Extensão do Tempo de Vida de Redes *Ad Hoc*

R.I. da Silva, J.C.B. Leite, M.P. Fernandez

Instituto de Computação
Universidade Federal Fluminense (UFF)
{rsilva, julius, marcial}@ic.uff.br

Abstract. *This paper presents a new algorithm, Extra, for extending the lifetime of ad hoc networks. Extra tries to conserve energy by identifying and switching off nodes that are momentarily redundant for message routing in the network. Extra is independent of the underlying routing protocol and uses solely information that is collected locally. Simulation studies conducted have shown promising results.*

Resumo. *Esse artigo apresenta um novo algoritmo, Extra, para a extensão do tempo de vida em redes ad hoc. Extra procura conservar energia através da identificação e desligamento dos nós que possam ser momentaneamente redundantes para o roteamento de mensagens na rede. Extra é independente do protocolo de roteamento escolhido e utiliza somente informações coletadas localmente. Simulações realizadas demonstraram resultados promissores.*

1. Introdução

Em anos recentes, as redes *ad hoc* de computadores têm recebido atenção crescente. Estas são redes sem topologia definida, cujos nós são móveis e comunicam-se através de canais de rádio. Não havendo, normalmente, nenhum elemento centralizador, todos os nós da rede devem cooperar para que haja o adequado roteamento de mensagens. Diversos protocolos de roteamento de mensagens em redes *ad hoc* foram propostos na literatura, e.g., DSDV [Perkins e Bhagwat 1994], DSR [Johson e Maltz 1996], TORA [Park e Corson 1997] e AODV [Perkins e Royer 1999]. As aplicações dessas redes são variadas, muitas delas na área militar. Contudo, aplicações em áreas civis podem ser visualizadas, como, por exemplo, o seu emprego por equipes de emergência atuando em áreas de desastre. Pode-se imaginar a situação de agentes dispersos em uma área geográfica onde haja necessidade de trocar informações visuais sobre algum tipo de estrago, de forma a melhor combater os problemas gerados por uma emergência. Um exemplo de tal situação seria a monitoração de uma região onde tivesse havido o rompimento de um duto de transporte de petróleo e onde parte do óleo atingisse um rio, daí espalhando-se para outros pontos.

Em função das características dos dispositivos móveis utilizados, normalmente dependentes de bateria, um dos aspectos mais importantes a serem tratados nas redes *ad hoc* é o consumo de energia. Dessa forma, além das métricas tradicionais de comparação de protocolos, como taxa de perda de pacotes, outras, como tempo de vida da rede, devem ser avaliadas. Os nós de uma rede *ad hoc* consomem energia não só quando estão transmitindo e recebendo mensagens, mas também quando se encontram em situação de escuta ociosa. Esse último caso ocorre em função da necessidade da eletrônica do rádio

estar energizada para que o nó mantenha a capacidade de receber mensagens. Diversos estudos indicam que a potência necessária para as funções de transmissão, recepção e escuta é, tipicamente, da ordem de 1,40W, 1,00W e 0,83W, respectivamente [e.g., Chen *et alli* 2002]. Contudo, o mesmo estudo acrescenta que, se o rádio for colocado em modo de dormência, esse consumo cai para 0,13W (outros autores indicam consumos ainda menores). Esses valores indicam que, para haver uma efetiva diminuição no gasto de energia da bateria, o rádio deverá ser colocado em dormência por períodos de tempo, durante a operação da rede.

Uma interessante comparação sobre o consumo de energia de diversos protocolos para rede *ad hoc* foi realizada em [Xu *et alli* 2001] (ver Figura 1). Nesse estudo, os autores observaram o comportamento dos protocolos AODV, DSR, DSDV e TORA, quanto ao consumo de energia, em duas situações diferentes: considerando a energia consumida no estado de escuta ociosa (barras escuras) e não considerando essa energia (barras claras). Da figura, nota-se que os quatro protocolos apresentam aproximadamente o mesmo comportamento em termos de gasto de energia quando a degradação causada pela escuta ociosa é levada em consideração. Dessa forma, percebe-se a importância de minimizar tais períodos, embora tendo-se em mente a necessidade de garantir a conectividade na rede ou de, pelo menos, tentar manter as métricas de desempenho em valores aceitáveis em relação à situação com todos os nós energizados.

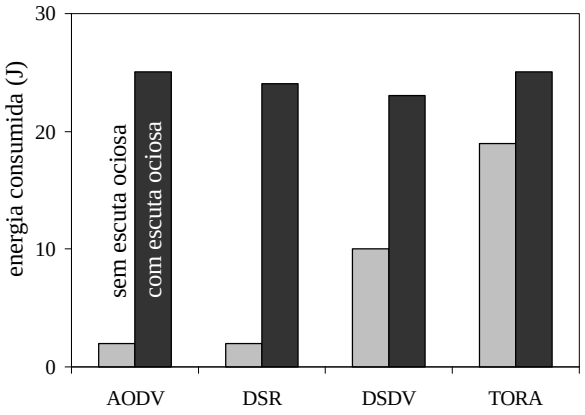

Figura 1. Comparação da energia consumida por quatro protocolos de roteamento em redes *ad hoc*

Nesse trabalho, é apresentado um algoritmo para tratar o problema da coordenação da ação de entrada em estado de dormência, para os nós de uma rede *ad hoc*. Esse algoritmo permite que cada nó da rede tome decisões de forma autônoma, com base em informações obtidas localmente. Adicionalmente, ele pode funcionar em conjunto com qualquer protocolo de roteamento para redes *ad hoc*.

2. Trabalhos Relacionados

A idéia de desligar o rádio-transmissor para conservar energia em redes *ad hoc* já foi explorada por outros autores. No algoritmo BECA (*Basic-Energy Conserving Algorithm*) [Xu *et alli* 2000], o objetivo é minimizar o consumo de energia mantendo o rádio desligado pelo máximo período possível, suportando uma latência mais elevada,

mas tendo um menor gasto de energia. Se o nó irá dormir ou não é uma decisão tomada puramente em bases locais. Inicialmente, o nó encontra-se no estado *Dormindo* e aí permanecerá por um tempo determinado, ao fim do qual transiciona para o estado *Escutando*. Nessa situação, se não notar nenhum tráfego, após determinado período, ele retorna ao estado anterior; caso contrário, transiciona para o estado *Ativo*. Ainda no estado *Dormindo*, se o nó tiver mensagens a transmitir, ele passa para o estado *Ativo*. Nesse último estado, ele permanecerá até ocorrer um determinado número de unidades de tempo (parametrizável) sem nenhuma transmissão. Após esse intervalo, o nó toma a decisão de dormir.

No algoritmo AFECA (*Adaptive Fidelity Energy-Conserving Algorithm*) [Xu *et alli* 2000], cada nó usa uma estimativa do número de nós existentes em sua vizinhança para aumentar o tempo em que permanecerá no estado *Dormindo*. Esse tempo é obtido através de uma variável aleatória uniformemente distribuída no intervalo (*1, número de nós vizinhos*) multiplicada por uma constante. Com o aumento dessa densidade de nós, mais energia pode ser economizada. Através da escolha de determinadas constantes, o algoritmo procura aumentar a probabilidade de que os nós em escuta formem um grafo conexo, de forma a manter a capacidade da rede de rotear pacotes.

Um outro algoritmo, GAF (*Geography-informed Energy Conservation for Ad Hoc Routing*) [Xu *et alli* 2001], produzido pelo mesmo grupo, utiliza informação de posicionamento geográfico (e.g., via GPS) para suportar o mecanismo de conservação de energia. Da mesma forma que AFECA e BECA, o GAF é independente do protocolo de roteamento utilizado. Nesse algoritmo, a área onde a rede estará atuante é dividida em uma malha virtual, composta por quadrados de dimensão fixa, independentemente da densidade de nós existentes. Os nós em cada um desses quadrados transicionam entre os estados *Dormindo* e *Ativo*, mas garantindo que, em cada instante, um deles esteja no estado *Ativo*, de forma a manter a conectividade da rede. Isso é possível, porque cada um dos nós pode saber em qual dos quadrados se encontra. A escolha de qual nó ficará ativo é feita de forma distribuída, através de um sistema de ordenação. Para tanto, um conjunto de regras foi estabelecido tendo como base informações sobre o estado do nó, como, por exemplo, o tempo esperado de vida de sua bateria. Simulações realizadas pelos autores mostram que esse algoritmo possibilita uma significativa extensão da vida da rede em relação ao uso do AODV puro, com pouca degradação em termos de perda de pacotes e tempo médio de entrega de pacotes em relação a este protocolo.

Em [Xu *et alli* 2003], é apresentado o CEC (*Cluster-based Energy Conservation Algorithm*). Nesse trabalho, os autores procuram evitar a dependência que o GAF apresenta em relação a informações de localização, como a fornecida por um GPS. Como é sabido, esse dispositivo não funciona em muitas situações, como em interiores e embaixo de árvores. Assim como também indicado em [Koushanfar *et alli* 2003], os autores reconhecem que o GAF, por fazer uma estimativa de conectividade (ou seja, não uma medida), é muito conservador na definição do número de nós que devem ficar acordados, conseqüentemente, aumentando o gasto de energia. Em CEC, os nós são organizados em *clusters*, cada um com um *cluster-head*, onde cada nó pode ser alcançado em não mais que dois saltos. Nós que pertençam a mais de um *cluster* atuam como *gateways*; quando há redundância, um deles será suprimido para haver economia de energia. Tanto os *cluster-heads* quanto os *gateways* são escolhidos de forma distribuída pelo grupo. Os nós que não foram escolhidos como *cluster-heads* ou

gateways são desligados para economia de energia. De tempos em tempos os nós que estavam dormindo acordam e um novo processo de escolha é realizado. Experimentos realizados pelos autores demonstram, entre outras conclusões, que CEC tem melhor desempenho em termos de sobrevivência da rede que GAF; que em caso de alta mobilidade GAF é melhor do que CEC; e que CEC tem melhor desempenho em situações de baixa mobilidade.

Os autores de *Span* [Chen *et alli* 2002] afirmam que uma rede *ad hoc* que leve em consideração a conservação de energia deve apresentar duas características. A primeira é que a técnica empregada deve permitir a tantos nós quanto possível desligarem seus rádios por um máximo de tempo. A segunda é que essa rede deve ter a capacidade de rotear pacotes com um mínimo de retardo, quando comparada ao caso de todos os nós estarem acordados. Isso implica que um determinado número de nós esteja acordado para formar um *backbone* conectado. Além disso, a capacidade oferecida por esse *backbone* deve ser suficiente para evitar congestionamentos. Para tanto, o algoritmo de *Span* elege nós, denominados *coordenadores*, que formam um conjunto dominante conexo do grafo formado pelos nós da rede *ad hoc*. Um conjunto dominante conexo *S* de um grafo *G* é um subgrafo conexo quando todo vértice *u* em *G*, ou está em *S* ou é adjacente a algum vértice *v* de *S*. Para suportar esse processo, o algoritmo prevê que cada nó tenha informações que indiquem, por exemplo, não somente quem são os seus vizinhos, mas também quem são os vizinhos de seus vizinhos. De forma a garantir que todos os nós compartilhem a tarefa de manter a conectividade global, é realizada uma rotação entre os nós atuantes como coordenadores.

Segundo [Koushanfar *et alli* 2003], os diversos trabalhos que discutiram as condições a que um nó deve atender para dormir, usando somente informações disponíveis localmente, e ainda preservando a conectividade da rede, são todos baseados em heurísticas, sem uma prova formal. No trabalho referenciado, os autores apresentam e demonstram um teorema com as condições necessárias e suficientes para um nó desligar o seu rádio, mantendo a conectividade da rede. De maneira que as informações para a tomada de decisão de entrada no estado de dormência estejam disponíveis, cada nó acordado, periodicamente, difunde uma mensagem que inclui, entre outros campos, a sua identificação, seu nível corrente de energia e uma lista de seus vizinhos correntes. Além disso, como o algoritmo prevê a formação de grupos de vizinhos, é também transmitida a identificação do grupo a que pertence o nó. Com base nessas informações, um nó pode ir dormir se e só se existir um caminho que passe por todos os seus vizinhos acordados e se todos os seus vizinhos que estão dormindo tiverem pelo menos um vizinho acordado. Os autores afirmam que esse algoritmo tem melhor desempenho em termos de economia de energia que o *Span*, para qualquer densidade de nós.

O algoritmo apresentado nesse trabalho, diferentemente de GAF, não utiliza nenhum equipamento para obtenção de posicionamento geográfico. Também de forma distinta a outros algoritmos anteriormente indicados, aqui não há a necessidade do conhecimento das vizinhanças dos nós vizinhos para a tomada de decisão, reduzindo assim o tráfego de mensagens de controle. Além disso, também não faz uso de algoritmos distribuídos de eleição, sendo todos os nós indistinguíveis, não havendo necessidade de nós especiais de coordenação ou centralização.

Nas próximas seções, serão apresentados o algoritmo Extra, um estudo comparativo de seu desempenho, as conclusões e algumas indicações de trabalhos que darão seqüência ao apresentado nesse texto.

3. O Algoritmo *Extra*

Nessa seção, será descrito o algoritmo *Extra* para implementação de mecanismo de economia de energia em redes *ad hoc*. Um ponto a ser notado é que esse algoritmo pode ser implementado sobre qualquer protocolo reativo, exigindo para isso poucas alterações no protocolo. Outro ponto importante é que o mecanismo só utiliza informações que podem ser obtidas localmente em cada nó, e também não é necessária qualquer comunicação com a aplicação.

O procedimento de economia de energia, conforme sugerido na seção 1, é baseado na colocação em estado de dormência de nós da rede. Ou seja, há o desligamento do rádio do nó, evitando assim a escuta ociosa, que consome praticamente tanta energia quanto a transmissão e a recepção de mensagens.

A principal característica deste mecanismo de economia de energia, e que o difere de outros propostos, é o momento de decisão e a duração do estado de dormência. Nesse algoritmo, em determinados instantes, cada nó terá uma probabilidade P, escolhida com base em heurísticas, de iniciar o procedimento de economia de energia. Sendo essa decisão tomada de forma distribuída, alguns nós entrarão em estado de dormência por um período conhecido, enquanto outros permanecerão ativos. Em redes altamente povoadas, tem-se então uma chance maior de que somente alguns nós permaneçam ativos, pouco afetando a conectividade da rede, enquanto outros economizam energia. Quando a densidade diminui, precauções são tomadas para reduzir a probabilidade de desconexões.

3.1. Operação do Algoritmo

Durante sua operação, um nó poderá estar em um de três estados: *Ativo*, *Escutando* e *Dormindo*, conforme indicado na Figura 2. Um nó inicia operação enviando uma mensagem de *hello*, informando a seus vizinhos que está no estado *Ativo*. Neste estado, o nó tem seu rádio ligado e por isso exerce normalmente todas as suas funções. Ele permanecerá assim durante T_a segundos. Se durante este período ele receber um pacote de dados (como o destino, ou parte de uma rota) ou uma resposta de rota, a contagem do tempo T_a é reiniciada. Decorrido esse tempo, e não tendo recebido nenhum tipo de mensagem, o nó executa o procedimento que determinará, com probabilidade P, se ele dormirá ou permanecerá ativo. Se a decisão for que o nó deva permanecer ativo, um outro ciclo de T_a unidades de tempo é iniciado; caso contrário, o nó inicia o procedimento de economia de energia, enviando a seus vizinhos uma mensagem de *hello*, contendo a informação de que seu estado será mudado para *Dormindo*. Isso permite que, durante toda a operação da rede, cada nó mantenha uma estimativa não só do seu número de vizinhos, mas também do número de vizinhos em estado ativo.

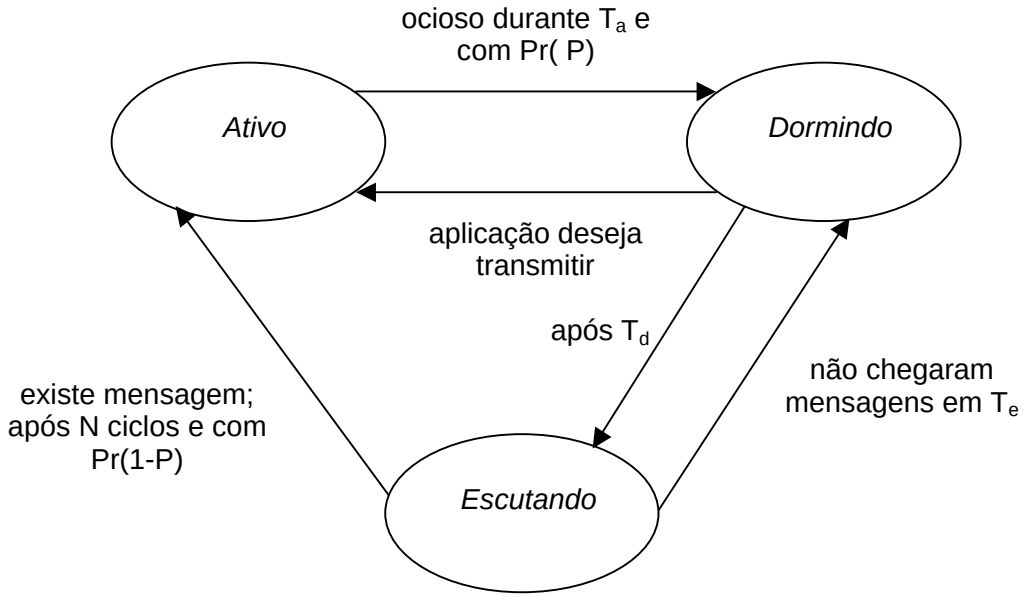

Figura 2. Transições de estado de um nó em *Extra*

Um nó permanece no estado *Dormindo* por T_d segundos. Ao final deste tempo, o nó muda seu estado para *Escutando*. Nessa situação, o rádio do nó é ligado, escuta o meio e verifica se existe alguma mensagem destinada a ele ou de difusão. Se isto ocorrer, o nó reinicia todo o ciclo, indo para o estado A*tivo*. Caso não haja mensagens durante um período de T_e segundos, o nó volta para o estado *Dormindo*. Este procedimento é repetido N vezes (ver exemplo na Figura 3). Depois disso, a função que define se o nó irá dormir é executada novamente e, de acordo com seu resultado, o nó volta ao estado A*tivo* ou inicia outra vez o procedimento de economia de energia. Um nó deve reiniciar o mecanismo sempre que receber ou transmitir uma resposta de rotas, receber um pacote destinado a ele ou retransmitir um pacote por uma rota. A qualquer momento, se a aplicação de um determinado nó necessitar fazer o envio de uma mensagem, este nó deve também reiniciar o mecanismo.

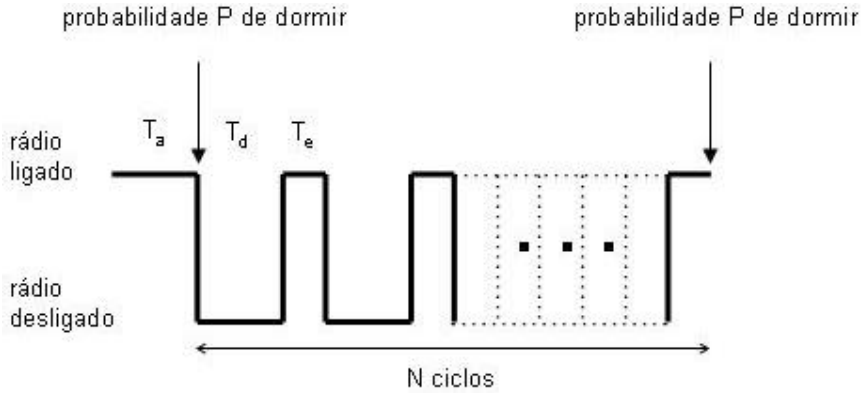

Figura 3. Exemplo de evolução temporal

Quando um nó ativo desejar transmitir um pacote destinado a um vizinho, a transmissão deve ser efetuada somente quando este encontra-se no estado *Escutando* ou no estado *Ativo*. Se tal vizinho estiver no estado *Dormindo*, a transmissão deverá ser atrasada até que o mesmo mude de estado. Com o intuito de fazer o controle de vizinhos e dos seus respectivos estados, cada nó possui um mecanismo de *cache* que é atualizado a partir dos pacotes recebidos. Um novo vizinho é inserido nesta *cache* quando o nó recebe uma requisição de rotas ou um *hello* dele oriundo. As informações de todos os nós vizinhos são sempre salvas junto com uma etiqueta de tempo do instante em que foram adquiridas e são atualizadas sempre que os nós recebem novas mensagens de seus vizinhos. Se, passado um intervalo, nenhuma nova mensagem for recebida de um determinado vizinho, as informações a ele associadas são consideradas obsoletas e retiradas da *cache*. O tempo de obsolescência das informações da *cache* deve ser igual ao tempo de vida de uma rota no protocolo sobre o qual o algoritmo aqui discutido está sendo implementado. Como mencionado no parágrafo anterior, um nó, ao receber um pacote de dados, reinicia o mecanismo indo para o estado A*tivo*. Por isso, quando um nó vai transmitir uma mensagem para um de seus vizinhos, a informação na *cache* sobre este vizinho também é atualizada.

3.2. Obtenção da Probabilidade P

O procedimento que calcula a probabilidade P é o responsável pela decisão de disparar o mecanismo de economia de energia. Os nós sempre têm uma probabilidade P de iniciar a economia de energia, contudo o valor de P pode variar entre implementações ou mesmo em tempo de execução. Esta variação é importante pois permite um ajuste adaptativo do mecanismo a vários tipos de rede, às variações de topologia, à energia corrente dos nós e às variações de cenário.

O objetivo de variar o valor de P é ajustar o mecanismo de economia de energia de acordo com as condições que a rede apresenta no momento. Para essa primeira implementação do algoritmo, somente informações referentes ao grau de conectividade e ao nível de energia do próprio nó são levadas em consideração. Simplificadamente, P deverá ser mais elevada quando o nível corrente de energia do nó, em relação à sua energia máxima, for baixo, e também quando o número de vizinhos for alto. Com base nessa última observação, algumas heurísticas foram desenvolvidas e são explicadas na seqüência.

Heurística 1: sua característica é tentar preservar a conectividade da rede. Assim, um valor maior de P e, conseqüentemente, uma maior probabilidade de dormir, é atribuído aos nós que estão em uma área de alta densidade e um valor menor aos nós que estão em áreas de baixa densidade. O valor de P é definido pela multiplicação do número de vizinhos do nó por uma constante K. Contudo, para evitar que todos os nós de uma região muito povoada durmam ao mesmo tempo, o valor de P é limitado por uma outra constante, L. Dessa forma:

$P(h_1) = K * numero\_vizinhos( )$
Se $(P(h_1) > L)$, então $P(h_1) = L$

onde *numero_vizinhos( )* é uma função que retorna o número de vizinhos do nó. Nos testes realizados para esse trabalho os valores escolhidos foram K = 0,1 e L = 0,9.

Heurística 2: sua característica também é tentar preservar a conectividade da rede. Nesse caso, o valor de P será a razão entre o número de vizinhos ativos e o número total de vizinhos do nó.

Heurística 3: o objetivo principal dessa heurística é economizar energia onde ela é mais escassa, apesar do efeito que possa provocar na conectividade. Aqui, o valor de P é obtido pela relação entre nível de energia atual e o nível de energia inicial do nó. Essa heurística pode ser vista como a do comportamento do egoísta. Dessa forma:

$P(h_3) = 1 - energia\_atual( ) / energia\_inicial( )$

onde *energia_atual( )* é uma função que retorna a quantidade de energia ainda armazenada pelo nó e *energia_inicial( )* retorna a energia inicial do nó.

Heurística 4: esta heurística é uma combinação da heurística 1 com a heurística 3. Ou seja, ela busca economizar energia balanceando os critérios de conectividade da rede e conservação da energia do nó. Sua definição é a seguinte:

se $(P(h_1) \geq ref)$ e $(P(h_3) \geq ref) \Rightarrow P(h_4) = \max(P(h_1), P(h_3))$
se $(P(h_1) \geq ref)$ e $(P(h_3) < ref) \Rightarrow P(h_4) = P(h_1)$
se $(P(h_1) < ref)$ e $(P(h_3) \geq ref) \Rightarrow P(h_4) = P(h_3)$
se $(P(h_1) < ref)$ e $(P(h_3) < ref) \Rightarrow P(h_4) = \min(P(h_1), P(h_3))$

onde $P(h_1)$, $P(h_3)$ e $P(h_4)$ são os valores de P retornados pelas heurísticas 1, 3 e 4, respectivamente, e *ref* é um parâmetro ajustável, cujo valor atribuído nas implementações foi 0,5.

Heurística 5: tem objetivo similar ao da heurística 4. É a combinação da heurística 2 e da heurística 3, também buscando economizar energia onde ela está mais em falta, embora tentando manter a conectividade da rede. Sua definição é a seguinte:

se $(P(h_2) \geq ref)$ e $(P(h_3) \geq ref) \Rightarrow P(h_5) = \max(P(h_2), P(h_3))$
se $(P(h_2) \geq ref)$ e $(P(h_3) < ref) \Rightarrow P(h_5) = P(h_2)$
se $(P(h_2) < ref)$ e $(P(h_3) \geq ref) \Rightarrow P(h_5) = P(h_3)$
se $(P(h_2) < ref)$ e $(P(h_3) < ref) \Rightarrow P(h_5) = \min(P(h_2), P(h_3))$

Finalmente, em todos os testes realizados, os valores atribuídos a N, T_a, T_e e T_d foram, respectivamente, 15, 4s, 0,05s e 0,5s.

4. Avaliação de Desempenho

4.1. Ambiente de Simulação

O algoritmo proposto na seção 3 foi avaliado através da técnica de simulação. O objetivo foi validar o algoritmo *Extra* para um grande número de cenários de redes *ad hoc*. Para avaliar o desempenho dessa proposta, foram feitas comparações com o protocolo AODV padrão, aqui admitido como o protocolo de referência, e com o algoritmo GAF básico, executando com protocolo AODV. Para validar o modelo de simulação, cada figura de mérito foi medida em 10 cenários diferentes, e o valor médio dessas medidas foi apresentado com intervalo de confiança de 95%.

O simulador utilizado nos testes foi o *Network Simulator - ns-2*, versão 2.26 [NS-2]. Para a representação da rede *ad hoc*, foi utilizado o pacote de roteamento

desenvolvido pela CMU, que faz parte da distribuição padrão do simulador *ns*. Essa mesma distribuição também inclui o código necessário à execução do GAF. No nível de enlace é assumida a interface 802.11 (DCF, 2Mbps). Para criar o cenário de simulação, foi utilizado o programa *BonnMotion*, versão 1.1 [BonnMotion 2003].

O cenário de simulação consistiu de 60 nós, movendo-se de forma aleatória (modelo *random way-point*) em um plano de 1200x600m. Foram definidos 4 nós estáticos para geração e consumo de tráfego CBR, posicionados nas extremidades do plano de simulação. A razão de se colocarem as fontes e drenos de tráfego em posição fixa é possibilitar a avaliação de medidas de forma mais coerente, já que os nós intermediários servirão exclusivamente para roteamento de mensagens. As situações com fontes e drenos móveis em cenários aleatórios tornam as medidas pouco apropriadas para uma avaliação, já que a variabilidade dos resultados tenderá a ser muito elevada.

As velocidades atribuídas foram 0 m/s, representando uma rede estática, usada como referência, 1 m/s, representando o equivalente a uma pessoa andando, e 10 m/s, representando um veículo em ambiente urbano (36 Km/h).

Uma importante razão na escolha do programa *BonnMotion* para criar os cenários é que ele permite o estabelecimento de velocidade fixa para os nós móveis. Isto é, ao definir a velocidade de 10 m/s, o nó somente se move a esta velocidade. Outra alternativa seria o programa *Setdest*, integrante do pacote *ns*. Esse programa, porém, implementa a velocidade como uma distribuição de valores entre 0 e a velocidade especificada. Ou seja, ao definir a velocidade de 10 m/s com o uso do *Setdest*, isso implica que o nó se movimenta com uma velocidade escolhida entre 0 e 10 m/s. A utilização do programa *Setdest* pode implicar resultados "melhores", já que a velocidade média será menor que a velocidade estipulada.

O tempo de simulação foi de 900 segundos, o que serviu para comparação com medidas semelhantes realizadas para o algoritmo GAF em [Xu *et alli* 2001]. Para os cenários empregados, foram utilizados tempos de pausa de 0, 30, 60, 120, 300, 600 e 900 segundos, que permitem representar situações variando do movimento contínuo (pausa 0s) até aquela de um nó com muito pouco movimento (pausa de 900s). O modelo de rádio utilizado admite um alcance de 250m, sendo que o modelo de propagação empregado foi o *two-ray-ground*.

Para o modelo de tráfego, foram implementadas duas fontes e dois drenos colocados nos nós fixos. As fontes de tráfego escolhidas foram do tipo CBR sobre transporte UDP. O tamanho dos pacotes utilizados foi fixado em 512 octetos. As taxas avaliadas foram 1 pacote/s, 10 pacotes/s e 20 pacotes/s, totalizando em cada simulação taxas de 2, 20 e 40 pacotes/s.

O modelo de energia escolhido é baseado nas medidas de [Stemm e Katz 1997] da placa WaveLAN 2 Mbps. Os valores obtidos no mencionado trabalho foram 1,6W para transmissão, 1,2W para recepção, 1,0W em escuta ociosa (*idle*) e 0,025W para o estado dormindo (*sleep*). Esses valores são os mesmos empregados na avaliação do GAF em [Xu *et alli* 2001], de forma a facilitar comparações. Como o consumo de energia no estado dormindo é muito menor que os demais, e para simplificar a implementação do algoritmo no *ns*, que não considera esse tipo de consumo na versão

distribuída, foi considerado que quando um nó é desligado o consumo de energia é nulo. Sendo isso feito para todos os algoritmos avaliados, a simplificação não implicará benefício relativo para nenhum deles.

A energia inicial atribuída a cada nó foi de 500J, o suficiente para manter a rede funcionando por um período de aproximadamente 450s para o protocolo AODV. Como o tempo de simulação foi de 900s, pode-se claramente observar o comportamento dos mecanismos de economia de energia. Já que o consumo de energia em escuta ociosa é significativo, pôde-se observar que os nós em uma rede com roteamento AODV puro morrem aproximadamente aos 450s, independentemente do tráfego encaminhado.

Não sendo interessante avaliar o comportamento da rede quando um par fonte/dreno de tráfego é desligado por falta de energia, a esses nós foi atribuída energia infinita, o que garante que sempre estarão transmitindo durante o período de simulação. Deve ser citado que uma outra razão para se colocar esses nós nas extremidades do plano é que nessa posição eles não conseguem rotear pacotes diretamente para o dreno. Se esses nós com energia infinita estivessem distribuídos pelo plano de simulação eles poderiam efetuar o roteamento sem a necessidade da participação dos nós com energia finita, desvirtuando as medidas realizadas.

4.2. Resultados Obtidos

Nos resultados aqui apresentados não é considerado nenhum controle de energia para o protocolo AODV e assume-se o GAF executando com AODV. A primeira medida apresentada indica a extensão do tempo de vida da rede obtida pelo algoritmo *Extra*. Na Figura 4 é feita uma comparação entre os desempenhos de AODV, GAF e *Extra*, esse último assumindo cada uma das heurísticas anteriormente definidas. Para a produção desse gráfico foi utilizado um tempo de pausa igual 0, admitiu-se que os nós móveis tinham velocidade de 1m/s e foi mantida uma taxa de 10 pacotes/s para cada uma das fontes. Dessa figura, vê-se claramente o benefício obtido com a utilização do algoritmo proposto.

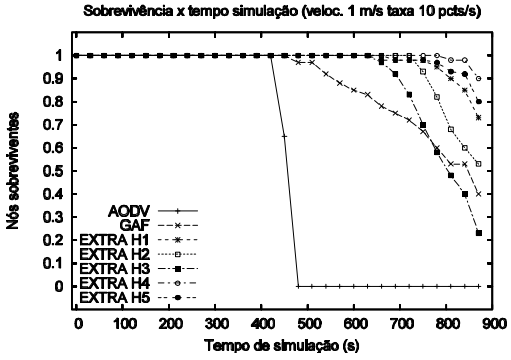

Figura 4. Comparação entre tempos de vida: AODV, GAF e *Extra* (para as várias heurísticas)

A Figura 5 mostra o efeito dos algoritmos GAF e *Extra* no tempo de vida da rede quando se leva em conta variações no tempo de pausa (Figura 5(a)), na taxa de transmissão (Figura 5(b)) e na velocidade dos nós (Figura 5(c)). Em todos esses casos, para permitir uma maior clareza das figuras, os resultados indicados são aqueles obtidos

usando somente a heurística 4. Esses gráficos representam o percentual de nós sobreviventes após 900s. Para esses casos, onde se aplicar, foi utilizada a combinação de pausa igual a 0, taxa de transmissão de 1 pacote/s e velocidade dos nós de 1m/s. Em relação ao tempo de pausa (Figura 5(a)), ambos os algoritmos têm comportamento similar quando esse parâmetro varia, mas *Extra* apresenta melhor desempenho. A Figura 5(b) indica que GAF tem um comportamento que deteriora à medida em que a taxa de transmissão aumenta e depois torna-se relativamente insensível a variações quando a taxa é mais elevada. Já *Extra* apresenta um bom desempenho quando as taxas são baixas, sofre deterioração à medida em que a taxa aumenta mas, em qualquer caso, tem desempenho superior ao GAF. Na Figura 5(c) vê-se que *Extra* foi pouco sensível a variações de velocidade, para toda a faixa de velocidades baixas, tendo um desempenho superior ao GAF em praticamente todo o intervalo indicado. Contudo, em velocidades mais altas, a noção de localidade fica afetada e Extra piora gradativamente seu desempenho relativamente ao GAF.

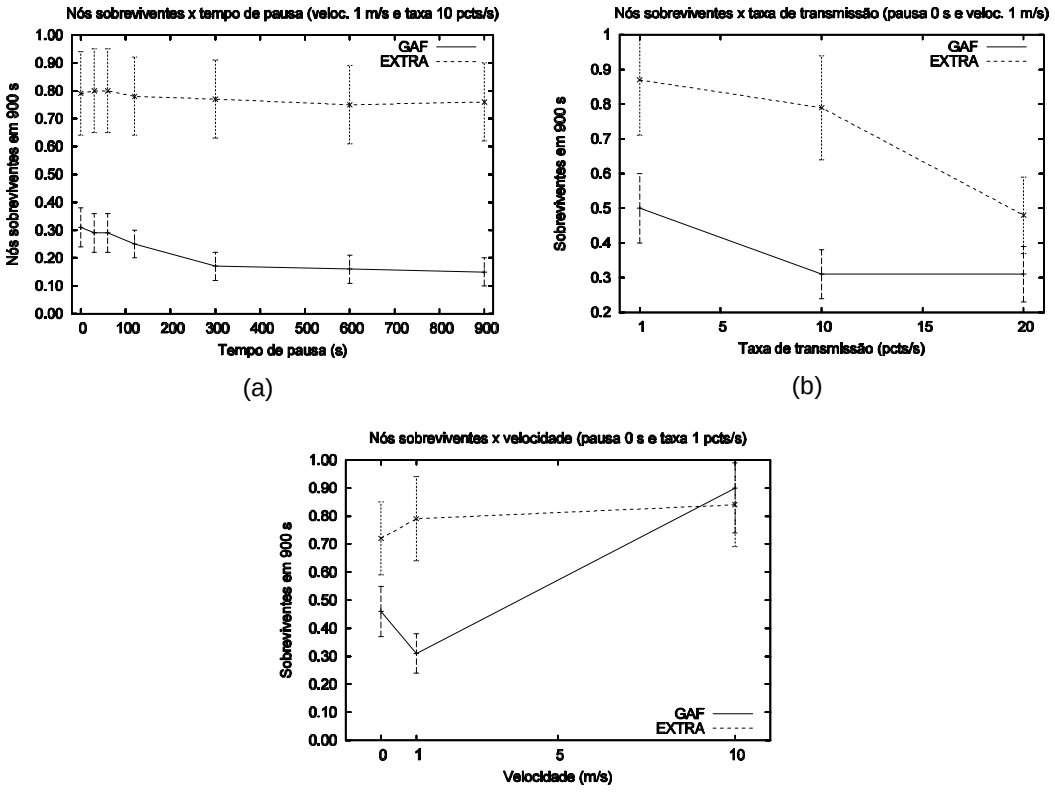

Figura 5. *Extra* e GAF: percentual de nós sobreviventes *versus* pausa, taxa de transmissão e velocidade dos nós

As Figuras de 6(a) a 6(f) comparam o descarte de pacotes e o percentil 90% do retardo quando há variações no tempo de pausa, na taxa de transmissão e na velocidade dos nós, para os algoritmos GAF e *Extra* (usando a heurística 4). Também para essas figuras, onde se aplicar, foi utilizada a combinação de pausa igual a 0, taxa de transmissão de 1 pacote/s e velocidade dos nós de 1m/s. Em todas elas, os valores só são coletados para tempos inferiores ao tempo de vida da rede AODV. Em relação à

variação do tempo de pausa (Figuras 6(a) e 6(b)), *Extra* e GAF apresentam resultados muito próximos. Já em relação ao comportamento quando há variação na taxa de transmissão, GAF apresenta melhor desempenho do que *Extra* para valores baixos, embora ambos sofram degradação de desempenho à medida em que a taxa aumenta. Finalmente, quando se trata da sensibilidade em relação à velocidade dos nós, *Extra* é superior a GAF em praticamente toda a faixa avaliada.

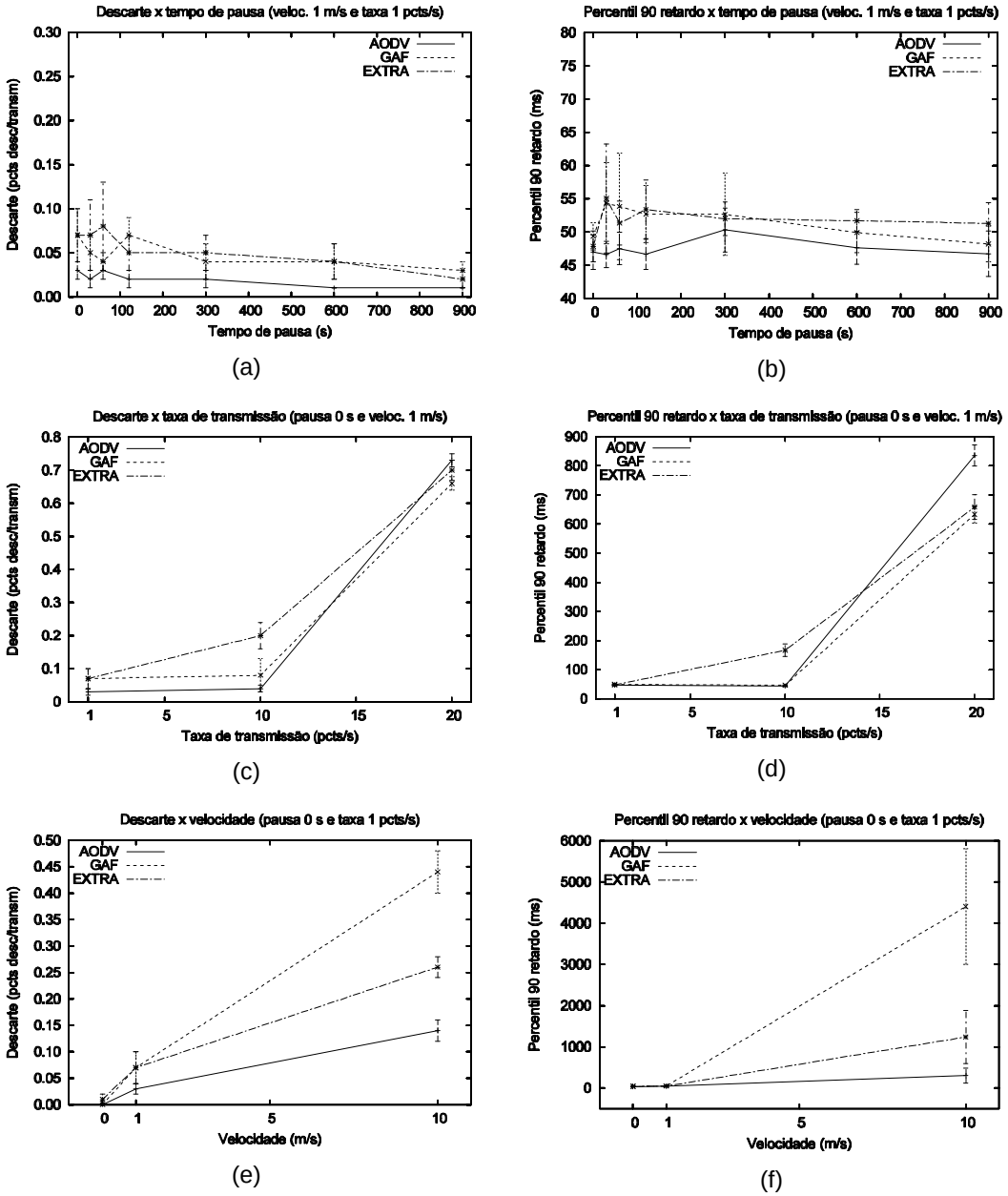

Figura 6. *Extra*, GAF e AODV: descarte de pacotes e retardo de transmissão *versus* pausa, taxa de transmissão e velocidade dos nós

5. Conclusões e Trabalhos Futuros

Esse trabalho apresentou um novo algoritmo, *Extra*, para a conservação de energia dos nós em redes *ad hoc*. Assim como vários outros algoritmos, seu objetivo é maximizar o tempo de vida da rede, mas tentando manter a conectividade. Seu funcionamento é controlado por uma função que define quando o nó deverá entrar em estado de dormência. Diferentemente de outros algoritmos, o tempo de dormência é constante, e a entrada nesse estado ocorre com uma dada probabilidade. Essa probabilidade é calculada em tempo de execução e, para a sua obtenção, heurísticas são utilizadas. Até o momento, cinco heurísticas diferentes foram testadas, todas elas com estrutura muito simples. Os testes até agora realizados mostram que é possível melhorar o desempenho do algoritmo, trabalhando-se na definição dessas heurísticas.

É claro que, sendo a abordagem aqui empregada probabilística, garantias absolutas de funcionamento ótimo não podem ser oferecidas. Contudo, os resultados preliminares obtidos na comparação com o GAF indicam que o mecanismo proposto apresenta vantagens significativas (e.g., informações coletadas são puramente locais, não há uso de GPS). Contudo, deve ser acrescentado, que nessa primeira fase ainda não foi possível fazer uma avaliação completa do algoritmo. Por exemplo, utilizamos tão somente o modelo de mobilidade *random way-point* e ainda temos de avaliar o impacto de variações nos parâmetros de configuração (T_a, T_d, etc.), entre outras tarefas.

Como visto no trabalho, em nenhuma das heurísticas avaliadas são utilizadas outras informações que não aquelas obtidas localmente. Ou seja, como trabalhos futuros, pode-se pensar em soluções onde os nós vizinhos troquem informações sobre a sua energia atual e onde a definição da densidade de vizinhos possa ser mais explorada. Nesse último caso, por exemplo, pode-se obter uma noção de quem são (ou quantos são) os vizinhos dos nós vizinhos. Obviamente, isso implicará em um aumento nas mensagens de controle, o que acarreta um maior gasto e energia. Contudo, é possível que esse efeito negativo não seja suficiente para degradar o aumento de desempenho obtido com a agregação ao algoritmo das informações não locais.

De qualquer forma, os testes realizados mostram a importância de manter a simplicidade na definição dos parâmetros de controle. Uma maior número destes implica em maior complexidade na sua obtenção e também na estrutura do algoritmo. Com base nas simulações, acreditamos que a seleção de um pequeno número desses parâmetros e a escolha de uma heurística simples podem levar à obtenção de efeitos significativos na extensão do tempo de vida da rede, afinal *entia non sunt multiplicanda praeter necessitatem* (princípio, navalha ou divisor de Occam; em uma tradução livre, "não multiplicar as entidades além do necessário", ou, em português coloquial, "vamos manter as coisas simples!").

Agradecimentos

Esse trabalho foi parcialmente apoiado pelo CNPq e pela Capes. Os autores gostariam de agradecer à Profa[a] Irenisia Oliveira por suas sugestões para a melhoria do texto. Finalmente, devemos dar crédito aos revisores anônimos por suas diversas observações, que contribuíram para uma maior clareza do texto final.

6. Referências

BonnMotion (2003) “A Mobility Scenario Generation and Analysis Tool”, http://www.informatik.uni-bonn.de/IV/BonnMotion/.

Chen, B. *et alli* (2002) “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks”, Wireless Networks, vol. 8, p. 481-494.

Johnson, D.B. e Maltz, D.A. (1996) “Dynamic Source Routing in Ad Hoc Wireless Networks”, em Mobile Computing, T. Imielinski e H. Korth (eds.), Kluwer, p. 153-181.

Koushanfar, F. *et alli* (2003) “Low Power Coordination in Wireless Ad-hoc Networks”, ISLPED'03, 25-27 de Agosto, Seul, Coréia, p. 475-480.

NS-2 (2003) “The Network Simulator - ns-2”, http://www.isi.edu/nsnam/ns/.

Park, V. e Corson, S. (1997) “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”, INFOCOM, Abril, p. 1405-1413.

Perkins, C.E. e Bhagwat, P. (1994) “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, ACM Conf. on Communications Architecture, Protocols and Applications, Agosto, p.234-244.

Perkins, C.E. e Royer, E.M. (1999) “Ad Hoc On-Demand Distance Vector Routing”, 2nd IEEE Work. on Mobile Computer Systems and Applications, Fevereiro, p. 90-100.

Stemm, M. e Katz, R.H. (1997) “Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices”, IEEE Trans. on Communications, E80-B(8), Agosto, p. 1125-1131.

Xu, Y. *et alli* (2000) “Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks”, Relatório de Técnico nº 527, USC/ISI, 12 de Outubro, Los Angeles, EUA, 14 p.

Xu, Y. *et alli* (2001) “Geography-informed Energy Conservation for Ad Hoc Routing”, 7th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking, 16-21 de Julho, Roma, Itália, p. 70-84.

Xu, Y. *et alli* (2003) “Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks”, Relatório Técnico nº 6, Center for Embedded Networked Sensing, 23 de Janeiro, Los Angeles, EUA, 18 p.