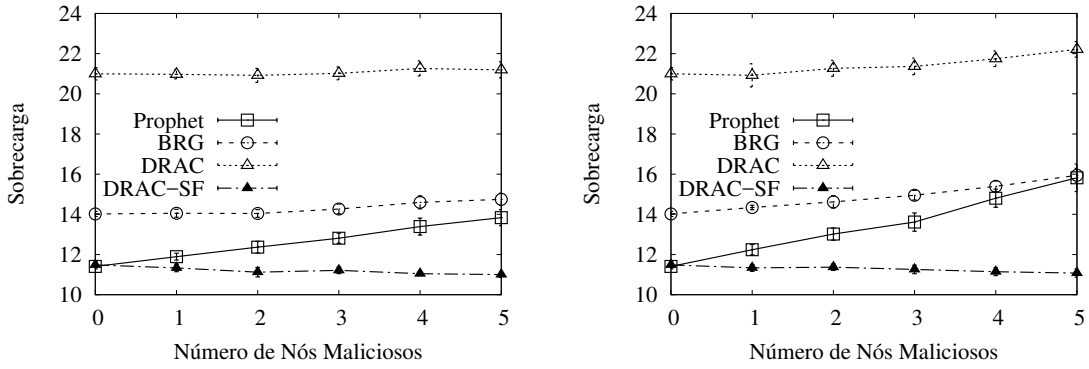
Figura 6.11: Sobrecarga para o protocolo Epidêmico no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

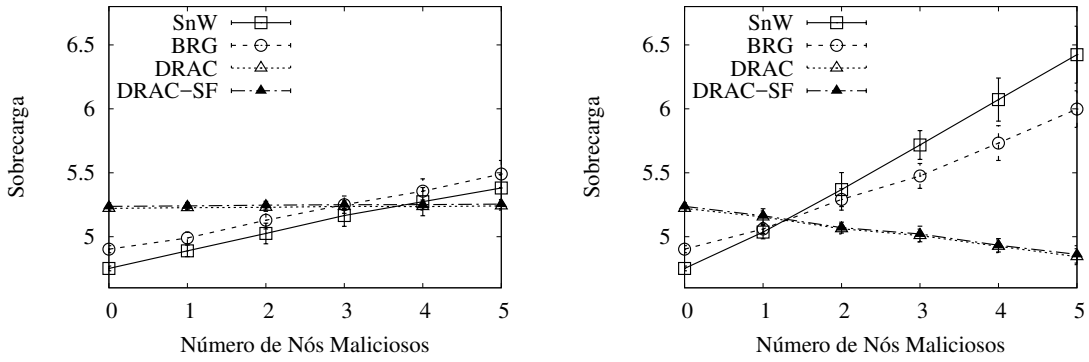
Figura 6.12: Sobrecarga para o protocolo MaxProp no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.13: Sobrecarga para o protocolo Prophet no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

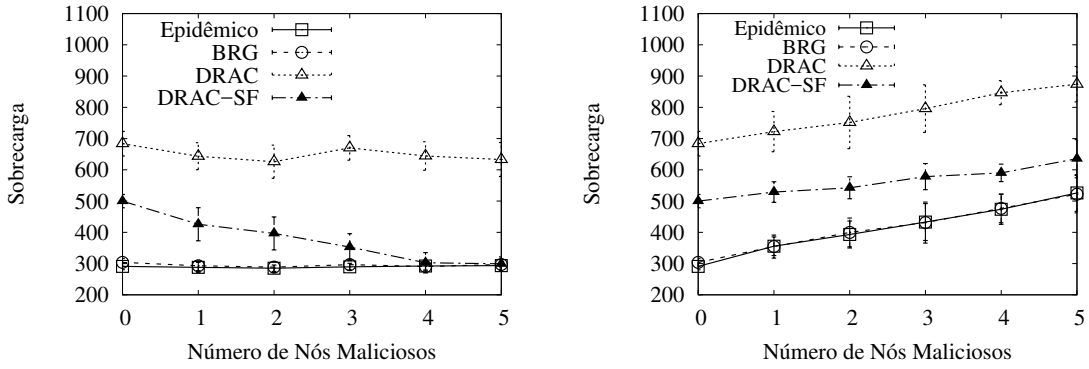
Figura 6.14: Sobrecarga para o protocolo *Spray and Wait* no cenário Rollernet.

6.3.2.2 Cenário Dieselnet

As Figuras 6.15 a 6.18 ilustram as sobrecargas obtidas no cenário Dieselnet. Nota-se na Figura 6.16 que a contramedida DRAC-SF obtém a menor sobrecarga para o protocolo MaxProp. Por sua vez, para o protocolo *Spray and Wait*, as propostas DRAC e DRAC-SF levam às menores sobrecargas.

Para o protocolo MaxProp, a menor sobrecarga é resultado da melhor eficiência da contramedida DRAC-SF, que entrega mais mensagens, junto a contramedida DRAC. No entanto, enquanto DRAC-SF reduz a sobrecarga ao não replicar mensagens para as quais reconhecimentos positivos foram recebidos, a DRAC continua a replicar, não beneficiando-se da diminuição da sobrecarga alcançada por DRAC-SF.

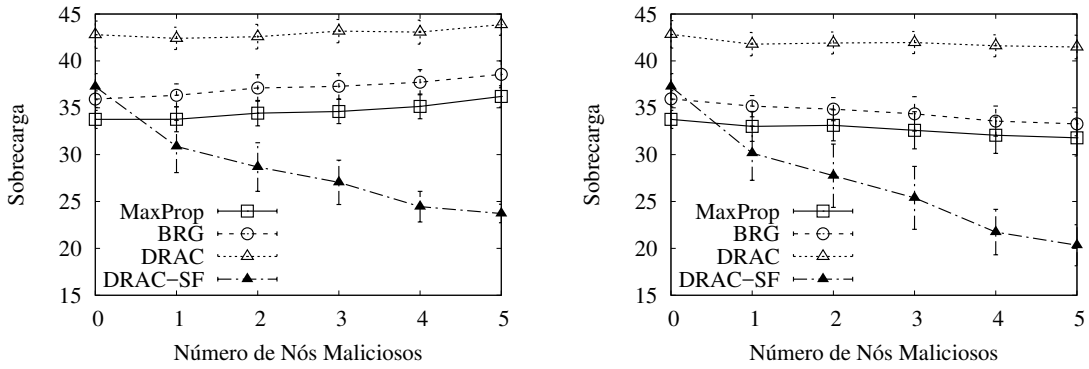
Embora tenha levado a uma melhoria na taxa de entrega para o protocolo Epidêmico, a contramedida DRAC-SF foi menos eficiente com relação a sobrecarga. As propostas também resultaram em maior sobrecarga com o protocolo Prophet. Como visto anteriormente neste trabalho, em cenários congestionados os ataques podem levar a melhorias contra-intuitivas no desempenho da rede. Especificamente nestes casos, ao optarem por não remover mensagens através de reconhecimentos positivos, as contramedidas resultam em uma maior sobrecarga.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

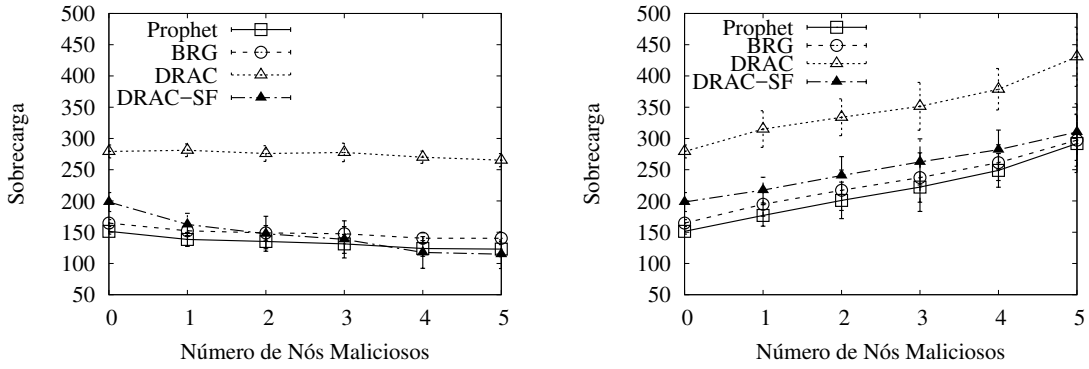
Figura 6.15: Sobrecarga para o protocolo Epidêmico no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.16: Sobrecarga para o protocolo MaxProp no cenário Dieselnet.

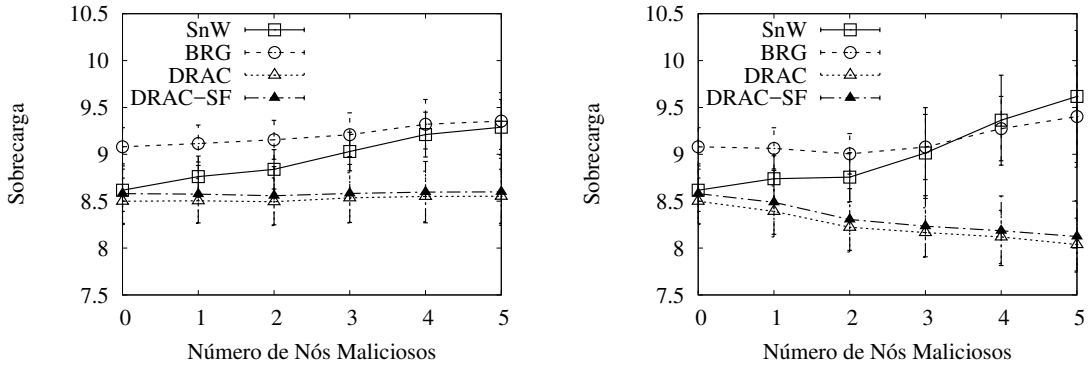


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.17: Sobrecarga para o protocolo Prophet no cenário Dieselnet.

Finalmente, embora tenham alcançado menores sobrecargas para o protocolo *Spray and Wait*, as propostas também levaram a piores taxas de entrega neste protocolo. Cabe lembrar que, dentre os cenários avaliados, este é o menos suscetível ao ataque de falsificação de reconhecimentos positivos. No entanto, as contramedidas propostas comportam-se sempre esperando que um ataque de esteja ocorrendo e isto prejudica o desempenho neste cenário.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.18: Sobrecarga para o protocolo *Spray and Wait* no cenário Dieselnet.

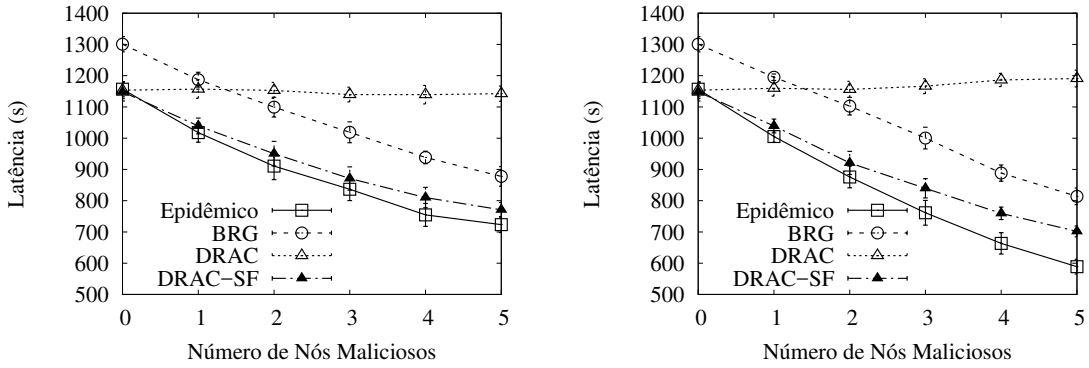
6.3.3 Atraso de Entrega

Nesta seção, são apresentados os resultados para a métrica atraso de entrega.

6.3.3.1 Cenário Rollernet

O atraso de entrega alcançado pelos protocolos no cenário Rollernet é apresentado nas Figuras 6.19 a 6.22. Os resultados para o protocolo Epidêmico são exibidos na Figura 6.19. Nesta figura é possível observar que a contramedida DRAC-SF reduz o atraso de entrega em comparação com a contramedida BRG. Isto acontece porque a DRAC-SF é mais eficiente, além de reduzir o congestionamento na rede através do mecanismo *Stop Forwarding*. Com um menor congestionamento, as mensagens que ainda não foram entregues ao destino fluem mais rapidamente na rede. No entanto, o protocolo Epidêmico sem contramedida alcança o menor atraso. Este desempenho aparentemente melhor é decorrente da baixa eficiência na entrega de mensagens, visto que o protocolo entrega principalmente as mensagens que podem ser entregues mais rapidamente, dado que quanto mais permanecem em *buffer*, mais as mensagens estão suscetíveis aos ataques. Como resultado, somente estas mensagens têm sua latência computada, levando a uma menor atraso de entrega médio.

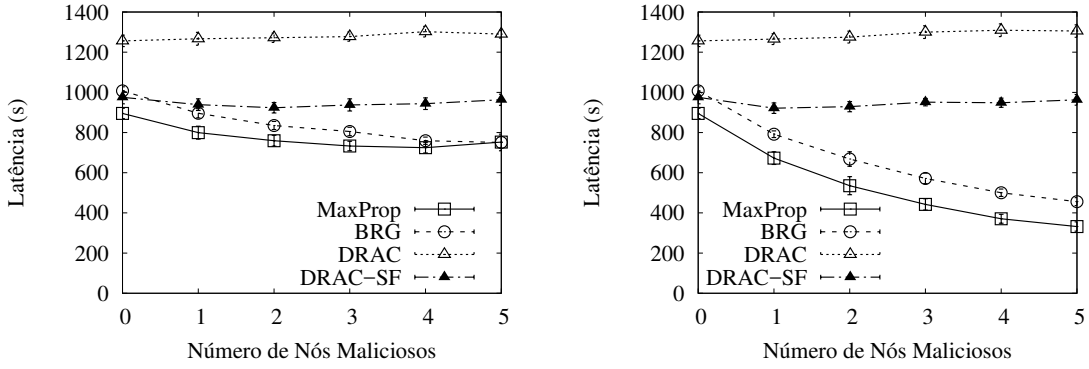
Cabe ressaltar que estes resultados são similares aos obtidos pelo protocolo Prophet, na Figura 6.21, o que dispensa esclarecimentos adicionais para este protocolo.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.19: Atraso de entrega para o protocolo Epidêmico no cenário Rollernet.

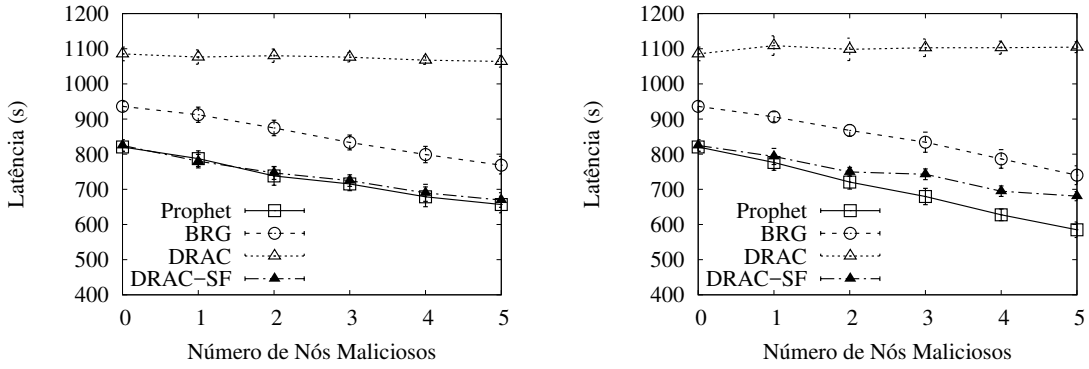


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.20: Atraso de entrega para o protocolo MaxProp no cenário Rollernet.

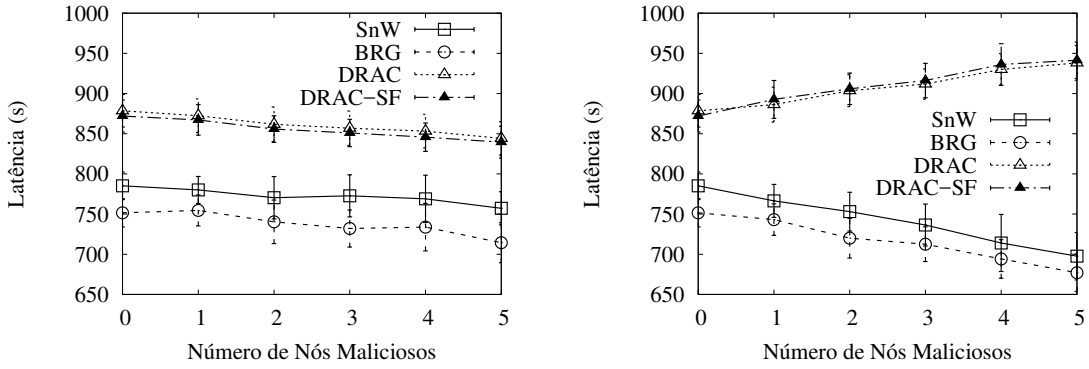
Resultados contrários são alcançados pelos protocolos MaxProp e *Spray and Wait*, ilustrados respectivamente nas Figura 6.20 e 6.22. Nestes resultados é possível observar que as contramedidas propostas alcançam atraso de entrega maior que a contramedida BRG e os protocolos MaxProp e *Spray and Wait* sem contramedidas. O aumento do atraso de entrega decorrente do aumento da taxa de entrega é um resultado comum que também aparece em vários outros trabalhos [96, 121, 80, 16]. Burns *et al.* argumentam que este aumento é esperado quando um protocolo de roteamento entrega mensagens que os outros falham em entregar. Além deste caso, as mensagens permanecem mais tempo em *buffer* com a utilização das contramedidas propostas, visto que elas optam por não remover mensagens através de reconhecimentos positivos com o objetivo de proteger a rede do ataque de falsificação de reconhecimentos positivos.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.21: Atraso de entrega para o protocolo Prophet no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

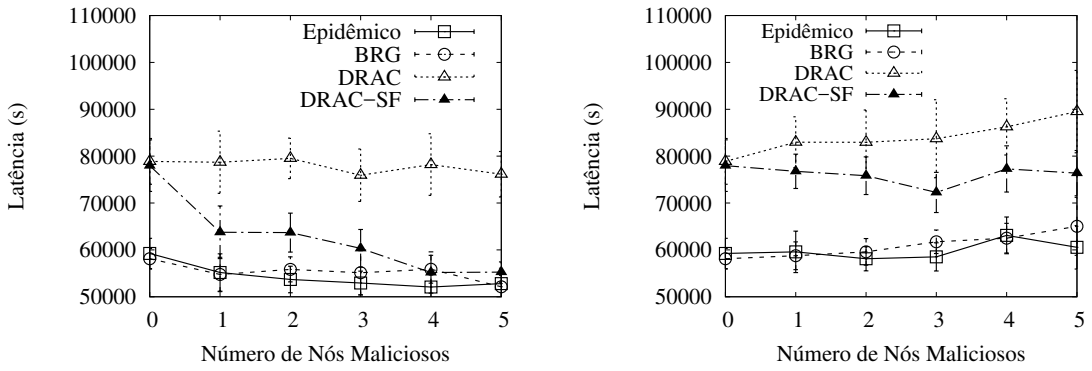
(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.22: Atraso de entrega para o protocolo *Spray and Wait* no cenário Rollernet.

6.3.3.2 Cenário Dieselnet

O atraso de entrega alcançado pelos protocolos no cenário Dieselnet é apresentado nas Figuras 6.23 a 6.26. É possível observar nestes resultados que as contramedidas propostas levam a um maior atraso de entrega para os protocolos avaliados. Para a contramedida DRAC-SF, como no cenário Rollernet, este aumento do atraso de entrega decorre do fato da proposta entregar mensagens que não são entregues pela contramedida BRG e pelos protocolos sem contramedidas, além do próprio projeto da contramedida que opta por manter as mensagens em *buffer* por um maior tempo. A propósito, esta característica também elucida o maior atraso de entrega da contramedida DRAC, mesmo em cenários onde não levou a ganhos na taxa de entrega. Adicionalmente, a contramedida DRAC continua replicando mensagens após receber reconhecimentos positivos, o que contribui para o aumento do congestionamento com consequente impacto negativo no atraso de

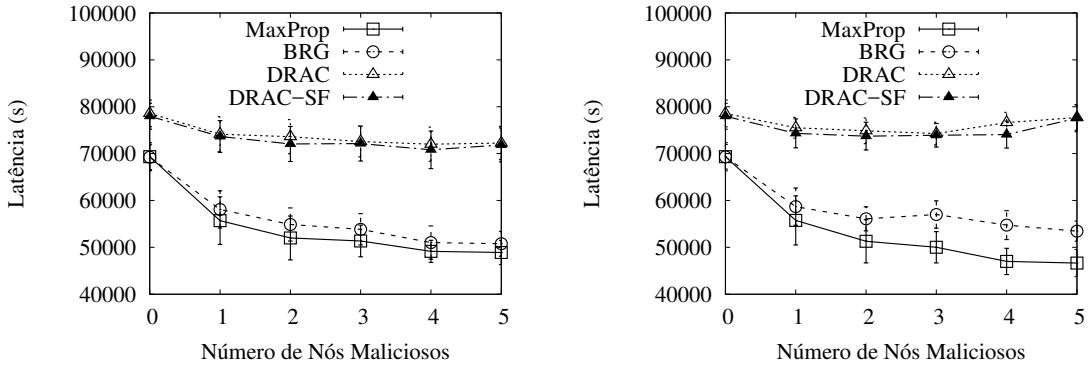
entrega.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

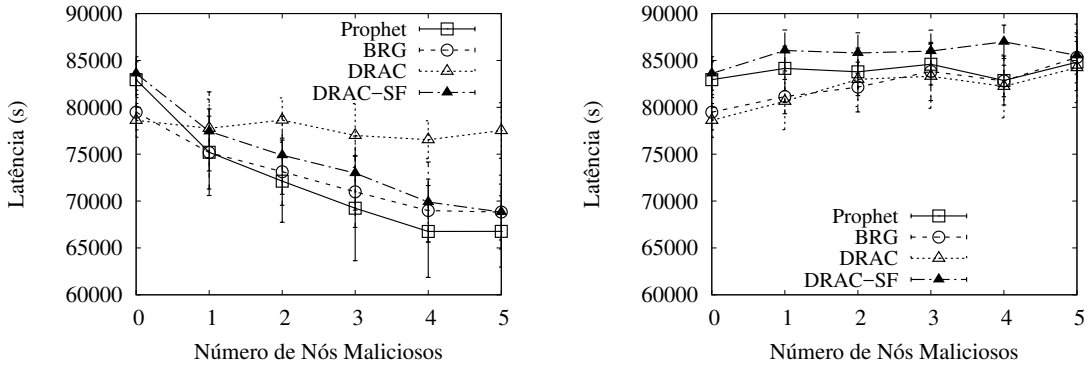
Figura 6.23: Atraso de entrega para o protocolo Epidêmico no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

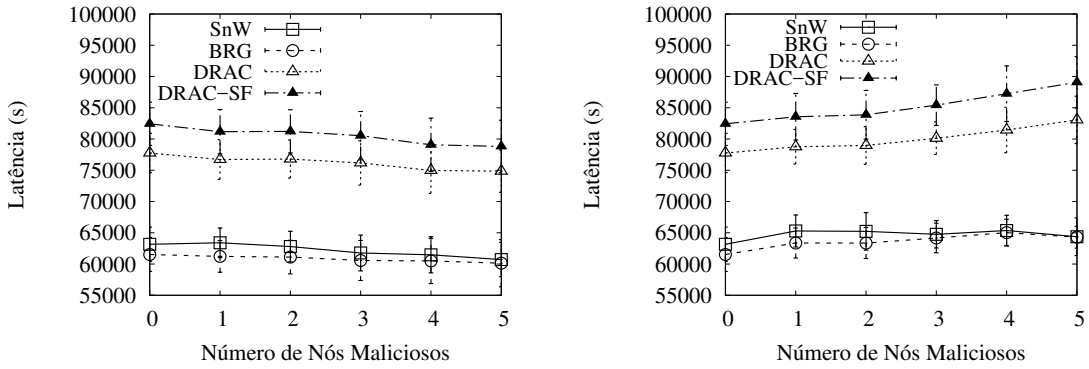
Figura 6.24: Atraso de entrega para o protocolo MaxProp no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.25: Atraso de entrega para o protocolo Prophet no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

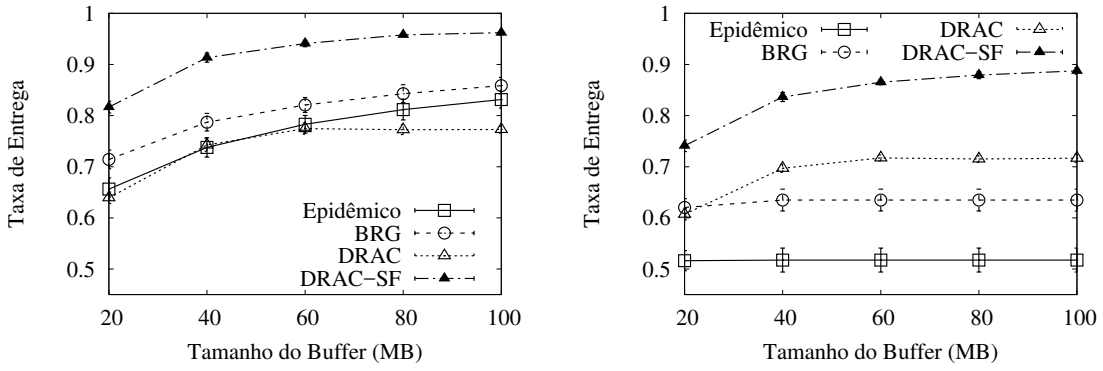
Figura 6.26: Atraso de entrega para o protocolo *Spray and Wait* no cenário Dieselnet.

6.3.4 O Efeito da Variação do Tamanho do *Buffer*

Com o objetivo de averiguar o efeito da variação do tamanho dos *buffers* dos nós no desempenho das contramedidas, simulações foram realizadas utilizando-se os protocolos Epidêmico, MaxProp e Prophet para os cenários Rollernet e Dieselnet. Para tanto, nestas simulações os tamanhos dos *buffers* dos nós foi configurado para 20, 40, 60, 80 e 100 M. Os resultados para a taxa de entrega são exibidos a seguir. Nas figuras exibidas, a menos que seja estabelecido o contrário, o eixo X apresenta os diferentes tamanhos de *buffer* utilizados e o eixo Y apresenta a taxa de entrega obtida.

6.3.4.1 Cenário Rollernet

As Figuras 6.27, 6.28 e 6.29 apresentam os resultados para os protocolos Epidêmico, MaxProp e Prophet, respectivamente. A seguir, algumas observações que são comuns a todas as figuras são feitas. É possível observar que a contramedida DRAC-SF alcança o melhor resultado em todos os cenários avaliados. Além disso, observa-se de modo geral que quanto maior o tamanho do *buffer*, maior a taxa de entrega de mensagens. Uma exceção a observação anterior é observada para o ataque de falsificação de reconhecimentos positivos com buraco negro, que mantém os desempenhos da contramedida BRG e dos protocolos nativos estabilizados independente do tamanho do *buffer*. Em outras palavras, a contramedida BRG e os protocolos nativos não são beneficiados pelo aumento do *buffer* quando o ataque de falsificação de reconhecimentos positivos com buraco negro está sendo realizado. Este comportamento é avaliado com maiores detalhes adiante nesta seção.

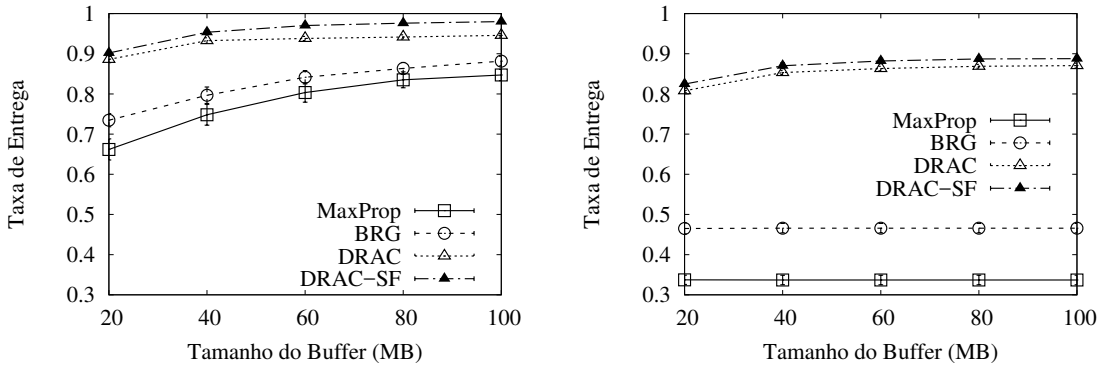


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.27: Taxa de entrega para o protocolo Epidêmico no cenário Rollernet.

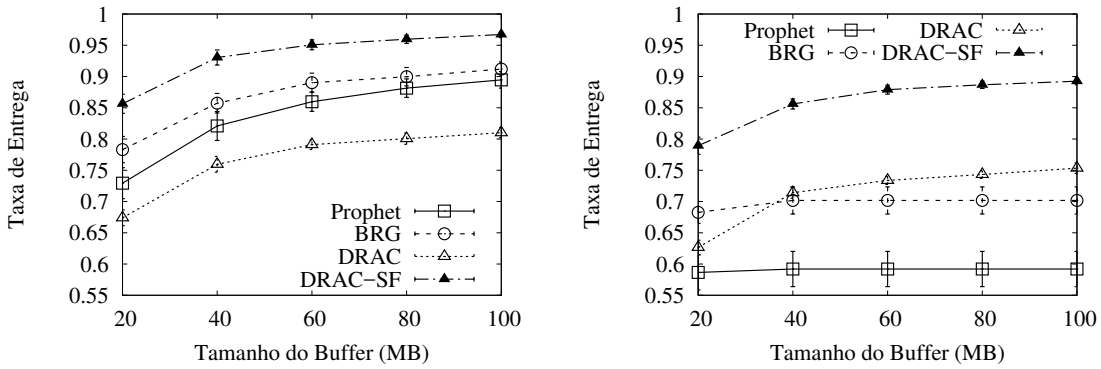
Com relação a contramedida DRAC, é possível observar que ela alcança o segundo melhor desempenho contra o ataque de falsificação de reconhecimentos positivos com buraco negro. No entanto, para o ataque sem buraco negro, esta proposta alcança o segundo melhor desempenho somente para o protocolo MaxProp.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.28: Taxa de entrega para o protocolo MaxProp no cenário Rollernet.



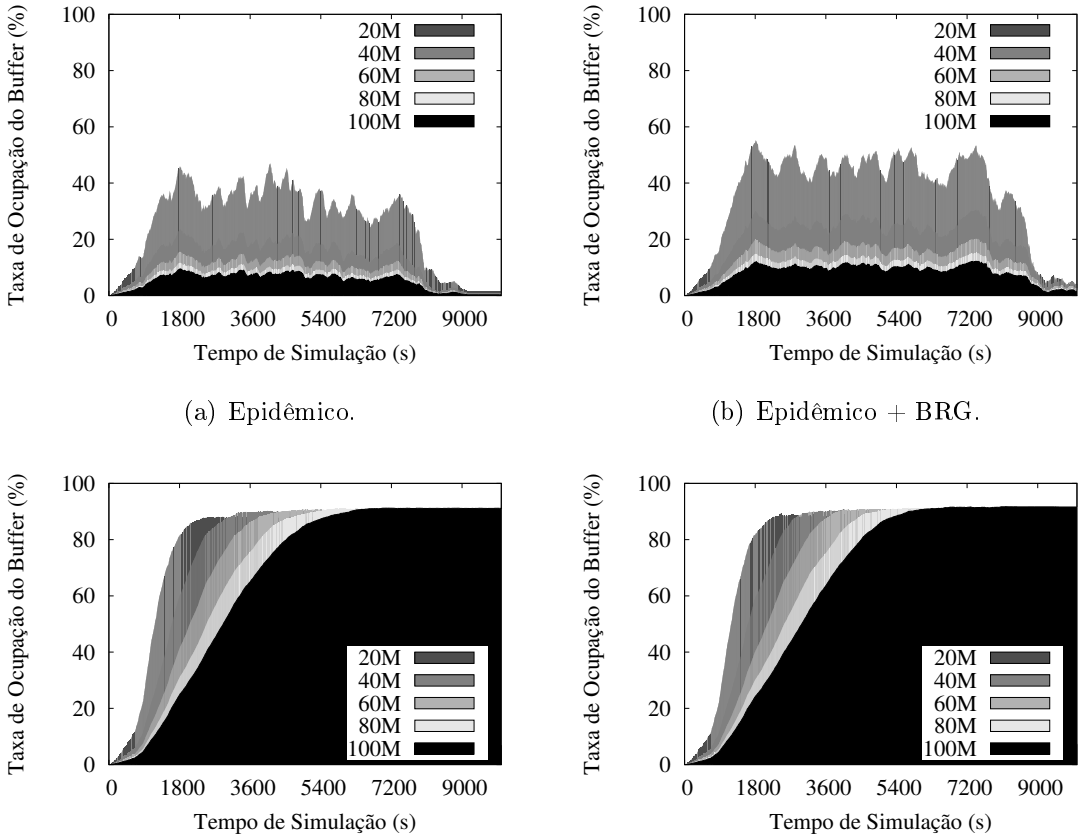
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.29: Taxa de entrega para o protocolo Prophet no cenário Rollernet.

Com o objetivo de detalhar a origem do problema dos protocolos nativos e da contramedida BRG, que não atingem melhores taxas de entrega com o aumento do tamanho do *buffer*, a taxa de ocupação média dos *buffers* dos nós é exibida na Figura 6.30. É possível observar que para o protocolo Epidêmico a taxa média de ocupação do *buffer* não ultrapassa 50% nem mesmo quando este é limitado a 20 M. Por sua vez, para a contramedida BRG, a taxa média de ocupação não ultrapassa 60%. Ainda para o protocolo Epidêmico e a contramedida BRG, o período de queda na taxa de ocupação próximo a 9.000 segundos é em razão do período final da simulação, onde novas mensagens não são geradas. Por sua vez, as propostas DRAC e DRAC-SF atingem uma taxa média de ocupação de *buffer* próxima a 90% e mantêm esta taxa de ocupação durante toda o restante da simulação. Este comportamento é esperado, visto que as propostas não removem mensagens atra-

vés de reconhecimentos positivos, enquanto que o protocolo Epidêmico e a contramedida BRG o fazem. Além disso, ambas estas estão suscetíveis ao ataque de falsificação de reconhecimentos positivos, o que pode reduzir ainda mais a taxa de ocupação dos *buffers*.



(a) Epidêmico.

(b) Epidêmico + BRG.

(c) Epidêmico + DRAC.

(d) Epidêmico + DRAC-SF.

Figura 6.30: Taxa de ocupação do *buffer* para o protocolo Epidêmico no cenário Rollernet durante o ataque de falsificação de reconhecimentos positivos com buraco negro.

No entanto, os resultados até aqui apresentados não esclarecem por que os protocolos nativos e a contramedida BRG não obtêm melhorias de desempenho contra o ataque de falsificação de reconhecimentos positivos com buraco negro conforme o tamanho dos *buffers* aumenta. A Figura 6.31 apresenta uma comparação entre a taxa média de ocupação dos *buffers* dos nós para o protocolo Epidêmico, no cenário Rollernet, quando o ataque de falsificação de reconhecimentos positivos com buraco negro é realizado por 5 nós e no mesmo cenário, exceto que sem nós maliciosos. Nesta figura, optou-se por apresentar somente os tamanhos de *buffer* extremos, isto é, 20 M e 100 M. Nela, observa-se que quando o ataque está sendo realizado, a taxa média de ocupação dos *buffers* dos nós é ainda mais reduzida. Isto indica que o ataque está sendo efetivo em remover mensagens que não deveriam ser removidas da rede, reduzindo a disponibilidade de réplicas de men-

sagens e levando a um impacto negativo na taxa de entrega. Isto também indica que o aumento do tamanho dos *buffers* dos nós não protege a rede contra o ataque. Finalmente, os resultados apontam que os protocolos nativos e a contramedida BRG não alcançam melhores desempenhos com o aumento do *buffer* neste cenário porque o tamanho restrito de *buffer* não é o principal problema deste cenário, visto que a taxa média de ocupação nunca aproxima-se do máximo. Por outro lado, a efetividade do ataque é um problema que precisa ser combatido, mas não é tratado com eficiência pela contramedida BRG.

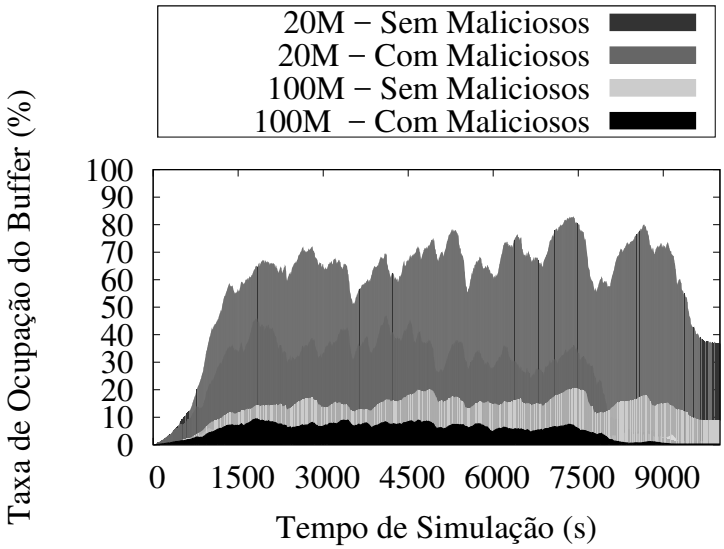
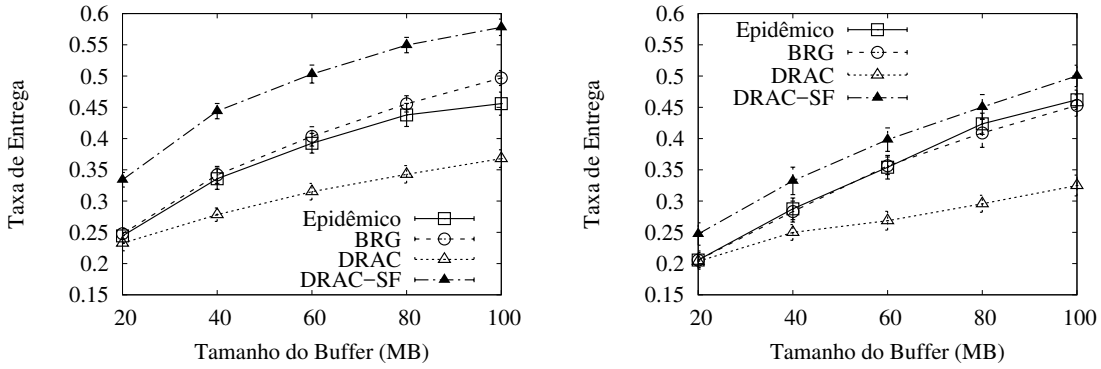


Figura 6.31: Diferença entre a taxa de ocupação de *buffer* para o protocolo Epidêmico no cenário Rollernet com 5 nós maliciosos e no mesmo cenário sem nós maliciosos.

6.3.4.2 Cenário Dieselnet

As Figuras 6.32 a 6.34 apresentam o impacto do tamanho do *buffer* na taxa de entrega para o cenário Dieselnet.

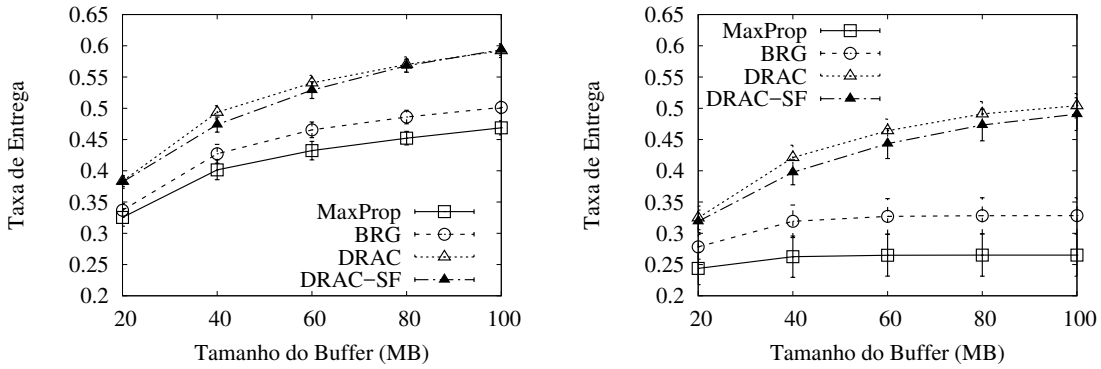


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.32: Taxa de entrega para o protocolo Epidêmico no cenário Dieselnet.

Observa-se que a contramedida DRAC-SF supera o desempenho da contramedida BRG para todos os cenários avaliados e que a diferença de desempenho é maior para o ataque sem a utilização de buracos negros. Cabe ressaltar novamente que as contramedidas avaliadas neste trabalho não foram projetadas especificamente para mitigarem o ataque de falsificação de reconhecimentos positivos.



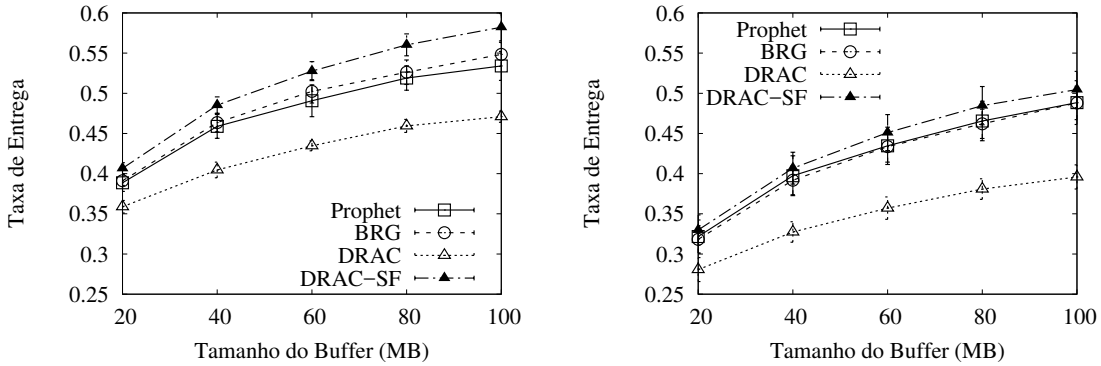
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.33: Taxa de entrega para o protocolo MaxProp no cenário Dieselnet.

Com relação a contramedida DRAC, é possível observar que esta supera o desempenho da contramedida BRG e do protocolo MaxProp nativo, para o protocolo MaxProp, mas obtém resultados inferiores para os demais protocolos. Ressalta-se novamente que esta contramedida é prejudicada pelo congestionamento incorrido na rede pelo protocolos Epidêmico e Prophet, visto que continua replicando mensagens indefinidamente, mesmo após os nos receberem reconhecimentos positivos. Por outro lado, o protocolo MaxProp

faz um controle maior do número de réplicas na rede e consequentemente o congestionamento é menor para este protocolo. Isto permite que a contramedida DRAC obtenha um desempenho melhor que nos demais protocolos.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.34: Taxa de entrega para o protocolo Prophet no cenário Dieselnet.

6.3.5 O Efeito da Variação do TTL

Com o objetivo de averiguar o efeito da variação do tamanho do TTL das mensagens no desempenho das contramedidas, simulações foram realizadas utilizando-se os protocolos Epidêmico, MaxProp e Prophet para os cenários Rollernet e Dieselnet. Nestas simulações o TTL das mensagens foi configurado como 5%, 10%, 15%, 20% e 25% do tempo total de simulação. Os valores configurados são apresentados na Tabela 6.5. Após arredondamento, estas porcentagens correspondem para o cenário Rollernet a 8, 17, 25, 33 e 42 minutos. Para o cenário Dieselnet os valores correspondentes são: 565, 1.130, 1.696, 2.261 e 2.826 minutos.

Tabela 6.5: Valores de TTL configurados para os cenários.

Cenário	**Tempo Total (m)**	**5%**	**10%**	**15%**	**20%**	**25%**
Rollernet	167	8	17	25	33	42
Dieselnet	11.306	565	1.130	1.696	2.261	2.826

Os resultados para taxa de entrega são apresentados nesta seção. A menos que seja estabelecido o contrário, para as figuras desta seção, o eixo X apresenta os diferentes valores de TTL utilizados e o eixo Y apresenta a taxa de entrega resultante.

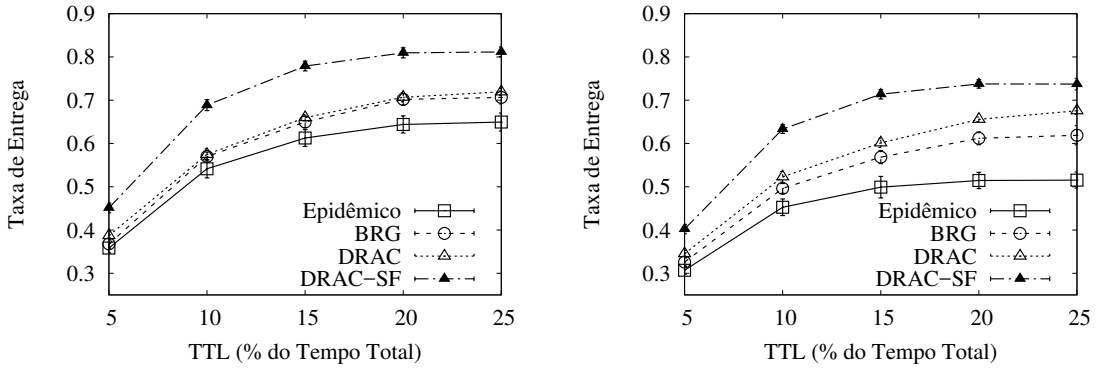
Um resultado é comum para todos os cenários avaliados nesta seção: quanto maior o

TTL utilizado, maior a taxa de entrega alcançada. Isto ocorre porque TTLs pequenos não permitem que as mensagens sejam disseminadas pela rede e cheguem ao destinatário. Por outro lado, conforme o TTL aumenta, aumenta também a chance das mensagens serem replicadas e permanecerem na rede até encontrarem o destinatário.

Cabe ressaltar que existe a possibilidade de que ao incrementar o TTL, o desempenho da rede seja prejudicado. Isto pode ocorrer em razão do congestionamento, visto que ao aumentar o TTL mais mensagens permanecerão na rede por mais tempo. Este efeito não é observado aqui, talvez em razão de o TTL não ter aumentado o suficiente, talvez em razão da utilização de ACKs que reduz o congestionamento. Visto que não é objetivo deste trabalho, uma avaliação minuciosa do efeito de diferentes valores de TTL em DTNs fica em aberto e pode ser conduzida em trabalhos futuros.

6.3.5.1 Cenário Rollernet

As Figuras 6.35, 6.36 e 6.37 apresentam a taxa de entrega alcançada pelos protocolos Epidêmico, MaxProp e Prophet, respectivamente.



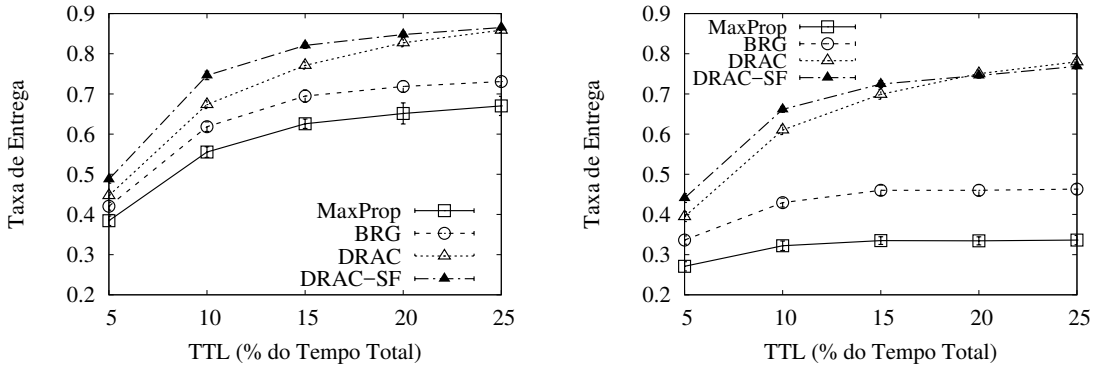
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.35: Taxa de entrega para o protocolo Epidêmico no cenário Rollernet.

As figuras mostram que a contramedida DRAC-SF leva a uma maior taxa de entrega em todas as circunstâncias avaliadas para o cenário Rollernet. Por sua vez, similarmente ao que foi demonstrado na avaliação do efeito do tamanho do *buffer* na taxa de entrega, a contramedida DRAC atinge um bom desempenho com o protocolo MaxProp, porém, não alcança bons resultados para os outros protocolos.

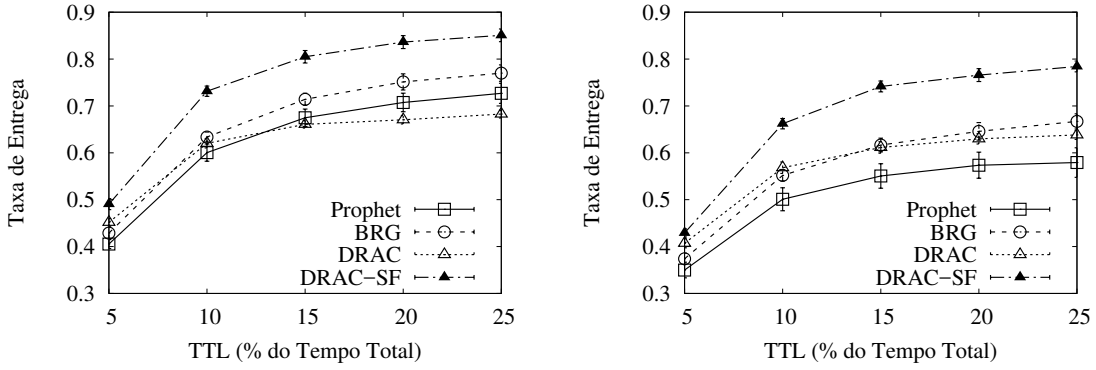
Especificamente para o protocolo MaxProp, contra o ataque de falsificação de reconhecimentos positivos com buraco negro, como ilustrado na Figura 6.36(b), as contramedidas



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.36: Taxa de entrega para o protocolo MaxProp no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

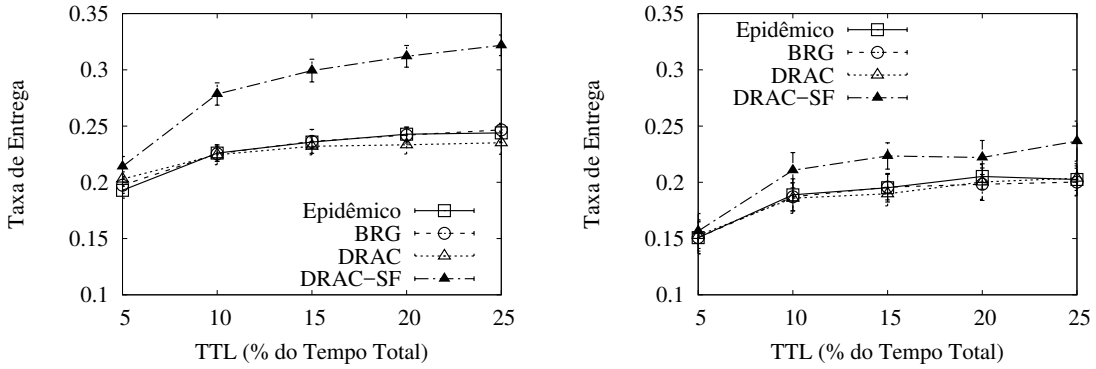
Figura 6.37: Taxa de entrega para o protocolo Prophet no cenário Rollernet.

propostas atingem aproximadamente 77% de taxa de entrega, contra 46% da contramedida BRG. Uma melhoria absoluta de cerca de 31%.

6.3.5.2 Cenário Dieselnet

Para o cenário Dieselnet, as Figuras 6.38, 6.39 e 6.40 apresentam a taxa de entrega alcançada pelos protocolos Epidêmico, MaxProp e Prophet, respectivamente.

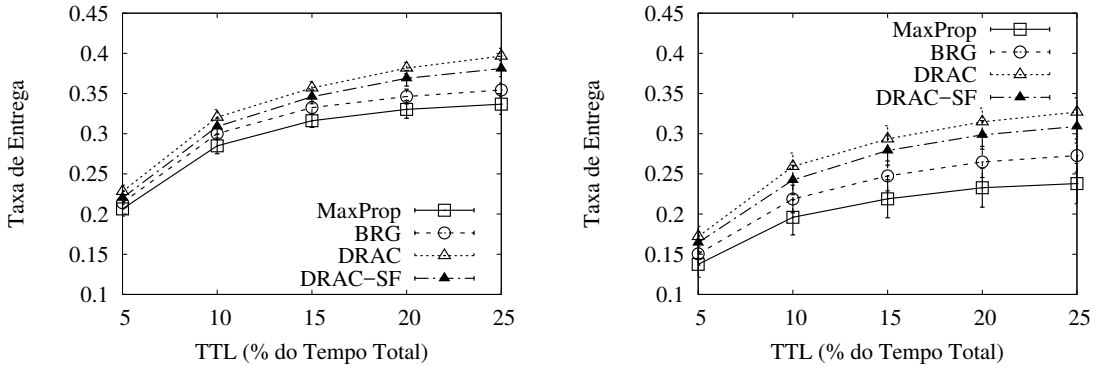
Para o protocolo Epidêmico, como ilustra a Figura 6.38, a contramedida DRAC-SF resulta em um melhor desempenho para ambos ataques, embora notavelmente mais expressivo para o ataque sem buraco negro. Por sua vez, as contramedidas DRAC e BRG não conseguem superar o protocolo Epidêmico sem contramedidas. A contramedida BRG não consegue identificar adequadamente quais ACKs são legítimos. O efeito disso é um desempenho similar ao protocolo Epidêmico sem contramedidas. Por sua vez, a



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.38: Taxa de entrega para o protocolo Epidêmico no cenário Dieselnet.



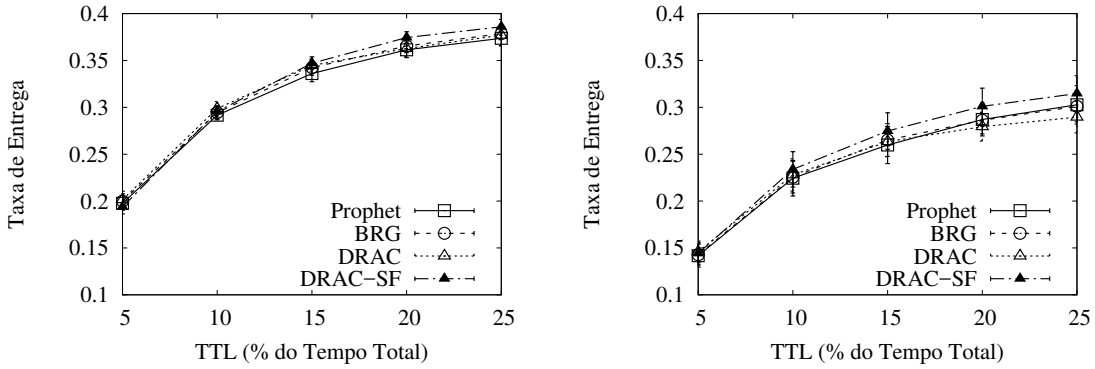
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.39: Taxa de entrega para o protocolo MaxProp no cenário Dieselnet.

contramedida DRAC continua a replicar as mensagens que foram reconhecidas, o que implica um maior congestionamento. Visto que este cenário é o menos conectado, o congestionamento tem consequências negativas com relação a taxa de entrega. Por outro lado, a contramedida DRAC-SF evita os reconhecimentos positivos forjados ao mesmo tempo que diminui o congestionamento ao interromper a replicação de mensagens que foram reconhecidas. Este equilíbrio leva esta proposta a atingir o melhor desempenho.

Para o protocolo MaxProp, como exibido na Figura 6.39, a contramedida DRAC-SF continua a superar a contramedida BRG, embora a diferença de desempenho não seja tão relevante quanto para o protocolo Epidêmico. Cabe observar que a contramedida DRAC superou a DRAC-SF neste cenário. Isto indica que o melhor controle de congestionamento realizado pelo protocolo MaxProp permite que a DRAC continue a replicar as mensagens para as quais reconhecimentos positivos foram recebidos sem prejudicar o desempenho da rede.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura 6.40: Taxa de entrega para o protocolo Prophet no cenário Dieselnet.

Por fim, para o protocolo Prophet, as diferenças de desempenho para todas as curvas são estatisticamente equivalentes. Isto pode indicar que o protocolo Prophet é mais apto a encontrar melhores caminhos quando as mensagens permanecem menos tempo nos *buffers* dos nós. Além disso, este comportamento pode significar que os ataques são menos efetivos neste cenário específico. De qualquer forma, é necessária investigação póstera adicional para elucidar este comportamento.

6.4 Conclusões

Dentre as propostas deste trabalho, destaca-se o desempenho da contramedida DRAC-SF, que supera a taxa de entrega obtida pela contramedida BRG, a principal contramedida contra o ataque de falsificação de reconhecimentos positivos da literatura, em 88% dos 336 cenários avaliados. É importante observar que se forem ignorados os resultados para o protocolo Spray and Wait e para os demais protocolos quando não existem nós maliciosos na rede, a proposta supera a BRG em 100% dos cenários.

A taxa de entrega também foi avaliada com diferentes tamanhos de *buffer*. Os resultados mostram que a contramedida DRAC-SF obtém o melhor desempenho. Além disso, os protocolos sem contramedidas e a contramedida BRG não são beneficiados pelo aumento do *buffer*. Isto ocorre porque a BRG e os protocolos sem contramedida são prejudicados pelos ataques, que removem muitas mensagens de seus *buffers*. A eficiência dos ataques fica clara quando mostra-se que a taxa de ocupação dos *buffers* nunca atinge valores próximos ao limite e que é menor ainda quando nós maliciosos estão realizando ataques.

A proposta DRAC-SF também mostra-se mais eficiente com relação a sobrecarga de entrega de mensagens, principalmente em cenários com maior conectividade. Especificamente para o protocolo MaxProp, a contramedida DRAC-SF obtém menores sobrecargas para todos os conjuntos de mobilidade utilizados na avaliação.

Com relação ao atraso de entrega, alguns autores relatam que em alguns casos os valores para esta métrica aumenta conforme também aumenta a taxa de entrega. Isto acontece porque os protocolos passam a entregar mensagens que antes não seriam entregues e que passam mais tempo armazenadas em *buffer*. Como a latência é calculada somente para mensagens entregues, estas mensagens contribuem para o aumento do atraso de entrega.

Capítulo 7

Conclusões e Considerações Finais

Duas contramedidas contra o ataque de falsificação de reconhecimentos positivos em DTNs foram propostas e avaliadas. Destacam-se os resultados da contramedida DRAC-SF, que obteve um desempenho superior a principal contramedida existente na literatura, a saber, BRG, em 88% dos 336 cenários avaliados neste trabalho. É importante apontar que a DRAC-SF não obtém resultados tão bons para o protocolo *Spray and Wait*. Isto acontece porque este protocolo é menos afetado pelo ataque de falsificação de reconhecimentos positivos e a proposta atua de modo conservador por não descartar mensagens para as quais reconhecimentos positivos são recebidos. No entanto, cabe destacar que conforme aumenta o número de nós maliciosos, a contramedida DRAC-SF passa a superar o desempenho da BRG em todos os cenários, exceto para um dos cenários avaliados. Este comportamento conservador também impacta negativamente nos cenários onde não existem nós maliciosos. No entanto, cabe ressaltar que se forem ignorados os resultados para o protocolo *Spray and Wait* e para os cenários onde não existem nós maliciosos na rede, a contramedida DRAC-SF passa a superar a BRG em 100% dos cenários avaliados.

Os ganhos alcançados pela contramedida DRAC-SF são expressivos. Por exemplo, para o protocolo MaxProp no cenário Infocom, a proposta alcança uma melhoria absoluta de 38%. Relativamente para este cenário, a melhoria chega a 135%. Adicionalmente, a contramedida DRAC foi proposta. Esta contramedida obtém um bom resultado para o protocolo MaxProp, no entanto, não é eficiente em protocolos com pouco controle de replicação, visto que acaba contribuindo para o congestionamento da rede. Além disso, duas avaliações a respeito da utilização de ACKs e dos efeitos do ataque de falsificação de reconhecimentos positivos foram realizadas. Cabe ressaltar que não foram encontradas na literatura avaliações tão amplas quanto as desenvolvidas neste trabalho, com o mesmo escopo.

A avaliação da utilização de reconhecimentos positivos mostrou que os ACKs melhoram a taxa de entrega e a eficiência da rede, pois reduzem a sobrecarga. Estes eram resultados esperados, mas que justificam a utilização de reconhecimentos positivos em vários protocolos para DTNs. Adicionalmente, observou-se que os protocolos que mais fazem uso de replicação são os mais beneficiados pelo uso de reconhecimentos positivos.

Posteriormente, observou-se que o desempenho dos protocolos de roteamento é comprometido pelo ataque de falsificação de reconhecimentos positivos. Além disso, a utilização de ataques conjuntos, como é o caso do ataque de falsificação de reconhecimentos positivos com buraco negro, mostrou-se mais prejudicial, visto que atua de modo mais agressivo, reduzindo a disponibilidade de réplicas na rede de maneira mais rápida. Com a avaliação dos resultados concluiu-se que os cenários mais conectados estão mais vulneráveis ao ataque, visto que as mensagens precisam alcançar os nós maliciosos para que estes possam forjar os ACKs. Analogamente, o protocolo *Spray and Wait* mostrou-se menos vulnerável ao ataque, devido ao seu modo de operação que controla estritamente o número de réplicas. Adicionalmente, o protocolo para de encaminhar mensagens quando estas atingem o limite de réplicas estabelecido, o que significa que existe mais chance de algumas mensagens nunca chegarem a nós maliciosos neste protocolo.

Finalmente, pretende-se estender este trabalho de forma a elaborar contramedidas adaptativas, que detectem a possibilidade de ataques e passem a agir desde então, para evitar os efeitos negativos apresentados principalmente quando nenhum nó malicioso está na rede mas as contramedidas estão em ação. Pretende-se desta forma tratar o problema de eficiência da contramedida DRAC-SF nos cenários onde não existem nós maliciosos e também para o protocolo *Spray and Wait*. Adicionalmente, pretende-se propor e avaliar novas contramedidas contra outros tipos de ataques realizados em DTNs.

Referências

[1] An, Y.; Huang, J.; Song, H.; Wang, J. A congestion level based end-to-end acknowledgement mechanism for delay tolerant networks. In *2012 IEEE Global Communications Conference (GLOBECOM)* (dec 2012), Institute of Electrical and Electronics Engineers (IEEE).

[2] Asokan, N.; Kostiainen, K.; Ginzboorg, P.; Ott, J.; Luo, C. Applicability of identity-based cryptography for disruption-tolerant networking. In *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking* (New York, NY, USA, 2007), MobiOpp '07, ACM, pp. 52–56.

[3] Asokan, N.; Kostiainen, K.; Ginzboorg, P.; Ott, J.; Luo, C. Towards securing disruption-tolerant networking. Tech. rep., Nokia Research Center, 2007.

[4] Balasubramanian, A.; Levine, B.; Venkataramani, A. Dtn routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev. 37*, 4 (Aug. 2007), 373–384.

[5] Bhutta, N.; Ansa, G.; Johnson, E.; Ahmad, N.; Alsiyabi, M.; Cruickshank, H. Security analysis for delay/disruption tolerant satellite and sensor networks. In *Proceedings of the International Workshop on Satellite and Space Communications* (2009).

[6] Birrane, E. "Streamlined" bundle security protocol. Lecture notes.

[7] Birrane, E. Streamlined bundle security protocol specification. Internet-Draft, December 2014. draft-birrane-dtn-sbsp-00.

[8] Birrane, E. Streamlined bundle security protocol (SBSP) discussion: History, recommendations, implementation, status, and todo. Lecture notes, March 2015.

[9] Birrane, E.; McKeever, K. Bundle protocol security specification. Internet-Draft draft-ietf-dtn-bpsec-05, IETF Secretariat, July 2017. `http://www.ietf.org/internet-drafts/draft-ietf-dtn-bpsec-05.txt`.

[10] Bitsch Link, J. A.; Viol, N.; Goliath, A.; Wehrle, K. Simbetage: utilizing temporal changes in social networks for pocket switched networks. In *Proceedings of the 1st ACM workshop on User-provided networking: challenges and opportunities* (New York, NY, USA, 2009), U-NET '09, ACM, pp. 13–18.

[11] Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *SIAM J. of Computing 32*, 3 (2003), 586–615. extended abstract in Crypto'01.

[12] Burgess, J.; Bissias, G. D.; Corner, M. D.; Levine, B. N. Surviving attacks on disruption-tolerant networks without authentication. In *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2007), MobiHoc '07, ACM, pp. 61–70.

[13] Burgess, J.; Gallagher, B.; Jensen, D.; Levine, B. N. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM* (April 2006).

[14] Burleigh, S. Bundle protocol. Internet-Draft, June 2015. draft-dtnwg-bp-00.

[15] Burleigh, S.; Ramadas, M.; Farrell, S. Licklider Transmission Protocol - Motivation. RFC 5325 (Informational), Sept. 2008.

[16] Burns, B.; Brock, O.; Levine, B. N. MV Routing and Capacity Building in Disruption Tolerant Networks. In *Proc. IEEE INFOCOM* (2005), pp. 398–408.

[17] Caini, C.; Cruickshank, H. S.; Farrell, S.; Marchese, M. Delay- and disruption-tolerant networking (DTN): An alternative solution for future satellite networking applications. *Proceedings of the IEEE 99*, 11 (2011), 1980–1997.

[18] Cao, Y.; Sun, Z. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys Tutorials, IEEE PP*, 99 (2012), 1 –24.

[19] Cao, Y.; Sun, Z. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys & Tutorials, IEEE 15*, 2 (2013), 654–677.

[20] Cerf, V.; Burleigh, S.; Hooke, A.; Torgerson, L.; Durst, R.; Scott, K.; Fall, K.; Weiss, H. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), Apr. 2007.

[21] Chen, B. B.; Chan, M. C. Mobicent: a credit-based incentive system for disruption tolerant network. In *Proceedings of the IEEE INFOCOM* (2010).

[22] Chen, C.; Tang, S.; Mitchell, C. J. Building general-purpose security services on emv payment cards. In *SecureComm* (2012), A. D. Keromytis and R. D. Pietro, Eds., vol. 106 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, pp. 29–44.

[23] Chen, L.-J.; Chiou, C.-L.; Chen, Y.-C. An evaluation of routing reliability in non-collaborative opportunistic networks. In *International Conference on Advanced Information Networking and Applications* (2009).

[24] Chen, X.; Sun, L.; Ma, J.; Ma, Z. A trust management scheme based on behavior feedback for opportunistic networks. *China Communications* (2015).

[25] Choo, F. C.; Chan, M. C.; Chang, E.-C. Robustness of DTN against routing attacks. In *Proceedings of the 2Nd International Conference on COMmunication Systems and NETworks* (Piscataway, NJ, USA, 2010), COMSNETS'10, IEEE Press, pp. 148–157.

[26] Cocks, C. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding* (London, UK, UK, 2001), Springer-Verlag, pp. 360–363.

[27] Daly, E.; Haahr, M. Social network analysis for information flow in disconnected delay-tolerant manets. *Mobile Computing, IEEE Transactions on 8*, 5 (may 2009), 606 –621.

[28] Daly, E. M.; Haahr, M. Social network analysis for routing in disconnected delay-tolerant MANETs. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing* (Montreal, Quebec, Canada, 2007), ACM Press, pp. 32–40.

[29] de Oliveira, E. C. R.; de Albuquerque, C. V. N. Análise do protocolo nectar em cenários com mobilidade e frequentes interrupções. In *Anais do 27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2009).

[30] de Oliveira, E. C. R.; de Albuquerque, C. V. N. Nectar: a dtn routing protocol based on neighborhood contact history. In *Proceedings of the 2009 ACM symposium on Applied Computing* (New York, NY, USA, 2009), SAC '09, ACM, pp. 40–46.

[31] de Oliveira, E. C. R.; Silva, E. F.; Passos, D.; Naves, J. F.; Muchaluat-Saade, D. C.; Moraes, I. M.; Albuquerque, C. Context-Aware Routing in Delay and Disruption Tolerant Networks. *International Journal of Wireless Information Networks 23*, 3 (2016), 231–245.

[32] Diep, P. T. N.; Yeo, C. K. Detecting colluding blackhole and greyhole attack in delay tolerant networks. In *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)* (2015).

[33] Dini, G.; Duca, A. L. Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network. *Ad Hoc Networks 10*, 7 (2012), 1167 – 1178.

[34] Douceur, J. R. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems* (London, UK, UK, 2002), IPTPS '01, Springer-Verlag, pp. 251–260.

[35] Fall, K. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (New York, NY, USA, 2003), SIGCOMM '03, ACM, pp. 27–34.

[36] Fall, K.; Farrell, S. DTN: An architectural retrospective. *IEEE J.Sel. A. Commun. 26*, 5 (June 2008), 828–836.

[37] Fall, K.; Hong, W.; Madden, S. Custody Transfer for Reliable Delivery in Delay Tolerant Networks. Tech. rep., 2003.

[38] Farrell, S. Security in the wild. *IEEE Internet Computing 15*, 3 (2011), 86–91.

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

[40] Farrell, S.; Cahill, V.; Geraghty, D.; Humphreys, I.; McDonald, P. When tcp breaks: Delay- and disruption- tolerant networking. *IEEE Internet Computing 10*, 4 (July 2006), 72–78.

[41] Farrell, S.; Ramadas, M.; Burleigh, S. Licklider Transmission Protocol - Security Extensions. RFC 5327 (Experimental), Sept. 2008.

[42] Farrell, S.; Symington, S. F.; Weiss, H.; Lovell, P. Delay-tolerant networking security overview. Internet-Draft, September 2009. draft-irtf-dtnrg-sec-overview-06.

[43] Friedman, A.; Birrane, E. Secure delay tolerant networking using sbsp and ipmeir: Enabling security, resiliency, and cost savings for space mission communications. Lecture notes.

[44] Galati, A.; Djemame, K.; Greenhalgh, C. Analysis of human mobility patterns for opportunistic forwarding in shopping mall environments. *Social Network Analysis and Mining 5*, 1 (apr 2015), 1–14.

[45] Gardner, M. Mathematical games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American 223* (1970), 120–123.

[46] Ginzboorg, P. *Security mechanisms in partially isolated networks.* Tese de Doutorado, Department of Communications and Networking, School of Electrical Engineering, Aalto University, 2014.

[47] Grasic, S.; Davies, E.; Lindgren, A.; Doria, A. The evolution of a dtn routing protocol – prophetv2. In *Proceedings of the 6th ACM workshop on Challenged networks* (2011), pp. 27–30.

[48] Gupta, A. K.; Bhattacharya, I.; Banerjee, P. S.; Mandal, J. K. A cooperative approach to thwart selfish and black-hole attacks in DTN for post disaster scenario. In *International Conference of Emerging Applications of Information Technology* (2014).

[49] Hui, P.; Chaintreau, A.; Scott, J.; Gass, R.; Crowcroft, J.; Diot, C. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking* (New York, NY, USA, 2005), WDTN '05, ACM, pp. 244–251.

[50] Hui, P.; Chaintreau, A.; Scott, J.; Gass, R.; Crowcroft, J.; Diot, C. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (New York, NY, USA, 2005), WDTN '05, ACM, pp. 244–251.

[51] Hui, P.; Crowcroft, J.; Yoneki, E. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing* (New York, NY, USA, 2008), MobiHoc '08, ACM, pp. 241–250.

[52] Jain, S.; Demmer, M.; Patra, R.; Fall, K. Using redundancy to cope with failures in a delay tolerant network. *SIGCOMM Comput. Commun. Rev. 35*, 4 (Aug. 2005), 109–120.

[53] Jain, S.; Fall, K.; Patra, R. Routing in a delay tolerant network. *SIGCOMM Comput. Commun. Rev. 34*, 4 (Aug. 2004), 145–158.

[54] Jain, S.; Fall, K.; Patra, R. Routing in a delay tolerant network. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2004), SIGCOMM '04, ACM, pp. 145–158.

[55] Jones, E. P. C.; Li, L.; Ward, P. A. S. Practical routing in delay-tolerant networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (New York, NY, USA, 2005), WDTN '05, ACM, pp. 237–243.

[56] Karvo, J.; Ott, J. Time scales and delay-tolerant routing protocols. *Proceedings of the third ACM workshop on Challenged networks CHANTS 08* (2008), 33–40.

[57] Kate, A.; Zaverucha, G. M.; Hengartner, U. Anonymity and security in delay tolerant networks. *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm* (2007), 504–513.

[58] Keränen, A.; Ott, J.; Kärkkäinen, T. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques* (New York, NY, USA, 2009), ICST.

[59] Khabbaz, M.; Assi, C. M.; Fawaz, W. Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges. *IEEE Communications Surveys and Tutorials* (2012), 607–640.

[60] Laitinen, P.; Ginzboorg, P.; Asokan, N.; Holtmanns, S.; Niemi, V. Extending cellular authentication as a service. In *Proceedings of the 1st IEEE Conference on Commercialising Technology and Inovation* (2005).

[61] Lee, F. C.; Goh, W.; Yeo, C. K. A queuing mechanism to alleviate flooding attacks in probabilistic delay tolerant networks. In *Proceedings of the 2010 Sixth Advanced International Conference on Telecommunications* (Washington, DC, USA, 2010), AICT '10, IEEE Computer Society, pp. 329–334.

[62] Li, F.; Wu, J.; Srinivasan, A. Thwarting blackhole attacks in distruption-tolerant networks using encounter tickets. In *Proceedings of the 28th IEEE INFOCOM Conference* (2009).

[63] Li, N.; Das, S. K. A trust-based framework for data forwarding in opportunistic networks. *Ad Hoc Networks 11*, 4 (2013), 1497 – 1509. 1. System and Theoretical Issues in Designing and Implementing Scalable and Sustainable Wireless Sensor Networks 2. Wireless Communications and Networking in Challenged Environments.

[64] Li, Q.; Gao, W.; Zhu, S.; Cao, G. To lie or to comply: Defending against flood attacks in disruption tolerant networks. *IEEE Transactions on Dependable and Secure Computing 10*, 3 (2013), 168–182.

[65] Lindgren, A.; Doria, A.; Schelén, O. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev. 7* (July 2003), 19–20.

[66] Lindgren, A.; Hui, P. The quest for a killer app for opportunistic and delay tolerant networks: (invited paper). In *Proceedings of the 4th ACM Workshop on Challenged Networks* (New York, NY, USA, 2009), CHANTS '09, ACM, pp. 59–66.

[67] Ma, D.; Tsudik, G. Security and privacy in emerging wireless networks. *Wireless Commun. 17*, 5 (Oct. 2010), 12–21.

[68] Mathurapoj, A.; Pornavalai, C.; Chakraborty, G. Fuzzy-spray: Efficient routing in delay tolerant ad-hoc network based on fuzzy decision mechanism. In *2009 IEEE International Conference on Fuzzy Systems* (aug 2009), Institute of Electrical and Electronics Engineers (IEEE).

[69] Miranda, E. D. S.; Naves, J. F.; Moraes, I. M.; Velloso, P. B. A joint custody-based forwarding policy for delay-tolerant networks. *2012 Global Information Infrastructure and Networking Symposium (GIIS)* (dec 2012), 1–6.

[70] Mota, V.; Macedo, D. F.; Ghamri-Doudane, Y.; Nogueira, J. M. Mineiro: Um mecanismo de incentivo para aplicações em redes oportunísticas. In *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2015).

[71] N4C Project. Networking for communications challenged communities. http://www.n4c.eu/. Último acesso em 01/02/2017.

[72] Nagrath, P.; Aneja, S.; Gupta, N.; Madria, S. Protocols for mitigating blackhole attacks in delay tolerant networks. *Wireless Networks* (2015), 1–12.

[73] Nagrath, P.; Aneja, S.; Purohit, G. N. Flooding attack in delay tolerant network. *International Journal of Emerging Technology and Advanced Engineering 4*, 7 (July 2014).

[74] Nagrath, P.; Aneja, S.; Purohit, G. N. Defending flooding attack in delay tolerant networks. In *2015 International Conference on Information Networking, ICOIN 2015, Siem Reap, Cambodia, January 12-14, 2015* (2015), pp. 40–45.

[75] Natarajan, V.; Yang, Y.; Zhu, S. Resource-misuse attack detection in delay-tolerant networks. In *Proceedings of the 30th IEEE International Performance Computing and Communications Conference* (Washington, DC, USA, 2011), PCCC '11, IEEE Computer Society, pp. 1–8.

[76] Naves, J. F.; Moraes, I. M. Mitigating the ACK Counterfeiting Attack in Delay and Disruption Tolerant Networks. In *Proceedings of the 22nd IEEE Symposium on Computers and Communications* (2017), pp. 1018–1023.

[77] Naves, J. F.; Moraes, I. M. Um Mecanismo Eficiente de Controle de Congestionamento para Redes Tolerantes a Atrasos e Desconexões. In *Anais do XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2017), pp. 315–328.

[78] Naves, J. F.; Moraes, I. M.; Albuquerque, C. LPS and LRF: Efficient buffer management policies for delay and disruption tolerant networks. In *Proceedings of the 2012 IEEE 37th Conference on Local Computer Networks (LCN 2012)* (Washington, DC, USA, oct 2012), LCN '12, IEEE Computer Society, pp. 368–375.

[79] Naves, J. F.; Moraes, I. M.; de Albuquerque, C. V. N. LPS e LRF: Políticas de gerenciamento de buffer eficientes para redes tolerantes a atrasos e desconexões. In *Anais do 30º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2012), pp. 233–246.

[80] Nelson, S. C.; Bakht, M.; Kravets, R. Encounter-based routing in DTNs. In *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM)* (2009).

[81] Nunes, C. M.; Link, E.; Dotti, F. L. Evaluating the impact of an acknowledgment strategy for APRP. In *Proceedings of the 5th International Latin American Networking Conference on - LANC '09* (2009), Association for Computing Machinery (ACM).

[82] Oliveira, C. T.; Moreira, M. D. D.; Rubinstein, M. G.; Costa, L. H. M. K.; B.Duarte, O. C. M. Redes tolerantes a atrasos e desconexões. In *Minicursos do Simpósio Brasileiro de Redes de Computadores* (May 2007).

[83] Ott, J.; Keränen, A.; Hyytiä, E. Beachnet: Propagation-based information sharing in mostly static networks. In *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition* (New York, NY, USA, 2011), ExtremeCom '11, ACM, pp. 15:1–15:6.

[84] Parris, I.; Henderson, T. The impact of location privacy on opportunistic networks. In *WoWMoM* (2011), IEEE, pp. 1–6.

[85] Parris, I.; Henderson, T. Privacy-enhanced social-network routing. *Comput. Commun. 35*, 1 (Jan. 2012), 62–74.

[86] Parris, I.; Henderson, T. Friend or flood? social prevention of flooding attacks in mobile opportunistic networks. In *Proceedings of the IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)* (2014).

[87] Partridge, C. Authentication for fragments. In *Proceedings of ACM SIGCOMM HotNets-IV workshop* (2005).

[88] Pereira, P. P.; Casaca, A.; Rodrigues, J. R.; Soares, V. S.; Triay, J. T.; Pastor, C. C. P. From delay-tolerant networks to vehicular delay-tolerant networks. *IEEE Communications Surveys and Tutorials 14*, 4 (January 2012), 1166–1182.

[89] Phe-Neau, T.; Dias de Amorim, M.; Campista, M. E. M.; Conan, V. Examining vicinity dynamics in opportunistic networks. In *Proceedings of the 8th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks* (New York, NY, USA, 2013), PM2HW2N '13, ACM, pp. 153–160.

[90] Pitkanen, M.; Keranen, A.; Ott, J. Message fragmentation in opportunistic DTNs. In *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks* (Washington, DC, USA, 2008), WOWMOM '08, IEEE Computer Society, pp. 1–7.

[91] Ramadas, M.; Burleigh, S.; Farrell, S. Licklider Transmission Protocol - Specification. RFC 5326 (Experimental), Sept. 2008.

[92] Raveneau, P.; Dhaou, R.; Chaput, E.; Beylot, A.-L. DTNs back: DTNs broadcasting ACK. In *2014 IEEE Global Communications Conference* (dec 2014), Institute of Electrical and Electronics Engineers (IEEE).

[93] Ren, Y.; Chuah, M. C.; Yang, J.; Chen, Y. Detecting blackhole attacks in disruption-tolerant networks through packet exchange recording. In *Proceedings of the 2010 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)* (2010).

[94] Ren, Y.; Chuah, M. C.; Yang, J.; Chen, Y. Detecting wormhole attacks in delay-tolerant networks. *Wireless Commun. 17*, 5 (Oct. 2010), 36–42.

[95] Ren, Y.; Chuah, M. C.; Yang, J.; Chen, Y. Muton: Detecting malicious nodes in disruption-tolerant networks. In *Proceedings of the 2010 IEEE Wireless Communications and Networking Conference (WCNC)* (2010).

[96] Sandulescu, G.; Nadjm-Tehrani, S. Opportunistic dtn routing with window-aware adaptive replication. In *Proceedings of the 4th Asian Conference on Internet Engineering* (New York, NY, USA, 2008), AINTEC '08, ACM, pp. 103–112.

[97] Scott, K.; Burleigh, S. Bundle Protocol Specification. RFC 5050 (Experimental), Nov. 2007.

[98] Seligman, M.; Fall, K.; Mundur, P. Storage routing for DTN congestion control. *Wireless Communications and Mobile Computing 7*, 10 (2007), 1183–1196.

[99] Sermpezis, P.; Spyropoulos, T. Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks. *Computer Communications 48*, 0 (2014), 71 – 83. Opportunistic networks.

[100] Seth, A.; Keshav, S. Practical security for disconnected nodes. In *Proceedings of the First International Conference on Secure Network Protocols* (Washington, DC, USA, 2005), NPSEC'05, IEEE Computer Society, pp. 31–36.

[101] Shamir, A. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in Cryptology* (New York, NY, USA, 1985), Springer-Verlag New York, Inc., pp. 47–53.

[102] Shevade, U.; Song, H. H.; Qiu, L.; Zhang, Y. Incentive-aware routing in DTNs. *Proceedings - International Conference on Network Protocols, ICNP* (2008), 238–247.

[103] Silva, A. P.; Burleigh, S.; Hirata, C. M.; Obraczka, K. DTN congestion control unplugged: A comprehensive performance study. In *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks* (2015), pp. 43–48.

[104] Silva, A. P.; Burleigh, S.; Hirata, C. M.; Obraczka, K. A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks 25, Part B*, 0 (2015), 480 – 494.

[105] Silva, A. P.; Burleigh, S.; Hirata, C. M.; Obraczka, K. A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks 25, Part B*, 0 (2015), 480 – 494. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication.

[106] Silva, A. P.; Burleigh, S.; Hirata, C. M.; Obraczka, K. Congestion control in disruption-tolerant networks: A comparative study for interplanetary and terrestrial networking applications. *Ad Hoc Networks 44* (2016), 1–18.

[107] Soelistijanto, B.; Howarth, M. Transfer reliability and congestion control strategies in opportunistic networks: a survey. *IEEE Communications Surveys and Tutorials 16*, 1 (2014), 538 – 555.

[108] Solis, J.; Ginzboorg, P.; Asokan, N.; Ott, J. Best-effort authentication for opportunistic networks. In *International Performance Computing and Communications Conference* (2011), S. Zhong, D. Dou, and Y. W. 0003, Eds., IEEE, pp. 1–6.

[109] Spyropoulos, T.; Psounis, K.; Raghavendra, C. S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN* (Aug. 2005), pp. 252–259.

[110] Symington, S. Delay-Tolerant Networking Metadata Extension Block. RFC 6258 (Experimental), May 2011.

[111] Symington, S.; Farrell, S.; Weiss, H.; Lovell, P. Bundle Security Protocol Specification. RFC 6257 (Experimental), May 2011.

[112] Templin, F.; Burleigh, S. DTN security key management - requirements and design. Internet-Draft, February 2015. draft-templin-dtnskmreq-00.

[113] Tournoux, P.-U.; Leguay, J.; Benbadis, F.; Conan, V.; de Amorim, M. D.; Whitbeck, J. The accordion phenomenon: Analysis, characterization, and impact on dtn routing. In *Proceedings of the 28th IEEE INFOCOM* (apr 2009), Institute of Electrical and Electronics Engineers (IEEE), pp. 1116–1124.

[114] Trifunovic, S.; Hossmann-Picu, A. Stalk me if you can: The anatomy of sybil attacks in opportunistic networks. In *Proceedings of the 9th ACM MobiCom Workshop on Challenged Networks* (New York, NY, USA, 2014), CHANTS '14, ACM, pp. 37–42.

[115] Trifunovic, S.; Hossmann-Picu, A. Stalk and lie—the cost of sybil attacks in opportunistic networks. *Computer Communications* (2015).

[116] Uddin, M. Y. S.; Khurshid, A.; Jung, H. D.; Gunter, C.; Caesar, M.; Abdelzaher, T. Making DTNs robust against spoofing attacks with localized countermeasures. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)* (2011).

[117] Vahdat, A.; Becker, D. Epidemic routing for partially-connected ad hoc networks. Tech. rep., Duke University, July 2000.

[118] Van Besien, W. L. Dynamic, non-interactive key management for the bundle protocol. In *Proceedings of the 5th ACM Workshop on Challenged Networks* (New York, NY, USA, 2010), CHANTS '10, ACM, pp. 75–78.

[119] Wood, L.; Eddy, W. M.; Holliday, P. A bundle of problems. In *Proceedings of the IEEE Aerospace Conference* (2009).

[120] Yinghui Guo, Sebastian Schildt, T. P.; Wolf, L. Detecting malicious behavior in a vehicular DTN for public transportation. In *Global Information Infrastructure Symposium* (2013).

[121] Yuan, Q.; Cardei, I.; Wu, J. Predict and relay: An efficient routing in disruption-tolerant networks. In *Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (New York, NY, USA, 2009), MobiHoc '09, ACM, pp. 95–104.

[122] Zhang, X.; Kurose, J.; Levine, B. N.; Towsley, D.; Zhang, H. Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking* (New York, NY, USA, 2007), MobiCom '07, ACM, pp. 195–206.

[123] Zhang, Z. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials 8*, 1-4 (2006), 24–37.

[124] Zhu, H. *Security in Delay Tolerant Networks.* Tese de Doutorado, University of Waterloo, 2009.

[125] Zhu, H.; Du, S.; Gao, Z.; Dong, M.; Cao, Z. A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks. *IEEE Transactions on Parallel and Distributed Systems 25*, 1 (2014), 22–32.

[126] Zhu, H.; Lin, X.; Lu, R.; Shen, X.; Xing, D.; Cao, Z. An opportunistic batch bundle authentication scheme for energy constrained DTNs. In *INFOCOM* (2010), IEEE, pp. 605–613.

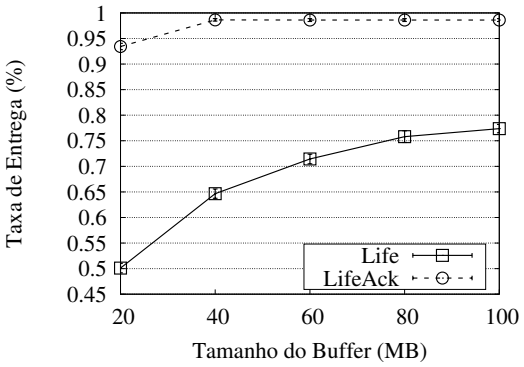
APÊNDICE A – Avaliação do Uso de Reconhecimentos Positivos

Este apêndice reúne resultados que foram omitidos na Seção 4.4 do Capítulo 4, que trata da avaliação do uso de reconhecimentos positivos em DTNs. Estes resultados foram omitidos para evitar a repetição desnecessária, visto que são similares com os resultados que foram apresentados. Para a completude deste trabalho, todos os resultados que foram omitidos na referida seção são exibidos a seguir, a saber, os resultados omitidos para os cenários Rollernet e Dieselnet e todos os resultados dos cenários Infocom05 e Shoppingmall.

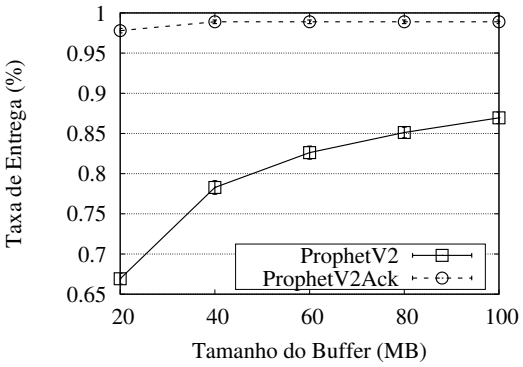
A.1 Taxa de Entrega

Tabela A.1: Melhoria na taxa de entrega para o cenário Infocom05.

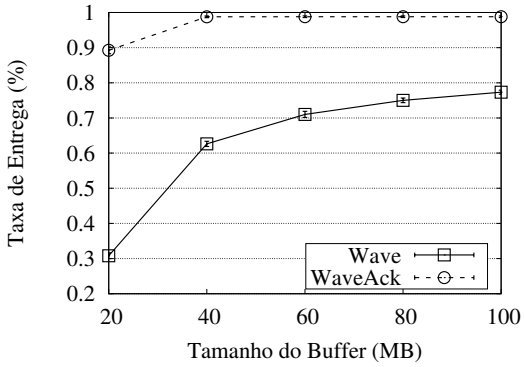
	Melhoria Absoluta (%)					**Melhoria Relativa (%)**				
Protocolo/Buffer (MB)	**20**	**40**	**60**	**80**	**100**	**20**	**40**	**60**	**80**	**100**
Epidêmico	17	30	37	40	40	64	83	87	86	80
Life	19	31	37	38	37	65	79	81	78	69
MaxProp	25	17	14	12	12	44	24	18	16	15
Prophet	24	32	36	38	37	67	74	74	74	70
ProphetV2	18	25	25	24	21	36	43	40	37	32
SnW	9	5	2	1	1	13	7	3	2	1
Wave	19	32	40	41	40	86	93	99	88	80



(a) Life



(b) ProphetV2

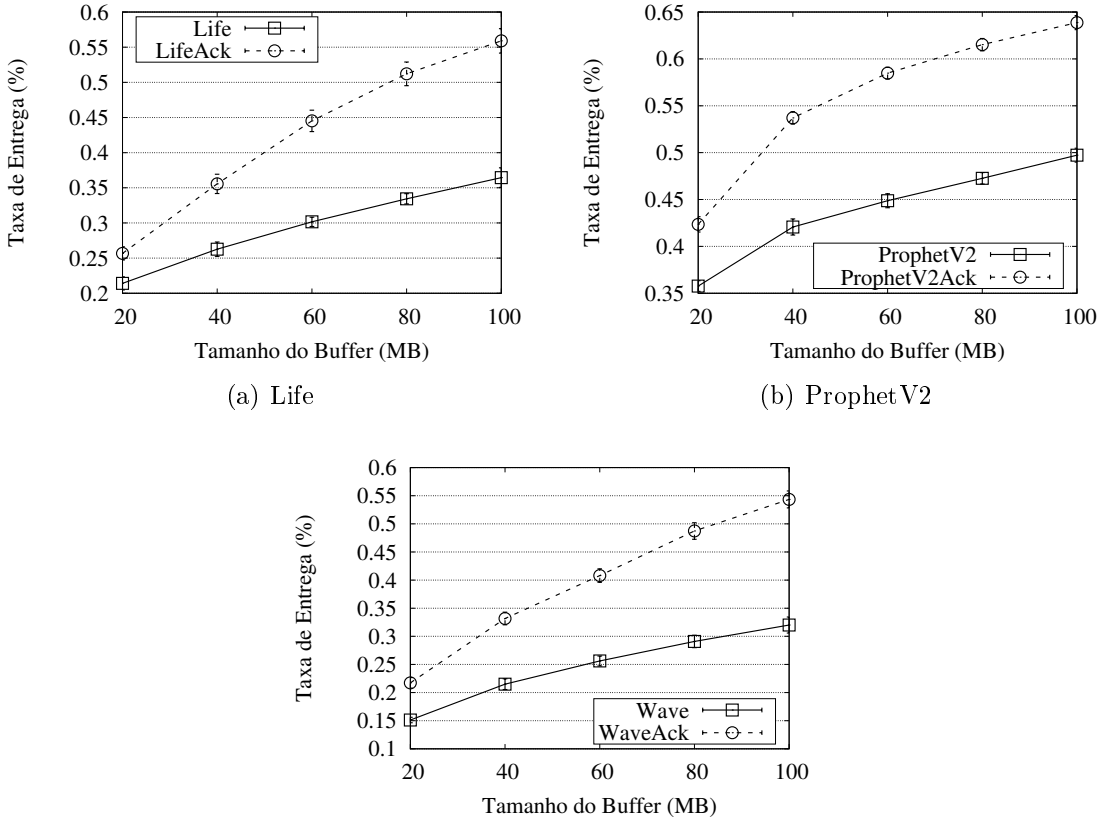


(c) Wave

Figura A.1: Taxa de entrega para o cenário Rollernet.

Tabela A.2: Melhoria na taxa de entrega para o cenário Shoppingmall.

	Melhoria Absoluta (%)					**Melhoria Relativa (%)**				
Protocolo/Buffer (MB)	**20**	**40**	**60**	**80**	**100**	**20**	**40**	**60**	**80**	**100**
Epidêmico	34	42	44	45	42	85	90	86	83	72
Life	19	25	26	26	23	30	35	37	35	30
MaxProp	10	6	5	4	4	12	6	5	5	4
Prophet	36	39	38	36	34	68	68	64	57	50
ProphetV2	21	20	18	16	14	29	26	23	19	17
SnW	4	1	1	0	0	5	1	1	0	0
Wave	56	54	51	45	42	335	159	115	82	72

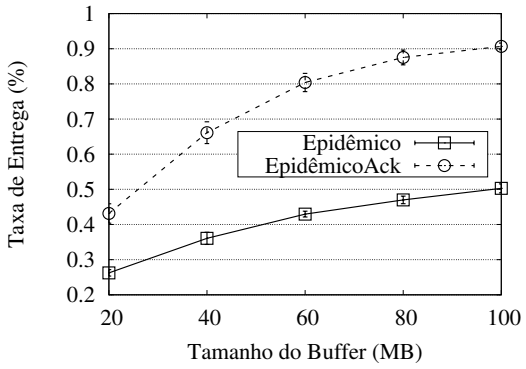


(a) Life

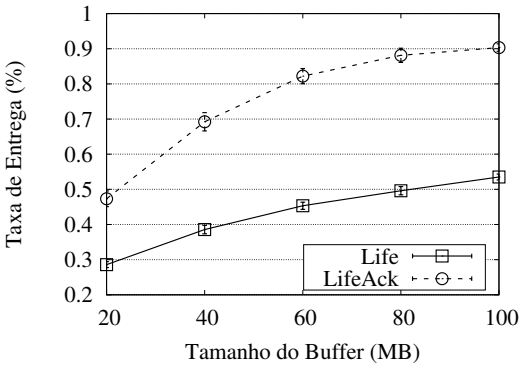
(b) ProphetV2

(c) Wave

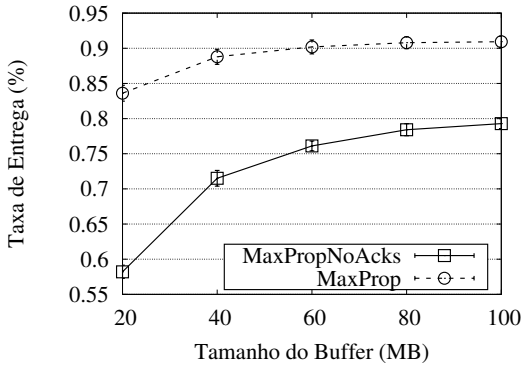
Figura A.2: Taxa de entrega para o cenário Dieselnet.



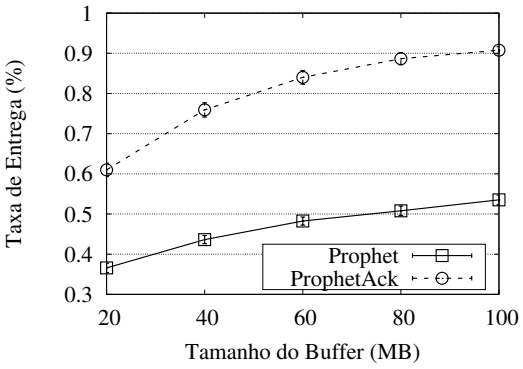
(a) Epidêmico



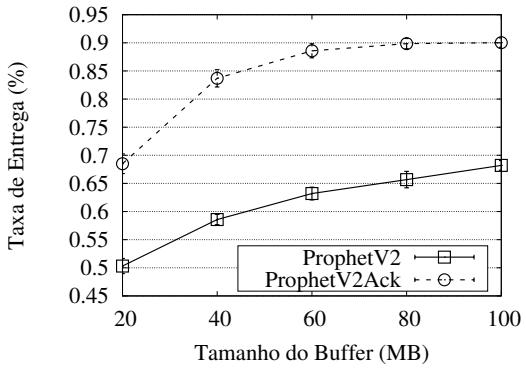
(b) Life



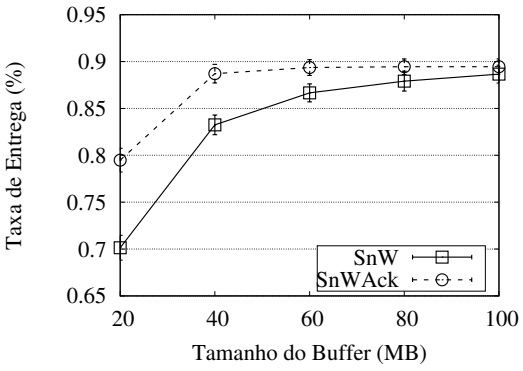
(c) MaxProp



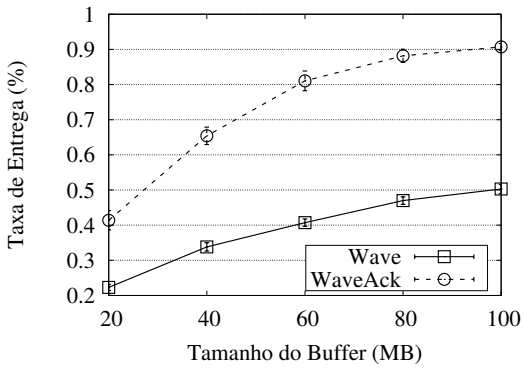
(d) Prophet



(e) ProphetV2

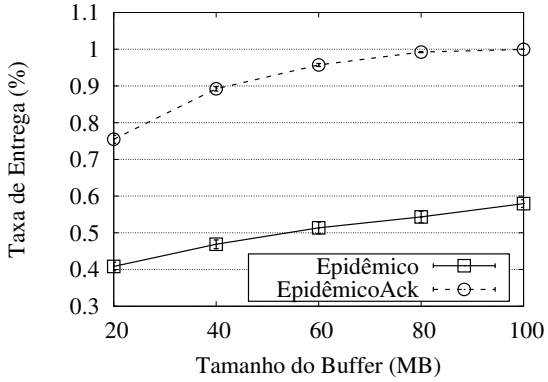


(f) Spray and Wait

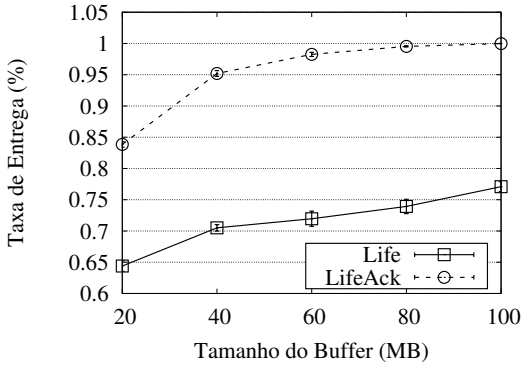


(g) Wave

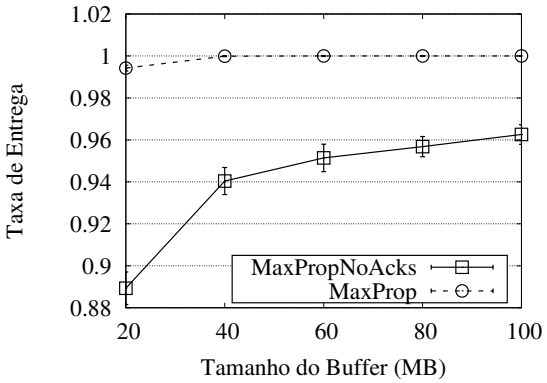
Figura A.3: Taxa de entrega para o cenário Infocom05.



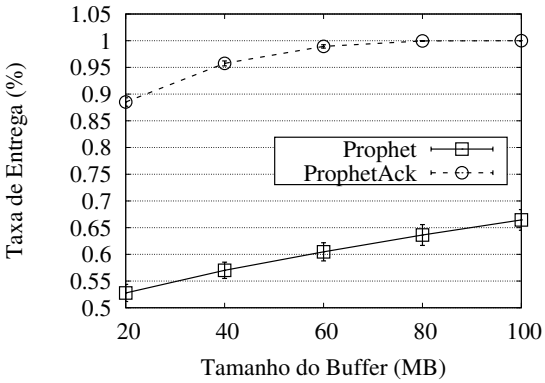
(a) Epidêmico



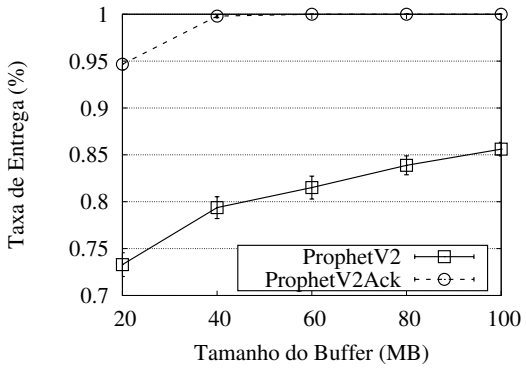
(b) Life



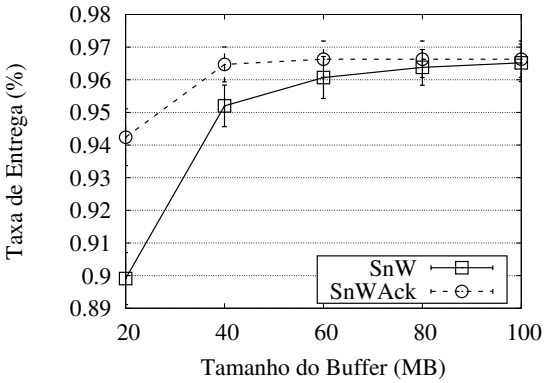
(c) MaxProp



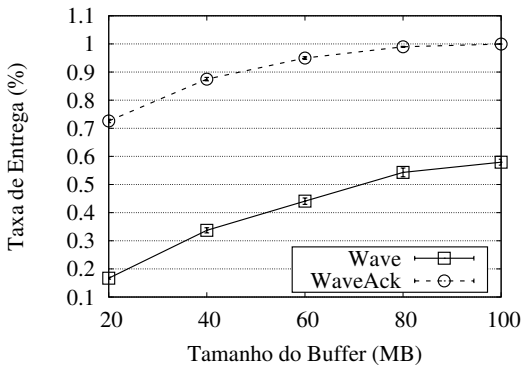
(d) Prophet



(e) ProphetV2



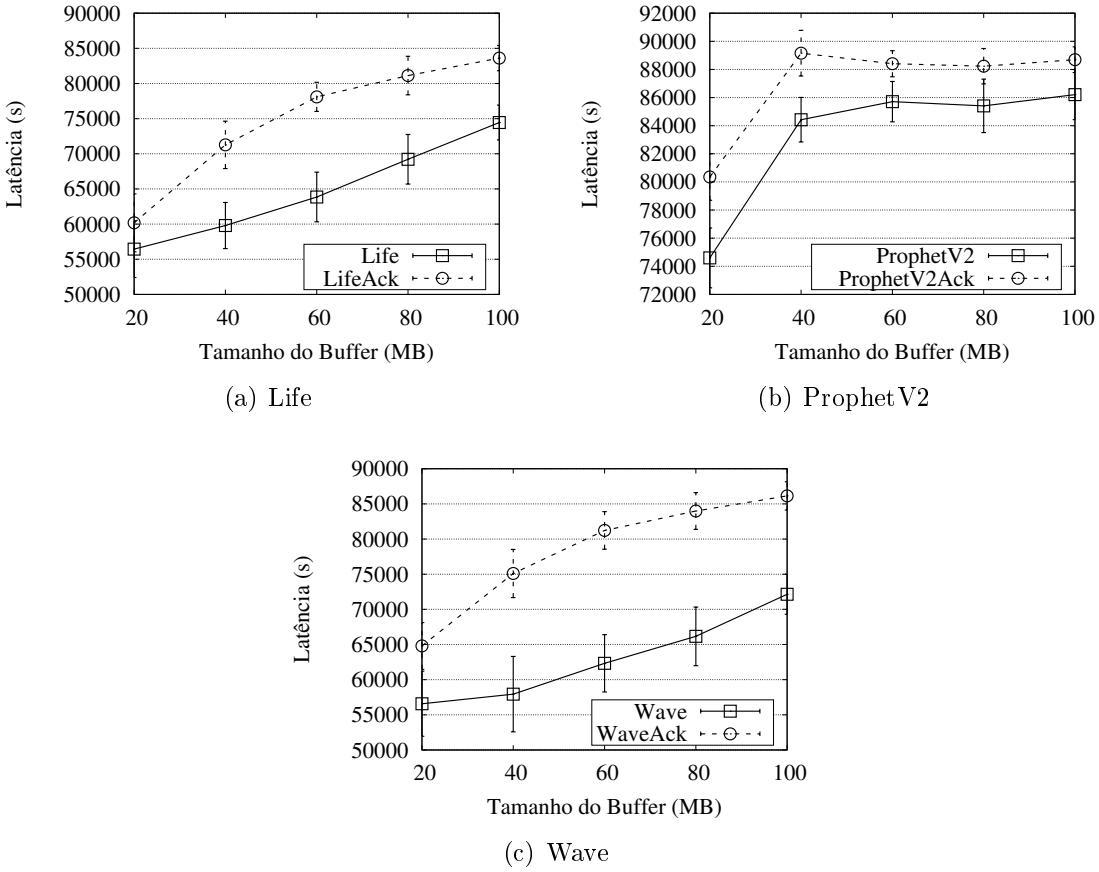
(f) Spray and Wait



(g) Wave

Figura A.4: Taxa de entrega para o cenário ShoppingMall.

A.2 Atraso de Entrega

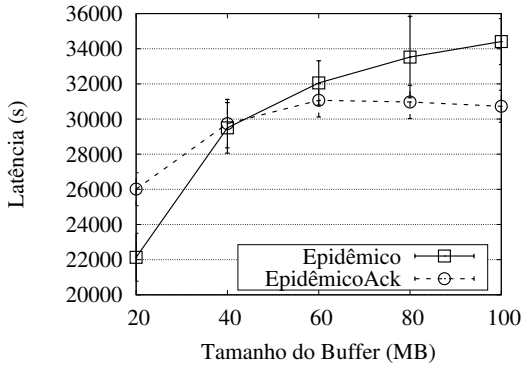


(a) Life

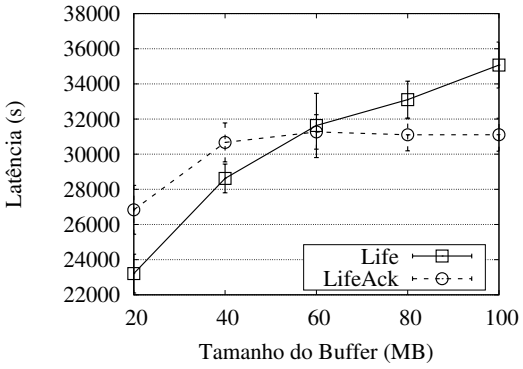
(b) ProphetV2

(c) Wave

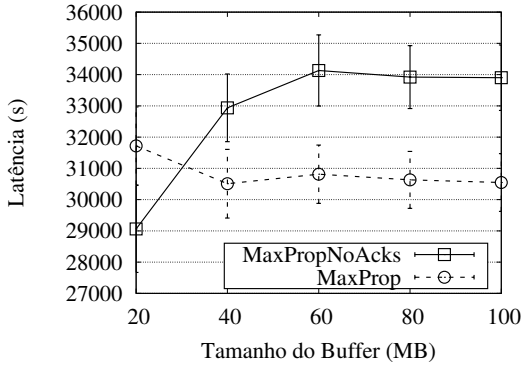
Figura A.5: Atraso de entrega para o cenário Dieselnet.



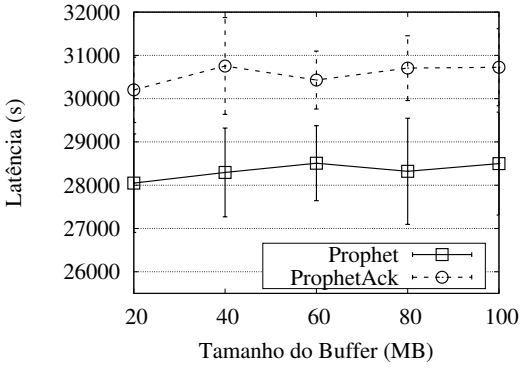
(a) Epidêmico



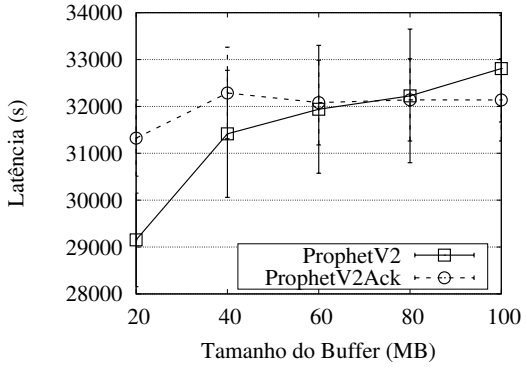
(b) Life



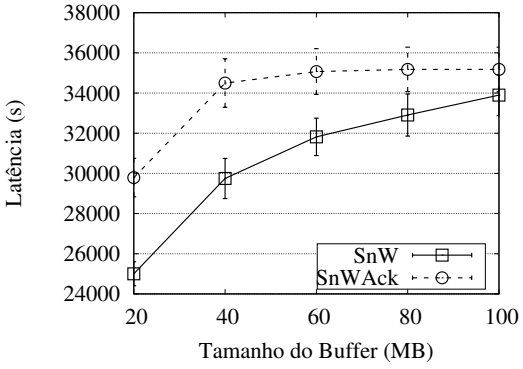
(c) MaxProp



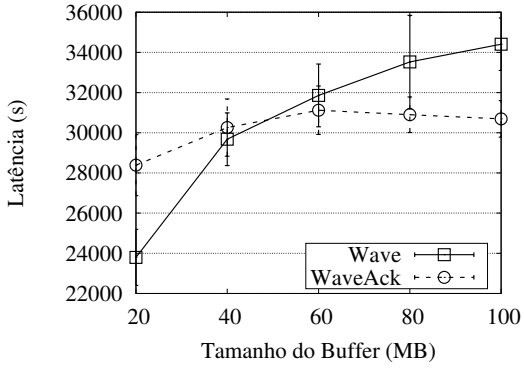
(d) Prophet



(e) ProphetV2

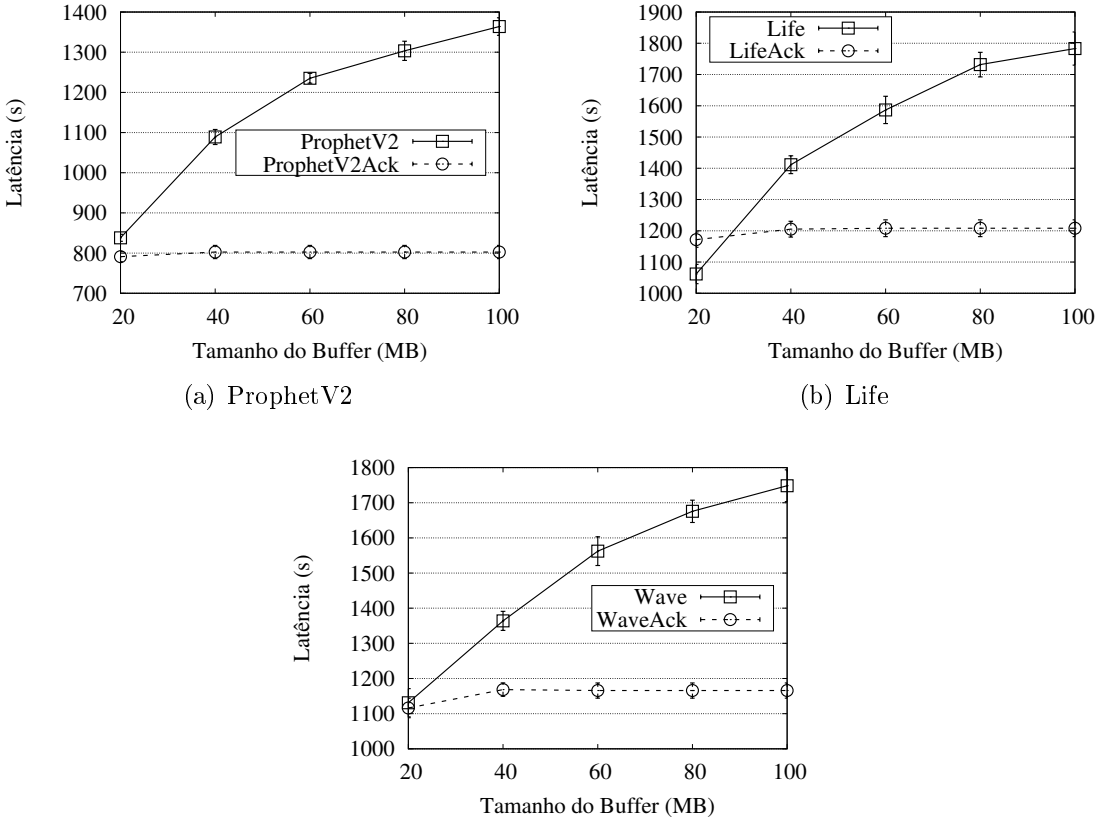


(f) Spray and Wait



(g) Wave

Figura A.6: Atraso de entrega para o cenário Infocom05.

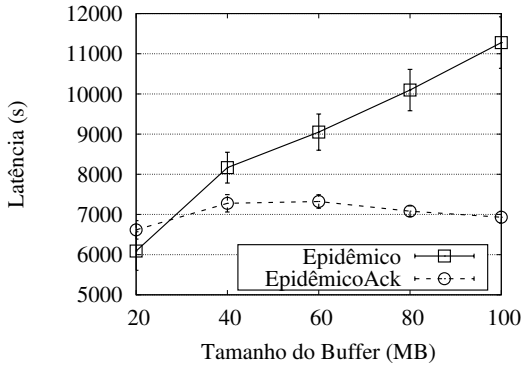


(a) ProphetV2

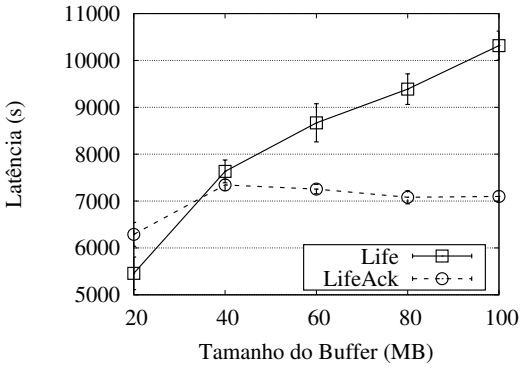
(b) Life

(c) Wave

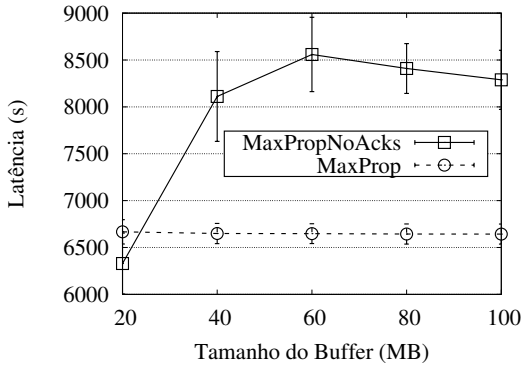
Figura A.7: Atraso de entrega para o cenário Rollernet.



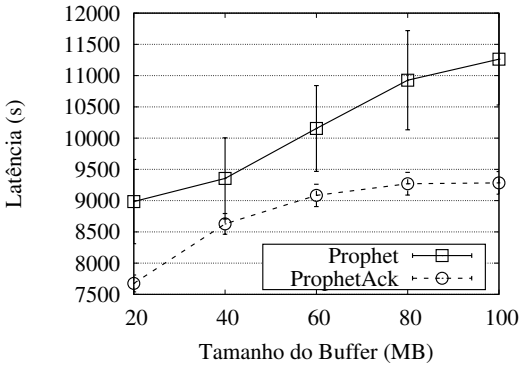
(a) Epidêmico



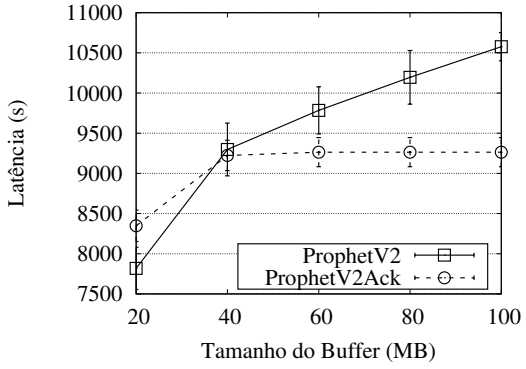
(b) Life



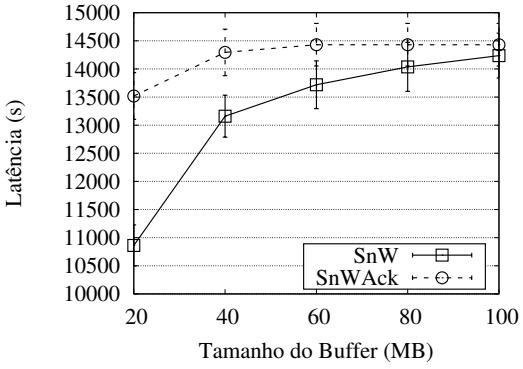
(c) MaxProp



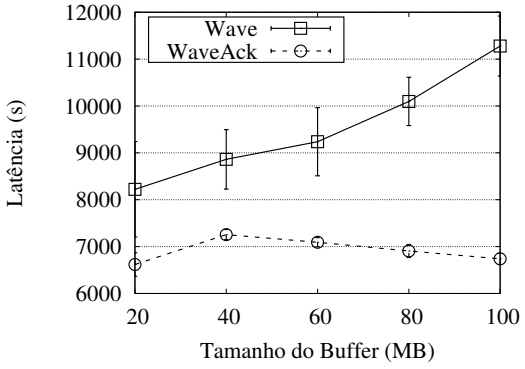
(d) Prophet



(e) ProphetV2



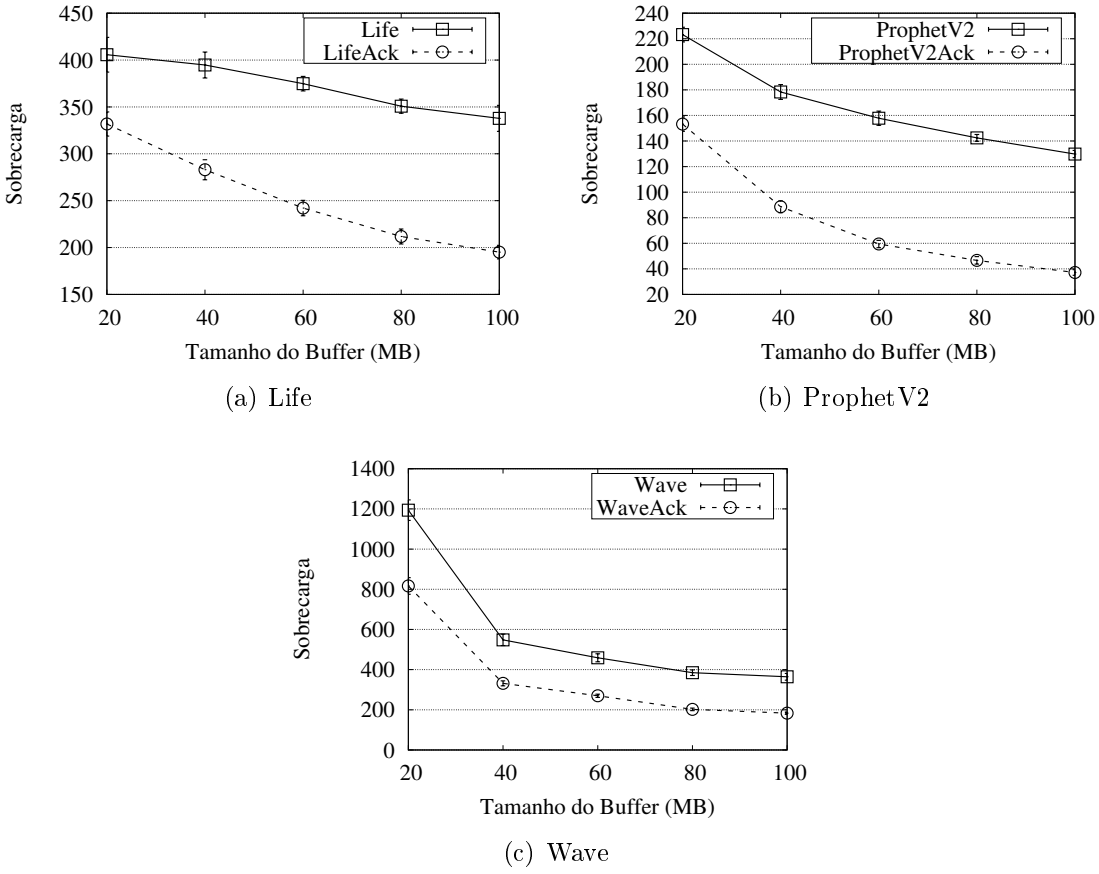
(f) Spray and Wait



(g) Wave

Figura A.8: Atraso de entrega para o cenário ShoppingMall.

A.3 Sobrecarga

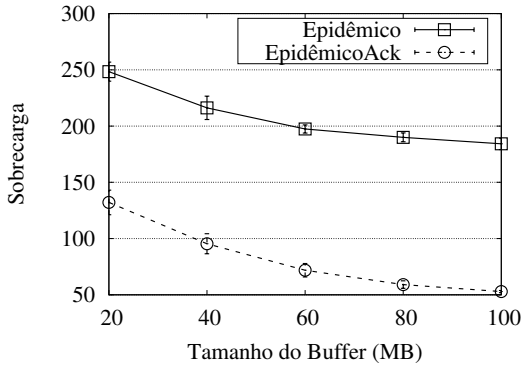


(a) Life

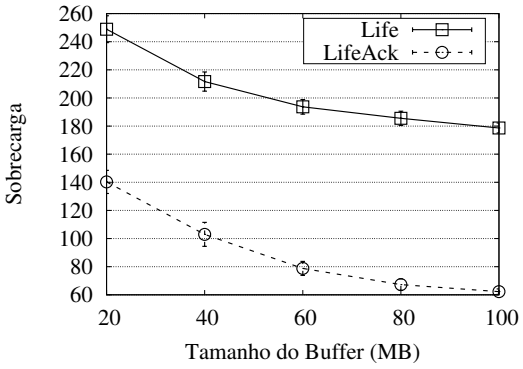
(b) ProphetV2

(c) Wave

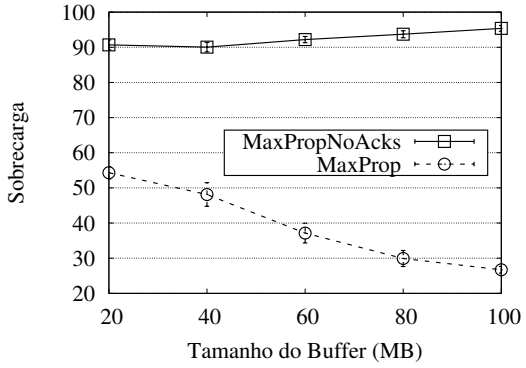
Figura A.9: Sobrecarga para o cenário Dieselnet.



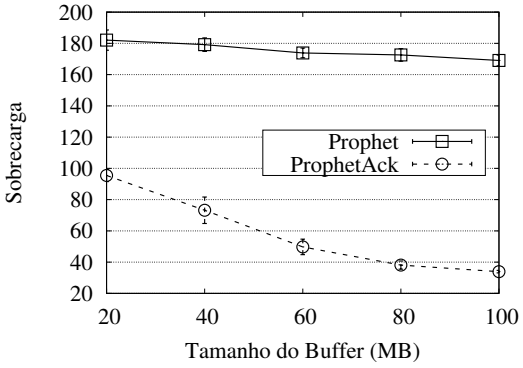
(a) Epidêmico



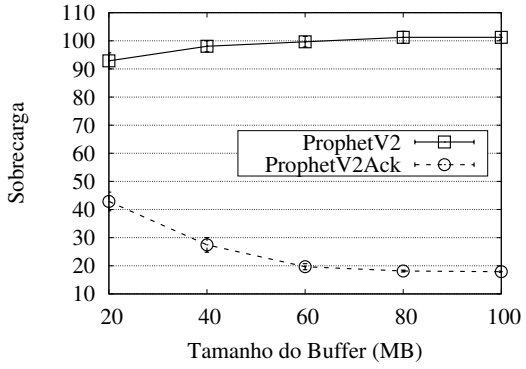
(b) Life



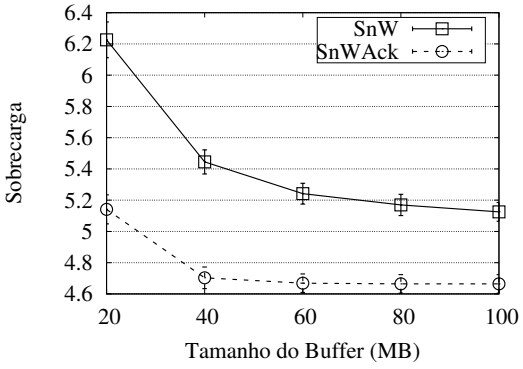
(c) MaxProp



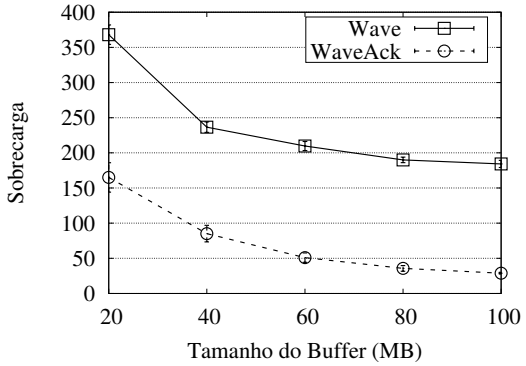
(d) Prophet



(e) ProphetV2

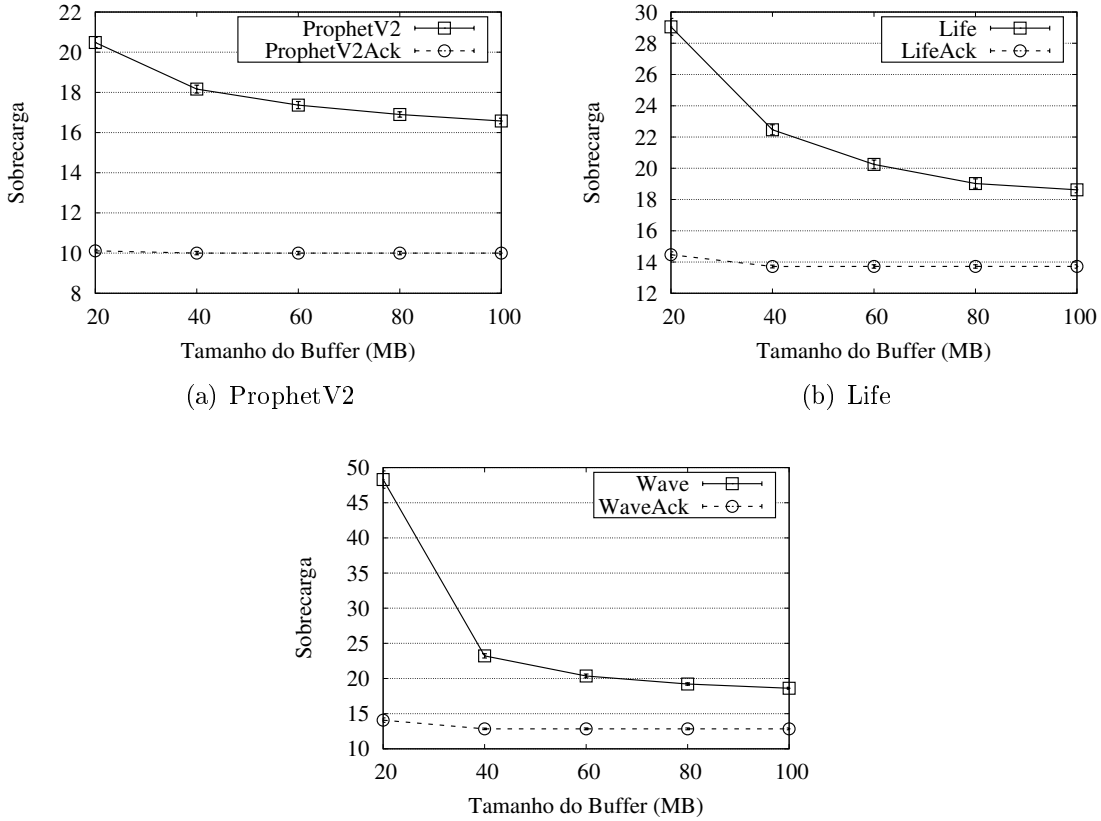


(f) Spray and Wait



(g) Wave

Figura A.10: Sobrecarga para o cenário Infocom05.

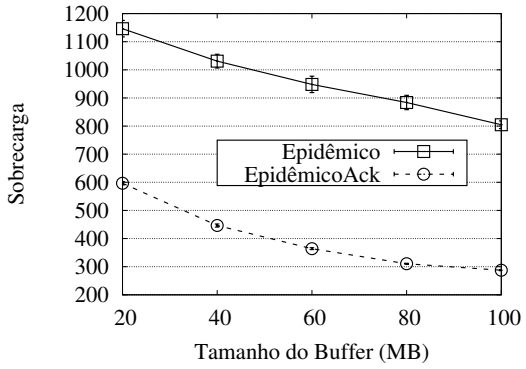


(a) ProphetV2

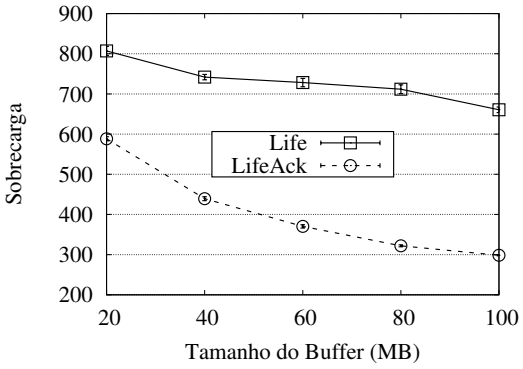
(b) Life

(c) Wave

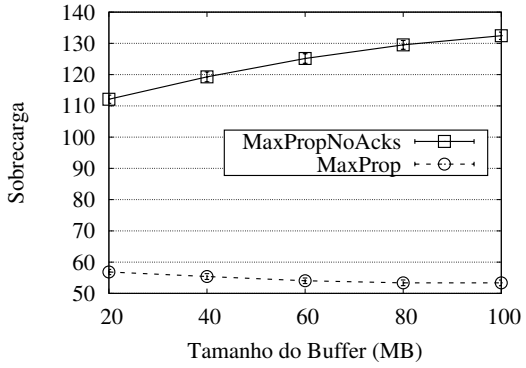
Figura A.11: Sobrecarga de entrega para o cenário Rollernet.



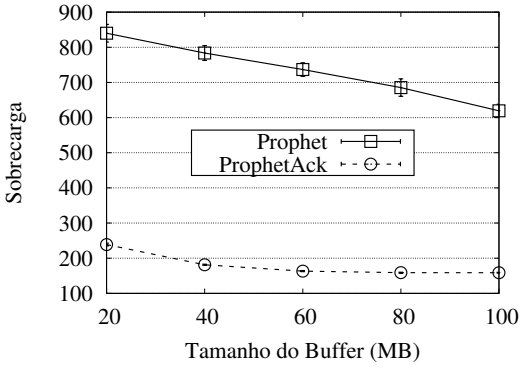
(a) Epidêmico



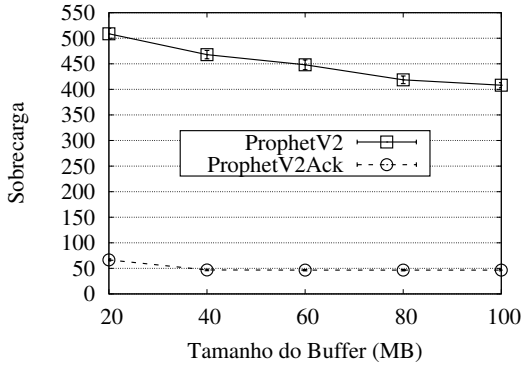
(b) Life



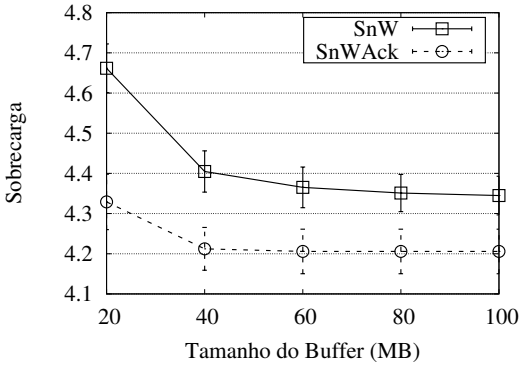
(c) MaxProp



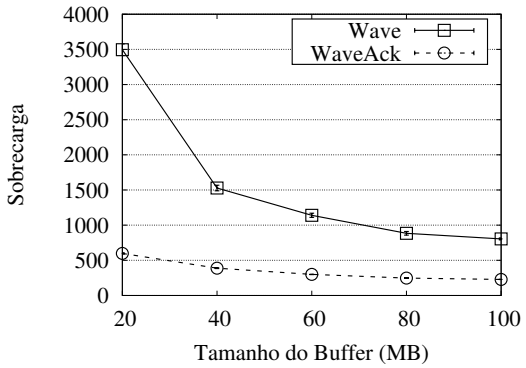
(d) Prophet



(e) ProphetV2



(f) Spray and Wait



(g) Wave

Figura A.12: Sobrecarga de entrega para o cenário ShoppingMall.

APÊNDICE B – Avaliação do Ataque de Falsificação de Reconhecimentos Positivos

Este apêndice reúne resultados que foram omitidos na Seção 5.4 do Capítulo 5, que trata da avaliação do impacto do ataque de falsificação de reconhecimentos positivos em DTNs. Estes resultados foram omitidos para evitar a repetição desnecessária, visto que são similares com os resultados que foram apresentados. Para a completude deste trabalho, todos os resultados que foram omitidos na referida seção são exibidos a seguir, a saber, os resultados omitidos para os cenários Rollernet e Dieselnet e todos os resultados dos cenários Infocom05 e Shoppingmall.

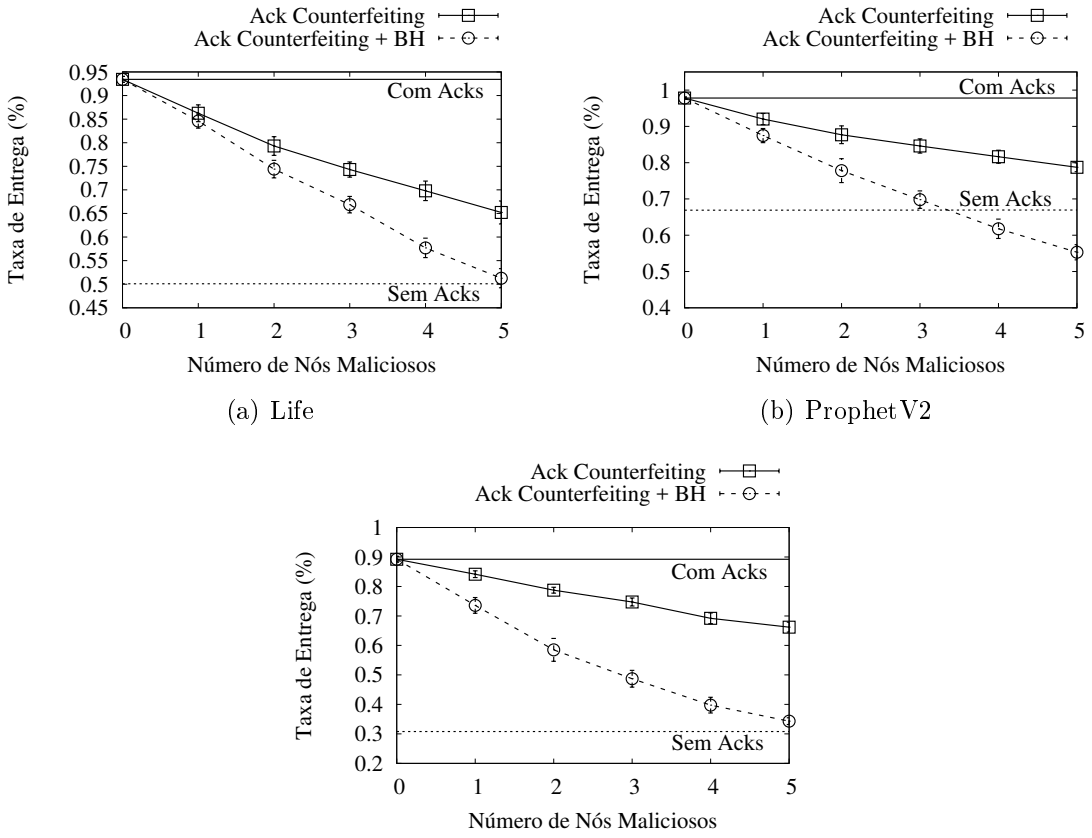
B.1 Taxa de Entrega

Tabela B.1: Impacto do ataque de falsificação de reconhecimentos positivos no cenário Infocom.

	Piora Absoluta					**Piora Relativa**				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	6.49	3.90	2.04	1.56	1.10	15.04	9.04	4.73	3.62	2.55
Life	2.63	0.70	-1.30	-2.25	-2.78	5.56	1.48	-2.75	-4.75	-5.87
MaxProp	-25.44	-34.44	-38.77	-39.86	-41.12	-30.43	-41.19	-46.37	-47.67	-49.18
Prophet	-1.97	-5.20	-7.63	-10.38	-11.03	-3.23	-8.53	-12.51	-17.02	-18.09
ProphetV2	-2.33	-5.23	-7.88	-9.38	-10.62	-3.40	-7.64	-11.50	-13.69	-15.50
Spray and Wait	-4.29	-8.50	-11.76	-14.48	-16.85	-5.40	-10.69	-14.80	-18.22	-21.20
Wave	6.85	4.43	3.10	1.78	0.84	16.55	10.71	7.49	4.30	2.03

Tabela B.2: Impacto do ataque de falsificação de reconhecimentos positivos com buraco negro no cenário Infocom.

	Piora Absoluta					**Piora Relativa**				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	6.71	4.57	2.21	-1.16	-3.64	15.55	10.59	5.12	-2.69	-8.44
Life	4.58	1.18	-2.33	-5.50	-8.83	9.68	2.49	-4.92	-11.62	-18.66
MaxProp	-31.34	-44.43	-52.00	-57.03	-60.41	-37.48	-53.14	-62.19	-68.21	-72.25
Prophet	-5.18	-10.89	-14.38	-18.53	-20.95	-8.49	-17.86	-23.58	-30.39	-34.36
ProphetV2	-3.77	-9.17	-15.16	-20.10	-23.94	-5.50	-13.39	-22.13	-29.34	-34.95
Spray and Wait	-8.09	-16.31	-22.93	-29.62	-35.14	-10.18	-20.52	-28.85	-37.27	-44.21
Wave	1.57	-6.70	-11.70	-16.23	-18.87	3.79	-16.19	-28.27	-39.22	-45.60



(a) Life

(b) ProphetV2

(c) Wave

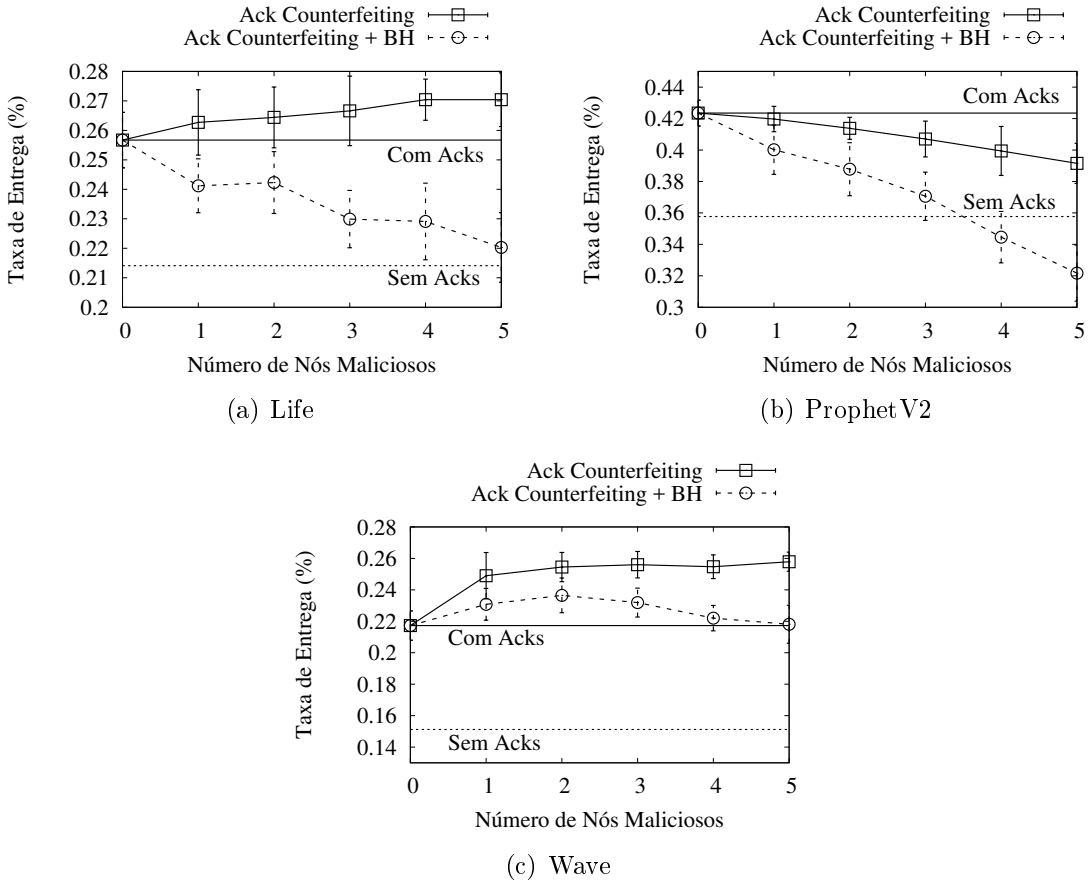
Figura B.1: Taxa de entrega para o cenário Rollernet.

Tabela B.3: Impacto do ataque de falsificação de reconhecimentos positivos no cenário Shopping.

	Piora Absoluta					Piora Relativa				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	-5.10	-8.27	-10.64	-12.29	-14.77	-6.76	-10.96	-14.10	-16.28	-19.57
Life	-4.52	-7.42	-8.01	-9.86	-11.19	-5.39	-8.85	-9.55	-11.76	-13.35
MaxProp	-9.73	-12.66	-13.60	-14.20	-14.25	-9.79	-12.73	-13.68	-14.28	-14.33
Prophet	-5.37	-9.58	-12.10	-14.16	-15.52	-6.07	-10.82	-13.67	-16.00	-17.53
ProphetV2	-1.93	-3.44	-5.12	-6.51	-7.15	-2.04	-3.63	-5.41	-6.88	-7.55
Spray and Wait	-4.71	-7.65	-9.95	-11.90	-12.06	-5.00	-8.12	-10.56	-12.63	-12.80
Wave	-6.02	-8.38	-10.19	-11.53	-12.92	-8.29	-11.54	-14.03	-15.88	-17.79

Tabela B.4: Impacto do ataque de falsificação de reconhecimentos positivos com buraco negro no cenário Shopping.

	Piora Absoluta					Piora Relativa				
Protocolos/ Nº de Nós Maliciosos	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Epidêmico	-3.76	-5.65	-7.51	-10.88	-14.57	-4.98	-7.49	-9.95	-14.42	-19.31
Life	-5.36	-9.19	-12.29	-17.92	-22.18	-6.39	-10.96	-14.66	-21.37	-26.45
MaxProp	-15.10	-22.26	-27.75	-32.50	-37.48	-15.19	-22.39	-27.91	-32.69	-37.70
Prophet	-4.96	-8.64	-12.11	-16.65	-21.04	-5.60	-9.76	-13.68	-18.81	-23.77
ProphetV2	-5.67	-9.67	-13.99	-18.94	-24.32	-5.99	-10.21	-14.78	-20.01	-25.69
SnW	-13.16	-24.06	-33.27	-42.53	-47.34	-13.96	-25.53	-35.30	-45.13	-50.23
Wave	-5.55	-8.26	-11.14	-14.06	-16.96	-7.67	-11.42	-15.40	-19.43	-23.44

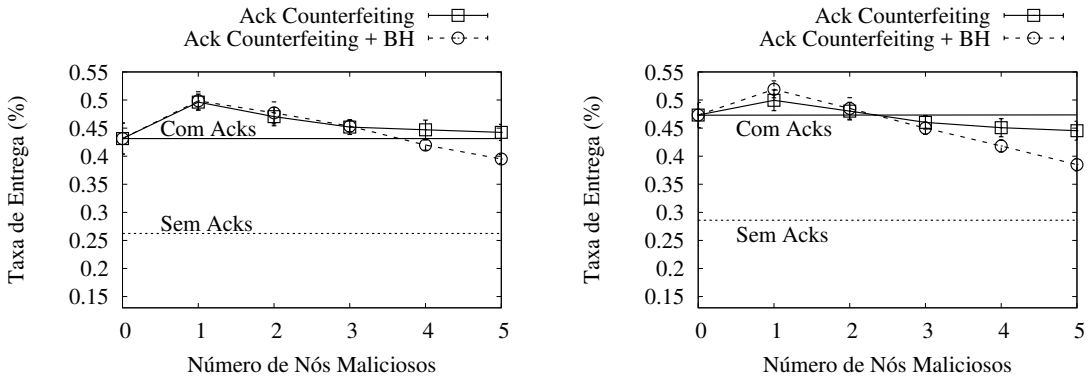


(a) Life

(b) ProphetV2

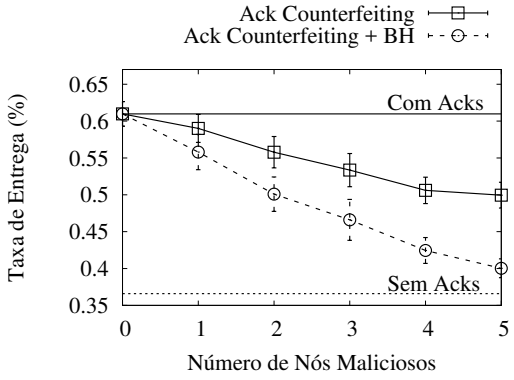
(c) Wave

Figura B.2: Taxa de entrega para o cenário Dieselnet.

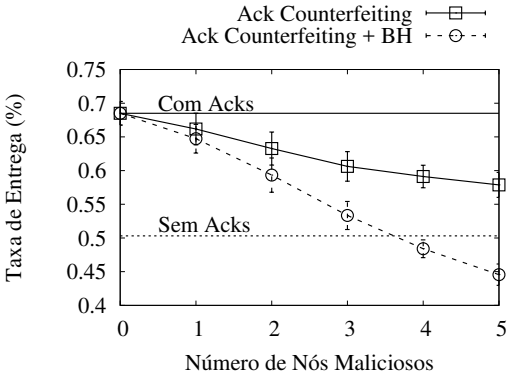


(a) Epidêmico

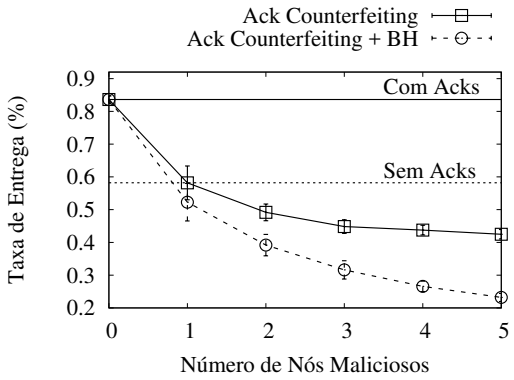
(b) Life



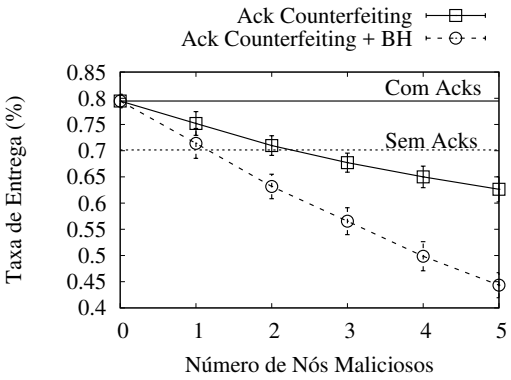
(c) Prophet



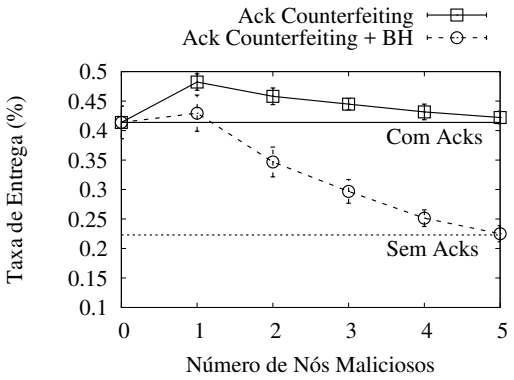
(d) ProphetV2



(e) MaxProp

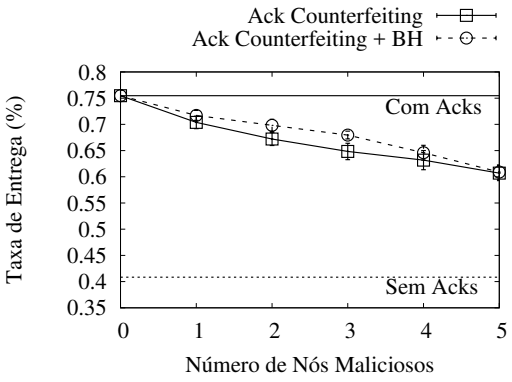


(f) Spray and Wait

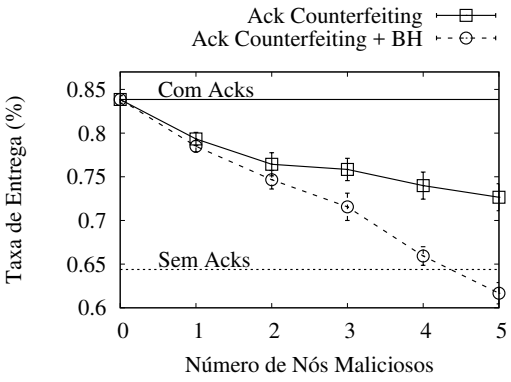


(g) Wave

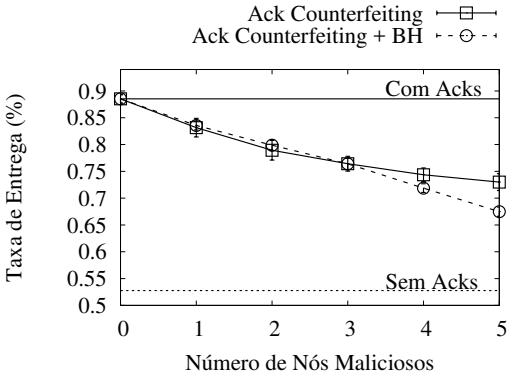
Figura B.3: Taxa de entrega para o cenário Infocom05.



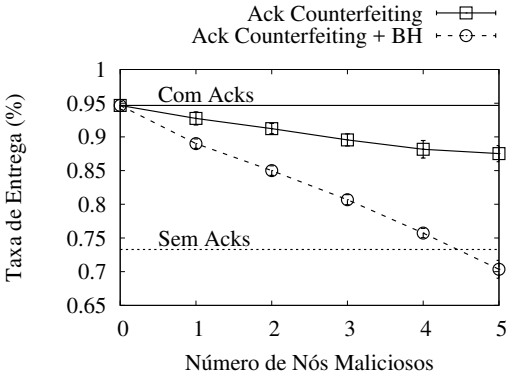
(a) Epidêmico



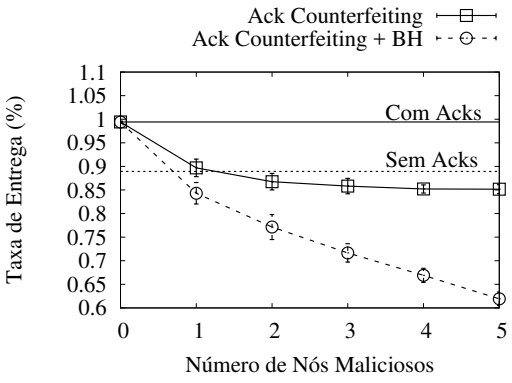
(b) Life



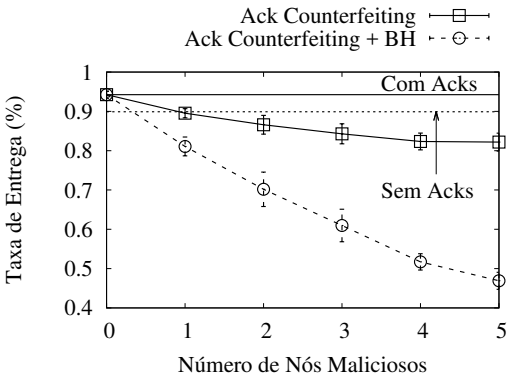
(c) Prophet



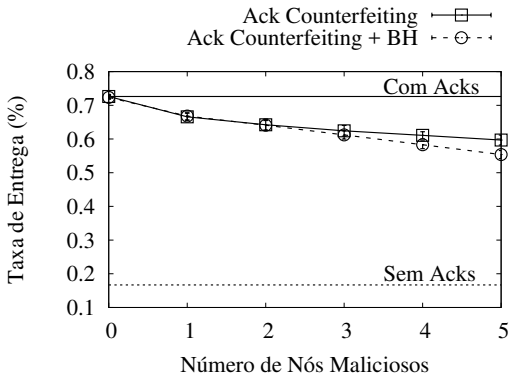
(d) ProphetV2



(e) MaxProp



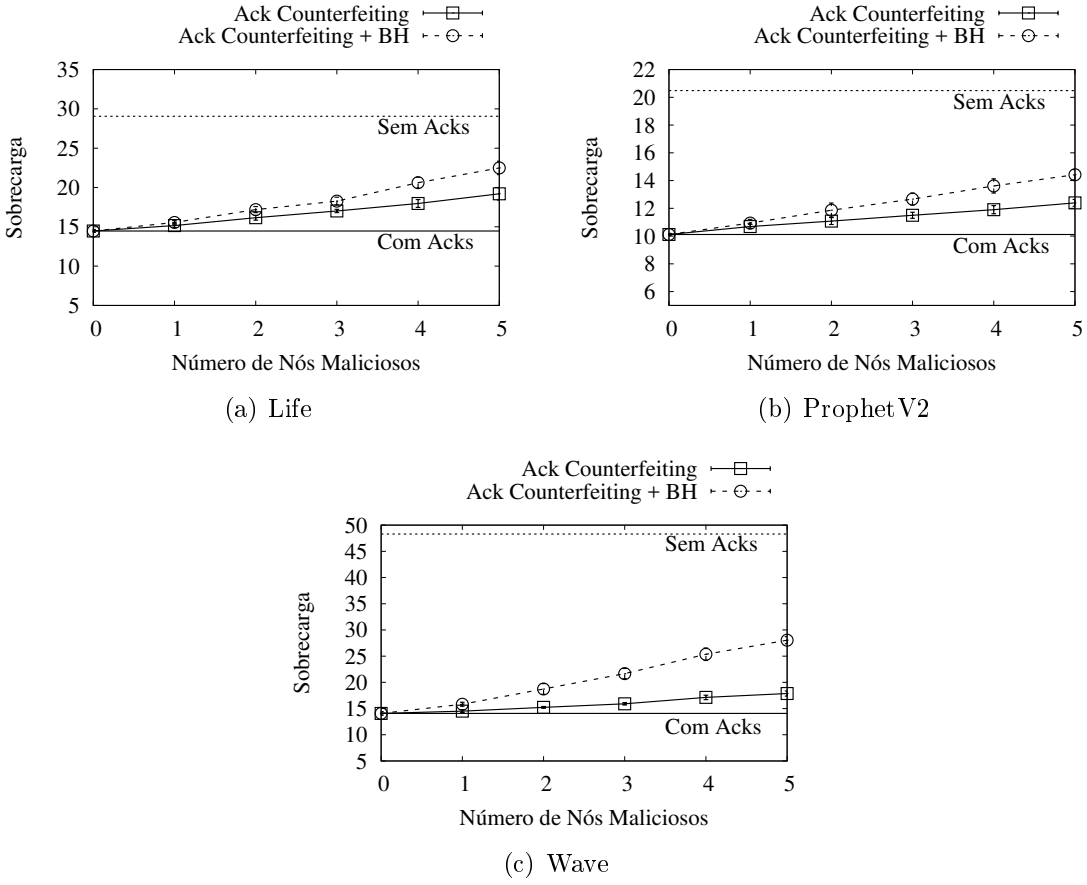
(f) Spray and Wait



(g) Wave

Figura B.4: Taxa de entrega para o cenário Shopping.

B.2 Sobrecarga

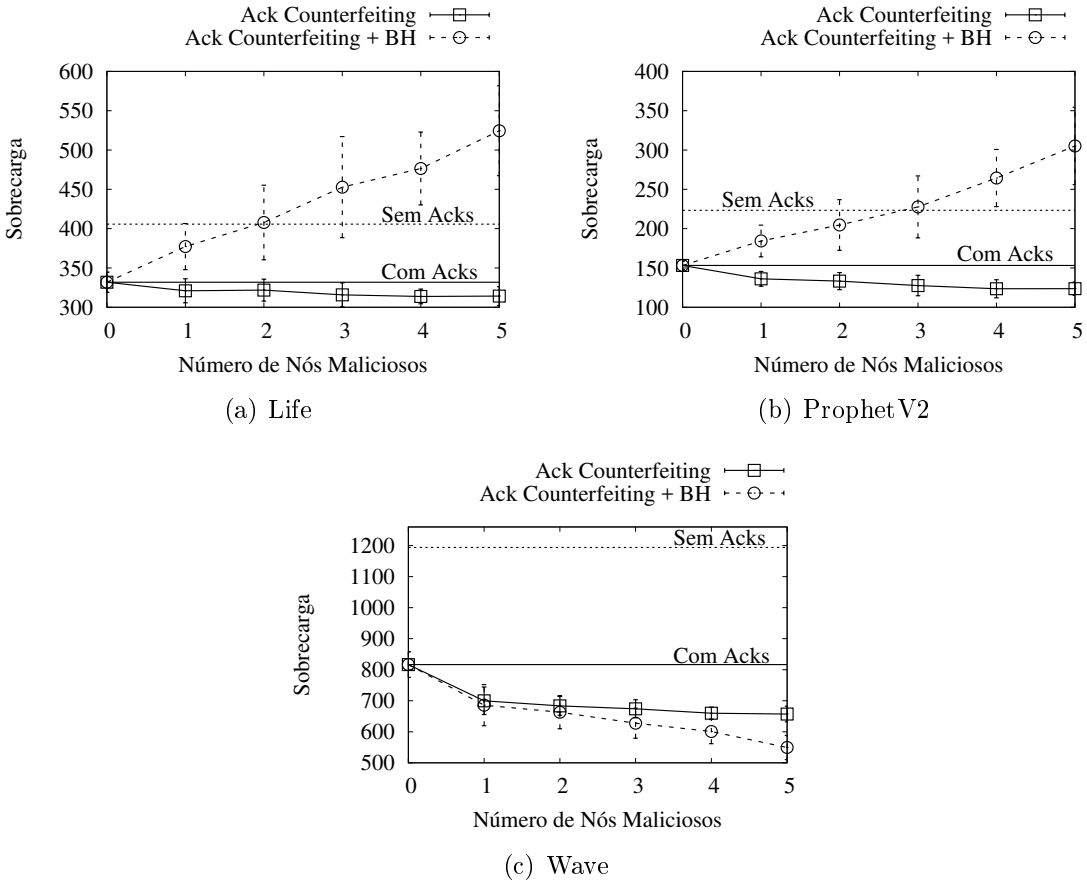


(a) Life

(b) ProphetV2

(c) Wave

Figura B.5: Sobrecarga para o cenário Rollernet.

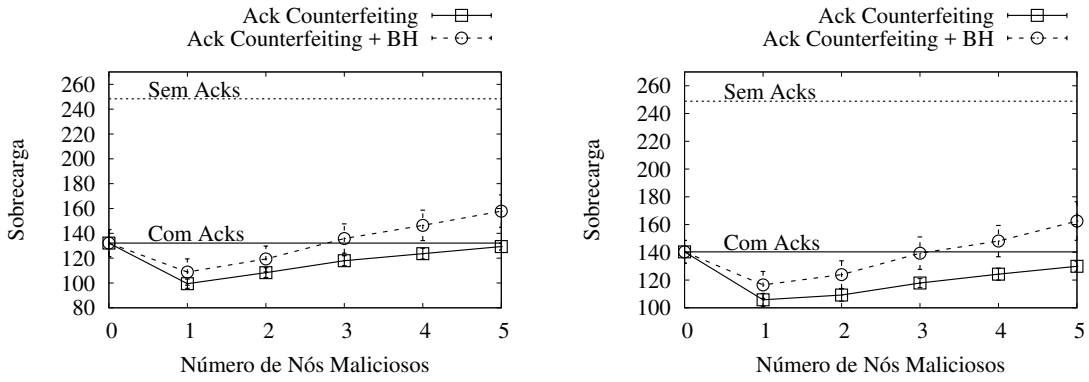


(a) Life

(b) ProphetV2

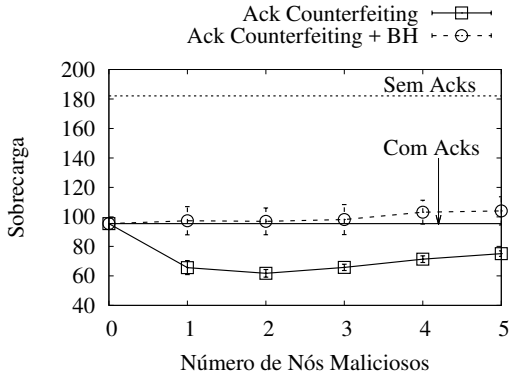
(c) Wave

Figura B.6: Taxa de entrega para o cenário Dieselnet.

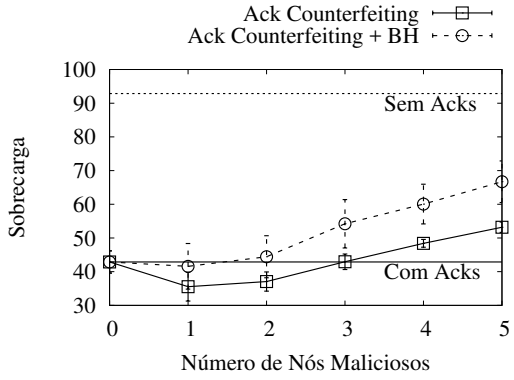


(a) Epidêmico

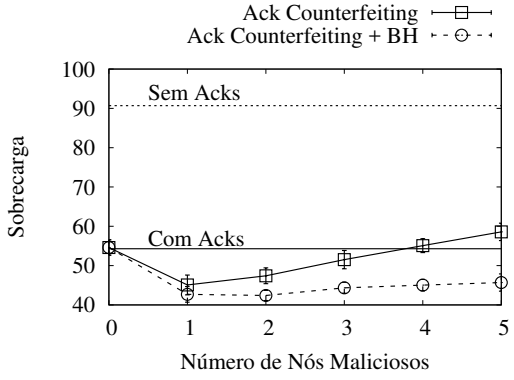
(b) Life



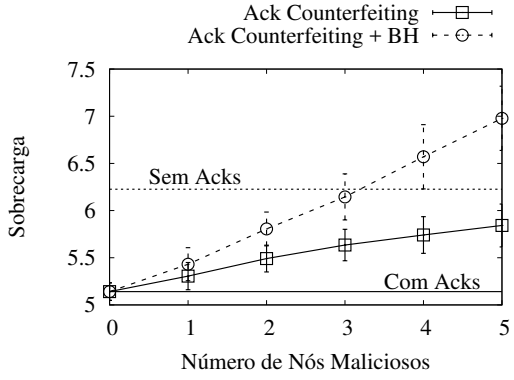
(c) Prophet



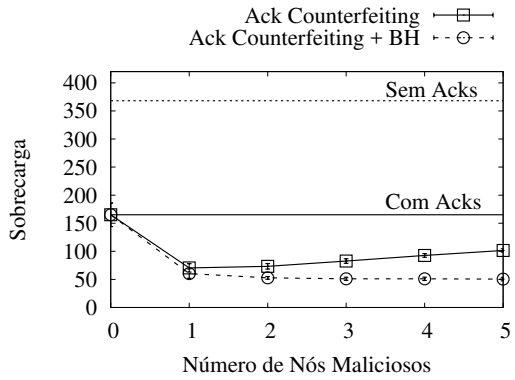
(d) ProphetV2



(e) MaxProp

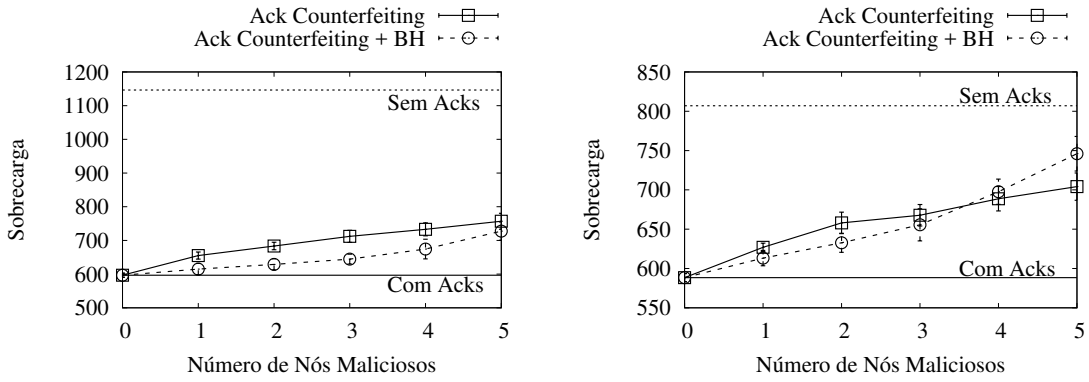


(f) Spray and Wait



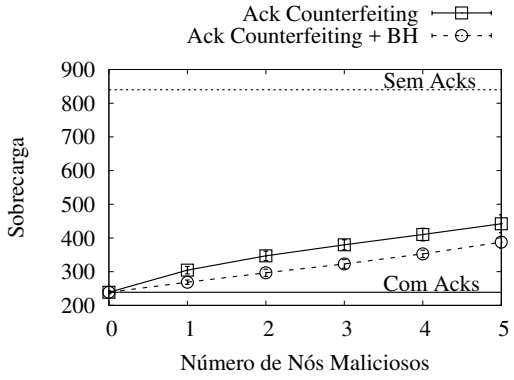
(g) Wave

Figura B.7: Sobrecarga para o cenário Infocom.

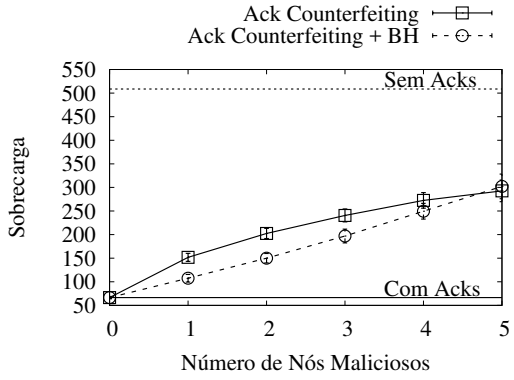


(a) Epidêmico

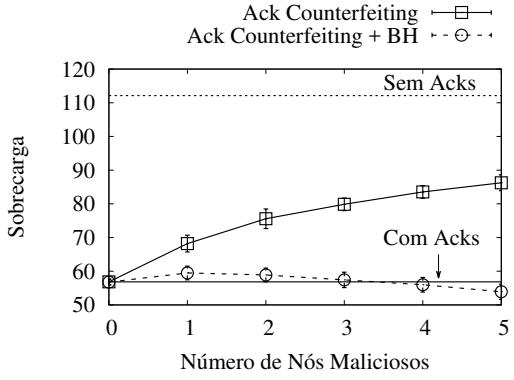
(b) Life



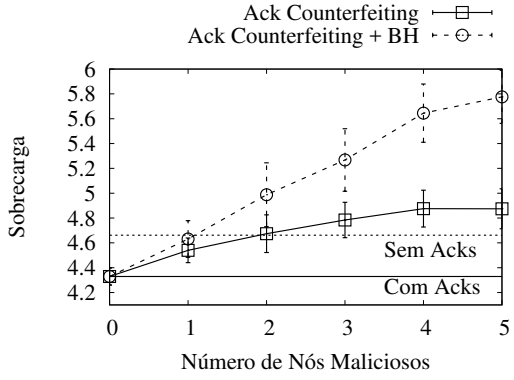
(c) Prophet



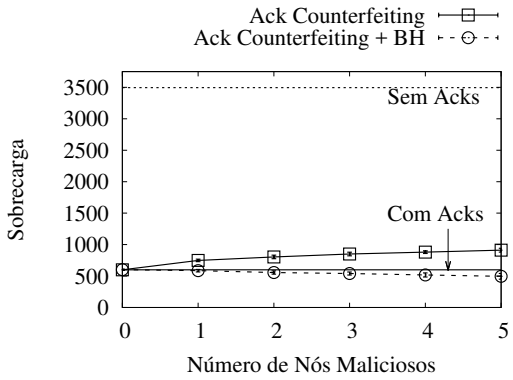
(d) ProphetV2



(e) MaxProp



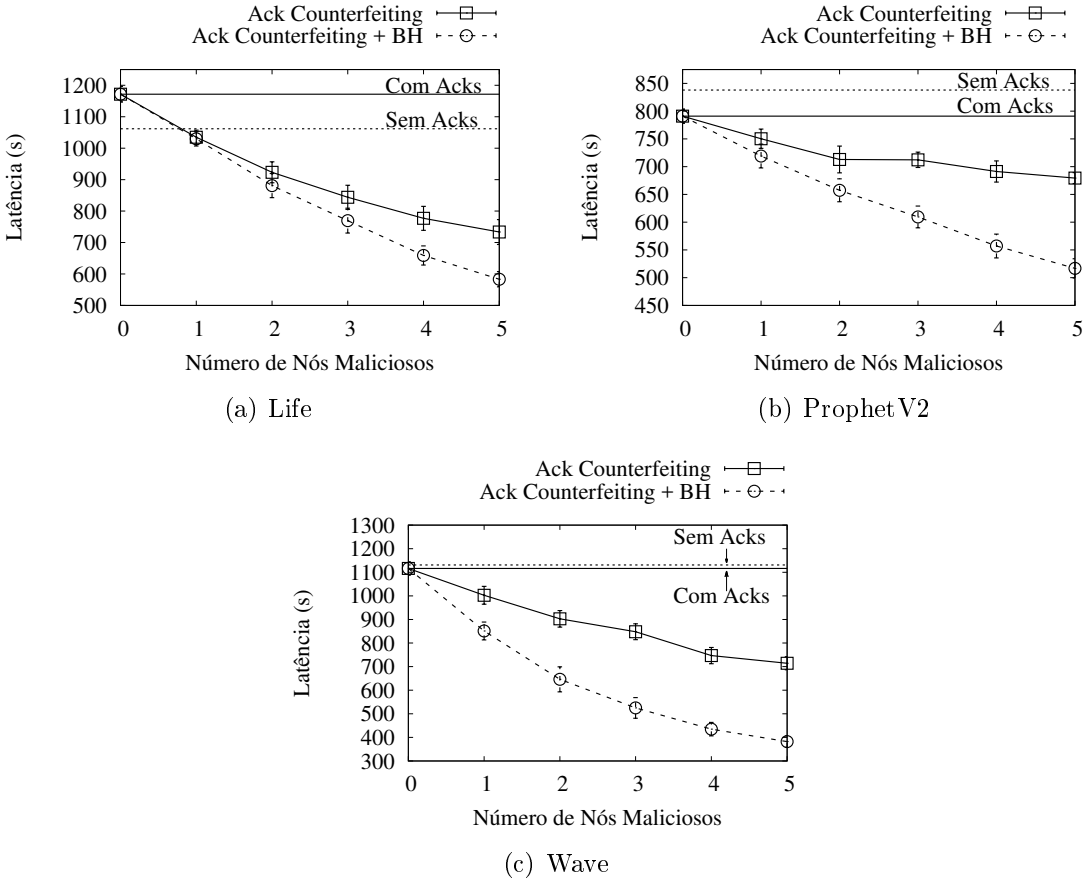
(f) Spray and Wait



(g) Wave

Figura B.8: Sobrecarga para o cenário Shopping.

B.3 Atraso de Entrega

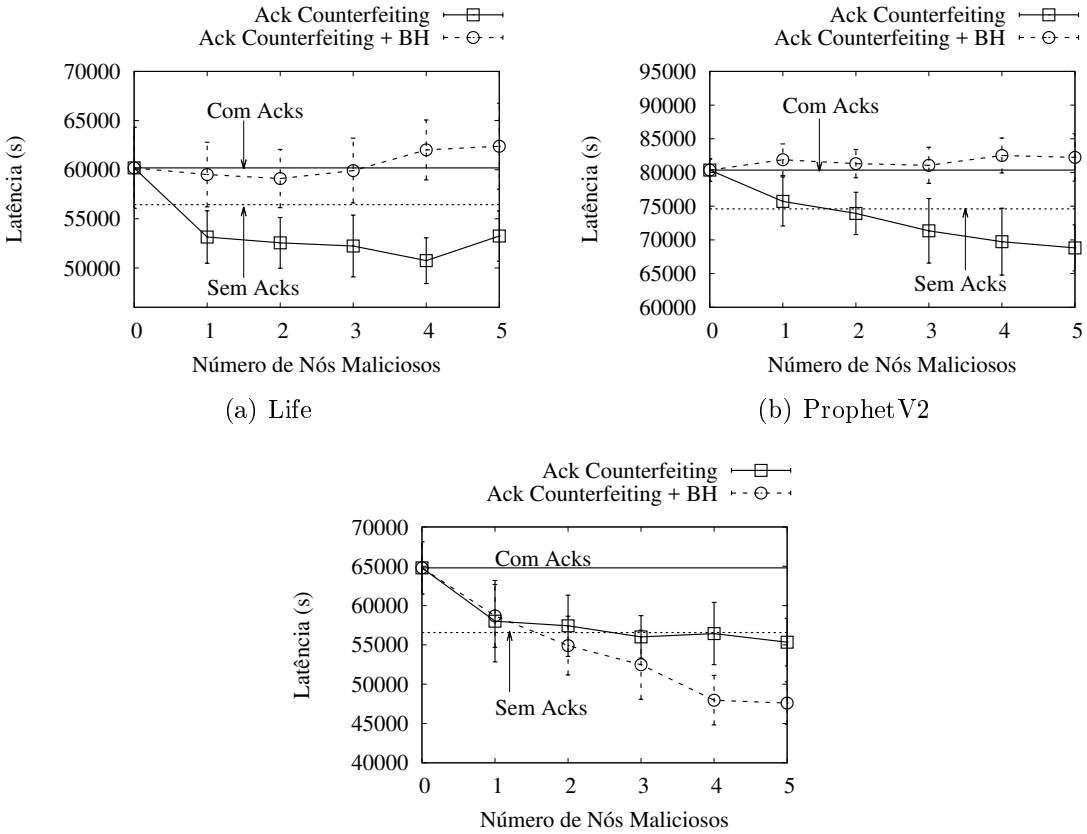


(a) Life

(b) ProphetV2

(c) Wave

Figura B.9: Sobrecarga para o cenário Rollernet.

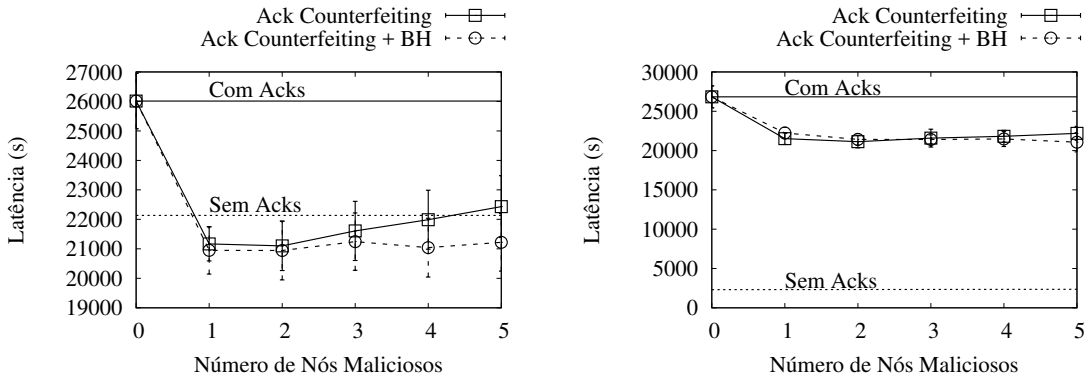


(a) Life

(b) ProphetV2

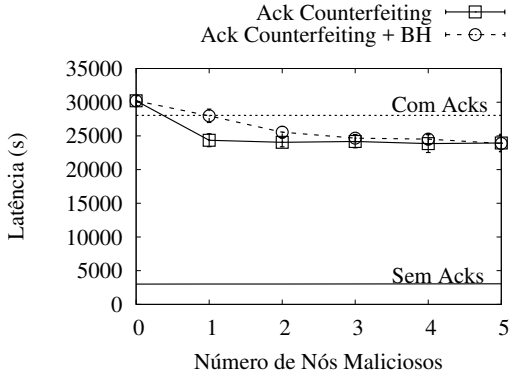
(c) Wave

Figura B.10: Taxa de entrega para o cenário Dieselnet.

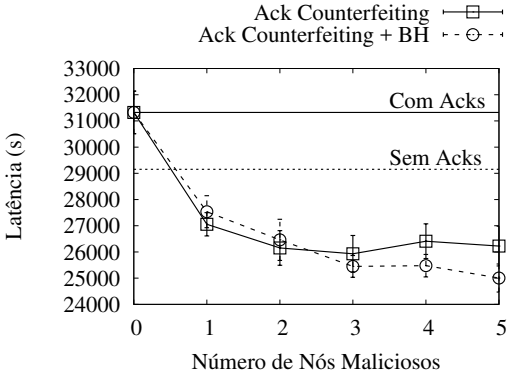


(a) Epidêmico

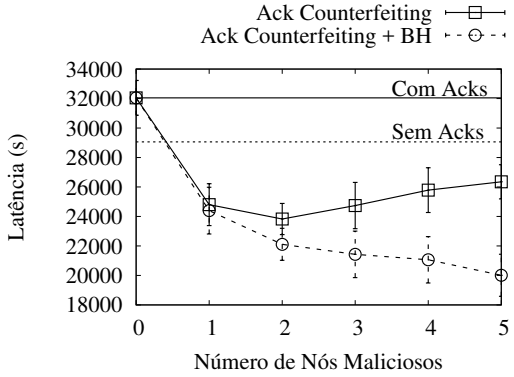
(b) Life



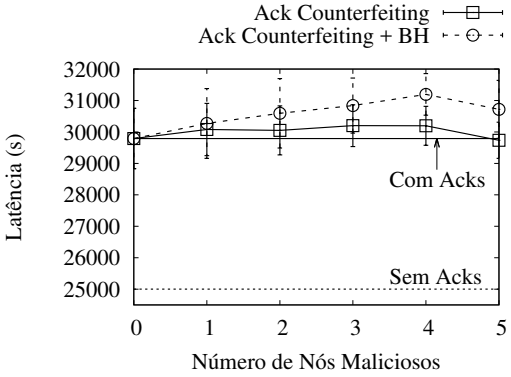
(c) Prophet



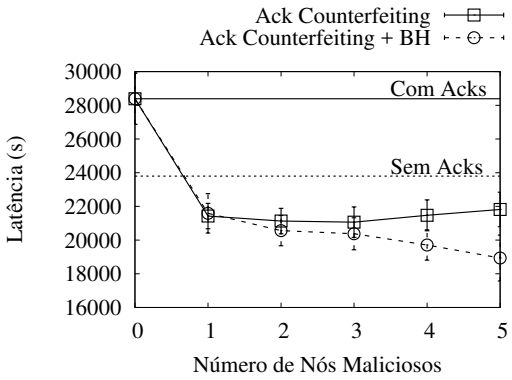
(d) ProphetV2



(e) MaxProp

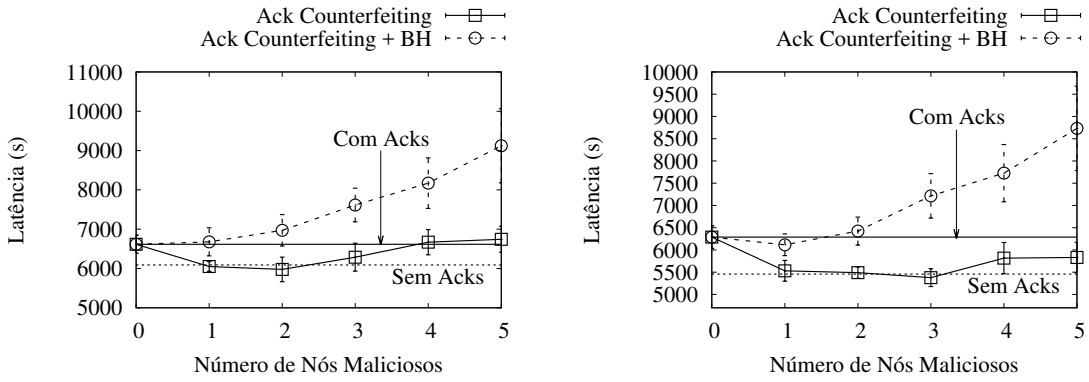


(f) Spray and Wait



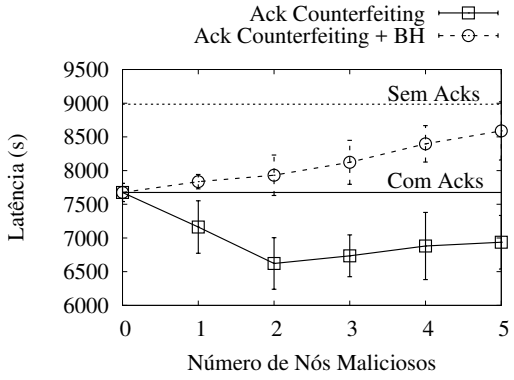
(g) Wave

Figura B.11: Sobrecarga para o cenário Infocom.

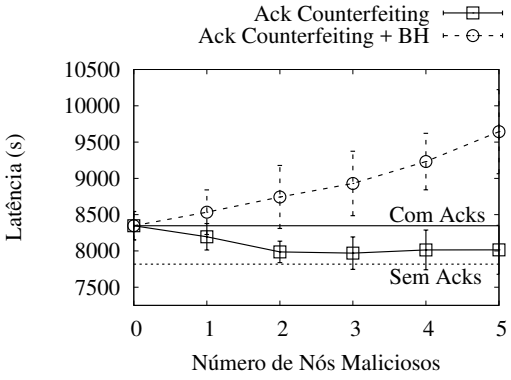


(a) Epidêmico

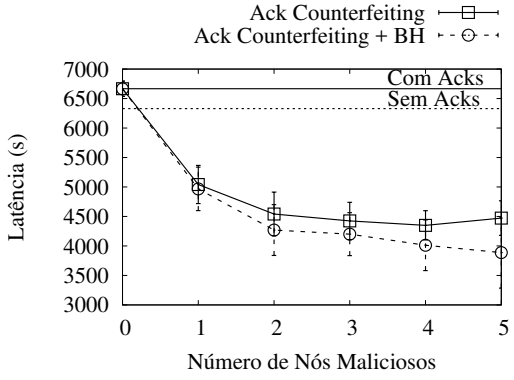
(b) Life



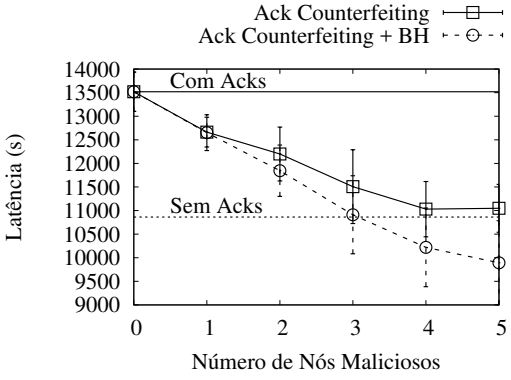
(c) Prophet



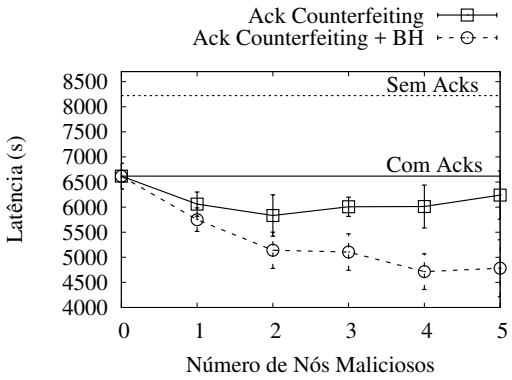
(d) ProphetV2



(e) MaxProp



(f) Spray and Wait



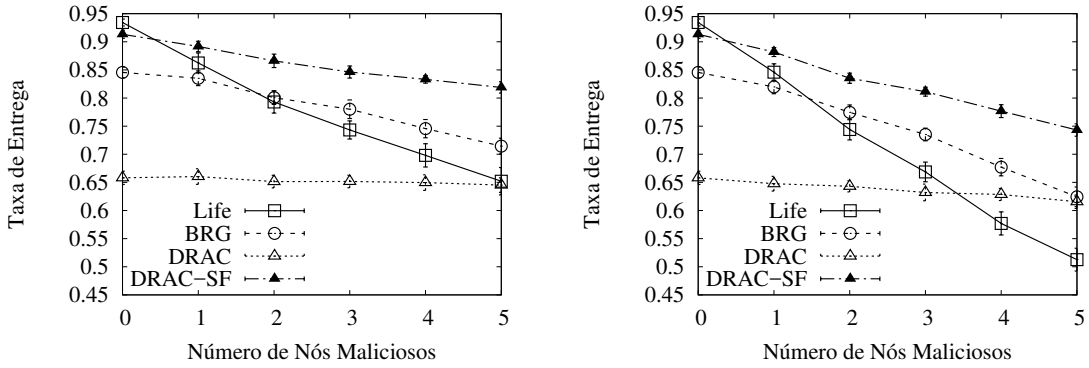
(g) Wave

Figura B.12: Sobrecarga para o cenário Shopping.

APÊNDICE C – Avaliação das Contramedidas Propostas

C.1 Taxa de Entrega

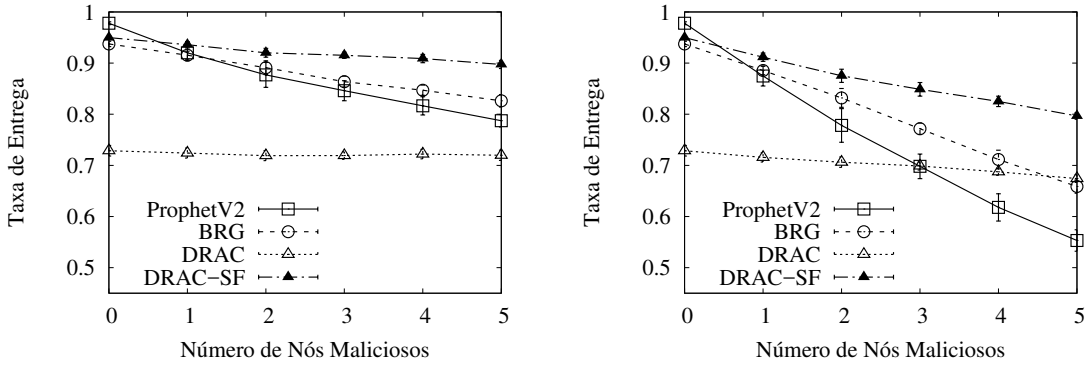
C.1.1 Cenário Rollernet



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

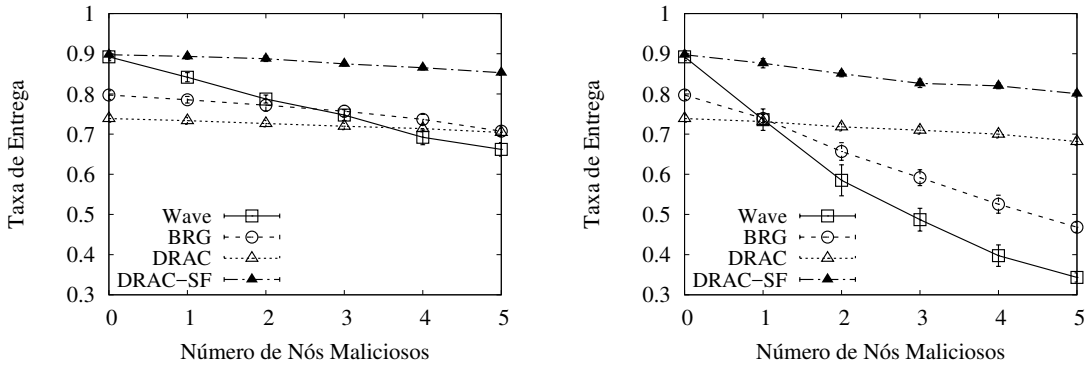
Figura C.1: Taxa de entrega para o protocolo Life no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.2: Taxa de entrega para o protocolo ProphetV2 no cenário Rollernet.

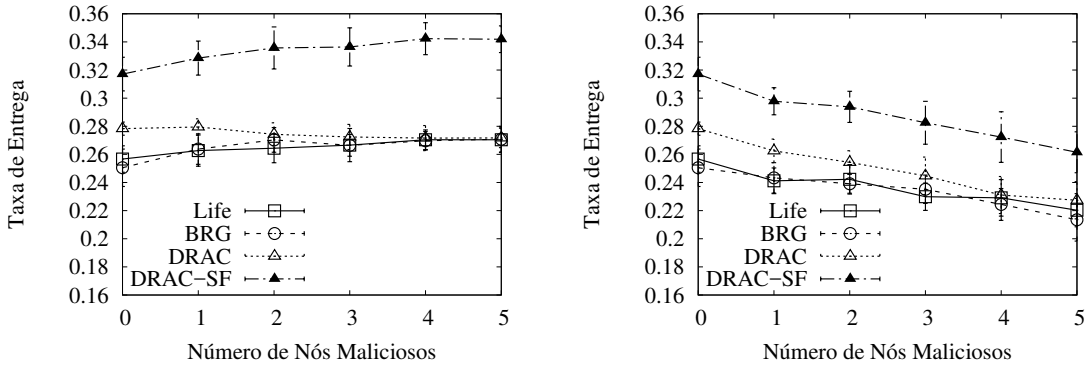


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.3: Taxa de entrega para o protocolo *Wave* no cenário Rollernet.

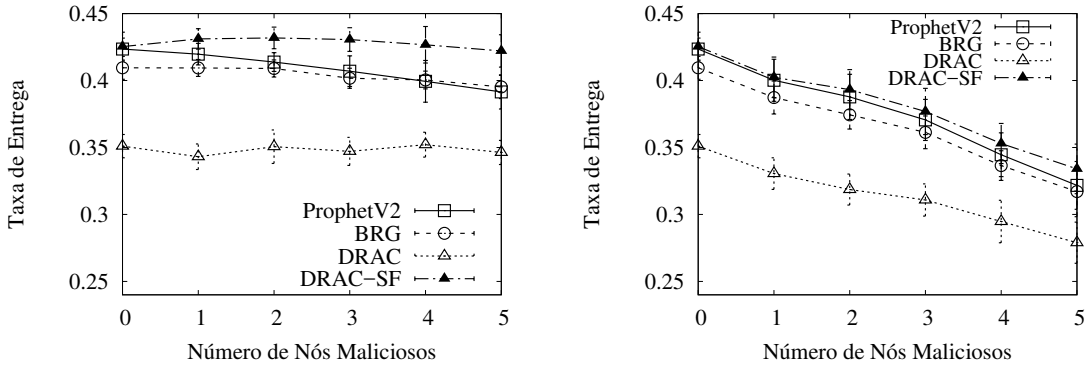
C.1.2 Cenário Dieselnet



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

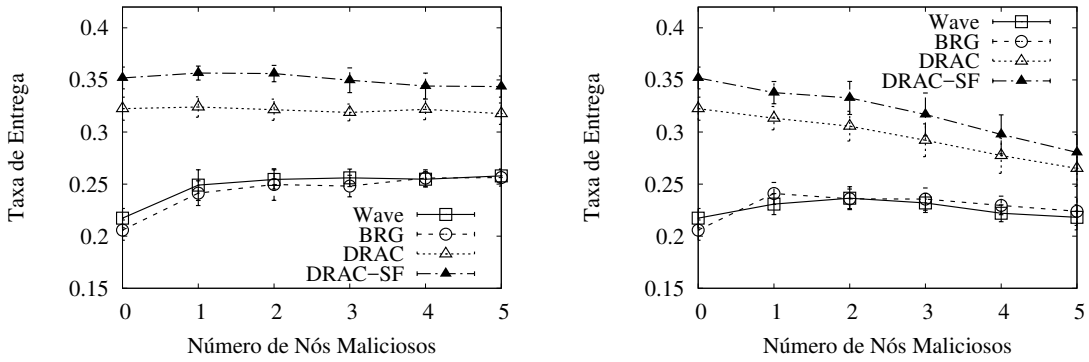
Figura C.4: Taxa de entrega para o protocolo Life no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.5: Taxa de entrega para o protocolo ProphetV2 no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

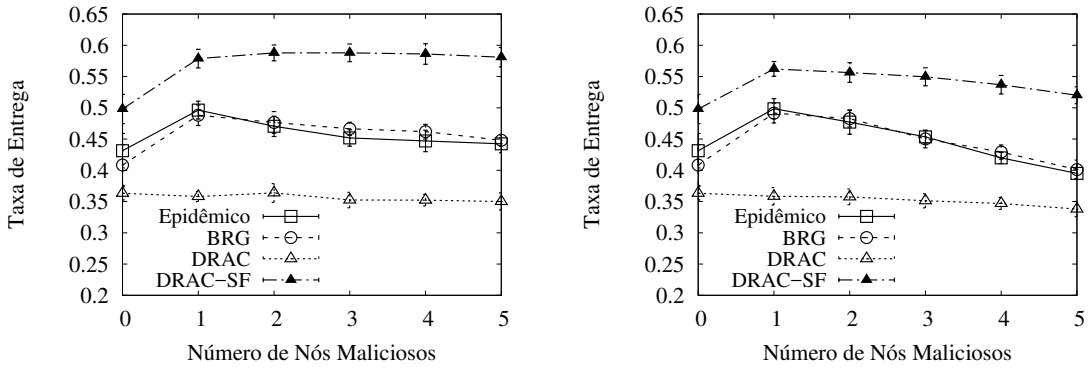
(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.6: Taxa de entrega para o protocolo *Wave* no cenário Dieselnet.

Tabela C.1: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Infocom e o ataque de falsificação de reconhecimentos positivos.

	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	9	9	11	12	12	13	22	19	23	26	27	30
Life	9	10	12	12	13	13	20	19	25	26	29	27
MaxProp	-5	16	24	28	29	30	-6	27	47	58	63	65
Prophet	3	7	10	12	13	13	5	11	17	21	24	25
ProphetV2	0	5	7	9	10	10	0	7	11	14	16	16
SnW	-10	-6	-3	0	2	5	-13	-9	-4	0	4	7
Wave	27	20	21	21	22	22	65	41	44	46	49	48

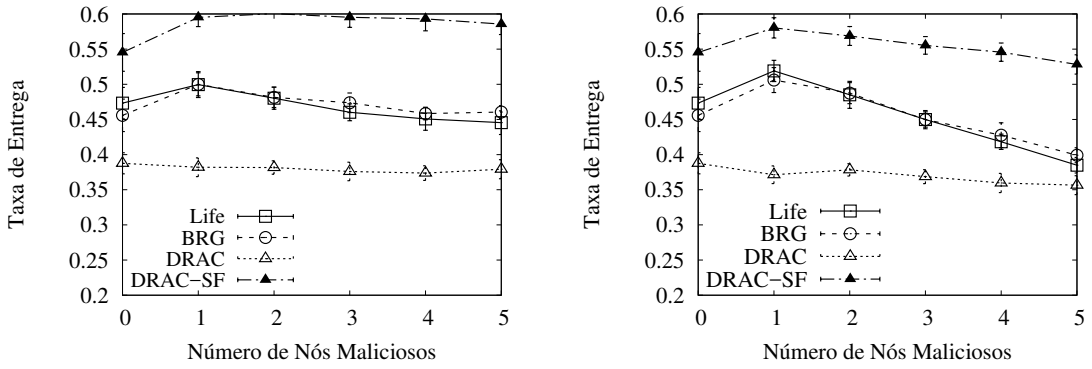
C.1.3 Cenário Infocom



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.7: Taxa de entrega para o protocolo Epidêmico no cenário Infocom.



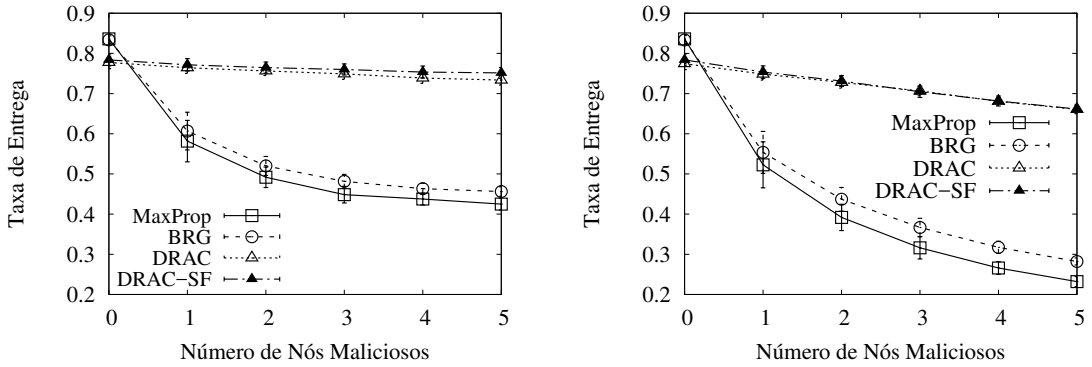
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.8: Taxa de entrega para o protocolo Life no cenário Infocom.

Tabela C.2: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Infocom e o ataque de falsificação de reconhecimentos positivos com buraco negro.

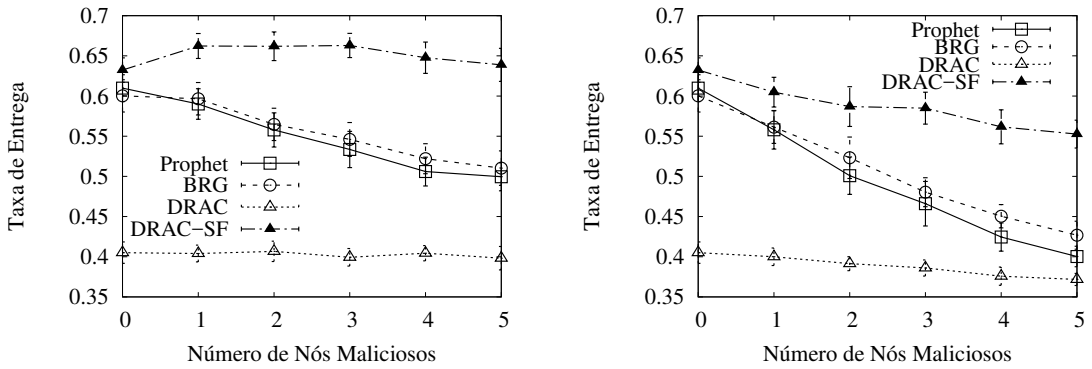
	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	9	7	7	10	11	12	22	14	15	22	25	30
Life	9	7	8	11	12	13	20	15	17	23	28	32
MaxProp	-5	20	29	34	36	38	-6	36	67	92	115	135
Prophet	3	4	6	10	11	13	5	8	12	22	25	29
ProphetV2	0	4	6	9	11	13	0	6	10	16	22	27
SnW	-10	-5	1	5	9	13	-13	-7	1	9	18	28
Wave	27	23	28	30	31	33	65	52	72	88	104	122



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

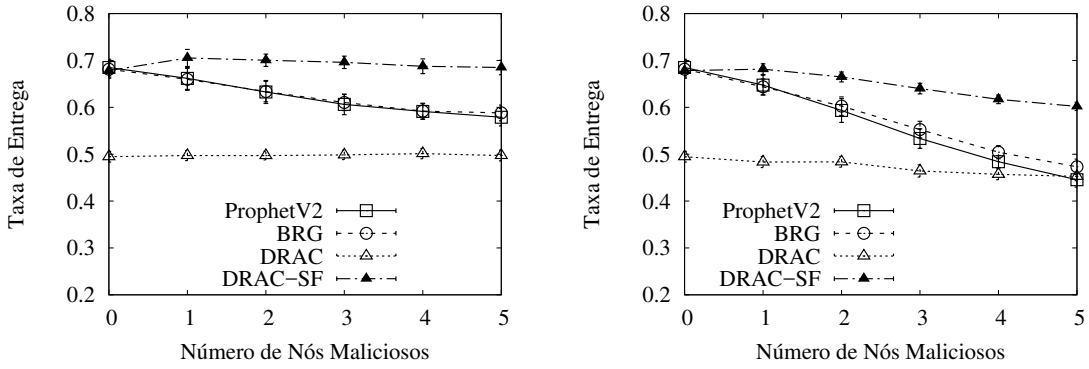
Figura C.9: Taxa de entrega para o protocolo MaxProp no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

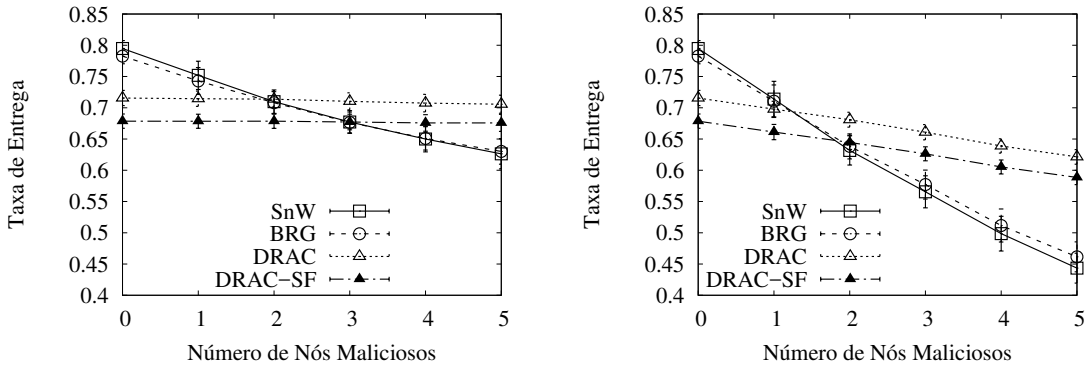
Figura C.10: Taxa de entrega para o protocolo Prophet no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

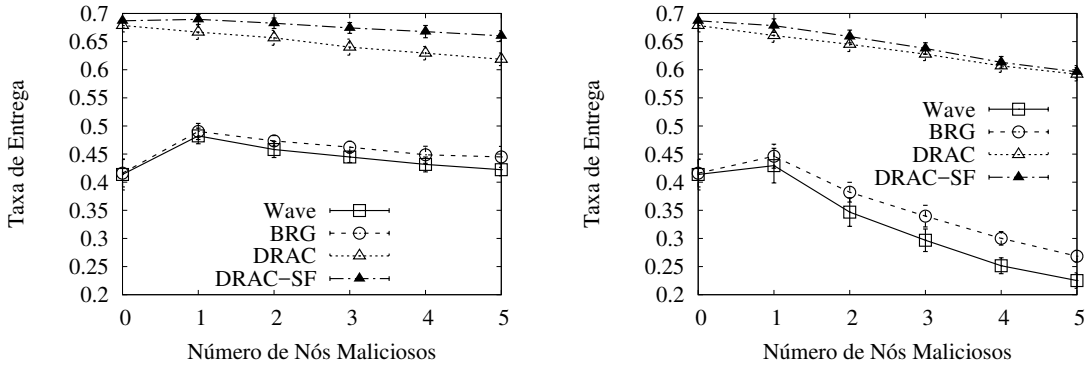
Figura C.11: Taxa de entrega para o protocolo ProphetV2 no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.12: Taxa de entrega para o protocolo *Spray and Wait* no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

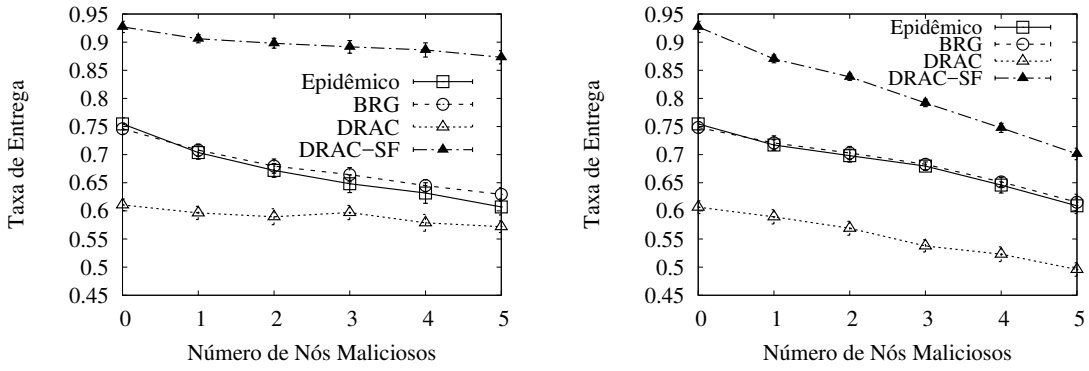
(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.13: Taxa de entrega para o protocolo *Wave* no cenário Infocom.

Tabela C.3: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Shopping e o ataque de falsificação de reconhecimentos positivos.

	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	18	20	22	23	24	24	24	28	32	34	37	39
Life	12	13	14	15	15	15	14	17	19	19	20	20
MaxProp	-1	5	7	7	7	6	-1	6	8	8	8	7
Prophet	7	9	10	12	12	13	8	11	13	16	16	17
ProphetV2	3	4	5	5	6	6	3	5	6	6	6	6
SnW	-15	-10	-6	-4	-1	-1	-16	-11	-7	-4	-2	-1
Wave	18	15	14	13	13	11	25	21	20	18	17	15

C.1.4 Cenário Shopping



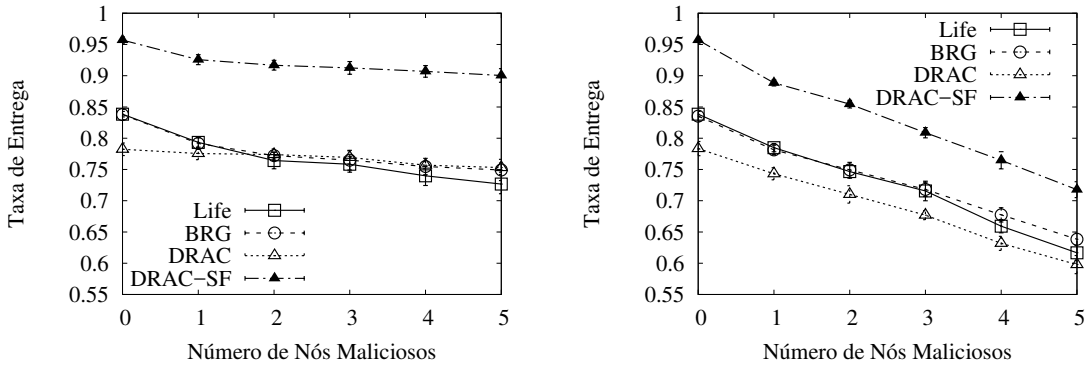
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.14: Taxa de entrega para o protocolo Epidêmico no cenário Shopping.

Tabela C.4: Diferenças absoluta e relativa entre os desempenhos das contramedidas DRAC-SF e BRG para o cenário Shopping e o ataque de falsificação de reconhecimentos positivos com buraco negro.

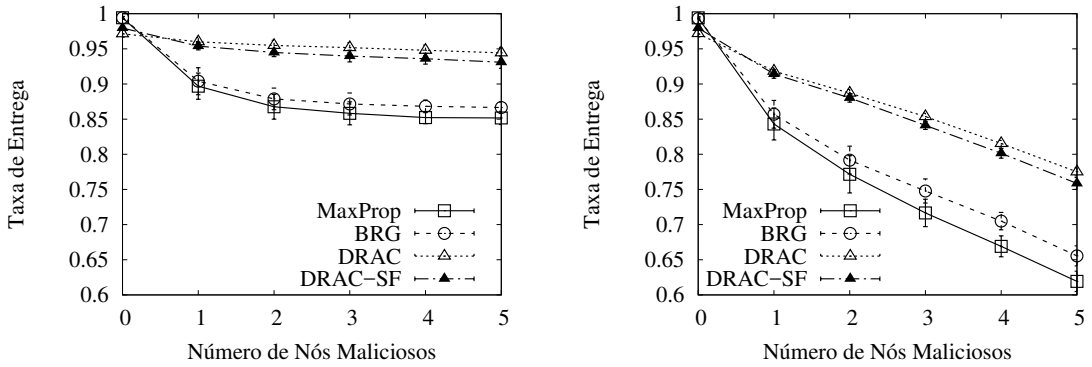
	Diferença Absoluta (%)						**Diferença Relativa (%)**					
Protocolo/ Nº Nós Maliciosos	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
Epidêmico	18	15	14	11	10	9	24	21	19	16	15	14
Life	12	11	11	9	9	8	15	14	14	13	13	12
MaxProp	-1	6	9	9	10	10	-1	7	11	12	14	16
Prophet	7	5	6	5	5	5	7	7	7	7	6	7
ProphetV2	3	4	4	4	4	4	3	4	4	5	5	6
SnW	-15	-8	-2	3	9	11	-16	-10	-2	5	15	21
Wave	18	16	14	13	11	10	25	24	21	20	18	18



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

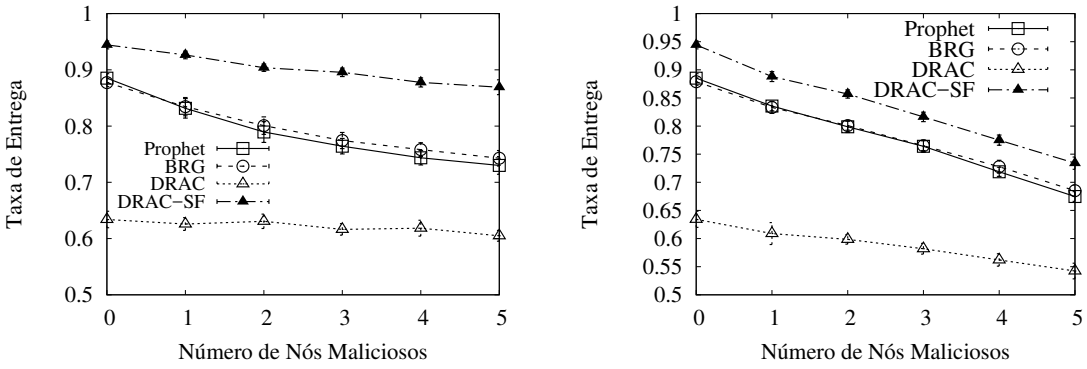
Figura C.15: Taxa de entrega para o protocolo Life no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

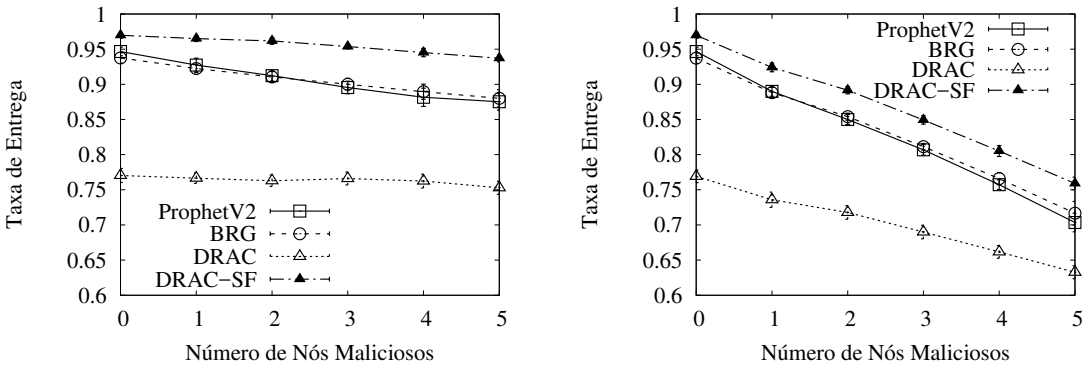
Figura C.16: Taxa de entrega para o protocolo MaxProp no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

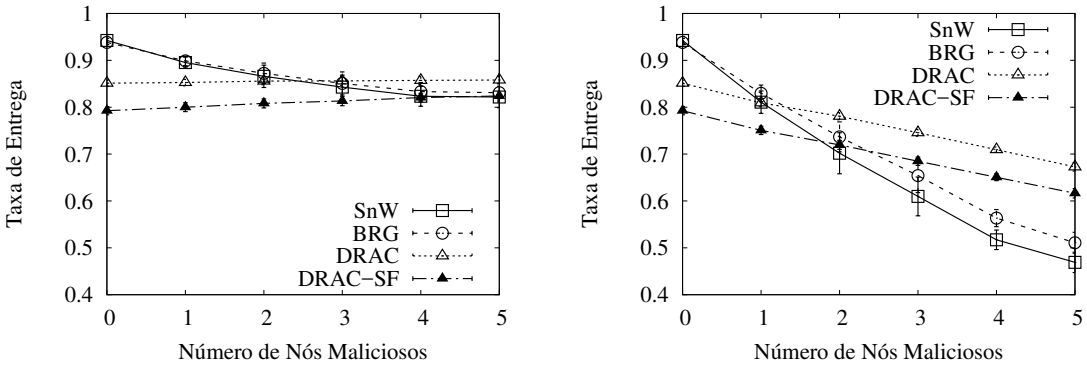
Figura C.17: Taxa de entrega para o protocolo Prophet no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

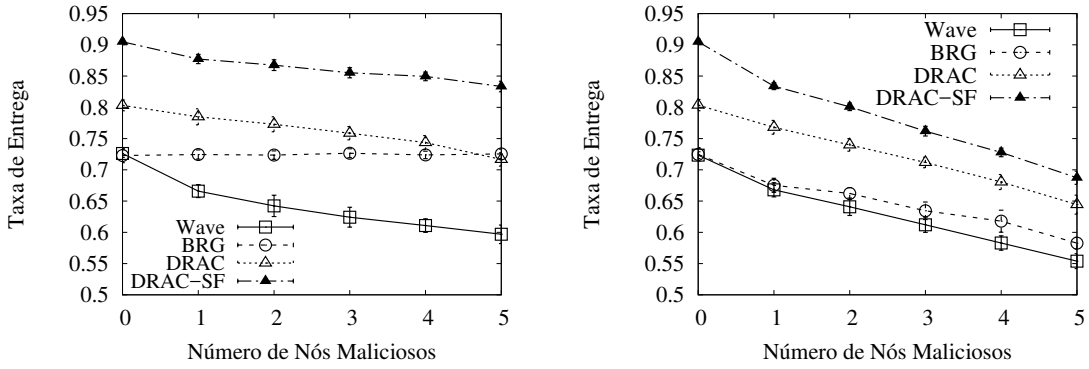
Figura C.18: Taxa de entrega para o protocolo ProphetV2 no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.19: Taxa de entrega para o protocolo *Spray and Wait* no cenário Shopping.



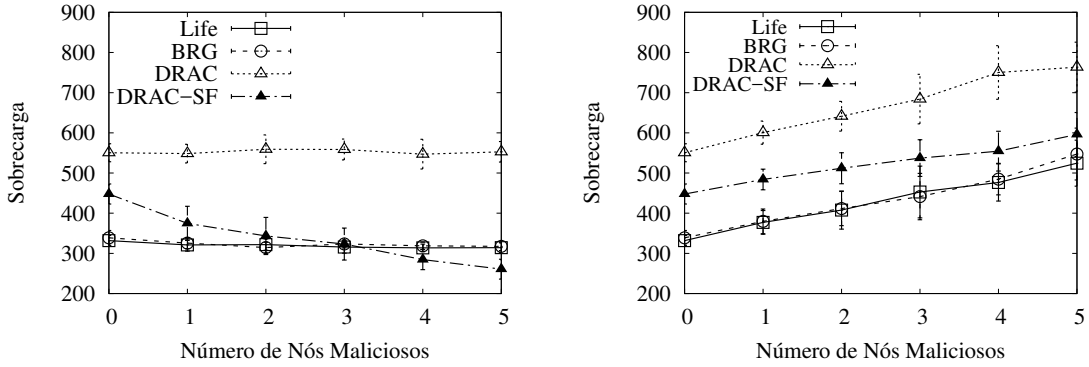
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.20: Taxa de entrega para o protocolo *Wave* no cenário Shopping.

C.2 Sobrecarga

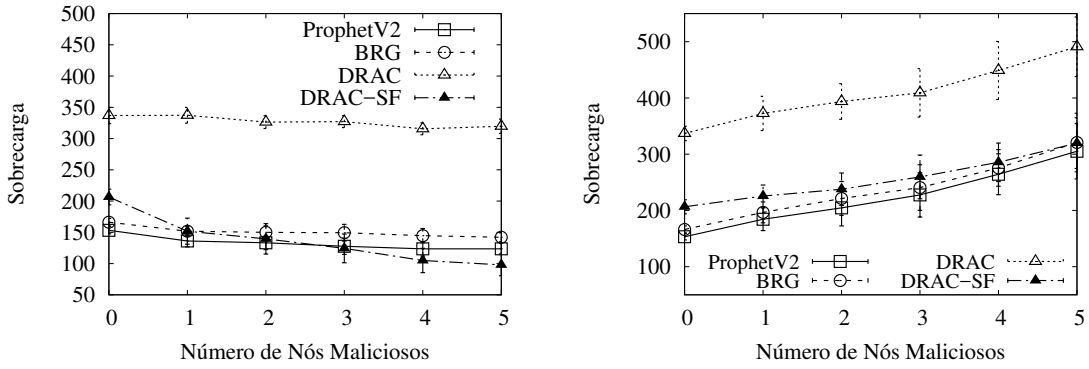
C.2.1 Cenário Dieselnet



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

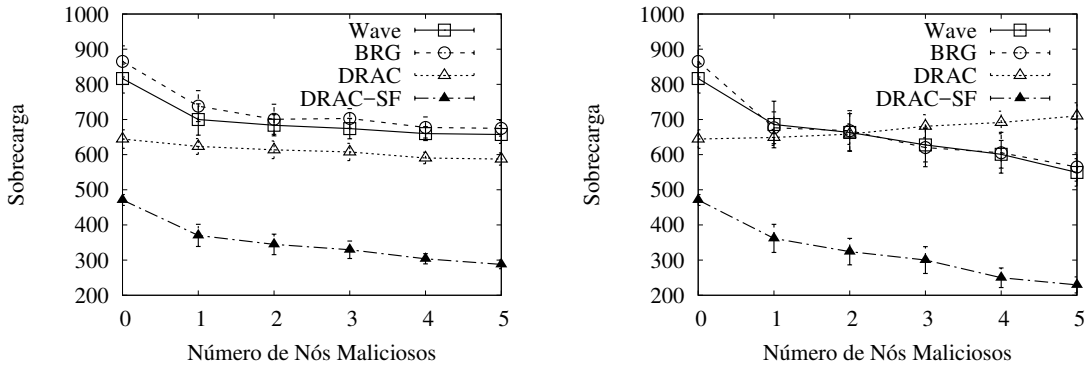
Figura C.21: Sobrecarga para o protocolo Life no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.22: Sobrecarga para o protocolo ProphetV2 no cenário Dieselnet.

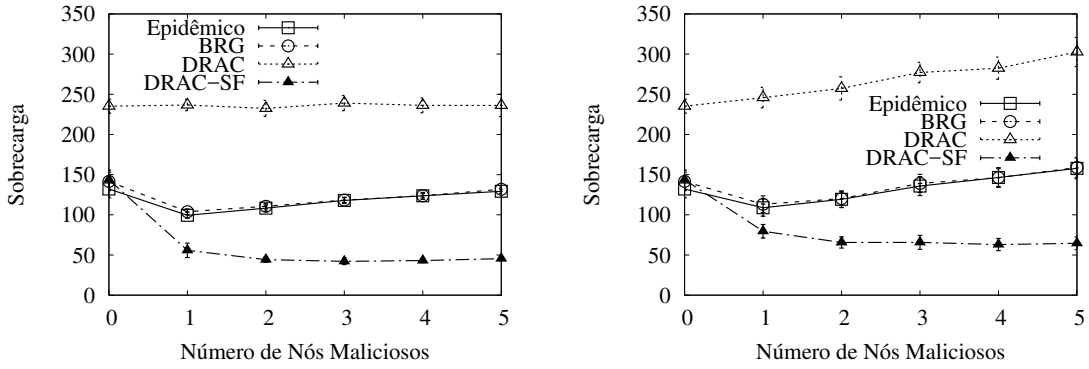


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.23: Sobrecarga para o protocolo *Wave* no cenário Dieselnet.

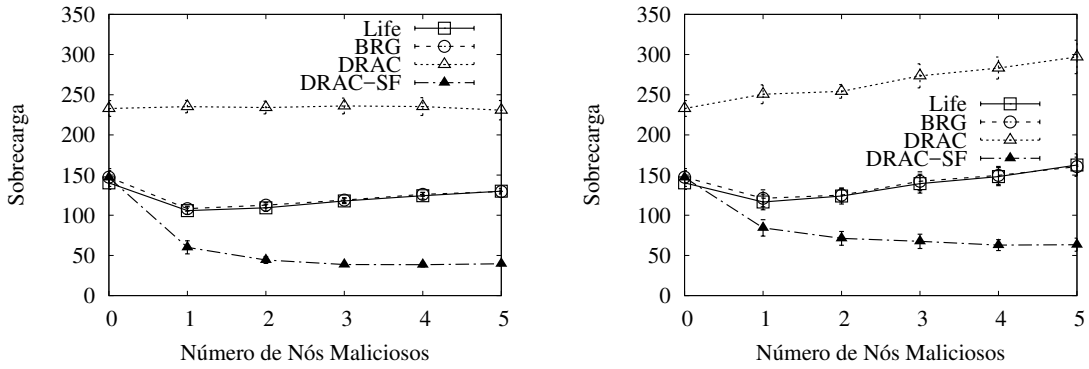
C.2.2 Cenário Infocom



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

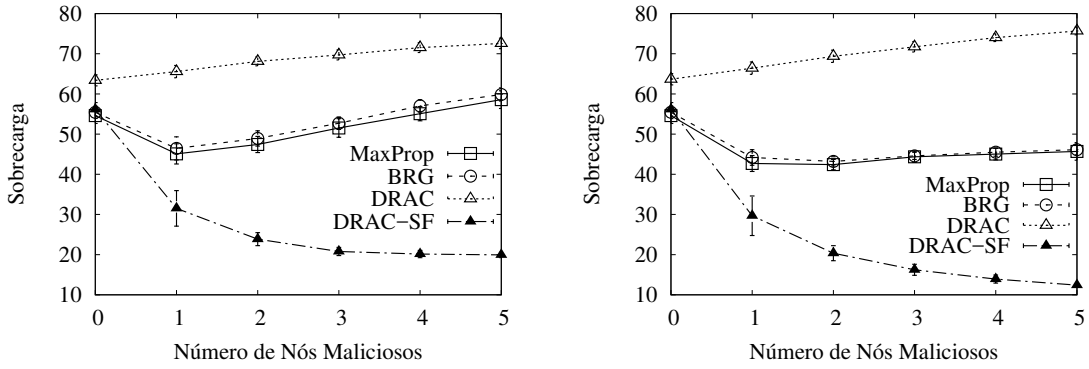
Figura C.24: Sobrecarga para o protocolo Epidêmico no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

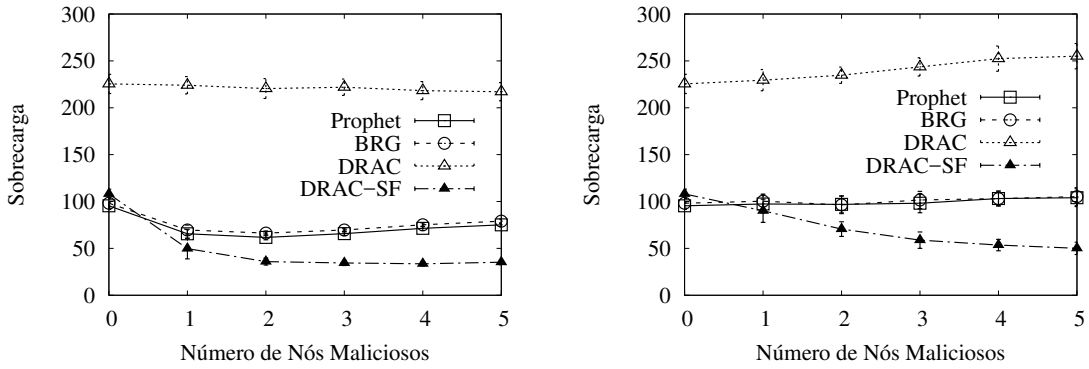
Figura C.25: Sobrecarga para o protocolo Life no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

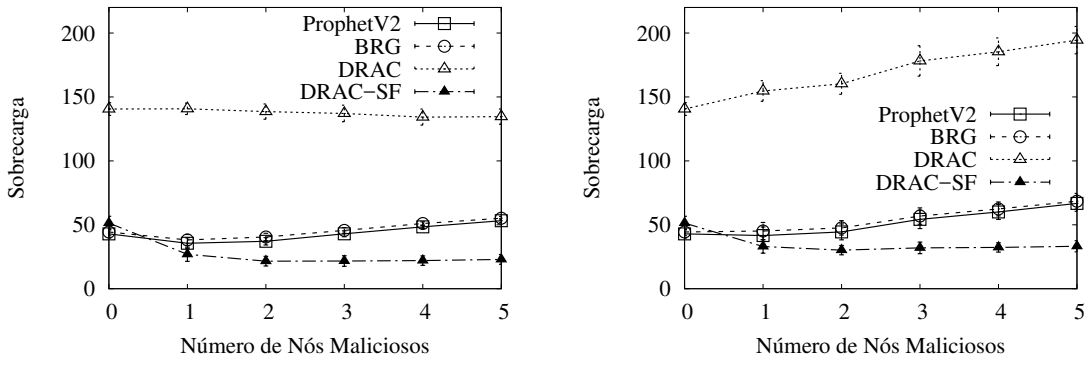
Figura C.26: Sobrecarga para o protocolo MaxProp no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

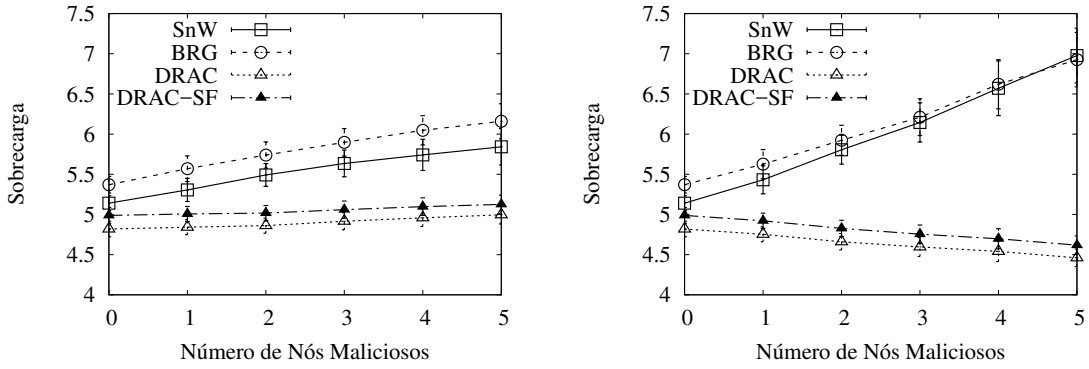
Figura C.27: Sobrecarga para o protocolo Prophet no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

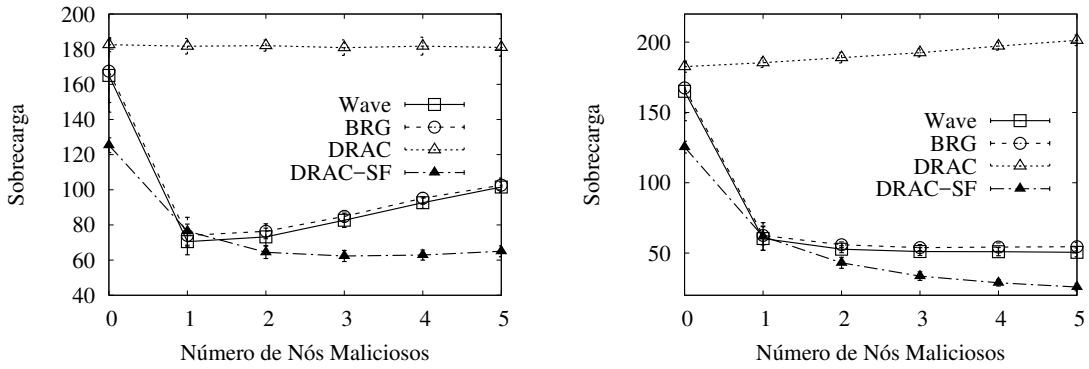
Figura C.28: Sobrecarga para o protocolo ProphetV2 no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.29: Sobrecarga para o protocolo *Spray and Wait* no cenário Infocom.

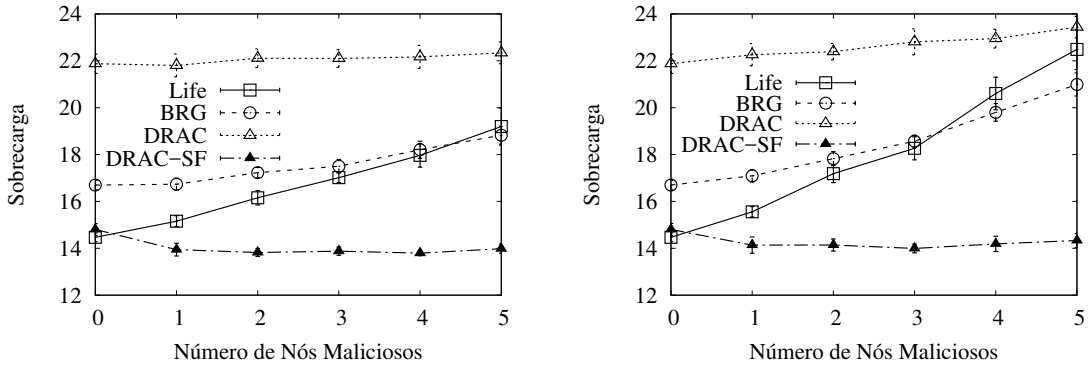


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.30: Sobrecarga para o protocolo *Wave* no cenário Infocom.

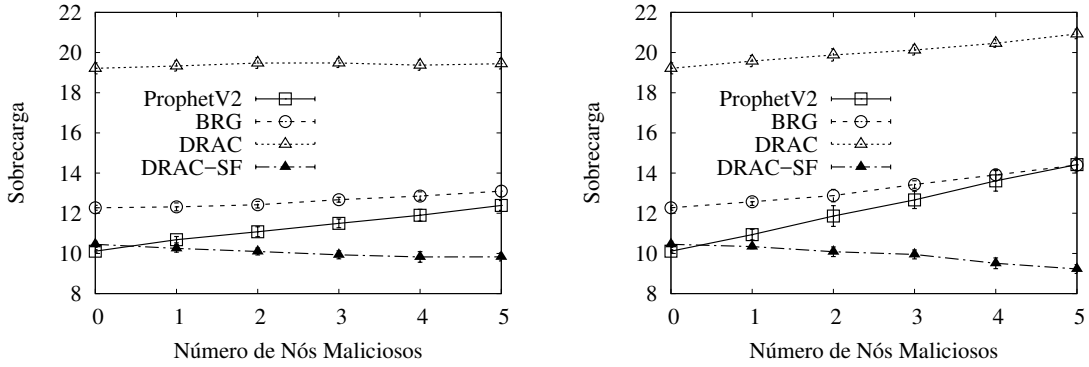
C.2.3 Cenário Rollernet



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

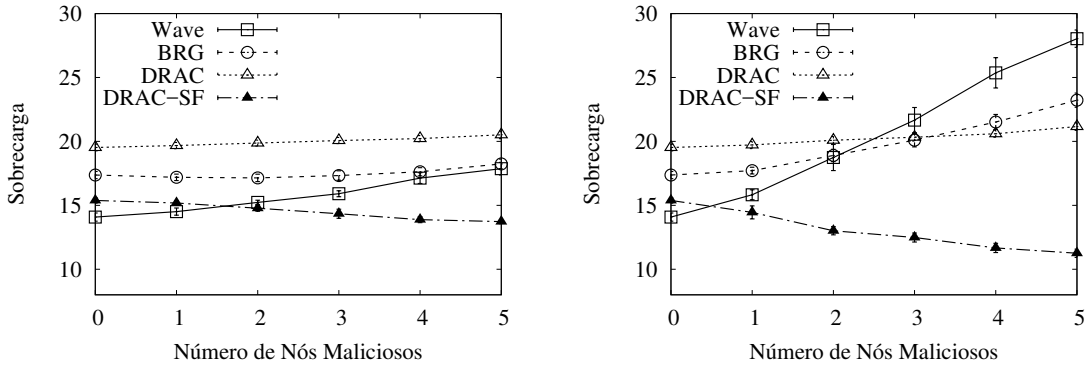
Figura C.31: Sobrecarga para o protocolo Life no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.32: Sobrecarga para o protocolo ProphetV2 no cenário Rollernet.

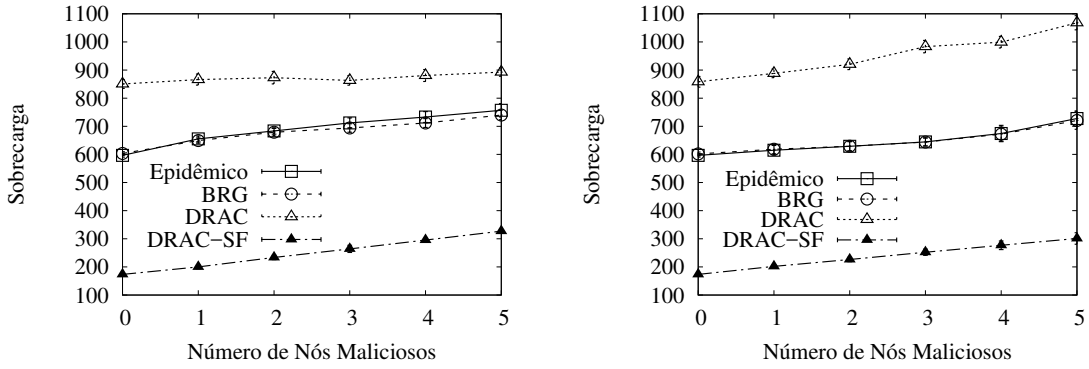


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.33: Sobrecarga para o protocolo *Wave* no cenário Rollernet.

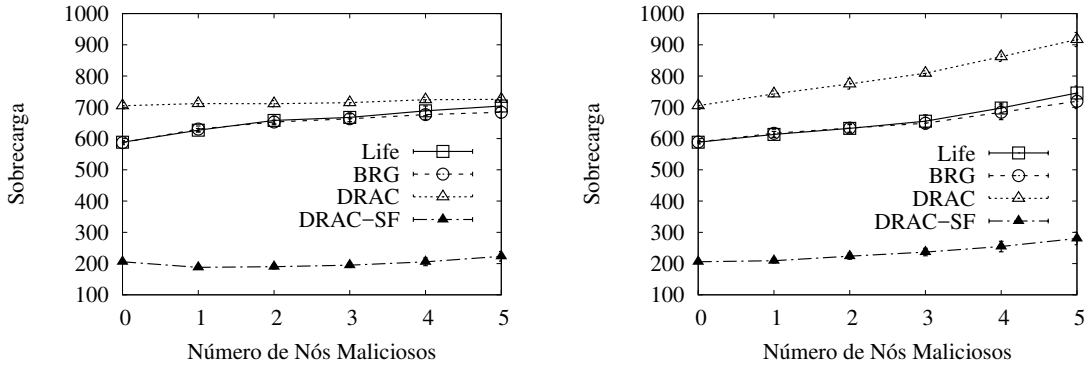
C.2.4 Cenário Shopping



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

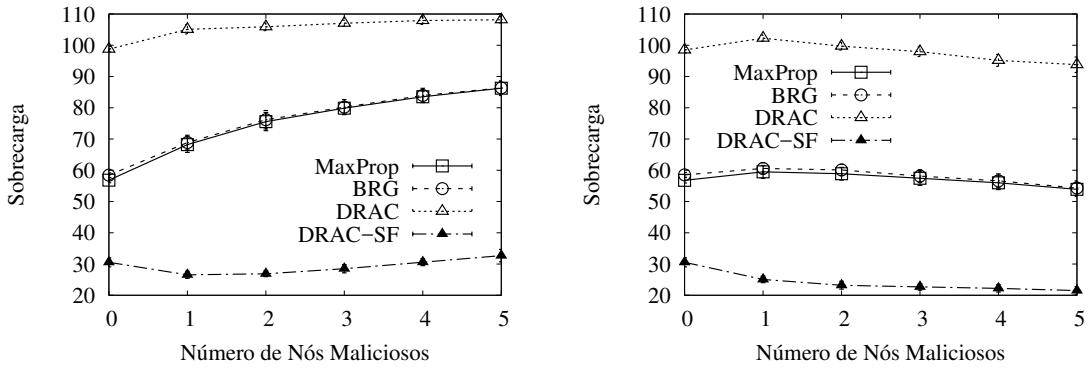
Figura C.34: Sobrecarga para o protocolo Epidêmico no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

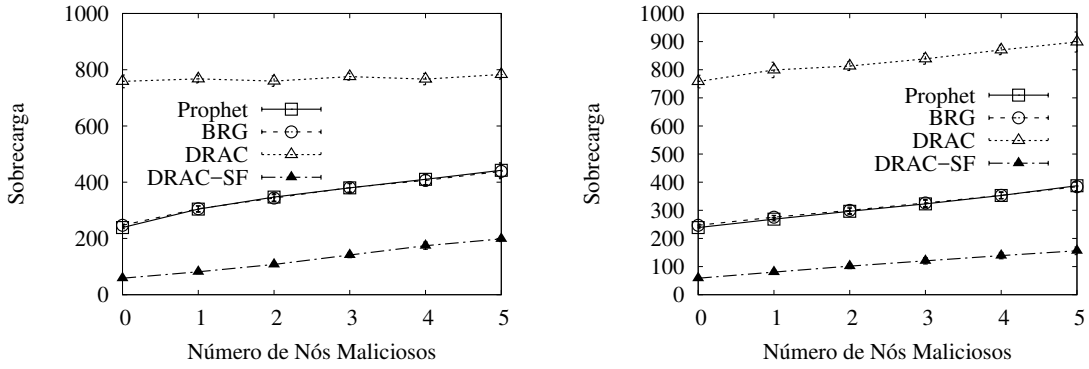
Figura C.35: Sobrecarga para o protocolo Life no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

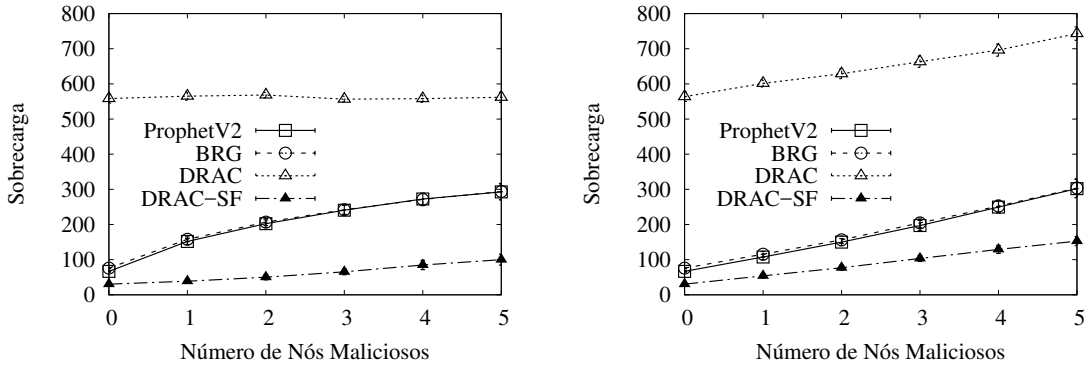
Figura C.36: Sobrecarga para o protocolo MaxProp no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

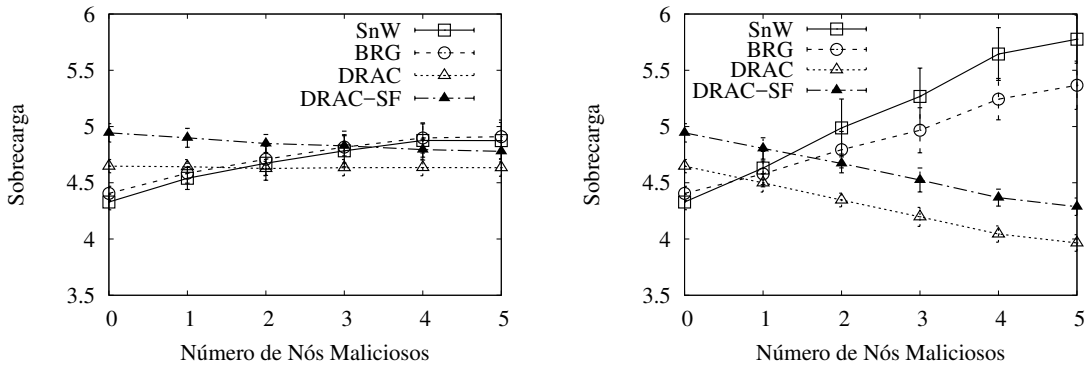
Figura C.37: Sobrecarga para o protocolo Prophet no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

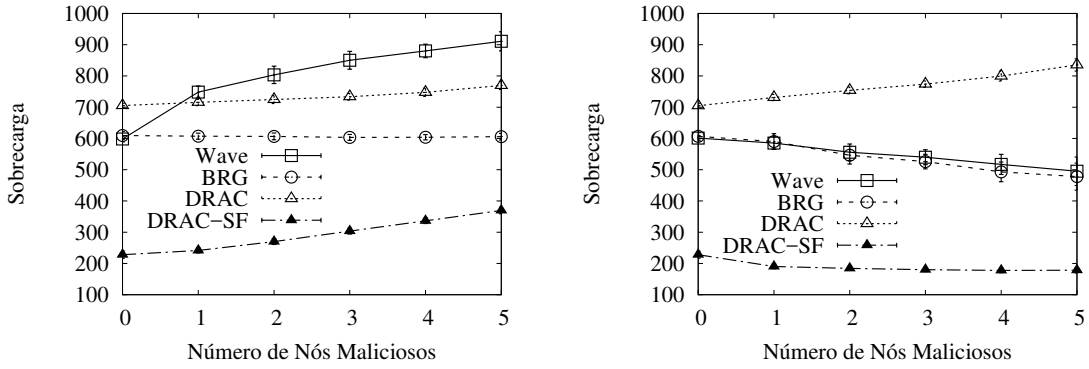
Figura C.38: Sobrecarga para o protocolo ProphetV2 no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.39: Sobrecarga para o protocolo *Spray and Wait* no cenário Shopping.



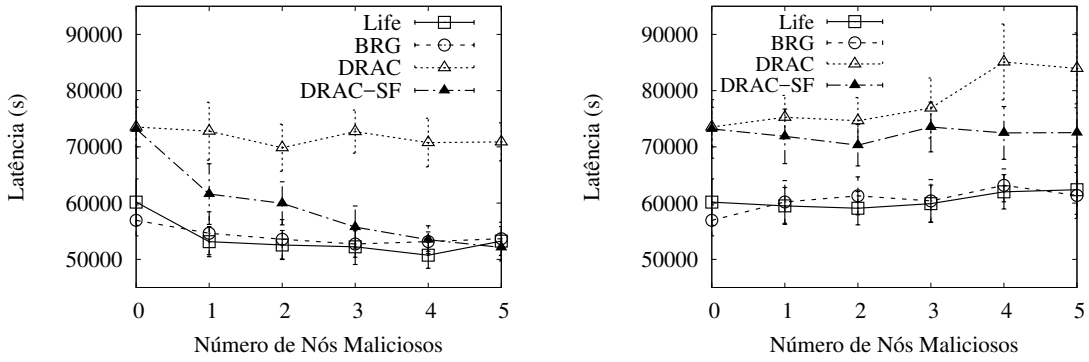
(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.40: Sobrecarga para o protocolo *Wave* no cenário Shopping.

C.3 Atraso de Entrega

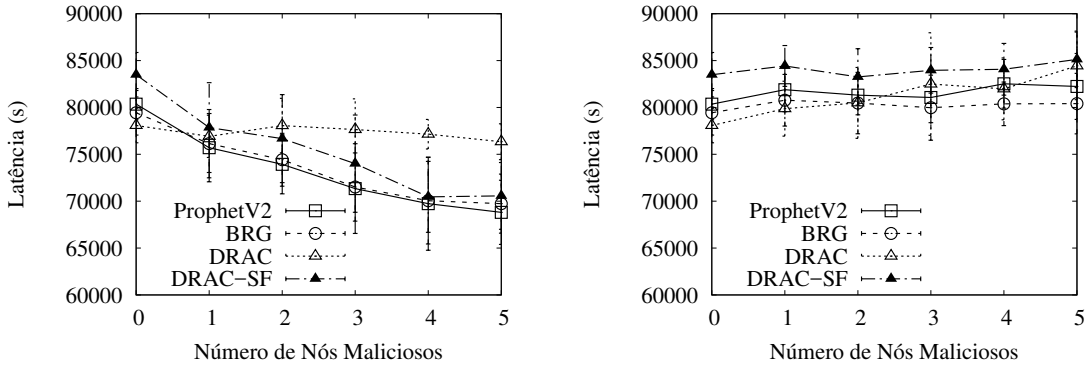
C.3.1 Cenário Dieselnet



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

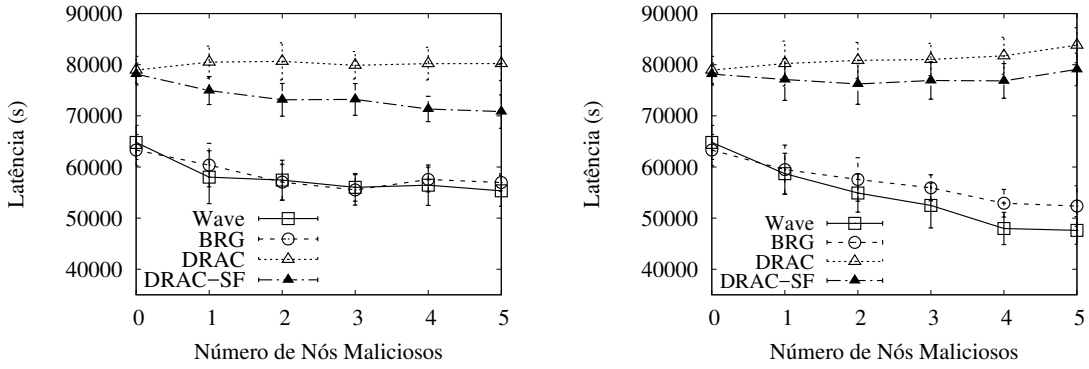
Figura C.41: Atraso de entrega para o protocolo Life no cenário Dieselnet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.42: Atraso de entrega para o protocolo ProphetV2 no cenário Dieselnet.

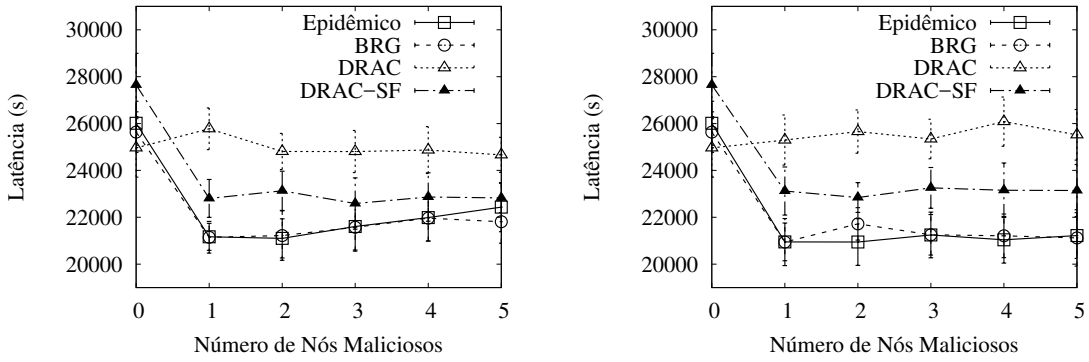


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.43: Atraso de entrega para o protocolo *Wave* no cenário Dieselnet.

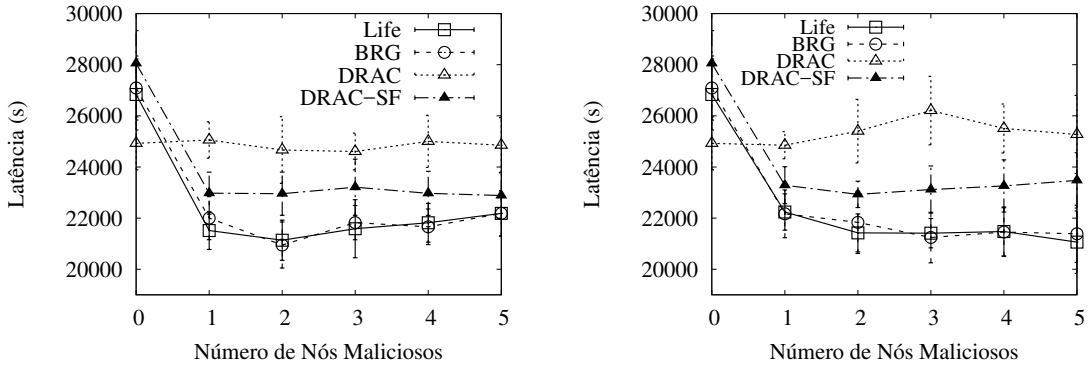
C.3.2 Cenário Infocom



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

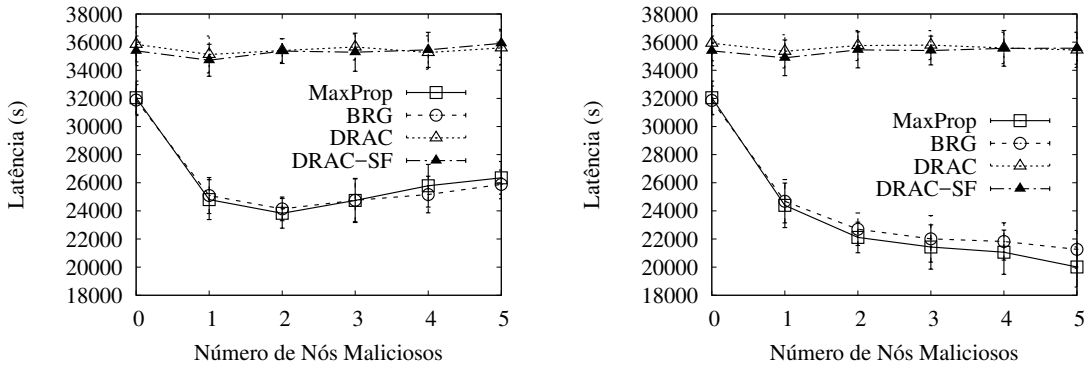
Figura C.44: Atraso de entrega para o protocolo Epidêmico no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

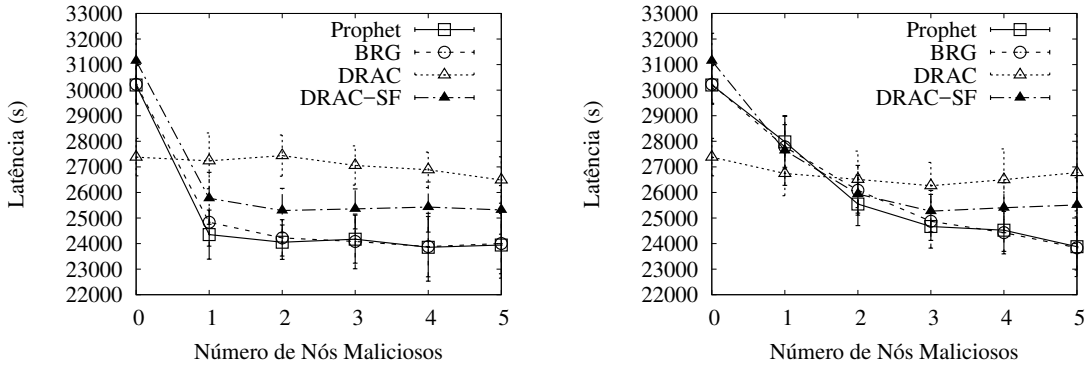
Figura C.45: Atraso de entrega para o protocolo Life no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

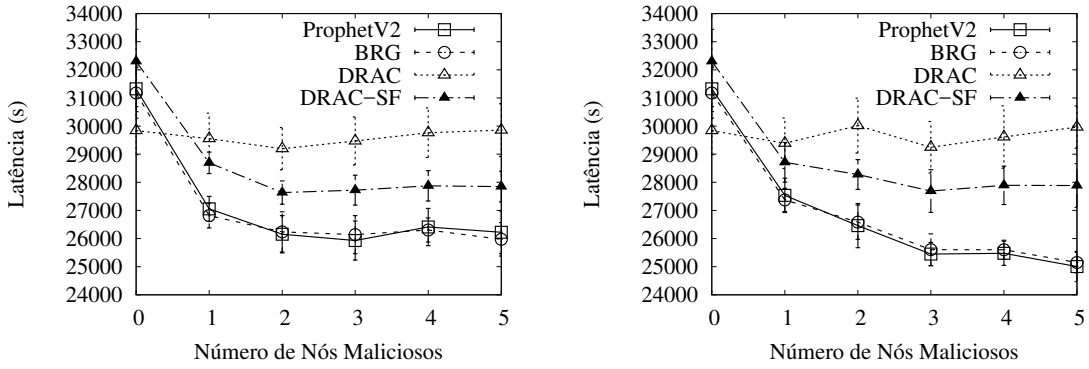
Figura C.46: Atraso de entrega para o protocolo MaxProp no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

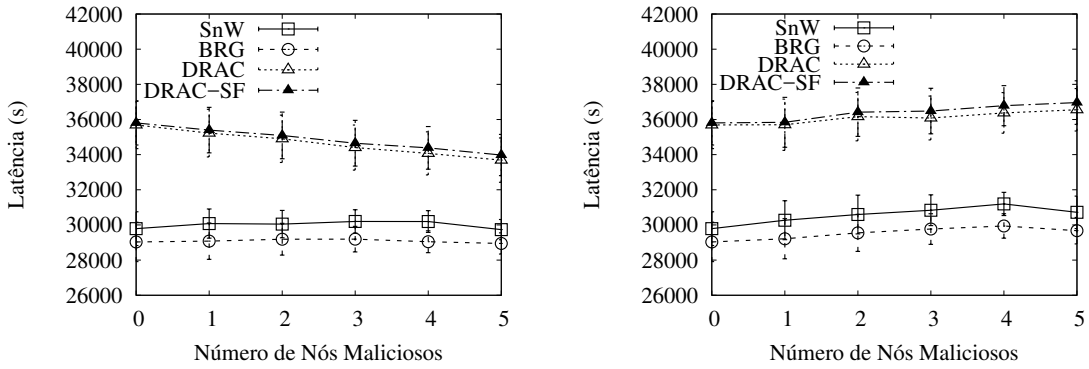
Figura C.47: Atraso de entrega para o protocolo Prophet no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

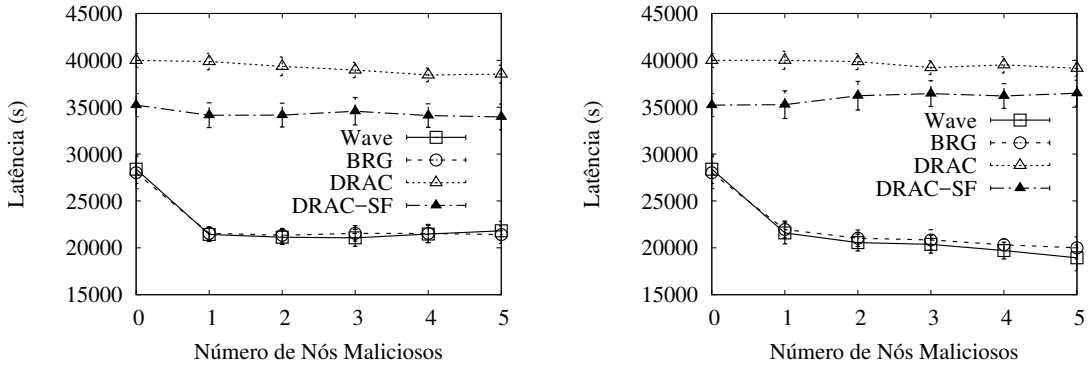
Figura C.48: Atraso de entrega para o protocolo ProphetV2 no cenário Infocom.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.49: Atraso de entrega para o protocolo *Spray and Wait* no cenário Infocom.

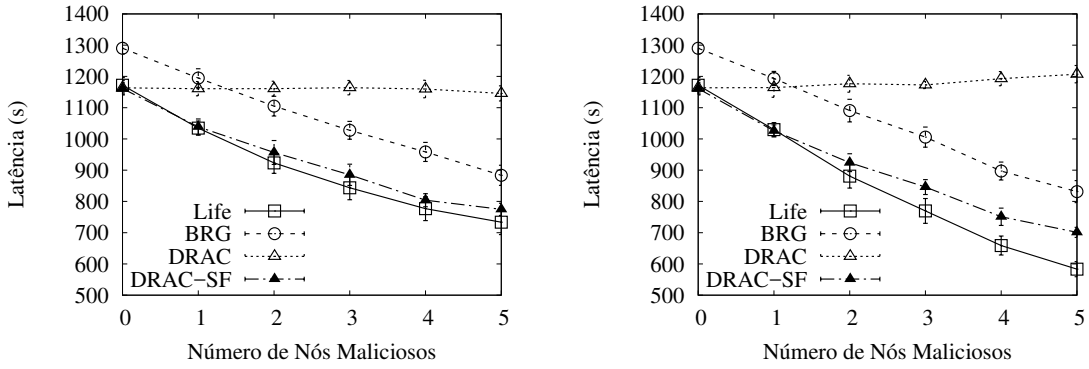


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.50: Atraso de entrega para o protocolo *Wave* no cenário Infocom.

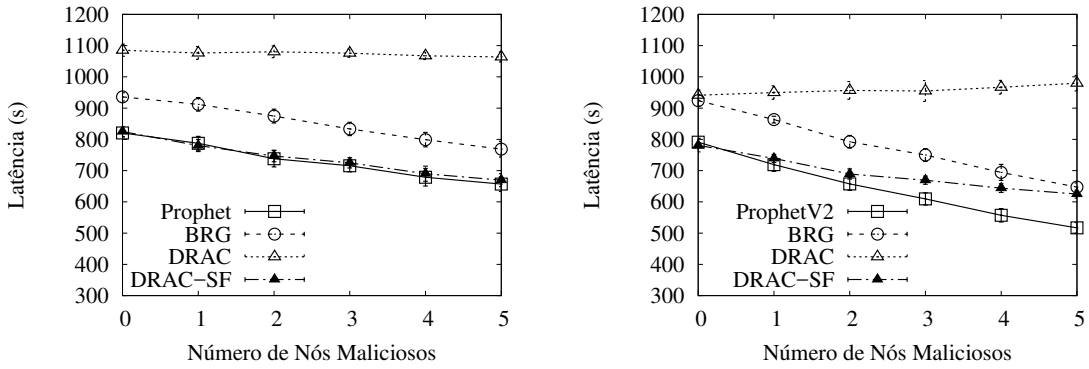
C.3.3 Cenário Rollernet



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

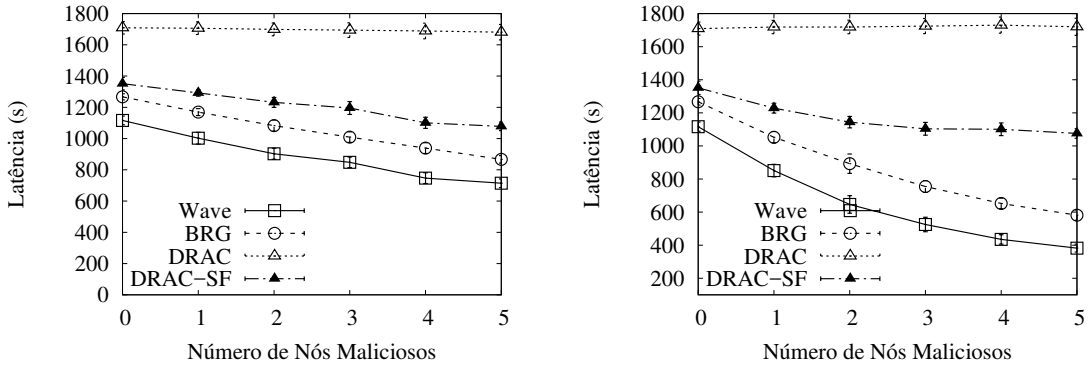
Figura C.51: Atraso de entrega para o protocolo Life no cenário Rollernet.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.52: Atraso de entrega para o protocolo ProphetV2 no cenário Rollernet.

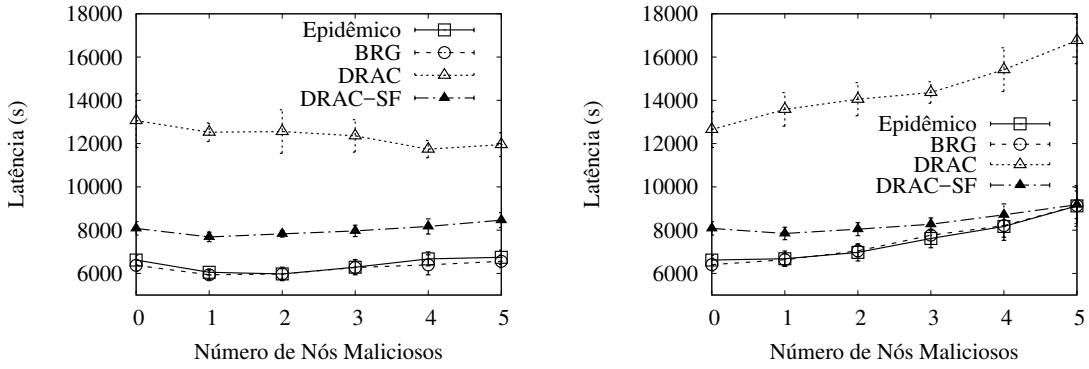


(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.53: Atraso de entrega para o protocolo *Wave* no cenário Rollernet.

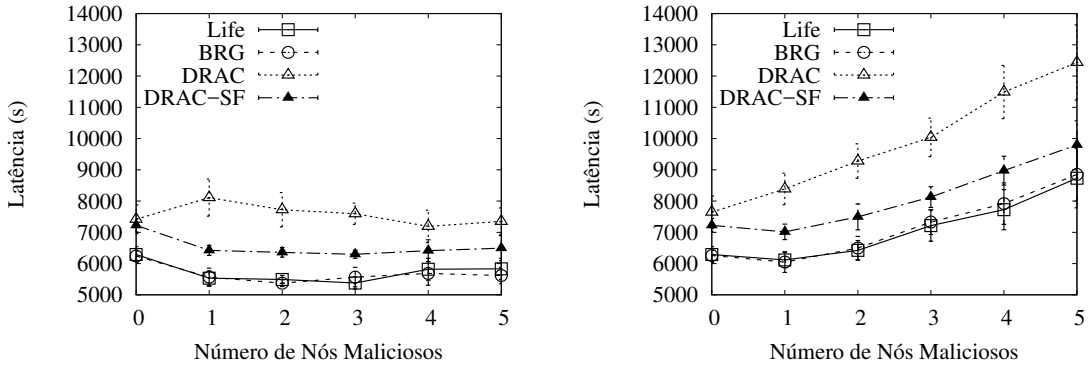
C.3.4 Cenário Shopping



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

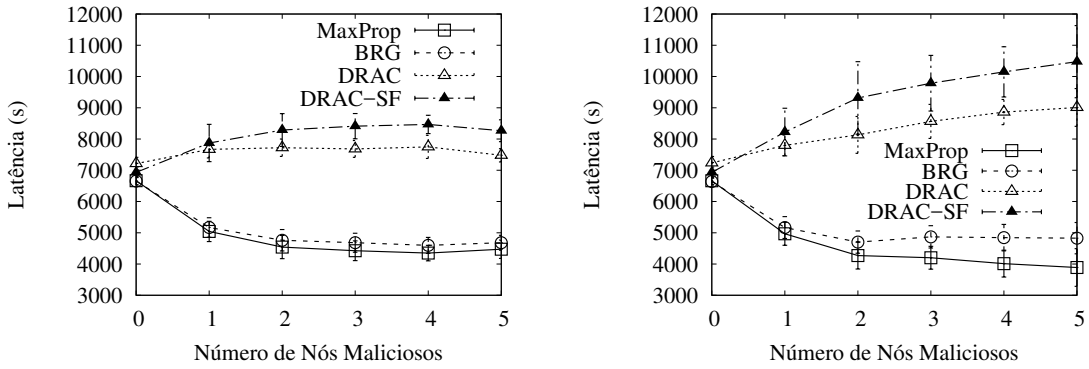
Figura C.54: Atraso de entrega para o protocolo Epidêmico no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

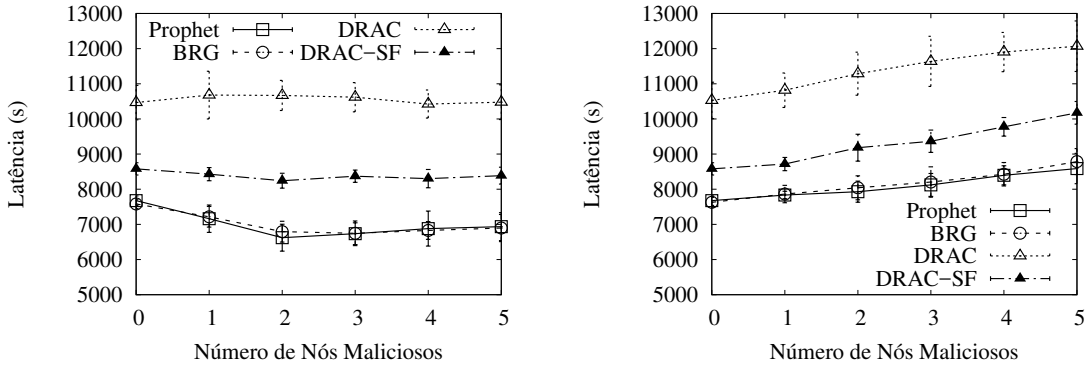
Figura C.55: Atraso de entrega para o protocolo Life no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

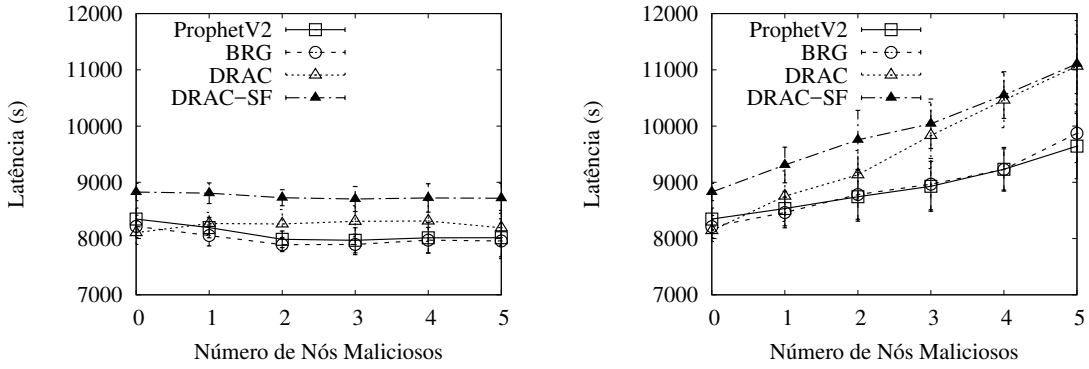
Figura C.56: Atraso de entrega para o protocolo MaxProp no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

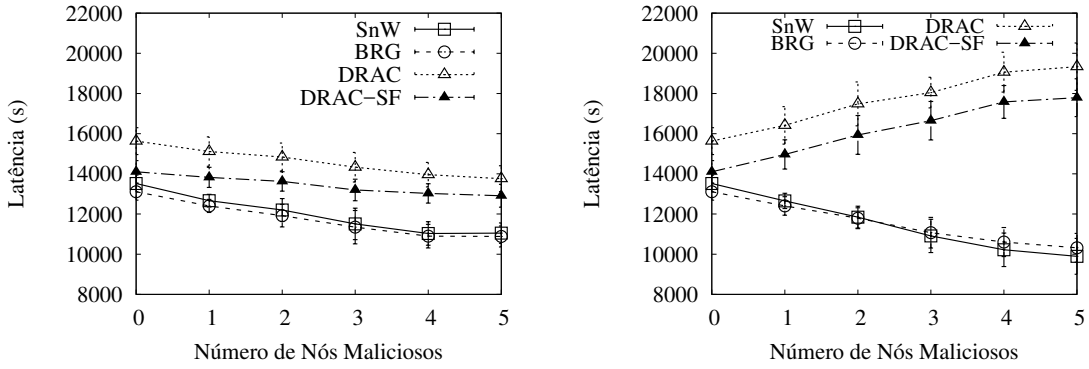
Figura C.57: Atraso de entrega para o protocolo Prophet no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

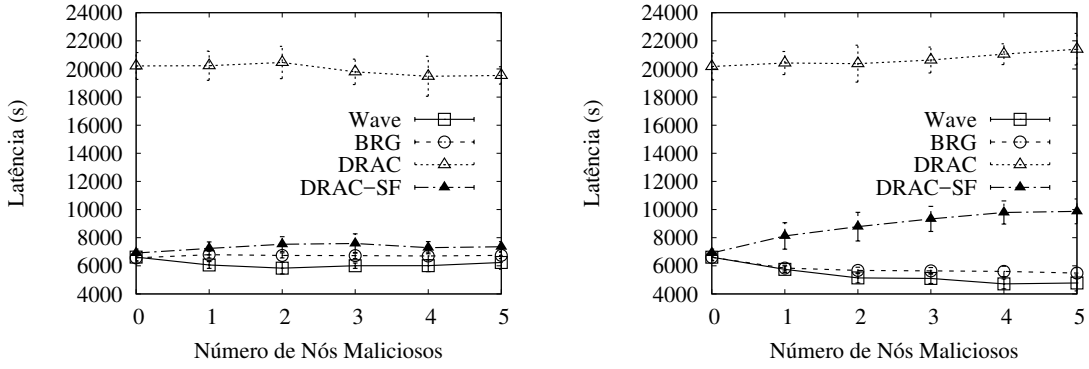
Figura C.58: Atraso de entrega para o protocolo ProphetV2 no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.59: Atraso de entrega para o protocolo *Spray and Wait* no cenário Shopping.



(a) Ataque de falsificação de reconhecimentos positivos.

(b) Ataque de falsificação de reconhecimentos positivos com buraco negro.

Figura C.60: Atraso de entrega para o protocolo *Wave* no cenário Shopping.