

Utilização de Enxame de Partículas para Roteamento em Redes Oportunistas

Alex Rosch de Faria, Ana Cristina B. Kochem Vendramin, Anelise Munaretto, Myriam Regattieri Delgado

Resumo—Redes Oportunistas (OPNETs - *Opportunistic Networks*) são redes compostas por nós sem fio onde as oportunidades de comunicação são intermitentes devido a alta mobilidade dos nós. Portanto, um caminho fim-a-fim entre um nó origem e um nó destino pode não existir. Essas características de rede tornam o roteamento um dos grandes desafios em OPNETs, pois a probabilidade de entrega de cada mensagem depende de uma seleção apropriada de um ou mais nós candidato(s) a encaminhador(es) de mensagens. Esse artigo propõe um protocolo de roteamento para OPNETs chamado PSONET (*PSO for Opportunistic Networks*) que utiliza a técnica denominada de otimização por enxame de partículas (PSO do inglês *Particle Swarm Optimization*) para obter informações sobre a conectividade dos nós e direcionar o tráfego da rede através de um subconjunto de bons encaminhadores de mensagens. Os resultados mostram que o PSONET obtém ganhos em termos da taxa de entrega de mensagens e relação de redundância de mensagens se comparado com os protocolos Epidêmico e PROPHET.

Palavras-Chave—Redes Oportunistas, Redes Tolerantes a Atraso, Enxame de Partículas.

Abstract—Opportunistic Networks (OPNETs) are networks comprised of wireless nodes where the communication opportunities are intermittent due to high mobility of nodes. Therefore, an end-to-end path between source destination nodes may not exist. These features make the network routing a major challenge in OPNETs since the probability of delivery of each message depends on a proper selection of one or more message forwarders. This paper proposes a routing protocol for OPNETs called PSONET (PSO for Opportunistic Networks) which uses Particle Swarm Optimization (PSO) to gather information about node connectivity and direct network traffic through a subset of good message forwarders. Results show that the proposed protocol obtains significant gains in terms of message delivery rate and message redundancy when compared to Epidemic and PROPHET protocols.

Keywords—Opportunistic Networks, Delay Tolerant Networks, Particle Swarm Optimization.

I. INTRODUÇÃO

Uma Rede Oportunista tem como principal objetivo suportar a mobilidade de usuários entre um grupo de redes heterogêneas para solucionar desafios como conectividade intermitente, mudança constante de topologia e atrasos longos [1]. Vários conceitos por trás das OPNETs são originados a partir de trabalhos de investigação sobre Redes Tolerantes a Atrasos (DTNs - *Delay Tolerant Networks*).

Para operar nessas redes, a transferência de custódia de mensagens é fornecida por nós intermediários até que o destino

Alex de Faria, Ana Cristina Vendramin, Anelise Munaretto, Myriam Delgado, CPGEI, Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba-PR, Brasil, E-mails: alex@alexfaria.eti.br, {anelise, criscochem, myriamdelg}@utfpr.edu.br. Este trabalho foi parcialmente financiado pelo CNPq.

destas seja alcançado. Até a ocorrência de uma oportunidade de contato, o nó poderá manter as mensagens de outros nós em seu *buffer* por um longo tempo (SCF - *Store-carry-forward*) [2]. As redes oportunistas são compostas por redes de nós sem fio auto-organizáveis, formando assim um ambiente tolerante a falhas sem a limitação de conectividade fim-a-fim. A comunicação ocorre de forma local, ou seja, decisões de roteamento são feitas por dispositivos com base em informações locais.

O termo “oportunista” em relação às redes sem fio implica na tendência de seus dispositivos em explorar os recursos disponíveis, tirando proveito de informações globais acessíveis localmente. Os dispositivos transmitem informações globais através de interações locais com outros dispositivos [3].

Essas premissas caracterizam um ambiente favorável para a aplicação de paradigmas baseados em população de soluções, como os oriundos da inteligência coletiva (*Swarm Intelligence*). Um dos exemplos mais conhecidos é a otimização por enxame de partículas (do inglês *Particle Swarm Optimization*) [4]. O PSO é uma técnica de otimização estocástica, desenvolvida por Kennedy e Eberhart [4] em 1995, na qual simula-se o comportamento social encontrado em bando de pássaros. O protocolo desenvolvido é inspirado nas equações de movimentação das partículas do PSO, modificado para tratar de caminhos ou rotas em redes oportunistas.

O documento está organizado da seguinte forma: a seção II apresenta os trabalhos relacionados. A seção III apresenta uma breve revisão bibliográfica sobre a técnica de otimização por enxame de partículas. A seção IV detalha o protocolo proposto. A seção V apresenta os resultados obtidos. Por fim, na seção VI apresenta-se a conclusão.

II. TRABALHOS RELACIONADOS

O roteamento é um dos principais desafios para OPNETs, devido às conexões intermitentes e à alta taxa de mobilidade dos nós [5]. Os protocolos de roteamentos para OPNETs podem ser classificados como: baseados no encaminhamento de mensagens (*Forwarding-based*), baseados em contexto (*Context-Based Forwarding*) e baseados em inundação (*Flooding-based*) [1]. Protocolos baseados em encaminhamento de mensagens direcionam a mensagem somente para um único nó a cada etapa do roteamento. Esta abordagem reduz a utilização do *buffer* dos nós e o número de mensagens transferidas na rede, porém podem ocorrer longos atrasos e uma baixa taxa de entrega de mensagens. *Minimum Estimated Expected Delay* (MEED) [6] é um exemplo de protocolo de roteamento que se enquadra neste grupo. Nos protocolos

baseados em contexto, as decisões de roteamento são baseadas em informações obtidas da rede. O protocolo PROPHET [7] é o protocolo mais conhecido desta categoria, pois ele se baseia no histórico de encontros dos nós para estimar a probabilidade de entrega das mensagens. Por fim, protocolos baseados em inundação geram múltiplas cópias de uma mesma mensagem e as distribuem na rede na tentativa de se obter alta taxa de entrega de mensagens. Essa abordagem implica em um alto custo em relação à quantidade de mensagens que são transmitidas na rede e à capacidade de armazenamento dos nós intermediários. O protocolo de roteamento Epidêmico, proposto por [8], utiliza esta abordagem.

Protocolos que utilizam inteligência coletiva são capazes de encontrar soluções de modo paralelo e iterativo devido ao método ser baseado em uma população de indivíduos. O comportamento que emerge é o de um grupo de indivíduos que interagem através regras simples e dinamicamente se auto-organizam mantendo as suas posições ao redor de localizações ótimas [9]. Como exemplos de protocolos de roteamento que utilizam inteligência coletiva pode-se citar: [10] e [11] propõem o uso de Otimização por Colônia de Formigas (ACO do inglês *Ant Colony Optimization*) para roteamento em redes móveis ad hoc com conectividade limitada; [12] empregam ACO para obter informações sobre a dinâmica de uma OPNET e auxiliar no roteamento, escalonamento e gerenciamento de *buffer*; [13] utilizam ACO e Algoritmos Culturais para prover qualidade de serviço no roteamento de uma topologia fixa de nós; [14] propõem o uso de ACO e Algoritmos Culturais para prover roteamento, escalonamento e gerenciamento de *buffer* nas OPNETs; e [15] propõe o uso do PSO em uma rede estática com topologia pequena de nós.

III. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

A população no PSO, denominada nuvem ou enxame, é composta por partículas, cujas posições representam as potenciais soluções para um problema estudado. Cada partícula n possui uma velocidade v_n^t responsável por comandar o processo de otimização. A velocidade de uma partícula determina a direção na qual a partícula se movimentará, com objetivo de melhorar sua posição atual.

A busca pela melhor posição (solução) é realizada através de um processo iterativo onde a posição de cada partícula é alterada adicionando-se uma velocidade à sua posição corrente, fazendo com que a partícula se movimente sobre o espaço em busca de melhores soluções. A cada iteração, uma função de avaliação (*fitness*) avalia a qualidade de cada posição ocupada pela partícula no espaço de busca. Cada partícula n armazena na memória a sua melhor posição já alcançada $pbest_n$ e a melhor posição já alcançada entre todas as partículas da nuvem $gbest$. Estes dois componentes em adição ao fator de inércia resultam na velocidade de movimentação da partícula, como definido por [4]:

$$v_n^t = w.v_n^{t-1} + c_1.r_1(pbest_n - x_n^{t-1}) + c_2.r_2(gbest - x_n^{t-1}), \quad (1)$$

onde w é o fator de inércia que controla o quanto a direção anterior da partícula irá influenciar na sua nova velocidade, c_1

é o fator cognitivo que indica a auto-confiança da partícula, c_2 é o fator social que força a partícula a seguir na direção da melhor partícula da nuvem, r_1 e r_2 são números aleatórios entre [0,1] que incrementam o caráter aleatório do processo.

A atualização da posição da partícula n é definida por [4]:

$$x_n^t = x_n^{t-1} + v_n^t. \quad (2)$$

A movimentação da partícula consiste simplesmente em adicionar velocidade à sua posição atual. O processo é repetido até que uma condição de parada seja satisfeita: (i) encontrar um valor aceitável para uma solução ótima; ou (ii) executar um número máximo de iterações.

Conforme é possível observar no Algoritmo 1, a nuvem de partículas é inicializada através da distribuição aleatória das partículas dentro do espaço de busca. Em seguida, inicia-se um processo iterativo onde a posição de cada partícula é alterada adicionando-se uma velocidade a sua posição corrente, fazendo com que a partícula se movimente sobre o espaço em busca de melhores soluções. Realiza-se o procedimento para verificar se houve melhora da solução da partícula e do enxame. Em caso afirmativo, os valores de $pbest$ e $gbest$ são atualizados. Este procedimento é realizado através do cálculo do *fitness* (o qual é dependente da aplicação - neste trabalho será calculado conforme (3)). Por fim, o algoritmo é encerrado caso o critério de parada seja atendido ou o número máximo de iterações seja alcançado.

Algoritmo 1 Pseudocódigo do algoritmo de PSO [4].

```

1: Atribui parâmetros
2: for  $n=1$  até o tamanho do enxame do
3:   Inicia  $\vec{x}_n$  como uma solução aleatória para o problema
4:   Inicia  $\vec{v}_n$  com uma velocidade aleatória  $< v_{max}$ 
5: end for
6: while Condição de parada do
7:   for Para cada partícula  $n$  do
8:     Cálculo da velocidade da partícula  $\vec{v}_n$  (1)
9:     Atualização da posição da partícula (2)
10:    Avaliação da nova partícula  $f(\vec{x}_n)$ 
11:    if  $x_n$  melhor do que  $pbest_n$  then
12:       $pbest_n \leftarrow x_n$ 
13:    end if
14:    if  $pbest_n$  melhor do que  $gbest$  then
15:       $gbest \leftarrow pbest_n$ 
16:    end if
17:  end for
18: end while

```

IV. PROTOCOLO PSONET

O PSONET interpreta as condições de comunicação da rede, adaptando-se aos recursos disponíveis. Portando, a variação na quantidade de nós no alcance de cada nó da rede poderá ser utilizada como parâmetro de definição do modo de operação do protocolo. Nos casos em que a rede esteja bastante esparsa, torna-se necessário estabelecer mecanismos que garantam a replicação das mensagens para todos (ou quase todos) os nós,

mesmo que esse procedimento resulte em uma sobrecarga de mensagens na rede. Já nos casos de alta densidade de nós, o protocolo verificará a maior probabilidade de cada candidato a encaminhador entregar as mensagens ao destino pretendido, para então definir o(s) melhor(es) encaminhador(es), o que ocasionará uma maior economia de recursos da rede.

Sendo assim, o PSONET tem como objetivo principal identificar o estado atual da rede e assegurar a melhor forma de entrega de cada mensagem ao seu destino, através de um dos seus modos de operação que estão detalhados na sequência.

A. Modos de Operação

A execução do PSONET é iniciada quando um nó A encontra seus nós vizinhos. Nesse momento, é analisada a quantidade de nós na área de alcance do nó A para definir o modo de operação do protocolo: esparso ou denso. No modo de operação esparso, cada nó transmite suas mensagens para todos os nós alcançáveis atuando como o protocolo Epidêmico [8]. A cada novo contato entre os nós, ocorre uma troca de vetores com o resumo das mensagens contidas em cada *buffer*. Ao receber este vetor, cada nó solicita ao outro nó as mensagens que não possui.

No modo de operação denso, cada nó transmite suas mensagens para o(s) nó(s) com maior probabilidade de entrega da mensagem ao seu destino. Nesse modo é executada a solução de encaminhamento de mensagens utilizando o PSO. Informações relacionadas aos encontros dos nós também são utilizadas para determinar os nós que possuem a maior probabilidade de entrega da mensagem ao destino. Esse processo visa reduzir a replicação de mensagens na rede.

Em ambos os modos, a cada encontro realizado ocorre a troca de informações referentes ao registros de encontros (*Histórico de Contatos*) entre os nós. No *Histórico de Contatos* são armazenadas informações referentes às conexões realizadas por cada nó: identificação da conexão; identificação do nó encontrado; tempo de início da conexão; e tempo final da conexão. Estas informações são utilizadas para o cálculo da probabilidade de entrega de mensagem de cada nó com cada destino de uma mensagem.

Para a solução do problema de encaminhamento de mensagens utilizando o PSO, foi necessário realizar algumas alterações nas equações utilizadas pelo Algoritmo 1 para possibilitar a codificação da partícula para a resolução de um problema discreto. O algoritmo responsável pela escolha de qual nó receberá a mensagem no modo denso utiliza o método de codificação da partícula apresentado em [15]. Assim, foi possível estabelecer um processo de decisão vinculado ao número de ocorrências de cada nó vizinho nos melhores caminhos para um destino sob diferentes métricas de avaliação.

B. Encaminhamento de Mensagens via Enxame de Partículas

A inicialização das partículas é realizada através da geração dos k melhores caminhos entre o nó portador da mensagem e o nó destino da mensagem, baseado no conhecimento adquirido através do *Histórico de Contatos* transformado em um grafo $G(V, A)$. Foi utilizado o algoritmo *KShortestPaths* [16] para determinar os k melhores caminhos por ordem de peso

crecente. O algoritmo é uma variante do algoritmo Bellman Ford [17], mas em vez de apenas armazenar o melhor caminho armazena o k melhores caminhos em cada passagem.

Para avaliar os melhores caminhos, foram utilizadas as seguintes métricas: idade do contato $e(i, j)$; frequência de contatos $f(i, j)$; e duração do contato $d(i, j)$. A métrica $e(i, j)$ de dois nós i e j representa o tempo decorrido desde o último encontro entre i e j . Esta métrica possibilita que o protocolo possa dar preferência para conexões mais recentes. $f(i, j)$ representa o número de encontros ocorridos entre os nós i e j , identificando assim as oportunidades de conexões. $d(i, j)$ representa o tempo total das conexões entre i e j . Esta métrica permite identificar contatos “duradouros” que possibilitam a transferência de uma maior quantidade de dados [18].

Após a geração das partículas, calcula-se o *fitness* de cada partícula (caminho) de acordo com as métricas definidas (valores normalizados). Neste processo, somam-se os valores das arestas dos nós que compõem a partícula (caminho) para cada métrica definida. O valor obtido pelo caminho é dividido pelo número de saltos conforme (3).

$$fitness = \frac{f(i, j) + d(i, j) + \frac{1}{e(i, j)}}{Nr_{saltos}}. \quad (3)$$

Cada partícula armazenará a sua melhor posição encontrada até o momento no *pbest*. Após avaliação de todas as partículas, define-se a melhor partícula encontrada no enxame (maior valor encontrado) como o *gbest*. Num contexto discreto e com partículas de diferentes tamanhos, algumas modificações são necessárias em relação ao algoritmo PSO padrão.

A melhor partícula encontrada até o momento é utilizada como uma partícula atratora no processo de atualização das demais partículas. Neste processo calcula-se a diferença das posições entre a melhor solução encontrada por todas as partículas (*gbest*) e a melhor posição da partícula até o momento (*pbest*). Isto é, verifica-se quais nós estão presentes no *gbest* mas que não fazem parte da melhor posição encontrada pela partícula *pbest*. Essa diferença de posições resultará na velocidade da partícula.

Para cada partícula que será atualizada, gera-se um grafo temporário $G(V, A)$ contendo o caminho do nó até o destino da mensagem, adicionam-se a este grafo os nós contidos na velocidade da partícula, respeitando as informações de vizinhança contidas no *Histórico de Contatos*. Com base nas informações contidas no grafo temporário geram-se novas partículas na tentativa de se encontrar melhores soluções.

Avalia-se cada nova partícula, atualiza-se o *pbest* e o *gbest*. Repete-se o processo de atualização das partículas até que o critério de parada seja atingido.

As melhores soluções apresentadas ao final do ciclo representam os possíveis nós encaminhadores de mensagem, isto é, os nós que foram melhores avaliados em todas as métricas utilizadas.

Com base nas informações dos melhores caminhos retornados pelo PSO, avalia-se a quantidade de vezes em que os nós que estão no alcance do nó portador da mensagem fizeram parte destes melhores caminhos. Encaminha-se a mensagem para o nó que aparecer mais vezes nos caminhos retornados pelo PSO.

V. AMBIENTE DE SIMULAÇÃO E RESULTADOS

O cenário utilizado neste artigo consiste em uma área 8800mx7800m dividida em cinco comunidades ou Pontos de Interesse (*PIs*). Esse cenário simula um grupo de pessoas em sua comunidade, que entrarão, eventualmente, em contato com alguma outra pessoa e trocarão dados. Existe uma pequena probabilidade de essas pessoas irem para outros *PIs* diferente da sua comunidade domiciliar, mas há uma grande probabilidade de que elas se encontrem em *PIs* em comum. Existem quatro grupos de nós, cada um com quarenta nós (40) inseridos de forma aleatória em um *PI*. Cada nó tem um *PI* domiciliar que é mais provável de ele visitar. Nós selecionam um destino e uma velocidade entre 0,5-1,5 m/s, se movem até lá, esperam por um período de tempo de 100-200 segundos e, em seguida, selecionam o próximo destino. Mensagens são configuradas com TTL - *Time to Live* de 600 minutos e têm tamanho igual a 500 kB. O tráfego é gerado a cada 5-10 segundos. O alcance da comunicação é de 10 metros e a velocidade de transmissão é de 250 Kbps. A simulação tem duração de 800.000 segundos. O PSO executa 5 iterações.

Neste trabalho estamos interessados em avaliar duas categorias de protocolos para OPNETs que podem operar em vários ambientes com diferentes modelos de mobilidade dos nós. Assim, o protocolo Epidêmico baseado em inundação de mensagens [8] e o protocolo PROPHET baseado em contexto [7] são utilizados para comparação de desempenho com o protocolo PSONET. Diferente do Epidêmico e do PROPHET, o PSONET tira vantagem do aprendizado adquirido pela experiência individual de cada nó e também pela aprendizagem coletiva para obter informações relevantes da rede. Com base nestas informações (locais e globais) o protocolo é capaz de analisar a utilidade de cada contato como encaminhador de mensagens e limitar o número de mensagens replicadas na rede.

As simulações foram realizadas através do simulador The ONE (do inglês *The Opportunistic Network Environment*) na versão 1.5. [19]. Foram variados o tamanho do *buffer* dos nós e o TTL das mensagens. Os resultados apresentados mostram valores médios de trinta execuções e intervalos de confiança em um nível de 95%.

A Fig. 1 mostra a taxa de entrega das mensagens em relação à variação do tamanho de *buffer* calculada de acordo com (4) para os três protocolos.

$$Entrega = \frac{M_{entregues}}{M_{criadas}} * 100, \quad (4)$$

onde $M_{entregues}$ representa o número de mensagens entregues ao seu destino final e $M_{criadas}$ o número de mensagens criadas durante a simulação.

Conforme o tamanho do *buffer* é incrementado, aumenta-se também o número de mensagens entregues com sucesso para os três protocolos. Isso porque quanto maior o tamanho do *buffer*, mais mensagens são armazenadas, reduzindo assim o risco de descarte de mensagens. O PSONET provê os melhores resultados devido à utilização do seu conhecimento da rede obtido através do *Histórico de Contatos*. Através do *Histórico de Contatos* foi possível a geração dos k melhores

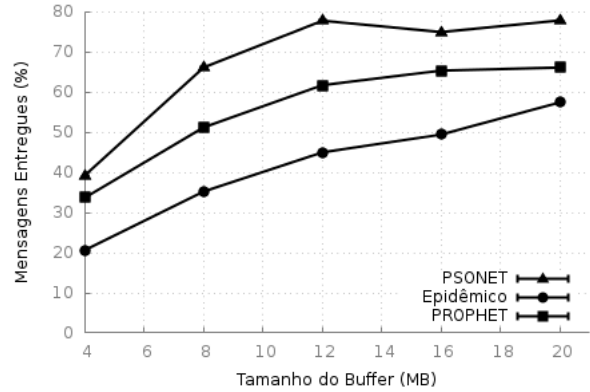


Fig. 1. Taxa de Entrega de Mensagens variando o Tamanho do *Buffer*.

caminhos entre um nó e o destino da mensagem, que por sua vez são otimizados através do PSO. A otimização dos caminhos iniciais via PSO é possível uma vez que a meta-heurística considera as três métricas simultaneamente, ao contrário do método utilizado pelo algoritmo KShortestPaths [16].

Para todos os tamanhos de *buffer*, o PSONET obteve um desempenho superior ao dos outros protocolos. Por exemplo, em um *buffer* de tamanho de 12MB, o PSONET entrega praticamente 80% das mensagens contra 60% do PROPHET e 45% do Epidêmico.

A Fig. 2 mostra a relação de redundância de mensagens calculada de acordo com (5) para os três protocolos.

$$Redundância = \frac{M_{replicadas} - M_{entregues}}{M_{entregues}}, \quad (5)$$

onde $M_{replicadas}$ representa o número de mensagens replicadas e $M_{entregues}$ o número de mensagens entregues.

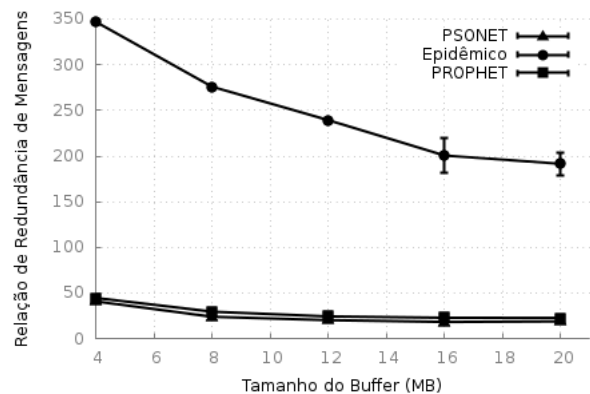


Fig. 2. Redundância de Mensagens variando o Tamanho do *Buffer*.

As Fig. 3 e 4 mostram respectivamente a taxa de entrega de mensagens e a relação de redundância de mensagens em relação à variação do TTL das mensagens, com um tamanho de *buffer* de 4 MB. Para todos os protocolos avaliados, quanto maior o TTL, menor a taxa de entrega das mensagens. Isso pode ser explicado pelo fato de que quando o TTL das mensagens cresce, essas mensagens permanecem por mais tempo no *buffer* dos nós, impedindo que outras mensagens que ainda não foram entregues cheguem em seus destinos.

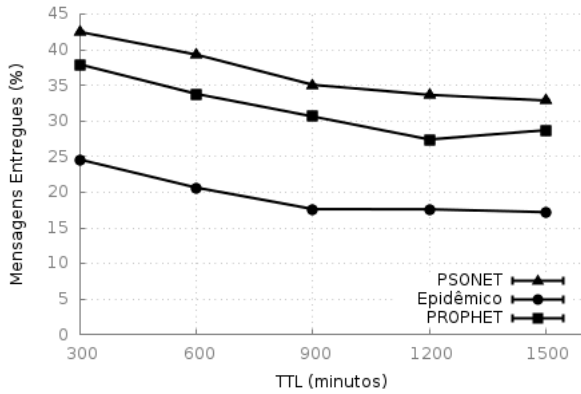


Fig. 3. Taxa de Entrega de Mensagens variando o TTL das Mensagens.

Na Fig. 4, para todos os valores de TTL, o PSONET gerou uma relação menor de redundância de mensagens. Os resultados foram equivalentes aos obtidos pelo protocolo PROPHET. Isso pode ser explicado pelo fato de que tanto o PROPHET quanto o PSONET buscam selecionar melhor o(s) melhor(es) contato(s), antes do encaminhamento das réplicas das mensagens. Já o protocolo Epidêmico replica as mensagens para todos os nós encontrados.

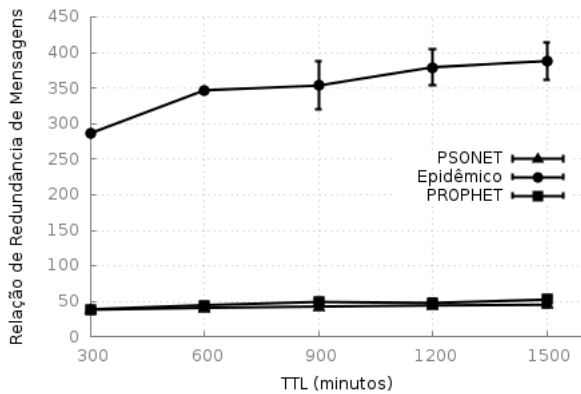


Fig. 4. Redundância de Mensagens variando o TTL das Mensagens.

VI. CONCLUSÕES

O presente artigo propõe um protocolo de roteamento para OPNETs, chamado PSONET, baseado em otimização por enxame de partículas. O objetivo do PSONET é selecionar os melhores encaminhadores de cada mensagem. As simulações mostraram que, em um modelo de simulação baseado em comunidades, o PSONET é capaz de alcançar uma taxa de entrega de mensagens superior e uma menor relação de redundância de mensagens se comparado aos protocolos Epidêmico e PROPHET. Isso se deve ao fato do PSONET interpretar as condições de comunicação da rede, adaptando-se aos recursos disponíveis e sendo assim um protocolo adaptativo. Como trabalhos futuros, pretende-se analisar o desempenho do PSONET em outros cenários de simulação.

AGRADECIMENTOS

Agradecimentos à Universidade Tecnológica Federal do Paraná (UTFPR) pela estrutura, à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao CNPq pelas concessões de bolsa de estudo e pesquisa.

REFERÊNCIAS

- [1] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, November 2006.
- [2] C.-M. Huang, K.-c. Lan, and C.-Z. Tsai, "A survey of opportunistic networks," in *Proceedings of the 22Nd International Conference on Advanced Information Networking and Applications - Workshops*, ser. AINAW '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 1672–1677.
- [3] K. S. Phanse and J. Nykvist, "Opportunistic wireless access networks," in *Proceedings of the 1st International Conference on Access Networks*, ser. AccessNets '06. New York, NY, USA: ACM, 2006.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, 1995, pp. 1942–1948 vol.4.
- [5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 27–34.
- [6] E. Jones, L. Li, J. Schmidtke, and P. Ward, "Practical routing in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 8, pp. 943–959, Aug 2007.
- [7] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003.
- [8] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep., 2000.
- [9] L. N. De Castro, *Fundamentals of natural computing: basic concepts, algorithms, and applications*, ser. Computer and Information Science Series. Boca Raton, Florida, USA: Chapman & Hall/CRC, 2006.
- [10] L. Rosati, M. Berioli, and G. Reali, "On ant routing algorithms in ad hoc networks with critical connectivity," *Ad Hoc Networks*, vol. 6, no. 6, pp. 827–859, 2008.
- [11] R. J. La and P. Ranjan, "Ant-based adaptive message forwarding scheme for challenged networks with sparse connectivity," in *Proceedings of the 28th IEEE Conference on Military Communications*. Boston, Massachusetts, USA: IEEE, 2009, pp. 1–7.
- [12] A. C. K. Vendramin, A. Munaretto, M. R. Delgado, and A. C. Viana, "GrAnt: Inferring best forwarders from complex networks' dynamics through a greedy ant colony optimization," *Comput. Netw.*, vol. 56, no. 3, pp. 997–1015, Feb. 2012.
- [13] M.-W. Zhang, X.-M. Sun, and X.-Y. Lv, "A QoS routing algorithm based on culture-ant colony algorithm," in *Proceedings of the International Conference on Computer Application and System Modeling*. Taiyuan, China: IEEE, 2010, pp. V12–198–V12–201.
- [14] A. C. B. K. Vendramin, A. Munaretto, M. Delgado, and A. C. Viana, "Cgrant: A swarm intelligence-based routing protocol for delay tolerant networks," in *Proceedings of the 2012 Genetic and Evolutionary Computation Conference*, Philadelphia, USA, 2012, pp. 33–40.
- [15] A. Toofani, "Article: Solving routing problem using particle swarm optimization," *International Journal of Computer Applications*, vol. 52, no. 18, pp. 16–18, August 2012, published by Foundation of Computer Science, New York, USA.
- [16] B. Naveh, "Kshortestpaths. jgrapht, a free java graph library." July 5 2007. [Online]. Available: <http://jgrapht.org/>
- [17] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.
- [18] T. Hossmann, T. Spyropoulos, and F. Legendre, "Putting contacts into context: Mobility modeling beyond inter-contact times," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '11. New York, NY, USA: ACM, 2011, pp. 18:1–18:11.
- [19] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. New York, NY, USA: ICST, 2009.