Introduzindo Tolerância a Interrupção em Redes Ad Hoc Móveis para Cenários de Emergência*

Vinícius F. S. Mota[1], Thiago H. Silva[1], José Marcos Silva Nogueira[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brazil

{vfsmota,thiagohs,jmarcos}@dcc.ufmg.br

Abstract. *In critical and emergency scenarios, such as disasters, first-responders can building mobile ad hoc networks addressing the lack of network infrastructure. Perhaps, communication in such scenarios may become susceptible to long interruptions. The Delay/Disruption Tolerant Networks are a proposal approach when communication is intermittent. In this paper, we develop a group mobility model for such scenarios and propose an algorithm based on hierarchical groups called ARGH. We compare ARGH to others similar protocols and noticed that it is scalable and has a higher rate of delivery that Epidemic and Prophet protocols when messages storage buffer is limited, like in wireless sensor Networks.*

Resumo. *Em cenários críticos e de emergência, tais como em desastres, equipes de resgate podem formar redes ad hoc móveis para a carência de infra-estrutura de rede. Porém, a comunicação nesses cenários pode se tornar suscetível a longas interrupções. As redes tolerantes a atrasos e interrupções (Delay/Disruption Tolerant Network - DTN) são uma abordagem proposta para situações em que a comunicação é intermitente. Neste trabalho, desenvolvemos um modelo de mobilidade em grupo para tais cenários e propomos um algoritmo hierárquico de roteamento baseado em grupos denominado ARGH. Comparamos o ARGH com outros protocolos de roteamento semelhantes e observamos que o ARGH é escalável e tem melhores taxas de entrega, comparado ao algoritmo Epidêmico e ao Prophet, quando o buffer de armazenamento de mensagens é limitado, situações que ocorrem normalmente em redes de sensores sem fios.*

1. Introdução

Em cenários de emergência, como os desastres naturais, tecnológicos ou causados pelo homem, equipes de busca e resgate da região afetada podem utilizar soluções de redes ad hoc móveis (*Mobile Ad Hoc Networks* - MANETs) para suprir eventuais carências de infra-estrutura de rede de comunicação. MANETs são redes autônomas e auto-organizáveis, cujos nós podem se mover aleatoriamente e auto organizarem suas tabelas de roteamento [man 2008].

Nesses cenários de emergência, devido a fatores como obstáculos e interferências, a conectividade fim-a-fim é altamente suscetível a interrupções. Isto faz com que sejam

*O presente trabalho foi realizado com apoio da CAPES, CNPq e FAPEMIG. Entidades do Governo Brasileiro e do Governo do Estado de Minas Gerais voltadas ao desenvolvimento científico e tecnológico.

ineficazes os protocolos de roteamento nas MANETs atuais, que necessitam estabelecer um caminho fim-a-fim para comunicação entre os nós, como por exemplo os protocolos AODV e DSR [Jain et al. 2004].

Redes de emergência são aquelas construídas sobre cenários de desastres e têm propriedades tais como comunicação robusta e resiliente, não são necessariamente infraestruturadas e principalmente oferece comunicação de dados e não somente voz (como ocorre com rádios atuais) [Rao et al. 2007].

Tais propriedades podem ser satisfeitas usando uma rede de comunicação tolerante a atrasos e interrupções (*Delay/Disruption Tolerant Network* - DTN). Uma DTN é um tipo de rede adequada para suportar longos atrasos, como em redes de satélites, e interrupções de comunicação, como em redes de sensores sem fios. Para possibilitar essa característica, foi adicionada uma nova camada na arquitetura, chamada *bundle*, entre as camadas de aplicação e transporte do modelo de camadas OSI [dtn 2008]. Essa camada é responsável por armazenar os pacotes (*bundles*) caso não seja possível encaminhá-los ao destino.

Uma DTN pode ter melhor alcançabilidade em redes com nós esparsos por utilizar políticas de comunicação assíncrona. Dessa forma, não há necessidade de estabelecimento de caminhos fim-a-fim, já que os *bundles* podem ficar armazenados nos nós até que haja uma conexão. Isso permite que as mensagens tenham atrasos longos e tolerância à altas taxas de erros [dtn 2008].

Os protocolos de roteamento em DTNs diferem dos protocolos equivalentes de MANETs tradicionais em suas capacidades de tolerar interrupções de comunicação. Jain e Patra formulam o problema de roteamento em DTNs em termos de um multi-grafo dirigido, no qual uma ou mais arestas podem existir entre um par de nós [Jain et al. 2004]. Contudo, o grafo que representa a topologia da rede pode ser um grafo desconexo formado por diversos componentes conexos.

O modelo de mobilidade comumente utilizado nas análises de protocolos de roteamento em DTN é o *Random Way Point*, no qual os nós se movimentam aleatoriamente em uma determinada área [Camp et al. 2002]. Este modelo por ser totalmente aleatório não representa o comportamento de um usuário normal.

O modelo de mobilidade *Reference Point Group* (RPGM) representa o movimento aleatório de grupos de nós, sendo utilizado para simular cenários de emergência como campos de batalha e catástrofes [Camp et al. 2002]. No RPGM, os movimentos dos nós de um grupo são baseados no caminho percorrido por um centro lógico do grupo. A fim de caracterizar melhor tais cenários, introduzimos no modelo RPGM a possibilidade de representação de regiões de interesse. Num determinado momento cada grupo de nós tem uma probabilidade ρ de ir para uma determinada região de interesse. Chamaremos este modelo de Modelo de Mobilidade de Emergência (MME). Explorando esse comportamento de mobilidade em grupo propomos um protocolo de roteamento hierárquico baseado em grupos, o qual batizamos de Algoritmo de Roteamento em Grupos Hierárquicos (ARGH).

A principal justificativa para este trabalho é o fato dos algoritmos tradicionais de roteamento epidêmicos inundarem a rede com replicações de mensagens, o que é uma característica indesejável pois sobrecarrega a rede. Portanto, reduzir o número de mensagens transmitidas na rede torna-se um desafio.

O foco do nosso trabalho é prover um protocolo de roteamento tolerante a interrupção

de comunicação. Para validarmos nossa solução utilizamos um cenário de desastre simplificado e aplicamos o MME. A contribuição do nosso trabalho é sumarizada a seguir:

- Proposição de uma extensão do modelo de mobilidade RPGM, acrescentando características que melhor representem cenários de emergência.
- Criação de um novo algoritmo hierárquico que considera o agrupamento dos nós para fazer o roteamento eficiente e escalável de mensagens.
- Avalição comparativa do desempenho do nosso algoritmo em relação ao algoritmos de roteamento em DTNs Epidêmico e ao Prophet.

Este trabalho está organizado como se segue: na Seção 2, apresentamos uma revisão da literatura sobre os algoritmos de roteamento em DTNs e os algoritmos de criação e manuntenção de agrupamentos. Nosso modelo de mobilidade é apresentado na Seção 3. A Seção 4 apresenta uma política de criação e manutenção de grupos (*clustering*) e o algoritmo de roteamento hierárquico desenvolvido. Na Seção 5 fazemos uma análise do desempenho e escalabilidade do algoritmo proposto. Por fim, concluímos e apresentamos propostas para trabalhos futuros na Seção 6.

2. Trabalhos Relacionados

2.1. Roteamento em Redes Tolerantes a Interrupção

O roteamento em DTNs criam novos desafios se comparado ao roteamento em redes tradicionais, visto que há incertezas sobre o tempo de duração da conectividade entre os nós. Os diversos protocolos de roteamento em DTNs diferem no conhecimento que os nós têm sobre a rede. Alguns assumem que os nós não têm nenhum conhecimento sobre o estado da rede (conhecidos como protocolos estocásticos). Outros assumem que os nós possuem informações tais como topologia da rede, tempo médio entre encontros sucessivos de dois nós e estimação do congestionamento dos nós (chamados de protocolos determinísticos). Revisões completas desses protocolos são apresentadas em [Jain et al. 2004] e [Zhang 2006].

Os algoritmos determinísticos baseiam-se nas informações que um nó tem sobre a rede, sendo utilizados quando é possível inferir quando haverá conectividade entre os nós [Handorean et al. 2004], [Liu and Wu 2007] e [Jain et al. 2004]. Em todas essas abordagens determinísticas, o caminho fim-a-fim é estabelecido antes do envio das mensagens, sendo dependente do momento em que foi prevista a possibilidade de conexão. No entanto, na maioria dos casos de redes *ad hoc* móveis não é possível prever a movimentação dos nós da rede.

Os algoritmos estocásticos são aplicáveis quando a rede tem um comportamento aleatório e pouco pode ser inferido sobre posições futuras dos nós. Esses protocolos variam desde o simples repasse da mensagem para todos os nós que se conseguir estabelecer contatos até a decisões baseadas no histórico, padrões de mobilidade ou outras informações [Vahdat and Becker 2000], [Lindgren et al. 2003], [Grossglauser and Tse 2002] e [Spyropoulos et al. 2005].

Em [Vahdat and Becker 2000] foi proposto um protocolo de roteamento estocástico Epidêmico para DTNs, no qual o nó origem difunde a mensagem para todos os seus vizinhos e cada um destes por sua vez repassa a mensagem para seus vizinhos. Esse ciclo

se repete até que a mensagem atinja o destino ou enquanto durar o tempo de vida (*time to live*/TTL) da mensagem, quando este for especificado. Desta forma, a mensagem é rapidamente distribuída em todos os nós alcançáveis. Com a mobilidade dos nós, espera-se que a mensagem também atinja partes da rede que não estavam acessíveis (outras sub-redes). Algoritmos epidêmicos se mostraram eficientes na entrega da mensagem, porém o número de mensagens extras enviadas pela rede (*overhead*) aumenta proporcionalmente à quantidade de nós.

Buscando diminuir esse *overhead*, propomos um protocolo (ARGH) que utiliza o agrupamento de nós para fazer o roteamento eficiente das mensagens. Uma mensagem só é encaminhada a um nó especial de um grupo e este nó é responsável por encaminhar a mensagem ao destino ou a um grupo diferente. Portanto, as mensagens não são replicadas entre nós de um mesmo grupo.

Um protocolo de roteamento probabilístico chamado Prophet (*Probabilist Routing Protocol using History of Encounters and Transitivity*) é proposto em [Lindgren et al. 2003]. O Prophet estima uma métrica probabilística denominada "previsora de entrega" $P_{(A,B)}$ sempre que um nó *A* estabelece uma conexão com um nó *B*. Esta métrica indica quais as chances um determinado nó (no caso, A) tem de entregar uma mensagem ao destino (no caso, B). As mensagens são repassadas somente para nós com maior previsibilidade de entrega ao destino.

No Prophet, o cálculo da "previsora de entrega" $P_{(A,B)}$ possui três fases e P_{init}, θ e γ são parâmetros configuráveis do algoritmo. Na primeira fase, quando o nó *A* encontra o nó *B*, $P_{(A,B)}$ é atualizado como mostrado na equação 1. Na fase 2, os nós trocam informações sobre outros nós que já conhecem. Com essa propriedade de transitividade o nó *A* atualiza a previsibilidade para um nó *C* que *B* já conhecia. A equação 2 mostra essa transitividade. Na terceira fase, a cada período *k* a previsora de entrega para os nós conhecidos é atualizada (eq. 3).

$$P_{(A,B)} = P_{(A,B)old} + (1 - P_{(A,B)old}) \times P_{init} \quad (1)$$

$$P_{(A,C)} = P_{(A,C)old} + (1 - P_{(A,C)old}) \times P_{A,B} \times P_{B,C} \times \theta \quad (2)$$

$$P_{(A,B)} = P_{(A,B)old} \times \gamma^k \quad (3)$$

O Prophet possui um compromisso na escolha de um valor inicial (P_{init}) para o cálculo da "previsora de entrega". Se este for baixo demais, atuam como meros epidêmicos; se for alto demais, o atraso aumentará [Spyropoulos et al. 2005].

Os algoritmos epidêmicos mostraram-se eficientes na entrega das mensagens porém a custo de um aumento do *overhead* na rede. Nos trabalhos citados não é feita uma avaliação sobre o impacto do tamanho do espaço disponível para armazenamento de mensagens (*buffer*). Em dispositivos com recursos limitados, os algoritmos epidêmicos podem ter baixos desempenhos de taxa de entrega de mensagens.

Nosso algoritmo pode ser classificado como estocástico, porém se difere dos demais algoritmos de mesma categoria devido a utilização de agrupamento de nós para fazer um roteamento sem excesso de replicações de mensagens na rede. Na Seção 5 mostramos que o algoritmo proposto tem melhor desempenho em situações em que recursos como o *buffer* são limitados.

2.2. Roteamento Hierárquico

Diversas abordagens de roteamento hierárquico foram propostas em MANETs [Pei et al. 1999], [Yu and Chong 2005]. Nesses trabalhos, uma hierarquia (virtual) é construída utilizando a formação de agrupamentos (*clustering*) multi-nível, permitindo uma abstração da topologia da rede para roteamento. Esses algoritmos propõem métodos de *clustering* em MANETs e políticas de repasse de mensagens entre os grupos que compõem a rede. Nos trabalhos apresentados, a formação de *clusters* para roteamento se mostrou eficiente, principalmente no controle de número de mensagens extras enviadas pela rede (*overhead*).

Para minimizar o número de mensagens necessárias para a formação e manutenção do *cluster*, Kwon e Gerla propõem um protocolo de criação de grupos chamado *Passive Clustering* (PC). No PC é feita a adição de um campo entre as camadas 3 e 4 nos pacotes de dados com a informação do estado do nó (membro, lider ou gateway) [Kwon and Gerla 2002]. Dessa forma, os *clusters* são criados somente sob demanda. O PC evita a inundação da rede com mensagens extras, porém não foi desenvolvido para dar suporte a roteamento hierárquico.

Baseado no *Passive Clustering*, Cramer *et al* apresentam um protocolo chamado "*On-Demand Group Mobility-Base Clustering*"(ODGMBC), que estende o PC modificando um campo da camada MAC para que os próprios pacotes de controle carreguem a informação sobre o estado do nó [Cramer et al. 2004]. O OGDMC consite em verificar e contar os quadros recebidos da camada de acesso ao meio (MAC) de um determinado nó, e a partir de um limite pré-estabelecido reconhecer se o nó é vizinho ou não. Quando um nó reconhece um vizinho, ele envia uma mensagem informando sobre isso e então o nó com o menor indentificador assume a liderança e envia mensagens periodicamente informando seu estado de liderança.

Um algoritmo hierárquico determinístico para redes tolerantes a interrupções é apresentado em [Liu and Wu 2007], onde é proposta a criação de uma árvore hierárquica para encaminhamento das mensagens. Os autores consideram que os nós são fixos ou que suas trajetórias são estritamente repetitivas. Dessa forma, um nó sabe quando entrará em contato com outro nó.

Em nosso trabalho, utilizamos um modelo de mobilidade em grupo mas não há informação sobre a movimentação dos nós e nem sobre suas posições futuras. O OGDMBC será usado como base para criação e manutenção do *clustering*.

3. Modelo de Redes de Emergência

O principal modelo de mobilidade utilizado para análise de protocolos de roteamento em MANETs é o random-WayPoint [Camp et al. 2002]. Este modelo, por ter características aleatórias, não representa bem cenários reais, pois desconsidera a existência de restrições geográficas (obstáculos), a dependência temporal do movimento do nó (correlação entre o histórico de movimento dos nós) e a dependência espacial entre os nós.

Para analisar o PROPHET, Lindgren *et al* criaram um modelo de mobilidade denominado "modelo de comunidade", no qual a área total é dividida em regiões e então cada nó segue para uma região com determinada probabilidade [Lindgren et al. 2003]. Porém, essa abordagem não considera a relação entre os movimentos dos nós.

Em cenários de emergência os nós que compõem a equipe de resgate tendem a se

mover em grupos e em geral há pontos de interesse, como o centro de controle e a região de busca e resgate.

3.1. Mobilidade em Cenários de Emergência

Utilizaremos neste trabalho um modelo que represente o comportamento dos movimentos de equipes de resgate em cenários de emergência. Para tal, consideramos as seguintes premissas:

- Existem grupos de resgate, quem podem ser mapeados em grupos de nós da rede. Tais agrupamentos podem ser representados pelo modelo RPGM, adicionando portanto, dependência temporal entre os nós. Porém, nesse modelo os grupos se movimentam aleatoriamente pela área simulada.
- Existem regiões de interesse, como centrais de atendimento, região a ser resgatada e pontos críticos a se pesquisar.
- Cada grupo possui um líder. Inicialmente será utilizada a política de menor identificador para escolha do líder. A formação de de grupos (*clustering*) é discutida na Seção 4.1.

A partir dessas premissas, propomos um modelo de mobilidade em grupo com áreas de interesse denominado Modelo de Mobilidade em cenários de Emergência (MME). Esse modelo estende o RPGM pela adição de pontos de interesse com probabilidade ρ de serem visitadas pelos grupos. Em cada ponto de interesse há um raio d_{max} indicando o raio da região em que os nós devem ficar. A Tabela 1 sumariza os parâmetros utilizados no MME.

Tabela 1. Principais Parâmetros do MME

g	Quantidade de grupos.
n	Número de nós no grupo.
δ	Distância máxima de um nó ao centro lógico de um grupo.
s	Velocidade mínima e máxima.
p	Ponto de interesse (coordenada x,y).
d	Raio de desvio do ponto de interesse.
ρ	Probabilidade do grupo ir para região do ponto de interesse p.

No RPGM, cada grupo de n nós possui um centro lógico representado por um dos nós do grupo, o qual escolhe aleatoriamente um destino (x_n,y_n) e velocidade aleatória. Os $n-1$ nós restantes seguem o nó central mantendo uma distância de $\pm\delta$.

No MME, pode haver um ou mais os pontos de interesses p. O nó considerado centro lógico de um grupo escolhe um p_i de acordo com sua respectiva probabilidade ρ_i. Para $\sum_{i=1}^{N} \rho_i \leq 1$, sendo N a quantidade de pontos de interesse. A Tabela 2 mostra exemplos de alguns pontos de interesse.

Tabela 2. Exemplo de Diversos Pontos de Interesse

Ponto (p)	(x_i, y_i)	(x_j, y_j)	*aleatório*
Probabilidade ρ	0.3	0.6	0.1

Um destino (x_i,y_i) para o centro lógico é escolhido conforme mostrado nas equações 4 e 5. O nó seleciona uma velocidade entre $[s_{min}, s_{max}]$ e os $n-1$ nós participantes do

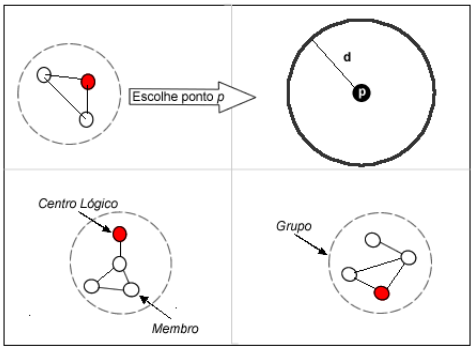


Figura 1. Mobilidade em grupo com áreas de interesse: O centro lógico do grupo escolhe um destino $p \pm d$ de acordo com a probabilidade de cada ponto de interesse. Os demais membos do grupo o seguem para a região escolhida.

grupo o seguem mantendo uma distância $\pm\delta$. A Figura 1 ilustra a movimentação dos grupos de nós com destino a uma área de interesse.

$$x_i \mid p_x - d \leq x_i \leq p_x + d \quad (4)$$

$$y_i \mid p_y - d \leq y_i \leq p_y + d \quad (5)$$

4. Algoritmo Hierárquico

O algoritmo de roteamento baseado em grupos hierárquicos (ARGH) proposto mantém uma tabela com informações dos nós que têm mais conectividade. Utilizando essa tabela, um nó constrói dinamicamente sua lista de vizinhos (Controle de Vizinhança). O líder de um grupo de vizinhos é escolhido pelo Controle de Vizinhança como sendo o vizinho com menor identificador no grupo. É importante ressaltar que o MME somente gera padrões de movimentação em grupo, não interferindo em nenhuma informação que um nó tenha sobre seus vizinhos.

Definimos uma política de roteamento de mensagens que segue um princípio básico de hierarquia: As mensagens somente são transmitidas ao nó líder de um grupo e este é responsável por entregá-las ao destino ou repassá-las a um nó que não seja participande do grupo. Dessa forma, somente os nós líderes fazem replicações das mensagens; os demais nós somente encaminham as mensagens para seus líderes.

O ARGH é composto por dois módulos: Controle de vizinhança e Política de roteamento das mensagens.

A Figura 2 apresenta a arquitetura do ARGH, onde as constantes α e β representam respectivamente o tempo máximo que um nó pode ficar desconectado sem ser desconsiderado como vizinho e o tempo mínimo acumulativo que dois nós devem manter conexão para serem considerados vizinhos. As sub-seções seguintes detalham cada um desses módulos.

4.1. Criação do Agrupamento

Cada nó ARGH possui um estado que o identifica em relação ao grupo, podendo ser: *INICIAL*, o nó não faz parte de nenhum grupo;*MEMBRO*, já é membro de um grupo; e *LIDER*, o líder do grupo.

Utilizamos como base para o nosso trabalho uma simplificação do ODGMBC [Cramer et al. 2004], pois como explicado na Seção 2, no ODGMBC a manuntenção do agrupamento não impacta no desempenho o algoritmo.

O ARGH possui um "Controle de vizinhança"(CV) que mantém uma tabela com os nós conhecidos. O estado inicial de cada nó é *INICIAL* e dessa forma todos são potenciais líderes. Em um primeiro contato, cada nó adiciona o endereço do outro em sua a tabela e marca o tempo (*t*) em que ocorreu a conexão. A cada nova conexão (ou a cada iteração onde a conexão está ativa) com o mesmo nó é executado o cálculo da diferença entre o novo tempo (*t'*) e o tempo marcado anteriormente. Dessa forma é possível verificar quanto tempo o contato está ativo (Δt). Um nó é considerado vizinho se somente se: $\Delta t = t - t'$; $\Delta t \geq \beta$ e $\Delta t \leq \alpha$. Sendo α o tempo máximo que define um nó como vizinho (isso evita que pequenas desconexões descaracterizem o nó como um vizinho) e β o tempo mínimo que um nó tem que manter conexão para ser considerado um vizinho.

As constantes α e β são parâmetros configuráveis do ARGH. Os nós aguardam um período de *warm up* para que todos os vizinhos possam ser reconhecidos. Após esse período, a partir da lista de vizinhos construída, o nó com o menor identificador assume a liderança e altera seu estado para *LIDER*. Diversas políticas de escolha de líder em MANETs vêm sendo propostas [Yu and Chong 2005]. Optamos por utilizar a política de líder com menor identificador devido à simplicidade deste método.

O nó líder envia uma mensagem ao seus vizinhos informando sobre sua liderança. Cada nó informado sobre o líder passa para o estado *MEMBRO* e informa aos seus vizinhos (que estão no estado *INICIAL*) sobre o líder; os nós informados também passam para o estado *MEMBRO*.

O CV verifica em cada iteração a conectividade de todos os nós considerados vizinhos. Quando um nó fica um longo tempo sem conectividade com um de seus vizinhos ($\Delta t\alpha$), este é desconsiderado vizinho. Se o vizinho analisado for o líder conhecido, o nó volta então ao estado de *INICIAL*.

4.2. Política de Roteamento em Grupo Hierárquico

Utilizando a hierarquia criada pelo Controle de Vizinhança, buscamos então minimizar o número de mensagens extras enviadas pela rede para que algoritmos de roteamento em DTN se tornem escaláveis, mas que não seja imposto um número máximo de replicações de mensagens.

A política de envio e repasse do ARGH é baseada no conhecimento que o nó possui sobre o destino. O ARGH age diferentemente se o destino da mensagem for um nó do mesmo grupo da origem ou se for para um grupo diferente.

O estado do nó determina a política de repasse de mensagens. Um nó MEMBRO somente repassa a mensagem ao destino ou ao LIDER, enquanto o nó LIDER é responsável

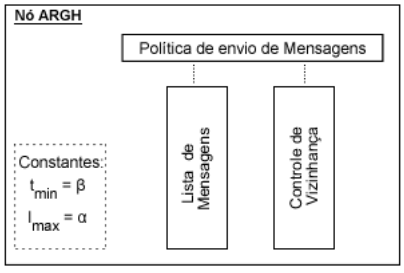


Figura 2. Arquitetura do ARGH

por distribuir as mensagens entre grupos diferentes. As definições 1 e 2 abaixo formalizam estes conceitos do envio de uma mensagem *m* e da participação de um nó *N* em um grupo *G*:

Definição 1 *Seja* $N_{a..z}(G) \mid N[a...z] \in G$*, um* N_k *nó pertencente ao cluster G.*

Definição 2 *Seja* $M(N_a, N_b)$ *o envio de uma mensagem (M) de* N_a *para* N_b*.*

Pela tabela de vizinhança mantida pelo CV, cada nó pode saber quem são seus vizinhos e quem é o LIDER do grupo. Se uma mensagem *M* deve ser enviada entre nós que fazem parte do mesmo grupo então o repasse de mensagem é feita diretamente para o destino. Como dois nós podem ser considerados vizinhos mesmo não havendo uma conexão por um tempo limite, se no momento do envio não houver conexão, a mensagem é repassada para o nó LIDER.

Se a mensagem é para ser enviada a um outro grupo, então ela é repassada para o líder e esse se responsabiliza por enviar a mensagem para um nó que não faça parte do grupo. O nó do grupo que recebeu a mensagem repete essas operações até que a mensagem seja entregue. O algoritmo 1 descreve o princípio básico do nosso protocolo.

Algoritmo 1 Repasse de Mensagens

Entrada: Nó $N_i(G)$ com M Mensagens para N destinos
para M_0 até M_{max} **faça**
  N_k Destino de M_j
  se $N_k \in G$ **então**
    Entrega M_j para N_k; FIM
    se N_i não está conectado com N_k **então**
      Entrega M_j para LIDER; FIM
    fim se
  fim se
  se $N_k \ni G$ **então**
    se $N_i(G)$ é líder **então**
      Envio M_j para nós que não pertencem a *G*; FIM
    fim se
  senão
    Envio M_j para LIDER; FIM
  fim se
fim para

Se uma mensagem é destinada para um nó do mesmo *cluster* do nó origem, no máximo a mensagem é repassada ao líder, que a repassa ao destino, havendo duas replicações da mensagem.

Se uma rede com *N* nós for formada por *n clusters* de tamanho g, no pior caso a mensagem será replicada para um nó de cada *cluster* e seus líderes, ou $2n$ replicações. No Epidêmico, chegará a *N*.

Tal análise demonstra que em redes nas quais o número de nós é pequeno a diferença do *overhead* de comunicação entre o ARGH e o Epidêmico pode ser insignificante. Porém, ao aumentar o tamanho da rede, a diminuição do *overhead* é conseguida de forma satisfatória.

5. Simulação e Análise

Para avaliar e comparar o desempenho do ARGH com diferentes algoritmos de roteamento, utilizamos o simulador "*Opportunistic Networking Evaluator*" (ONE)

[Keranen and Ott 2007]. O ONE simula um modelo de comunicação tolerante a interrupções, onde os nós seguem o paradigma *armazenar-segurar-repassar* mensagens (*store-carry-foward*), podendo mantê-las em um *buffer* caso o nó não tenha conexão direta com o destino. Todos os resultados apresentados possuem 95% de intervalo de confiança. Cada teste foi executado quinze vezes, alterando a semente geradora do padrão de mobilidade em cada vez.

Nossa avaliação se concentrou em comparar o ARGH com o protocolo Epidêmico [Vahdat and Becker 2000] e com o Prophet [Lindgren et al. 2003], utilizando duas métricas:

- Entrega de mensagens, que consiste em calcular a quantidade de mensagens enviadas que atinjiram o destino.
- *Overhead* relativo, calculado por $(\frac{\text{mensagens transmitidas} - \text{mensagens entregues}}{\text{mensagens entregues}})$. Ou seja, quantas mensagens tiveram que ser transmitidas na rede para cada mensagem entregue corretamente ao destino.

Antes de receber uma mensagem, um nó verifica se existe espaço de armazenamento em seu *buffer*. Caso esse esteja cheio, as mensagens mais antigas são descartadas até que haja espaço para a nova mensagem. Esta política de fila "primeiro a entrar - primeiro a sair" para descarte de mensagens foi utilizada em todos os algoritmos de roteamento analisados. Nos quais um nó somente aceita mensagens que ele não esteja armazenadas em seu *buffer*.

Analisamos a influência do tamanho do *buffer* disponível para armazenamento de mensagens e em seguida fazemos uma análise da escalabilidade do ARGH com diversos tamanhos de *buffers*.

5.1. Configuração do Experimento

Para simularmos um cenário de emergência, consideramos uma área de 3000x3000m, com um ponto de interesse com raio de 300m. Os nós podem se mover com velocidade entre 2 e 5m/s. O tempo simulado foi de 6000s. Utilizamos os valores padrões do 802.11 para alcance de rádio e largura de banda, com 250m e 1Mbps respectivamente.

As mensagens foram geradas com distribuição uniforme entre 20 e 35s e com tamanho entre 50kb e 500kb, que pode representar troca de arquivos de texto e imagens de pequena resolução.

No PROPHET, foram utilizados os parâmetros $P_{init} = 0.75$, $\theta = 0.25$, $\gamma = 0.98$ e k = 30s como em [Lindgren et al. 2003].

Testes preliminares indicaram escolha dos parâmetros do Controle de Vizinhança do ARGH, sendo fixados em $\alpha = 500s$ e $\beta = 60s$ para todas as simulações realizadas. No MME, foram formados grupos de 5 nós e variada a quantidade de grupos de nós na rede.

5.2. Resultados e Análises

5.2.1. Impacto da Capacidade de Armazenamento do *Buffer* de Mensagens

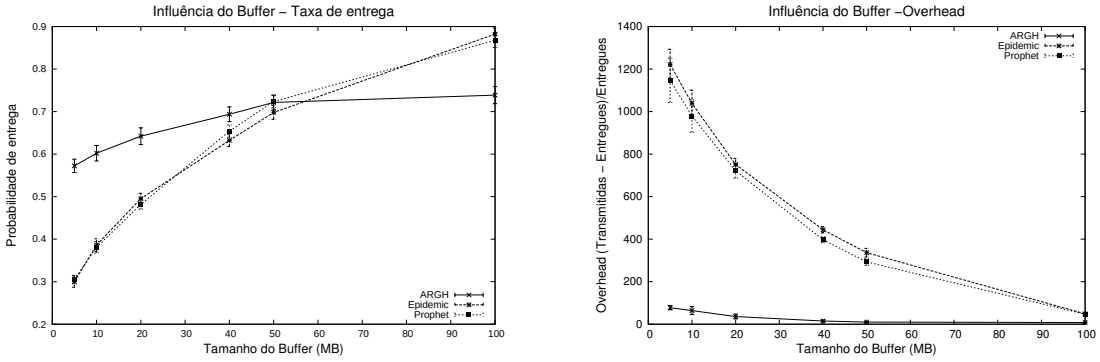
Neste experimento variamos a capacidade de armazenamento de mensagens entre 5 e 100MB em cada nó. Executamos os testes com 50 e 100 nós.

Os gráficos das Figura 3(a) e 3(b) mostram o desempenho dos algoritmos em relação à capacidade do *buffer* em uma rede com 50 nós. O algoritmo Epidêmico e o Prophet, por causarem inundação na rede, têm um baixo desempenho com capacidade de *buffers* pequenos. Contudo, o *overhead* relativo teve valores extremamente altos para os dois algoritmos, com 5MB de capacidade de *buffer* o Epidêmico enviou em média 1200 mensagens para cada uma que entregou. Tal fato ocorre devido à política de descarte de mensagens do *buffer*. Quando um nó recebe uma mensagem e está com a capacidade do *buffer* completamente utilizada, o nó exclui a mais antiga do *buffer*. Dessa forma, quando a capacidade de armazenamento é baixa, o nó está excluindo e recebendo uma mesma mensagem várias vezes.

O ARGH manteve o melhor desempenho de taxa de entrega de mensagens nas situações em que as capacidades de armazenamento dos *buffers* eram pequenas (até 60MB). Contudo, manteve sua taxa de *overhead* praticamente constante. Com 5MB de capacidade de *buffer* o ARGH foi superior ao Epidêmico e ao Prophet em média de 65% na taxa de entrega e teve um *overhead* 1400% inferior. No outro extremo, com 100MB de capacidade de *buffer*, o ARGH teve um desempenho na taxa de entrega de aproximadamente 17% inferior em relação aos outros analisados, mas manteve um *overhead* de 58% abaixo dos apresentados pelos demais algoritmos.

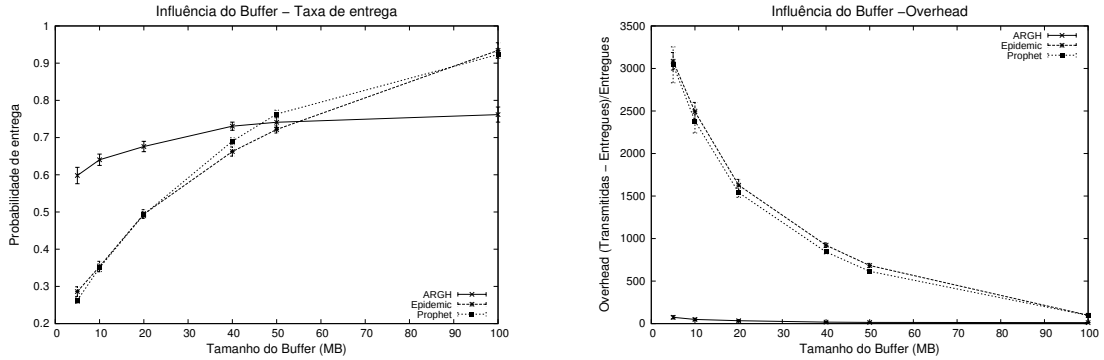
Os gráficos das Figuras 4(a) e 4(b) mostram o mesmo experimento, porém com 100 nós participando da rede. Percebemos um comportamento bastante similar entre os cenários com 50 e 100 nós para taxa de entrega e *overhead* dos algoritmos Epidêmico e Prophet. Em ambos os cenários, o *overhead* relativo manteve um comportamento de distribuição exponencial inversa e o ARGH uma constante aproximada. Ressaltamos que com 100 nós o número de mensagens extras transmitidas na rede praticamente triplicou no algoritimo Epidêmico e no Prophet.

Para *buffer* de armazenamento de mensagens com capacidade maiores que 100MB, os resultados mantêm-se constantes. Os mesmos testes foram refeitos utilizando *buffer* com capacide de 500MB. Os resultados mostraram que, com o padrão de carga de tráfego de mensagens utilizados, o aumento de capacidade de *buffer* para mais que 100MB não causa diferença nos resultados das métricas utilizadas.



(a) Taxa de entrega (b) Overhead

Figura 3. Influência da capacidade do *buffer* de mensagens com 50 nós na rede



(a) Taxa de entrega (b) Overhead

Figura 4. Influência da capacidade do *buffer* de mensagens com 100 nós na rede

5.2.2. Escalabilidade

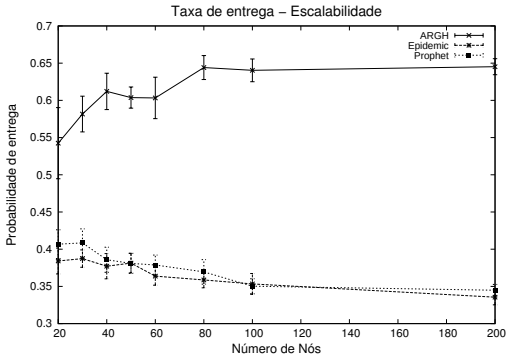
Analisamos a escalabilidade dos algoritmos em dois cenários. Primeiramente, observamos o comportamento dos algoritmos quando o *buffer* de mensagens tem capacidade limitada (10MB). Em seguida fizemos a mesma análise utilizando *buffers* com capacidade de 100MB de armazenamento. Em ambos os testes, variamos a quantidade de nós participantes da rede de 20 a 200 nós.

Com a capacidade de *buffer* de 10MB, o ARGH mostrou-se eficiente na entrega dos dados, mantendo a taxa de entrega crescente em relação à quantidade de nós na rede, enquanto o *overhead* relativo manteve-se praticamente constante. Como esperado, em situações com *buffer* limitado, o algoritmo Epidêmico tem o pior desempenho de taxa de entrega e de *overhead* relativo entre os algoritmos analisados. Os gráficos das Figuras 5(a) e 5(b) mostram os resultados nesse cenário.

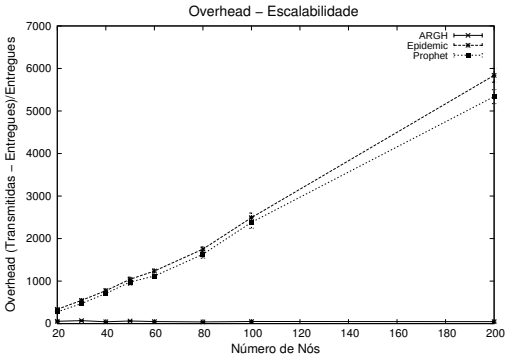
O algoritmo Epidêmico e o Prophet apresentaram taxas de entrega descrescentes em relação ao número de nós. Isso ocorre pois o aumento do número de nós causa o aumento de mensagens transmitidas na rede. Porém, com a capacidade de *buffer* limitada, o *buffer* de mensagens estará constantemente cheio o que causará descarte das mensagens na fila de entrega.

Os gráficos das Figuras 6(a) e 6(b) mostram a escalabilidade dos algoritmos analisados utilizando um *buffer* de mensagens com capacidade de 100MB. O Epidêmico e o Prophet mantêm uma taxa de entrega superior ao ARGH. Entretanto, o *overhead* de transmissão de dados para o Epidêmico e o Prophet é crescente (linearmente) em relação ao número de nós. Isso significa que nesses algoritmos, quando há espaço suficiente de armazenamento no *buffer* de mensagens, uma mensagem é replicada até N vezes para cada mensagem entregue, sendo N o número de nós na rede. Já o ARGH manteve-se praticamente constante no *overhead* de transmissão comunicação. A pequena variação é causada pela quantidade de grupos existentes em cada quantidade de nós.

Apesar do ARGH ter um desempenho inferior quando a capacidade do *buffer* de mensagens é alto, o ARGH se mostrou um algoritmo escalável em todos os cenários analisados. Um compromisso entre memória disponível para armazenamento de mensagens e taxa de entrega, fazem do ARGH uma solução atraente quando os dispositivos que compõem a rede possui recursos limitados.

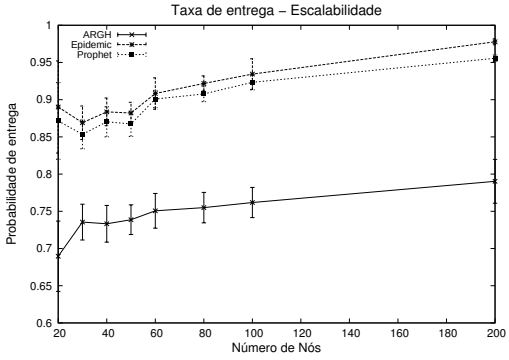


(a) Taxa de entrega

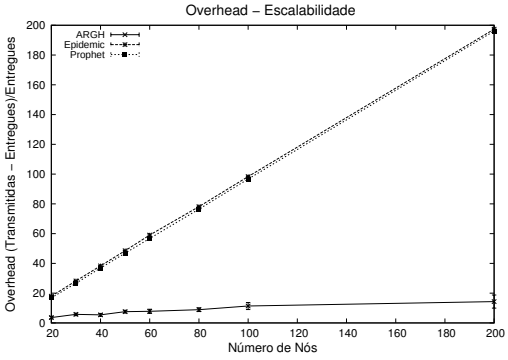


(b) Overhead

Figura 5. Escalabilidade dos Algoritmos com *Buffer* de 10MB



(a) Taxa de entrega



(b) Overhead

Figura 6. Escalabilidade dos Algoritmos com *Buffer* de 100MB

6. Conclusões

Neste trabalho propusemos um novo algoritmo de roteamento para redes tolerantes a interrupções, batizado de ARGH, que tem como objetivo aumentar a taxa de entrega de dados em redes intermitentes sem afetar o *overhead* de comunicação de rede. Esta diminuição no *overhead* de transmissão é essencial quando os dispositivos que formam a rede *ad hoc* possuem recursos limitados, como por exemplo em redes de sensores sem fios.

Analisamos nosso algoritmo utilizando um modelo de mobilidade que reflete propriedades de cenários de emergência. O ARGH se mostrou escalável em todos os cenários analisados, visto que o aumento do número de nós não afeta o *overhead* de transmissão de dados como ocorre com os algoritmos Epidêmico e Prophet.

O Epidêmico e o Prophet exigem que se tenha capacidade de armazenamento de mensagens proporcional ao número de nós e ao padrão de tamanho e frequência de mensagens na rede. Em dispositivos de rede com recursos limitados o ARGH se mostrou eficiente na taxa de entrega sem afetar o *overhead* de comunicação da rede. Percebemos um compromisso entre taxa de entrega e *overhead* de comunicação.

Como trabalhos futuros pretendemos adicionar características probabilísticas ao ARGH, de forma que os nós líderes repassem uma determinada mensagem para nós de outros grupos que tenham maiores probabilidades de entregá-la ao destino.

A escolha do líder é baseada na política de nó vizinho com menor identificador. A utilização de um metódo mais eficiente, como nó com maior quantidade de vizinhos conectados, também poderia trazer melhorias na taxa de entrega. Além disso, pretendemos fazer uma análise sobre o impacto de políticas de descarte de mensagens que sigam outros modelos de filas além do modelo "Primeiro-entrar, Primeiro-Sair" utilizado neste trabalho.

Referências

(http://www.dtnwg.org, 2008). *Delay-Tolerant Network work Group*. DTN.

(http://www.ietf.org/html.charters/manet-charter.html, 2008). *IETF Working Group on Mobile Ad-hoc Networks*. IETF.

Camp, T., Boleng, J., and Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502.

Cramer, C., Stanze, O., Weniger, K., and Zitterbart, M. (2004). Demand-driven clustering in manets. In *International Conference on Wireless Networks*, pages 81–87.

Grossglauser, M. and Tse, D. (2002). Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 10(4):477–486.

Handorean, R., Gill, C., and Roman, G. (2004). Accommodating Transient Connectivity in Ad Hoc and Mobile Settings. *Pervasive Computing: Sec. International Conference, Pervasive 2004, April 18-23, 2004: Proceedings*.

Jain, S., Fall, K., and Patra, R. (2004). Routing in a delay tolerant network. *Proceedings of ACM SIGCOMM'04*.

Keranen, A. and Ott, J. (2007). Increasing reality for DTN protocol simulations. *Networking Laboratory, Helsinki University of Technology, Tech. Rep*.

Kwon, T. and Gerla, M. (2002). Efficient flooding with Passive Clustering (PC) in ad hoc networks. *ACM SIGCOMM Computer Communication Review*, 32(1):44–56.

Lindgren, A., Doria, A., and Schelen, O. (2003). Probabilistic routing in intermittently connected networks. *Mobile Computing and Communications Review*, 7(3):19–20.

Liu, C. and Wu, J. (2007). Scalable routing in delay tolerant networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 51–60. ACM Press New York, NY, USA.

Pei, G., Gerla, M., Hong, X., and Chiang, C. (1999). A wireless hierarchical routing protocol with group mobility. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, pages 1538–1542.

Rao, R., Eisenberg, J., and Schmitt, T. (2007). Improving Disaster Management: The Role of IT in Mitigation, Preparedness, Response, and Recovery. *National Research Council, National Academy of Sciences Washington DC, ISBN: 0-309-66744-5*.

Spyropoulos, T., Psounis, K., and Raghavendra, C. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM.

Vahdat, A. and Becker, D. (2000). Epidemic routing for partially connected ad hoc networks. *Tec. Report, Duke University*.

Yu, J. and Chong, P. (2005). A survey of clustering schemes for mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 7(1):32–48.

Zhang, Z. (2006). Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials, IEEE*, 8(1):24–37.