

APRP-Group: Roteamento para Redes DTN com Repasse baseado em Agrupamento de Nodos por Potencial de Entrega

Cristina M. Nunes, Fernando Luís Dotti, João Batista S. de Oliveira

¹Faculdade de Informática

Pontifícia Universidade Católica do Rio Grande do Sul (PUC-RS)

Av. Ipiranga, 6681 – Porto Alegre – RS – Brasil

{cristina.nunes,fernando.dotti,joao.souza}@pucrs.br

Abstract. *In this paper we propose APRP-Group, a new routing protocol for Delay Tolerant Networks (DTN) based in the number of meetings with other nodes over a time interval (named delivery potential). This protocol, an improvement of APRP and APRP-Ack, identifies clusters of nodes with similar potential, and considers the cluster potential in the forwarding decision. In comparison to Spray and Wait, APRP-Group presents, in sparse networks, reducing the overhead (up to 20 %) and increasing in delivery rate (up to 9 %) at the price of an increase in delivery delay (up to 22 %).*

Resumo. *Neste artigo é proposto o APRP-Group, um novo protocolo de roteamento para Redes Tolerantes a Atrasos (DTN) que se baseia no número de encontros com outros nodos em um período de tempo (chamado de potencial de entrega). Esse protocolo, uma melhoria do APRP e do APRP-Ack, busca identificar grupos de nodos com potencial de entrega similares e utiliza os potenciais desses grupos para decidir sobre o repasse das mensagens. Comparado ao Spray and Wait, o APRP-Group apresenta, em redes esparsas, redução de overhead (até 20%) e aumento na taxa de entrega (até 9%) ao custo de um aumento no atraso de entrega (até 22%).*

1. Introdução

Assim como as redes *ad hoc*, as redes DTN (*Delay Tolerant Network*) são formadas por múltiplos nodos que se comunicam sem a necessidade de estações base ou de uma rede com infra-estrutura fixa pré-existente. Contudo, segundo [Jain et al. 2004], as desconexões nas redes DTN são frequentes, muitas vezes não existindo um caminho entre uma origem e um destino que desejem se comunicar. As principais razões mencionadas na literatura para as desconexões são a mobilidade dos nodos e a operação em ambientes hostis [Zhang 2006]. De acordo com [Spyropoulos et al. 2008], as redes DTN podem ser aplicadas nos seguintes cenários: redes de sensores que monitoram a vida selvagem, redes militares, redes *ad hoc* veiculares (VANETs), redes para fornecer acesso a Internet de baixo custo a comunidades remotas, entre outras.

A falta de conectividade entre os nodos também é considerada em redes *ad hoc*, entretanto a abordagem utilizada é diferente. Para resolver o roteamento em redes *ad hoc*, supõe-se que um caminho contemporâneo válido entre origem e destino é possível. Diversos protocolos foram especificados para tais redes, como: AODV, DSR e DSDV [Liu and Kaiser 2005]. Tais protocolos são variações de protocolos de roteamento para

redes tradicionais e utilizam o mecanismo *store and forward* no repasse das mensagens. Por outro lado, nas redes DTN esse mecanismo não funciona adequadamente, dada a inexistência de caminhos contemporâneos de comunicação entre os nodos. O mecanismo utilizado em DTN é chamado de *store-carry-forward* [Zhang 2006], segundo o qual um nodo, ao receber uma mensagem, pode armazená-la até encontrar outro nodo propício para o repasse em direção ao destinatário.

Segundo [Zhang 2006], os protocolos de roteamento utilizados nas redes DTN podem ser divididos em determinísticos e estocásticos. Os determinísticos consideram que as mudanças topológicas são conhecidas no tempo e pode-se calcular as jornadas das mensagens *a priori* [Bui-Xuan et al. 2003]. Já nos protocolos estocásticos as mudanças topológicas não são conhecidas *a priori* e surgem técnicas para tentar aumentar a probabilidade de sucesso no roteamento, utilizando conhecimento histórico do comportamento da rede para tomar a decisão.

A literatura apresenta diversos trabalhos que propõem formas de roteamento estocástico para redes DTN, como em [Abdulla and Simon 2007], [Lindgren et al. 2003], [Spyropoulos et al. 2008] e [Vahdat and Becker 2000]. Entretanto, o problema de roteamento em redes DTN ainda está em aberto e formas mais eficientes têm sido investigadas.

Em trabalho prévio [Nunes and Dotti 2009, Nunes et al. 2009], foram propostos dois protocolos para redes DTN estocásticas: o APRP (*Adaptive Potential Routing Protocol*) e o APRP-Ack, sendo que este último usa mensagens de confirmação para liberar espaço de armazenamento em nodos intermediários. Ambos utilizam como principal informação para decisão sobre o roteamento o número de encontros em que os nodos estão envolvidos em um período de tempo (chamado de “potencial de entrega”). Assim, os nodos priorizam o repasse de pacotes a nodos com mais alto potencial, aumentando a probabilidade da mensagem ser entregue ao destino. Conforme Seção 3, o APRP-Ack tem melhores índices de desempenho que o APRP e, quando comparado a outros protocolos referenciados na literatura, tem uma maior taxa de entrega, gerando menos cópias de mensagens na rede. Entretanto, o atraso médio na entrega das mensagens é um pouco maior, já que o repasse não acontece imediatamente no próximo encontro com outro nodo, mas depende também do nodo encontrado ter maior potencial de entrega.

Neste artigo, é proposta uma nova estratégia de roteamento, chamada de APRP-Group, também baseada no potencial de entrega dos nodos. Esta estratégia visa diminuir o atraso médio na entrega, sem prejudicar a taxa de entrega das mensagens.

Conforme discutido na Seção 4, observa-se que dependendo da movimentação, indicada pela velocidade dos nodos, existem nodos com valores de potenciais de entrega agrupados. A estratégia de roteamento aqui proposta se utiliza dessa característica, agrupando logicamente nodos com potenciais de entrega semelhantes e definindo uma nova estratégia de repasse baseada nestes grupos. Além disso, com o objetivo de aumentar a disseminação de mensagens na rede, um novo mecanismo para administração de cópias de uma mensagem foi definido. O APRP-Group apresenta uma redução significativa no atraso (de até 31%) se comparado ao APRP-Ack, além de aumentar ou manter a taxa de entrega das mensagens. Já em comparação ao *Spray and Wait*, um dos principais protocolos para DTN apresentado na literatura, o APRP-Group desempenha de forma análoga em redes densas e melhor em redes esparsas. Em redes esparsas o APRP-Group apresenta

redução de *overhead*¹ (até 20%) e aumento na taxa de entrega (até 9%) ao custo de um aumento no atraso de entrega (até 22%).

Este artigo está organizado como segue. A Seção 2 descreve trabalhos relacionados. A Seção 3 descreve o protocolo APRP e o APRP-Ack. A Seção 4 apresenta o mecanismo utilizado para agrupar nodos de acordo com o potencial de entrega. A Seção 5 descreve o APRP-Group, apresentando a estratégia de repasses utilizada. A Seção 6 discute os resultados obtidos por simulação para dois e três tipos de nodos, apresentando as seguintes métricas: atraso na entrega das mensagens, probabilidade de mensagens entregues e número de cópias na rede. Por fim, a Seção 7 apresenta as conclusões e discute trabalhos futuros.

2. Trabalhos Relacionados

A literatura apresenta vários estudos que propõem mecanismos diferentes para realizar o roteamento em redes DTN. Um exemplo de protocolo é o PROPHET (*Probabilistic Routing Protocol using History of Encounters and Transitivity*) [Lindgren et al. 2003]. Esse protocolo usa uma métrica probabilística que indica a capacidade de um nodo entregar uma mensagem para um outro nodo específico. Tal métrica é chamada de previsibilidade de entrega.

Vários protocolos baseados no PROPHET podem ser encontrados na literatura. Um exemplo é o protocolo MV (*Meeting Visit*) [Burns et al. 2005], que tem o objetivo de aprender a frequência de encontros entre pares de nodos e suas visitas em regiões geográficas particulares. A história de contatos é também usada em [Oliveira and Albuquerque 2009].

O MaxProp [Burgess et al. 2006] estima a probabilidade de entrega, semelhante ao PROPHET, a qual é usada para indicar as mensagens que deveriam ser transmitidas primeiro. Diferentemente dos protocolos de roteamento mencionados acima, toda mensagem entregue para o destinatário é notificada com um *acknowledgement*, o qual é replicado para todos os nodos da rede. No *Predict and Relay* (PER) [Yuan et al. 2009], os nodos determinam a distribuição de probabilidade dos tempos de contato futuros e escolhem um próximo *hop* apropriado com o objetivo de melhorar a probabilidade de entrega.

Nos protocolos mencionados o custo de manutenção das informações topológicas em termos de comunicação e armazenamento é considerável, uma vez que os mesmos acumulam informações sobre os nodos da rede e rotas para alcançá-los. Por outro lado, existem abordagens mais simplistas que não armazenam esse tipo de informação e, conseqüentemente, realizam sua decisão de repasse com base em políticas de roteamento e informações locais aos nodos que se encontram. Dessa forma, nenhuma informação topológica é mantida nos nodos.

A estratégia de Transmissão Direta [Abdulla and Simon 2007] é considerada a mais simples e visa entregar as mensagens fazendo uma entrega direta ao destino. Além dessa, a estratégia de *flooding* também é usada, de uma forma restrita, por muitos protocolos. Um exemplo de protocolo que usa *flooding* é chamado de Epidêmico [Vahdat and Becker 2000]. A operação desse protocolo é semelhante a uma doença epidêmica: as mensagens são repassadas aos nodos encontrados.

¹Considera-se *overhead* neste texto o número de cópias geradas na rede para cada mensagem criada.

O *Spray and Wait* [Spyropoulos et al. 2008] propõe uma abordagem para controlar o *flooding* limitando o número de cópias da mensagem na rede. Para cada mensagem, um nodo pode copiá-la na rede C vezes, onde C é configurado. Assim que todas as cópias forem distribuídas, cada cópia pode ser entregue somente de forma direta ao destino. Uma variação do *Spray and Wait* é chamada de DASW (*Density-Aware Spray-and-Wait*) [Tournoux et al. 2009]. Esse protocolo tenta definir o número de cópias da mensagem que pode ser espalhada pela rede.

Um outro protocolo que usa informações locais para decidir sobre o repasse das mensagens é chamado de *Rapid* [Balasubramanian et al. 2007]. Este protocolo determina, a cada oportunidade de transferência, se a replicação da mensagem justifica o uso dos recursos. O *Rapid* controla o número de cópias de uma mensagem e dá uma prioridade mais alta para as mensagens que foram repassadas em menos tempo, removendo primeiro do *buffer* mensagens com prioridade mais baixa.

A única menção encontrada ao uso de agrupamento em redes DTN refere-se a identificação de grupos de nodos fisicamente próximos. Um protocolo de roteamento para DTN que usa a ideia de grupo é chamado de HYMAD [Whitbeck and Conan 2009]. De acordo com [Whitbeck and Conan 2009], HYMAD periodicamente procura mudanças na topologia da rede e constrói grupos disjuntos temporários de nodos conectados. O roteamento dentro de um grupo é manipulado por um protocolo de roteamento *ad hoc* convencional, enquanto a entrega entre grupos é manipulada por um protocolo para DTN. Embora HYMAD trabalhe com grupos de nodos, a noção de grupos e consequentemente o processo de identificação desses grupos é bastante diferente da estratégia apresentada neste artigo.

3. APRP and APRP-Ack

O protocolo APRP foi definido em [Nunes and Dotti 2009] e, como os citados na Seção 2, considera redes cuja mobilidade é desconhecida. Nesse protocolo cada nodo de uma rede DTN monitora o número de encontros com outros nodos em um período de tempo. Um encontro acontece toda vez que um nodo entra no seu alcance de comunicação. A sinalização de um encontro é possível através de tecnologias de nível 1 e 2 [Trainin et al.]. O valor do número de encontros foi chamado, em [Nunes and Dotti 2009], de “potencial de entrega”. A estratégia de repasse assume que o comportamento de encontros de um nodo está relacionado com a história recente do mesmo. Portanto, nodos com potencial de entrega mais alto tem uma probabilidade maior de encontrar o destino em um futuro próximo.

O cálculo do potencial de entrega é feito da seguinte forma: cada nodo registra o número de encontros que teve em uma determinada janela de tempo. Quando um nodo encontra um novo vizinho, ele registra o novo encontro e remove encontros antigos quando o tempo da janela foi ultrapassado.

A decisão de repasse leva em consideração o potencial dos nodos e o limite do número de cópias da mensagem na rede. Um nodo repassa uma mensagem para nodos com potencial de entrega maior do que um certo valor, de forma que os recursos usados na transferência sejam justificados. Assim, o repasse de uma mensagem entre dois nodos ocorre se

$$P_r > P_o + \delta(P_{max-o} - P_o),$$

onde o e r são, respectivamente, origem e destino do repasse; P_o é o potencial de entrega de o ; P_r é o potencial de entrega de r ; P_{max-o} é o potencial máximo conhecido por o ; e δ é um valor de ajuste para o diferencial de potencial desejado. O valor de P_{max-o} é local e depende dos encontros registrados por o .

Complementarmente, o APRP limita o número de cópias de uma mensagem. A estratégia usada é semelhante a do protocolo *Spray and Wait* [Spyropoulos et al. 2008] com *spraying* binário. A origem de uma mensagem inicia com um crédito de C cópias; quando o repasse de uma mensagem é feito, ela delega $\lfloor C/2 \rfloor$ dos créditos para o outro nodo e mantém $\lceil C/2 \rceil$ para si. Caso não haja espaço em *buffer* para armazenar a mensagem, a estratégia FIFO é utilizada para descarte.

O protocolo APRP-Ack [Nunes et al. 2009] é uma extensão que utiliza mensagens de *acknowledgement* para liberar espaço em *buffer* e melhorar a taxa de entrega. Cada vez que uma mensagem é entregue ao destino, um *ACK* é emitido através da rede. Um nodo, ao receber tal mensagem, remove a mensagem identificada de seu *buffer*.

Em comparação com os protocolos Epidêmico, PROPHET, and *Spray and Wait*, o APRP e o APRP-Ack provêm custos mais baixos tanto em termos de comunicação como armazenamento, mantendo ou causando aumento em taxas de entrega [Nunes and Dotti 2009, Nunes et al. 2009]. Este comportamento é especialmente observado em redes esparsas - de maior importância no caso de redes DTN - sendo que o APRP-Ack desempenha melhor que o APRP. Comparando o APRP-Ack ao *Spray and Wait*, em uma rede com raio de alcance de 30 metros, a probabilidade de entrega das mensagens aumenta de 2 a 6%, com uma redução de 6 a 58% no *overhead*. Contudo, o atraso médio na entrega das mensagens é mais elevado, uma vez que o nodo precisa encontrar outro com maior potencial de entrega para realizar o repasse.

4. Caracterização e Agrupamento de Potenciais

Em muitas situações os nodos podem ser agrupados de acordo com o seu potencial de entrega. Observou-se que grupos de nodos que se movem de forma semelhante tendem a valores de potenciais médios aproximados [Nunes and Dotti 2009]. Os nodos que possuem maior mobilidade em geral possuem um valor de potencial mais elevado, pois seu contato com outros nodos é maior. Já os nodos que se movimentam menos possuem esse valor reduzido.

A Figura 1a apresenta os potenciais para dois tipos de nodos quando seus intervalos de velocidade são aproximados. Neste caso, um tipo varia a velocidade entre [0.5,1.0] m/s, enquanto o outro varia entre [2.0,5.0] m/s. Já a Figura 1b apresenta os potenciais quando há uma maior diferença no intervalo das velocidades, com intervalos entre [0.5,1.0] m/s e [20.0,22.0] m/s². Observa-se que quanto maior é a diferença entre os intervalos de velocidades dos tipos de nodos, melhor é a separação entre grupos.

A divisão dos potenciais em grupos é usada com o intuito de reduzir o atraso gerado pelo APRP-Ack. Para tanto, o repasse é feito para qualquer nodo que esteja em um grupo de potencial superior, com exceção dos nodos que estão no grupo de maior potencial que repassam mensagens para nodos do próprio grupo.

²Os demais parâmetros utilizados nessas simulações são os mesmos apresentados na Seção 6.

A identificação de grupos é um processo distribuído, que deve rodar em cada nodo da rede. A cada encontro, os nodos trocam seus valores de potenciais, colecionando amostras <nodo, potencial>. Com o tempo, um nodo coleciona valores de potenciais de outros nodos (chamada de lista de contatos) e com essa coleção os grupos podem ser identificados. A identificação de grupos pode utilizar técnicas estatísticas, reconhecimento de padrões e mineração de dados [Berkhin 2002]. Este trabalho propõe o uso de uma abordagem que usa programação dinâmica, adaptada de [Oliveira 2008]. Tal abordagem minimiza tanto os desvios padrão de cada grupo como a diferença entre esses desvios. Para tanto, são analisadas as divisões em todos os pontos de corte possíveis para k grupos, como descrito abaixo.

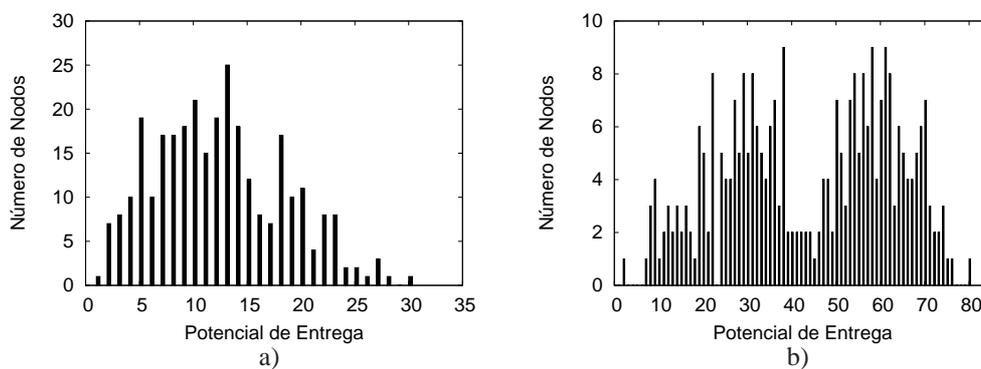


Figura 1. Distribuição de potenciais para dois tipos de nodos. a) Resultado da simulação com intervalo de velocidades aproximado. b) Resultado da simulação com maior diferença no intervalo de velocidades.

Assume-se um conjunto de potenciais e um valor inteiro k de grupos a serem formados. Para acelerar o processo, os valores dos desvios calculados vão sendo armazenados em uma matriz M , de forma que esses valores não precisem ser calculados novamente. Cada elemento $M_{n,k}$ representa o menor desvio padrão máximo de um grupo para todas as possíveis divisões de n valores em k grupos. Para cada desvio padrão armazenado em M , a posição de corte é armazenada em uma outra matriz (*break_list*). Portanto, a partir da *break_list* obtém-se as posições de corte considerando-se divisões de 1 a k grupos. Para o algoritmo utilizado no APRP-Group é passado um valor de k que corresponde à metade dos valores de potenciais que o nodo armazena, de forma que grupos pequenos, com menos de dois valores de potenciais, não sejam formados.

A escolha do melhor número de grupos é realizada da seguinte forma: realiza-se as somas S_{k-1}, S_k, S_{k+1} dos desvios padrão aos grupos escolhidos para os casos $k - 1, k$ e $k + 1$. Enquanto $S_{k-1}/S_k > S_k/S_{k+1}$, continua-se o processo, passando para $k + 1$. Isso significa que a diminuição dos desvios padrão dos grupos é significativa a cada passo. O processo termina quando o resultado da segunda divisão for menor que a primeira, indicando uma redução na diminuição dos desvios padrão dos grupos. Observa-se que esse mecanismo não escolhe a divisão em apenas um grupo, pois deseja-se que no mínimo dois grupos sejam formados.

Dada a amostra com uma distribuição hipotética de potenciais apresentada na Figura 2 à esquerda acima, observa-se que os dados separam-se naturalmente em 3 grupos, ou seja, o k ideal deveria ser 3. O cálculo usado na escolha de k está apresentado na

Figura 2. A tabela à esquerda apresenta os valores dos desvios padrão da distribuição hipotética para a separação em vários grupos, bem como o valor calculado como critério de parada. Já à direita da mesma figura está apresentado o gráfico com esses últimos valores, onde seleciona-se o valor mais alto. Neste caso, a escolha seria o valor correspondente a $k = 3$, indicando uma divisão em 3 grupos.

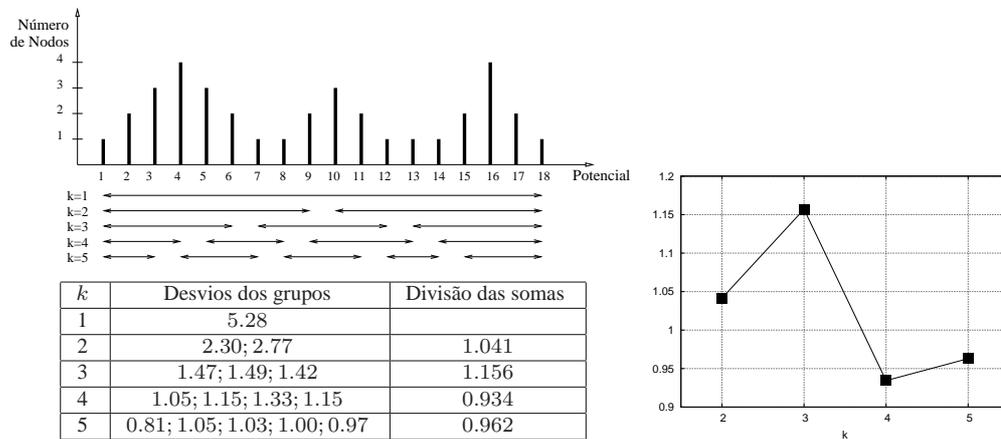


Figura 2. Distribuição hipotética de potenciais e valores dos desvios padrão se for efetuada a divisão em $k = 1, 2, 3, 4$ e 5 grupos.

5. APRP-Group

Conforme indicado na Seção 3, os protocolos APRP e APRP-Ack geram um atraso médio na entrega das mensagens superior a outros protocolos referenciados na literatura. Há duas hipóteses para esse atraso ocorrer:

- i. em qualquer nodo, as mensagens ficam retidas até que se encontre um nodo de maior potencial;
- ii. em decorrência de (i), à medida que o potencial do nodo cresce, decresce a probabilidade de encontrar outro nodo de maior potencial, aumentando o atraso.

Para contornar a primeira hipótese, sugere-se realizar uma divisão nos créditos das mensagens, ou seja, dedicar parte dos créditos para tentar um repasse com baixo atraso, independente do potencial, e outra parte segundo a estratégia do potencial. Já para contornar a segunda hipótese, sugere-se utilizar a técnica de repasse baseada em grupos.

Grupos de Potenciais Na estratégia baseada em grupos, o repasse de uma mensagem ocorre quando ela sair do grupo atual e atingir um grupo de potencial superior. Para nodos do grupo de maior potencial, é permitida a troca de mensagens dentro do mesmo grupo, já que não existe um grupo superior para repassar a mensagem. Essa estratégia evita que as mensagens fiquem retidas durante muito tempo em um único nodo com alto potencial de entrega. Os resultados obtidos com a utilização dessa estratégia estão apresentados na Seção 6, com a identificação APRP-Group1.

Créditos das Mensagens Os créditos das mensagens são divididos em tipos diferentes para alcançar uma maior redução no atraso médio na entrega das mesmas, sem que haja redução na taxa de entrega. Quando uma mensagem é criada, um número C de créditos é configurado, o qual é dividido em CP (Créditos de Potencial) e CA (Créditos de Atraso). Os créditos serão decrementados sempre que ocorrer o repasse de uma mensagem, sendo

que apenas créditos de um tipo podem ser utilizados a cada repasse, conforme detalhado no Algoritmo 1.

5.1. Roteamento no APRP-Group

A estratégia de roteamento definida pelo APRP-Group se utiliza dos dois mecanismos descritos anteriormente. O Algoritmo 1 apresenta essa estratégia. Sempre que uma mensagem é criada, os créditos de entrega (CP) e de atraso (CA) são configurados. O $STATUS$ é configurado com “ BUF ”, indicando que a mensagem está armazenada no *buffer* para ser entregue. As informações resumidas das mensagens criadas pelos nodos são mantidas em um *summary vector* no seguinte formato: $\langle m, CP, CA, STATUS \rangle$.

Ao encontrar um nodo n , $nodeMeetingNeighbor()$ é executado. A identificação, o potencial, o *summary vector* e a lista de contatos do nodo encontrado são obtidos por $obtain(n, P_n, summary_vector_n, contact_list_n, time)$. O nodo encontrado é adicionado em $contact_list_p$ e em $contact_list_g$ e os contatos antigos são removidos de acordo com a respectiva janela de tempo. Além disso, adiciona-se na $contact_list_g$ do nodo os contatos da $contact_list_n$ que sejam válidos na janela. A troca das listas de contatos entre os nodos ocorre para que haja uma convergência mais rápida no processo de identificação de grupos. A sincronização da lista de mensagens entregues ao destino final, bem como a remoção das mesmas do *buffer* do nodo é realizado por $updateListAckedIds(summary_vector_n, summary_vector_o)$. As mensagens entregues possuem o $STATUS$ “ ACK ” no *summary vector*.

O agrupamento é realizado por $clusterKGroups(contact_list_g, k, break_list)$, que retorna o número de grupos ideal e a matriz com as posições de corte dos grupos. O potencial de entrega do nodo origem é calculado por $calculateP_o(contact_list_p, P_o)$ e o potencial de seu grupo é calculado por $calculateGroup(P, P_g, k, break_list)$. Caso haja mensagens para serem entregues, o procedimento $directDelivery(summary_vector_o, n, summary_vector_n)$ identifica se o nodo encontrado é destino de alguma mensagem presente em sua lista para que seja feita a entrega direta, a remoção da mesma e a alteração de seu $STATUS$ no *summary vector*. A entrega somente será realizada caso o nodo encontrado ainda não possua a mensagem a ser entregue. Além disso, calcula-se o potencial do grupo do nodo encontrado. Se esse potencial for maior que o do grupo do nodo origem ou ambos estiverem no grupo de maior potencial, é possível repassar as mensagens usando um dos dois tipos de créditos configurados. Caso contrário, somente os créditos de atraso são usados. O repasse somente será realizado caso o nodo encontrado ainda não possua a mensagem a ser entregue. Ao repassar uma mensagem em $sendMessageToNeighbor(summary_vector_o, n, m, Cx, type)$, $\lfloor C/2 \rfloor$ dos créditos fica com o nodo que recebe a mensagem e $\lceil C/2 \rceil$ fica com o nodo que está transmitindo a mensagem, semelhante ao APRP e APRP-Ack. O tipo do crédito é transmitido juntamente com a mensagem e somente os créditos do tipo repassado são decrementados.

O procedimento $receiveMessage(m, C, type)$ é executado sempre que uma mensagem for recebida. Se houver espaço em *buffer*, o nodo armazena a mensagem e inclui a identificação, os créditos, bem como o $STATUS$ da mensagem no *summary vector* do nodo. Sempre que o crédito de um tipo for recebido, o crédito de outro tipo pode ser utilizado apenas para entrega direta da mensagem.

Algoritmo 1 Mecanismo de Roteamento do APRP-Group.

```

// Estruturas mantidas no nodo:
o: identificação do próprio nodo;  $P_o$ : potencial de entrega do próprio nodo;  $P_{go}$ : potencial de grupo do próprio nodo;
k: número ideal de grupos; break_list: matriz com as posições de corte dos grupos;
contact_list_g: lista de contatos para agrupamento, no formato  $\langle n, P_n, time \rangle$ ;
contact_list_p: lista de contatos que informa quem são os vizinhos do próprio nodo, no formato  $\langle n, P_n, time \rangle$ ;
summary_vector_o: lista com a identificação das mensagens e o número de créditos que estão sendo armazenadas pelo próprio
nodo, no formato  $\langle m, C \rangle$ ;
 $W_p$ : tamanho da janela de tempo usada na atualização da contact_list_p;
 $W_g$ : tamanho da janela de tempo usada na atualização da contact_list_g.

// Variáveis:
n: identificação do nodo encontrado;  $P_n$ : potencial de entrega do nodo encontrado;  $P_{gn}$ : potencial de grupo do nodo encontrado;
time: tempo em que houve o contato com o nodo encontrado;
contact_list_n: lista de contatos do nodo encontrado, no formato  $\langle n, P_n, time \rangle$ ;
m: identificação da mensagem a ser repassada ou entregue ao destino final;
summary_vector_n: lista com a identificação das mensagens que estão sendo armazenadas pelo nodo encontrado;
C: número de créditos que uma mensagem possui; CP: créditos de entrega; CA: créditos de atraso;
STATUS: indica se a mensagem foi entregue ("ACK") ou se ainda está em buffer ("BUF");
type: tipo de crédito.

procedure nodeMeetingNeighbor()
  obtain( $n, P_n, summary\_vector\_n, contact\_list\_n, time$ )
  updateContactList(contact_list_p, contact_list_g, contact_list_n,  $n, P_n, time, W_p, W_g$ )
  updateListAckedIds(summary_vector_n, summary_vector_o)
  clusterKGroups(contact_list_g, k, break_list)
  calculate $P_o$ (contact_list_p,  $P_o$ )
  calculateGroup( $P_o, P_{go}, k, break\_list$ )
  if (nodeHasMessage(summary_vector_o)) then
    directDelivery(summary_vector_o,  $n, summary\_vector\_n$ )
    calculateGroup( $P_n, P_{gn}, k, break\_list$ )
    if ( $(P_{gn} > P_{go}) \vee ((P_{gn} == P_{go}) \wedge (isGreaterGroup(P_{go}, k)))$ ) then
      for all  $\langle m, CP, CA \rangle$  in summary_vector_o do
        if ( $(CP > 1) \wedge (!neighborHasMessage(m, summary\_vector\_n))$ ) then
          sendMessageToNeighbor(summary_vector_o,  $n, m, CP, "CP"$ )
        else
          if ( $CA > 1$ ) then
            sendMessageToNeighbor(summary_vector_o,  $n, m, CA, "CA"$ )
          end if
        end if
      end for
    else
      for all  $\langle m, CP, CA \rangle$  in summary_vector_o do
        if ( $(CA > 1) \wedge (!neighborHasMessage(m, summary\_vector\_n))$ ) then
          sendMessageToNeighbor(summary_vector_o,  $n, m, CA, "CA"$ )
        end if
      end for
    end if
  end if

procedure directDelivery(summary_vector_o,  $n, summary\_vector\_n$ )
  for all m in summary_vector_o do
    if ( $(destination(m) = n) \wedge (!neighborHasMessage(m, summary\_vector\_n))$ ) then
      sendDirectMessage( $m, n$ )
      removeMessageAndUpdateList(summary_vector_o,  $m, "ACK"$ )
    end if
  end for

procedure sendMessageToNeighbor(summary_vector_o,  $n, m, Cx, type$ )
  sendMessage( $n, m, \lfloor Cx/2 \rfloor, type$ )
  updateMsgList(summary_vector_o,  $m, \lceil Cx/2 \rceil$ )

procedure receiveMessage( $m, C, type$ )
  makeRoomForNewMessage( $m$ )
  if ( $type = "CP"$ ) then
    storeInBuffer(summary_vector_o,  $m, C, 1, "BUF"$ )
  else
    if ( $type = "CA"$ ) then
      storeInBuffer(summary_vector_o,  $m, 1, C, "BUF"$ )
    end if
  end if

```

Algoritmo 2 Mecanismo de Roteamento do APRP-Group - Descrição dos procedimentos.

procedure *calculateGroup*($P, P_g, k, break_list$) // Calcula o potencial do grupo do nodo passado por parâmetro.
procedure *calculateP_o*($contact_list_p, P_o$) // Calcula o potencial do nodo origem.
procedure *clusterKGroups*($contact_list_g, k, break_list$) // Gera a distribuição de frequência dos potenciais, calcula o número de grupos ideal e a lista com as posições das divisões dos grupos.
procedure *createNewMessage*($m, summary_vector_o, C$) // Inclui a identificação de nova mensagem na lista de mensagens, configura os tipos de créditos da mensagem ($CP = \lceil C/2 \rceil$ e $CA = \lfloor C/2 \rfloor + 1$), bem como seu *STATUS*.
procedure *isGreaterGroup*(P_{go}, k) // Verifica se potencial do grupo pertence ao maior grupo encontrado.
procedure *neighborHasMessage*($m, summary_vector_n$) // Verifica se a mensagem m está presente no *summary vector* do nodo encontrado. Caso esteja retorna *true*, caso contrário retorna *false*.
procedure *nodeHasMessage*($summary_vector_o$) // Verifica se há mensagens na lista, retornando *true* ou *false*.
procedure *obtain*($n, P_n, summary_vector_n, contact_list_n, time$) // Retorna a identificação, o potencial e o *summary vector* do nodo encontrado, bem como o tempo do encontro.
procedure *removeMessageAndUpdateList*($summary_vector_o, m, "ACK"$) // Remove do *buffer* a mensagem entregue ao destino e altera o *STATUS* dessa mensagem para "ACK" no *summary vector* do próprio nodo.
procedure *sendDirectMessage*(m, n) // Envia mensagem para o destino.
procedure *sendMessage*($n, m, C, type$) // Repassa mensagem para o nodo.
procedure *storeInBuffer*($summary_vector_o, m, CP, CA, STATUS$) // Armazena mensagem recebida na lista de mensagens.
procedure *updateContactList*($contact_list_p, contact_list_g, contact_list_n, n, P_n, time, W_p, W_g$) // Inclui a identificação do nodo encontrado, potencial do nodo e tempo de contato em *contact_list_p* e *contact_list_g*. Remove contatos antigos de acordo com as janelas de tempo (W_p e W_g). Adiciona em *contact_list_g* os contatos do nodo encontrado, válidos na janela de tempo.
procedure *updateListAckedIds*($summary_vector_n, summary_vector_o$) // Atualiza o *summary vector* do próprio nodo com as mensagens confirmadas que estão no *summary vector* do nodo encontrado, além de remover do *buffer* as mensagens confirmadas que foram atualizadas.
procedure *updateMsgList*($summary_vector_o, m, C$) // Atualiza a lista de mensagens dando o novo crédito para a mensagem repassada.

6. Resultados do APRP-Group

A avaliação do APRP-Group foi realizada através do simulador ONE (*Opportunistic Network Environment*) [Keränen and Ott 2007], um ambiente de simulação de eventos discretos específico para DTN, sendo utilizado pela comunidade acadêmica [Lee et al. 2009], [Polat et al. 2009]. Devido aos resultados comentados na Seção 3 sobre o desempenho do APRP-Ack, este artigo apresenta a comparação do APRP-Group com APRP-Ack e *Spray and Wait*. Os resultados apresentados do APRP-Group estão divididos em duas partes. A primeira, chamada de APRP-Group1, que utiliza apenas a estratégia de repasses usando grupos, enquanto a segunda, chamada de APRP-Group2, que utiliza a estratégia completa descrita na Seção 5.1.

A janela de tempo para cálculo do potencial (W_p) foi configurada em 240 s, pois esse tempo é suficiente para caracterizar diferentes valores de potenciais para diferentes densidades [Nunes and Dotti 2009]. A janela de tempo para atualização da lista de contatos (W_g) usada no agrupamento foi configurada com $4 \times W_p$, pois conforme testes realizados, com esse tempo obtém-se convergência na identificação distribuída dos grupos (em cada nodo). Um período de transiência de 1000 s foi utilizado. Durante esse período os nodos registram encontros com outros nodos, mas não é gerado tráfego. De acordo com a janela de tempo W_g , na primeira troca de mensagens o agrupamento terá sido realizado uma vez. O número de créditos (C) foi configurado nos protocolos APRP-Group, APRP-Ack e *Spray and Wait* em 6 por mensagem, conforme [Keränen and Ott 2007], permitindo maior comparação de resultados [Spyropoulos et al. 2008]. Experimentos com diferentes valores de δ foram realizados [Nunes and Dotti 2009] e o valor de 0.2 forneceu melhor relação entre as métricas analisadas.

A área simulada tem um tamanho de 4500×3400 metros. O número de nodos varia entre 30, 50, 100 e 300 nodos. Os nodos se movimentam de acordo com três padrões

de mobilidade: *Random Waypoint* [Bettstetter et al. 2003], *Shortest Path Map Based Movement* [Keränen and Ott 2007] e *Map Based Movement* [Keränen and Ott 2007]. O raio de alcance utilizado varia em 10, 30 e 80 metros.

Experimentos com dois e três tipos de nodos foram realizados. Nos experimentos com dois tipos de nodos, metade deles representa automóveis e a outra metade representa pessoas. Os automóveis diferem das pessoas em termos de velocidade e tamanho de *buffer*. A velocidade das pessoas varia entre [0.5, 1.0] m/s e a velocidade dos automóveis varia entre [2.7, 13.9] m/s. Quanto ao tamanho dos *buffers*, as pessoas conseguem armazenar 20Mbytes, enquanto os automóveis conseguem armazenar 40Mbytes. Já nos experimentos com três tipos de nodos, além dos dois tipos mencionados foi incluído um terceiro tipo com velocidades entre [18, 25] m/s, com capacidade de armazenamento de até 50Mbytes. Esse terceiro tipo foi incluído de forma a analisar o desempenho da proposta baseada em grupos. Além disso, um número total de 5000 mensagens de 100Kbytes a 2Mbytes são enviadas, sem considerar TTL (*Time to Live*). O momento do envio de cada mensagem, bem como a origem e o destino das mesmas foram escolhidos aleatoriamente.

Os experimentos rodam por aproximadamente 55 horas simuladas, tendo sido ajustado para atingir a distribuição espacial de nodo compatível com a obtida através de métodos analíticos em [Delamare et al. 2006], considerando os mesmos parâmetros de mobilidade. Para cada cenário simulado, foram realizadas sete rodadas de simulação. Os resultados apresentados nesta seção foram obtidos pela média de todas as rodadas. Para cada valor apresentado, as barras verticais associadas representam um intervalo de confiança de 95%. Algumas vezes o intervalo de confiança está bastante próximo do valor plotado, sendo difícil sua visualização nas figuras.

A seguir estão apresentados os resultados para três métricas: atraso médio na entrega das mensagens, percentual de mensagens entregues e *overhead*. São apresentados resultados para dois e três tipos de nodos, com raio de alcance de 30 metros e com padrão *Map Based Movement*. Os resultados com outros raios de alcance, bem como outros padrões de mobilidade foram omitidos por restrições de espaço e porque eles não trazem informações adicionais para entender o comportamento do protocolo proposto.

6.1. Atraso

A Figura 3 apresenta o atraso médio na entrega das mensagens. A partir dessa figura pode-se observar que o APRP-Group1 reduz o atraso em relação ao APRP-Ack, contudo uma redução mais significativa ocorre com o APRP-Group2 que se aproxima do *Spray and Wait*. Essa redução ocorre tanto para o cenário com dois tipos de nodos (Figura 3a) quanto para o cenário com três tipos de nodos (Figura 3b), ficando mais evidenciada nos cenários esparsos. Isto confirma as hipóteses de atraso tal como colocadas na Seção 5.

6.2. Percentual de Mensagens Entregues

A Figura 4 apresenta o percentual de mensagens entregues. Observa-se que o percentual de entrega, tanto do APRP-Group1 quanto do APRP-Group2, se mantém similar ou ligeiramente superior ao APRP-Ack na maioria dos casos, sendo que o APRP-Group2 é melhor em cenários esparsos. Esse comportamento é devido ao APRP-Group1 e APRP-Group2 possibilitarem em uma maior disseminação de mensagens na rede, conforme Figura 5 e ainda realizar o repasse a nodos de potencial crescente. Contudo, conforme a rede se torna mais densa, seja por um número maior de nodos ou devido a um maior raio

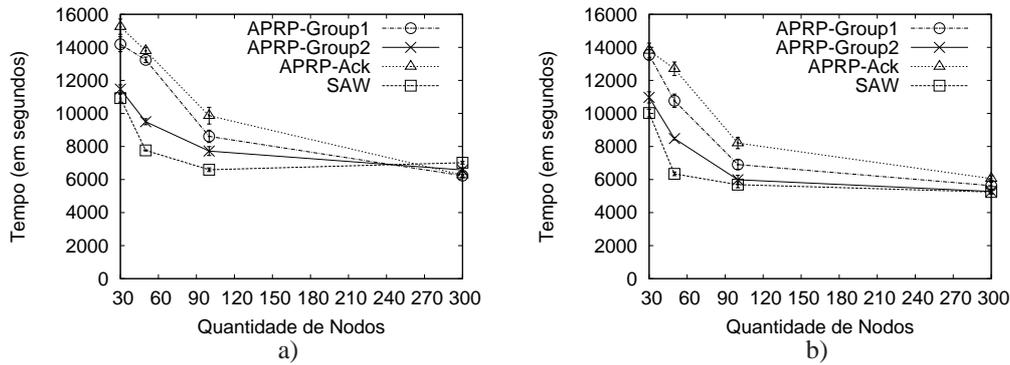


Figura 3. Atraso na entrega das mensagens. a) Dois tipos de nodos. b) Três tipos de nodos.

de alcance na comunicação, a probabilidade de encontrar o destino é mais alta e isso faz com que haja um maior percentual de entrega em todos os protocolos.

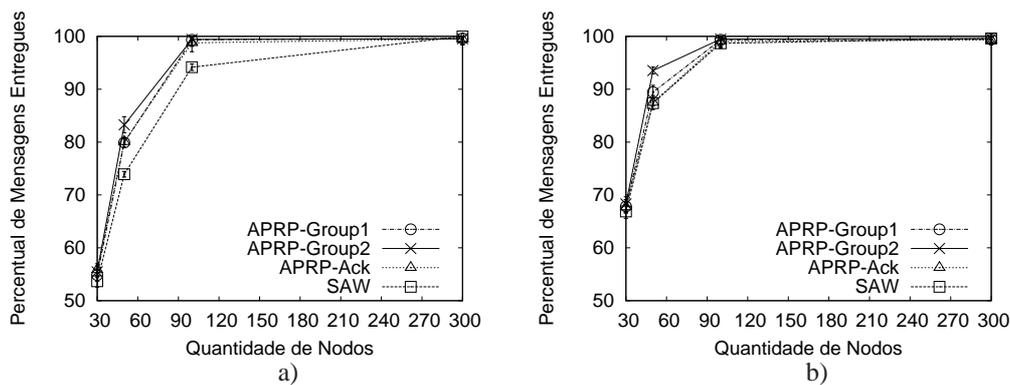


Figura 4. Percentual de entrega das mensagens. a) Dois tipos de nodos. b) Três tipos de nodos.

6.3. Overhead

O *overhead* está apresentado na Figura 5. Apesar do *overhead* gerado tanto pelo APRP-Group1 quanto pelo APRP-Group2 ter aumentado em relação ao APRP-Ack, ele continua menor que no *Spray and Wait*. O aumento do número de cópias acontece no APRP-Group1 pois os nodos de maior potencial tem a possibilidade de repassar as mensagens para nodos de mesmo grupo, não tendo que ficar a espera de um nodo de maior potencial como no APRP-Ack. Já no APRP-Group2, uma parte das mensagens pode ser repassada a nodos de menor potencial. Apesar disso, o *overhead* se mantém muito baixo, não ocasionando sobrecarga nos recursos da rede e nodos.

7. Conclusões

Neste artigo foi proposto um novo protocolo de roteamento para redes DTN que agrupa nodos com potencial de entrega semelhantes, além de utilizar tipos de créditos para decidir sobre o repasse das mensagens. Tal protocolo evita, dessa forma, que mensagens fiquem retidas por muito tempo nos nodos. Experimentos foram realizados com dois e três tipos de nodos e observou-se, em relação ao APRP-Ack: uma melhoria significativa no atraso; e a manutenção ou aumento no percentual de mensagens entregues, com um aumento no

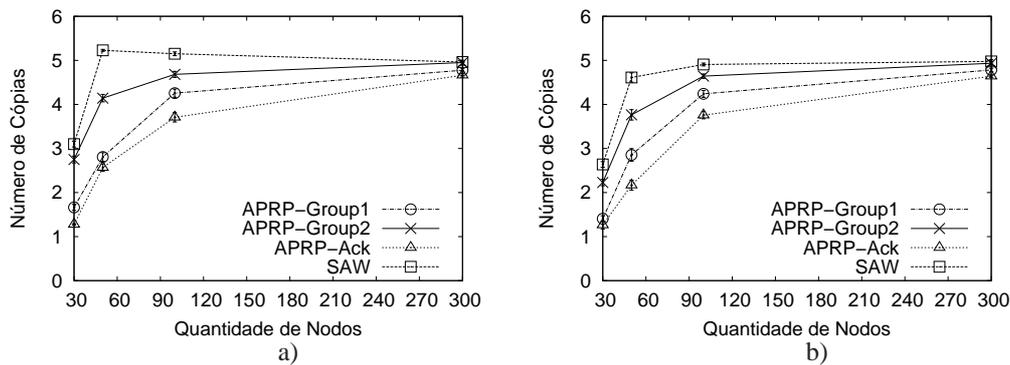


Figura 5. **Overhead.** a) Dois tipos de nodos. b) Três tipos de nodos.

overhead. Desta forma, os objetivos do APRP-Group foram atingidos. Já em comparação ao *Spray and Wait*, o APRP-Group apresenta redução de até 20% no *overhead*, aumento de até 9% no percentual de entrega e de até 22% no atraso na entrega das mensagens.

Referências

- Abdulla, M. and Simon, R. (2007). The impact of the mobility model on delay tolerant networking performance analysis. In *40th Annual Simulation Symposium*, pages 177–184.
- Balasubramanian, A., Levine, B., and Venkataramani, A. (2007). DTN routing as a resource allocation problem. *SIGCOMM Computer Communication Review*, 37(4):373–384.
- Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA.
- Bettstetter, C., Resta, G., and Santi, P. (2003). The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269.
- Bui-Xuan, B., Ferreira, A., and Jarry, A. (2003). Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267–285.
- Burgess, J., Gallagher, B., Jensen, D., and Levine, B. N. (2006). MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *IEEE Infocom*, pages 1–11.
- Burns, B., Brock, O., and Levine, B. N. (2005). MV routing and capacity building in disruption tolerant networks. In *IEEE Infocom*, pages 398–408.
- Delamare, F., Dotti, F. L., Fernandes, P., Nunes, C. M., and Ost, L. C. (2006). Analytical modeling of random waypoint mobility patterns. In *Third ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pages 106–113.
- Jain, S., Fall, K., and Patra, R. (2004). Routing in a delay tolerant network. In *ACM SIGCOMM*, pages 145–158.
- Keränen, A. and Ott, J. (2007). Increasing reality for DTN protocol simulations. Technical report, Helsinki University of Technology.

- Lee, C., Chang, D., Shim, Y., Choi, N., Kwon, T., and Choi, Y. (2009). Regional token based routing for DTNs. In *The International Conference on Information Networkin*, pages 1–5.
- Lindgren, A., Doria, A., and Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20.
- Liu, C. and Kaiser, J. (2005). A survey of mobile ad hoc network routing protocols. Technical report, University of Magdeburg.
- Nunes, C. M. and Dotti, F. L. (2009). Uma nova estratégia de roteamento para redes tolerantes a atrasos. In *XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 757–770.
- Nunes, C. M., Link, E., and Dotti, F. L. (2009). Evaluating the impact of an acknowledgment strategy for APRP. In *5th International Latin American Networking Conference*, pages 77–86.
- Oliveira, E. C. R. d. and Albuquerque, C. V. N. (2009). NECTAR: A DTN routing protocol based on neighborhood contact history. In *24th Annual ACM Symposium on Applied Computing*, pages 40–46.
- Oliveira, J. B. S. d. (2008). Two algorithms for automatic document page layout. In *ACM Symposium on Document Engineering*, pages 141–149.
- Polat, B. K., Khan, M., Tuli, N., and Kailay, P. (2009). ZBMF: Zone based message ferrying for disruption tolerant networks. *Communications and Networking in China*, 26:48–59.
- Spyropoulos, T., Psounis, K., and Raghavendra, C. (2008). Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90.
- Tournoux, P. U., Leguay, J., Benbadis, F., Conan, V., de Amorim, M. D., and Whitbeck, J. (2009). The accordion phenomenon: Analysis, characterization, and impact on DTN routing. In *IEEE Infocom*, pages 1116–1124.
- Trainin, S., Cordeiro, C., and (2009), A. K. Arrangements for association and re-association in a wireless network. <http://www.wipo.int/pctdb/en/wo.jsp?wo=2009114621>.
- Vahdat, A. and Becker, D. (2000). Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University.
- Whitbeck, J. and Conan, V. (2009). HYMAD: Hybrid dtn-manet routing for dense and highly dynamic wireless networks. In *Third International IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications*, pages 1–7.
- Yuan, Q., Cardei, I., and Wu, J. (2009). Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks. In *10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.
- Zhang, Z. (2006). Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys & Tutorials*, 8(1):24–37.