

# Um Mecanismo de Remoção de Mensagens Obsoletas para as Redes Tolerantes a Atrasos e Interrupções

Elenilson da N. Gomes<sup>1</sup>, Rafael M. S. Fernandes<sup>2</sup>,  
Carlos Alberto V. Campos<sup>1</sup>, Aline C. Viana<sup>3</sup>

<sup>1</sup>Departamento de Informática Aplicada (DIA)  
Universidade Federal do Estado do Rio de Janeiro (UNIRIO) – Rio de Janeiro, Brasil

<sup>2</sup>Faculdade SENAC RIO  
Rio de Janeiro, Brasil

<sup>3</sup>Institut National de Recherche en Informatique et en Automatique (INRIA)  
Orsay, France

{elenilson.gomes, beto}@uniriotec.br, rafaelmsf@gmail.com, aline.viana@inria.fr

**Abstract.** *Depending on the routing protocol used in Delay Tolerant Networks (DTNs), many messages are replicated throughout the network by intermediate nodes. This is performed with the goal of increasing the likelihood of message delivery. However, even after the arrival of messages at their destination, redundant copies of delivered messages continue to circulate in the network and to occupy relevant space in the buffer of network nodes. This affects the delivery of non-delivered messages. Within this context, this paper proposes a mechanism for removing obsolete messages in DTNs. The proposed mechanism was implemented in the simulator ONE and an evaluation was performed to assess its impact on the performance of DTN routing protocols, such as: Epidemic, PROPHET, and Spray and Wait. Results show the good performance of our removal mechanism when applied together with the three evaluated protocols.*

**Resumo.** *Dependendo do protocolo de roteamento utilizado nas redes tolerantes a atrasos e interrupções, muitas mensagens são replicadas ao longo da rede pelos nós intermediários. Este fato visa aumentar a probabilidade de entrega das mensagens. Entretanto, depois que uma mensagem chega no seu destino, as suas réplicas continuam ocupando espaço no buffer dos nós e sendo repassadas na rede, o que pode prejudicar o encaminhamento ou a entrega dessas mensagens. Dentro desse contexto, neste artigo é proposto um mecanismo de remoção de mensagens obsoletas para as redes tolerantes a atrasos e interrupções. O mecanismo proposto foi implementado no simulador ONE e uma avaliação de desempenho foi realizada para verificar seu impacto no desempenho dos protocolos Epidêmico, PROPHET e Spray and Wait.*

## 1. Introdução

Existem situações de uso de uma rede sem fio, onde, a comunicação entre os dispositivos (nós) é intermitente podendo não existir uma conexão fim-a-fim entre a origem e o destino. Para estas situações surgem as redes tolerantes a atrasos e interrupções (*Delay*

*Tolerant Networks – DTN*), pois, tem como objetivo manter a comunicação entre os dispositivos mesmo sob grandes atrasos de transmissão, altas taxas de erro e sob interrupções na conectividade da rede. Além disso, o caminho entre origem e destino pode não estar acessível o tempo todo, ou talvez nunca chegue a estar. Devido a estas características o mecanismo usado é o *store-carry-and-forward* [Vahdat and Becker 2000], que faz com que as mensagens sejam armazenadas no *buffer* para somente serem enviadas no contato com um outro nó. Nas DTNs os nós normalmente tem pouco espaço de armazenamento e baixa capacidade de processamento.

O roteamento tradicional não pode ser utilizado e no seu lugar, na maioria das vezes, usa-se os protocolos de roteamento replicadores de mensagens. Esses protocolos geram várias cópias da mensagem original com o intuito de aumentar a probabilidade de entrega e, até mesmo, diminuir o atraso na entrega de mensagens entre a origem e o destino. Como exemplo de um protocolo replicador tem-se o protocolo de roteamento Epidêmico [Vahdat and Becker 2000]. Outros protocolos de roteamento probabilísticos e protocolos replicadores controlados foram proposto. Mesmo assim, mensagens podem ter sido replicadas na rede e quando a mensagem original ou uma de suas réplicas chegar ao destino, as outras cópias se tornarão obsoletas, mas continuarão no *buffer* dos nós ocupando espaço desnecessariamente. Assim quando uma mensagem chegar ao seu destino, a rede pode usar esta informação para comunicar aos nós que esta mensagem foi entregue e que todos os nós que tiverem uma cópia da mesma devem removê-las, pois estas cópias não são mais necessárias e estão obsoletas nos *buffers*.

Dentro desse contexto, neste artigo é proposto um mecanismo que usa a informação de confirmação de recebimento de uma mensagem pelo destino para a remoção das possíveis cópias desta mensagem que estão, de maneira obsoleta, armazenadas no *buffer* dos nós. Estas informações são armazenadas numa lista que é trocada a cada contato efetuado entre os nós, possibilitando uma atualização de ambos, similar ao modelo matemático Vaccine [Haas 2006]. Nele, podemos observar que utilizando protocolos de roteamento que disseminam mensagens pela rede é possível melhorar o uso do *buffer* em relação aos que não disseminam mensagens, logo, necessita-se de mais 14% de espaço em *buffer*. A principal contribuição do nosso artigo é o uso do mecanismo de remoção de mensagens obsoletas independente do protocolo de roteamento utilizado. Sendo assim, este mecanismo foi implementado no simulador The One levando em consideração a característica de independência do funcionamento do roteamento usado na DTN.

Este artigo está organizado da seguinte maneira: na Seção 2, os trabalhos relacionados são apresentados e discutidos; na Seção 3, o mecanismo de remoção de mensagens obsoletas é proposto e um detalhamento sobre seu funcionamento é mostrado; uma avaliação do mecanismo proposto é apresentada na Seção 4, onde os resultados obtidos são mostrados e discutidos. Por fim, a Seção 5 contém a conclusão e os trabalhos futuros.

## **2. Trabalhos Relacionados**

Em [Haas 2006] é apresentado o modelo SWIM(*Shared Wireless Infostation Model*) que apresenta uma avaliação de um novo paradigma da comunicação, que ao contrário de conexões estabelecidas fim-a-fim, como em uma rede de celular, o SWIM

cria o que se refere a comunicações virtuais em uma rede DTN. O SWIM explora a mobilidade do sistema, produzindo os enlaces virtuais para *offload* de dados quando um nó está perto de um posto de coleta. As relações virtuais são realizadas por ligações físicas dos nós que quando entram em contato, propagam as informações para os destinos através da replicação das informações. Primeiramente, foi apresentado um estudo sobre o tempo de vida dos pacotes (*Time To Live - TTL*), Paralelamente, foi examinado o percentual de uso do *buffer* no armazenamento no sistema, desenvolvendo-se modelos matemáticos para cinco diferentes métodos de descartar pacotes, JUST-TTL, FULL-ERASE, IMMUNE, IMMUNE-TX e VACCINE.

Em [Mundur and Seligman 2008], foi modificado o protocolo de roteamento Epidêmico. O foco foi usar a informação das mensagens entregues, através de uma lista, visando impedir trocas futuras entre elas. Através desta técnica mostra-se a melhoria na taxa de entrega das mensagens, por consequência da melhoria do uso do *buffer*.

A preocupação em remoção de mensagens é abordada em [Rashid and Ayub 2010], onde é proposta uma política de gerência de *buffer* em DTN. Quando o *buffer* do nó está cheio e o mesmo precisa de espaço para armazenar uma nova mensagem, a maior mensagem no *buffer* será descartada. Esta estratégia foi chamada de *droplargest (DLA)*. Entretanto, o mecanismo prejudica o funcionamento das aplicações que geram mensagens grandes.

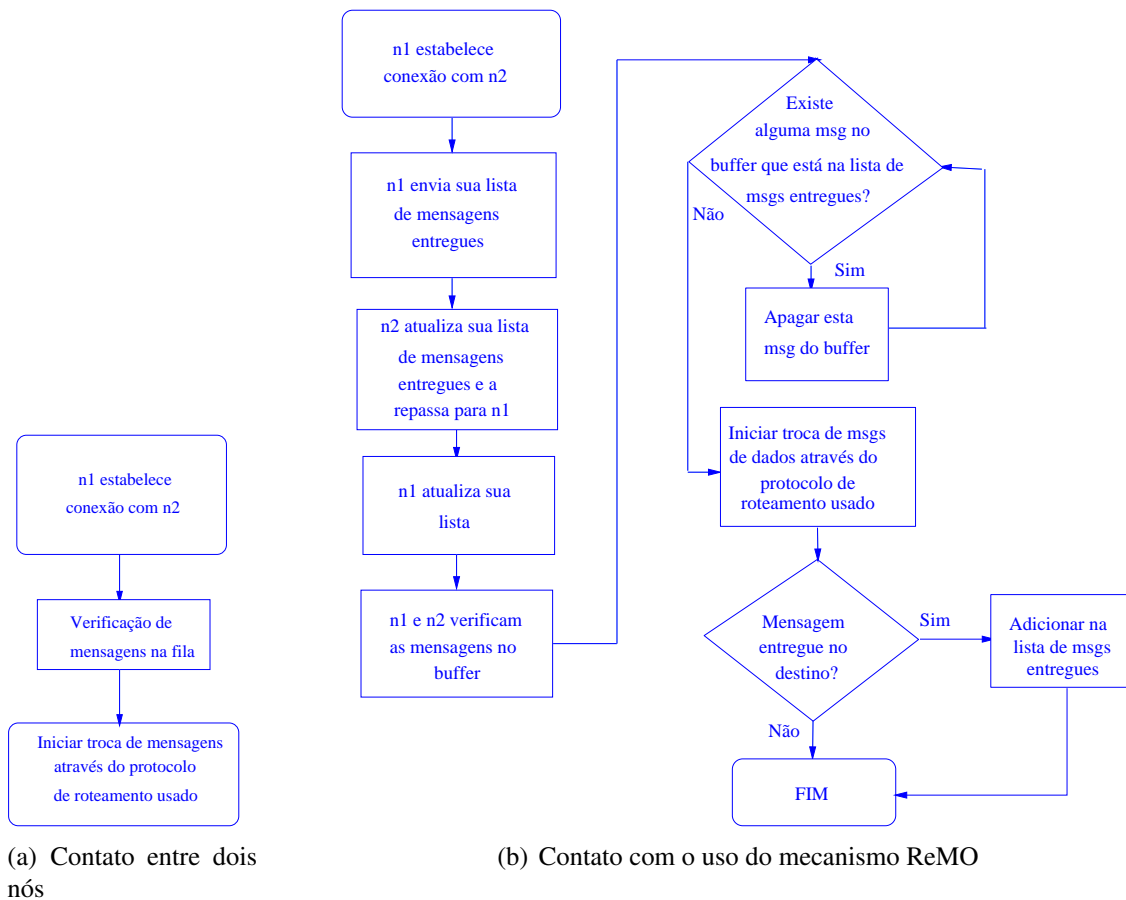
Em [Yuen 2010], é mostrado que numa rede dependente da mobilidade dos nós, o tempo de entrega pode variar de minutos a horas. Assim, para agilizar a entrega das mensagens, as mesmas são replicadas durante os encontros dos nós, mesmo quando o nó receptor não for o destino. Com réplicas de mensagens redundantes, o tempo de entrega pode ser significativamente reduzido. Para limitar o número de réplicas de mensagens, algumas destas replicas podem ser eliminadas através do uso do TTL. Assim, foi definido um regime baseado no tempo e outro baseado no número de saltos para o TTL. O objetivo é analisar os dois regimes em termos métricas de custo de energia e tempo de armazenamento, através de simulações. Considera-se também o desempenho quando um canal de retorno com baixa taxa de dados está disponível. Quando uma mensagem é entregue com êxito, o sistema notifica todos os nós para apagarem suas réplicas desta mensagem através de um canal de retorno, como o canal de paginação de uma rede de telefonia celular.

Um mecanismo de remoção de mensagem também é o foco em [Kaveevivitchai et al. 2010], onde é apresentada a eficiência no mecanismo para eliminação de mensagem que pode ser aplicado independente do protocolo de roteamento. O projeto foi separado em duas partes: (i) Métodos para a distribuição de mensagens de confirmação (ACK), (ii) A utilização de nós especiais denominados *helper* para retransmitir mensagens de ACK.

### **3. Mecanismo de Remoção de Mensagem Obsoletas - ReMO**

O mecanismo de remoção visa informar aos nós intermediários que uma mensagem foi entregue ao nó de destino, possibilitando a remoção de suas réplicas da rede.

Conforme o funcionamento de alguns protocolos de roteamento, temos como premissa que quando um nó entra em contato com outro, é feita uma breve verificação das mensagens em sua fila. Se possuem mensagens diferentes então irão trocá-las, conforme descrito na Figura 1a.



**Figura 1. Fluxogramas da comunicação entre dois nós durante um contato DTN.**

### 3.1. O algoritmo do mecanismo ReMO

As DTNs trocam mensagens quando ocorrer um contato entre dois ou mais nós. Um contato pode ser definido da seguinte maneira. Sejam  $n_1$  e  $n_2$  nós e  $p_{n_1}(t)$  e  $p_{n_2}(t)$  suas respectivas posições no instante  $t$ . Sejam ainda  $A_{n_1}$  e  $A_{n_2}$  suas áreas de cobertura. Se no instante  $t$ ,  $p_{n_2}(t) \subset A_{n_1}$  e  $p_{n_1}(t) \subset A_{n_2}$  diz-se que  $n_1$  e  $n_2$  estão em contato no instante  $t$ .

O ReMO cria uma lista de mensagens entregues para cada nó, onde são armazenadas as informações das mensagens que já foram entregues no destino. Esta lista é trocada a cada contato e após a sua troca é efetuada a remoção de mensagens obsoletas do *buffer*. Após esta fase os nós transmitirão as mensagens que estejam nos seus respectivos *buffers*, seguindo o protocolo de roteamento que os mesmos estão executando.

É importante observar que o ReMO independe do protocolo de roteamento utilizado pelos nós. O algoritmo funciona de maneira distribuída, onde a informação de uma mensagem entregue vai ser repassada, salto-a-salto, para todos os nós. A Figura 1b apresenta o fluxograma de funcionamento do mecanismo ReMO no instante de contato entre os nós  $n_1$  e  $n_2$ .

### 3.2. Detalhes da implementação do mecanismo ReMO no simulador ONE

O ReMO foi implementado no simulador The ONE [Keränen et al. 2009] que é um dos mais utilizados em DTN. O The ONE é um simulador de eventos discretos desenvolvido na linguagem de programação JAVA e de código aberto.

Para a troca de informações entre os nós no mecanismo ReMO referentes as mensagens que já chegaram nos seus respectivos destinos foi adicionada uma lista de mensagens entregues, esta lista foi configurada para armazenar (N) registros e ocupa (N) bytes do *buffer*, cada registro contém o identificador único da mensagem já entregue em cada nó. Quando uma mensagem é entregue no destino, este nó de (destino) insere a identificação desta mensagem na sua lista de mensagens entregues, assim como o identificador do nó que entregou a mensagem. Com isso, as mensagens já entregues estarão na lista de mensagens entregues de ambos os nós. Adicionalmente, a informação de mensagens entregues desse nó poderá ser repassada para os outros nós da rede.

Em cada contato entre nós na rede, o primeiro passo é a comparação e troca das informações das listas de mensagens entregues entre os nós que estão em contato. Após essa troca, os dois nós que estão em contato possuirão a mesma lista de mensagens entregues (ver Figura 1b). No próximo passo, os nós verificam se existe alguma mensagem em seu *buffer* que está relacionada na sua lista de mensagens entregues. Se houver, significa que o nó possui mensagens obsoletas e então a mensagem será removida, liberando espaço de armazenamento para outras mensagens. Após a remoção das mensagens obsoletas, os nós trocarão as mensagens conforme o protocolo de roteamento que esteja sendo executado na rede. Se alguma mensagem de dados for entregue ao nó destino na etapa de troca de mensagens de dados, o nó destino e o nó que está entregando a mensagem irão incluir na lista de mensagens entregues. Além disso, nos próximos contatos destes nós a informação será propagada conforme descrito anteriormente.

Na seção seguinte será descrita uma avaliação do mecanismo ReMO contendo os resultados obtidos por simulação.

#### **4. Avaliação de desempenho de alguns protocolos de roteamento sob a influência do mecanismo ReMO**

Espera-se que, com a remoção de mensagens obsoletas das DTNs será possível utilizar melhor o *buffer* dos nós, o que pode permitir um melhor desempenho da rede e dos protocolos de roteamento que usam a técnica de replicação de mensagens. Para a avaliação do mecanismo ReMO, foi utilizado o simulador The ONE. Assim, no restante desta seção serão apresentados: protocolos de roteamento usados, descrição dos parâmetros de simulação, métricas de desempenho e, finalmente, os resultados obtidos.

##### **4.1. Protocolos de Roteamento**

Em [Vahdat and Becker 2000] foi definido um protocolo de roteamento chamado Epidêmico, com o objetivo de disseminar mensagens em DTN maximizando a taxa de entrega e minimizando o tempo de transmissão de uma mensagem da origem até o seu respectivo destino. O funcionamento desse protocolo é semelhante a uma doença epidêmica, ou seja, enquanto houver espaço em *buffer* as mensagens são repassadas a cada encontro dos nós. Cada nodo da rede possui uma lista com as mensagens que o

mesmo armazena. Essa lista é trocada entre vizinhos que estão no mesmo alcance de transmissão e serve para que um nó determine quais as mensagens existentes no *buffer* do vizinho ele ainda não possui, para então solicitá-las. Esse processo de troca de mensagens se repete sempre que um nó entra em contato com outro. Embora o roteamento Epidêmico possa ter uma alta taxa de entrega e baixo atraso, não é raro ver a capacidade de armazenamento dos nós ser ultrapassada ocasionando assim o descarte das mensagens e a consequente queda da taxa de entrega.

O protocolo *Probabilistic Routing Protocol using History of Encounters and Transitivity* (PROPHET) [Lindgren and Doria 2003] usa uma métrica que indica a probabilidade de um nó entregar uma mensagem a um destinatário. Essa métrica é chamada de previsibilidade de entrega. Quando dois nós se encontram, uma mensagem é enviada para o outro nó se a previsibilidade de entrega ao destino for mais alta no outro nó. Contudo, como o nó que repassou a mensagem pode encontrar um nó melhor ou o próprio destino final no futuro, a mensagem repassada não é removida do nó, mas fica armazenada em *buffer* (desde que haja espaço disponível). Além disso, uma função de transitividade também é usada.

Em [Spyropoulos and Psounis 2008] outro protocolo de roteamento é o *Spray and Wait*, onde o nó origem da mensagem inicialmente começa com  $L$  cópias. Qualquer nó que possui  $n > 1$  cópias da mensagem e encontra outro nó sem cópias dessa mensagem, repassa  $\lfloor n/2 \rfloor$  de suas cópias e mantém  $\lceil n/2 \rceil$ ; até que haja somente uma cópia. Então, ele troca para o roteamento de transmissão direta. Segundo [Spyropoulos 2007], o roteamento *Spray and Wait* não tem um bom desempenho para cenários onde a mobilidade é pequena e localizada. Uma definição importante a ser feita nesse tipo de roteamento é a quantidade de cópias que pode ser gerada para cada mensagem criada.

## 4.2. Descrição dos parâmetros da simulação

Com o objetivo de avaliar uma DTN em um cenário onde pessoas estivessem usando dispositivos móveis e se locomovendo a pé em uma cidade, foi utilizado o mapa da cidade de Helsink na Finlândia (mapa disponível no simulador The ONE) para representar esse cenário urbano. Além disso, os valores dos parâmetros dos componentes da simulação referentes: ao cenário, ao movimento dos nós (representando pessoas caminhando), ao raio de alcance dos nós, à geração e características das mensagens, às características específicas de cada protocolo de roteamento avaliado, ao tempo da simulação e ao número de rodadas para cada cenário de simulação avaliado, são mostrados na Tabela 1.

## 4.3. Métricas de Desempenho Utilizadas

Neste trabalho serão utilizadas três métricas de desempenho (probabilidade de entrega, atraso médio e sobrecarga de mensagens) para avaliar a influência do mecanismo proposto no funcionamento da rede sob diferentes protocolos de roteamento e a métrica ocupação do *buffer* para investigar a eficiência do mecanismo na remoção de mensagens obsoletas. Estas métricas serão definidas a seguir.

**Probabilidade de entrega de mensagens:** número de mensagens entregues dividido pelo número de mensagens criadas;

**Atraso médio:** tempo médio que uma mensagem leva para ser entregue, desde quando é gerada até o seu destino;

Parâmetro	Valor
Tempo de simulação	12 horas
Velocidade de transmissão	250kbps
Raio de alcance	100 metros
Número de nós	100
Tamanho de <i>buffer</i>	10MB, 50MB, 100MB e 150MB
Modelo de mobilidade	<i>Shortest Path Map Based Movement</i>
Velocidade dos nós	entre 0,5 e 1,5 m/s
Geração das mensagens	a cada 20 segundos
Tamanho das mensagens	100KB e 1MB
Quantidade de rodadas de simulação	10
Tamanho da área simulada	4500 x 3400 metros <sup>2</sup>
Intervalo de tempo para encaminhamento de mensagens no PROPHET	30 segundos
Número de cópias das mensagens no <i>Spray and Wait</i>	6 cópias

**Tabela 1. Valores dos parâmetros componentes/protocolos da simulação**

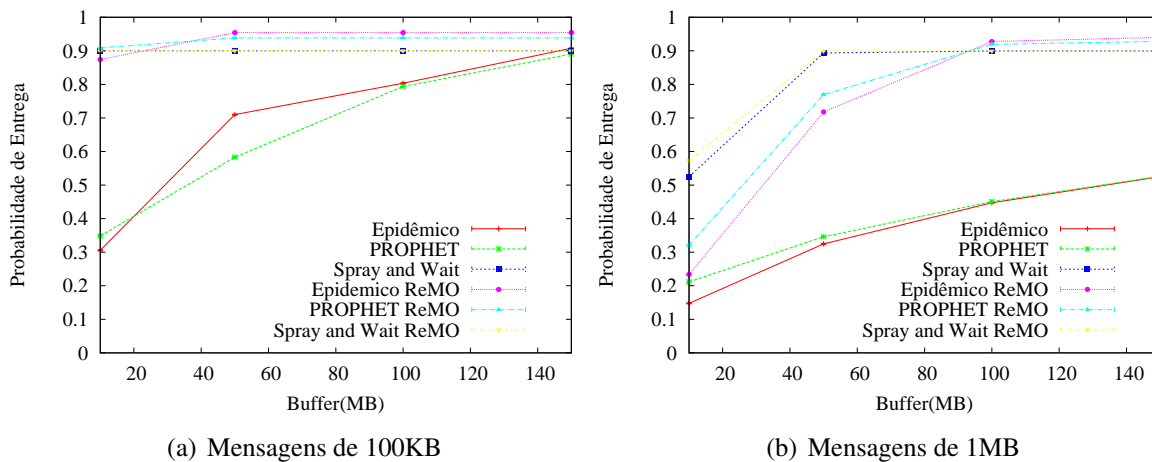
**Sobrecarga de mensagens:** quantidade de mensagens repassadas dividido pela quantidade de mensagens entregues;

**Ocupação do *buffer*:** quantidade de dados armazenados dividido pela capacidade total do *buffer*, em um determinado instante de tempo.

#### 4.4. Resultados Obtidos

##### 4.4.1. Probabilidade de Entrega

As Figuras 2a e 2b ilustram os resultados das simulações para a métrica probabilidade de entrega.



**Figura 2. Probabilidade de entrega das mensagens sob a influência dos protocolos Epidêmico, PROPHET e *Spray and Wait* variando o tamanho das mensagens.**

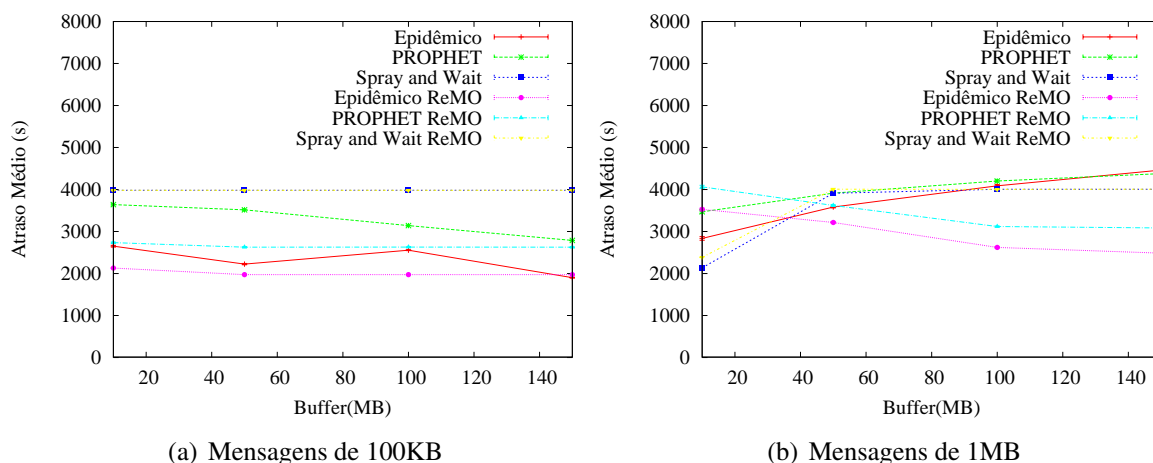
Na Figura 2a é analisada a métrica probabilidade de entrega, variando o *buffer*,

com o tamanho das mensagens definido em 100KB. É possível observar que a probabilidade de entrega para os protocolos Epidêmico e PROPHET é maior com o uso do mecanismo ReMO. Isso ocorre devido a menor taxa de ocupação do *buffer* dos nós, proporcionando maiores possibilidades de troca de mensagens. Já para o protocolo *Spray and Wait*, a probabilidade de entrega foi igual tanto com o uso do ReMO, quando sem o mesmo. Isso ocorre devido a baixa taxa de utilização do *buffer* dos nós, que é uma característica do protocolo *Spray and Wait*, fazendo com que o ReMO não seja tão efetivo. Uma análise mais detalhada sobre a ocupação do *buffer* será apresentada na Seção 4.4.4.

Já a Figura 2b são utilizadas mensagens de tamanho 1MB, com o objetivo de saturar o *buffer* dos nós e verificar a influência do ReMO na capacidade de entrega das mensagens. Mais uma vez os protocolos Epidêmico e PROPHET obtiveram uma maior probabilidade de entrega com o uso do ReMO. Já o *Spray and Wait* teve uma pequena diferença quando analisado o *buffer* de 10MB e para *buffers* maiores, obteve um comportamento similar ao da Figura 2b. Essa diferença no *buffer* de 10MB é devido ao pequeno tamanho do *buffer* em relação as mensagens de 1MB, entretanto, quando o tamanho do *buffer* aumenta isso deixar de ser importante.

#### 4.4.2. Atraso Médio

Os resultados para o atraso médio podem ser observados nas Figuras 3a e 3b. Novamente a primeira figura ilustra o gráfico da simulação para as mensagens de tamanho 100KB e a segunda para mensagens de tamanho 1MB.



**Figura 3. Análise do atraso médio da entrega das mensagens sob a influência dos protocolos Epidêmico, PROPHET e *Spray and Wait* variando o tamanho das mensagens.**

Na Figura 3a, referente à mensagens de 100KB, pode-se observar que os protocolos Epidêmico e PROPHET provocaram um menor atraso na entrega das mensagens utilizando o ReMO. Com a remoção de algumas mensagens, os nós obtiveram mais espaço nos *buffers*, possibilitando mais trocas de mensagens e, conseqüentemente, fazendo algumas mensagens chegarem mais rapidamente aos seus respectivos destinos. Para o proto-

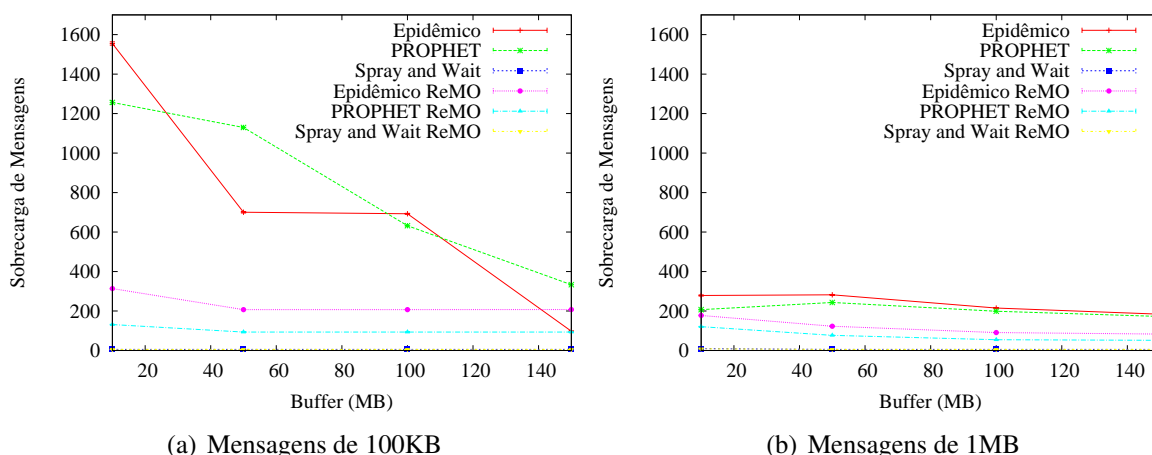


colo *Spray and Wait*, foi observado que não houveram alterações no atraso médio, mais uma vez, devido à baixa ocupação do *buffer* dos nós provocada pelo número restrito de cópias que o protocolo gera.

Já a Figura 3b apresenta a mesma métrica para mensagens de tamanho 1MB. O protocolo *Spray and Wait* continuou com o mesmo comportamento, mantendo o atraso médio igual. Já os protocolos Epidêmico e PROPHET tiveram uma melhora no atraso médio, como ocorreu nos resultados da Figura 3a.

#### 4.4.3. Sobrecarga de Mensagens

Com uma menor ocupação do *buffer* utilizando-se o ReMO, mais mensagens serão trocadas e mais mensagens poderão ser entregues. Essas informações estão diretamente ligadas com a métrica sobrecarga.



**Figura 4. Sobrecarga de mensagens causada pelos protocolos Epidêmico, PROPHET e *Spray and Wait* variando o tamanho das mensagens.**

A Figura 4a apresenta o resultado para mensagens de tamanho 100KB. É possível observar que a sobrecarga das mensagens quando usado o ReMO é bem menor. Um dado interessante é o protocolo *Spray and Wait*, pois como ele só gera  $L$  cópias das mensagens (definiu-se  $L=6$ ), a sobrecarga ficará próxima deste valor.

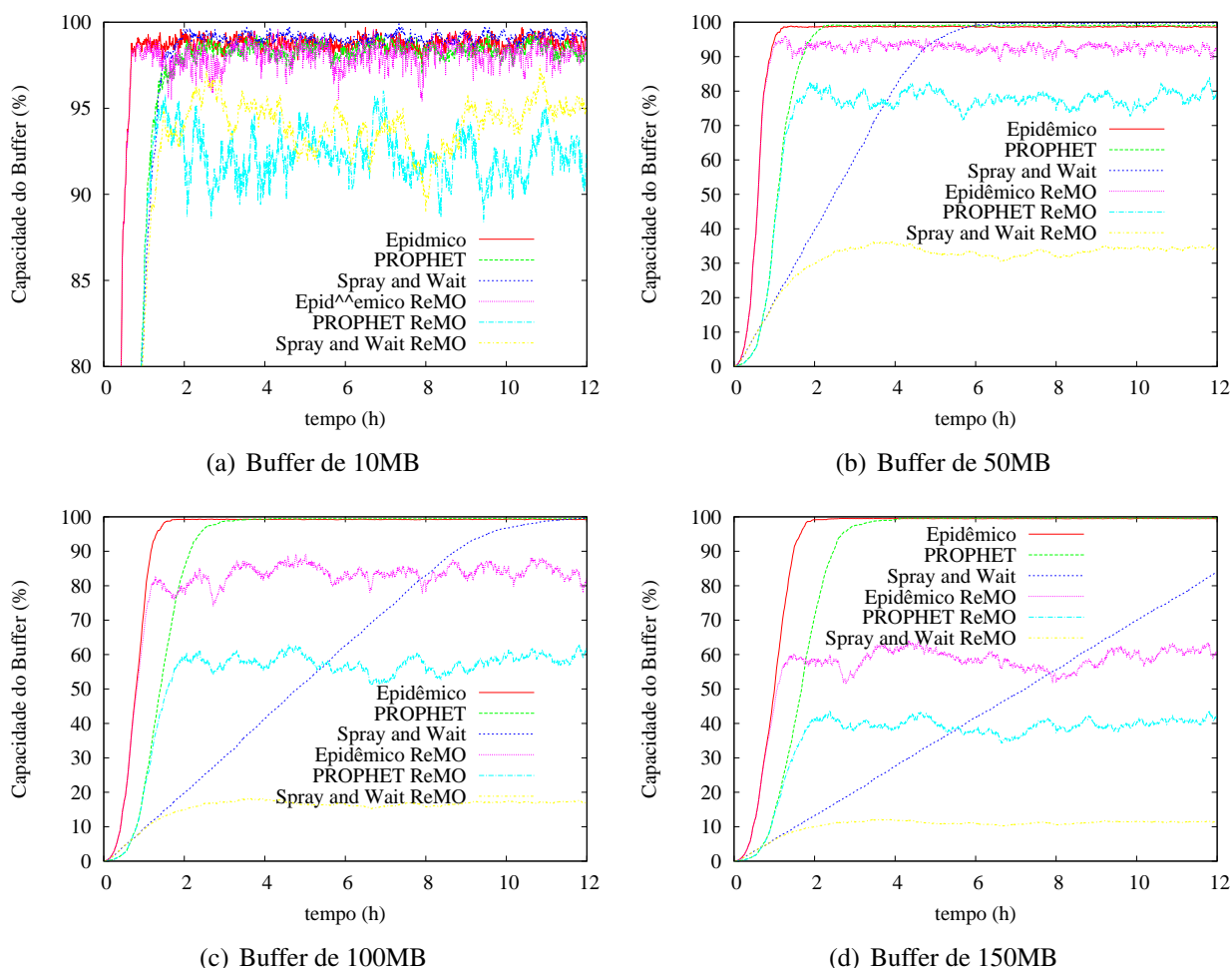
Na Figura 4b, com mensagens de tamanho 1MB, tem-se um comportamento similar ao da Figura 4a.

#### 4.4.4. Análise da influência do mecanismo ReMO no percentual de ocupação do *buffer*

Para investigar, em profundidade, a eficiência do mecanismo ReMO será analisada a taxa de ocupação do *buffer* com diferentes cargas de tráfego e de tamanho de *buffer*. Na Figura 5 são usadas mensagens com tamanho de 1MB com o objetivo de avaliar uma alta carga

de dados e na Figura 6 são utilizadas mensagens de 100KB para avaliar uma baixa carga de dados na rede.

Na Figura 5a cada nó possui um *buffer* de 10 MB. Nesta situação, o percentual de ocupação do *buffer* é bastante alto devido a capacidade de armazenamento de cada nó ser pequena e o grande tamanho das mensagens. O protocolo Epidêmico, sem o uso do ReMO, apresenta uma ocupação de 99% do *buffer* dos nós após, aproximadamente, uma hora de simulação. Esta é uma característica esperada deste protocolo, já que o mesmo tem por princípio de funcionamento a replicação das mensagens em cada contato dos nós, ocasionando muitas cópias das mensagens na rede. Quando o ReMO é aplicado, o percentual de ocupação médio é de 97%, possibilitando um maior repasse de mensagens. Essa pequena diferença na ocupação média do *buffer* dos nós permitiu uma melhoria de 9,5% na probabilidade de entrega das mensagens, conforme visto na Figura 2b.



**Figura 5. Influência do mecanismo ReMO na ocupação do *buffer* para cada protocolo de roteamento com mensagens de 1MB de tamanho.**

Os resultados para o protocolo de roteamento PROPHET, que também é um protocolo replicador como o Epidêmico, também apresentaram um melhor resultado no percentual de ocupação do *buffer* com o uso do ReMO, conforme Figura 5a. A porcentagem média de ocupação diminuiu 4% quando se usa o ReMO, em comparação com a simulação sem o uso do ReMO. Essa diferença ocasionou uma melhora de 10% na en-

trega das mensagens utilizando o protocolo PROPHET. Já para o protocolo de roteamento *Spray and Wait*, que também é replicador, porém com uma quantidade limitada de réplicas na rede, apresentou uma melhora de 5% na taxa de ocupação do *buffer*. Este aumento influenciou positivamente o roteamento dos nós, como pode ser observado na métrica de probabilidade de entrega (ver Figura 2b), onde a probabilidade de entrega aumentou 5% quando se usa o ReMO em relação ao resultado sem o mesmo.

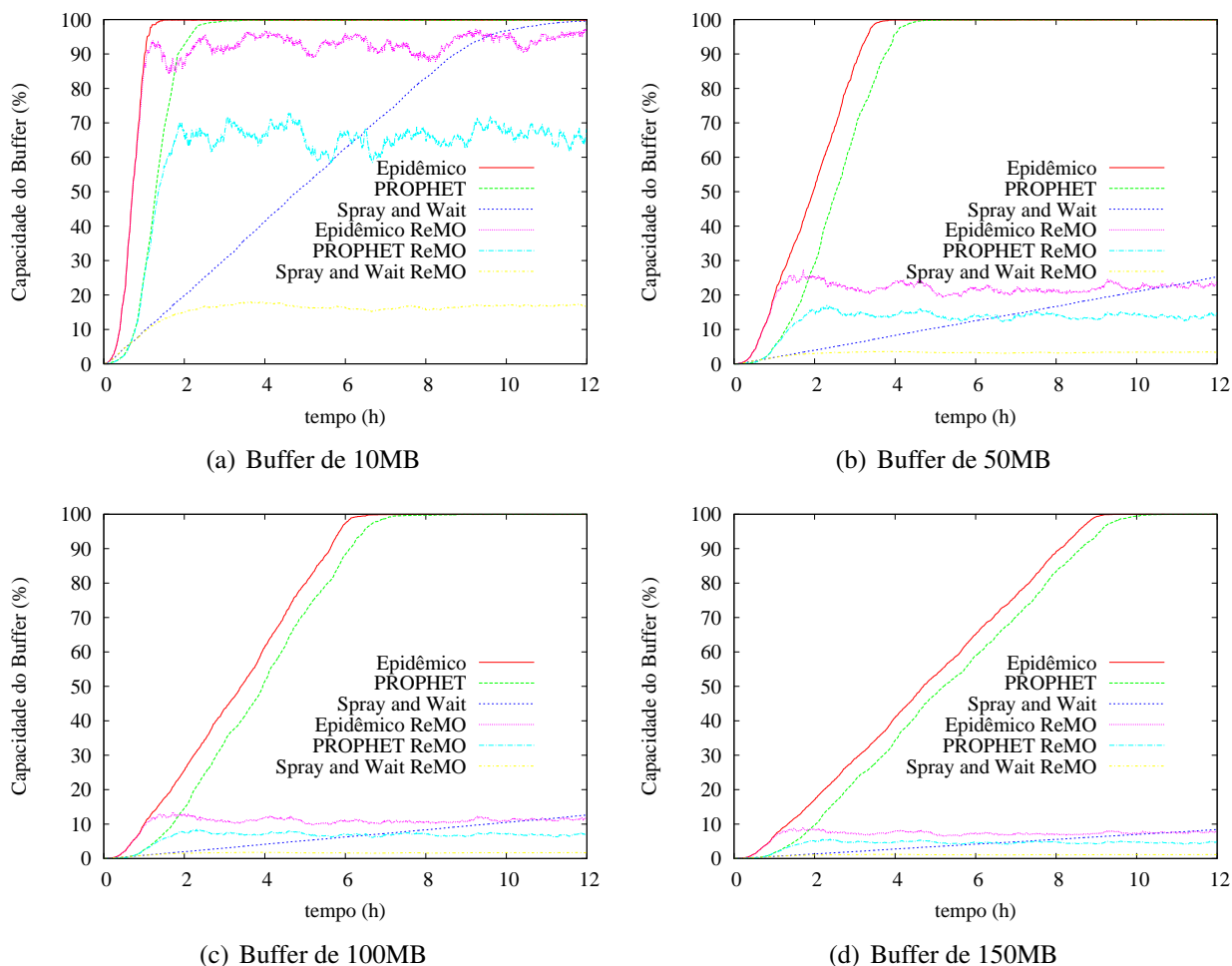
Já na Figura 5b cada nó possui um *buffer* de 50 MB. Com esta alteração, os nós terão mais espaço para armazenar as mensagens e os *buffers* dos nós demorarão mais tempo para ficarem cheios. Para esta situação, o protocolo de roteamento Epidêmico apresentou um percentual de ocupação do *buffer* de 99% sem o uso do ReMO e de 93%, com o uso do mesmo. Este menor percentual de ocupação permitiu ao protocolo Epidêmico entregar mais do dobro de mensagens. Sem o ReMO, a probabilidade de entrega é de 30% e com o ReMO foi de 71%. Já para o PROPHET, a melhora no percentual de ocupação foi de 28%, passando de 98% de média de percentual de ocupação dos *buffers* dos nós para 80% de média com o ReMO. Com isso, na métrica de probabilidade de entrega (ver Figura 2b), houve uma melhora de mais de 100% na probabilidade de entrega das mensagens.

A Figura 5c mostra o percentual de ocupação para *buffer* com 100 MB de tamanho. Pode-se observar que para os protocolos Epidêmico e PROPHET, o uso do ReMO diminuiu muito o percentual médio de ocupação dos nós, influenciando as métricas de desempenho avaliadas. A ocupação do *buffer* para o protocolo *Spray and Wait* não teve melhora com o uso do ReMO, pois como há um limite de cópias das mensagens e o *buffer* está com um valor alto, não houve ocupação maior que 12% do *buffer* sem o uso do ReMO, fazendo com que o protocolo funcione com um bom desempenho, mesmo sem o uso do ReMO.

Por fim, na Figura 5d cada nó possui um *buffer* de 150 MB. Nesse cenário a ocupação de *buffer* para os protocolos Epidêmico e PROPHET foi de 99% e com uso do ReMO foi de 56% e 48%, respectivamente; proporcionando uma redução de 77% na ocupação do *buffer* para o Epidêmico e uma redução de 106% na ocupação para o PROPHET. Mais uma vez, esses protocolos tiveram seu desempenho superior com o uso do ReMO e o protocolo *Spray and Wait*, conforme explicado no parágrafo anterior, não teve melhora devido a baixa taxa de ocupação do *buffer* do mesmo.

As Figuras 6a, 6b, 6c e 6d, apresentam o percentual de ocupação média do *buffer*, no tempo, de todos os nós dos cenários simulados, nos quais o tamanho das mensagens transmitidas na rede é de 100KB e o tamanho do *buffer* foi de 10MB, 50 MB, 100 MB e 150MB, respectivamente. O *buffer* da Figura 6a é o que possui menor capacidade de armazenamento entre os tamanhos analisados. Pode-se observar que os nós, utilizando os protocolos Epidêmico e PROPHET, atingiram porcentagens muito altas de ocupação do *buffer* (99%), logo nas primeiras horas de simulação. Isso aconteceu porque esses protocolos geram muitas cópias das mensagens. Já se observar os gráficos dos mesmos protocolos com o uso do ReMO, a porcentagem de ocupação do *buffer* diminuiu para 90% no Epidêmico e para 67% no PROPHET. Além disso, estes resultados influenciaram as métricas de desempenho anteriormente analisadas. Para a métrica de probabilidade de entrega, o uso do ReMO melhorou em 200% este resultado, passando de 30% de probabilidade de entrega para 90%. Já o protocolo *Spray and Wait* gera uma quantidade

limitada de cópias das mensagens, o que levou a uma menor ocupação do *buffer* dos nós. Isto aconteceu também na avaliação do *Spray and Wait* com o uso do mecanismo ReMO.



**Figura 6. Influência do mecanismo ReMO na ocupação do *buffer* para cada protocolo de roteamento com mensagens de 100KB de tamanho.**

Já na Figura 6b cada nó possui um *buffer* de 50 MB, no qual a capacidade de armazenamento dos nós é cinco vezes maior. Para os protocolos Epidêmico e PROPHET Figura 6, houve uma melhora nas métricas de desempenho, porém o *buffer* dos nós continua chegando próximo a 100% de ocupação. Assim, o uso do ReMO nesses protocolos é bem eficiente, apresentando uma melhoria de aproximadamente 30% na probabilidade de entrega de mensagens. Mais uma vez, para o protocolo *Spray and Wait* o ReMO não trouxe melhorias para as métricas analisadas.

A Figura 6c representa os nós com *buffer* de 100 MB. Mesmo com este tamanho de *buffer*, os protocolos de roteamento Epidêmico e PROPHET chegaram perto da ocupação máxima dos nós. Com isso, o ReMO diminuiu a ocupação do *buffer*, o que proporcionou uma maior probabilidade de repasse de mensagens e isso refletiu na métrica de probabilidade de entrega (ver Figura 2a), na qual houve 18% a mais de entregas das mensagens.

Por fim, na Figura 6d onde cada nó possui um *buffer* de 150 MB, teve um resultado similar aos analisados anteriormente. Entretanto, o *buffer* dos nós chegou perto da saturação somente próximo ao fim da simulação para os protocolos Epidêmico e

PROPHET, ocasionando uma melhoria de 5% na entrega das mensagens quando utilizado o ReMO.

## 5. Conclusão

Neste trabalho é mostrada a necessidade do uso de uma política para remoção de réplicas de mensagens que ficam obsoletas após a sua entrega no destino em DTN. Assim, se não existir um mecanismo para a remoção destas mensagens, o transbordamento desses *buffers* com mensagens obsoletas pode ocorrer. Por consequência disso, uma diminuição no desempenho da rede, como por exemplo, um aumento na probabilidade de perda de mensagens ou no tempo de entrega pode acontecer.

Assim, dentro deste contexto, o mecanismo proposto para remoção de mensagens assume que o nó destino quando em contato com outros nós comunica o recebimento da mensagem e assim, os nós que possuem cópias desta mensagem irão removê-las e ainda propagarão o recebimento da mensagem para outros nós.

Para avaliar a eficiência do mecanismo ReMO, o mesmo foi implementado no simulador ONE e os protocolos de roteamento Epidêmico, PROPHET e *Spray and Wait* foram avaliados sem o uso do ReMO e com seu uso.

Analisando os resultados obtidos, pode-se verificar que para os protocolos Epidêmico e PROPHET, que geram muitas cópias das mensagens durante sua execução, o ReMO obteve resultados importantes. Já para o protocolo *Spray and Wait*, o ReMO auxilia-o quando o tamanho do *buffer* é pequeno, mas conforme o tamanho do *buffer* aumenta, a contribuição do ReMO não é significativa.

Foram feitas comparações dos resultados do percentual de ocupação do *buffer* com os resultados da probabilidade de entrega, mostrando que para protocolos de roteamento que geram muitas cópias das mensagens, o desempenho da probabilidade de entrega está diretamente relacionado com a ocupação do *buffer* e conseqüente utilização do ReMO. Análises similares foram feitas para as métricas de atraso médio e sobrecarga das mensagens e pode-se observar que o desempenho dos protocolos de roteamento melhorou com o uso do ReMO.

## Referências

- Haas, Z. (2006). A new networking model for biological applications of ad hoc sensor networks. *Networking, IEEE/ACM Transactions on*, 14(1):27–40.
- Kaveevivitchai, S., Ochiai, H., and Esaki, H. (2010). Message Deletion and Mobility Patterns for Efficient Message Delivery in DTNs. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 760–763. IEEE.
- Keränen, A., Ott, J., and Kärkkäinen, T. (2009). 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. ICST.
- Lindgren, A. and Doria, A. (2003). Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing*, 7(3):19.

- Mundur, P. and Seligman, M. (2008). Epidemic routing with immunity in Delay Tolerant Networks. *Conference, 2008. MILCOM*, pages 1–7.
- Rashid, S. and Ayub, Q. (2010). Efficient Buffer management Policy DLA for DTN Routing Protocols under Congestion. *International Journal*, 2(9):118–121.
- Spyropoulos, T. and Psounis, K. (2008). Efficient routing in intermittently connected mobile networks: the multiple-copy case. *Networking, IEEE/ACM*, pages 1–14.
- T. Spyropoulos, K. Psounis, C. R. (2007). Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. *In IEEE PerCom, on the International Workshop on Intermittently Connected Mobile Ad hoc Networks (ICMAN)*.
- Vahdat, A. and Becker, D. (2000). Epidemic routing for partially connected ad hoc networks. *Science*.
- Yuen, W. (2010). Message Replication and Deletion in Delay Tolerant Networks under Hop-based and Time-based TTL Schemes. *Columbia University*, pages 1–12.