

# Um Mecanismo Eficiente de Controle de Congestionamento para Redes Tolerantes a Atrasos e Desconexões

Juliano F. Naves<sup>1,2</sup>, Igor M. Moraes<sup>1</sup>

<sup>1</sup>Laboratório MídiaCom, PGC-TCC / Instituto de Computação  
Universidade Federal Fluminense

<sup>2</sup>Núcleo Informatizado de Memória e Pesquisa do IFRO (NIMPI)  
Instituto Federal de Rondônia, *campus* Vilhena

juliano.naves@ifro.edu.br, igor@ic.uff.br

**Resumo.** *Protocolos de roteamento para Redes Tolerantes a Atrasos e Desconexões frequentemente utilizam a replicação para aumentar a resiliência e a taxa de entrega de mensagens. No entanto, a replicação não controlada de mensagens pode levar ao congestionamento da rede, impactando negativamente o seu desempenho. Este artigo propõe um mecanismo híbrido de controle de congestionamento simples e eficiente que tem como objetivo evitar a replicação desnecessária de mensagens e atenuar o congestionamento na rede. O mecanismo proposto baseia-se no fato de que a remoção de mensagens dos buffers dos nós deve ser persistente, ou seja, um nó não deve tornar a receber mensagens que já foram removidas de seu buffer. A análise considera quatro registros reais de mobilidade e três protocolos de roteamento bem conhecidos na literatura. Os resultados mostram que o mecanismo proposto resulta em uma redução da sobrecarga de transmissão de até 98,5%, além de aumentar a taxa de entrega e reduzir o atraso de entrega de mensagens.*

**Abstract.** *Routing protocols for Delay and Disruption Tolerant Networks frequently use message replication in order to increase the network resilience and the message delivery rate. However, uncontrolled replication may lead to network congestion, negatively impacting network performance. This paper proposes a simple and efficient hybrid network congestion control mechanism that aims at avoiding unwanted message replication and to reduce network congestion. The proposal is based on the fact that the removal of messages from nodes buffers must be persistent, i.e. a node must not receive messages that have already been removed from its buffer again. The analysis consider four real mobility traces and three well-known routing protocols. Results show the proposed mechanism reduces transmission overhead in up to 98.5% in addition to increasing delivery rate and decrease delivery latency.*

## 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) [Oliveira et al. 2007]. As redes DTN, são baseadas no paradigma armazenar-carregar-e-encaminha. Nesse 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. Essas oportunidades de encaminhamento são chamadas de contatos.

Vários protocolos de roteamento, com diferentes características, foram propostos para operarem em redes DTN [Cao e Sun 2013]. É comum que estes protocolos façam uso da replicação das mensagens como forma de aumentar a probabilidade de entrega de mensagens. No entanto, tal medida pode esgotar rapidamente os recursos da rede, levando ao congestionamento da mesma [Naves et al. 2012a, Naves et al. 2012b]. Portanto, mecanismos de controle de congestionamento são importantes para garantir que estas redes atinjam o desempenho desejado. Estes mecanismos são projetados para evitar ou mitigar os efeitos negativos causados pelo congestionamento em DTNs.

As técnicas de controle de congestionamento em DTNs podem ser divididas em reativas, proativas e híbridas. As técnicas proativas utilizam medidas que visam prevenir o acontecimento do congestionamento. Por outro lado, as técnicas reativas usualmente aguardam o acontecimento do congestionamento para que alguma ação seja tomada. Por sua vez, técnicas de controle de congestionamento híbridas combinam técnicas reativas e proativas. Neste contexto, segundo Silva *et al.*, adotar uma abordagem proativa pode melhorar significativamente o desempenho da rede [Silva et al. 2016a].

O objetivo deste trabalho, portanto, é propor e avaliar um mecanismo híbrido de controle de congestionamento para DTNs. O mecanismo baseia-se na premissa de que a remoção de mensagens dos *buffers* dos nós deve ser persistente, isto é, um nó não deve tornar a receber mensagens que já foram removidas de seu *buffer*. Para tanto, na proposta deste trabalho, os nós da rede mantêm uma lista contendo os identificadores das mensagens que foram removidas dos seus *buffers*. Desta forma, durante um contato os nós não solicitam novamente mensagens que já foram removidas. Ademais, ressalta-se que o mecanismo proposto é independente do protocolo de roteamento. Para a avaliação, são utilizados quatro registros reais de mobilidade: Rollernet, Infocom05, DieselNet e Shopping. Além disso, três protocolos de roteamento bem conhecidos na literatura são utilizados, são eles: Epidêmico, MaxProp e Prophet. Os resultados demonstram que o mecanismo proposto é eficiente, reduzindo a sobrecarga de transmissão de mensagens em até 98,5%, além de aumentar a taxa de entrega e reduzir o atraso de entrega de mensagens.

O restante deste trabalho é organizado como a seguir. Na Seção 2, os trabalhos relacionados são descritos. O mecanismo de controle de congestionamento proposto neste trabalho é apresentado na Seção 3. A Seção 4 apresenta o ambiente de avaliação da proposta, assim como os registros de mobilidade reais utilizados nas simulações. Os resultados obtidos através das simulações são apresentados na Seção 5. Finalmente, as conclusões alcançadas no decorrer do presente trabalho, bem como possíveis extensões a

este trabalho e prováveis trabalhos futuros são apresentadas na Seção 6.

## 2. Trabalhos Relacionados

O controle de congestionamento é uma preocupação em DTNs desde a concepção desta arquitetura de rede. Recentemente, esforços foram empregados na análise, caracterização [Soelistijanto e Howarth 2014, Silva et al. 2015b] e também na avaliação [Silva et al. 2016a, Silva et al. 2015a] de mecanismos de controle de congestionamento propostos pela comunidade. Silva *et al.* classificam os mecanismos de controle de congestionamento em proativos, reativos e híbridos [Silva et al. 2015b]. Segundo os autores, mecanismos de controle de congestionamento proativos, também conhecidos como mecanismos de prevenção de congestionamento (*congestion avoidance*), agem de forma preventiva para tentar evitar que o congestionamento ocorra na rede. Por outro lado, os mecanismos de controle de congestionamento reativos esperam o acontecimento do congestionamento para entrarem em ação. Além disso, são mencionados mecanismos híbridos, que combinam as abordagens proativa e reativa.

Em trabalhos posteriores, Silva *et al.* fazem estudos comparativos de mecanismos de congestionamento em DTNs [Silva et al. 2016a, Silva et al. 2015a]. Segundo os autores, a adoção de uma abordagem proativa ou híbrida pode melhorar significativamente o desempenho da rede. Com relação aos mecanismos reativos, estes podem não alcançar um desempenho satisfatório devido aos grandes atrasos de comunicação que podem existir em DTNs. Como os mecanismos reativos esperam o acontecimento do congestionamento, estes podem sofrer um grande impacto do atraso de comunicação inerente às DTNs, o que atrasa as decisões tomadas por estes mecanismo e conseqüentemente agrava o congestionamento na rede.

É interessante notar que embora existam propostas restritas ao controle de congestionamento, protocolos de roteamento frequentemente especificam mecanismos de controle de congestionamento para sua operação. Este é o caso específico do protocolo MaxProp [Burgess et al. 2006], que especifica um mecanismo híbrido de controle de congestionamento. Neste protocolo, as mensagens possuem uma lista de saltos (*hop list*) na qual a mensagem armazena os nós pelos quais ela já passou. Desta forma, um nó  $A$  envia uma mensagem  $M_1$  para um determinado nó  $B$ , se e somente se,  $B$  não está na lista de saltos de  $M_1$ . Além disso, o nó  $A$  envia a mensagem  $M_1$  para  $B$ , se e somente se já não tiver enviado esta mensagem para  $B$  anteriormente. Este caso em específico difere do caso anterior, visto que a mensagem  $M_1$  pode ter trafegado por diferentes caminhos. O mecanismo da lista de saltos foi implementado para a avaliação conduzida neste trabalho e este mecanismo será mencionado daqui em diante como *Hop List* (HL). Os outros dois mecanismos utilizados pelo protocolo MaxProp para lidar com o congestionamento da rede são o envio de reconhecimentos positivos [Burgess et al. 2007, Naves e Moraes 2014] e a política de gerenciamento de *buffer*.

A despeito das inúmeras pesquisas existentes, o congestionamento em DTNs continua sendo objetivo de trabalhos recentes. Este é o caso do trabalho de Silva *et al.*, que propõe um mecanismo de controle de congestionamento que tem como base a aprendizagem de máquina [Silva et al. 2016b]. Por sua vez, Liu *et al.* propõem um mecanismo de controle de congestionamento que considera as características sociais do nó, nomeado *Social Awareness based Congestion Control* (SACC) [Liu et al. 2016]. As características

sociais também são abordadas por Wei *et al.* em sua proposta denominada *Context-Aware Congestion Control Approach* (CACC) [Wei et al. 2014]. No entanto, o número de variáveis que representam o contexto da rede e que são utilizadas por estes mecanismo pode dificultar a implementação dos mesmos.

Sendo assim, este trabalho propõe um mecanismo de controle de congestionamento híbrido e fácil de ser implantado, visto que utiliza somente informações locais e não exige a troca de informações entre os nós da rede.

### 3. O Mecanismo Proposto

O objetivo de uma política de gerenciamento de *buffer* é maximizar o desempenho da rede, mantendo em *buffer* as mensagens com maior probabilidade de serem entregues ao destino. Desta forma, evitando causar impacto negativo pelo descarte inadequado de uma mensagem. Em casos específicos, como quando a mensagem já chegou ao destino, o descarte de uma mensagem pode ter impacto positivo no desempenho da rede, visto que impede que uma mensagem que já chegou ao destino continue a ser replicada, diminuindo a eficiência da rede. Em face do exposto, é razoável considerar que uma mensagem que foi descartada do *buffer* de um determinado nó utilizando uma política de gerenciamento de *buffer* que atenda ao objetivo exposto anteriormente possui baixa probabilidade de entrega. Isto porque a política visa manter as mensagens com maior probabilidade de entrega e portanto descartar as mensagens com menor probabilidade de entrega. Logo, é plausível assumir que o envio de uma determinada mensagem  $M_1$  para um determinado nó  $A$ , após este nó já ter removido esta mensagem de seu *buffer* através de uma política de gerenciamento de *buffer*, é ineficiente. Em outras palavras, o descarte de uma mensagem deve ser permanente. Esta é a premissa utilizada pela proposta deste trabalho.

No mecanismo proposto, daqui em diante chamado de *Message List* (ML), os nós da rede mantêm uma lista contendo os identificadores de todas as mensagens que já descartaram. Desta forma, no início de um contato entre dois nós, durante a troca dos vetores de mensagens, ambos não solicitam o envio de mensagens que já receberam em algum momento e que foram descartadas de seus *buffers*. Ressalta-se que o mecanismo ML é um mecanismo de controle de congestionamento híbrido, visto que atua tanto reativamente quanto proativamente. A característica reativa advém do fato do mecanismo basear-se em uma política de gerenciamento de *buffer* para o seu funcionamento. Por outro lado, o mecanismo ML é também proativo, visto que impõe *a priori* um limite máximo de réplicas e de encaminhamentos na rede. Isto é, seja  $n$  o número de nós presentes na rede, uma mensagem só poderá ser replicada  $n$  vezes na rede. Com relação aos encaminhamentos, estas estão limitadas a  $n - 1$ , visto que um nó pode transmitir uma mensagem somente uma vez para todos os seus vizinhos.

Neste trabalho, assume-se que os nós da rede possuem um espaço reservado suficientemente grande para armazenar todos os identificadores das mensagens descartadas por eles, de modo que o mecanismo ML funcione plenamente. No entanto, a seguir é feita uma breve análise da sobrecarga de armazenamento resultado do mecanismo ML. As mensagens podem ser identificadas unicamente através do identificador da fonte da mensagem e da marca de tempo (*timestamp*) desta mensagem. A especificação do protocolo *Bundle* [Scott e Burleigh 2007] restringe o tamanho máximo dos identificadores a 2046 bytes, representados como *strings*. Com relação à marca de tempo, na especificação

do protocolo *Bundle* ela possui tamanho variável. No entanto, a utilização de 4 bytes é suficiente para representar cerca de 136 anos, como acontece com a marca de tempo padrão Unix. Isto impõe uma sobrecarga de 2.050 bytes, ou 16.400 bits, por mensagem. Uma forma de reduzir esta sobrecarga é utilizar uma função de dispersão. Uma alternativa é a utilização do *Message-Digest algorithm 5* (MD5). Desta forma, aplicando-se a função MD5 ao par identificador da fonte e marca de tempo, a sobrecarga pode ser reduzida a 128 bits por mensagem.

A possibilidade de que os nós não tenham espaço suficiente para armazenar todos os identificadores das mensagens descartadas durante toda a operação da rede é admissível. Neste caso, quando a lista utilizada pelo mecanismo de controle de congestionamento proposto neste trabalho tiver tamanho limitado, ela pode ser gerenciada de modo similar ao modo em que o *buffer* dos nós é gerenciado, isto é, através das políticas de gerenciamento de *buffer*. Então, quando ocorrer a sobrecarga da lista em questão, isto é, quando um novo identificador de mensagem precisar ser armazenado e a lista não tiver espaço suficiente, uma política para o gerenciamento desta lista é acionada para escolher um identificador a ser descartado, de forma a liberar espaço para o recebimento do novo identificador. Entre as heurísticas que podem ser utilizadas para o gerenciamento da lista, cita-se a bem conhecida FIFO (*First In First Out*). Além disso, uma nova heurística que considere o identificador utilizado menos recentemente, similar ao bem conhecido algoritmo de gerenciamento de *cache Least Recently Used* (LRU) ou a política de gerenciamento de *buffer Least Recently Forwarded* [Naves et al. 2012a].

Para efeito de análise preliminar, simulações foram executadas com memória finita para armazenar os identificadores das mensagens. O espaço reservado na memória para o armazenamento dos identificadores foi variado entre 10 e 500 identificadores. Duas estratégias para substituição de identificadores na lista foram implementadas: FIFO e LRU. Os resultados indicam que a partir de 100 identificadores o desempenho torna-se equivalente a utilização de memória infinita. De qualquer forma, assumir um espaço reservado de memória infinito na rede serve de guia para comprovar que a premissa da proposta está correta: as mensagens devem ser permanentemente descartadas.

Finalmente, observa-se que como não utiliza informações específicas do protocolo de congestionamento, como por exemplo, a probabilidade de entrega calculada por alguns destes protocolos, o mecanismo proposto é independente de protocolo de roteamento. Portanto, é possível classificá-lo como de fácil implantação.

## 4. Cenários de Avaliação

### 4.1. Conjuntos de Dados e Roteamento

Três protocolos bem conhecidos da literatura foram utilizados na avaliação do presente estudo, são eles, Epidêmico, Prophet e MaxProp. Destaca-se que os três protocolos lidam com a replicação de mensagens de maneira distinta e portanto, atingem diferentes níveis de congestionamento. O protocolo Epidêmico não impõe qualquer controle na replicação de mensagens, sendo o mais suscetível ao congestionamento. O protocolo Prophet utiliza 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 protocolo Prophet é menos suscetível ao congestionamento do que o protocolo Epidêmico. Por último, o protocolo MaxProp utiliza mecanismos que impõem um limite estrito de réplicas e encaminhamentos, como detalhado na Seção 2, logo, entre os protocolos avaliados, é o protocolo menos suscetível ao congestionamento.

Quatro registros reais de mobilidade, são utilizados para avaliar o desempenho da proposta. O conjunto de registros Rollernet [Tournoux et al. 2009] é resultado de um experimento no qual foram distribuídos 62 iMotes para voluntários. Este conjunto tem duração aproximada de 3 horas. No conjunto de registros Infocom05 [Hui et al. 2005] 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 [Zhang et al. 2007, Burgess et al. 2006]. Por último, o conjunto Shopping [Galati et al. 2015] é resultado de um experimento no qual 25 dispositivos foram distribuídos em um *shopping*.

#### 4.2. Ambiente de Simulação

A proposta deste trabalho foi implementada no simulador *Opportunistic Network Environment* (ONE). As simulações foram executadas para os quatro conjunto de dados distintos, com os três protocolos de roteamento escolhidos, resultando em 12 cenários diferentes. Para efeito de comparação, o mecanismo *Hop List*, presente no protocolo MaxProp, como descrito na Seção 2, foi implementado nos protocolos Epidêmico e Prophet. Adicionalmente, uma versão do protocolo MaxProp sem a utilização do mecanismo HL também foi implementada. Desta forma, pretende-se avaliar os efeitos destes diferentes mecanismos em três protocolos que lidam de maneira distinta com a replicação de mensagens.

Os parâmetros de configuração das simulações também são resumidos na Tabela 1. O padrão de tráfego de mensagens para cada conjunto de registros é o seguinte. Para o Rollernet, 500 mensagens são geradas durante as duas primeiras horas de um tempo total de aproximadamente 3 horas de simulação. Para os conjuntos de dados DieselNet, Infocom05 e Shopping, foram geradas 1.000 mensagens nas primeiras, respectivamente, 143, 52 e 125 horas, aproximadamente, de um tempo de simulação total de aproximadamente 187, 72 e 132 horas. 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. Para os cenários Dieselnet, Infocom05 e Shopping, o tamanho do *buffer* foi configurado para 20 M, enquanto que no cenário Rollernet, devido a seu tempo de duração mais curto e ao menor número de mensagens geradas, o tamanho do *buffer* foi configurado para 10 M.

### 5. Resultados

O mecanismo de controle de congestionamento proposto neste trabalho é comparado através de simulações com o mecanismo HL, implementado pelo protocolo de

**Tabela 1. Parâmetros das simulações e características dos cenários.**

Parâmetros/Conjuntos	DieselNet	Infocom05	Rollernet	Shopping
Dispositivo	802.11	iMote	iMote	<i>Bluetooth</i>
Duração ( $\approx$ )	7,8 dias	3 dias	3 horas	5,5 dias
Número de nós	31	41	62	25
Número de mensagens	1.000	1.000	500	1.000
Tamanho das mensagens (MB)	1	1	1	1
Taxa de Transmissão (Mbps)	1	1	1	1
TTL	$\infty$	$\infty$	$\infty$	$\infty$
Tamanho do <i>Buffer</i> (MB)	20	20	10	20

roteamento MaxProp, como observado na Seção 2. Além disso, o mecanismo é comparado com uma versão não modificada dos protocolos escolhidos. Para avaliar estes mecanismos 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. 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%, representado por barras verticais.

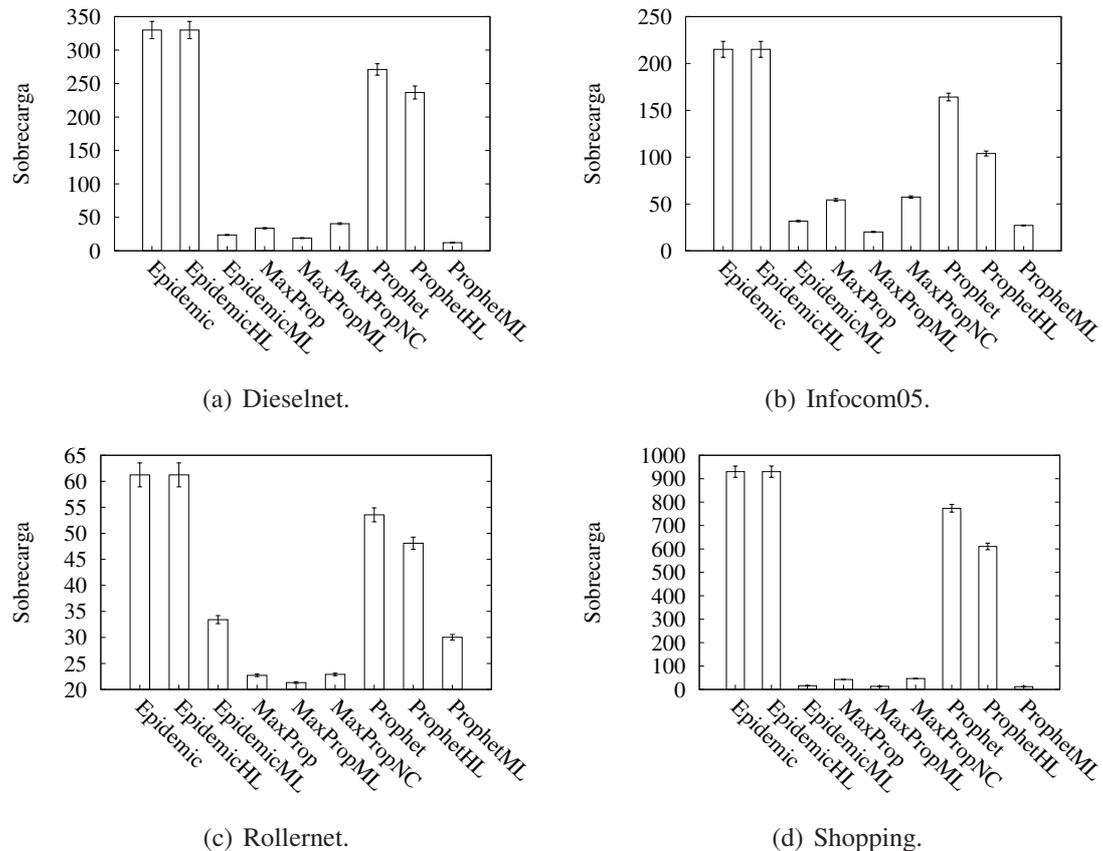
Nas figuras exibidas nesta seção, assume-se a seguinte nomenclatura. Epidemic, MaxProp e Prophet são os protocolos escolhidos para avaliação sem nenhuma modificação nos respectivos modos de operação. EpidemicML, MaxPropML e ProphetML são os protocolos dotados do mecanismo de controle de congestionamento proposto neste trabalho. EpidemicHL e ProphetHL são os protocolos Epidêmico e Prophet, respectivamente, dotados do mecanismo HL, que já é implementado no protocolo MaxProp por padrão. Por último, o nome MaxPropNC é atribuído a uma versão modificada do protocolo MaxProp que não implementa o mecanismo HL.

### 5.1. Sobrecarga

A sobrecarga  $S$  é calculada pela razão entre a diferença do número de mensagens encaminhadas  $m_t$  e mensagens entregues  $m_e$  e o número de mensagens entregues  $m_e$ , ou seja,  $S = \frac{m_t - m_e}{m_e}$ . Essa relação expressa a quantidade de mensagens adicionais que foram transmitidas para cada mensagem que chegou ao destino. Nos protocolos avaliados nesse trabalho, essa relação também indica a quantidade de réplicas criadas para cada mensagem entregue. Isto se deve ao fato destes protocolos replicarem as mensagens cada vez que a encaminham. Ou seja, mantêm a mensagem armazenada em *buffer*, mesmo após o encaminhamento. Adicionalmente, é interessante ressaltar que a sobrecarga de transmissão pode representar uma medida da eficiência energética dos protocolos, visto que a maior parte da energia de um nó sem-fio é consumida nos estados de transmissão e recepção de pacotes [Cunha et al. 2005].

A Figura 1 apresenta a sobrecarga de encaminhamentos para todos os cenários avaliados. Especificamente, a Figura 1(a) apresenta os resultados para o cenário Dieselnet, a Figura 1(b) apresenta os resultados para o cenário Infocom05, a Figura 1(c) apresenta os resultados para o cenário Rollernet e finalmente, a Figura 1(d) ilustra os resultados para o cenário Shopping.

É interessante observar que o mecanismo de controle de congestionamento pro-



**Figura 1. Sobrecarga de encaminhamento para os cenários avaliados.**

posto neste trabalho diminui a sobrecarga para todos os cenários avaliados. Para o cenário Dieselnet, como é possível observar na Figura 1(a), os protocolos ProphetML, MaxPropML e EpidemicML obtiveram a menor sobrecarga de transmissão. Observa-se que enquanto o protocolo Prophet original tem uma sobrecarga de cerca de 271 encaminhamentos, o protocolo ProphetML, que implementa a proposta deste trabalho, obtém uma sobrecarga de cerca de 12 encaminhamentos. Isto representa uma redução de cerca de 96% na sobrecarga. Por sua vez, o protocolo Epidêmico resultou em uma sobrecarga de 330 encaminhamentos enquanto que o protocolo EpidemicML resultou em uma sobrecarga de cerca de 24 mensagens, o que representa uma redução de cerca de 93% na sobrecarga de transmissão de mensagens. Por último, o protocolo MaxProp incorre em uma sobrecarga de transmissão de mensagens de cerca de 34 encaminhamentos, enquanto que o protocolo MaxPropML resulta em cerca de 19 encaminhamentos por mensagem entregue e isto uma redução de cerca de 44% na sobrecarga. É importante ressaltar que esta redução na sobrecarga de encaminhamento não tem impacto negativo na taxa de entrega. Pelo contrário, o mecanismo proposto leva a um aumento na taxa de entrega porque torna a rede mais eficiente ao evitar a replicação e o encaminhamento de mensagens com pouca probabilidade de entrega, como será discutido na Seção 5.2.

Passando para o cenário Infocom05, exibido na Figura 1(b), observa-se que os protocolos Epidêmico, Prophet e MaxProp resultam em uma sobrecarga de cerca de 215, 164 e 54 encaminhamentos, respectivamente. Por outro lado, as implementações que uti-

lizam a proposta deste trabalho, isto é, EpidemicML, ProphetML e MaxPropML, resultam em uma sobrecarga de cerca de 32, 27 e 20 encaminhamentos, respectivamente. Ou seja, a utilização do mecanismo ML resulta em uma redução de cerca e 85% da sobrecarga de transmissão para o protocolo Epidêmico, 83% para o protocolo Prophet e 63% para o protocolo MaxProp.

Seguindo com a avaliação do cenário Rollernet, ilustrado na Figura 1(c), é possível observar que este é o cenário com a menor sobrecarga de transmissão. Ainda assim, a utilização do mecanismo ML resultou em redução da sobrecarga de transmissão. Especificamente, a sobrecarga caiu de cerca de 61 encaminhamentos por mensagem entregue para o protocolo Epidêmico para cerca de 33 encaminhamentos para o protocolo EpidemicML, uma redução de cerca de 46%. Quando implementado no protocolo Prophet, o mecanismo proposto reduziu a sobrecarga de encaminhamento de cerca de 53 encaminhamentos para cerca de 30 encaminhamentos, uma redução de aproximadamente 43%. O protocolo MaxProp foi o protocolo que menos beneficiou-se da implementação da proposta no cenário Rollernet. Para este protocolo, a sobrecarga média caiu de cerca 22,7% para 22,89%, uma redução de menos de 1%.

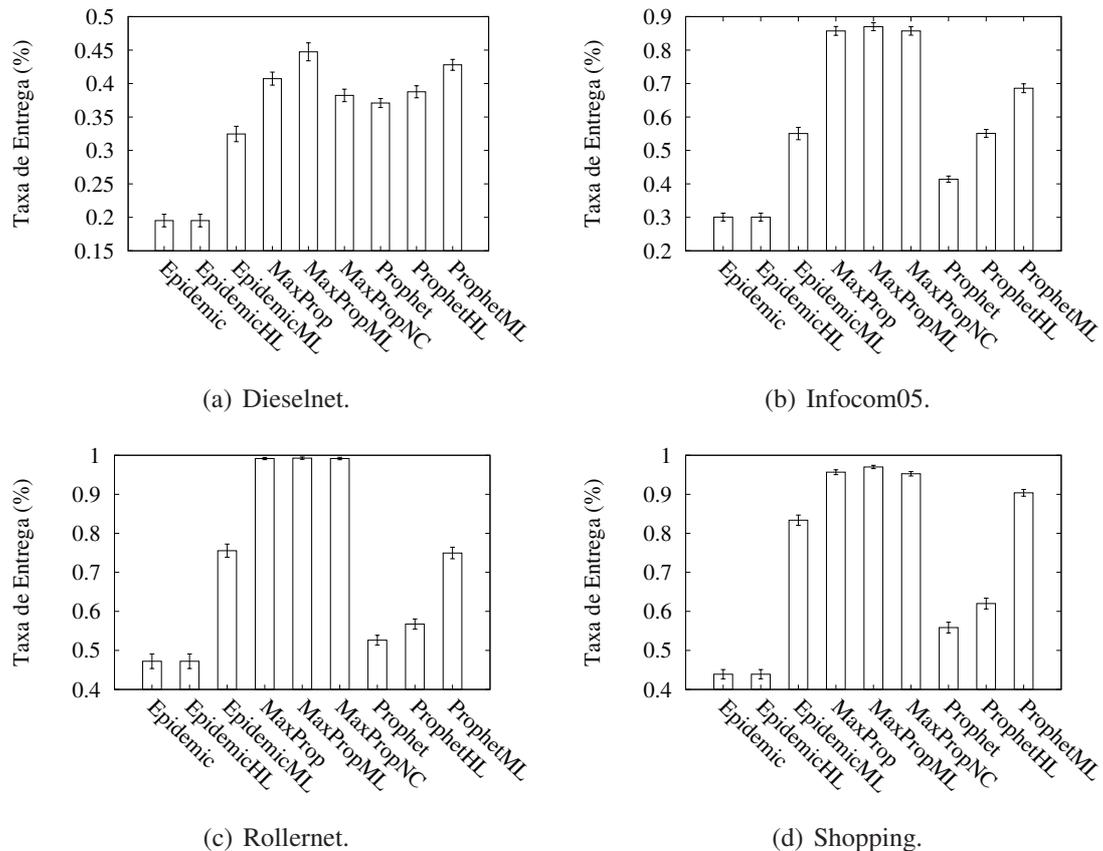
Por último, os resultados para o cenário Shopping são ilustrados na Figura 1(d). É possível observar que os protocolos Epidêmico, Prophet e MaxProp resultam, respectivamente, em uma sobrecarga de cerca de 930, 773 e 42 encaminhamentos. Por outro lado, quando utilizam o mecanismo de controle de congestionamento proposto neste trabalho, isto é, os protocolos EpidemicML, ProphetML e MaxPropML, resultam em uma sobrecarga de cerca de 16, 11 e 13 encaminhamentos, respectivamente. Ou seja, a utilização do mecanismo ML resulta em uma redução de cerca e 98% da sobrecarga de transmissão para o protocolo Epidêmico, 98,5% para o protocolo Prophet e 69% para o protocolo MaxProp.

Com relação ao mecanismo HL, embora a utilização deste mecanismo resulte em uma menor sobrecarga de encaminhamentos para os protocolos MaxProp e Prophet, nota-se que este mecanismo é superado pela proposta deste trabalho, que resulta em melhores resultados. Além disso, a proposta deste trabalho não incorre em sobrecarga de comunicação, diferentemente do mecanismo HL, visto que a lista de saltos é parte da mensagem e portanto, replicada junto com a mensagem.

## 5.2. Taxa de Entrega

A taxa de entrega é definida como o percentual de mensagens criadas que chegaram ao destino durante a simulação. Para as figuras aqui apresentadas, o eixo  $Y$  representa a taxa de entrega de mensagens obtida e os protocolos modificados e sem modificação são representados pelas barras. A Figura 2 apresenta as taxas de entrega para todos os cenários avaliados. Especificamente, a Figura 2(a) apresenta os resultados para o cenário Dieselnet, a Figura 2(b) apresenta os resultados para o cenário Infocom05, a Figura 2(c) apresenta os resultados para o cenário Rollernet e finalmente, a Figura 2(d) ilustra os resultados para o cenário Shoppingmall.

De modo geral, é possível observar que o mecanismo proposto, ML, aumenta a taxa de entrega alcançada para todos os protocolos avaliados. Isto ocorre porque o mecanismo proposto aumenta a eficiência da rede ao evitar o encaminhamento e a replicação de mensagens com baixa probabilidade de entrega. Além disso, é possível observar que



**Figura 2. Taxa de entrega para os cenários avaliados.**

o ganho de desempenho do mecanismo proposto é maior para os protocolos que atingem um nível de congestionamento maior, como é o caso dos protocolos Epidêmico e Prophet, assim como mencionado na Seção 4.1.

No cenário Dieselnet, a implementação do protocolo MaxProp que utiliza o mecanismo proposto para controle de congestionamento, denominada na figura como MaxPropML, obtém o melhor desempenho com relação a taxa de entrega entre todos os protocolos avaliados. Neste cenário, o MaxPropML alcançou uma taxa de entrega de cerca de 45%, seguido em desempenho pelo ProphetML, que atingiu cerca de 43% de taxa de entrega. Com relação ao protocolo MaxProp original, o protocolo MaxPropML obteve uma melhora de 5 pontos percentuais, ou neste caso, uma melhora de 12,5%.

Para o cenário Infocom05, o protocolo MaxPropML alcançou 87% de taxa de entrega, contra cerca de 86% do protocolo MaxProp original. Por sua vez, o protocolo ProphetML alcançou cerca de 69% contra cerca de 41% do protocolo Prophet original. Finalmente, o protocolo EpidemicML alcançou cerca de 55% de taxa de entrega contra uma taxa de entrega de 30% alcançada pelo protocolo original. Isto representa um acréscimo de 83% na taxa de entrega.

Por sua vez, o mecanismo HL mostra alguns resultados interessantes que serão comentados a seguir. A primeira observação a ser feita é que para o protocolo Epidêmico, o mecanismo HL não trouxe melhorias. Esta observação é válida para as demais

métricas de avaliação, abordadas nas Seções 5.3 e 5.1. Isto ocorre porque o protocolo Epidêmico não faz nenhum controle do número de réplicas ou mensagens encaminhadas na rede, desta forma, uma mensagem pode conter um número significativo de réplicas e cada uma destas réplicas pode ter trafegado por um caminho diferente. Sendo assim, é comum que mesmo que um determinado nó  $X$  já tenha recebido uma réplica da mensagem  $m_1$  e posteriormente descartado esta réplica, novas réplicas de  $m_1$  cheguem a  $X$  por caminhos distintos, que não passaram anteriormente por  $X$  e que por consequência, não se enquadram na premissa utilizada pelo mecanismo HL. Com relação ao protocolo Prophet, o mecanismo HL melhora o desempenho em todos os cenários avaliados. O caso do protocolo MaxProp é específico, pois este já implementa o mecanismo HL. Desta forma, para avaliar corretamente a melhoria de desempenho do mecanismo HL, decidiu-se implementar uma versão deste protocolo, chamada neste trabalho de MaxProNC. Desta forma, conseguiu-se observar que o mecanismo HL leva a ganhos de desempenho no protocolo MaxProp, principalmente em cenários menos conectados e por consequência mais congestionados, como exibido no cenário Dieselnet e apresentado na Figura 2(a).

### 5.3. Atraso de Entrega

O atraso de entrega ilustra a quantidade de tempo, em segundos, que decorre desde a criação de uma mensagem até o momento em que ela chega ao destinatário. Para os gráficos de atraso de entrega, apresentados na Figura 3, o eixo  $Y$  representa o atraso médio de entrega em segundos e os protocolos modificados e sem modificação são representados pelas barras. Especificamente, a Figura 3(a) apresenta os resultados para o cenário Dieselnet, a Figura 3(b) apresenta os resultados para o cenário Infocom05, a Figura 3(c) apresenta os resultados para o cenário Rollernet e finalmente, a Figura 3(d) ilustra os resultados para o cenário Shopping.

É possível observar que o mecanismo de controle de congestionamento proposto, ML, diminui o atraso de entrega na maioria dos cenários avaliados. Isto ocorre porque o mecanismo ML evita encaminhar e replicar mensagens com baixa probabilidade de entrega. Ao dar prioridade as mensagens com maior probabilidade de entrega, o mecanismo contribui para que estas mensagens sejam entregues mais rapidamente. As principais diferenças são apresentadas pelo protocolo Epidêmico. No cenário Dieselnet, apresentado na Figura 3(a), o atraso de entrega cai de cerca de 60.000 segundos, quando o protocolo original está sendo utilizado, para cerca de 40.000 segundos quando o mecanismo proposto é implementado no protocolo, uma redução de aproximadamente 5,5 horas ou cerca de 33%. Para o cenário Infocom05, apresentado na Figura 3(b), o atraso de entrega cai de cerca de 26.000 segundos para cerca de 21.000 segundos. Uma redução de cerca, de 1,4 horas, ou de cerca de 20%. Por sua vez, no cenário Rollernet, o atraso de entrega médio cai de cerca de 861 segundos para cerca de 777 segundos, quando o mecanismo de controle de congestionamento está sendo utilizado, o que corresponde a uma redução de cerca de 10% no atraso de entrega. Por último, no cenário Shopping, como exibido na Figura 3(d), o atraso de entrega médio obtido pelo protocolo Epidêmico cai de cerca de 6.026 segundos para 3.338 segundos, quando o mecanismo proposto é utilizado. Isto corresponde a uma redução de aproximadamente 45% do atraso de entrega médio.

Com relação ao protocolo Prophet, assim como no protocolo Epidêmico, a implementação da proposta reduziu o atraso de entrega em todos os 4 cenários avaliados. No entanto, algumas observações devem ser feitas a respeito do protocolo MaxProp. Neste

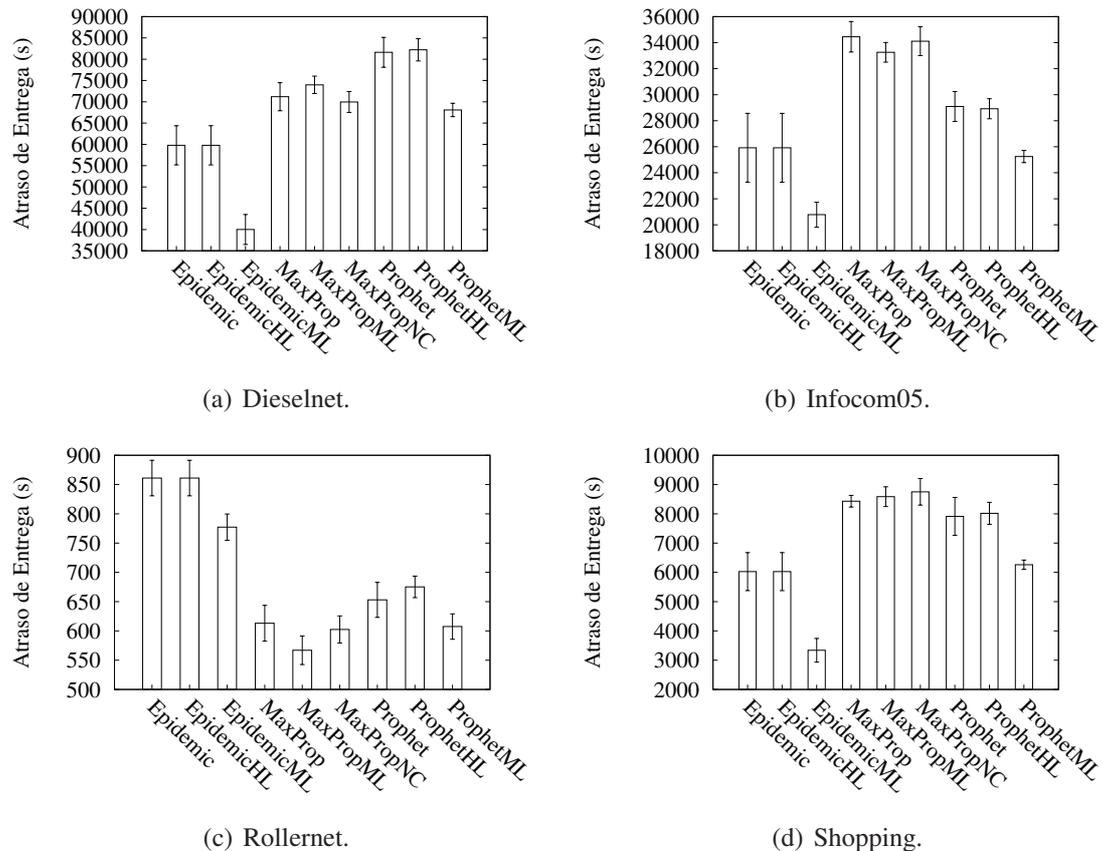


Figura 3. Atraso de entrega para os cenários avaliados.

protocolo, a utilização do mecanismo ML levou a um aumento no atraso de entrega em dois dos cenários avaliados, são eles, Dieselnet e Shopping, ilustrados nas Figuras 3(a) e 3(d), respectivamente. No entanto, os resultados são estatisticamente equivalentes, como é possível observar através do intervalo de confiança calculado.

Também é possível observar que o protocolo Epidêmico obteve o melhor desempenho com relação ao atraso de entrega para os cenários avaliados, exceto para o cenário Rollernet. Esta exceção ocorre devidos aos fatos que serão descritos a seguir. O conjunto de mobilidade Rollernet é notavelmente conectado, alterando períodos de alta conectividade com períodos menos conectados, fenômeno que os autores chamaram de “efeito acordeão” [Tournoux et al. 2009]. Esta alta conectividade ocasiona congestionamento, visto que possibilita que os nós repliquem exaustivamente as mensagens que possuem armazenadas em *buffer*. Observando-se que o protocolo Epidêmico não possui mecanismos para o controle de replicação, conclui-se que este cenário, combinando protocolo Epidêmico ao conjunto de mobilidade Rollernet, é o mais suscetível ao congestionamento dentre os cenários avaliados. Por outro lado, a despeito do aumento na taxa de entrega, o protocolo Epidêmico que implementa o mecanismo proposto atinge o menor atraso de entrega nos cenários Dieselnet, Infocom05 e Shopping. Isto demonstra que o mecanismo proposto oferece um equilíbrio entre o número de réplicas e a capacidade da rede, diminuindo o atraso de entrega enquanto aumenta ou mantém a taxa de entrega.

Por último, com relação ao mecanismo HL, é possível observar que este meca-

nismo levou a um pior desempenho com relação ao atraso de entrega para todos os cenários avaliados.

## 6. Conclusões e Trabalhos Futuros

Este trabalho propõe e avalia um novo mecanismo de controle de congestionamento para DTNs, nomeado ML. Esta proposta baseia-se na premissa de que o descarte das mensagens dos *buffers* dos nós deve ser permanente. Além disso, a proposta não incorre em sobrecarga de comunicação.

O mecanismo ML foi implementado em três protocolos bem conhecidos da literatura e seu desempenho foi comparado através de simulações com outro mecanismo utilizado para controle de congestionamento, o qual tem complexidade baixa, similar a proposta. Para as simulações, foram utilizados 4 conjuntos reais de mobilidade e para a avaliação três importantes métricas de desempenho foram utilizadas: taxa de entrega, atraso de entrega e sobrecarga. A avaliação demonstrou que o mecanismo é capaz de reduzir a sobrecarga de transmissão de mensagens em até 98,5%, com melhorias também na taxa de entrega de mensagens e no atraso de entrega de mensagens.

Para trabalhos futuros, pretende-se avaliar o mecanismo proposto com outros protocolos de roteamento, além de compará-lo a mecanismos de controle de congestionamento mais complexos existentes na literatura e que exigem um maior nível de conhecimento do contexto da rede. Além disso, pretende-se avaliar mais detalhadamente a premissa assumida por nossa proposta, ou seja, de que a remoção das mensagens dos *buffers* dos nós deve ser persistente. Adicionalmente, pretende-se avaliar a variação de outras variáveis da rede, como o tempo de vida das mensagens e o tamanho do *buffer* dos nós.

## Agradecimentos

Esse trabalho é apoiado pela TBE/ANEEL, CAPES, CNPq, FAPERJ, CTIC, PROPESP/IFRO e Proppi/UFF. Os autores agradecem pelos dados obtidos do arquivo CRAWLAD de Dartmouth College.

## Referências

- Burgess, J., Bissias, G. D., Corner, M. D., e Levine, B. N. (2007). Surviving attacks on disruption-tolerant networks without authentication. Em *Mobihoc*, páginas 61–70.
- Burgess, J., Gallagher, B., Jensen, D., e Levine, B. N. (2006). MaxProp: Routing for vehicle-based disruption-tolerant networks. Em *IEEE INFOCOM*, páginas 1–11.
- Cao, Y. e Sun, Z. (2013). Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys & Tutorials, IEEE*, 15(2):654–677.
- Cunha, D. d. O., Costa, L. H. M. K., e Duarte, O. C. M. B. (2005). *Analyzing the Energy Consumption of IEEE 802.11 Ad Hoc Networks*, páginas 473–484. Springer US.
- Galati, A., Djemame, K., e Greenhalgh, C. (2015). Analysis of human mobility patterns for opportunistic forwarding in shopping mall environments. *Social Network Analysis and Mining*, 5(1):1–14.
- Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., e Diot, C. (2005). Pocket switched networks and human mobility in conference environments. Em *ACM SIGCOMM WDTN Workshop*, páginas 244–251.

- Liu, Y., Wang, K., Guo, H., Lu, Q., e Sun, Y. (2016). Social-aware computing based congestion control in delay tolerant networks. *Mobile Networks and Applications*, páginas 1–12.
- Naves, J. F. e Moraes, I. M. (2014). Uma avaliação do ataque de falsificação de reconhecimentos positivos em redes tolerantes a atrasos e desconexões. Em *Anais do 32º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 105–118.
- Naves, J. F., Moraes, I. M., e Albuquerque, C. (2012a). LPS and LRF: Efficient buffer management policies for delay and disruption tolerant networks. Em *IEEE 37th Conference on Local Computer Networks (LCN)*, páginas 368–375.
- Naves, J. F., Moraes, I. M., e de Albuquerque, C. V. N. (2012b). LPS e LRF: Políticas de gerenciamento de buffer eficientes para redes tolerantes a atrasos e desconexões. Em *Anais do 30º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, páginas 233–246.
- Oliveira, C. T., Moreira, M. D. D., Rubinstein, M. G., Costa, L. H. M. K., e B. Duarte, O. C. M. (2007). Redes tolerantes a atrasos e desconexões. Em *Minicursos do Simpósio Brasileiro de Redes de Computadores*.
- Scott, K. e Burleigh, S. (2007). Bundle Protocol Specification. RFC 5050.
- Silva, A. P., Burleigh, S., Hirata, C. M., e Obraczka, K. (2015a). DTN congestion control unplugged: A comprehensive performance study. Em *Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks*, páginas 43–48.
- Silva, A. P., Burleigh, S., Hirata, C. M., e Obraczka, K. (2015b). A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks*, 25, Part B(0):480 – 494.
- Silva, A. P., Burleigh, S., Hirata, C. M., e Obraczka, K. (2016a). Congestion control in disruption-tolerant networks: A comparative study for interplanetary and terrestrial networking applications. *Ad Hoc Networks*, 44:1–18.
- Silva, A. P., Obraczka, K., Burleigh, S., e Hirata, C. M. (2016b). Smart congestion control for delay- and disruption tolerant networks. Em *13th Annual IEEE International Conference on Sensing, Communication, and Networking*, páginas 1–9.
- Soelistijanto, B. e Howarth, M. (2014). Transfer reliability and congestion control strategies in opportunistic networks: a survey. *IEEE Communications Surveys and Tutorials*, 16(1):538 – 555.
- Tournoux, P.-U., Leguay, J., Benbadis, F., Conan, V., de Amorim, M. D., e Whitbeck, J. (2009). The accordion phenomenon: Analysis, characterization, and impact on DTN routing. Em *IEEE INFOCOM*, páginas 1116–1124.
- Wei, K., Guo, S., e Xu, K. (2014). CACC: A context-aware congestion control approach in smartphone networks. *Communications Magazine, IEEE*, 52(6):42–48.
- Zhang, X., Kurose, J., Levine, B. N., Towsley, D., e Zhang, H. (2007). Study of a bus-based disruption-tolerant network: Mobility modeling and impact on routing. Em *ACM MobiCom*, páginas 195–206.