Escassez de Recursos em Redes Tolerantes a Atrasos e Interrupções *

Raphael M. Guedes[1], **Marcel W. R. da Silva**[1] **e José Ferreira de Rezende**[1]

[1]Grupo de Teleinformática e Automação (GTA) – COPPE
Universidade Federal do Rio de Janeiro (UFRJ)

{raphael,marcel,rezende}@gta.ufrj.br

Abstract. *Disruption/Delay Tolerant Networks (DTNs) have become increasingly important in the context of advanced networks research. These networks have as main important feature the robustness against the lack of an end-to-end connection. Thus, existing solutions make use of intermediate storage and opportunistic forwarding to deliver end-to-end messages. However, since they are mainly focused on scenarios with mobile devices, one of the problems that can affect their performance is resource constraint such as energy, transmission rate and storage space. This paper discusses the efficiency of some mechanisms that may represent major gains in terms of resources saving on DTNs. For this purpose, we have developed a simulator on top of NS-2 that focuses on resources scarcity problems of DTNs.*

Resumo. *As redes tolerantes a atrasos e interrupções (RTAIs) têm se tornado cada vez mais importantes no âmbito das pesquisas em redes avançadas. Estas redes possuem como principal característica a robustez à falta de conectividade fim-a-fim. Para isso, as soluções desenvolvidas realizam o armazenamento intermediário e o encaminhamento oportunista das mensagens geradas. Entretanto, por serem voltadas para cenários com dispositivos móveis, um dos problemas que podem afetar o seu desempenho é a restrição de recursos como energia, taxa de transmissão e capacidade de armazenamento. Este trabalho discute a eficiência de mecanismos que podem trazer ganhos em termos de economia de recursos nessas redes. Para esta finalidade foi desenvolvido um simulador, que utiliza funcionalidades básicas do NS-2, tendo como principal característica levar em conta os problemas de escassez de recursos nos nós que compõem essas redes.*

1. Introdução

Uma premissa básica para o bom funcionamento de redes "convencionais" [Lindgren et al. 2003], como as redes cabeadas e até mesmo as redes *ad hoc*, é a existência de uma conectividade fim-a-fim entre os terminais que desejam se comunicar. Na maioria dos protocolos de roteamento, o sucesso na descoberta de uma rota e, portanto, o início de uma comunicação, só é possível caso exista um caminho entre a fonte e o destino. Entretanto existem ambientes "desafiadores" onde nem sempre é possível garantir essa conectividade fim-a-fim, ou por ela nunca existir ou por ser intermitente. Entre as redes que apresentam alguns desses problemas, podemos citar: redes de satélite, redes que

*Este trabalho recebeu recursos do CNPq, CAPES, FAPERJ, FINEP e RNP.

utilizam gerenciamento de energia (redes de sensores), redes submarinas, redes interplanetárias, redes *ad hoc* esparsas, etc..

Estes desafios têm estimulado grupos de pesquisa na procura por soluções para estes problemas, como o DTNRG - *Delay Tolerant Network Research Group* criado pelo IRTF - *Internet Research Task Force*. A essas redes, que levam em consideração longos atrasos e freqüentes desconexões, dá-se o nome em inglês de DTN (*Delay and Disruption Tolerant Networks*) que mais recentemente foram denominadas Redes com Desafios (CHAllenged NeTworkS - CHANTS) [Chen and Murphy 2001] ou ainda Redes Oportunistas [Ramanathan et al. 2007]. Ao longo desse texto, será utilizado o termo em português RTAIs para Redes Tolerantes a Atrasos e Interrupções.

A solução para contornar a dificuldade das desconexões na entrega de mensagens é o armazenamento persistente em memória (*buffer*) dessas mensagens nos nós intermediários e o encaminhamento oportunista das mesmas, explorando a mobilidade dos próprios nós na tentativa de entregar a mensagem ao nó de destino [Lindgren et al. 2003]. Assim, nas RTAIs, quando se deseja enviar uma mensagem, esta é armazenada em *buffer* e encaminhada nó a nó desde a origem até o destino aproveitando oportunamente a disponibilidade de tais enlaces. Desta forma, o correto funcionamento dessas redes depende da capacidade de armazenamento dos seus nós, assim como da capacidade desses nós em transferir dados durante o tempo de contato entre eles.

Para que o desempenho dessas redes seja satisfatório, os recursos dos nós devem ser bem aproveitados. Como as mensagens podem levar muito tempo para serem entregues, é crucial que as mesmas permaneçam armazenadas pelo nó que as gerou assim como pelos nós intermediários. Com isso, se um nó possui pouco espaço em memória para este armazenamento é provável que o mesmo tenha de recorrer a políticas de descarte para poder acomodar novas mensagens. Se a mensagem descartada ainda não havia sido entregue, o desempenho da rede pode cair devido ao aumento da perda de mensagens. Quando dois nós entram em contato, o desejável é que cada um deles obtenha mensagens ainda não vistas. No entanto, mesmo que os nós possuam *buffer* suficiente, o tempo desse contato pode ser curto em relação ao tempo despendido na transmissão de cada mensagem, tornando-se impossível a transmissão de todas as mensagens desejadas. Desta forma, a difusão das mensagens pela rede fica prejudicada o que aumenta o tempo necessário para a entrega, levando eventualmente ao descarte dessas mensagens. O gasto de energia também pode representar um problema em RTAIs. Como a principal característica dessas redes é a mobilidade dos seus nós, supõe-se que tais nós sejam dispositivos restritos em energia. Então é muito importante evitar desperdícios com transmissões desnecessárias pois quando a energia de um nó acaba, todas as mensagens em seu *buffer* são perdidas.

Em resumo, a restrição de recursos em RTAIs é um problema importante que afeta diretamente o seu desempenho. A maioria dos mecanismos existentes para o aumento do desempenho dessas redes se vale direta ou indiretamente da conservação dos recursos disponíveis. Técnicas como, por exemplo, o uso de confirmações de mensagens entregues ou o emprego de políticas de encaminhamento e descarte de mensagens, nada mais são do que meios de reduzir ou aperfeiçoar o uso dos recursos disponíveis nestas redes. Visando estudar os impactos da restrição de recursos em RTAIs, este trabalho apresenta um conjunto de mecanismos simples para poupar recursos destas redes. Para permitir analisar tais mecanismos, foi desenvolvido um ambiente de simulação, utilizando alguns dos

recursos básicos do NS-2 e usando um nível de abstração que permitiu avaliar a influência das restrições de recursos nas RTAIs.

A Seção 2 discute alguns dos trabalhos em RTAIs relacionados ao problema de restrição de recursos. A Seção 3 apresenta em maiores detalhes os mecanismos relacionados às restrições de recursos propostos e avaliados. O simulador desenvolvido é objeto de estudo da Seção 4. Em seguida, a Seção 5 traz os resultados das simulações realizadas, assim como uma discussão a respeito. E, por fim, a Seção 6 apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Os protocolos tradicionais de roteamento não foram projetados para redes com freqüentes desconexões, e por isso não apresentam um desempenho adequado neste tipo de ambiente [Boice et al. 2007]. Devido a isso, novos protocolos foram concebidos, tais como o *Rapid* [Balasubramanian et al. 2007], SWIM [Small and Haas 2005], PROPHET [Lindgren et al. 2003], MV [B. Burns and Levine 2005] e MaxProp [Burguess et al. 2006], alguns modificados, tais como o SCaTR [Boice et al. 2007], que surge a partir do AODV, e o apresentado em [Jain et al. 2004] que é uma modificação do algoritmo de Dijsktra. De uma forma geral, essas diferentes propostas foram concebidas levando em consideração o grau de informação disponível da topologia da rede ao longo do tempo [Zhang 2006, Oliveira et al. 2007]. Assim sendo, essas soluções são geralmente voltadas para cenários determinísticos ou estocásticos. O cenário é dito determinístico quando se tem informação completa dos instantes de contato entre os nós e a duração dos mesmos. Caso contrário, o cenário é classificado como estocástico.

A primeira proposta, voltada para cenários estocásticos, foi o chamado roteamento Epidêmico [Mitchner and Vadhat 2000]. Em seu funcionamento, quando dois nós estão ao alcance um do outro, eles trocam um vetor (*summary vector*) que contém o resumo das mensagens contidas no *buffer* de cada um. De posse dessa informação, cada nó solicita as mensagens que não possui ao outro nó e, ao fim dessa troca de mensagens, o *buffer* dos dois nós tornam-se "espelhados", salvo restrições quanto à capacidade de armazenamento, duração do contato e taxa de transmissão. Este processo se repete a cada novo encontro, onde novas réplicas das mensagens são criadas permitindo uma rápida distribuição pela rede, exatamente como uma epidemia. Resultados de avaliação deste protocolo constatam uma alta taxa de entrega, porém sendo o seu maior problema o alto custo em número de transmissões e espaço em *buffer* utilizado.

Outra classificação dos protocolos de roteamento é com relação a quais mensagens devem ser encaminhadas a cada contato. Por esse critério, os protocolos classificam-se em replicadores, tais como, o *Rapid*, MV, SWIM, Epidêmico, MobySpace [Leguay et al. 2005] e *Spray and Wait* [Spyropoulos et al. 2005], ou encaminhadores, como o PROPHET, o MEED [Jones et al. 2005] e o MaxProp. Os replicadores tentam repassar cópias de todas as suas mensagens a cada encontro como numa inundação. Os encaminhadores utilizam procedimentos que definem qual mensagem deve ser transmitida. Por exemplo, pode ser utilizada uma probabilidade associada ao encontro de cada nó com os outros nós da rede e a partir disso decidir sobre o encaminhamento ou não de uma determinada mensagem. É relevante notar que os protocolos replicadores tendem a consumir mais recursos da rede que os encaminhadores.

A Tabela 1 apresenta alguns protocolos, dividindo-os entre replicadores ou encaminhadores, além de classificá-los em relação a vazão durante as oportunidades de transferência e, a capacidade de armazenamento dos nós, ambos como finita(o) ou ilimitada(o). Semelhante abordagem é feita em [Balasubramanian et al. 2007], contudo a informação entre parênteses refere-se ao tipo de política de descarte aplicada pelo protocolo, e não ao tipo de modelo de movimentação empregado nas simulações. As políticas de descarte indicam qual mensagem deve ser descartada quando o *buffer* se encontra cheio e, uma nova mensagem chega ao nó. Em [Davis et al. 2001], algumas políticas de descarte são enumeradas: descarte da mensagem mais antiga (FIFO), descarte aleatório (ALEAT) e descarte da mensagem cujo recebimento tenha sido recente. Em [Lindgren and Phanse 2006], algumas políticas foram propostas e avaliadas: mensagem mais encaminhada (MOFO), mensagem com o menor tempo de vida, mensagem com o menor valor de probabilidade de entrega, dentre outras.

<table>
<tr><th>Tipo</th><th>Armazenamento</th><th>Vazão</th><th>Roteamento (descarte)</th></tr>
<tr><td rowspan="2">Replicador</td><td rowspan="2">Finito</td><td>Finita</td><td>Rapid (menor utilidade)</td></tr>
<tr><td rowspan="3">Ilimitada</td><td>Davis et al. (várias políticas), MV (FIFO), SWIM (msg já entregue)</td></tr>
<tr><td rowspan="2">Ilimitado</td><td>Epidêmico(-), Spray and Wait(-)</td></tr>
<tr><td rowspan="3">Encaminhador</td><td>Alg. Dijkstra Modificado
Jain et al (overflow), MobySpace(-)</td></tr>
<tr><td rowspan="2">Finito</td><td>PROPHET (FIFO)</td></tr>
<tr><td>Finita</td><td>MEED (FIFO), MaxProp (maior número de saltos, menor prob.)</td></tr>
</table>

Tabela 1. Classificação de protocolos de roteamento

Existem outros protocolos que, motivados pela problemática de consumo de recursos, tentam impor um controle no número de réplicas que cada mensagem pode ter. Em [Harras et al. 2005], é colocado um limite na quantidade máxima de encaminhamentos de cada mensagem. Existem também propostas de modificação do protocolo Epidêmico [Spyropoulos et al. 2005, Grossglauser and Tse 2002, Ramanathan et al. 2007]. Em [Grossglauser and Tse 2002], para se reduzir o número de nós "contaminados" por uma mensagem é implementado um limite de salto em apenas dois, ou seja, um salto para o primeiro contato e outro posteriormente apenas ao destino. Desta forma, nenhuma mensagem é replicada mais de uma vez. Outra proposta é encontrada no protocolo *Spray and Wait*, onde somente a fonte pode replicar uma mensagem, tornando a quantidade de replicações proporcional ao número de nós na rede [Boice et al. 2007].

Alguns protocolos aplicam um procedimento de "limpeza" dos *buffers*, removendo mensagens que não são mais necessárias. Entre os protocolos que utilizam este método podemos citar o SWIM e o MaxProp. O protocolo SWIM faz uso de mecanismos, denominados VACCINE e IMMUNE_TX, cujo funcionamento é semelhante a uma lista de *ack's*. Antes da troca de mensagens, os dois nós trocam esta lista de *ack's*, que contém os identificadores das mensagens que já foram entregues ao nó de destino e não necessitam mais continuar sendo armazenadas. A diferença entre os dois protocolos está na forma de composição da lista. No MaxProp, assim que um nó encaminha

uma mensagem para o destino, esta mensagem é acrescida a lista de *ack's*. No protocolo SWIM, a mensagem é colocada na lista de *ack's* quando entregue a um nó intermediário, denominado coletor, e não ao destino final. Em [Harras et al. 2005], um mecanismo, denominado de cura passiva (*passive cure*), tenta apagar do *buffer* dos nós mensagens já entregues, através da "inundação" de *ack's*.

Uma característica que pode ser percebida nas propostas para melhoria de desempenho de RTAIs é que muitas giram em torno do aperfeiçoamento do uso dos recursos disponíveis. O mau uso dos recursos disponíveis pode fazer com que o espaço em *buffer* seja ocupado por mensagens desnecessárias e com que muitas transmissões sejam realizadas, causando um gasto energético excessivo. Indo mais além, uma união dos dois problemas mencionados anteriormente pode fazer com que os *buffers* dos nós da rede fiquem sobrecarregados, causando descartes desnecessários de mensagens que ainda não foram entregues ao destino final.

Em cenários onde os recursos da rede são restritos, os nós contam com pouco espaço em memória, limitações de energia e taxas de transmissão baixas. Com isso, os problemas causados pelo uso inadequado dos recursos da rede ficam ainda mais evidentes. Além disso, podem surgir outros problemas como a dificuldade de enviar mensagens à um outro nó, devido a necessidade de um tempo maior de contato ocasionado pelas baixas taxas de transmissão. Em cenários de alta mobilidade, onde o tempo de contato é curto, esta característica pode ser um problema. Entretanto, quando tais recursos são preservados ou utilizados de maneira "inteligente", os ganhos podem ser muito expressivos, resultando em maiores taxas de entrega de mensagens e/ou maior tempo de vida dos nós da rede.

3. Mecanismos de Gerenciamento de Recursos em RTAIs

Como já foi mencionado, o problema da escassez de recursos em RTAIs é importante pois pode afetar diretamente seu funcionamento e desempenho. Com esta finalidade serão apresentadas algumas propostas, que podem fornecer melhorias de desempenho na economia de recursos e que foram implementadas e avaliadas no simulador desenvolvido.

Uma das maneiras de melhorar a utilização dos recursos é aperfeiçoar a utilização do espaço em memória. Um mecanismo simples, proposto no protocolo MaxProp, é a troca de confirmações (*ack's*) sobre mensagens que já alcançaram seu destino final. Toda vez que um nó entrega uma mensagem a seu destino final, o mesmo adiciona um *ack* a sua lista de mensagens entregues. Tal mecanismo gera pouca sobrecarga, pois os nós sincronizam suas listas de identificadores das mensagens que já possuem confirmação conhecida e esta informação pode ser armazenada e trocada na forma de uma tabela *hash*. Com isto, os nós podem descartar do seu *buffer* todas as mensagens que já foram entregues ao destino final, aumentando o espaço disponível para mensagens ainda não entregues e evitando o descarte de mensagens que ainda são relevantes.

Outro problema importante para o caso de redes com recursos escassos é a determinação da prioridade das mensagens a serem enviadas. Dependendo do padrão de mobilidade existente em uma determinada rede, da capacidade de transmissão dos nós e do tamanho das mensagens, o tempo em que os nós permanecem em contato pode ser insuficiente para o envio de todas as mensagens desejadas. Para contornar o problema utilizam-se políticas de prioridade no encaminhamento de mensagens. Isto faz com que

haja uma garantia de que as mensagens com maior importância, como por exemplo, as destinadas aquele nó receptor ou as menos enviadas, sejam enviadas primeiro aumentando a possibilidade de serem entregues durante aquele contato de curta duração.

Para os nossos experimentos implementamos no simulador uma política em etapas. Inicialmente são priorizadas as mensagens destinadas àquele nó receptor, visando aumentar a taxa de entrega. Depois disso, as mensagens que foram menos encaminhadas, com o intuito de aumentar a propagação das mensagens pela rede. Se houver empate de várias mensagens com o mesmo número de encaminhamentos, priorizam-se aquelas originadas pelo próprio nó emissor. E por fim, escolhe-se a mensagem mais antiga do *buffer* (FIFO).

O uso de mecanismos que tentam garantir espaço em *buffer* para as mensagens recebidas, nem sempre evita enchimento de *buffer*. Assim, o descarte de mensagens pode ser inevitável. Nestes casos é necessário que os nós utilizem políticas de descarte, para garantir que mensagens mais importantes sejam preservadas. Alguns tipos de políticas de descarte de mensagens já existentes na literatura foram mencionadas na Seção 2, para os nossos experimentos implementamos no simulador as políticas: FIFO, que descarta a mensagem mais antiga do *buffer* e MOFO, que descarta a mensagem que já foi mais encaminhada por aquele nó.

Um outro tipo de abordagem para evitar que mensagens sejam descartadas, apesar de trivial, nunca foi mencionado ou utilizado na literatura em protocolos de roteamento para RTAIs. Nesta outra abordagem, que denominamos **encaminhamento proativo**, são enviadas apenas as mensagens que se sabe que o receptor pode armazenar. Isto pode evitar que mensagens sejam descartadas e que muitas transmissões sejam realizadas em possível detrimento da taxa de entrega, que pode cair devido a menor troca de mensagens.

Para implementar este tipo de mecanismo, os nós em contato podem trocar mensagens no início da negociação da troca de mensagens. Por exemplo, no caso do roteamento Epidêmico, a informação adicional a respeito do tamanho do espaço disponível em *buffer* pode ser trocado entre os nós junto dos vetores de sumário (*summary vectors*). Assim, além de saber as mensagens que precisam ser enviadas para o outro nó, um dos nós saberá também quantas mensagens seu par ainda pode comportar. Desta forma, o nó emissor pode empregar as suas políticas de encaminhamento e selecionar para o envio apenas as mensagens que não irão gerar descarte ao chegar no nó receptor.

Apesar de trivial e simples, este mecanismo proposto possui um apelo forte no que diz respeito à economia de recursos de RTAIs. Quando o *buffer* de um nó encontra-se cheio, o mesmo só terá a chance de retirar mensagens dele quando encontrar com o destino de alguma delas ou quando encontrar com outro nó que possui o *ack* de alguma delas (no caso do uso de listas de *ack's*). Isso poderá fazer com que menos transmissões sejam realizadas, economizando a energia despendida nesta tarefa. Além disso, fará com que nenhuma mensagem seja descartada, evitando um possível desperdício de alguma informação que ainda não foi entregue.

Em contrapartida, o encaminhamento proativo pode possuir o efeito indesejado de impedir que as mensagens sejam disseminadas rapidamente pela rede, o que pode aumentar o atraso na entrega da informação e reduzir a taxa de entrega. Portanto para este mecanismo pode existir um compromisso entre a possível redução na taxa de entrega e a

redução do gasto de energia.

Para avaliar o impacto dos mecanismos apresentados e do encaminhamento proativo proposto nesta seção no problema da escassez de recursos em RTAIs, desenvolvemos um simulador capaz de levar em conta tais fenômenos nestas redes.

4. Simulador

Para avaliar o problema da escassez de recursos em RTAIs, desenvolvemos um ambiente de simulação que utiliza recursos básicos do NS-2 (*Network Simulator*), um simulador a eventos discretos. Aproveitando-se desta característica do NS-2, o simulador desenvolvido evolui em intervalos fixos de tempo que denominamos de *snapshots*. A cada *snapshot*, todos os contatos existentes entre os nós são determinados e, baseado nisso, todas as trocas de mensagens possíveis entre os nós são realizadas. Após essa troca, a simulação avança para o próximo *snapshot*. A lista de contatos de um determinado nó, construída a cada *snapshot*, é composta pelos nós que se encontram a uma distância igual ou inferior ao alcance de transmissão.

Em cada *snapshot*, a troca de mensagens é realizada de maneira seqüencial do seguinte modo. No início do *snapshot*, todos os nós da rede podem trocar mensagens, ou seja, fazem parte de uma lista de nós habilitados para a comunicação. Então, um nó, escolhido aleatoriamente desta lista, verifica seus contatos e, dentre estes, escolhe um como par de comunicação. A cada *snapshot* apenas uma mensagem pode ser trocada entre esses dois nós e tanto os nós vizinhos do nó fonte quanto os vizinhos do nó receptor ficam impedidos de trocar mensagens naquele mesmo *snapshot*. Com isso, simulamos a existência de um protocolo simples de controle de acesso ao meio que evita colisões e que permite transmissões concorrentes no mesmo *snapshot* apenas quando todos os nós envolvidos estão fora dos alcances de interferência. Por exemplo, um nó A escolhido aleatoriamente em um *snapshot*, verifica inicialmente quais são seus vizinhos de um salto. Dentre estes vizinhos, o nó A escolhe um nó B para ser seu par de comunicação. Depois de realizada a troca de uma das mensagens a serem trocadas entre os nós A e B, os vizinhos de A e de B ficam impedidos de se comunicar com qualquer outro nó naquele mesmo *snapshot*. Em seguida a essa troca de mensagem, todos os nós "envolvidos" nessa comunicação (nós A, B e seus vizinhos) são retirados da lista de nós habilitados a realizar uma comunicação ainda nesse *snapshot*. Então, um novo nó é escolhido aleatoriamente dessa lista e assim sucessivamente até que a lista se esvazie por completo. Quando isso ocorre, a simulação avança para o próximo *snapshot*. No exemplo anterior, em *snapshots* sucessivos, os nós A e B continuam trocando mensagens até que todas as mensagens escolhidas tenham sido trocadas ou quando A e B saírem de alcance.

A escolha do intervalo de tempo entre *snapshots* pode ser vista como uma maneira de modelar a restrição de recursos da rede, pois o intervalo determina quanto tempo é necessário para a transmissão de uma única mensagem. Por exemplo, se utilizarmos mensagens de 10 KBytes e o intervalo de tempo do *snapshot* for de 0.1 s, isto equivale a uma taxa de transmissão de 800 Kbps. Se aumentarmos para um intervalo de tempo de 0.5 s, a taxa de transmissão passa a ser de 160 Kbps.

Cada mensagem possui um identificador único na rede. Além disso, todos os nós mantém informações atualizadas para todas as mensagens presentes em seus *buffers*, sobre quantos saltos ela já fez até chegar a este nó, a quantidade de vezes que foi enca-

minhada pelo nó em questão, o instante em que foi gerada, etc.. Na implementação, cada nó possui dois *buffers*, um para as mensagens próprias, ou seja, as geradas pelo próprio nó (*source buffer*) que é ilimitado e outro para as mensagens sujeitas a encaminhamento (*relay buffer*), este finito. A utilização de um *source buffer* ilimitado é uma escolha de implementação, justificada pelo fato de que o nó que gerou a mensagem não tem interesse em descartá-la antes que ela seja repassada.

No instante da troca de mensagens, como o nó só pode enviar uma mensagem por *snapshot*, é de extrema importância o uso de alguma política de encaminhamento. Para determinar quais mensagens devem ser repassadas no caso de um contato, adotamos o protocolo Epidêmico. Além disso, como já foi citado na Seção 3, alguns mecanismos para a contenção de recursos em RTAIs já existentes e o encaminhamento proativo foram implementados no simulador.

5. Resultados de Simulação

Para os experimentos realizados foram utilizados 50 nós, com alcance de 100 metros, se movimentando em uma área quadrada de 1000 metros de lado durante 1000 s. Para a movimentação dos nós dois tipos de cenários foram gerados, com baixa e alta mobilidade, ambos utilizando o modelo de mobilidade *Random Waypoint*, que cria um tipo de movimentação aleatória simples e não tendenciosa. Para os cenários com baixa mobilidade os nós se movimentam com velocidade máxima de 3 m/s e tempo de pausa de 10 s, simulando um grupo de pessoas portando algum tipo de equipamento portátil, como apresentado em [Leguay et al. 2006], que consistia na propagação de um jornal eletrônico entre alunos de uma universidade. Já no cenário de alta mobilidade, a velocidade máxima era de 10 m/s e o tempo de pausa de 60 s, simulando um cenário de redes veiculares em ruas fechadas de uma cidade.

Cada mensagem gerada tinha um tamanho fixo de 10 KBytes, e dois tipos de tráfego foram utilizados, com carga baixa, onde cada nó gerou apenas uma mensagem destinada a outro nó aleatório; e carga alta, onde cada nó gerou 5 mensagens com destinos aleatórios. Os instantes de geração de mensagens foram distribuídos durante os primeiros 20% segundos iniciais de cada simulação. Além disso, foram criados 30 cenários distintos para cada configuração de mobilidade. Com isso, todos os resultados apresentados serão médias dos resultados dos 30 cenários com intervalos de confiança de 95%.

A política de encaminhamento de mensagens utilizada nos experimentos foi aquela descrita na Seção 3. Já para as políticas de descarte testamos a MOFO e a FIFO. Entretanto, devido a pouca diferença nos resultados, adotamos apenas a política MOFO.

5.1. Avaliação da Duração do *Snapshot*

Para avaliar a influência do tamanho do *snapshot* realizamos simulações variando este parâmetro e fixando o tamanho do *relay buffer* em 10 mensagens e utilizando carga baixa (1 mensagem por nó). A métrica avaliada nestas simulações foi a taxa de entrega. Foram realizadas simulações com valores entre 0,25 s e 1 s, equivalente à cenários onde a taxa dos nós varia de 320 Kbps até 80 Kbps, com o intuito de avaliar o impacto deste tipo de restrição de recursos nos cenários simulados. Além disso, não foram habilitados nenhum tipo de mecanismo para previnir o descarte de mensagens ou transmissões desnecessárias.

Os resultados obtidos mostram que em cenários de baixa mobilidade a taxa de entrega pouco varia de acordo com a taxa de transmissão, ficando em torno de 50% em todas as faixas de tamanhos de *snapshot*. Isto ocorre, pois com a baixa mobilidade o tempo médio de contato é grande e permite a troca de várias mensagens antes do fim do contato. Já no caso de alta mobilidade, nota-se alguma variação na taxa de entrega. Com um tamanho de *snapshot* de 0,25 s, a taxa de entrega ficou em torno de 79%, e para um *snapshot* de 1 s, ficou em torno de 73%, sem sobreposição dos intervalos de confiança. Com isso, constatou-se que o aumento da mobilidade e o aumento da duração do *snapshot* tornou mais difícil a entrega de todas as mensagens desejadas durante o período de contato, causando uma queda na taxa de entrega. Além disso, também podemos perceber que com o aumento da mobilidade houve um aumento na taxa de entrega absoluta. Isto é fruto da maior fluidez de mensagens pela rede, causada pelo aumento da quantidade de contatos realizados e da maior troca de mensagens.

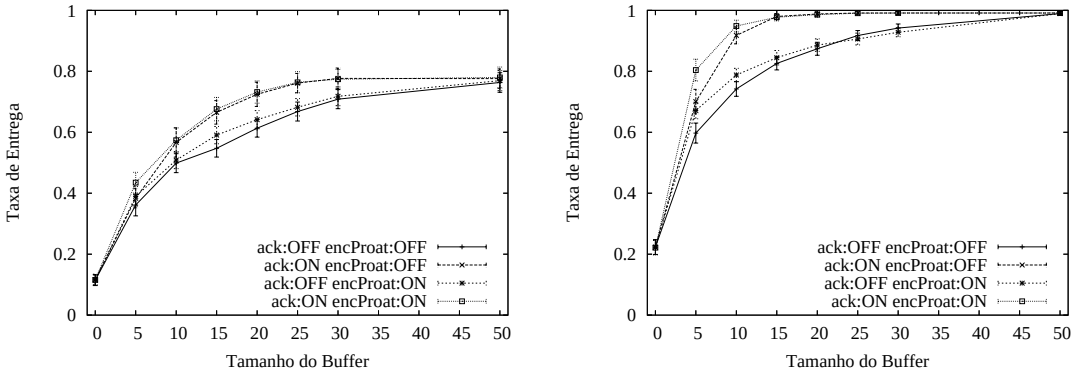
5.2. Avaliação do Tamanho do *buffer*

Para avaliar o efeito da restrição da quantidade de armazenamento em *buffer* dos nós realizamos simulações variando este parâmetro de zero até 50 mensagens. O uso de *buffer* de tamanho zero permitiu avaliar o desempenho das redes no caso em que apenas a transmissão direta de mensagens fosse permitida. Nestes casos, apenas as oportunidades geradas pela mobilidade dos nós é que fazem com que as mensagens sejam entregues. Isso é comprovado pelos resultados de número médio de saltos das mensagens entregues que, no caso *buffer* de tamanho zero, é igual a 1. O tamanho de *buffer* igual a 50 corresponde ao número total de mensagens geradas nos cenários de baixa carga. Nesses cenários, com esses valores de tamanho de *buffer*, é como se estivéssemos utilizando *buffers* de tamanho infinito.

Além disso, para estas simulações mantivemos o intervalo de tempo do *snapshot* em 1 s e serão apresentados os resultados para testes com dois dos mecanismos de economia de recursos discutidos na Seção 3: a lista de *ack's* e o encaminhamento proativo proposto. As métricas avaliadas nesses experimentos foram a taxa de entrega, o número de mensagens enviadas, número de mensagens confirmadas na origem (mensagens cujo *ack* foi recebido pelo seu nó emissor) e número médio de saltos das mensagens entregues. As Figuras 1, 2, 3 e 4 mostram os resultados para as métricas avaliadas no caso de carga baixa. Já as Figuras 5, 6, 7 e 8 apresentam o resultado para as mesmas métricas no caso de carga alta.

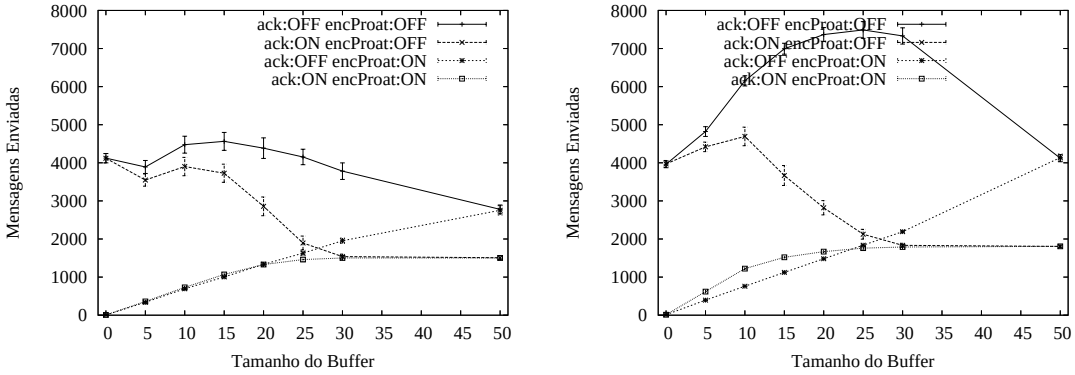
Os gráficos da Figura 1 constatam melhores resultados com a lista de *ack's* e encaminhamento proativo acionados, sendo que em baixa mobilidade, o uso da lista de *ack's* se torna mais influente. A taxa de entrega nos cenários de baixa mobilidade é menor que nos de alta mobilidade devido justamente a diferença no número de oportunidades de novos pares de comunicação. Em baixa mobilidade os nós tendem a trocar mais mensagens com o mesmo contato, ao contrário de cenários de alta mobilidade onde as oportunidades de novos pares é maior, garantindo uma maior fluidez das mensagens pela rede o que reflete numa taxa de entrega maior. Em carga alta, a taxa de entrega do esquema com ambos os mecanismos ativados se distancia dos demais esquemas à medida que o tamanho dos *buffers* aumenta.

A partir da Figura 2 vemos que, tanto em carga baixa como em carga alta, o número de mensagens enviadas com o uso de *ack's* e encaminhamento proativo diminui



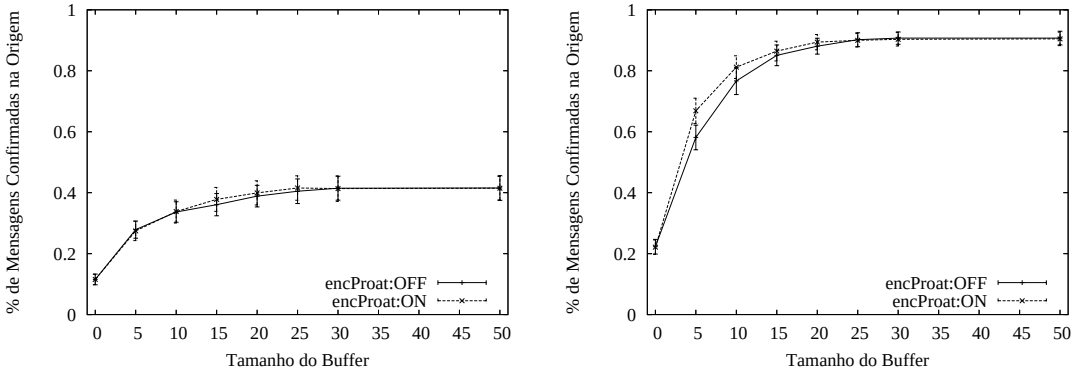
(a) Baixa mobilidade. (b) Alta mobilidade.

Figura 1. Taxa de entrega (carga = 50 mensagens).



(a) Baixa mobilidade. (b) Alta mobilidade.

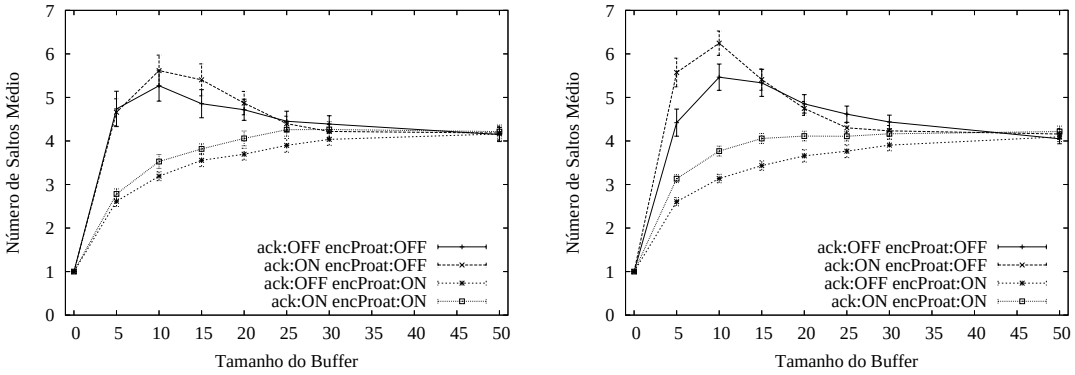
Figura 2. Número de mensagens enviadas (carga = 50 mensagens).



(a) Baixa mobilidade. (b) Alta mobilidade.

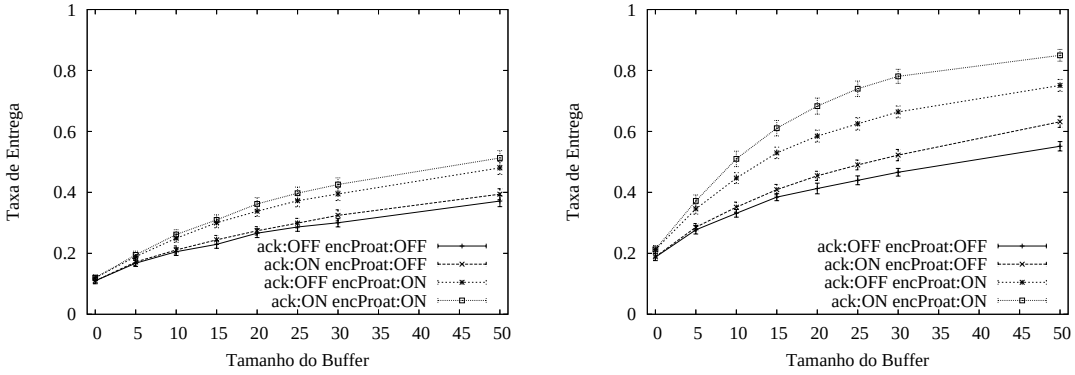
Figura 3. Número de mensagens confirmadas na origem (carga = 50 mensagens).

bastante. Isto ocorre, pois o uso de *ack's* indica quais mensagens podem ser apagadas de maneira segura dos nós intermediários, o que diminui a quantidade de mensagens em *buffer* e evita transmissões desnecessárias. Além disso, o encaminhamento proativo permite que um nó só envie o número de mensagens que o seu par suporta, diminuindo



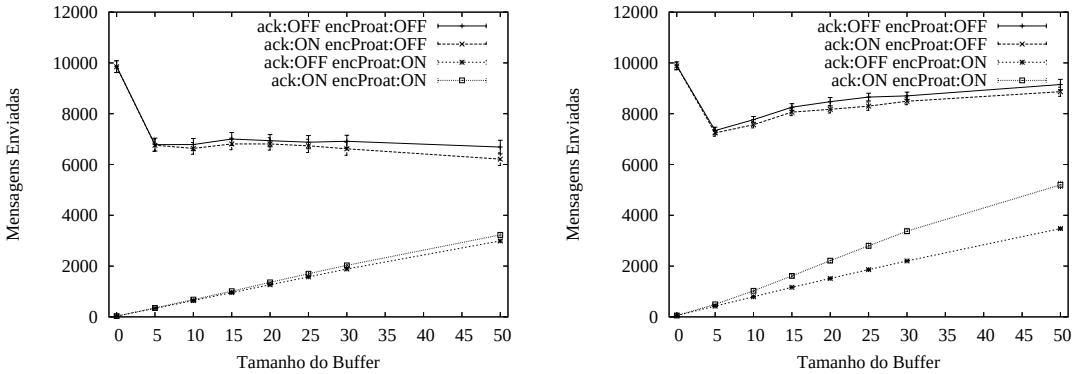
(a) Baixa mobilidade. (b) Alta mobilidade.

Figura 4. Número médio de saltos das mensagens entregues (carga = 50 mensagens).



(a) Baixa mobilidade. (b) Alta mobilidade.

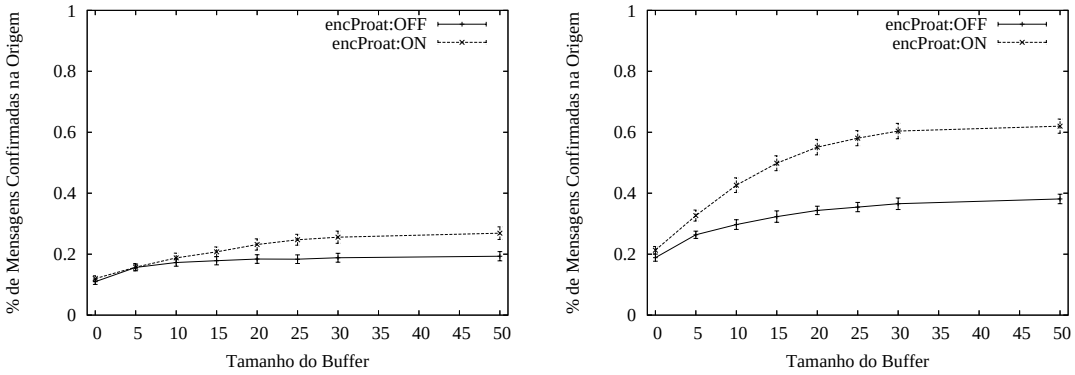
Figura 5. Taxa de entrega (carga = 250 mensagens).



(a) Baixa mobilidade. (b) Alta mobilidade.

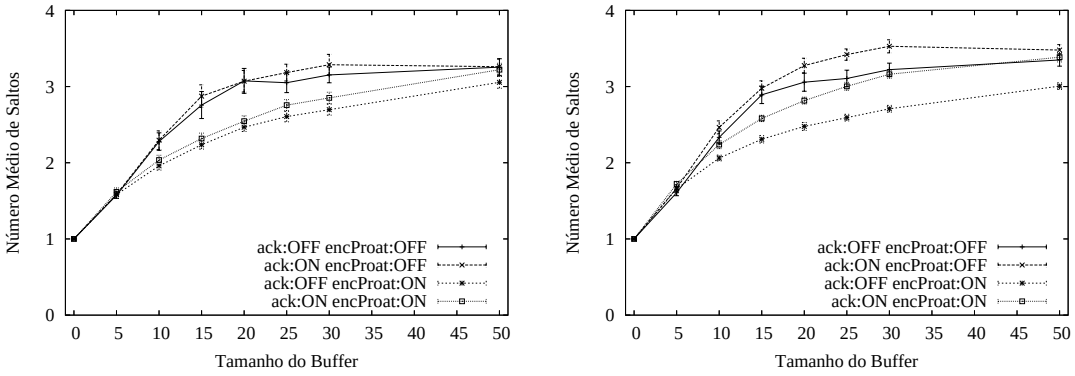
Figura 6. Número de mensagens enviadas (carga = 250 mensagens).

a quantidade de mensagens enviadas. Com as políticas desligadas, quando o tamanho do *buffer* aumenta, ocorre um aumento no número de transmissões, pois o espaço possível de armazenamento é maior e o vetor de sumários contém mais mensagens. Entretanto, a



(a) Baixa mobilidade. (b) Alta mobilidade.

Figura 7. Número de mensagens confirmadas na origem (carga = 250 mensagens).



(a) Baixa mobilidade. (b) Alta mobilidade.

Figura 8. Número médio de saltos das mensagens entregues (carga = 250 mensagens).

partir de um dado instante, os *buffers* tendem a ficar "espelhados" e por isso o número de transmissões decai.

O número de mensagens enviadas em carga alta, nos dois cenários de mobilidade, é menor com o encaminhamento proativo acionado, sendo mais expressivo com valores de *buffer* maiores em alta mobilidade. Com carga baixa e valores de *buffer* menores, nos dois cenários, o desempenho é próximo.

O número de mensagens confirmadas na origem, nos dois cenários, apresentou melhores resultados com o encaminhamento proativo acionado, e em cenários de alta mobilidade obteve-se melhores valores. Em carga alta, o desempenho foi menor, porém vale ressaltar que o tempo de simulação foi o mesmo para os dois cenários de carga.

O número médio de saltos das mensagens entregues tende ao mesmo valor com o aumento do tamanho dos *buffers* nos dois cenários em diferentes condições de carga. No entanto, para buffer menores, os esquemas com encaminhamento proativo ativado tendem a fornecer um menor número médio de saltos.

6. Conclusões

As RTAIs são um tipo de rede de grande importância por não necessitarem de requisitos de conectividade fim-a-fim, conseguindo aproveitar de maneira oportunista as características de mobilidade dos nós para o seu funcionamento. Entretanto, são redes onde o problema da escassez de recursos pode afetar bastante o seu desempenho. Neste trabalho, desenvolvemos um ambiente de simulação de RTAIs que leva em conta os problemas gerados pela escassez de recursos intrínseca a essas redes. Isto permitiu a avaliação do impacto de tais restrições no seu desempenho através da avaliação da influência de alguns mecanismos já existentes na literatura e de uma proposta simples de encaminhamento proativo, até então não empregada ou avaliada em outros trabalhos.

Com os resultados obtidos pode-se constatar como o emprego de mecanismos de gerência de recursos em RTAIs, como o uso de *ack's* e o encaminhamento proativo, fornecem ganhos significativos. O uso de *ack's* e do encaminhamento proativo se mostraram maneiras eficientes de reduzir a quantidade de transmissões realizadas pelos nós e de reduzir o número de saltos na entrega de mensagens mantendo ou melhorando a taxa de entrega. Isto mostra a importância do emprego de tais mecanismos para a preservação dos recursos das RTAIs melhorando o gerenciamento dos mesmos.

Como trabalhos futuros pretende-se avaliar o impacto de outros tipos de mecanismos de preservação de recursos em RTAIs. Além disso, pretende-se também realizar o mesmo estudo em outros protocolos de roteamento, como por exemplo, o PROPHET.

Referências

B. Burns, O. B. and Levine, B. N. (2005). MV Routing and Capacity Building in Disruption Tolerant Networks. In *Proceeding of IEEE INFOCOM*.

Balasubramanian, A., Levine, B., and Venkataramani, A. (2007). DTN Routing as a Resource Allocation Problem. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 373–384, New York, NY, USA. ACM.

Boice, J., Garcia-Luna-Aceves, J., and Obraczka, K. (2007). On-Demand Routing in Disrupted Environments. In *Proceedings of IFIP Networking 2007*, pages 155–166. Springer.

Burguess, J., Gallagher, B., Jensen, D., and Levine, B. (2006). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proceedings of IEEE INFOCOM*.

Chen, X. and Murphy, A. L. (2001). Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks. In *Proceedings of the Workshop on Principles of Mobile Computing (POMC), co-located with PODC*.

Davis, J. A., Fagg, A. H., and Levine, B. N. (2001). Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks. In *Proceedings of the International Symposium on Wearable Computers (IEEE ISWC)*.

Grossglauser, M. and Tse, D. N. C. (2002). Mobility Increases the Capacity of Ad Hoc Wireless Networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486.

Harras, K., Almeroth, K., and Belding-Royer, E. (2005). Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks. In *International Conferences on Networking (IFIP 2005)*.

Jain, S., Fall, K., and Patra, R. (2004). Routing in a delay tolerant network. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2004)*.

Jones, E. P. C., Li, L., and Ward, P. A. S. (2005). Practical Routing in Delay-Tolerant Networks. In *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN 2005)*.

Leguay, J., Friedman, T., and Conan, V. (2005). DTN Routing in a Mobility Pattern Space. In *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN 2005)*, pages 276–283, New York, NY, USA. ACM.

Leguay, J., Lindgren, A., Scott, J., Friedman, T., and Crowcroft, J. (2006). Opportunistic Content Distribution in an Urban Setting. In *Proceedings of SIGCOMM Workshop on Challenged Networks (CHANTS 2006)*, pages 205–212, New York, NY, USA. ACM.

Lindgren, A., Doria, A., and Scheln, O. (2003). Probabilistic Routing in Intermittently Connected Networks. In *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*.

Lindgren, A. and Phanse, K. S. (2006). Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks. In *Proceedings of the International Conference on Communication System Software and Middleware (Comsware 2006)*.

Mitchner, W. and Vadhat, A. (2000). Epidemic Routing for Partially Connected Ad Hoc Networks. Technical report, Duke University.

Oliveira, C. T., Moreira, M. D. D., Rubinstein, M. G., Costa, L. H. M. K., and Duarte, O. C. M. B. (2007). Redes Tolerantes a Atrasos e Desconexões. In *Minicursos do Simpósio Brasileiro de Redes de Computadores (SBRC 2007)*.

Ramanathan, R., Hansen, R., Basu, P., Rosales-Hain, R., and Krishnan, R. (2007). Prioritized Epidemic Routing for Opportunistic Networks. In *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp 2007)*, pages 62–66, New York, NY, USA. ACM.

Small, T. and Haas, Z. (2005). Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks. In *Proceedings ACM WDTN*.

Spyropoulos, T., Psounis, K., and Raghavendra, C. S. (2005). Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proceedings of ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN 2005)*, pages 252–259, New York, NY, USA. ACM.

Zhang, Z. (2006). Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communication Surveys and Tutorials*.