RONE ILÍDIO DA SILVA

EXTRA – ECONOMIA DE ENERGIA E EXTENSÃO DO TEMPO DE VIDA DE REDES *AD HOC*

Dissertação apresentada ao Curso de Pós-Graduação em Computação Aplicada da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre.

Orientador: JULIUS C. B. LEITE

NITERÓI
2004

Dedico este trabalho a meus pais, irmãos e a Rosângela.

Agradecimentos

Logicamente, inicio agradecendo a Deus. Acredito que suas vontades são realizadas através de seus filhos, ou seja, cada um de nós é uma ferramenta a providência divina. Por isso, agradeço a Deus pelas inúmeras pessoas que passaram pelo meu caminho durante essa jornada e que contribuíram direta ou indiretamente para a conclusão, me ajudando e me dando forças para continuar.

Agradeço a meu Orientador, Prof. Ph.D. Julius Leite. Além do conteúdo técnico transmitido durante todo o mestrado, sua postura com profissional é um exemplo para todos os alunos. Sua dedicação, respeito e compromisso com os trabalhos de seus orientados colaboram para obtenção de ótimos resultados e para, principalmente, a formação de profissionais competentes. Agradeço ao Prof. Dr. Marcial Fernandez que contribuiu diretamente para realização dos trabalhos, principalmente durante a fase de simulações e testes. A todos os professores e funcionários do Departamento de Pós-Graduação em Computação da Universidade Federal Fluminense deixo minha gratidão, em especial aos professores que me dera aula e à secretária Izabela pelas várias colaborações e pela amizade.

A partir deste ponto, torna-se muito difícil escrever este agradecimento, pois com certeza esquecerei o nome de alguém que muito me ajudou. Iniciarei pelos meus pais Waldyr e Hilda, seus ensinamentos são e serão utilizados em todos os instantes de minha vida, o amor dessas pessoas me fortalece e me deixa muito mais feliz. Agradeço, de todo coração à amada Rosângela. Sua presença em meus pensamentos sempre foi um grande motivo para continuar. Nos momentos mais difíceis só de ouvir sua voz meiga, mesmo a distância, eu me sentia mais tranqüilo e forte. Saber que tem alguém que você ama te esperando no destino dá forças durante a jornada. Aos meus irmãos, Sandro, Denise e de tabela "mermão" Frank, agradeço o amor, as conversas e a seus exemplos de garra para viver. A minha avó Julieta, agradeço a todo carinho sempre dispensado a mim. E aos demais familiares, meus sobrinhos Pedro e André, meus tios, tias e primos fica minha gratidão pela força nos momentos difíceis.

Agradecer aos amigos é a parte mais complicada, uma vez que graças a Deus são muitos e esquecer de escrever os nomes de alguns será inevitável. Contudo, meus amigos são sempre lembrados e acredito que me perdoaram não tiverem o nome aqui. Iniciarei pelos companheiros de Mouseville (Sidney, Jonivan, Eduardo, Áthila e André), a colaboração desses amigos foi imprescindível em todos os momentos. Tanto a ajuda pessoal com a ajuda técnica foram importantíssima para conclusão do mestrado. Agradeço aos amigos de Mouseville2 (Alex,

Glauco, Renatinha e os outros que saíram), às meninas da “Casa da Cris” (Cristiane, Idalmis, Ilza e Gabi) e aos amigos de mestrado Adriana (valeu pelo inglês), Allyson, Bruno, Igor, Jonildo.

Finalizando, agradeço a todos os meus irmãos da República Canaan (Ouro Preto – MG) pelos “bondes”, pela amizade e pela ajuda na minha formação pessoal.

Entia non sunt multiplicanda praeter necessitatem

(em uma tradução livre, “não multiplicar as entidades além do necessário”,
ou, em português coloquial, “vamos manter as coisas simples!”)

William of Occam

Sumário

Lista de Figuras iii
Resumo vi
Abstract vii
Capítulo 1 1
1.1) Introdução 1
Capítulo 2 - Economia de Energia em Redes Ad Hoc 5
2.1) Introdução 5
2.2) Algoritmos que adormecem os nós de redes ad hoc 6
2.2.1) GAF 7
2.2.1.1) Divisão da área onde se encontram os nós 7
2.2.1.2) Estados dos nós 8
2.2.2) CEC 10
2.2.2.1) Classificação dos nós 10
2.2.2.2)Princípio de funcionamento 11
2.2.2.3) Características importantes 11
2.2.3) On-demand power management for ad hoc networks 12
2.2.3.1) Transição entre os estados dos nós 12
2.2.3.2) Framework sobre demanda 12
2.2.4) BECA e AFECA 13
2.2.4.1) BECA 13
2.2.4.2) AFECA 14
2.2.5) Span 15
2.3) Algoritmos que buscam enviar pacotes por nós com níveis de energia mais altos . 15
2.3.1) RDRP e MMRP 16
2.3.1.1) RDRP 16
2.3.1.2) MMRP 16
2.3.2) AODV-E 17
2.3.2.2) Mecanismo de descoberta de rotas alterado 17
2.4) Comentários 18
Capítulo 3 - O Algoritmo Extra 19
3.1) Introdução 19

3.2) Operação do algoritmo 19
3.3) Obtenção da probabilidade P 22
3.4) Analise das heurísticas 24
Capítulo 4 – Experimentos 25
4.1) Introdução 25
4.2) Ambiente de simulação 26
4.3) Variação dos parâmetros 28
4.3.1) Variação do parâmetro N 29
4.3.2) Variação do parâmetro Ta 33
4.3.3) Variação do parâmetro Td 37
4.3.4) Variação do parâmetro Te 40
4.3.5) Análise dos parâmetros 44
4.4) Análise das heurísticas 45
4.5) Validando o algoritmo Extra 48
4.5.1) Percentual de nós sobreviventes durante a simulação 49
4.5.2) Percentual de nós sobreviventes ao final da simulação 50
4.5.3) Percentil 90 de retardo..... 53
4.5.4) Descarte de pacotes 56
4.5.5) Número de pacotes úteis por unidade de energia 58
4.5.6) Análise dos experimentos de validação do Extra 61
4.6) Analisando os riscos de desconexão da rede 61
4.6.1) Comportamento da heurística 4 para várias taxas de transmissão 61
4.6.2) Analisando o comportamento de cada heurística 67
4.7) Análise dos experimentos 68
Capítulo 5 - Conclusões e Trabalhos Futuros 69
Capítulo 6 – Referências 71

Lista de Figuras

Figura 1: comparação da energia consumida por quatro diferentes protocolos de roteamento em redes ad hoc 3
Figura 2: divisão de uma área em três sub-áreas quadradas de lado r 8
Figura 3: cálculo de r..... 8
Figura 4: diagrama de estados 9
Figura 5: tipos de nós 10
Figura 6: variação do temporizador keep-alive 13
Figura 7: estados do BECA 14
Figura 8: transições de estado de um nó com o Extra 20
Figura 9: exemplo de evolução temporal 21
Figura 10: cenário criado para os experimentos 27
Figura 11: nós sobreviventes em 900s de simulação por taxa de transmissão 29
Figura 12: percentil 90 de retardo por taxa de transmissão, para cinco valores de N 31
Figura 13: taxa de descarte de pacotes por taxa de transmissão, para cinco valores de N 31
Figura 14: pacotes que alcançaram o destino por unidade de energia x taxa de transmissão, para cinco valores de N 32
Figura 15: porcentagem de nós sobreviventes por taxa de transmissão, para os valores de Ta 34
Figura 16: percentil 90 de retardo por taxa de transmissão, para os valores de Ta 35
Figura 17: taxa de descarte de pacote por taxa de transmissão, para os valores de Ta 36
Figura 18: pacotes que alcançaram o destino por unidade de energia x taxa de transmissão, para os cinco valores analisados de Ta 36
Figura 19: porcentagem de nós sobreviventes por taxa de transmissão de pacotes, para os cinco valores de Td 38
Figura 20: percentil 90 de retardo por taxa de transmissão, para os cinco valores de Td 39
Figura 21: taxa de descarte de pacote por taxa de transmissão para os cinco valores analisados de Td 39
Figura 22: número de pacotes que alcançaram o destino com um Joule por taxa de transmissão, para os cinco valores analisados de Td 40
Figura 23: porcentagem de nós sobreviventes em 900s de simulação por taxa de transmissão de pacotes, para os cinco valores de Te 41

Figura 24: percentil 90 de retardo por taxa de transmissão, para os cinco valores de Te 42
Figura 25: taxa de descarte de pacotes por taxa de transmissão de pacotes, para os cinco valores de Te 43
Figura 26: número de pacotes que alcançaram o destino com um Joule por taxa de transmissão de pacotes, para os três valores de Te 44
Figura 27: porcentagem de nós sobreviventes em 900s de simulação por taxa de transmissão, para as cinco heurísticas 45
Figura 28: percentil 90 de retardo por taxa de transmissão, para as cinco heurísticas 46
Figura 29: taxa de descarte de pacotes por taxa de transmissão, para as cinco heurísticas 47
Figura 30: número de pacotes que alcançaram o destino com um Joule por taxa de transmissão de pacotes, para as cinco heurísticas 48
Figura 31: nós sobreviventes por tempo de simulação para as cinco heurísticas do Extra, para o GAF e para o AODV 50
Figura 32: percentual de nós sobreviventes após 900s de simulação variando-se o tempo de Pausa 51
Figura 33: percentual de nós sobreviventes após 900s de simulação variando-se a taxa de transmissão de pacotes 52
Figura 34: percentual de nós sobreviventes após 900s de simulação variando-se a velocidade dos nós 52
Figura 35: percentil 90 do retardo variando-se o tempo de pausa 54
Figura 36: percentil 90 do retardo variando-se a taxa de transmissão de pacotes 54
Figura 37: percentil 90 do retardo variando-se a velocidade dos nós 55
Figura 38: descarte de pacotes variando-se o tempo de pausa 56
Figura 39: descarte de pacotes variando-se a taxa de transmissão de pacotes 57
Figura 40: descarte de pacotes variando-se a velocidade dos nós 58
Figura 41: número de pacotes úteis por unidade de energia variando-se o tempo de pausa 59
Figura 42: número de pacotes úteis por unidade de energia variando-se a taxa de transmissão de pacotes 60
Figura 43: número de pacotes úteis por unidade de energia variando-se a velocidade dos nós 60
Figura 44: nós ativos em cada instante da simulação, para a heurística 4 e taxa de transmissão de pacotes de 1 pct/s 63

Figura 45: nós dormindo em cada instante da simulação, para a heurística 4 e taxa de transmissão de pacotes de 10 pct/s 64
Figura 46: nós dormindo em cada instante da simulação, para a heurística 4 e taxa de transmissão de pacotes de 20 pct/s 64
Figura 47: nós dormindo em cada instante da simulação, para a heurística 1 e taxa de transmissão de pacotes de 10 pct/s 66
Figura 49: nós dormindo em cada instante da simulação, para a heurística 3 e taxa de transmissão de pacotes de 10 pct/s 67
Figura 50: nós dormindo em cada instante da simulação, para a heurística 5 e taxa de transmissão de pacotes de 10 pct/s 67

Resumo

Esse trabalho apresenta um novo algoritmo, *Extra*, para economia de energia e extensão do tempo de vida em redes ad hoc. *Extra* procura conservar energia através de um mecanismo que faz com que cada nó decida de forma autônoma se continuará ativo ou se entrará em estado de dormência, onde o consumo energético é consideravelmente menor. Além disso, a economia é efetuada de forma a elevar o tempo de vida da rede, ou seja, a aumentar o tempo gasto até que o primeiro nó morra por falta de energia. Todas as informações utilizadas por esse algoritmo são obtidas na camada de roteamento, não sendo necessária qualquer comunicação com outras camadas da pilha de protocolos. Adicionalmente, ele pode ser utilizado em conjunto com qualquer protocolo reativo de redes *ad hoc*, sem a necessidade de alterar o modo de operação de tal protocolo. Devido à sua simplicidade, o *Extra* pode ser implementado em vários tipos de dispositivos, até mesmo naqueles bastante simples e baratos. Através de simulações, foram realizados vários experimentos que comprovaram a eficácia desse algoritmo no controle do consumo de energia. Os resultados obtidos mostram que o *Extra* minimiza o gasto energético dos nós quando comparado a outros mecanismos similares.

Palavras-chave: redes *ad hoc*, economia de energia, protocolos de roteamento, extensão do tempo de vida da rede.

Abstract

This work proposes a new algorithm, called *Extra*, for power saving and *ad hoc* network life time extension. *Extra* is able to save power by allowing each nod to decide whether it showd be active or to changed to the sleeping state, in which the power consumption is significantly lower. The power saving is done between the nodes in order to extend the network life time, that is, extend de period of time consumed until the death of the first node. All used information can be obtained only from the routing level. Besides, *Extra* can be used together with any other ad hoc network reactive protocol and it is not necessary to change its features. As it is very simple, it can be used by any kind of hardware, even the most simple and not expensive ones. The simulations tests show that Extra is very effective in network power saving compared with other similar mechanisms.

Keywords – *ad hoc* networks, energy conserving, routing protocols, extends the network life time

Capítulo 1

1.1) Introdução

Verificou-se nos últimos anos o surgimento de vários trabalhos referentes a redes *ad hoc* de computadores. Esse tipo de rede se caracteriza pelo uso de canais de rádio para promover comunicação entre seus nós e pela ausência de dispositivos fixos destinados a centralizar as comunicações da rede. Com isso, quando um nó se encontra dentro do raio de alcance de outro, e vice-versa, eles podem se comunicar diretamente. Porém, quando dois nós se encontram muito afastados, de forma que não consigam se comunicar diretamente, os demais nós da rede podem participar desta comunicação como intermediários, realizando a função de roteamento das mensagens entre os nós origem e destino da comunicação. Para garantir o roteamento de mensagens entre todos os nós e aumentar a eficiência na comunicação, vários protocolos podem ser encontrados na literatura. Pode-se citar como exemplos DSDV [Perk 94], DSR [John 96], TORA [Park 97] e AODV [Perk 99].

Essas características citadas acima permitem que os nós possam se mover dentro de uma determinada área, alterando a topologia da rede dinamicamente, e ainda manter a conectividade. Tudo isso amplia o número de possíveis aplicações das redes *ad hoc*. Pode-se citar, como exemplo, a utilização de robôs autônomos em operações de reconhecimento e estudo de ambientes inadequados à presença humana. Nesta situação, torna-se necessária a troca de informações entre esses robôs e, para isso, redes *ad hoc* podem ser empregadas. Como outro exemplo de aplicação, 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 sobre redes *ad hoc* é o consumo de energia. De acordo com [Cano 00], nos últimos anos a capacidade de armazenamento de energia das baterias não evoluiu de forma tão significativa quanto a miniaturização e o poder de processamento dos dispositivos semicondutores. Com isso, o consumo da energia deve ser controlado de forma a minimizar os gastos e melhor equilibrar o nível de energia entre os nós da rede. Os estudos realizados em [Jone 01] mostram que a fonte de energia dos componentes de

uma rede sem fio de computadores é utilizada, principalmente, com a manutenção dos recursos visuais (vídeo), com a alimentação do processador, com a manutenção da memória principal e com a interface sem fio. Os protocolos de roteamento de redes *ad hoc* utilizam esta interface em seus mecanismos de descoberta de rotas de várias formas e por isso variam quanto à utilização desse recurso.

Esses protocolos de roteamento se dividem, principalmente, em dois grupos: protocolos pró-ativos e protocolos reativos, [Cama 99]. Os primeiros são os protocolos que mantêm em cada nó informações sobre a topologia de toda a rede, ou seja, mantêm rotas na memória para todos os possíveis destinos da rede. Para isso, periodicamente, cada nó envia a seus vizinhos informações sobre alterações em suas tabelas de rotas, consumindo assim parte da banda de transmissão com tais informações de controle. Pode-se citar como exemplo desse tipo de protocolo o DSDV [Perk 94]. Os protocolos que compõem o segundo grupo só mantêm na memória as rotas que foram requisitadas e que estão sendo utilizadas pelos nós. Novas rotas só são adquiridas quando requisitadas. Esses protocolos economizam a banda de transmissão pois diminuem o número de mensagens de controle que são enviadas. De acordo com os estudos efetuados em [Cano 00], eles também apresentam melhor desempenho no quesito economia de energia, gastando menos energia durante o tempo de operação da rede. Por esta razão, esse trabalho, que trata do consumo de energia de redes *ad hoc*, dedicou-se mais ao estudo de protocolos reativos.

Os dispositivos de comunicação que compõem redes móveis, quando em operação, consomem energia não só quando estão transmitindo e recebendo mensagens, mas também quando se encontram ociosos. Esse último caso ocorre em função da necessidade da eletrônica do rádio estar energizada para que o nó mantenha a capacidade de receber mensagens. Estudos indicam que a potência tipicamente necessária para as funções de transmissão é da ordem de 1,40W, recepção de 1,00W e manutenção em estado ocioso de 0,83W, [Chen 02]. 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. A Figura 1 apresenta uma interessante comparação sobre o consumo de energia de diversos protocolos para rede *ad hoc* que foi realizada em [Xu 01]. As barras claras representam o consumo energético da rede quando o consumo ocioso é levado em consideração; as barras escuras representam o consumo da rede quando o consumo ocioso não é considerado. Pode-se perceber que a energia consumida quando

o nó está ocioso representa uma parte significativa de toda a energia consumida na rede, isso para qualquer tipo de protocolo de roteamento utilizado.

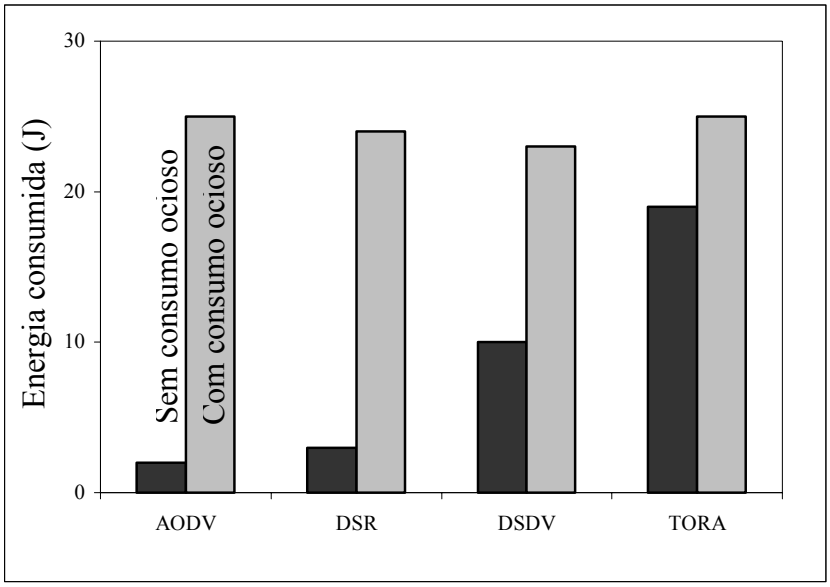

Figura 1: comparação da energia consumida por quatro diferentes protocolos de roteamento em redes *ad hoc*

Nesse trabalho é apresentado *Extra*, 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. Tais informações são adquiridas a partir dos protocolos de roteamento sobre os quais o algoritmo pode ser implementado, não sendo necessária nenhuma comunicação com camadas superiores (aplicação e transporte), inferiores (acesso ao meio e física) ou com outros dispositivos, como GPS. Nenhuma informação que não possa ser adquirida pelos nós em tempo de execução é utilizada. Com isso, a implementação do *Extra* se torna possível em dispositivos bastante simples e baratos.

Adicionalmente, ele pode funcionar em conjunto com qualquer protocolo de roteamento para redes *ad hoc*, sendo que, de acordo com suas características, ele se adapta melhor a protocolos reativos.

A seguir tem-se o estudo do estado da arte cujo enfoque foi economia de energia em redes *ad hoc*. Uma maior atenção foi dada aos trabalhos que tratam do problema na camada de roteamento. No entanto, vários outros trabalhos relacionados também estão sendo apresentados. O capítulo 3 traz a explicação das características e do princípio de funcionamento de todos os mecanismos que compõem o algoritmo *Extra*. No capítulo 4 são apresentados os

resultados dos testes efetuados com esse algoritmo. Tais testes se dividiram em três etapas: a primeira visou encontrar os melhores valores a serem associados aos parâmetros de configuração do *Extra*, a segunda visa comprovar a eficácia desse algoritmo em comparação com outros algoritmos encontrados na literatura e a terceira buscou analisar os riscos de possíveis desconexões na rede que poderiam ser causadas com utilização do *Extra*. No capítulo 5 são apresentadas as conclusões e são discutidos possíveis trabalhos futuros. Finalizando, no capítulo 6 são apresentadas as referências bibliográficas desse trabalho.

A partir das idéias aqui apresentadas foram elaborados artigos científicos que foram publicados no 22º Simpósio Brasileiro de Rede de Computadores (SBRC-2004) [Silv 04a], no 15º Congresso Brasileiro de Automática (CBA-2004) [Silv 04b] e no 11º International Conference on Telecommunications (ICT-2004) [Silv 04c].

Capítulo 2 - Economia de Energia em Redes Ad Hoc

2.1) Introdução

Na literatura pode-se encontrar vários trabalhos referentes ao consumo de energia em redes *ad hoc*. Alguns desses trabalhos apresentam estudos sobre como a energia é consumida. Como exemplos temos [Cano 00] que faz análises e comparações dos comportamentos de alguns protocolos de redes *ad hoc* e [Woo 98] que estabelece métricas que podem ser utilizadas pelos protocolos no roteamento de redes *ad hoc* na tentativa de controlar o consumo de energia. Outros trabalhos, como por exemplo [Xu 01] e [Xu 03], apresentam algoritmos que visam melhorar o consumo de energia. Esses algoritmos podem ser classificados em três categorias, de acordo com seus mecanismos de controle de energia:

⇒ algoritmos que controlam a transição dos nós para o estado de dormência, ou seja, para um estado com menor consumo de energia;

⇒ algoritmos que controlam a potência de transmissão de pacotes, diminuindo-a quando os nós que estão diretamente se comunicando estão separados por distâncias menores; e,

⇒ algoritmos que buscam enviar pacotes por nós com níveis de energia mais altos.

Os algoritmos que controlam a transição dos nós para o estado de dormência tentam evitar o gasto de energia com a escuta ociosa, ou seja, procuram evitar que os nós escutem pacotes que não são endereçados a eles. Além disso, existe a necessidade de que os componentes dos rádios de cada um dos nós permaneçam energizados para que eles mantenham a capacidade de receber transmissões. Isto leva a um gasto energético (consumo ocioso) que pode ser minimizado se os nós puderem se manter em estado de dormência por mais tempo. Pode-se citar como exemplos desse tipo de algoritmo o GAF [Xu 01], o CEC [Xu 03] e o *Span* [Chen 02].

Esta forma de economizar energia se mostra bastante eficiente, realmente diminuindo o consumo de energia em cada nó da rede e, se bem aplicada, aumentando consideravelmente o tempo de vida da rede. Contudo, ela leva a um aumento no tempo de transmissão de pacotes entre a origem e o destino e a um aumento na taxa de descarte de pacotes, o que pode ser observado na maioria dos trabalhos citados.

Os algoritmos que controlam a potência na transmissão de pacotes possuem mecanismos que calculam qual a potência mínima necessária para que uma transmissão, entre dois nós vizinhos, possa ocorrer. Esse tipo de algoritmo diminui a energia gasta em cada

transmissão, conseqüentemente diminuindo seu raio de alcance. Com transmissões alcançando menores distâncias, o número de nós que as escutam tende a cair, diminuindo assim a energia gasta com escuta ociosa. Contudo, redes que utilizam esse tipo algoritmo sofrem aumento na taxa de descarte causada pela movimentação dos nós, pois, como a potência mínima para a transmissão é utilizada, se ocorrer um pequeno afastamento entre nós que estão se comunicando a transmissão não consegue alcançar seu destino e o pacote é descartado.

Um exemplo desse tipo de algoritmo pode ser encontrado em [Jung 02], onde é apresentado um protocolo da camada MAC de redes sem fio de computadores denominado PCM (*Power Control Mac*). Tal protocolo utiliza suas mensagens de controle para calcular qual é a potência mínima necessária para efetuar uma transmissão entre nós vizinhos. Esta potência é então utilizada até o fim da transmissão dos dados.

A terceira categoria é composta por algoritmos que buscam enviar pacotes por nós com maior nível de energia. Esta técnica visa fazer um balanceamento no consumo entre os nós da rede, evitando a morte precoce, por falta de energia, de alguns nós que estão sendo muito utilizados, em detrimento de pouca ou nenhuma utilização de outros. Com isso, o tempo de vida da rede, que é o tempo gasto até que o primeiro nó morra, é aumentado. Porém, o consumo de energia da rede como um todo pode aumentar, pois a rota composta pelos nós com maior nível de energia pode não ser a menor rota, ou seja, com um maior número de nós participando de uma transmissão o consumo global aumenta. Como exemplo desse tipo de algoritmo temos [Cunh 02].

Os estudos efetuados para esse trabalho foram direcionados principalmente para algoritmos que se enquadram na primeira categoria, ou seja, aqueles que controlam a transição dos nós para o estado de dormência, visando principalmente aqueles que trabalham com informações adquiridas na camada de roteamento de redes *ad hoc*. Os algoritmos classificados como da terceira categoria também foram estudados, uma vez que eles também trabalham com informações da camada de roteamento. Contudo, os algoritmos dessa categoria não serão apresentados aqui, uma vez que não se enquadram no escopo desse trabalho, pois também trabalham com informações da camada MAC. A seguir serão explicados alguns desses algoritmos, sendo feita a divisão conforme as categorias citadas acima.

2.2) Algoritmos que adormecem os nós de redes ad hoc

Todos os algoritmos desta seção controlam a transição dos nós da rede para um estado de dormência, onde o consumo energético é consideravelmente menor [Chen 02]. Nesse estado, o rádio dos nós é desligado. Isso leva a uma redução no consumo de energia de duas

formas: a primeira acontece pois, mesmo quando um nó em estado não dormente não está participando de transmissões, os dispositivos que compõem seu rádio devem permanecer energizados para que o nó tenha a capacidade de receber mensagens. A segunda forma de economia de energia acontece pois, como o rádio é desligado, o nó deixa de receber as transmissões que ocorrem entre seus vizinhos, ou seja, diminui a escuta ociosa.

2.2.1) GAF

Geographical Adaptive Fidelity, ou GAF [Xu 01], é um algoritmo que identifica nós que são redundantes em relação à sua função de roteamento de pacotes. Tais nós são colocados em estado de dormência para economizar energia. Esse algoritmo busca sempre conservar energia sem prejudicar o roteamento. Para realizar essas funções, o GAF requer dados de localização (que podem ser fornecidos por um dispositivo como um GPS localizado em cada um dos nós da rede).

O GAF, de acordo com os estudos realizados em [Xu 01], consegue uma economia de energia de 40% a 60%. Esse valor pode variar de acordo com a densidade da rede, de forma que redes mais densas têm maior economia de energia. Além disso, ele pode ser utilizado junto com vários tipos de protocolos de roteamento, como o AODV e o DSR, proporcionando bons resultados.

2.2.1.1) Divisão da área onde se encontram os nós

Inicialmente, o GAF divide a área onde se encontram os nós em uma grade virtual formada por sub-áreas quadradas de mesmo tamanho. Todos os nós que estão em uma determinada sub-área podem se comunicar com os nós que estão nas sub-áreas vizinhas. A figura 2 ilustra essa divisão. Nela são mostrados cinco nós (1, 2, 3, 4 e 5) em uma área que foi dividida em três sub-áreas (A, B e C) de lado *r*. Os nós 2, 3 e 4 podem trocar pacotes com os nós 1 e 5, mas esses últimos não podem trocar pacotes diretamente entre si. Se, por exemplo, o nó 1 enviar um pacote para o nó 5, esse pacote deve primeiramente ser transmitido para um dos nós da sub-área B, para depois ser transmitido para 5. Ou seja, como qualquer um dos nós 2, 3 ou 4 pode participar do roteamento desse pacote, os outros dois nós são considerados redundantes e podem adormecer para economizar energia, sem prejudicar o roteamento de pacotes.

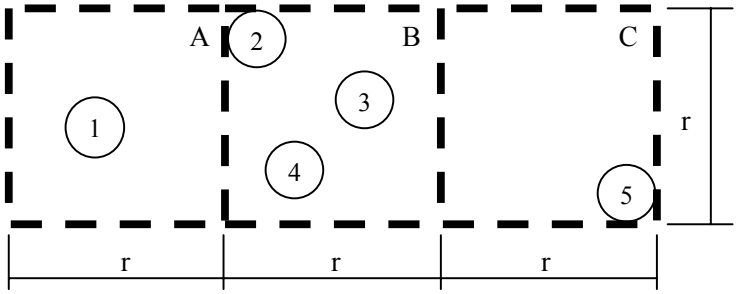

Figura 2: divisão de uma área em três sub-áreas quadradas de lado r

Torna-se necessária a definição do valor de *r*. Esse valor é definido em relação ao raio de alcance dos rádios dos nós, *R*. A maior distância entre dois nós de sub-áreas vizinhas ocorre quando eles se encontram nos vértices opostos de suas sub-áreas (como os nós 2 e 5 da figura 2). Pode-se concluir então que a distância entre esses dois vértices deve ser menor ou igual ao raio de alcance dos rádios para que eles possam se comunicar. A figura 3 mostra como o calculo de *r* é feito, a partir de um triângulo retângulo formado pelos lados consecutivos das sub-áreas, pelo lado de uma das sub-áreas e por uma linha traçada entre os dois vértices. Assim,

$$R^2 \geq r^2 + (2r)^2 \Rightarrow r \leq R / 5^{1/2}$$

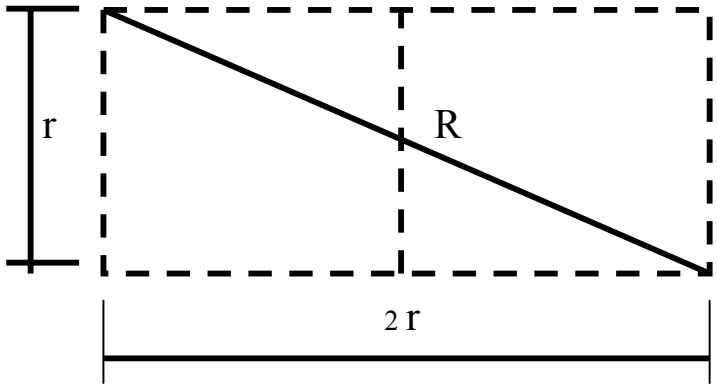

Figura 3: cálculo de r

2.2.1.2) Estados dos nós

GAF define três estados para os nós: *sleeping, active* e *discovery*. Inicialmente um nó fica em estado *discovery*, onde seu rádio fica ligado e uma mensagem de "descoberta" é transmitida com o intuito de encontrar nós que compartilham a mesma sub-área. Essa mensagem é formada pelo identificador do nó, pelo identificador da sub-área, pelo tempo que esse nó pode

permanecer em atividade (de acordo com suas reservas de energia), ou *enat*, e pelo o estado em que o nó se encontra no momento. Depois de T_d segundos esse nó decide se deve entrar no estado *active* ou adormece entrando no estado *sleeping* para economizar energia. Esta decisão é tomada a partir de um mecanismo de *ranking* que será explicado a seguir. No estado *active*, o rádio fica ligado e o nó continua transmitindo mensagens de descoberta a cada T_d segundos. Um nó permanece no estado *active* por T_a segundos, depois desse tempo ele volta para o estado *discovery*. Um nó no estado *discovery* pode decidir que deve ir para o estado *sleeping,* onde permanece por T_s segundos e depois volta para o estado *discovery.* A figura 4 mostra um diagrama com os três estados que um nó pode assumir:

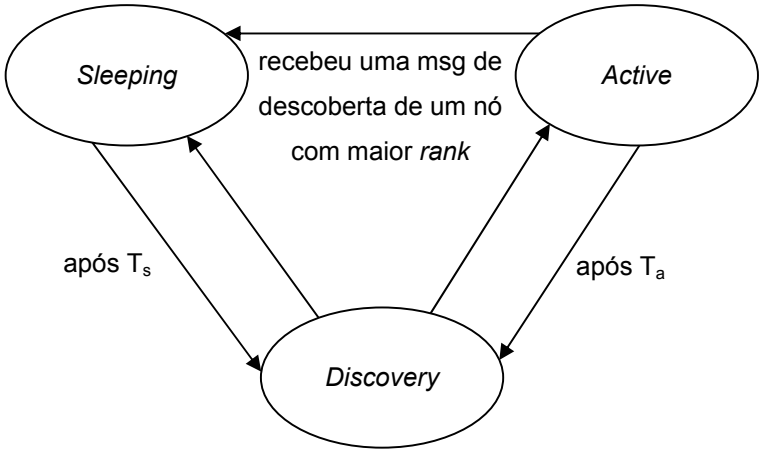

Figura 4: diagrama de estados

O mecanismo de *ranking*, mencionado anteriormente, faz a escolha de qual nó deve dormir e qual nó deve permanecer ativo dentro de uma sub-área. O nó com maior *rank* fica no estado *active,* os demais vão para o estado *sleeping.* Para determinar qual é o *rank* de um nó, esse mecanismo utiliza duas regras:

⇒ nós que já estão no estado *active* têm maior *rank* que nós no estado *discovery*;

⇒ entre nós com o mesmo estado, possui maior *rank* aquele com maior *enat*.

A primeira regra foi criada para que rapidamente somente um nó por área fique no estado *active*. Contudo, esse nó sempre é substituído após um determinado tempo. Esta substituição ocorre quando esse nó sai do estado *active* e volta para o estado *discovery.* Nesse momento suas reservas de energia já foram bastante consumidas, o que não ocorre com os nós que estão no estado *sleeping*. Por essa razão, um outro nó com maior reserva de energia entra no estado *active* e o primeiro nó vai para o estado *sleeping*.

Esse mecanismo é executado toda vez que um nó recebe uma mensagem de descoberta. Essa transição entre estados, além de economizar energia, faz com que a energia da rede seja balanceada, uma vez que os nós ficam se alternando entre o estado *sleeping* e os outros estados.

2.2.2) CEC

Cluster-based Energy Conservation, ou CEC [Xu 03], é um algoritmo para economia de energia em redes *ad hoc* criado pelos mesmos autores do GAF [Xu 01]. O CEC se apóia em um mecanismo de agrupamento dos nós, ou seja, a rede é dividida em grupos de nós ou *clusters*. A partir desta divisão, os nós são classificados em tipos diferentes, cada um com uma função dentro de seu grupo. A economia de energia acontece quando alguns desses nós são colocados em estado de dormência.

2.2.2.1) Classificação dos nós

Os nós podem ser classificados em três tipos: *cluster-heads, gateways* e normais. Os *cluster-heads* são os nós principais em um grupo. Eles são responsáveis por receber todas as comunicações feitas pelos nós que compõem seu grupo e transmiti-las a outros *cluster-heads* de outros grupos, se necessário. Com isso, todos os nós ao transmitirem uma mensagem devem transmiti-la para o *cluster-head* pois ele é o encarregado de retransmitir a mensagem a outros nós da rede. *Gateways* são nós que conseguem se comunicar com dois ou mais *cluster-heads.* Por esta característica, são utilizados para fazer a comunicação entre grupos vizinhos. Os demais nós são denominados normais. A figura 5 ilustra os três tipos de nós e os grupos formados.

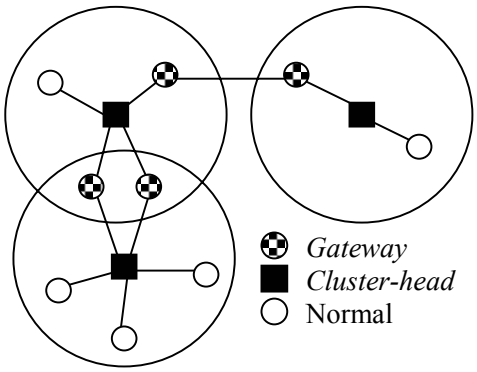

Figura 5: tipos de nós

2.2.2.2)Princípio de funcionamento.

Inicialmente, os nós fazem *broadcasts* de mensagens de descoberta. Essas mensagens são utilizadas pelos nós para a classificação e divisão dos nós em grupos. Os primeiros nós que são definidos são os *cluster-heads*. Depois são escolhidos os *gateways* e, finalmente, os demais nós são classificados como normais. Os nós normais entram, então, no estado *sleeping*. Eles permanecem nesse estado durante T_s segundos e depois acordam. Esse tempo é calculado a partir da razão entre o tempo estimado de vida do *cluster-head* (*enlt*) e uma constante definida pelo usuário (x). Existe uma outra situação onde um nó pode ir para o estado *sleeping*. Ela ocorre quando dois ou mais nós podem ser *gateways*. Com isso, só um permanece ativo, ou outros são adormecidos.

Quando um nó acorda, ele, provavelmente, possui melhores reservas de energia que seu *cluster-head.* Ocorre então uma troca de função, onde o *cluster-head* passa a ser um nó normal e o nó que era normal assume a posição do *cluster-head*. Essa característica foi criada para balancear a energia da rede, uma vez que os *cluster-heads* são mais utilizados que os nós normais e por isso consomem mais energia, morrendo precocemente.

Duas situações que devem ser tratadas podem ocorrer. A primeira é quando um nó está dormindo e deseja transmitir. O protocolo faz com que esse nó acorde, transmita e depois durma outra vez. A segunda situação corre quando chega um pacote para um nó que está dormindo. A medida adotada para resolver esse problema é criar um *buffer* de pacotes no *cluster-head* de forma ele armazene todos os pacotes destinados aos nós de seu grupo.

2.2.2.3) Características importantes

De acordo com [Xu 03], foi criado para esse algoritmo um mecanismo para torná-lo mais adequado a redes com maior mobilidade. Contudo, esse mecanismo não é sempre utilizado. Ele funciona calculando uma estimativa do tempo T que o *cluster-head* deve permanecer na rede. Esse valor é obtido pela razão entre o raio de transmissão do rádio e a velocidade de cada nó. Com isso define-se que os nós que não desempenham o papel de *cluster-head* podem dormir por, no máximo, T segundos.

2.2.3) On-demand power management for ad hoc networks

Em [Zhen 03] é apresentado um *framework* para economia de energia em redes *ad hoc*. Nele, a escolha dos nós que irão adormecer é feita através da observação das mensagens de controle que estão trafegando pela rede. Tais mensagens podem ser da camada MAC ou da camada de roteamento, o que torna possível a implementação desse *framework* sobre qualquer tipo de protocolo dessas camadas.

2.2.3.1) Transição entre os estados dos nós

Na utilização desse *framework,* os nós ficam variando entre os estados *power saving* (PS) e *active* (AM). Quando um nó está no estado AM ele permanece com seu rádio ligado podendo receber e transmitir mensagens a outros nós a qualquer momento. No estado PS o nó tem seu rádio desligado, ou seja, o nó entra em estado de dormência. Periodicamente, um nó no estado PS liga seu rádio para verificar a existência de algum pacote destinado a ele ou para retransmissão. Com isso, os pacotes endereçados a nós que estão dormindo sofrerão pequenos atrasos em seus envios.

A transição do estado AM para o estado PS, ou seja, a ação de adormecer um determinado nó, é determinada por um temporizador denominado *keep-alive.* Caso um nó não receba nenhum pacote destinado a ele, esse nó permanece acordado pelo período definido por esse temporizador. Ao final desse tempo o nó vai para o estado PS. Quando no estado AM, se um nó recebe um pacote destinado a ele, esse temporizador é reiniciado.

A transição inversa, ou seja do estado PS para o estado AM, ocorre quando um determinado nó que está dormindo verifica que existe uma mensagem destinada a ele. Isso ocorre pois um nó no estado PS periodicamente escuta o meio verificando se existe alguma mensagem para ele. Esta mensagem pode ser uma requisição de rotas (típica de protocolos de roteamento reativos), uma RTS (*request to send*, típica de protocolos da camada MAC), dados ou outro tipo de mensagem predefinido. Ao perceber a existência de tal mensagem esse nó muda seu estado para o estado AM, recebe a mensagem e reinicia seu temporizador.

2.2.3.2) Framework sobre demanda

Esse *framework* pode ser implementado de três formas distintas, de acordo com o valor associado ao temporizador *keep-alive.* Quando é associado um valor igual ao tempo de vida

da rede, os nós permanecem constantemente acordados e com isso a economia de energia não é realizada. Esta implementação é denominada de *always-on*. Quando o valor associado é igual a zero, os nós permanecem dormindo pelo maior tempo possível e só mudam para o estado AM quando existem mensagens destinadas a eles ou para retransmissão. Dessa forma, os tempos gastos para entrega de pacotes tendem a aumentar, ou seja, aumenta o retardo fim-a-fim. Essa implementação é denominada *always-off*.

O artigo original apresenta uma implementação denominada sob demanda (*on-demand*) onde se busca um valor intermediário para o temporizador *keep-alive.* O intuito desta implementação é economizar energia, mas tentando diminuir o atraso na entrega das mensagens. A figura 6 ilustra os tipos de implementações possíveis.

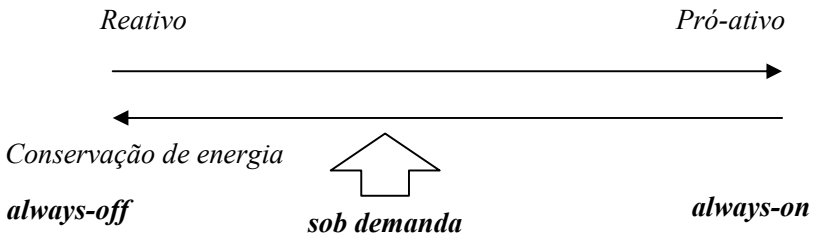

Figura 6: variação do temporizador *keep-alive*

2.2.4) BECA e *AFECA*

Esses algoritmos são apresentados em [Xu 00]. Eles se baseiam em mecanismos que desligam o rádio de determinados nós de redes *ad hoc* com alta densidade, com o intuito de economizar energia e estender o tempo de vida da rede. Foram criados de forma a se adaptarem às variações de densidade causadas pela movimentação dos nós. Trabalham juntamente com protocolos reativos de roteamento já conhecidos na literatura, como o DSR [John 96] e o AODV [Perk 99], além de utilizarem informações da camada MAC para determinar por quanto tempo os nós permanecerão com seus rádios desligados.

2.2.4.1) BECA

BECA, *Basic Energy-Conserving Algorithm*, se baseia na idéia de que os nós não necessitam estar consumindo energia quando não estão enviando, recebendo ou encaminhando pacotes por uma rota. Por isso, devem ficar adormecidos pelo maior tempo possível. Contudo, esse algoritmo pode elevar consideravelmente o tempo de transmissão de pacotes entre a origem e

o destino. Os nós variam entre três estados: *sleeping*, *listening* e *active.* A figura 7 mostra um diagrama contendo tais estados, os tempos associados a eles e as possíveis transições.

Inicialmente, um nó fica no estado *sleeping*, e permanece nesse estado durante T_s segundos. Após esse tempo, ele muda para o estado *listening*, mantendo seu rádio ligado. Se durante esse estado a aplicação do nó desejar transmitir algum pacote ou esse nó receber um pacote para ser encaminhado, ele migra para o estado *active*, caso contrário, depois de T_l segundos, ele volta para o estado *sleeping*. No estado *active* o nó fica com seu rádio ligado e permanece assim durante T_a segundos. Se ele não precisar transmitir nenhum pacote, ele volta para o estado *sleeping*, iniciando o ciclo. Quando existe um pacote para um determinado nó que está no estado *sleeping,* esse pacote é retransmitido R vezes a cada T_o segundos. Para manter a interação entre o BECA e os protocolos de roteamento, T_l é ajustado para receber o mesmo valor que T_o, T_s para receber um valor K vezes T_o e R é ajustado de forma a assegurar que pelo menos uma requisição, dentre as R que serão enviadas, alcançará o destino.

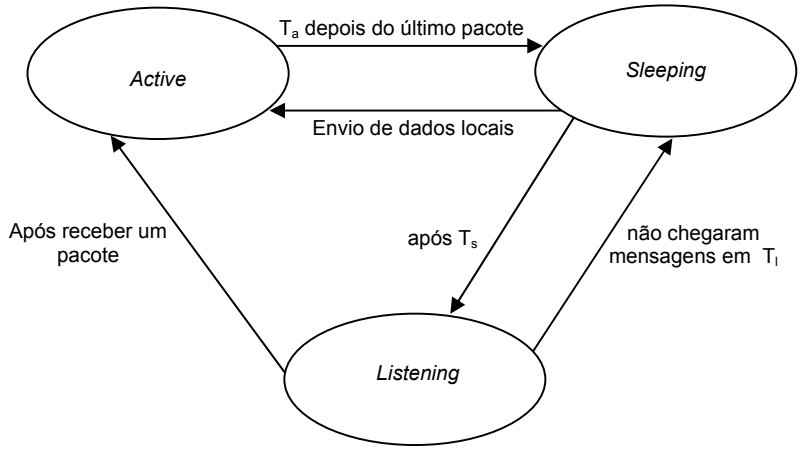

Figura 7: estados do BECA

2.2.4.2) AFECA

Esse algoritmo visa aumentar a economia de energia observando o aumento da densidade da rede, ou seja, em redes muito densas teremos uma maior economia de energia. Para isto, um nó aumenta o tempo que permanecerá no estado *sleeping* de forma proporcional à densidade de sua vizinhança, ou seja, é necessário que os nós consigam estimar o número de vizinhos. Para adquirir tal informação, os nós escutam as transmissões que estão ocorrendo em sua vizinhança e, a partir delas, montam uma lista de vizinhos. O AFECA possui um diagrama de transição de estados semelhante ao do BECA (Figura 7), alterando somente o tempo que ele permanece no estado *sleeping*.

2.2.5) Span

Apresentado em [Chen 02], *Span* é um algoritmo que busca economizar energia em redes *ad hoc* a partir da seguinte observação: "em regiões onde a densidade de nós é grande um pequeno número de nós consegue manter a conectividade da rede". Com isso, esse pequeno conjunto de nós deve permanecer ativo, enquanto os demais devem ficar em estado de dormência para economizar energia. Além de manter a conectividade, tais nós devem ser capazes de rotear todos os pacotes que passam por eles, sem que haja congestionamento na rede.

Para executar tal tarefa, esse algoritmo deve permitir que o maior número possível de nós desligue seus rádios pelo maior tempo possível. Além disso, depois de ter muitos de seus nós desligados, a rede deve manter a capacidade de rotear pacotes com um mínimo de retardo, quando comparada ao caso de todos os nós estarem acordados. Para tanto, o algoritmo *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 sub-grafo 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. Durante esse processo, cada nó decide sozinho se dever permanecer acordado ou se deve dormir, utilizando para isso informações de seu protocolo de roteamento.

2.3) Algoritmos que buscam enviar pacotes por nós com níveis de energia mais altos

O intuito desse tipo de algoritmo é distribuir o consumo de energia entre os nós da rede. Aqueles nós que possuem níveis de energia mais altos têm maior probabilidade de participar de rotas, consumindo mais energia. Os nós que possuem níveis mais baixos de energia são poupados. A partir disso, o tempo de vida da rede, ou seja, o tempo que leva até que o primeiro nó da rede morra por falta de energia, tende a aumentar. Um ponto negativo desses algoritmos é que o consumo de energia da rede como um todo pode aumentar, pois quando a rota

que possui nós com melhores níveis de energia é escolhida esta rota não é necessariamente a de menor caminho.

2.3.1) RDRP e MMRP

Em [Yu 02] são apresentados dois protocolos de roteamento em redes *ad hoc*, *Request-Delay Routing Protocol* (RDRP) e *Max-Min Routing Protocol* (MMRP), que visam fazer o balanceamento do consumo de energia entre os nós. Tais protocolos são baseados no DSR [John 96], ao qual foram adicionadas alterações em seu mecanismo de descoberta de rotas. O DSR foi escolhido para ser alterado pois, de acordo com os estudos realizados em [Cano 00], esse protocolo oferece bom desempenho em relação ao consumo de energia quando comparado com outros protocolos de roteamento encontrados na literatura.

2.3.1.1) RDRP

A idéia principal deste protocolo é: quando um nó receber um pacote de requisição de rotas ele deve atrasar a retransmissão deste pacote em um período de tempo inversamente proporcional ao nível de sua bateria. Este mecanismo foi construído a partir do fato de que um nó destino de uma rota somente aceita a primeira requisição recebida, descartando as demais. Com isso, nós que possuem níveis de energia mais altos retransmitem requisições de rotas mais rapidamente do que nós com níveis de energia mais baixos. Ou seja, as respostas de rotas que passam por esses nós com maiores níveis de energia, provavelmente, chegarão primeiro. As demais respostas de rotas serão, então, descartadas. Isto leva a uma melhor distribuição da energia entre os nós da rede, pois os nós que consumiram mais energia terão menor chance de participar de rotas.

2.3.1.2) MMRP

No DSR, quando um nó recebe um pacote de requisição de rotas, antes de retransmiti-lo, ele insere nesse pacote seu número de identificação. Isso ocorre quando o nó não possui em *cache* uma rota para o destino requisitado e quando esse nó não é o destino da requisição. No caso do nó ser o destino, ele responde somente à primeira requisição de uma determinada origem, descartando as outras. O DSR considera que a primeira rota que chega ao destino é a melhor, por isso ela é a escolhida.

No MMRP, quando um nó recebe um pacote de requisição de rotas, além de seu identificador, ele também insere nesse pacote o nível de energia de sua bateria. O nó destino que

receber uma requisição deve aguardar, por um tempo predefinido, a chegada de outras requisições vindas do mesmo nó origem. De posse de várias requisições vindas de uma mesma origem, o destino procura em cada uma delas o nó que possui o menor nível de energia. A partir daí, a cada requisição será associado um valor denominado $min(P_i)$, que é o menor nível de energia entre os nós pelos quais a requisição P_i passou. A requisição que será respondida é aquela que possuir o maior valor de $min(P_i)$. No caso de empate, a requisição com o segundo maior valor será respondida. O intuito deste protocolo é preservar os nós que estão com suas reservas energéticas bastante escassas, para isso utilizando outras rotas.

O MMRP pode ser implementado com um número limite de rotas que o nó destino de uma requisição receberá. Além do temporizador, esse número evita que um nó receba muitas rotas e com isso o tempo necessário para aquisição de uma rota sofra um acréscimo muito grande.

2.3.2) AODV-E

O protocolo AODV-E [Cunh 03] é uma variação do protocolo AODV. As modificações foram feitas no mecanismo de descoberta de rotas do protocolo original, para que ele levasse em consideração os níveis de energia dos nós que compõem tais rotas.

O AODV original possui um mecanismo que descarta rotas redundantes, ou seja, quando o nó que requisitou uma rota recebe mais de uma resposta, ele assume a primeira rota recebida como a melhor, descartando as demais. Partindo desse princípio, o AODV-E inseriu um tempo de espera antes que um nó retransmita um pacote de requisição de rotas. Esse tempo é inversamente proporcional à energia do nó naquele instante. Com isso, respostas de rotas compostas por nós com níveis de energia baixos demorarão mais para retornar à origem do que respostas de rotas compostas por nós com altos níveis de energia. Isto aumenta a probabilidade das rotas com baixos níveis de energia serem descartadas.

2.3.2.2) Mecanismo de descoberta de rotas alterado

O AODV original [Perk 98] é um protocolo reativo, ou seja, só descobre rotas quando elas são necessárias. Quando um determinado nó origem deseja enviar um pacote a outro nó destino, o nó origem envia por *broadcast* a todos os seus vizinhos um pacote de descoberta de rotas. Quando um nó recebe um desses pacotes ele, verifica se é o destino informado na requisição de rotas ou se possui a rota requisitada em sua *cache*. Se uma dessas duas

possibilidades for verdadeira, esse nó transmite para o nó origem, pelo caminho inverso ao caminho feito pela requisição, uma resposta de rota contendo a rota requisitada. Caso contrário, esse nó retransmite, também em *broadcast* para seus vizinhos, a requisição de rotas recebida.

O AODV-E alterou o protocolo AODV de forma que, antes que um nó retransmita uma requisição de rotas, é inserido um atraso inversamente proporcional ao nível de energia daquele nó. Esse mecanismo é o mesmo que foi criado para o protocolo RDRP [Yu 02] e explicado na seção anterior.

Contudo, no AODV-E, para o cálculo desse atraso foram analisadas várias situações. A fórmula escolhida foi a que obtém o atraso a partir da energia do nó no instante que ele recebe a requisição, da energia inicial do nó e do atraso máximo definido pelo usuário. Tal fórmula, quando os nós estão com nível máximo de energia, oferece atraso zero. As outras possibilidades analisadas ofereciam atraso mínimo igual à metade do atraso máximo, o que aumentava ainda mais o tempo gasto pelo protocolo para aquisição de rotas.

2.4) Comentários

De acordo com o que se pode observar nos vários trabalhos discutidos nesse capítulo, economizar energia no nível dos protocolos de roteamento é uma tarefa que pode ser realizada por algoritmos que coordenam a transição dos nós para um estado de dormência, onde o gasto energético é muito menor. Para tal coordenação várias informações podem ser utilizadas. A forma pela qual cada algoritmo utiliza tais informações, escolhe quais nós dormirão e determina por quanto tempo cada nó permanecerá neste estado é o que determina a sua eficiência na economia de energia.

Além de economizar energia, torna-se importante balancear o consumo de energia na rede, para evitar que alguns nós sejam muito utilizados e morram por falta de energia, em detrimento de pouca ou nenhuma utilização de outros. Tal tarefa é realizada por outros algoritmos apresentados neste capítulo. Contudo, as técnicas empregadas por tais algoritmos podem levar a um aumento no consumo energético da rede, uma vez que as rotas formadas por nós com maiores níveis de energia nem sempre são as de menor caminho.

No próximo capítulo será apresentado um algoritmo que coordena a transição dos nós de redes *ad hoc* para o estado de dormência para economizar energia. Para isso, somente são utilizadas informações da camada de roteamento. Tal economia é distribuída entre os nós da rede de forma que o tempo de vida da rede também aumente.

Capítulo 3 - O Algoritmo *Extra*

3.1) Introdução

Nessa seção será descrito o algoritmo *Extra* (Extensão do Tempo de vida de Redes Ad hoc) 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, exigindo para isso poucas alterações nesse protocolo. Além disso, ele só utiliza informações que podem ser obtidas localmente na camada de roteamento, não sendo necessária, então, qualquer comunicação com camadas superiores (como transporte e aplicação) ou com camadas inferiores (como as camadas de acesso ao meio e física).

O procedimento de economia de energia 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. Além destas três situações onde ocorre gasto energético, existe também um consumo associado à necessidade do rádio de cada nó de permanecer energizado para que seja capaz de receber pacotes. Esse consumo é denominado consumo ocioso, o qual também é minimizado quando se coloca o nó em estado de dormência.

A principal característica desse 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.

3.2) Operação do algoritmo

Durante sua operação, um nó poderá estar em um de três estados: *Ativo*, *Escutando* e *Dormindo*, conforme indicado na Figura 8. Um nó inicia operação enviando uma

mensagem de *hello*, informando a seus vizinhos que está no estado *Ativo*. Nesse 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 esse 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.

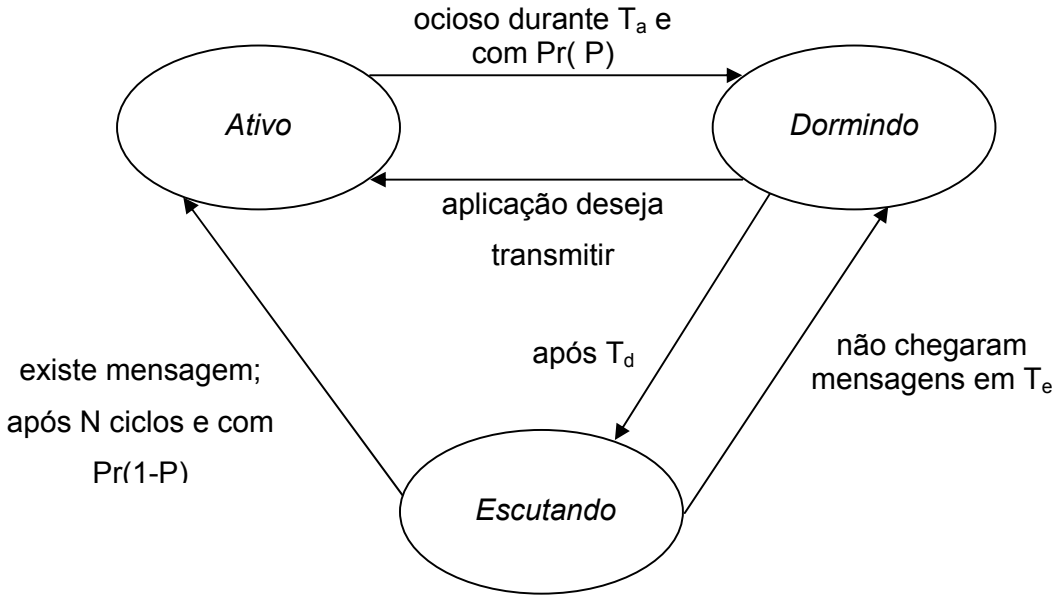

Figura 8: transições de estado de um nó com o *Extra*

Um nó permanece no estado *Dormindo* por T_d segundos. Ao final desse 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*. Esse procedimento é repetido N vezes (ver exemplo na Figura 9). 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 mudar para o estado *Ativo* e reiniciar a contagem de T_a 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 esse nó deve também reiniciar o mecanismo.

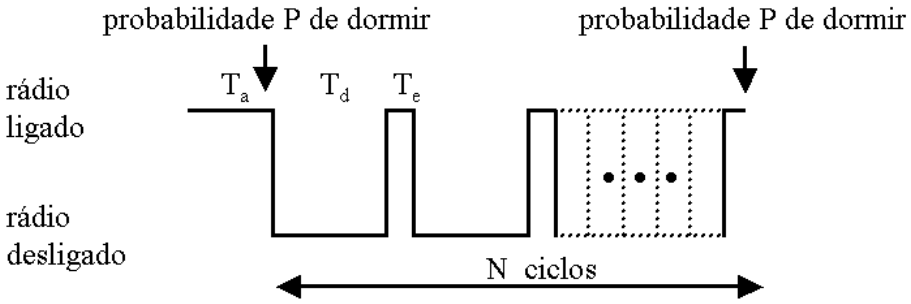

Figura 9: exemplo de evolução temporal

Quando um nó ativo desejar transmitir um pacote destinado a um vizinho, a transmissão deve ser efetuada somente quando esse se encontrar 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 esse vizinho também é atualizada.

3.3) 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. Tais adaptações podem ser comprovadas pelos experimentos efetuados na seção 4.3.

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 para as duas constantes acima referenciadas 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. A partir daí temos:

$$P(h_2) = \text{numero_vizinhos_ativos( )} / \text{numero_vizinhos( )}$$

onde numero_vizinhos_ativos( ) é uma função que retorna o número de vizinhos que estão em estado ativo e numero_vizinhos( ) retorna o número total de vizinhos do nó. Pode-se notar que esta fórmula sempre terá como resultado um valor entre 0 e 1, uma vez que o número de vizinhos ativos do nó sempre será um número menor que o número total de vizinhos.

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:

$$P(h_3) = 1 - \textit{energia_atual( )} / \textit{energia_inicial( )}$$

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. Também se pode notar que esta fórmula sempre terá como resultado um valor entre 0 e 1, uma vez que a energia atual do nó sempre será um número menor que energia inicial.

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:

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

onde $P(h_1)$, $P(h_3)$ e $P(h_4)$ são os valores da probabilidade P que são 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 $(P(h_3) < ref) \Rightarrow P(h_5) = \min(P(h_2), P(h_3))$

3.4) Analise das heurísticas

As heurísticas 1 e 2 buscam ajustar o valor de P utilizando somente informações referentes à conectividade da rede, ou seja, só analisam o número de vizinho e o número de vizinhos em estado *Ativo*. A heurística 3, que pode ser interpretada como de comportamento egoísta, pois somente chega a um valor de P utilizando o estado da bateria de cada nó, sem analisar sua vizinhança para executar tal tarefa.

Para encontrar valores de P que utilizassem tanto informações da vizinhança quanto informações da bateria do nó foram criadas as heurísticas 4 e 5. Elas são as combinações dos resultados entre as heurísticas 1 e 3 e as heurísticas 2 e 3, respectivamente. Ambas foram criadas para que seja utilizado o maior valor de P quando as condições forem propícias para o nó dormir (conectividade alta e energia baixa) e o menor valor de P quando as condições de não forem propícias para que o nó durma (baixa conectividade e muita energia). Quando a conectividade estiver alta (alta probabilidade do nó dormir) e a energia também estiver alta (baixa probabilidade nó dormir) é escolhido o valor de P referente à conectividade, pois se sua vizinhança está cheia um nó pode dormir, mesmo que ainda tenha muita energia. Já para o inverso, conectividade baixa (baixa probabilidade nó dormir) e energia também baixa (alta probabilidade nó dormir), é escolhido o valor de P referente à energia, buscando sempre economizar energia quando a bateria estiver descarregada. Foi considerada probabilidade alta ou baixa, aquela que for maior ou menor, respectivamente, que 50% da energia total. Esse valor é ajustável através do parâmetro *ref*.

Capítulo 4 - Experimentos

4.1) Introdução

Os experimentos foram realizados em três etapas. Na primeira etapa foram realizados experimentos com o intuito de encontrar os valores de parâmetros (T_a, T_d, T_e e N) a serem passados ao algoritmo *Extra* de forma que ele alcance os melhores resultados. Esses parâmetros estão relacionados com o ciclo de estados pelo qual os nós que estão executando o algoritmo necessariamente têm que seguir. Além disso, foram realizados nesta fase experimentos que buscavam, dentre as cinco heurísticas apresentadas, a que apresentasse melhores resultados.

Na segunda etapa dos experimentos buscou-se validar o algoritmo, utilizando para isso um grande número de variações de cenários. Nesta parte foram utilizados, para servir de base de comparação, o protocolo AODV [Perk 99] puro e o algoritmo GAF [Xu 01] rodando sobre o AODV, sendo que o *Extra* também foi implementado sobre esse mesmo protocolo. A terceira parte é composta pelos experimentos que foram criados com o intuito de analisar o risco da rede ficar desconexa. Esse estudo se tornou relevante pois o *Extra* utiliza um mecanismo de probabilidade para adormecer os nós da rede, o que poderia levar a rede a uma completa desconexão. Nesta parte serão comentados os riscos desta desconexão.

Todos os experimentos foram executados através de simulação. O simulador utilizado nos testes foi o *Network Simulator - ns-2*, versão 2.26 [NS 03]. 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 [Xu 01]. No nível de enlace é assumida a interface 802.11 (DCF, 2Mbps). 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%.

Nas duas primeiras etapas de experimentos foram consideradas quatro métricas para análise de desempenho. A primeira métrica é a percentagem de nós sobreviventes ao final da simulação. Com essa métrica torna-se possível avaliar o impacto do uso dos algoritmos sobre a economia de energia. A segunda métrica utilizada foi o Percentil 90 do retardo sobre o tempo total de transmissão dos pacotes entre as origens e os destinos. Pode-se com isso observar se a utilização dos algoritmos aumentou o retardo na entrega de pacote. A terceira métrica é taxa de descarte de pacotes. Esse valor é obtido pela razão entre número de pacotes que saíram das fontes mas que não alcançaram o destino e o número total de pacotes que saíram das fontes. Com a análise dos valores coletados com essa métrica, torna-se possível verificar se a utilização dos

algoritmos afeta o número de pacotes perdidos pelos nós da rede durante as transmissões. A última métrica é o número de pacotes que alcançaram o destino com uma unidade de energia (Joule). Esse valor é obtido somando o número de pacotes úteis (pacotes que conseguiram alcançar o destino) e dividindo esse valor pelo total da energia gasta na rede durante toda a simulação. A análise dos valores coletados para esta métrica torna possível verificar como os algoritmos utilizaram a energia da rede, pois quanto maior for o número de pacote que alcançaram o destino com uma unidade de energia, melhor estão sendo utilizados os recursos energéticos da rede.

4.2) Ambiente de simulação

O cenário de simulação consistiu de 60 nós, movendo-se de forma aleatória (modelo *random way-point*) em um plano de 1200x600m (figura 10). 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. Isso ocorre pois em alguns cenários os nós fonte e dreno poderiam ser posicionados ou se deslocarem um ao encontro do outro, de forma que a distância permita a comunicação direta entre eles. Em outros casos, esses nós podem permanecer, durante todo o tempo de execução, separados por maiores distâncias, o que obriga a comunicação através de nós intermediários. Esses dois tipos de cenários citados apresentam resultados muito diferentes. 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.

Para criar o cenário de simulação foi utilizado o programa *BonnMotion*, versão 1.1 [Bonn 03]. Uma importante razão na escolha desse programa 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.

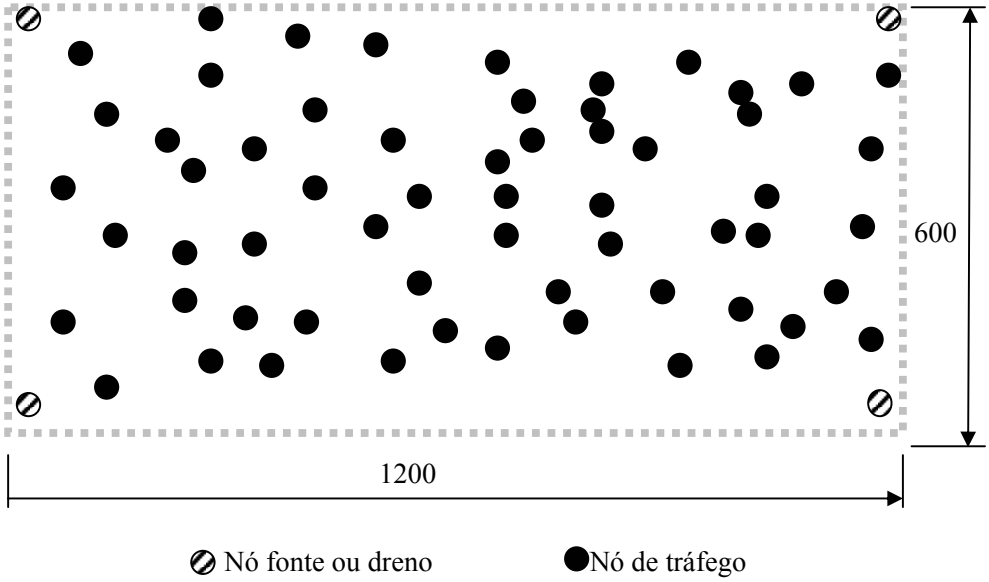

Figura 10: cenário criado para os experimentos

O tempo de simulação foi de 900 segundos, o que serviu para comparação com medidas semelhantes realizadas para o algoritmo GAF em [Xu 01]. 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.

O modelo de energia escolhido é baseado nas medidas de [Stem 97] da placa WaveLAN 2 Mbps. Os valores obtidos no mencionado trabalho foram 1,6W para transmissão, 1,2W para recepção e 1,0W quando ocioso. Esses valores são os mesmos empregados na avaliação do GAF em [Xu 01], de forma a facilitar comparações. Como o consumo de energia no estado dormindo é muito menor que nos demais (0,025 W), 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 entra no estado *Dormindo* seu 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 quando o nó está ocioso é significativo, pôde-se observar que os nós em uma rede com roteamento AODV puro morrem aproximadamente aos 450s, independentemente do tráfego encaminhado.

4.3) Variação dos parâmetros

Como mencionado anteriormente, o *Extra* possui quatro parâmetros relacionados a seu ciclo de operação. Nesses primeiros experimentos, o intuito foi encontrar valores para esses parâmetros de forma a alcançar os melhores resultados possíveis. Tais parâmetros são:

- ⇒ **T_a ou tempo ativo**: é o parâmetro que informa qual o tempo que o nó deve permanecer ativo, sem receber ou transmitir pacotes, antes de executar o mecanismo que escolherá, com probabilidade P, se esse nó continuará ativo ou se iniciará o mecanismo de economia de energia;
- ⇒ **T_d ou tempo dormindo**: esse parâmetro estabelece o tempo que os nós permanecerão dormindo;
- ⇒ **T_e ou tempo escutando**: tempo que os nós permanecerão escutando para saber se existe algum pacote para eles ou para retransmissão;
- ⇒ **N**: parâmetro que estabelece o número de vezes que o nó fará a transição entre os estados *Dormindo* e *Escutando* até executar o mecanismo que determina se ele ficará ativo ou não.

Os parâmetros básicos escolhidos foram N=15 ciclos, T_a=4s, T_d=0,5s e T_e=0,05s. Para os testes de um parâmetro, enquanto ele variava, os demais permaneceram com os valores citados acima. A heurística utilizada em todos os experimentos foi a 4. Também foi assumida velocidade constante para todos os nós de 1 m/s (equivalente a uma pessoa caminhando, típico em redes *ad hoc*), o tempo de pausa foi de 0s (nós sempre se movimentando, pior caso) e a taxa de transmissão de pacotes foi variada em 1, 10 e 20 pcts/s.

Esses valores para os parâmetros e a heurística foram escolhidos durante a fase de implementação do algoritmo, onde vários experimentos foram realizados utilizando tal configuração e obtendo-se bons resultados.

4.3.1) Variação do parâmetro N

Esse parâmetro foi variado entre os valores 1, 5, 10, 15 e 25. O primeiro gráfico apresentado (figura 11) mostra o número de nós sobreviventes após 900s de simulação para os cinco valores de N. Pode-se verificar que quanto maior o valor de N maior é o número de nós sobreviventes ao final da simulação. Isto acontece pois quanto maior for o ciclo entre os estados *Dormindo* e *Escutando*, maior será o número de vezes que um determinado nó ficará no estado *Dormindo*, conseqüentemente levando a uma maior economia de energia. Além disto, pode-se notar uma diminuição do número de nós vivos com o aumento da taxa de transmissão, o que se torna mais acentuado para valores menores de N, sendo que para N = 1 e taxas de 10 e 20 pacotes/s todos os nós da rede haviam morrido por falta de energia.

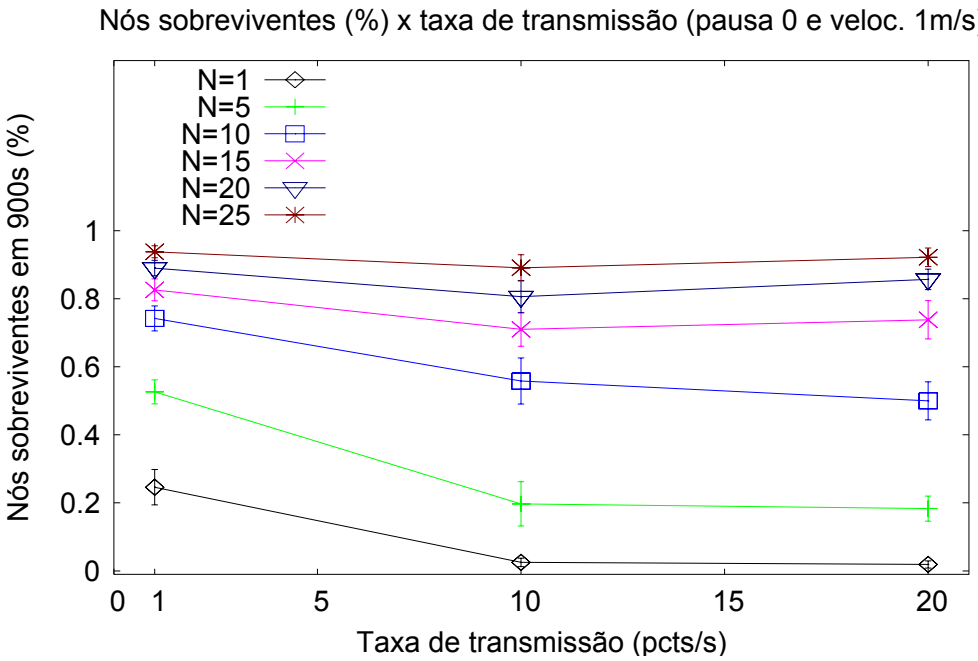

Figura 11: nós sobreviventes em 900s de simulação por taxa de transmissão

O próximo gráfico (figura 12) apresenta o retardo na entrega de pacotes para os cinco valores de N em questão. Nele pode-se notar que ocorreu uma diminuição do retardo para taxas de transmissão de 10 pacotes/s, com exceção da curva que representa N = 1. Esse comportamento ocorre pois com taxas menores de transmissão de pacotes os nós ficam mais tempo ociosos, aumentando a probabilidade de dormir, o que proporciona maior atraso na transmissão. Além disso, com taxas de transmissão maiores o congestionamento é maior, o que também aumenta o retardo. A curva que representa o parâmetro N = 1 apresentou um

comportamento um pouco diferenciado, sempre aumentando o retardo com o aumento da taxa de transmissão. Tal fato pode ser explicado pois quando as taxas de transmissão assumem menores valores, o retardo para entrega de pacotes tende a ser menor, uma vez que os nós dormem menos. Contudo, um ciclo menor aumenta o número de mensagens de *hello* transmitidas pelos nós. Para taxas de transmissão menores, tais mensagens não influenciam muito no congestionamento da rede. Porém, para taxas de transmissão maiores, onde o congestionamento já é grande, qualquer aumento no tráfego de pacotes também aumenta o congestionamento, atrasando ainda mais a entrega de pacotes.

Pode-se verificar no gráfico que as curvas possuem grandes faixas de valores para seus intervalos de confiança, tanto para taxas de transmissão de pacotes mais altas quanto para taxas mais baixas. Isso mostra que ocorreu uma variação muito acentuada no atraso médio entre valores coletados com os vários cenários utilizados nas simulações, o que explica as inversões ocorridas entre as curvas referentes a N = 5, 10 e 20.

Nota-se que as curvas para N = 20 e N = 25 apresentam tendência a aumentar o descarte para taxas de transmissão mais baixas. Contudo, para a taxa em torno de 10 pcts/s essa tendência é menos acentuada. Outro ponto importante a se verificar é que a curva que representa N = 15 mostrou o mesmo comportamento das demais, aumentando o retardo tanto para taxas menores quanto para maiores. Contudo, esse comportamento ocorreu de forma menos acentuada. Além disso, para todas as taxas de transmissão, esse valor de N não se apresentou com os piores resultados.

O gráfico da figura 13 apresenta a taxa de descarte de pacotes para os valores de N analisados. Todas as curvas apresentam comportamento semelhante, aumentando a taxa de descarte de pacotes com o aumento da taxa de transmissão de pacotes. Nota-se que para uma determinada curva a faixa de valores do intervalo de confiança de um de seus pontos no gráfico abrange praticamente todos as outras curvas. Com isso conclui-se que o valor de N não influi na taxa de descarte de pacotes.

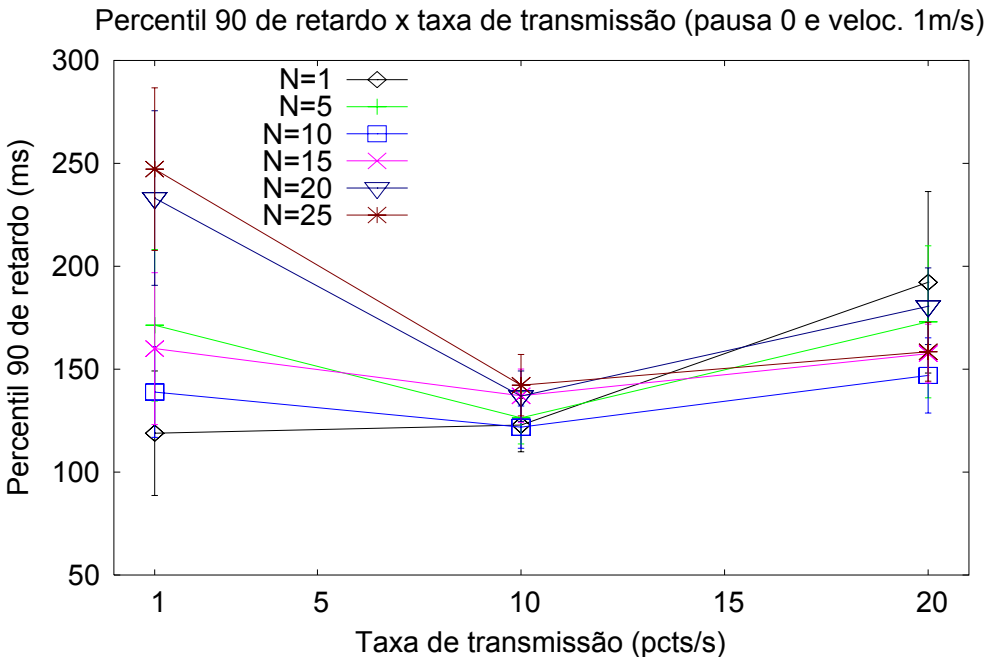

Figura 12: percentil 90 de retardo por taxa de transmissão, para cinco valores de N

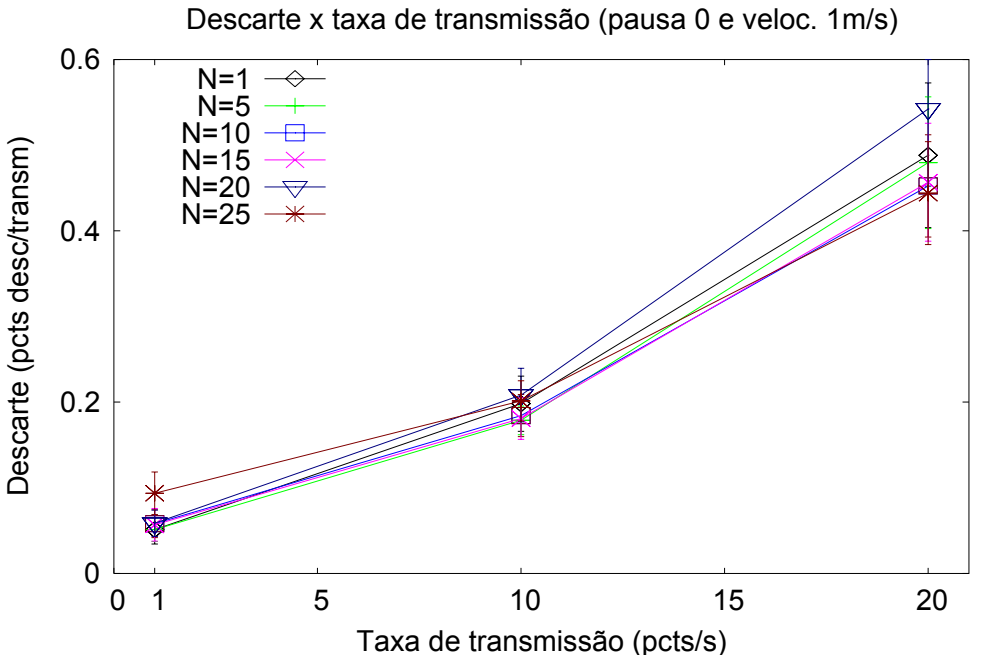

Figura 13: taxa de descarte de pacotes por taxa de transmissão, para cinco valores de N

O gráfico da figura 14 mostra o número de pacotes úteis transmitidos por unidade de energia, ou seja, quantos pacotes alcançaram seu destino para cada Joule gasto na rede. A primeira coisa que se pode notar é que com o aumento da taxa de transmissão de pacotes também aumenta a relação de pacotes transmitidos por Joule. Isto ocorre pois, com taxas de transmissão menores o maior gasto energético passa a ser feito pelo consumo ocioso de cada nó. Pode-se verificar que o aumento do valor de N melhora a utilização da energia da rede. Dos valores analisados, o que levou a melhores resultados foi N = 25. Conclui-se que, com N assumindo valores mais altos os nós dormem mais e conseqüentemente economizam a energia. Tal economia faz com que a rede passe a gastar mais sua energia com a transmissão de pacotes e menos com o consumo ocioso.

Pode-se notar no gráfico que ocorreu uma inversão entre as curvas que representam os valores de N = 15 e N =20, para taxa de transmissão de 20 pcts/s. Tal fato é explicado pois a faixa de valores do intervalo de confiança dessas curvas é grande, o que faz com que uma curva se encontre dentro do intervalo de confiança outra. Isso ocorreu pois os experimentos apresentaram uma grande variabilidade de resultados para a taxa de transmissão em questão.

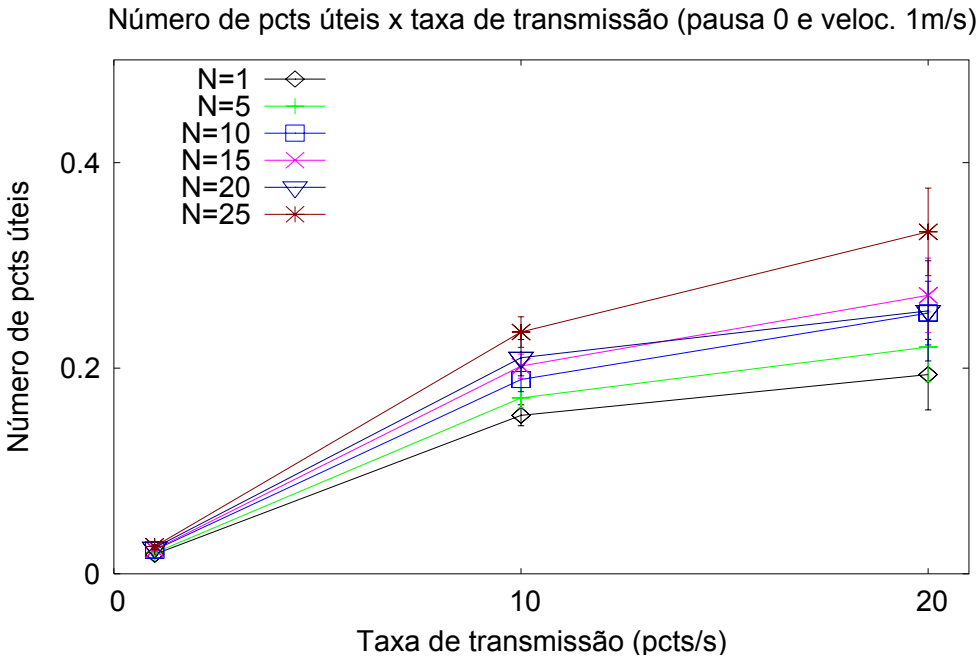

Figura 14: pacotes que alcançaram o destino por unidade de energia x taxa de transmissão, para cinco valores de N

A partir da análise desses gráficos, pode-se concluir que quanto maior o valor de N maior é a economia de energia. Contudo o retardo também tende a aumentar, principalmente para taxas de transmissão de pacotes menores. Verifica-se que N = 25 e N = 20 proporcionam bons resultados em relação ao número de nós sobreviventes após 900s de simulação e ao número de pacotes transmitidos por unidade de energia. Contudo, N= 15 será escolhido pois apresenta bons resultados em relação a economia de energia da rede e não mostra tendência de aumento de retardo.

4.3.2) Variação do parâmetro T_a

Esse parâmetro estabelece o tempo que os nós devem permanecer ativos, sem receber ou transmitir mensagens, antes que eles executem o mecanismo que decide se vão permanecer ativos ou iniciar o procedimento de economia de energia. Foram escolhidos os seguintes valores para o parâmetro T_a: 2, 4, 8 e 16 segundos.

A figura 15 apresenta o gráfico com a porcentagem de nós sobreviventes após 900s de simulação. Nos experimentos onde T_a = 8s, para taxa de transmissão igual a 1 pct/s, verificam-se alguns nós ainda com energia ao final da simulação, mas para as demais taxas de transmissão todos os nós morreram por falta de energia. Com T_a = 16s todos os nós morreram por falta de energia antes do final da simulação, para todas as taxas de transmissão de pacotes estudadas. Quando T_a assumiu valores iguais a 4 segundos, cerca de 80% dos nós ainda tinham energia ao final da simulação, para todas as taxas de transmissão de pacotes. Para T_a igual a 2 segundos, praticamente todos os nós da rede ainda tinham energia ao final da simulação. Esse comportamento é explicado pelo fato de que quanto maior o valor de T_a maior será o tempo que o nó permanecerá ativo, conseqüentemente gastando energia. Para valores mais baixos a economia é maior pois o nó fica menos tempo ativo.

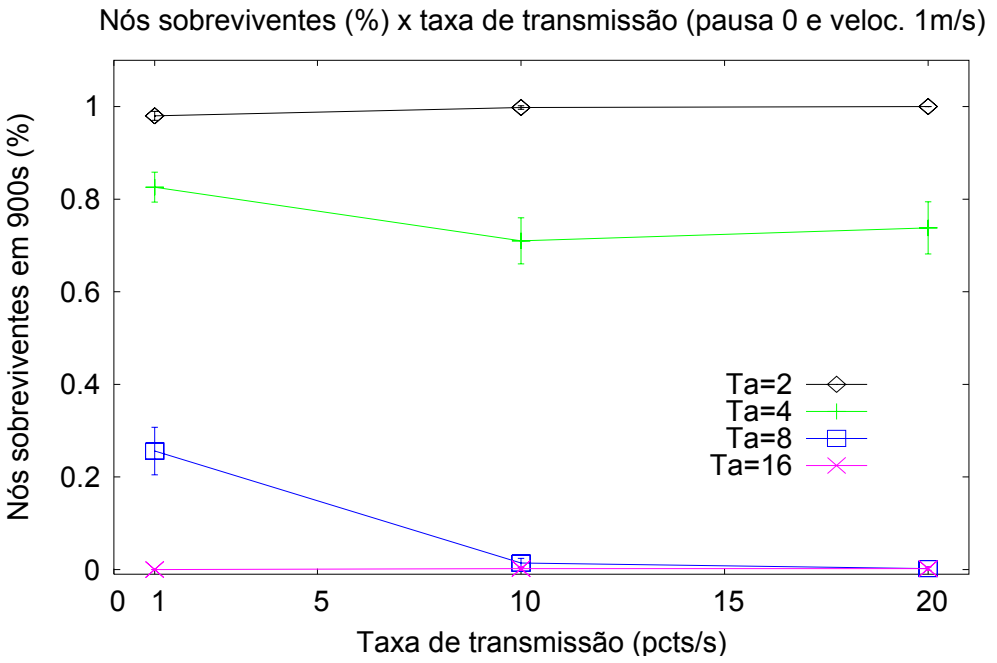

Figura 15: porcentagem de nós sobreviventes por taxa de transmissão, para os valores de T_a

O próximo gráfico, figura 16, apresenta o resultado do retardo na entrega de pacotes para os valores de T_a estudados. Destaca-se nesse gráfico o comportamento da curva referente a T_a = 2s, para taxas mais baixas de transmissão de pacotes. Nota-se um aumento no retardo, em comparação aos demais valores de T_a. Isso ocorre pois com T_a assumindo tal valor e com taxas de transmissão pequenas os nós que compõem uma determinada rota tendem a dormir mais, aumentando assim o retardo na entrega de pacotes. O aumento da taxa de transmissão faz com que os nós formadores de uma rota estejam sempre recebendo pacotes e conseqüentemente reiniciando a contagem de T_a.

Para os outros valores de T_a pode-se notar uma queda no retardo para taxas de transmissão de pacotes em torno de 10 pcts/s. Isso ocorre pois com taxas de transmissão mais baixas os nós dormem mais e com taxas mais altas ocorre congestionamento na rede, tudo isso proporciona aumento do retardo. Contudo, essa variação foi menos acentuada quando T_a assumiu valores iguais a 4 segundos.

Para taxas de transmissão de 1 e 20 pcts/s, pode-se notar inversões entre as curvas do gráfico. Contudo, todas as curvas se encontram dentro das faixas de intervalo de confiança umas das outras, o que ocorre devido à variabilidade nos resultados obtidos com tais experimentos.

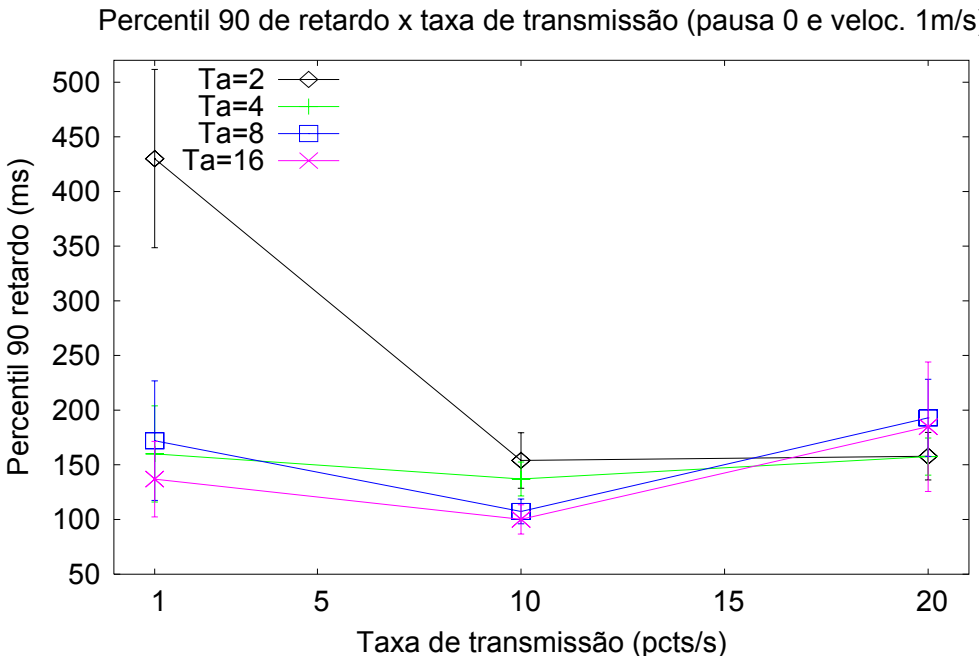

Figura 16: percentil 90 de retardo por taxa de transmissão, para os valores de T_a

O próximo gráfico, figura 17, apresenta a taxa de descarte de pacotes para os quatro valores de T_a estudados. Com exceção da curva para T_a = 2 s, os demais valores de T_a apresentam taxas de descarte equivalentes, estando sempre uma curva dentro da faixa de valores do intervalo de confiança da outra. Pode-se concluir que ocorre aumento no descarte para valores de T_a mais baixos, pois os nós estão menos tempo ativos e mais tempo dormindo, o que aumenta a perda de pacotes.

A figura 18 apresenta o número de pacotes úteis, ou seja, número de pacotes de alcançaram o destino por unidade de energia. Como em todos os experimentos para análise dessa métrica, os melhores resultados são obtidos com maiores taxas de transmissão de pacotes, pois os nós gastam mais energia com transmissões e recepções de pacotes e menos com o consumo ocioso. O melhor desempenho apresentado foi para T_a = 2s segundos, seguido por T_a igual a 4s, 8s e 16 s. Pode-se, então notar nesse experimento que o desempenho é inversamente proporcional ao valor de T_a. Isso ocorre pois com os nós menos tempo ativos, maior é a economia de energia, ou seja, o gasto com o consumo ocioso diminui.

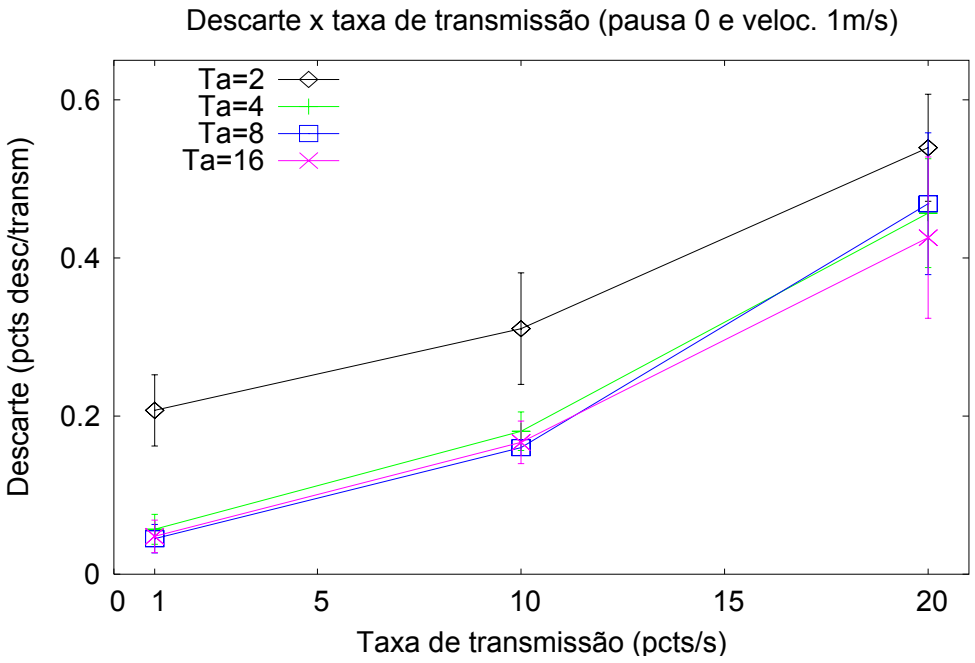

Figura 17: taxa de descarte de pacote por taxa de transmissão, para os valores de T_a

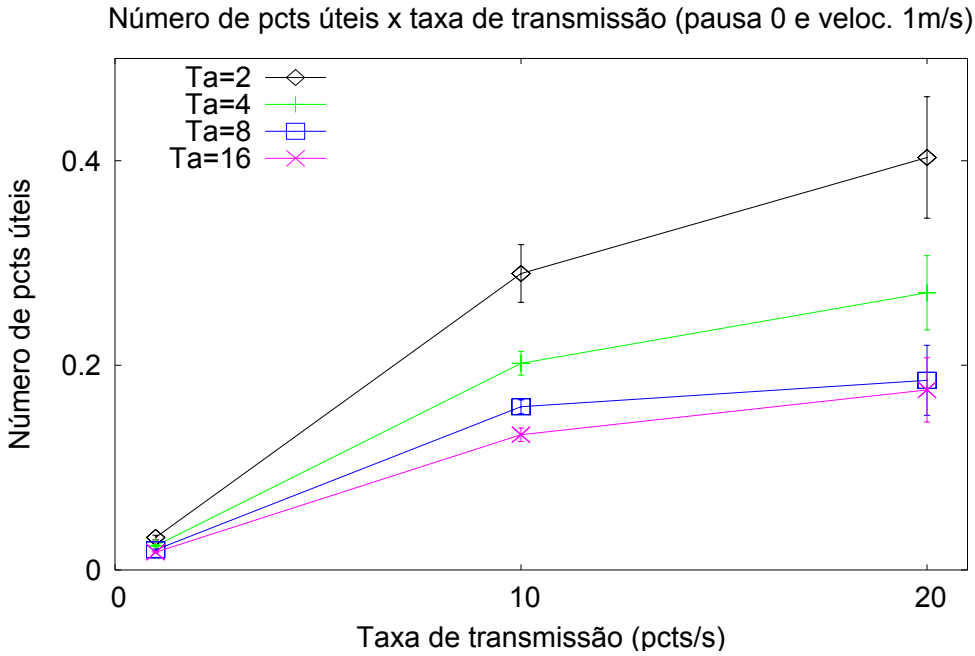

Figura 18: pacotes que alcançaram o destino por unidade de energia x taxa de transmissão, para os cinco valores analisados de T_a

A partir da análise dos gráficos pode-se verificar que T_a = 2s apresenta os melhores resultados em relação ao número de nós vivos em 900s de tempo de simulação e ao número de pacotes que alcançaram o destino por unidade de energia. Contudo, os gráficos também mostram que para T_a = 2s a taxa de descarte de pacote é a maior para todas as taxas de transmissão estudadas e o que o retardo também é o maior para taxas de transmissão de 1 e 10 pcts/s. Com T_a = 4s, o número de nós sobreviventes em 900s de simulação e o número de pacotes que alcançaram o destino por unidade de energia apresentaram o segundo melhor resultado. Em relação ao descarte de pacotes esse valor de T_a proporcionou resultados equivalentes aos demais e em relação ao retardo, além de apresentar resultados equivalentes, ele se mostrou pouco sensível à variação da taxa de transmissão. Por esta razão, T_a = 4s foi escolhido como o melhor valor para esse parâmetro.

4.3.3) Variação do parâmetro T_d

Esse parâmetro estabelece o tempo que um nó permanece no estado *Dormindo*. Os valores atribuídos a T_d foram: 0,05s, 0,1s, 0,5s, 1s e 2s. O primeiro gráfico com a variação de T_d, figura 19, mostra a porcentagem de nós sobreviventes em 900s de simulação, para os quatro valores mencionados. A primeira coisa que se pode verificar é que as curvas que representam T_d = 0,05s e T_d =0,1s ficam sempre em zero. Ou seja, para esses valores a economia de energia não foi suficiente para que após 900s de simulação algum nó ainda permanecesse vivo. Com o incremento do valor de T_d para 0,5s temos um aumento considerável na porcentagem de nós que ainda têm energia ao final da simulação. Tal porcentagem variou entre 75 e 80% dos nós, para a variação da taxa de transmissão. Com valores de T_d iguais a 1s e 2s temos que praticamente toda a rede permanece viva, ou seja, para esses valores os nós dormiram mais e a economia de energia é maior.

Na figura 20 tem-se o gráfico mostrando o percentil 90 de retardo para os valores atribuídos a T_d. A curva que representa T_d = 2s apresenta maiores retardos que as demais mostradas nesse gráfico. Isso ocorre pois com o aumento do tempo que o nó permanece dormindo o tempo de entrega de pacotes entre nós vizinhos tende a crescer, uma vez que o nó transmissor tem que atrasar a transmissão de pacotes até que seu vizinho mude de estado. Com T_d assumindo valores menores o retardo tende a cair. Além disso, pode-se observar que essas outras curvas apresentaram comportamentos equivalentes.

As curvas que representam os menores valores atribuídos a T_d (0,1s e 0,05s) possuem comportamento diferenciado. O retardo para esses valores cresce com o aumento da taxa

de transmissão de pacotes. Tal crescimento é maior para T_d = 0,05s. Tal fato pode ser explicado pois quanto menor for o tempo que os nós ficam dormindo, menor será o tempo que eles gastam para completar seu ciclo de economia de energia. Isso aumenta o número de mensagens de *hello* que são transmitidas na rede. Tais mensagens são sempre transmitidas no início de um ciclo novo, tanto quando o nó decide dormir quanto quando ele decide ficar ativo. Com taxas de transmissão de pacotes maiores essas mensagens congestionam ainda mais a rede, aumentando o retardo. Esse congestionamento também influencia a taxa de descarte de pacotes, o que pode ser visto na figura 21. Neste gráfico nota-se que as curvas que representam os menores valores de T_d não proporcionam menor taxa de descarte de pacotes, como esperado.

O gráfico da figura 22 mostra número de pacotes que alcançaram o destino por unidade de energia, para os cinco valores de T_d analisados. Como todos os demais gráficos para análise dessa métrica, todas as curvas mostram melhora no desempenho com o aumento da taxa de transmissão. Pode-se notar que os melhores resultados são aqueles que foram obtidos quando T_d assumiu os maiores valores dentre os estudados. Conclui-se que quanto maior for o tempo que os nós permanecerem dormindo, maior será a economia de energia com o consumo ocioso e conseqüentemente aumentará a razão entre o número de pacotes úteis e a energia consumida.

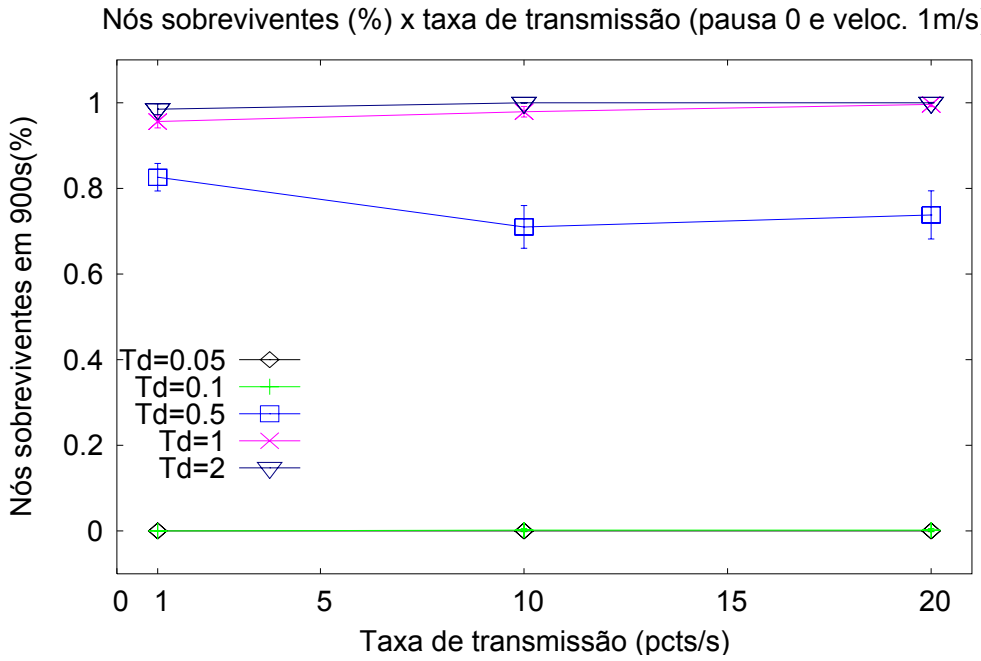

Figura 19: porcentagem de nós sobreviventes por taxa de transmissão de pacotes, para os cinco valores de T_d

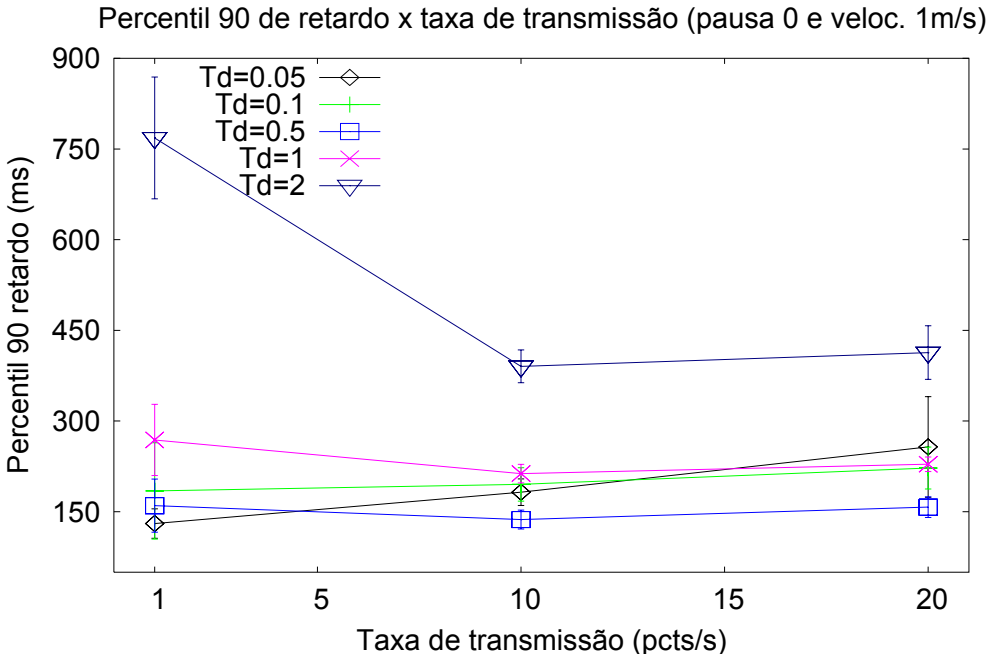

Figura 20: percentil 90 de retardo por taxa de transmissão, para os cinco valores de T_d

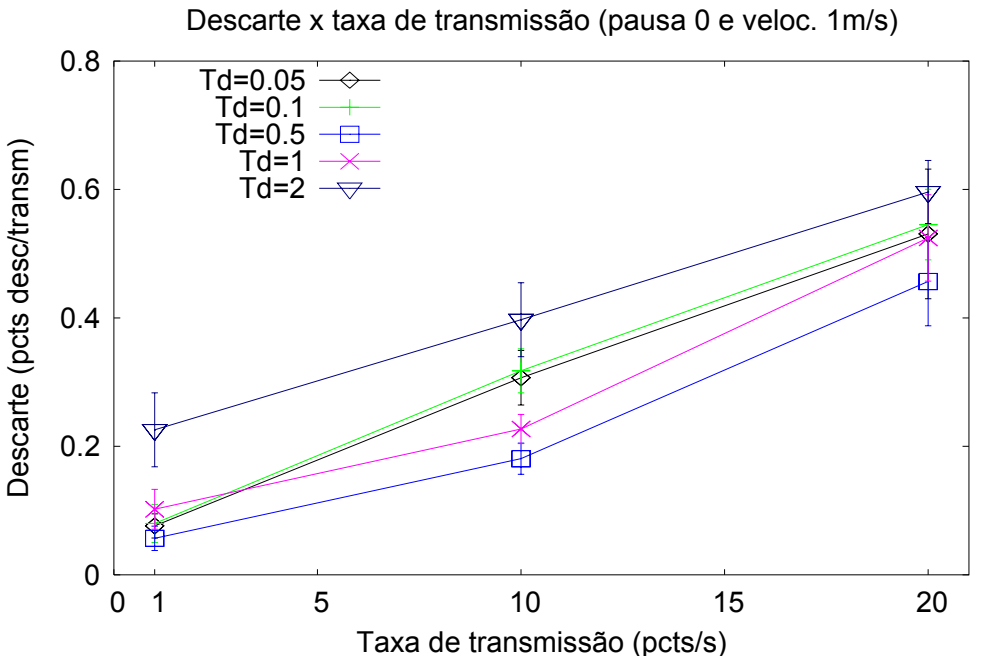

Figura 21: taxa de descarte de pacote por taxa de transmissão para os cinco valores analisados de T_d

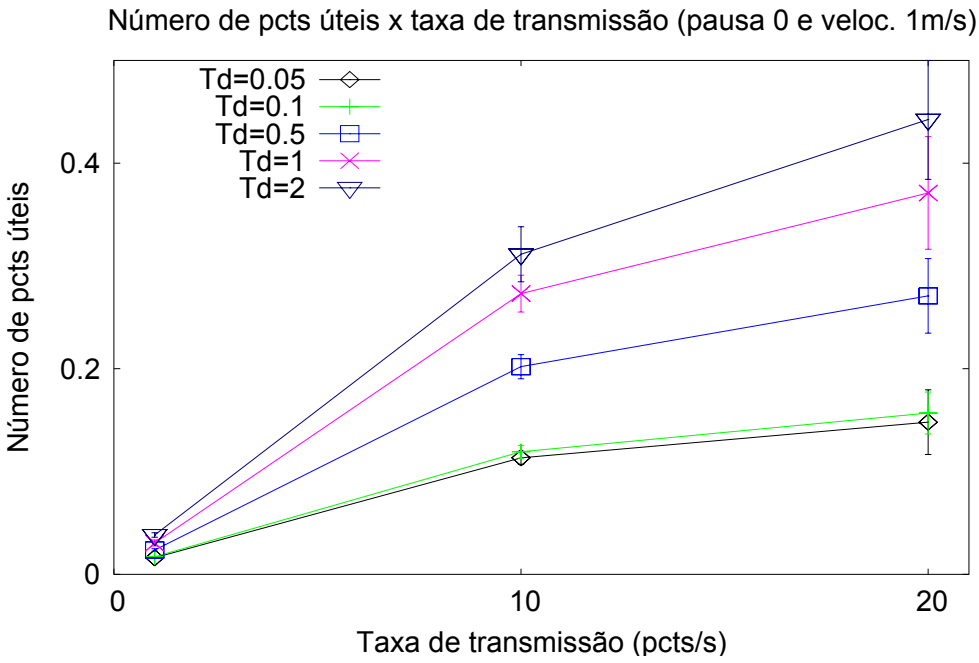

Figura 22: número de pacotes que alcançaram o destino com um Joule por taxa de transmissão, para os cinco valores analisados de T_d

A partir da análise dos resultados dos experimentos apresentados nesta seção, pode-se notar que T_d = 1s e Td = 2s fornecem os melhores resultados no que diz respeito ao número de nós sobreviventes em 900s de simulação e ao número de pacotes transmitidos por unidade de energia. Contudo, esses valores de T_d também levam a maiores taxas de descarte de pacotes e maiores retardos, em comparação aos demais valores analisados. O valor atribuído a este parâmetro que proporciona um bom compromisso em todas as métricas é T_d = 0,5s.

4.3.4) Variação do parâmetro T_e

Esse parâmetro corresponde ao tempo que os nós permanecem em estado *Escutando* durante o ciclo de economia de energia. Foram considerados os valores de 0,005s, 0,01s, 0,05s, 0,1s e 0,5s segundo. A figura 23 mostra o gráfico com a porcentagem de nós sobreviventes após 900s de simulação. Pode-se verificar que quanto maior o valor associado a T_e menor será a porcentagem de nós sobreviventes ao final da simulação, de forma que para o maior valor, T_e = 0,5s, ao final dos 900s de simulação nenhum nó se encontra vivo. Isso ocorre pois, quanto maior for o tempo que o nó escutar o meio, menor será sua economia de energia, o que

leva a mortes precoces por falta de energia. Um outro ponto que merece destaque é que para todos os valores atribuídos a esse parâmetro ocorreu degradação no desempenho com o aumento da taxa de transmissão de 1 para 10 pcts/s. Contudo, para taxas de transmissão maiores, o desempenho permanece praticamente constante. Tal fato pode ser explicado pois baixas taxas de transmissão de pacotes levam a um maior número de nós dormindo e, conseqüentemente, consomem menos energia.

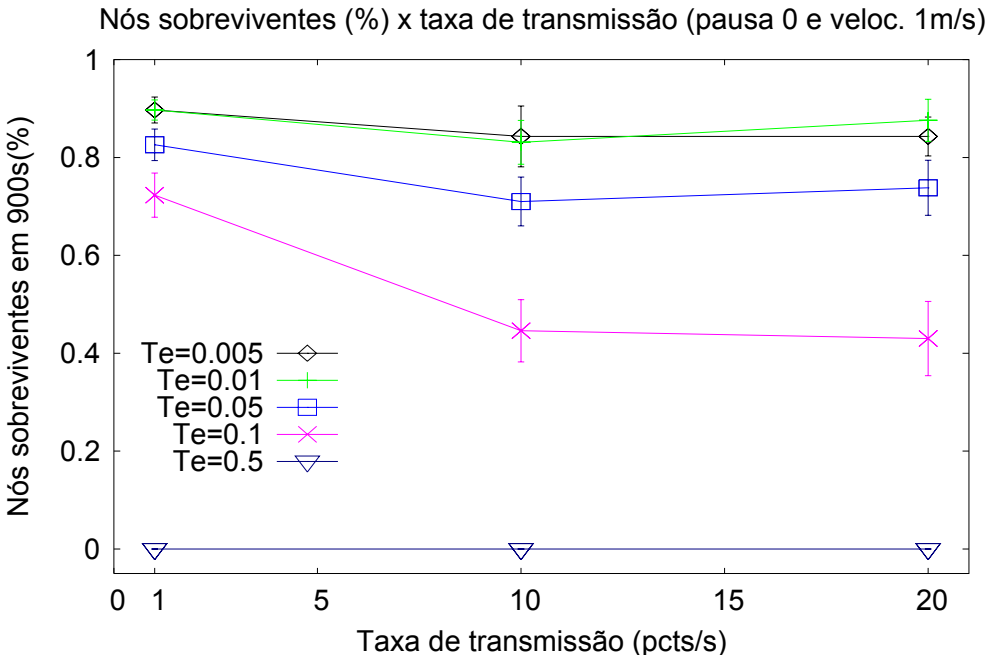

Figura 23: porcentagem de nós sobreviventes em 900s de simulação por taxa de transmissão de pacotes, para os cinco valores de T_e

O próximo gráfico, mostrado na figura 24, apresenta o retardo na entrega de pacotes para cada um dos valores associados a T_e. Os dados coletados com tais experimentos apresentaram uma variabilidade muito grande de resultados para todos os valores de T_e estudados. Isso pode ser verificando observando-se pelo tamanho das faixas de valores dos intervalos de confiança de cada curva. Verificou-se que tal comportamento recebe muita influência das variações de cenário. Seria necessário, então, aumentar o número de cenários utilizados para buscar valores mais conclusivos. Isso, contudo, implicaria em um aumento brutal no tempo de simulação. Para este estudo, foi considerado que dez execuções fornecem uma precisão adequada.

O único comportamento que se pode observar é que, para todos os valores de T_e, o retardo é menor quando a taxa de transmissão de pacotes está em torno de 10 pcts/s. Isso, como já mencionado anteriormente, pode ser explicado pelo fato de que com taxas de transmissão

menores os nós tendem a dormir mais e com taxas maiores ocorre congestionamento da rede, implicando em aumento no retardo.

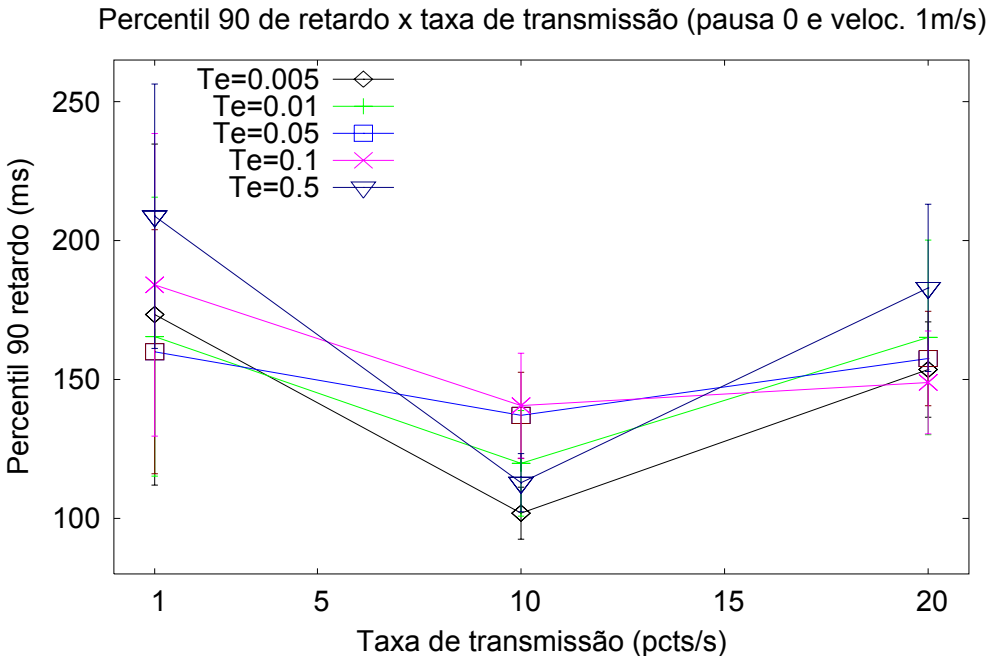

Figura 24: percentil 90 de retardo por taxa de transmissão, para os cinco valores de T_e

O gráfico da figura 25 mostra o descarte de pacotes para cada um dos valores atribuídos a T_e. Pode-se perceber que tal parâmetro influência pouco no descarte de pacotes, uma vez que para todas as taxas de transmissão de pacotes analisadas as curvas correspondentes à variação de T_e tiveram comportamento bastante semelhante. Importante ainda mencionar que em praticamente para todas as taxas de transmissão analisadas os intervalos de confiança de cada uma das curvas se interpõem, o que explica as inversões que ocorreram entre tais curvas.

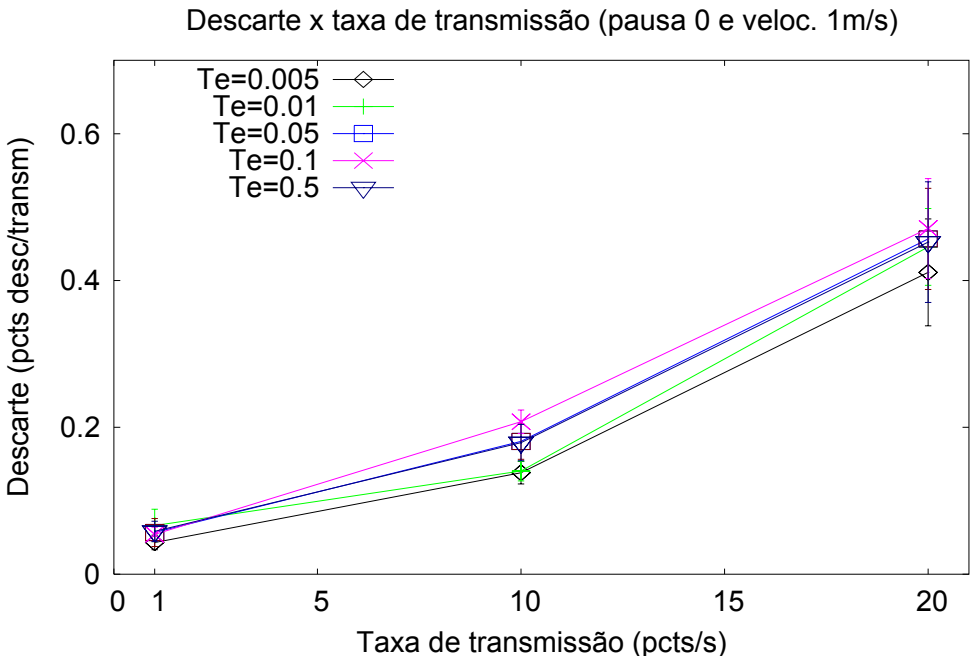

Figura 25: taxa de descarte de pacotes por taxa de transmissão de pacotes, para os cinco valores de T_e

O gráfico da figura 26 mostra o número de pacotes que alcançaram o destino por unidade de energia, para cada um dos valores associados a T_e. Nota-se, que quanto maior o valor atribuído a T_e, menor é o desempenho da rede em relação à utilização dos recursos energéticos. Com T_e menor, logicamente, os nós permanecem menos tempo escutando, de forma que a rede economiza a energia que seria gasta com o consumo ocioso. Gastando menos energia com consumo ocioso e mais com transmissão e recepção de pacote, o número de pacotes úteis da rede tende a aumentar, o que é mostrado no gráfico.

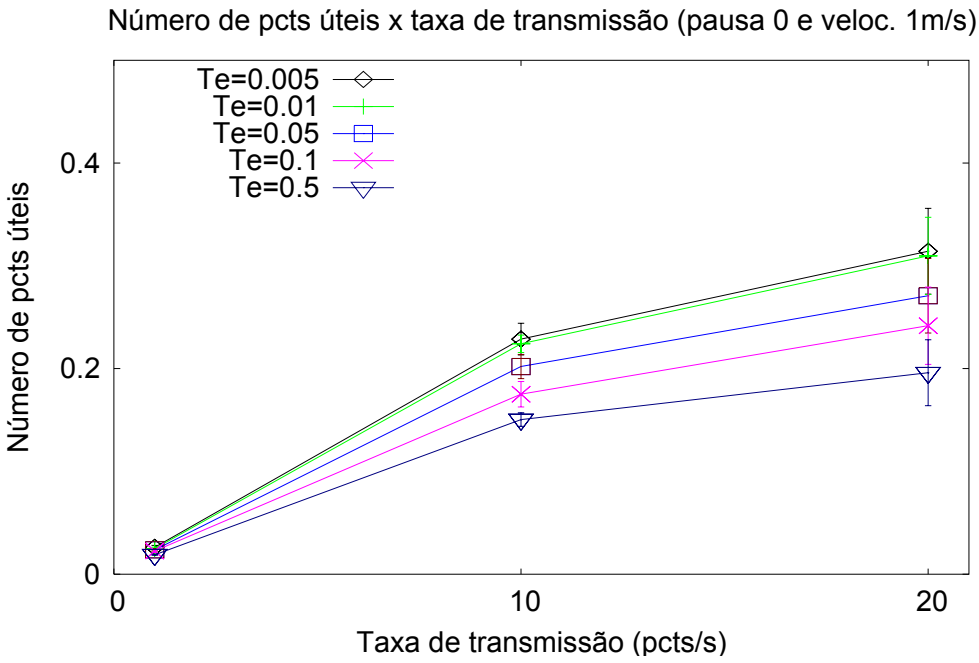

Figura 26: número de pacotes que alcançaram o destino com um Joule por taxa de transmissão de pacotes, para os três valores de T_e

Nesses experimentos pode-se notar que todos os valores atribuídos a T_e apresentaram comportamento semelhante quando a taxa de descarte de pacotes e o retardo foram analisados. Contudo, os gráficos que apresentam os resultados dos experimentos em relação ao número de pacotes úteis por unidade de energia e à porcentagem de nós sobreviventes em 900s de simulação mostram que quando T_e assumiu valores iguais a 0,05s, 0,01s e 0,005s, os resultados obtidos foram bastante semelhantes. Assim, T_e = 0,05 representa uma boa solução de compromisso.

4.3.5) Análise dos parâmetros

Esses experimentos foram criados com o intuito de buscar os valores que fossem associados aos parâmetros do *Extra* de forma a oferecer os melhores resultados possíveis. Para a escolha de tais valores, utilizou-se um critério de equilíbrio de perdas e ganhos, buscando-se com isso valores que economizassem energia da rede mas que não levassem a retardos maiores ou a altas taxas de descarte de pacotes.

A partir de análises dos gráficos, pode-se verificar que N = 15, T_a = 4s, T_d = 0,5s e T_e = 0,05s proporcionam boa economia de energia e razoáveis taxas de descarte de pacotes e de

retardo. Tais valores foram utilizados nos próximos experimentos, os quais visam validar a eficácia do algoritmo *Extra*, comparando seu desempenho com o algoritmo GAF e o protocolo AODV.

4.4) Análise das heurísticas

No capítulo 3 foi apresentado o algoritmo *Extra* onde foram implementadas cinco heurísticas para o cálculo do valor da probabilidade P, através de informações que os nós adquirem em tempo de execução. Para esta seção foram feitos experimentos para avaliar o desempenho destas heurísticas no cenário mencionado anteriormente.

O primeiro gráfico apresentado (figura 27) exibe o percentual de nós sobreviventes em 900s de simulação para as cinco heurísticas criadas. A que apresentou melhor desempenho foi a heurística 4 com cerca de 80% dos nós ainda vivos no final dos 900s. A heurística 5 também apresentou bom resultado, um pouco inferior ao apresentado pela 4. As demais heurísticas apresentaram piores resultados, com exceção da heurística 1 quando a taxa de transmissão estava em 1 pct/s. Nesse ponto, esta heurística apresentou desempenho equivalente às heurísticas 4 e 5.

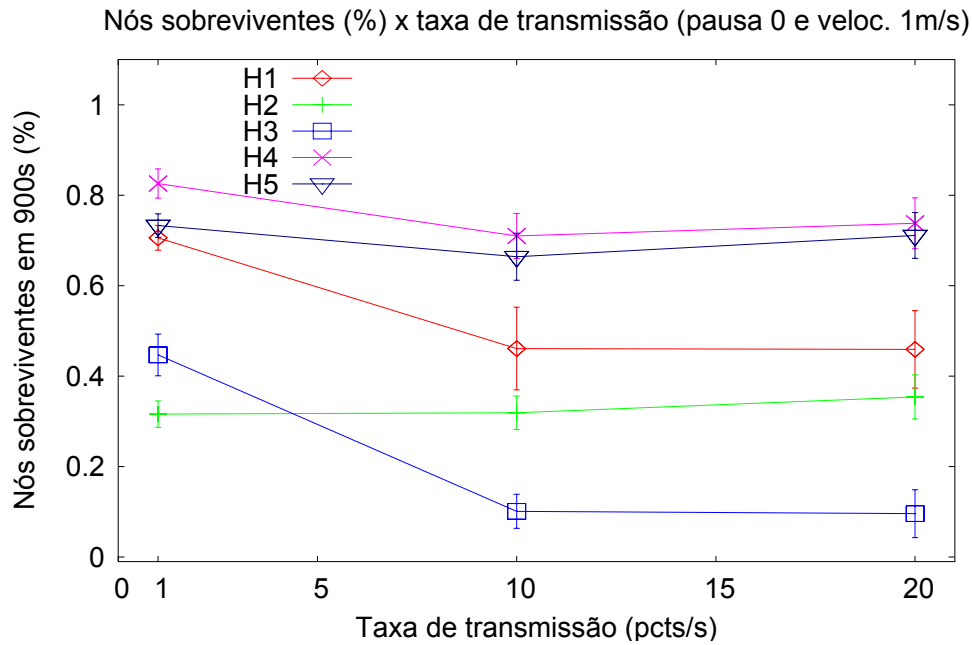

Figura 27: porcentagem de nós sobreviventes em 900s de simulação por taxa de transmissão, para as cinco heurísticas

O gráfico da figura 28 mostra o desempenho das heurísticas em relação ao retardo. O comportamento de todas as cinco heurísticas foi semelhante, apresentando maior retardo quando a taxa de transmissão estava em 1 e em 20 pcts/s, o que também pôde ser observado em todos os gráficos anteriores onde tal métrica foi analisada. Para taxas de 10 e 20 pcts/s, os resultados se encontram dentro da faixa de intervalo de confiança uns dos outros, o que mostra desempenhos equivalentes para todas as heurísticas. Para taxa de transmissão de 1 pct/s, o desempenho se diferencia, de forma que a heurística 5 ofereceu o maior retardo e a heurística 1 o menor. A heurística 4, que é utilizada como heurística básica nos outros experimentos, apresentou o segundo menor retardo.

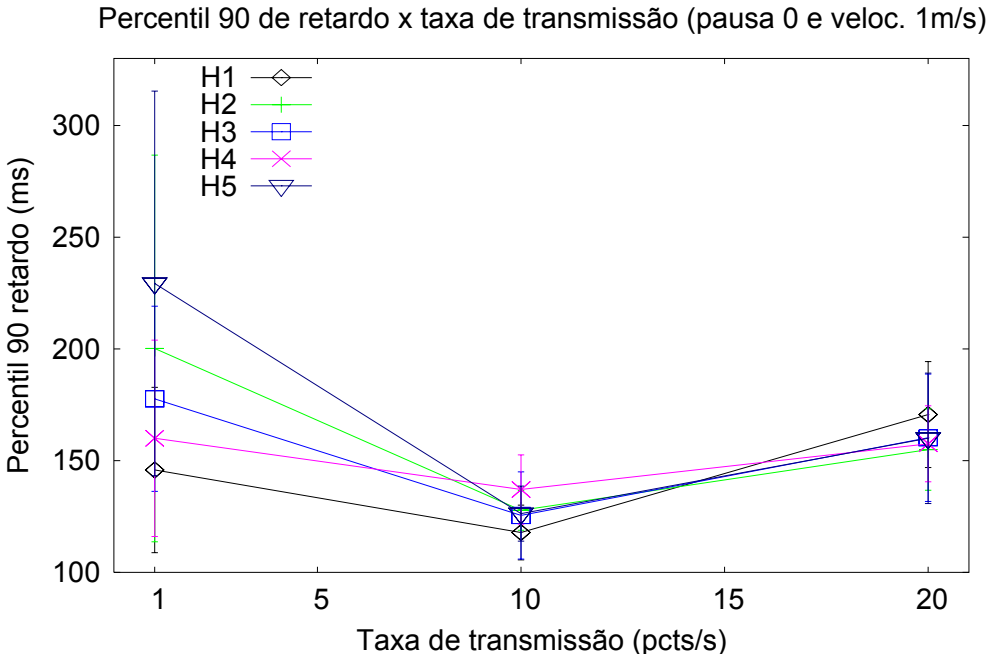

Figura 28: percentil 90 de retardo por taxa de transmissão, para as cinco heurísticas

O gráfico a seguir (figura 29) apresenta a taxa de descarte de pacotes para as cinco heurísticas. Todas aumentaram o descarte com o aumento da taxa de transmissão, fato fácil de ser entendido pois com maiores taxas o congestionamento da rede é maior, levando a uma maior taxa de descarte. Todas também apresentaram valores semelhantes de descarte para todas as taxas de transmissão estudas.

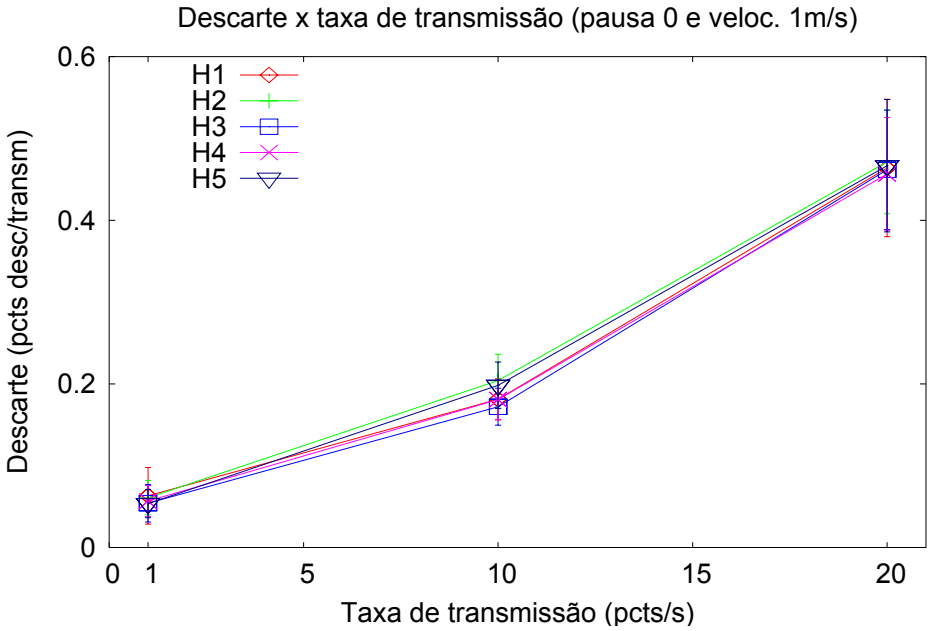

Figura 29: taxa de descarte de pacotes por taxa de transmissão, para as cinco heurísticas

O próximo gráfico, apresentado na figura 30, mostra o número de pacotes que foram transmitidos por unidade de energia (Joule). O desempenho de todas as cinco heurísticas melhora com o aumento da taxa de transmissão de pacotes, o que pôde ser observado em todos os experimentos anteriores que analisaram tal métrica. A heurística que apresentou o melhor desempenho foi a 4, para todas as taxas de transmissão de pacotes estudas. Com isso conclui-se que esta é a heurística que melhor faz uso dos recursos energéticos da rede, para os cenários estudados aqui. A heurística 3 apresentou os piores resultados, para todas as taxas de transmissão de pacote.

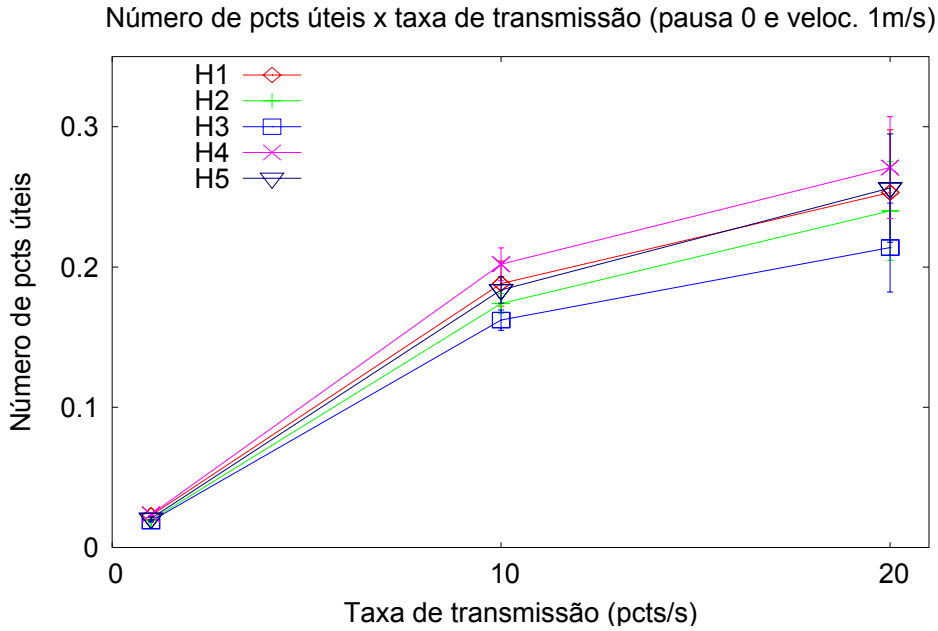

Figura 30: número de pacotes que alcançaram o destino com um Joule por taxa de transmissão de pacotes, para as cinco heurísticas

A partir desses experimentos apresentados pode-se verificar que a heurística que apresentou melhores resultados foi a 4. Esta heurística levou a uma maior economia de energia, o que pode ser comprovado pelo gráfico que mostra o número de sobreviventes em 900s de simulação (figura 27). Além disso, ela também se mostrou como a heurística que melhor faz uso dos recursos energéticos (figura 30). Com relação ao descarte, todas as heurísticas apresentaram resultados bastante semelhantes. Em relação ao retardo, todas as heurísticas também apresentaram comportamento semelhante para taxas de transmissão de pacote de 10 e 20 pcts/s. Para taxa de 1 pct/s a heurística 1 se destacou como a melhor e a 5 como a pior.

4.5) Validando o algoritmo *Extra*

O objetivo desses experimentos 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, sem a adaptação à mobilidade proposta no artigo original. A escolha do AODV foi feita com base em vários experimentos encontrados na literatura onde esse protocolo é utilizado como referência. O fato desse protocolo ser classificado como reativo

também é um fator importante para a sua escolha, uma vez que tais tipos de protocolo consomem menos energia que os protocolos pró-ativos, de acordo com [Cano 00]. O GAF foi escolhido pois, além de oferecer um bom desempenho e de ser bem referenciado na literatura, sua implementação já se encontra no ns-2, o simulador de redes utilizado nos experimento. Importante mencionar que o GAF foi utilizado em sua forma padrão, não sendo considerada a adaptação à mobilidade sugerida em seu artigo original.

Tanto as heurísticas quanto os valores dos parâmetros utilizados para esses experimentos foram escolhidos a partir dos experimentos apresentados na seção 4.1. Ou seja, foram utilizados a heurística 4 e os parâmetros N = 15, T_a = 4s, T_d = 0,5s e T_e = 0,05s para todos os experimentos a seguir. A partir daí foram variadas as características do cenário para se verificar o desempenho do algoritmo, sendo utilizados como valores básicos a taxa de transmissão de pacotes de 10 pcts/s, velocidade de 1 m/s e tempo de pausa de 0 segundo.

As velocidades utilizadas foram 0 m/s, representando uma rede estática, 1 m/s, representando o equivalente a uma pessoa caminhando, e 10 m/s, representando um veículo se movimentando em ambiente urbano. Também, para esses movimentos, 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 nenhum movimento (pausa de 900s). As taxas de transmissão de pacotes avaliadas foram 1 pacote/s, 10 pacotes/s e 20 pacotes/s, totalizando em cada simulação taxas de 2, 20 e 40 pacotes/s. Como medidas de desempenho foram utilizadas as mesmas das seções anteriores, ou seja:

1. percentual de nós que sobreviveram após 900 segundos de simulação;
2. o retardo na transmissão de pacotes entre a fonte e o dreno;
3. taxa de descarte de pacotes da rede; e
4. número de pacotes que conseguiram alcançar o destino (pacotes úteis) por unidade de energia gasta.

4.5.1) Percentual de nós sobreviventes durante a simulação

Esse experimento (figura 31) mostra o percentual de nós que ainda tinham alguma energia a cada instante dos 900 segundos de simulação. A primeira curva que chama a atenção é a que apresenta os resultados do protocolo AODV. Com ela pode-se observar que depois de cerca de 450 segundos de simulação todos os nós da rede morrem praticamente ao mesmo tempo. Isso ocorre pois o consumo ocioso degrada continuamente os recursos energéticos da rede. Os resultados obtidos com o GAF, coincidentemente, foram muito próximos aos

resultados obtidos com o *Extra* utilizando a heurística 1. Pode-se verificar em tais curvas que ao final da simulação cerca de 46% dos nós ainda tinham energia e que os nós começaram a morrer só a partir de 550 segundos, aproximadamente.

O *Extra* utilizando as heurísticas 2 e 3 leva a resultados piores que o GAF, sendo que ao final da simulação apenas cerca de 35% e 10% dos nós, respectivamente, ainda tinham alguma energia. Contudo, o *Extra* utilizando as heurísticas 4 e 5 apresenta os melhores resultados, ultrapassando o resultado obtido com o GAF. A heurística 4 se destacou como a que mais economiza energia e mais aumenta o tempo de vida da rede. Os valores apresentados são o resultado das médias obtidas em cada série de 10 experimentos.

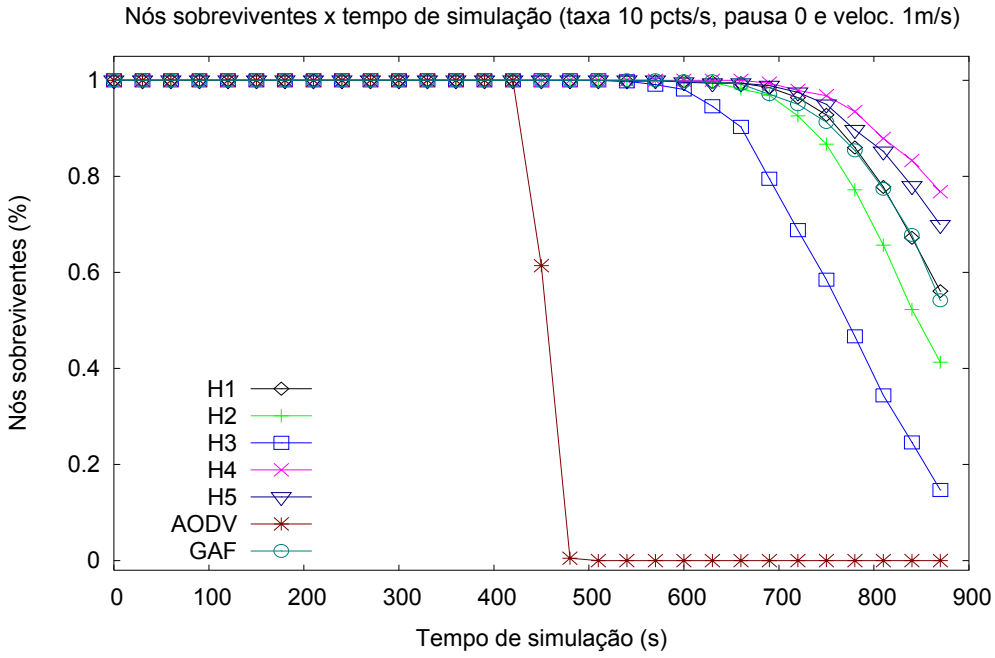

Figura 31: nós sobreviventes por tempo de simulação para as cinco heurísticas do *Extra*, para o GAF e para o AODV

4.5.2) Percentual de nós sobreviventes ao final da simulação

Os gráficos a seguir apresentam o resultados dos experimentos que mediram o número de nós que sobreviveram após 900s de simulação. Para tais experimentos o protocolo AODV não será utilizado como base de comparação pois após 500s de simulação todos os nós da rede já haviam morrido por falta de energia. Pode-se verificar no gráfico a seguir, figura 32, os resultados obtidos variando-se o tempo de pausa. Para todos os valores obtidos o *Extra*, além de

se mostrar menos sensível à variação deste parâmetro, apresentou melhor desempenho que o GAF, sendo em alguns pontos ele se mostra várias vezes mais eficiente.

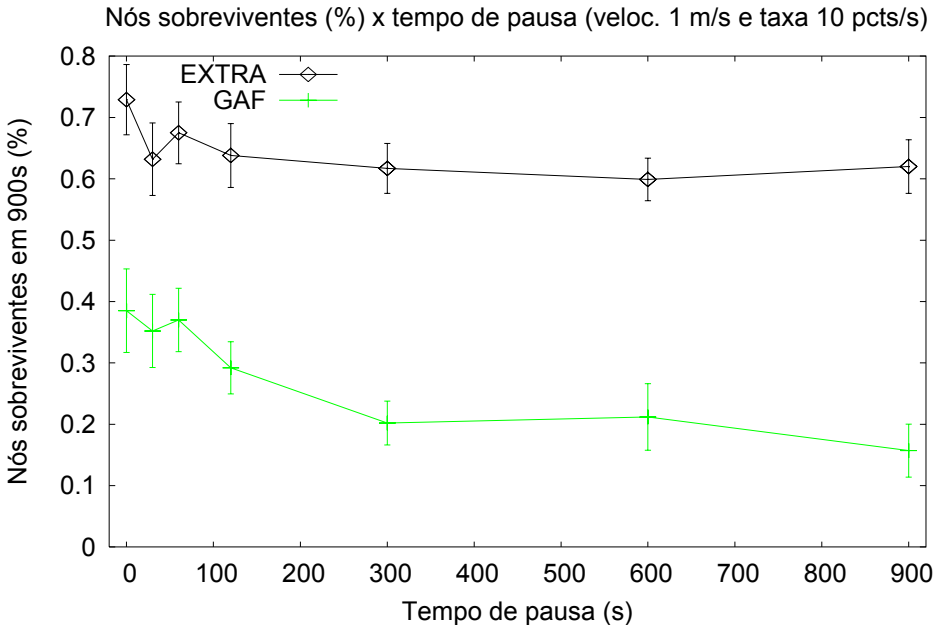

Figura 32: percentual de nós sobreviventes após 900s de simulação variando-se o tempo de pausa

A figura 33 mostra o gráfico com o percentual de nós sobreviventes após 900 segundos, variando-se a taxa de transmissão de pacotes em 1, 10 e 20 pcts/s. Pode-se verificar, na curva que representa o GAF, que com o aumento da taxa de transmissão de pacotes os nós gastam mais energia e o número de nós sobreviventes ao final da simulação é menor. Tal comportamento só pode ser observado no *Extra* para taxas entre 1 e 10 pcts/s, e para a faixa de 10 a 20 pcts/s esse algoritmo praticamente não alterou seu desempenho. Verifica-se que para todas as taxas estudas o *Extra* apresenta melhor desempenho que o GAF.

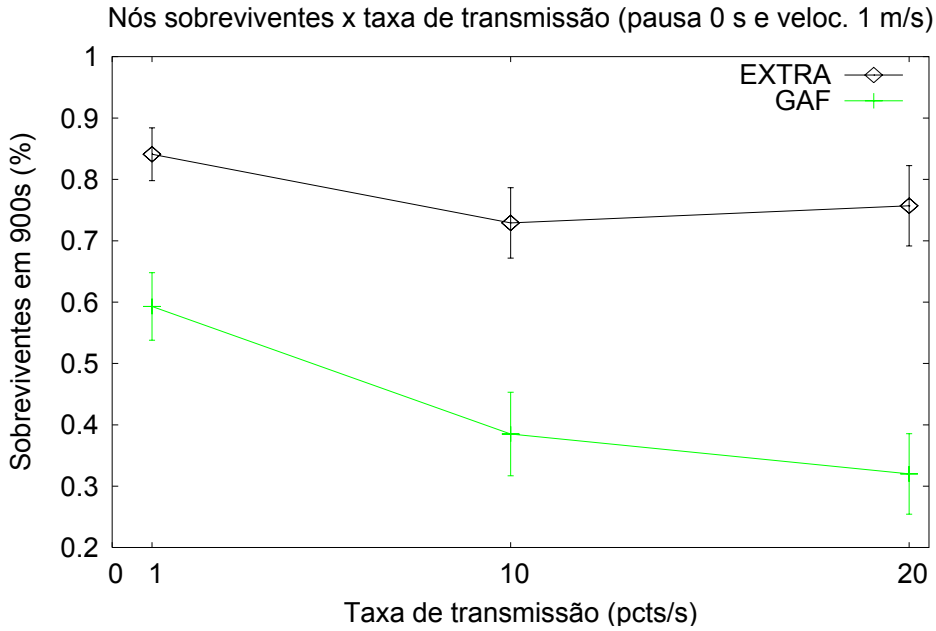

Figura 33: percentual de nós sobreviventes após 900s de simulação variando-se a taxa de transmissão de pacotes

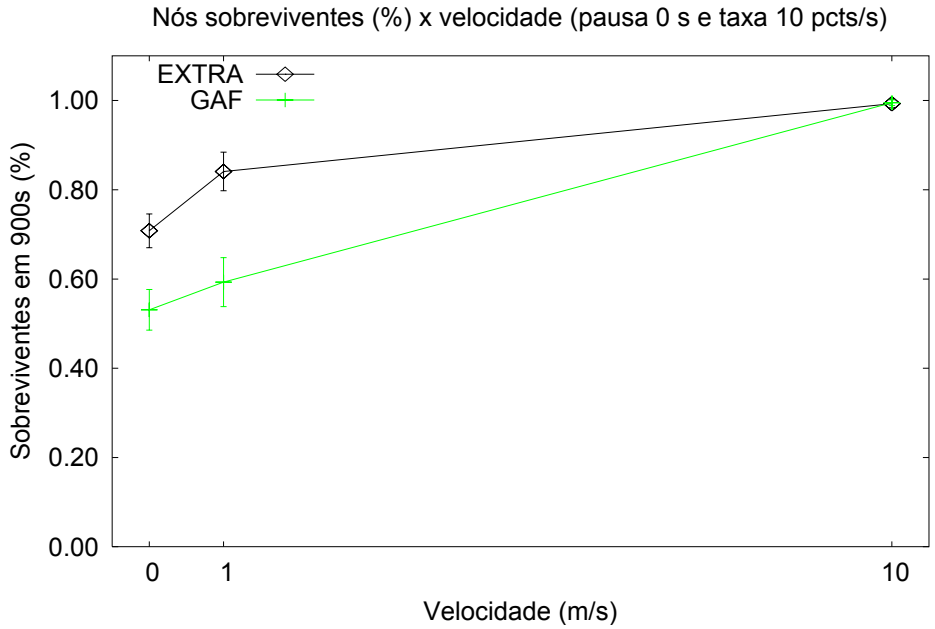

Figura 34: percentual de nós sobreviventes após 900s de simulação variando-se a velocidade dos nós

A figura 34 mostra o gráfico com o percentual de nós sobreviventes após 900 segundos de simulação, variando-se a velocidade dos nós entre 0, 1 e 10 m/s. Vê-se que tanto para o *Extra* quanto para o GAF o percentual de nós sobreviventes cresceu com o aumento da velocidade, sendo que para o GAF tal crescimento se apresenta mais acentuado, a ponto de se tornar equivalente ao desempenho do *Extra* quando velocidades maiores são admitidas. Tal fato ocorre pois quando um pacote é descartado no meio de uma rota os demais nós dessa rota não gastam energia com sua transmissão, ou seja, a rede a economiza energia. Com o aumento da velocidade a taxa de descarte de pacotes é maior para os dois algoritmos (figura 40), conseqüentemente, a porcentagem de nós que sobreviveram a 900s de simulação também é maior.

4.5.3) Percentil 90 de retardo

A seguir são apresentados os resultados obtidos medindo-se o percentil 90 do retardo nas transmissões de pacotes entre os nós origem e destino. Através da análise da figura 35 pode-se verificar que AODV e o GAF apresentam resultados equivalentes e que o *Extra* aumenta o retardo para qualquer um dos valores de tempo de pausa estudados, em relação aos demais algoritmos. Esse aumento do retardo é conseqüência da economia de energia, a qual é alcançada com a dormência dos nós, conseqüentemente aumentando o retardo na entrega de pacotes.

O gráfico da figura 36 mostra o percentil 90 do retardo variando-se a taxa de transmissão de pacotes. Para o AODV, com o aumento da taxa de transmissão de pacotes ocorre um aumento do retardo. Para o *Extra* e o GAF, o retardo é maior para taxas de transmissão de 1 e 20 pcts/s. Com taxas mais baixas, um maior número de nós adormece, aumentando-se o retardo. Já para taxas mais altas o que proporciona o aumento no retardo é o congestionamento da rede. O *Extra* apresentou desempenho equivalente ao GAF tanto para taxas menores quando para maiores, uma vez que as faixas de valores dos intervalos de confiança de suas curvas se sobrepõem. Já para taxas intermediárias o *Extra* apresenta maior retardo que os demais.

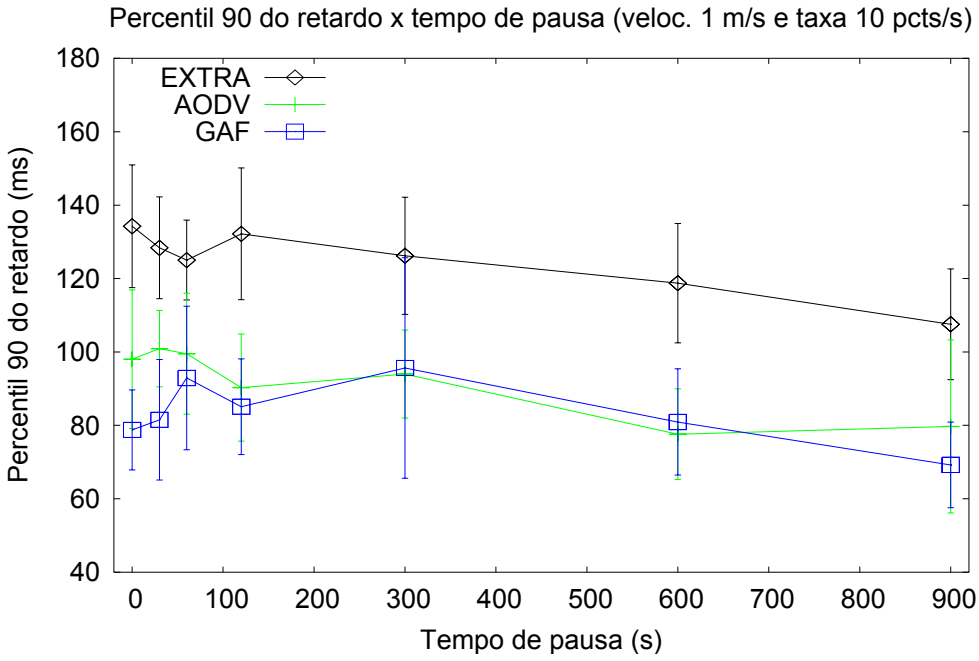

Figura 35: percentil 90 do retardo variando-se o tempo de pausa

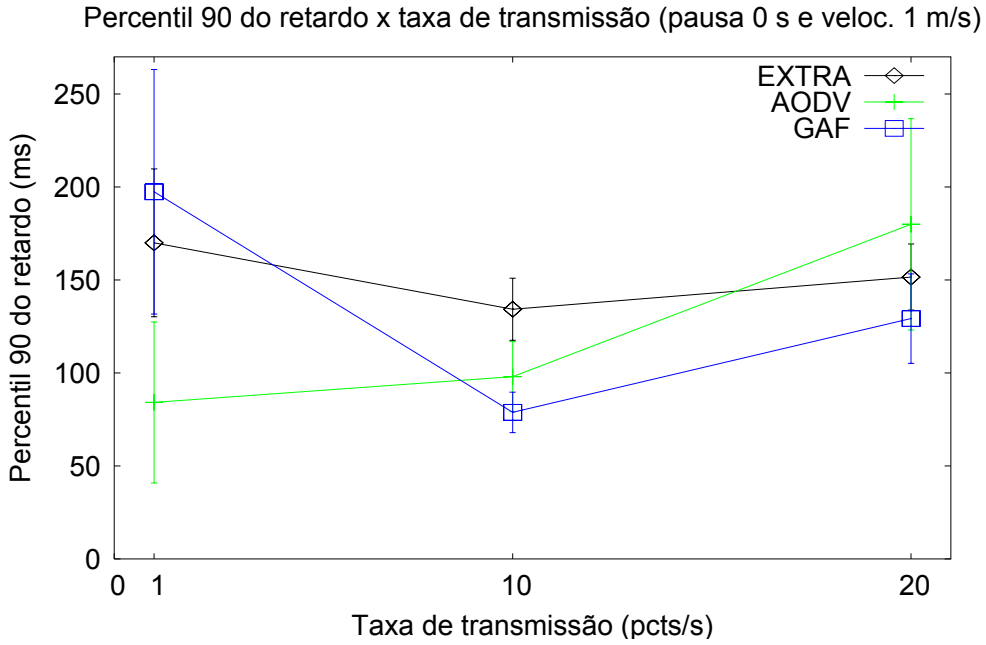

Figura 36: percentil 90 do retardo variando-se a taxa de transmissão de pacotes

A figura 37 mostra a influência da variação da velocidade dos nós sobre o retardo. Torna-se importante mencionar que para todos os experimentos o GAF utilizado segue

sua forma padrão, não sendo considerada sua adaptação à mobilidade, como mostrado em seu artigo original. Esse algoritmo sofreu uma degradação acentuada de desempenho com o aumento da velocidade, se mostrando muito sensível a ela. O mesmo não ocorre com o *Extra*, que mostrou um comportamento levemente inferior ao do AODV. Isso mostra que tal algoritmo, em relação ao retardo e ao aumento da velocidade dos nós, não altera as características do protocolo sobre o qual ele foi implementado.

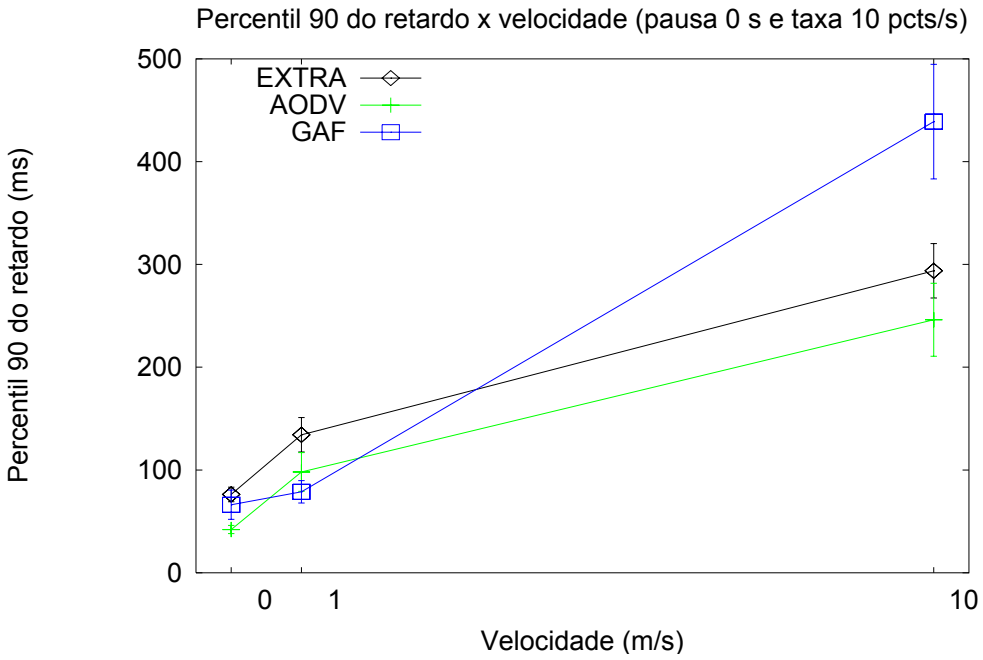

Figura 37: percentil 90 do retardo variando-se a velocidade dos nós

Com a análise dos resultados obtidos com esses experimentos, pode-se verificar que o *Extra* tem um pior comportamento em relação ao retardo que os outros algoritmos. Contudo, tal comportamento é esperado uma vez que para se economizar energia esse algoritmo adormece os nós da rede, o que leva ao aumento do retardo. Quanto maior for o número de nós adormecidos na rede, e o tempo que eles permanecem nesse estado, maior será o retardo na entrega de pacotes e, conseqüentemente, a economia de energia.

4.5.4)Descarte de pacotes

Os resultados a seguir foram obtidos medindo-se a taxa de descarte de pacotes nas simulações. Essa taxa é a razão entre o total de pacotes descartados e o total de pacotes transmitidos pelas fontes. No gráfico da figura 38 pode-se verificar que para todos os tempos de pausa estudados o *Extra* apresentou taxa de descarte de pacotes mais elevada que os outros algoritmos. O grande número de nós que são adormecidos com este algoritmo aumenta o descarte de pacotes, característica que pode ser encontrada em vários outros algoritmos que utilizam tal técnica para economizar de energia.

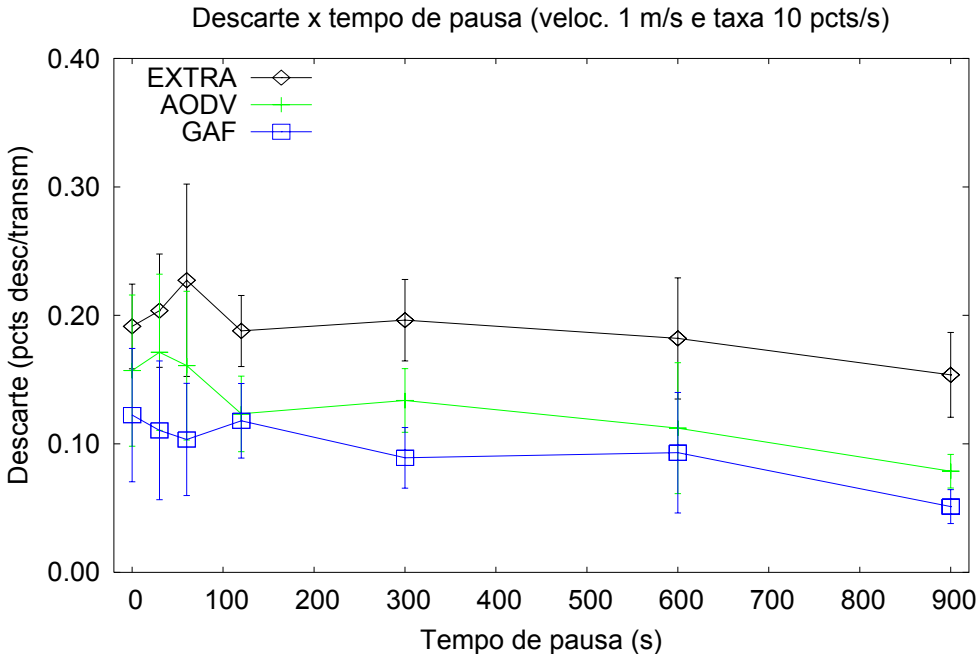

Figura 38: descarte de pacotes variando-se o tempo de pausa

O gráfico da figura 39 mostra o descarte de pacotes em relação ao aumento da taxa de transmissão. Pode-se verificar que o *Extra* apresenta um maior descarte de pacotes em relação aos demais algoritmos, para todas as taxas estudadas. Contudo, pode-se verificar que a taxa de descarte de pacotes do *Extra* sempre aumentou na mesma proporção que a do AODV, o protocolo sobre o qual ele foi implementado. Esse comportamento mostra o impacto causado pelo adormecimento de alguns nós sobre o descarte de pacotes.

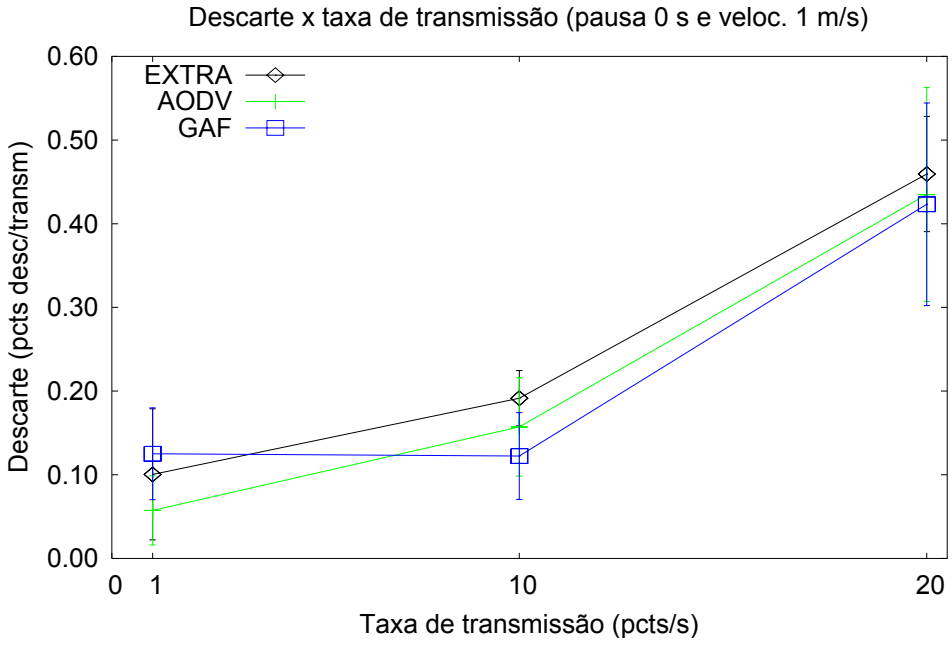

Figura 39: descarte de pacotes variando-se a taxa de transmissão de pacotes

O gráfico a seguir, figura 40, mostra o descarte de pacotes em relação ao aumento da velocidade dos nós. O *Extra*, para todas as velocidades estudadas, apresentou taxa de descarte um pouco maior que o AODV, havendo, como esperado, um incremento do descarte com o aumento da velocidade. Contudo, esse aumento foi sempre proporcional ao aumento ocorrido com o AODV. O GAF apresentou baixas taxas de descarte para velocidades menores, mas tal taxa cresceu acentuadamente quando velocidades mais altas eram admitidas, ultrapassando o descarte dos outros algoritmos.

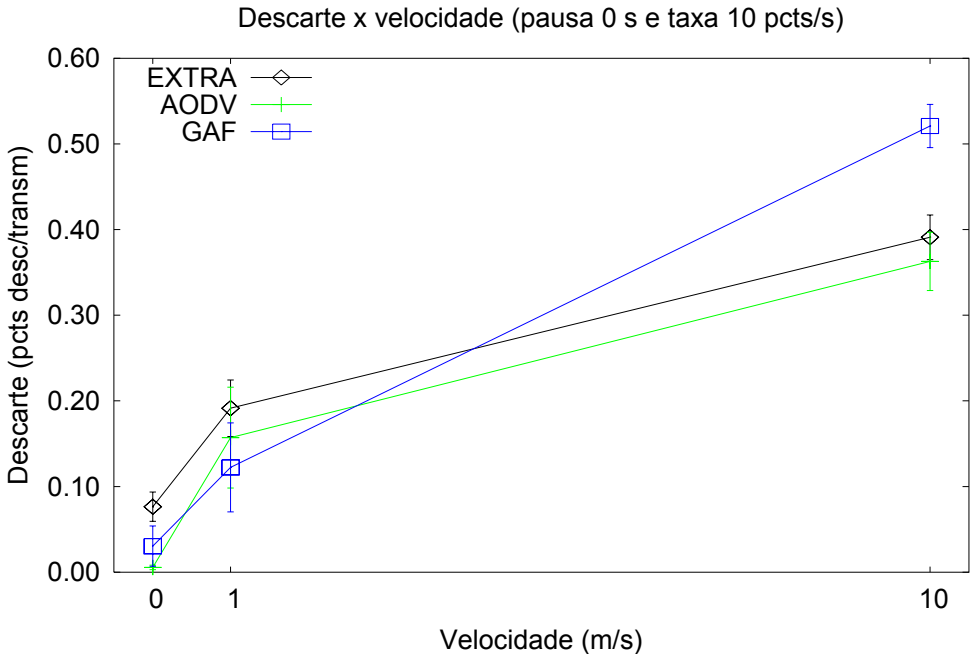

Figura 40: descarte de pacotes variando-se a velocidade dos nós

Pode-se verificar que o descarte de pacotes aumentou com a utilização do *Extra* em relação ao AODV puro. Contudo esse aumento foi sempre proporcional ao ocorrido com o AODV para a variação tanto da velocidade dos nós quanto da taxa de transmissão de pacotes. Tal fato mostra que, em relação ao descarte de pacotes, o *Extra* não altera as características do protocolo sobre o qual ele foi implementado.

4.5.5) Número de pacotes úteis por unidade de energia

Nos experimentos a seguir foram contados quantos pacotes foram transmitidos pelas fontes e conseguiram alcançar seu destino, ou seja, verificou-se o número de pacotes úteis da rede. Os gráficos a seguir apresentam a razão entre esse valor e o total de energia gasta em cada simulação. Pretende-se com isso obter o número de pacotes que alcançaram o destino por unidade de energia gasta na rede. Pode-se concluir que aquele algoritmo que apresentar melhores resultados é aquele que melhor utiliza os recursos energéticos da rede.

O gráfico da figura 41 mostra o comportamento dos três algoritmos para tal medida de desempenho com a variação do tempo de pausa. O *Extra*, para todos os tempos de pausa estudados, apresentou melhores resultados, seguido pelo GAF que obteve desempenho pouco superior ao desempenho apresentado pelo AODV.

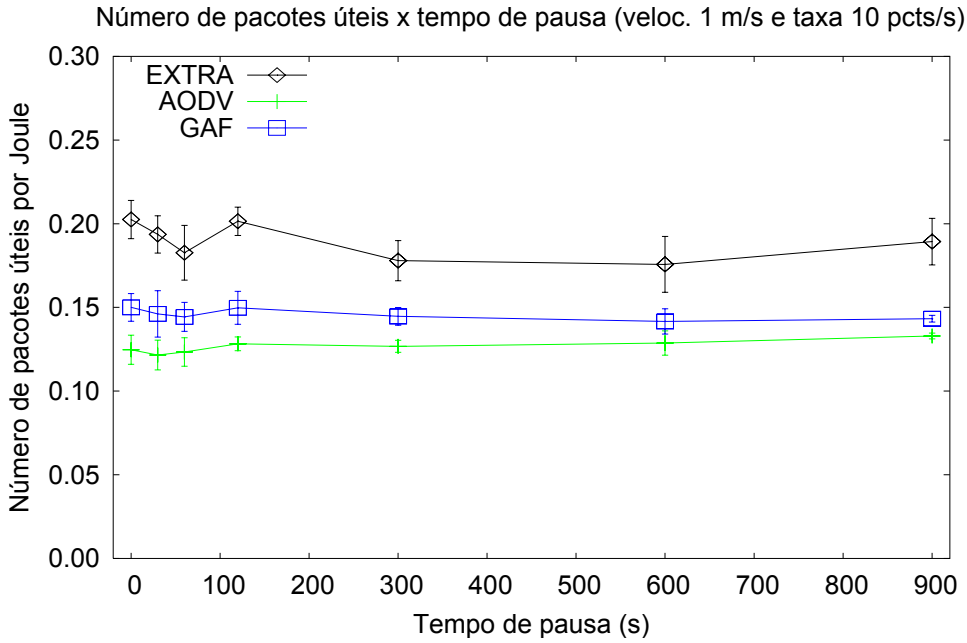

Figura 41: número de pacotes úteis por unidade de energia variando-se o tempo de pausa

O gráfico da figura 42 mostra o número de pacotes úteis por unidade de energia com a variação da taxa de transmissão de pacotes. A primeira coisa a se notar é que para taxas menores de transmissão de pacotes todos os algoritmos apresentaram seus piores resultados. Isso é proporcionado pelo consumo ocioso, pois com taxas de transmissão mais baixas seu consumo energético supera, em muito, o consumo energético proporcionado pela transmissão e recepção de pacotes. Quando ocorre um aumento na taxa de transmissão de pacotes, o impacto do consumo ocioso diminui em relação à troca de mensagens. Para todas as taxas estudadas o *Extra* apresentou os melhores resultados, melhorando seu desempenho em relação aos demais com o aumento da taxa de transmissão de pacotes.

No gráfico da figura 43 temos o comportamento dos três algoritmos para o número de pacotes úteis por unidade de energia com a variação da velocidade dos nós. Verifica-se que o *Extra* foi superior aos outros algoritmos para todas as velocidades estudadas. Nota-se também que o aumento da velocidade não levou a grandes variações do desempenho. Isso se deve ao fato de que o ráio de alcance dos rádios dos nós é grande (250 m) em relação às velocidade admitidas, com isso a topologia não se altera muito rapidamente.

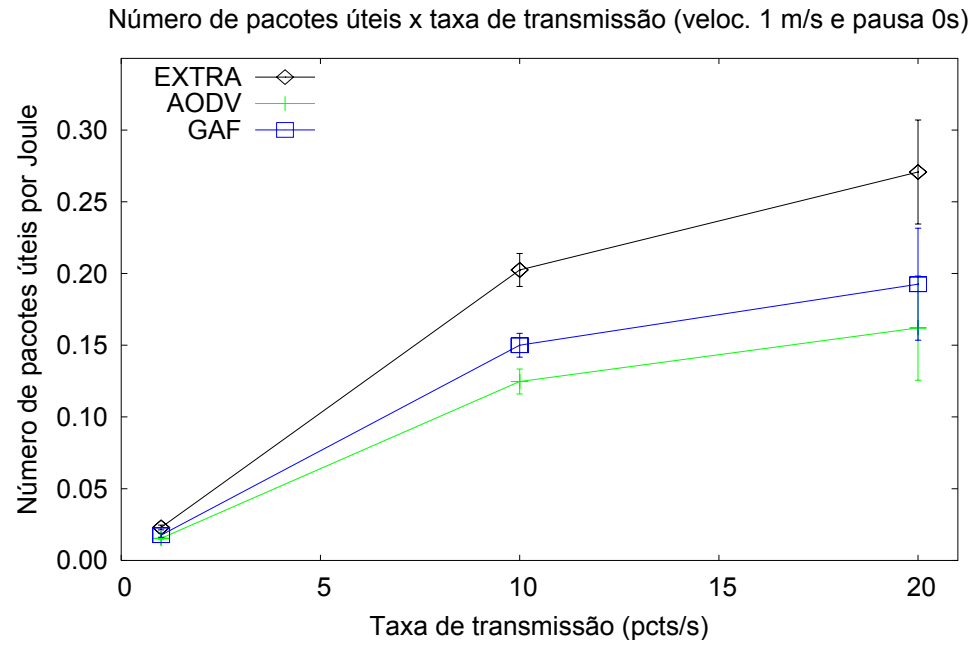

Figura 42: número de pacotes úteis por unidade de energia variando-se a taxa de transmissão de pacotes

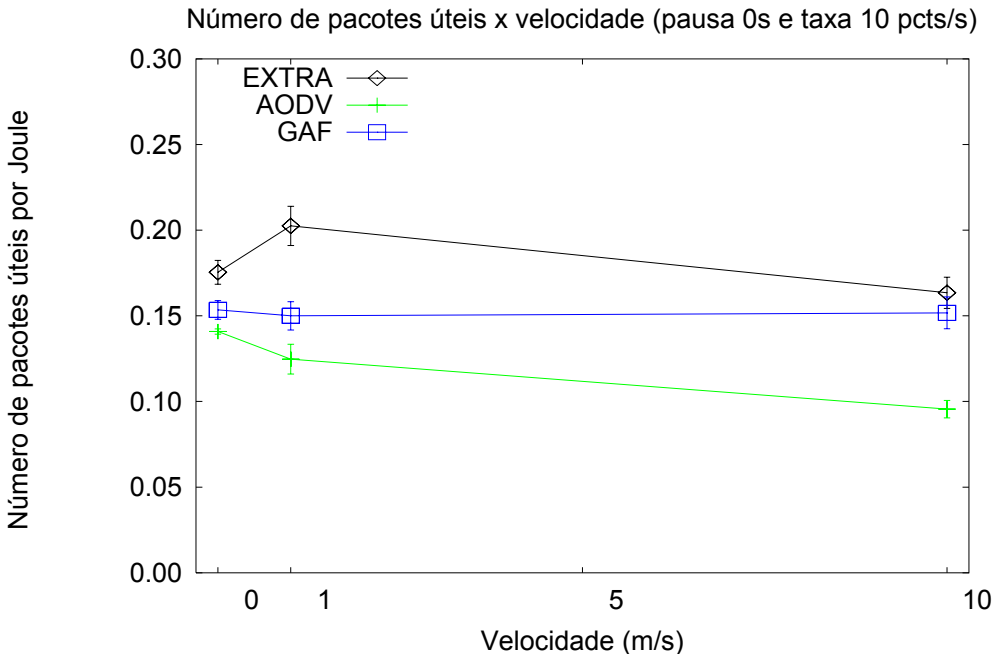

Figura 43: número de pacotes úteis por unidade de energia variando-se a velocidade dos nós

Com a análise desses gráficos, pode-se verificar que o *Extra* apresentou melhor desempenho que os demais algoritmos em todos os experimentos realizados. Pode-se concluir que o *Extra* utiliza melhor os recursos da rede que os outros algoritmos aqui comparados, diminuindo o consumo ocioso e fazendo com que a energia seja efetivamente utilizada com transmissões e recepções de pacotes.

4.5.6) Análise dos experimentos de validação do Extra

Com a análise destes experimentos pode-se verificar que o *Extra* realmente aumenta o tempo de vida da rede. No gráfico que apresenta o número de nós ainda com energia a cada instante de simulação, pode-se perceber que o tempo de vida da rede tende a ser maior com o *Extra* do que os demais algoritmos estudados. Isso mostra uma melhor distribuição do consumo de energia entre os nós da rede.

Nos gráficos que apresentam o percentual de nós sobreviventes ao final dos 900s de simulação vê-se que o *Extra* aumenta tal percentual. Isso mostra que sua utilização realmente economiza a energia da rede. Contudo, como já esperado, tal economia eleva a taxa de descarte e o retardo na entrega de pacotes, uma vez que um maior número de nós adormece. No entanto, para verificar se a economia proporcionada pelo *Extra* é compensatória, foram criados os experimentos que mostram o número de pacotes que alcançam o destino com uma unidade de energia. Neles verifica-se que o *Extra* utiliza melhor os recursos energéticos que os demais algoritmos estudados, ou seja, economiza energia que seria gasta com o consumo ocioso e faz com que a rede utilize esta energia economizada com trabalho útil (transmissão e recepção de pacotes).

4.6) Analisando os riscos de desconexão da rede

4.6.1) Comportamento da heurística 4 para várias taxas de transmissão

Como mencionado anteriormente, o algoritmo *Extra* se baseia em um mecanismo que determina de forma probabilística quais os nós da rede que devem adormecer e quais permanecerão acordados mantendo a conectividade da rede. Com isso, poderiam ocorrer situações onde todos os nós adormeçam ao mesmo tempo levando a uma completa desconexão da rede.

Para analisar os riscos desta desconexão a estratégia utilizada foi contar, a cada segundo de simulação, o número de nós que estão com seus rádios ligados. Foram considerados para isso aqueles nós que estiverem no estado *Ativo* ou no estado *Escutando*. Nessa contagem não são incluídos nem os nós em estado *Dormindo,* nem os que já morreram por falta de energia. A partir daí, é feita a razão entre o número de nós com o rádio ligado e o número de nós ainda vivos na rede. O valor obtido nesta operação mostra a porcentagem de nós vivos, que estão mantendo a conectividade da rede.

Nos gráficos gerados a partir desses dados, para oferecer uma melhor visualização, os intervalos de confiança não foram inseridos e as curvas indicam a média dos valores coletados nos experimentos. Foi utilizado um valor para pausa de 0 segundo e admitiu-se que os nós sempre tinham velocidade de 1 m/s. Os parâmetros N, T_a, T_d e T_e receberam os valores considerados básicos nos experimentos da seção 4.3.

Primeiramente serão apresentados os gráficos que mostram o comportamento da rede quando a heurística 4 é utilizada para taxas de transmissão de pacotes de 1 pct/s (figura 44), 10 pcts/s (figura 45) e 20 pcts/s (figura 46). Em todos esses gráficos pode-se verificar que, inicialmente, a porcentagem de nós que adormecem simultaneamente é grande (cerca de 80% do total). Com o passar do tempo esse número começa a diminuir até atingir um patamar, ficando em torno de 50%, que é mantido por quase todo restante da simulação. Esse comportamento é explicado pelo fato de que as transmissões são iniciadas no instante zero quando todos os nós estão acordados. Nesse ponto duas rotas são estabelecidas e utilizadas pelas duas fontes de dados existentes na rede. Contudo, a grande maioria dos nós não faz parte destas rotas, e por isso estão ociosos desde o início da simulação. Depois de T_a segundos tais nós terão probabilidade P de dormir. Nessa situação os nós têm muita energia e, de acordo com o funcionamento da heurística 4, o valor de P é calculado a partir do número de vizinhos de cada nó. Como a rede possui alta densidade, o valor de P é alto, resultando assim em um grande número de nós adormecendo. Com o passar do tempo outras rotas vão sendo estabelecidas e por isso outros nós vão acordando, até chegar a um ponto onde a variação entre os nós em operação de instante para instante é muito menor, permanecendo assim por quase todo o final da simulação. Depois de cerca de 750s de simulação o número de nós em operação começa a cair, conseqüência dos nós que vão morrendo por falta de energia. Tal comportamento é mais acentuado para taxas de transmissão mais altas (10 e 20 pcts/s)

Pode-se considerar então que o *Extra* passa por uma fase de ajuste no início das simulações, chegando depois a um valor de P que seja mais apropriado para as condições da rede. O período de duração desta fase varia de acordo com o a taxa de transmissão, o que pode-se notar

nos três gráficos a seguir. Quanto maior for a taxa de transmissão, menor será a fase de ajuste. Depois disso, o número de nós em operação varia em menor amplitude com o passar do tempo, mas sempre dentro de uma faixa limitada de valores, mostrando que a rede adquiriu uma certa estabilidade, a qual só é quebrada quando alguns nós no final da simulação começam a morrer por falta de energia.

Partindo-se das análises feitas nesses gráficos pode-se perceber que não existe o risco de desconexão completa da rede, uma vez que a porcentagem de nós em operação sempre varia dentro de uma pequena faixa estabelecida durante a fase inicial da simulação, ou seja, sempre existirão nós ativos. Contudo, podem ocorrer situações isoladas onde uma certa região da rede fica desconexa, deixando somente alguns nós fora da rede. Porém, essa situação é momentânea uma vez que os nós que estiverem dormindo acordarão em algum momento, seja para entrar no estado Escutando, seja para permanecer no estado Ativo.

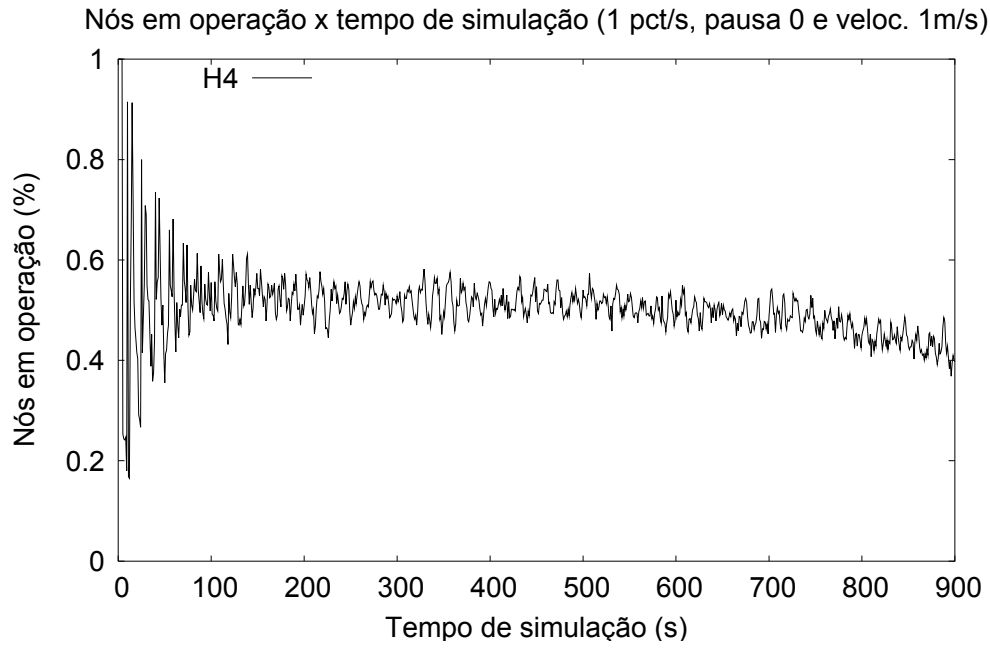

Figura 44: nós ativos em cada instante da simulação, para a heurística 4 e taxa de transmissão de pacotes de 1 pct/s

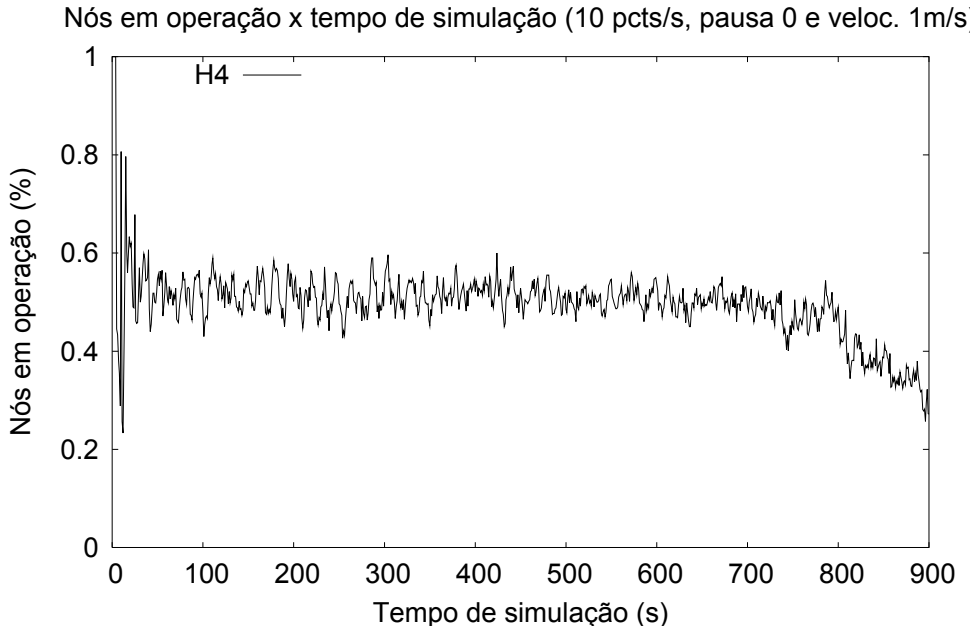

Figura 45: nós dormindo em cada instante da simulação, para a heurística 4 e taxa de transmissão de pacotes de 10 pct/s

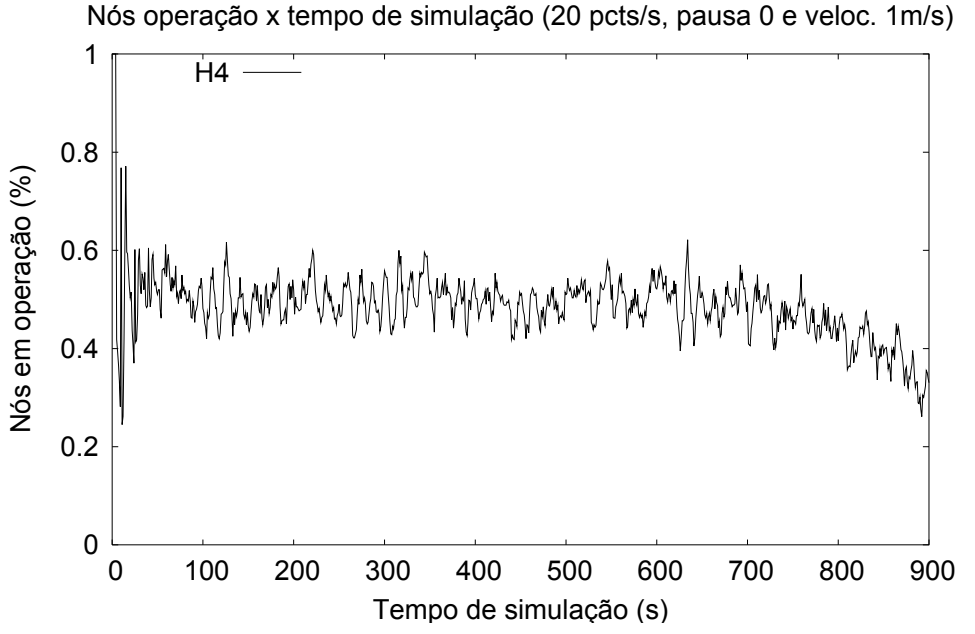

Figura 46: nós dormindo em cada instante da simulação, para a heurística 4 e taxa de transmissão de pacotes de 20 pct/s

4.6.2) Analisando o comportamento de cada heurística

Inicialmente foi analisado o comportamento da heurística 4 com a variação da taxa de transmissão de pacotes. Será agora analisado o comportamento das demais heurísticas em relação à porcentagem de nós em operação durante todo o tempo de simulação. Os gráficos a seguir mostram os resultados obtidos, utilizando para isso taxa de transmissão de 10 pcts/s.

Com a análise dos gráficos verifica-se que as heurísticas 4 e 5 se adaptam melhor às condições da rede. Isso pode ser observado pelo fato de que durante praticamente todo o tempo de simulação, quando os nós da rede possuem boas reservas energéticas, a percentagem de nós em operação varia dentro de uma pequena faixa fixa. Contudo, ao final da simulação quando alguns nós já estão mortos por falta e energia, a porcentagem de nós em atividade tende a cair, economizando assim a pouca energia que ainda resta nos nós vivos.

Na figura 47 temos o comportamento da heurística 1. Pode-se perceber que esta heurística também faz com que a rede tenha uma fase de ajuste nos instantes iniciais da simulação, passando por uma fase de estabilidade que vai até o final da simulação, onde é mantida uma pequena faixa de variação da porcentagem de nós em operação. O mesmo comportamento pode ser observado para a heurística 2 na figura 48, uma vez que esta heurística obtém o valor de P para cada nó através da divisão do número de vizinhos ativos pelo número total de vizinhos. Contudo, para esta heurística o número de nós em operação na rede durante a fase de estabilidade é maior do que para as heurísticas 4 e 5, o que leva a um maior consumo de energia por parte da heurística 2.

De acordo com as análises feitas na seção 4.4, a heurística 3 proporcionou os piores desempenhos dentre todas. Para esta heurística o valor de P é inversamente proporcional à quantidade de energia existente em cada nó naquele instante. O gráfico da figura 49 apresenta o comportamento desta heurística em relação à porcentagem de nós em operação durante o período de funcionamento da rede. Pode-se notar que essa porcentagem começa muito alta e vai caindo com o passar do tempo. Isso ocorre porque os nós iniciam com muita energia e com o passar do tempo tais reservas vão sendo utilizadas. O valor de P vai aumentando de acordo com o que cada nó vai consumindo de energia, fazendo com que a porcentagem de nós em operação diminua. Ao final da simulação, como o número de nós vivos é pequeno, qualquer alteração no número de nós em operação se torna significativa, o que leva a uma maior variação da porcentagem de nós em atividades.

Na figura 50 temos o comportamento da heurística 5, que é muito parecido com o comportamento da heurística 4. Isso comprova o motivo pelo qual elas apresentaram desempenho bastante similar durante as análises da seção 4.4.

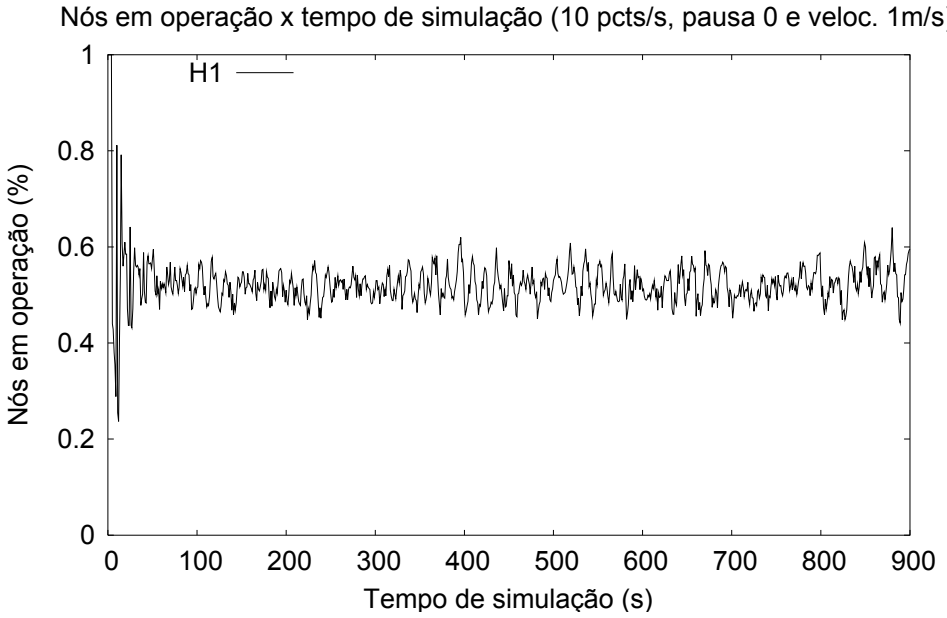

Figura 47: nós dormindo em cada instante da simulação, para a heurística 1 e taxa de transmissão de pacotes de 10 pct/s

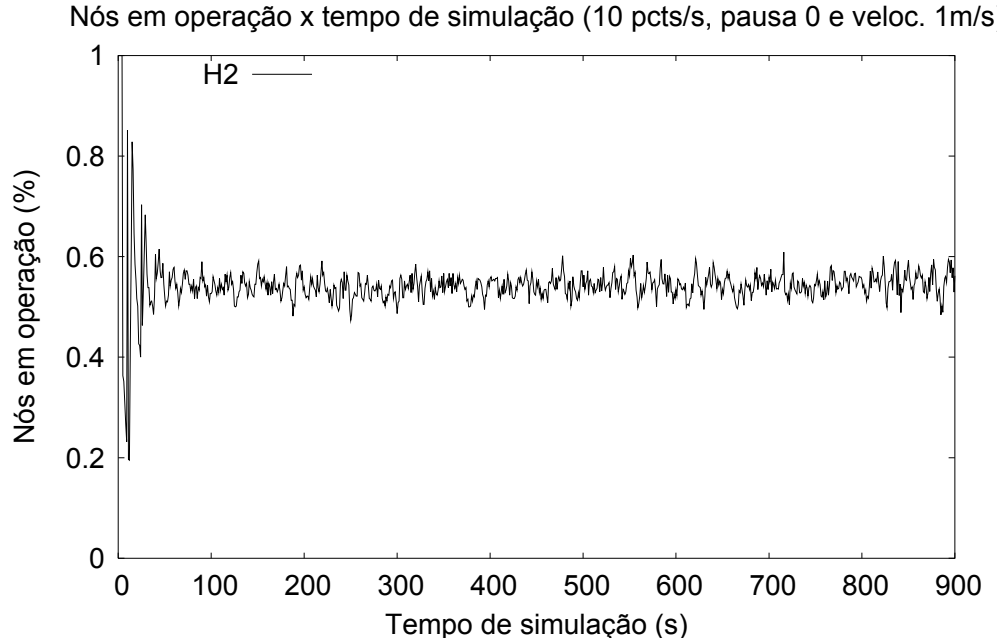

Figura 48: nós dormindo em cada instante da simulação, para a heurística 2 e taxa de transmissão de pacotes de 10 pct/s

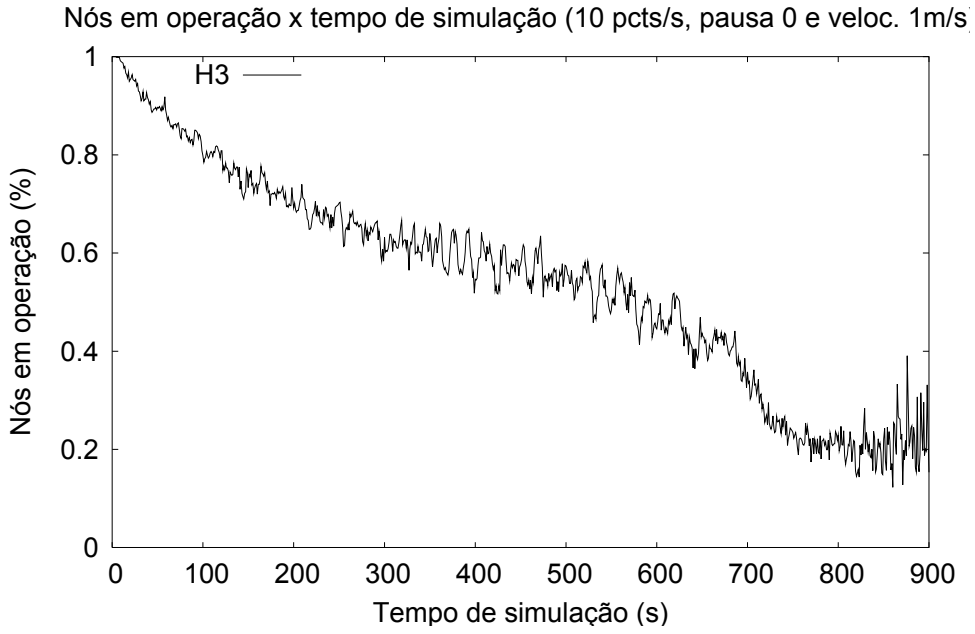

Figura 49: nós dormindo em cada instante da simulação, para a heurística 3 e taxa de transmissão de pacotes de 10 pct/s

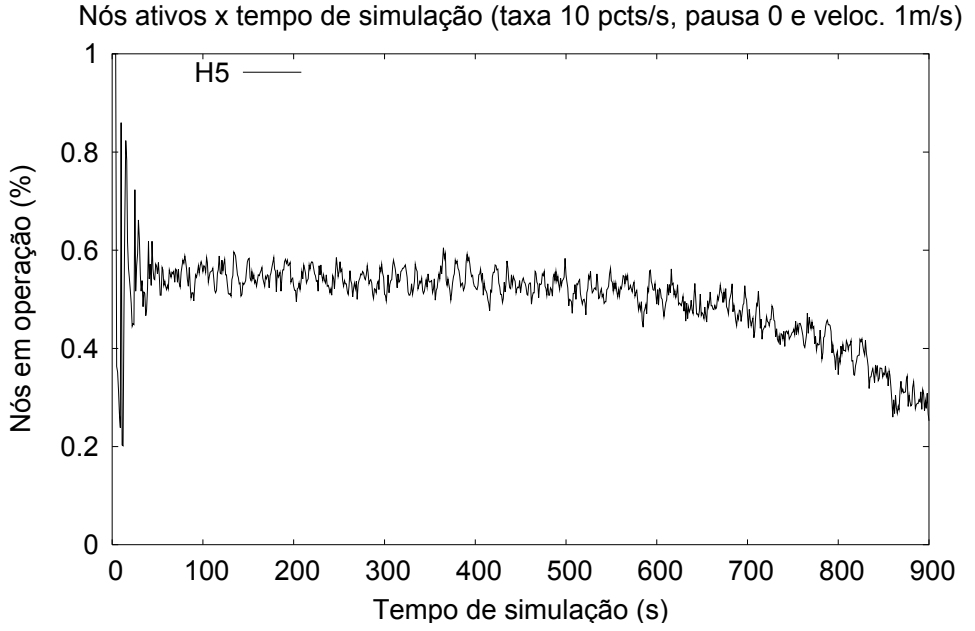

Figura 50: nós dormindo em cada instante da simulação, para a heurística 5 e taxa de transmissão de pacotes de 10 pct/s

4.7) Análise dos experimentos

Esses experimentos foram criados para buscar valores dos parâmetros do *Extra* que oferecessem melhores resultados. De posse de tais valores, vários experimentos foram criados com o intuito validar a eficácia do algoritmo, comparando-o com outros algoritmos já conhecidos na literatura. Na última seção de experimentos, analisa-se os risco de desconexão da rede, uma vez que o *Extra* usa um mecanismo probabilístico para adormecer os nós da rede.

Nota-se que os parâmetros do *Extra* influenciam sensivelmente no desempenho da rede e na economia de energia. Verificou-se que os parâmetros adquiridos durante a fase de implementação do algoritmo, e utilizados como parâmetros básicos, levam a bons resultados e por esse motivo foram utilizados nos experimentos que comprovaram a eficácia do Extra.

A utilização desse algoritmo realmente economiza energia. Além disso, tal economia ocorre de forma distribuída entre os nós, o que faz aumentar o tempo de vida da rede. Em comparações com o AODV e com o GAF, o *Extra* apresentou melhores resultados em praticamente todos os experimentos, mostrando que ele faz melhor uso dos recursos energéticos da rede. Contudo, o preço pago pela economia de energia é o aumento do retardo e da taxa de descarte de pacotes. No entanto, esses aumentos podem ser controlados com a escolha adequada dos parâmetros. Verificou-se também que o *Extra* oferece baixo risco de desconexão da rede, uma vez que um percentual quase constante de nós sempre permanece ativo, mantendo a conectividade .

Capítulo 5 - Conclusões e Trabalhos Futuros

Esse trabalho apresenta um novo algoritmo, *Extra*, para economizar energia e prolongar o tempo de vida de redes *ad hoc*. Sua função é coordenar a transição dos nós para o estado de dormência, onde o consumo energético é muito inferior ao consumo quando esses nós estão ativos. Nesse último estado, os nós têm a necessidade de manter energizados os dispositivos eletrônicos que compõem seus rádios para continuarem tendo a capacidade de receber transmissões. Esta característica faz com que os nós consumam energia mesmo quando não estão participando de transmissões ou recepções de pacotes, o chamado consumo ocioso. O *Extra* evita o consumo ocioso adormecendo determinados nós da rede, ou seja, desligando os rádios de alguns desses nós por períodos preestabelecidos. A escolha de quais nós devem adormecer ocorre de forma distribuída. Cada nó executa uma função que estabelece com probabilidade P se deve desligar seu rádio ou não. O valor de P é obtido em tempo de execução, através de heurísticas bem simples que utilizam somente informações locais, que podem ser adquiridas em tempo de execução e que não são obtidas com o auxílio de outros dispositivos, como GPS. Devido à sua simplicidade, o *Extra* pode ser implementado sobre qualquer tipo de protocolo de roteamento para redes *ad hoc* e ser executado por *hardwares* bem simples.

Esse algoritmo foi implementado sobre o protocolo de roteamento AODV [Perk 99]. Foram efetuadas simulações com o intuito de comprovar sua eficácia. Através delas, verificou-se que a utilização do *Extra* proporciona uma economia considerável de energia e que o tempo de vida da rede é prolongado, em comparação com o AODV em sua forma original. Nosso algoritmo também foi comparado com outro algoritmo para economia de energia já conhecido na literatura, o GAF [Xu 01], o qual também foi executado sobre o AODV. Pode-se verificar que o *Extra* obteve melhores resultados que o GAF em praticamente todos os experimento relacionados à economia de energia e extensão do tempo de vida da rede.

Como o *Extra* possui um mecanismo probabilístico que determina quais os nós adormeceram, foram criados experimentos para analisar os riscos de uma possível desconexão total da rede. Verificou-se que durante as simulações sempre existirão nós acordados e que a quantidade de nós nesta situação sempre varia dentro de uma pequena faixa de valores. Isso mostra que o risco de desconexão total da rede é muito baixo. Contudo, existe o risco de desconexão parcial, ou seja, uma certa região pode ficar sem comunicação. Porém, esse estado é temporário, uma vez que necessariamente os nós mudam de estado. Verifica-se também que, no início das simulações, o número ativos varia muito até o algoritmo encontre um valor adequado

para P (fase de ajuste). Como trabalhos futuros, pode-se criar mecanismos ou ajustar os parâmetros para que o tempo gasto pela fase de ajuste seja reduzido.

Como visto, todas as heurísticas aqui propostas utilizam informações simples e locais. Em trabalhos futuros poderiam ser criadas novas heurísticas que utilizam informações mais complexas e com maior dificuldade de aquisição, como, por exemplo, o número de vizinhos de cada nó vizinho, o estado energético da vizinhança, a movimentação do nó e de sua vizinhança e o fluxo de passagem de pacotes pelos nós. Contudo, a utilização dessas novas informações pode gerar novas mensagens de controle, o que aumenta o tráfego da rede e conseqüentemente o consumo de energia. Será, então, imprescindível a análise do impacto causado por tais mensagens no desempenho da rede.

Durante os experimentos foi utilizado um cenário de dimensões e número de nós fixos, onde foram variados somente parâmetros relacionados à movimentação dos nós, à taxa de transmissão de pacotes e parâmetros que determinam os tempos nos estados pelos quais os nós passam quando estão executando o *Extra*. Vários outros tipos de cenários, com características bem distintas, podem ser criados, para os quais a eficácia do algoritmo pode ser analisada.

Como mencionado anteriormente, o *Extra* foi implementado sobre o protocolo AODV. Qualquer protocolo de roteamento de redes *ad hoc* pode ser utilizado para esta implementação. Porém, as características do algoritmo apresentado nesse trabalho fazem com que ele se adapte melhor a protocolos reativos. No entanto, não é impossível sua implementação sobre protocolos pró-ativos. Além disso, a utilização do *Extra* não está restrita a redes móveis *ad hoc*. Sua implementação pode ser feita sobre protocolos de redes de sensores. Nesse tipo de rede os nós são a origem das transmissões, a quais são direcionadas a estações base que são computadores em pontos fixos que não têm restrições quanto à energia [Cord 02].

De qualquer forma, os trabalhos realizados até o momento comprovam que a energia de redes *ad hoc* pode ser economizada sem causar grandes impactos em seu desempenho. Para isso, a simplicidade na utilização de informações e na criação de heurísticas se mostrou eficiente, o que pôde ser comprovado pelos experimentos apresentados nesse trabalho. No entanto, muito ainda pode ser criado, melhorando ainda mais os resultados obtidos até o momento.

Capítulo 6 - Referências

[Bonn 03] "BonnMotion: A Mobility Scenario Generation and Analysis Tool", http://www.informatik.uni-bonn.de/IV/BonnMotion/.

[Cama 99] Câmara, D., Loureiro, A. A. F., "Redes de Computação Móvel Ad Hoc", curso apresentados nas Jornadas de Atualização em Informática, SBC, Brasília - DF, 28 p., Julho 1999.

[Cano 00] Cano, J.C., Manzoni, P., "A Performance Comparison of Energy Consumption for Mobile Ad Hoc Network Routing Protocols", 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, San Francisco – Califórnia, EUA, p. 57-64, Agosto 2000.

[Chen 02] Chen, B., Jamieson, K., Balakrishnan, H.,Morris, R., "*Span*: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", ACM Wireless Networks Journal, vol. 8, p. 481-494, Setembro 2002.

[Cord 02] Cordeiro, C.M., Agrawal, D.P., "Mobile Ad Hoc Networking", 20º Simpósio Brasileiro e Redes de Computadores, Búzios - RJ, p. 125-187, Maio 2002.

[Cunh 03] Cunha, O.D., Costa, L.H.M.k., Duarte, O.C.M.B., "Um Mecanismo de Roteamento para o Consumo Balanceado de Energia em Redes Móveis Ad Hoc", 21º Simpósio Brasileiro e Redes de Computadores, Natal – RN, p. 747-760, Maio 2003.

[Feen 01] Feeney, L.M., Nilsson, M., "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment", IEEE INFOCOM, Anchorage – AK, EUA, p. 1548-1557, Abril 2001.

[John 96] Johnson,B., Maltz, D.A., "Dynamic Source Routing in Ad Hoc Wireless Networking", Mobile Computing, T. Imielinski e H. Korth (eds.), Kluwer, p. 153-181, 1996.

[Jone 01] Jones, C.E., Sivalingam, K.M., Agrawal, P., Chen, J.C., “A Survey of Energy Efficient Network Protocols for Wireless Networks”, Wireless Networks Journal, vol. 7, nº4, p. 343-358, Julho 2001.

[Jung 02] Jung, E.S., Vaidya, N.H., "A Power Control MAC Protocol for Ad Hoc Networks", ACM MOBICOM, p. 36-47, Atlanta, EUA, Setembro 2002.

[Kar 03] Kar, K., kodialam, M., Lakshman, T.V., Tassiulas, L., “Routing for Network Capacity Maximization in Energy-Constrained Ad Hoc Networks”, IEEE Infocom, San Francisco - California, EUA, p. 673-681, Abril 2003.

[NS 03] “The Network Simulator – ns 2”, http://www.isi.edu/nsnam/ns/.

[Park 99] Park, V., Corson, M., “Temporally-Ordered Routing Algorithm (TORA) Version 1 - Functional Specification”, Internet Draft, MANET Working Group, draft-ietf-manet-tora-spec-02.txt, Outubro 1999.

[Perk 94] Perkins, C.E., Bhagwat, P., “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, ACM Conf. on Communications Architecture, Protocols and Applications, Londres, UK, p. 234-244, Agosto/Setembro 1994.

[Perk 99] Perkins, C.E., Royer, E.M., "Ad-Hoc On Demand Distance Vector Routing," IEEE WMCSA'99, New Orleans – LA, EUA, p. 90-100, Fevereiro 1999.

[Sing 98] Singh, S., Raghavendra, C.S. “PAMAS - Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks”, ACM Computer Communications Review, vol. 28, nº 3, p. 5-26, Nova York – NY, EUA, Julho 1998.

[Silv 04a] Silva, R.I., Leite, J.C.B., Fernandez, M.P., “Extensão do Tempo de Vida de Redes Ad Hoc”, 22º Simpósio Brasileiro de Redes de Computadores (SBRC-2004), Gramado – RS, p. 337-350, Maio 2004.

[Silv 04b] Silva, R.I., Leite, J.C.B., Fernandez, M.P., "Economia de Energia em Redes Ad Hoc", a ser publicado, 15º Congresso Brasileiro de Automática (CBA-2004), Gramado – RS, Setembro 2004.

[Silv 04c] Silva, R.I., Leite, J.C.B., Fernandez, M.P., "A Heuristic Approach to Energy Saving in Ad Hoc Networks", a ser publicado, 11º International Conference on Telecommunications (ICT-2004), Fortaleza – CE, Agosto 2004.

[Stem 97] Stemm, M., Katz, R.H., "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices", IEEE Trans. on Communications, vol. E80-B, nº 8, p. 1125-1131, Agosto 1997.

[Woo 98] Woo, S.S., Raghavendra, C.S., "Power-Aware Routing in Mobile Ad Hoc Networks", Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Dallas – Texas, EUA, p. 181-190, Outubro 1998.

[Xu 00] Xu, Y., Heidemann, J., Estrin, D., "Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks", Relatório Técnico nº 527, USC/ISI, Los Angeles - EUA, 14 p., Outubro 2002.

[Xu 01] Xu,Y., Heidemann, J., Estrin, D., "Geography-Informed Energy Conservation for Ad Hoc Routing", ACM/IEEE International Conference on Mobile Computing and Networking, Roma, Itália, p.70-84, Julho 2001.

[Xu 03] Xu,Y. et alli, "Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks", http://www.isi.edu/~johnh/PAPERS/Xu03a.html, Janeiro 2003.

[Yu 02] Yu, W., Lee, J., "DSR-Based Energy-aware Routing Protocols in Ad Hoc Networks", International Conference on Wireless Networks (ICWN 02), Las Vegas - Nevada, EUA, Junho 2002.

[Zhen 03] Zheng, R., Kravets, R., "On-demand Power Management for Ad Hoc Networks**",** IEEE Infocom, San Francisco – Califórnia, EUA, vol. 1, p. 481-491, Abril 2003.