

Pedro Olmo Stancioli Vaz de Melo

# Teoria dos Jogos na Comunicação de Dados em Redes de Sensores Sem Fio

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Informática.

Belo Horizonte

Março 2007

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



FICHA CATALOGRÁFICA

Biblioteca da Pontifícia Universidade Católica de Minas Gerais

V393t

Vaz de Melo, Pedro Olmo Stancioli

Teoria dos jogos na comunicação de dados em redes de sensores sem fio. / Pedro Olmo Stancioli Vaz de Melo. – Belo Horizonte, 2007. VIII; 80f. : il.

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Raquel Aparecida de Freitas Mini  
Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-Graduação em Informática, Belo Horizonte. Bibliografia.

1. Sistemas de comunicação sem fio. 2. Arquitetura não-hierárquica (Rede de computador). 3. Teoria dos jogos. 4. Teoria da informação. I. Mini, Raquel Aparecida de Freitas. II. Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-Graduação em Informática. III. Título.

CDU: 681.3.092

Bibliotecária : Erica Fruk Guelfi – CRB 6/2068

## Resumo

Um importante desafio no projeto de redes de sensores sem fio é desenvolver técnicas que façam uso eficiente da sua energia e, assim, estendam o seu tempo de vida. Reduzir a comunicação entre nós é uma maneira de se atingir esse objetivo, uma vez que a comunicação é o principal responsável pelo consumo da energia da rede. Uma técnica interessante para evitar comunicações é construir um modelo que use a informação local do nó sensor nas decisões dos mesmos. Uma vez que a teoria dos jogos lida com a otimização de um resultado global a partir de decisões locais, ela se torna uma ferramenta apropriada para desenvolver esse modelo. Este trabalho propõe e avalia um modelo baseado na teoria dos jogos para construir um mapa de energia para ser usado por protocolos de roteamento em redes de sensores sem fio. O mapa de energia é a informação sobre a energia residual em cada parte da rede e é atualizado por pacotes enviados pelos nós sensores. Uma solução para construir um mapa de energia precisa enviar um número reduzido de pacotes, pois, caso contrário, a rede gastaria muita energia na comunicação desses pacotes e, provavelmente, a utilidade da informação contida no mapa de energia não compensaria o custo para construí-lo. O modelo proposto neste trabalho considera que uma atividade de comunicação de dados deve ser realizada apenas se esta for trazer benefícios para o nó sensor que a executa, ou seja, se for economizar a sua energia. O modelo faz um nó sensor enviar um pacote de atualização do mapa de energia somente se isso for retirá-lo do processo de roteamento, uma vez que o gasto de energia com o roteamento é maior que o gasto com o envio do pacote de atualização. Assim, um nó sensor que está economizando a sua energia também está contribuindo para estender o tempo de vida da rede. Resultados de simulação revelam que o uso do modelo baseado na teoria dos jogos reduz significativamente a quantidade de pacotes requeridos para construir o mapa de energia, economizando a energia da rede. Além disso, verifica-se que o mapa construído por esse modelo é capaz de fornecer informações suficientes para a geração de rotas eficientes em energia.

## **Abstract**

An important issue in the design of a wireless sensor network is to devise techniques to make efficient use of its energy, and thus, extend its lifetime. Reduction in the communication among nodes is one way to achieve this goal, once communication is the main responsible for network energy consumption. An interesting technique to avoid communication is to build a model that uses the sensor node local information in its decisions. Once game theory deals with the optimization of a global result from local decisions, it becomes an appropriate tool to develop this model. This work proposes and evaluates a game theoretic model to build an energy map to be used by om 17792(g)e193121244.6.8051(y)-315.079

# Agradecimentos

Ao assistir uma cerimônia de premiação, sempre tive preguiça daqueles que desdobram uma lista gigantesca de nomes para serem agradecidos. Hoje, depois de concluído o meu mestrado, penso diferente. Penso que, durante a realização de um trabalho longo e árduo, obtemos auxílios que são determinantes para a conclusão desse trabalho com sucesso. Penso assim porque, depois de dois anos de muito trabalho, vejo que não teria conseguido concluir o meu mestrado sem a ajuda dada por muitas pessoas. Ajudas essas que, cada qual da sua maneira, me empurraram para a conclusão dessa difícil etapa da minha vida e serão sempre lembradas com carinho por mim. Assim, desejo expressar os meus sinceros agradecimentos:

Ao Programa de Mestrado em Informática da PUC Minas pelo voto de confiança, tendo me aceitado como aluno e oferecendo toda a estrutura necessária para a melhor realização do meu trabalho.

À todos os professores, que sempre me ajudaram com as minhas dúvidas e dificuldades durante o curso.

À Giovanna, que é uma segunda mãe para todos os alunos do mestrado, sempre trabalhando com invejável competência.

Aos meus colegas do mestrado, que mostraram-se verdadeiros companheiros durante o curso. Um agradecimento especial à Anne José, Cristiano José, Carol José, José José, Leonardo José, Michelle José, Sandro José e Wanderley José, que de colegas tornaram-se amigos. Contem comigo sempre que precisarem!

Ao Max e ao Daniel Guidoni, que foram fundamentais para o meu sucesso. Sou eternamente grato ao esforço gratuito que vocês fizeram para me ajudar na minha pesquisa. Estou muito feliz de ser colega de vocês hoje!

À Raquel Mini, que foi um para mim mais que uma orientadora, foi um exemplo de vida. A sua competência e confiança no meu trabalho foram essenciais para que eu tenha a vida que tenho hoje.

Ao professor Antônio Loureiro pelos valiosos comentários feitos durante o mestrado e pelo voto de confiança dado para o ingresso no doutorado.

Aos integrantes do grupo de pesquisa em redes de sensores sem fio da PUC Minas, que tiveram paciência para assistir a várias apresentações minhas, sempre colaborando com importantes comentários.

Ao Rigel e aos colaboradores da empresa Arte Informática, que me deram todo o apoio necessário para ingressar no mestrado, mesmo que isso não fosse o melhor para eles. Vocês perderam um funcionário mas ganharam um amigo.

À minha namorada Lorena, que esteve comigo durante os momentos mais difíceis. O seu apoio é a minha sustentação para querer melhorar sempre! Muito obrigado por tudo que você representa para mim!

Finalmente, agradeço aos meus pais Paulo Roberto e Maria José, que me deram todo o suporte necessário para enfrentar as muitas dificuldades advindas nesses dois anos de mestrado. Sou muito agradecido à Deus por ter a família que tenho!



# Conteúdo

Lista de Figuras	vi
Lista de Tabelas	ix
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivo . . . . .	4
1.3 Organização do Trabalho . . . . .	5
<b>2 Trabalhos Relacionados</b>	<b>6</b>
2.1 Comunicação de Dados em RSSFs . . . . .	6
2.2 Mapa de Energia . . . . .	8
2.3 Teoria dos Jogos em RSSFs . . . . .	10
2.3.1 Introdução . . . . .	10
2.3.2 Teoria dos Jogos em Protocolos de Roteamento . . . . .	10
2.3.3 Teoria dos Jogos no Controle do Acesso ao Meio . . . . .	12
2.3.4 Teoria dos Jogos em Outras Aplicações . . . . .	13
2.3.5 Conclusões . . . . .	14
<b>3 Teoria dos Jogos</b>	<b>15</b>
3.1 Introdução . . . . .	15
3.2 Taxonomia . . . . .	18
3.3 Solução por Dominância Estrita Iterada . . . . .	19
3.4 Solução por <i>Backward Induction</i> . . . . .	21
3.5 Solução por Equilíbrio de Nash . . . . .	23
3.5.1 Equilíbrio de Nash e Estratégias Puras . . . . .	23

3.5.2	Equilíbrio de Nash e Estratégias Mistas . . . . .	23
3.5.3	Considerações Finais . . . . .	24
<b>4</b>	<b>Teoria dos Jogos na Construção do Mapa de Energia</b>	<b>26</b>
4.1	Modelo . . . . .	26
4.2	O Algoritmo Pred-GT . . . . .	28
4.2.1	Algoritmo Base . . . . .	28
4.2.2	Utilizando Informações do Mapa de Energia (Pred-GT- $\alpha$ ) . . . . .	29
4.2.3	Utilizando Informações dos Vizinhos (Pred-GT- $\beta$ ) . . . . .	30
4.2.4	Utilizando Níveis de Energia (Pred-GT- $\gamma$ ) . . . . .	31
4.3	Equilíbrio de Nash na Construção do Mapa de Energia . . . . .	33
4.4	Conclusões . . . . .	35
<b>5</b>	<b>Resultados de Simulação</b>	<b>36</b>
5.1	Cenário . . . . .	36
5.2	Topologia com Distribuição Homogênea de Energia . . . . .	39
5.3	Topologia com Distribuição Aleatória de Energia . . . . .	47
5.4	Topologia com uma Região de Baixa Energia . . . . .	54
5.5	Topologia que Forma uma Região de Baixa Energia . . . . .	63
5.6	Conclusões . . . . .	71
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>74</b>
6.1	Conclusões . . . . .	74
6.2	Trabalhos Futuros . . . . .	75
	<b>Bibliografia</b>	<b>76</b>

# Lista de Figuras

1.1	Uma rede de sensores sem fio que detecta incêndios [WCE04]. . . . .	2
2.1	Curvas de roteamento no mapa de energia. . . . .	7
2.2	Exemplos de mapa de energia de uma rede. . . . .	9
3.1	Matriz de <i>payoffs</i> do jogo “A Batalha dos Sexos”. . . . .	17
3.2	Jogo na forma estratégica. . . . .	20
3.3	Jogo na forma estratégica depois da eliminação de uma coluna. . . . .	20
3.4	Jogo na forma estratégica depois da eliminação de duas linhas e duas colunas. . . . .	21
3.5	Solução de um jogo na forma extensiva por <i>backward induction</i> . . . . .	22
4.1	Comportamento do algoritmo de roteamento que usa as informações do mapa de energia. . . . .	28
5.1	Exemplos de topologias usadas nas simulações. . . . .	38
5.2	Topologia que forma uma região com baixa reserva de energia. . . . .	39
5.3	Curvas geradas para uma topologia com distribuição homogênea de energia. . . . .	40
5.4	Exemplo de um mapa de energia que representa uma topologia em que os nós sensores vizinhos ao nó monitor estão mortos. . . . .	41
5.5	Número de EIPs enviados para o nó monitor. . . . .	41
5.6	Energia média dos nós roteadores. . . . .	42
5.7	Tempo em equilíbrio de Nash. . . . .	44
5.8	Número de nós enviaram EIPs e continuaram a rotear. . . . .	45
5.9	<i>Payoff</i> médio. . . . .	47
5.10	<i>Payoff</i> cumulativo. . . . .	47

5.11	A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário com distribuição homogênea de energia. . . . .	48
5.12	Curvas geradas para uma topologia com uma distribuição aleatória das suas energias iniciais. . . . .	49
5.13	Número de EIPs enviados para o nó monitor. . . . .	50
5.14	Energia média dos nós roteadores. . . . .	51
5.15	Tempo em equilíbrio de Nash. . . . .	52
5.16	Número de nós enviaram EIPs e continuaram a rotear. . . . .	52
5.17	<i>Payoff</i> médio. . . . .	53
5.18	<i>Payoff</i> cumulativo. . . . .	53
5.19	A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário com distribuição aleatória de energia. . . . .	55
5.20	Curvas geradas para uma topologia com uma região de baixa energia. . . . .	56
5.21	Número de EIPs enviados para o nó monitor. . . . .	57
5.22	Energia média dos nós roteadores. . . . .	57
5.23	Número de transmissões dentro da região de baixa energia. . . . .	58
5.24	Número de EIPs enviados dentro da região de baixa energia. . . . .	59
5.25	Porcentagem de nós sensores localizados na região de baixa energia que executaram ações de roteamento e de envio de EIPs. . . . .	60
5.26	Tempo em equilíbrio de Nash. . . . .	60
5.27	Número de nós enviaram EIPs e continuaram a rotear. . . . .	61
5.28	<i>Payoff</i> médio. . . . .	62
5.29	<i>Payoff</i> cumulativo. . . . .	62
5.30	A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário com uma região de baixa energia. . . . .	64
5.31	Curvas geradas numa topologia que forma uma região com baixa reserva de energia. . . . .	65
5.32	Número de EIPs enviados para o nó monitor. . . . .	66
5.33	Energia média dos nós roteadores. . . . .	66
5.34	Número de transmissões dentro da região de baixa energia. . . . .	67
5.35	Número de EIPs enviados dentro da região de baixa energia. . . . .	67

5.36 Porcentagem de nós sensores localizados na região de baixa energia que

# Lista de Tabelas

3.1	Taxonomia da teoria dos jogos. . . . .	20
4.1	Relação da energia real de um nó sensor que tem energia inicial de $40J$ com sua energia convertida para um sistema de 2 níveis de energia e 10 níveis de energia. . . . .	32
5.1	Valores padrão usados nas simulações. . . . .	37
5.2	Comparação entre os resultados obtidos na topologia com distribuição homogênea de energia e a topologia com distribuição aleatória de energia. . .	54
5.3	Comparação entre os resultados obtidos na topologia com distribuição aleatória de energia e a topologia com uma região de baixa energia. . . . .	63
5.4	Comparação entre os resultados obtidos na com uma região de baixa energia e a topologia que forma uma região de baixa energia. . . . .	69

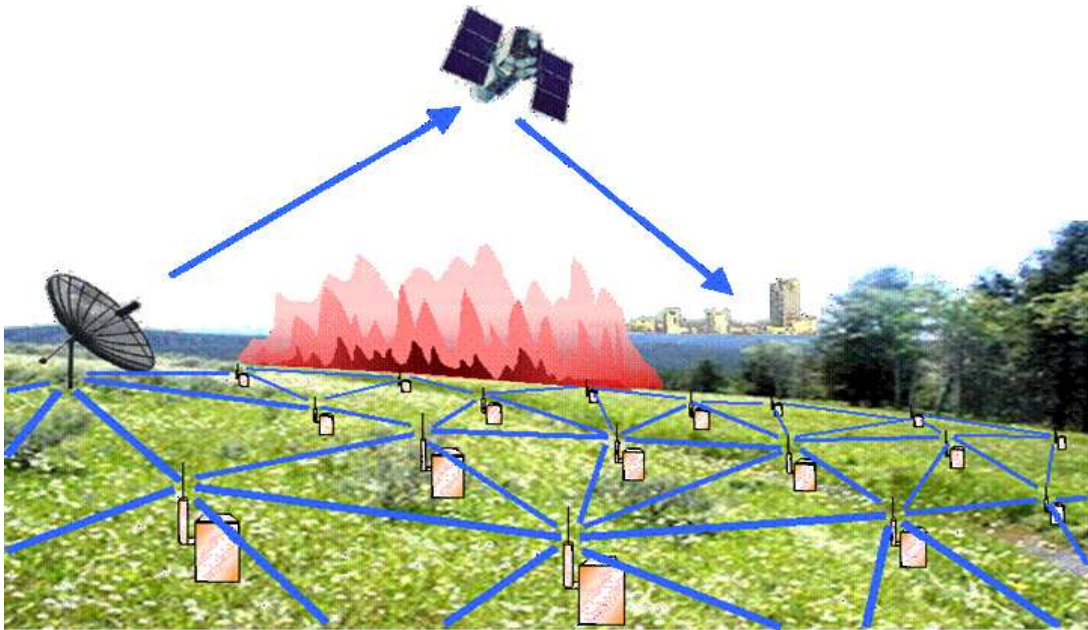
# Capítulo 1

## Introdução

### 1.1 Motivação

Redes de Sensores Sem Fio (RSSFs) [ADL<sup>+</sup>98, EGHK99, IEGH02, IGE00, KKP99, PK00a, PK00b] representam novos desafios de pesquisa relacionados ao projeto de algoritmos, protocolos de rede e *software* que possibilitarão o desenvolvimento de aplicações baseadas em sensores. As RSSFs são basicamente redes compostas de nós sensores, que são dispositivos computacionais de baixo custo e de baixa capacidade de processamento e de memória, e de nós monitores, que não possuem restrições de capacidade computacional. Os nós sensores são depositados em uma determinada região com o intuito de detectar variações das condições físicas da mesma, tais como temperatura, campo magnético ou umidade. Os nós monitores, por sua vez, são nós responsáveis por receber os dados coletados pelos nós sensores e disponibilizá-los para o usuário final. A figura 1.1 ilustra um exemplo de uma RSSF depositada para detectar incêndios, em que os dados coletados pelos nós sensores são enviados para o nó monitor, que é representado pela antena, aonde são processados e enviados para o usuário final.

As RSSFs podem ser disponibilizadas em regiões de difícil acesso ou até mesmo nocivas aos seres humanos, como vulcões e regiões contaminadas por radiação ou agentes químicos. Isso é possível porque a posição dos nós sensores não necessita ser previamente definida,



**Figura 1.1:** Uma rede de sensores sem fio que detecta incêndios [WCE04].

possíveis exemplos, uma aplicação pode ser desenvolvida para monitorar a presença de forças inimigas em campos de batalha, detectar focos de incêndio em alguma reserva florestal, notificar a perda de consciência de algum paciente ou até mesmo na construção de lares inteligentes que se adaptem às necessidades e costumes dos seus moradores [ASSC02].

Segundo [ASSC02], as RSSFs formam um novo tipo de rede *ad hoc* com um novo conjunto de características e desafios, tais como:

- Número consideravelmente elevado de nós sensores, dificultando a existência de um identificador global (ID) para cada nó.
- Nós sensores são depositados formando uma topologia densa.
- Nós sensores são passíveis de falhas, tais como término de energia e danos físicos, fazendo com que a topologia da rede varie constantemente.
- Uso do paradigma de comunicação por *broadcasting* é o principal, enquanto nas redes *ad-hoc* tradicionais, usa-se primariamente o paradigma de comunicação ponto a ponto.
- Nós sensores são limitados em energia, capacidade computacional e em memória.

Nós sensores possuem uma quantidade de energia limitada e, por estarem geralmente cercados por centenas de outros nós, aliado ao fato de estarem disponibilizados em áreas



de acesso restrito, não podem ter suas fontes de energia recarregadas ou substituídas. Os protocolos das RSSFs devem, então, priorizar a economia de energia em troca de fatores de *QoS*, tais como baixa latência e alto *throughput*, que são prioridades em redes tradicionais. Nesse contexto, a atividade de comunicação de dados é uma das maiores preocupações no projeto dessas redes porque o seu custo de energia é o mais significativo. Segundo [PK00a], o custo de energia da comunicação é aproximadamente três ordens de grandeza maior que o do processamento.

Uma vez que a comunicação apresenta o maior consumo de energia em RSSFs, trocas de mensagens devem ser evitadas sempre que possível. Isso pode ser feito a partir de um modelo que considera as informações locais dos nós sensores para determinar quais dados devem ser transmitidos. Como a teoria dos jogos [FT91] lida com a otimização de resultados globais a partir de tomadas de decisões locais, ela se torna uma abordagem favorável para a construção desse modelo. Na teoria dos jogos, os agentes são modelados como jogadores egoístas e racionais que tomam decisões visando exclusivamente o seu próprio benefício (*payoff*). Os benefícios são mapeados a partir de uma função de utilidade, que considera as estratégias escolhidas por todos os jogadores. Cada jogador possui um conjunto de estratégias que corresponde às ações que ele pode escolher para alcançar o seu objetivo.

Um dos problemas clássicos da teoria dos jogos é o “Dilema dos Prisioneiros” [FT91]. Nesse jogo, dois criminosos são presos e colocados em celas diferentes. A polícia oferece a eles a confissão do crime. Caso ambos confessem, eles pegarão cinco anos de prisão. Caso ambos não confessem, eles pegarão dois anos de prisão. Se um confessar e o outro não, aquele que confessou sairá livre e o que não confessou pegará dez anos de prisão. O interessante desse jogo é que o seu resultado natural é o pior possível. Ambos os jogadores tendem a escolher a estratégia de confessar e, conseqüentemente, estar na cadeia por cinco anos cada. A teoria dos jogos provê maneiras para entender esse jogo e o seu resultado natural e, a partir disso, alterar as suas características para os jogadores cooperarem entre si e aumentarem os seus *payoffs*. Se, por exemplo, o jogo se repetir infinitas vezes, os jogadores tenderão a aceitar a cooperação, escolhendo a estratégia de não confessar, uma vez que isso lhes dá melhores *payoffs*. Se um jogador optar pela estratégia de confessar, o outro alteraria também a sua estratégia na rodada seguinte e o resultado voltaria a ser o pior para ambos. Assim, nenhum jogador possui incentivos para quebrar o acordo de cooperação quando o jogo se repete infinitas vezes.

A utilização da teoria dos jogos nas RSSFs pode, por exemplo, definir um nó sensor como um jogador egoísta e racional de um jogo que paga um melhor *payoff* para os jogadores que economizarem mais energia. Assim, cada nó sensor sempre tomaria a decisão

que o leva ao menor gasto possível de energia. Se um pacote é recebido por esse jogador e pedido para ser encaminhado, é racional acreditar que o nó sensor descartará o pacote para economizar a sua energia, não contribuindo para o funcionamento global da rede. Contudo, como feito no Dilema dos Prisioneiros, a teoria dos jogos possui meios para forçar o jogador a cooperar e, nesse caso, encaminhar o pacote que ele recebeu. Além disso, o paradigma “*sensor-centric*” afirma que, para maximizar a utilização e a viabilização da informação em uma RSSF, sensores precisam cooperar para alcançar os objetivos globais da rede enquanto maximizam os seus tempos de vida individuais, uma vez que quanto mais um nó sobrevive, melhor é para a rede [KSI04]. Essas restrições, aliadas ao fato que nas RSSFs a comunicação deve ser evitada sempre que possível, fazem das mesmas um cenário favorável para ser estudado a partir dos conceitos da teoria dos jogos.

## 1.2 Objetivo

A economia de energia em RSSFs é um fator primordial para que elas funcionem de maneira eficiente. O maior consumidor de energia de uma RSSF é a comunicação. É desejável, então, que somente dados relevantes sejam trafegados pela rede. Dados com informações rotineiras e repetidas não devem ser enviados e encaminhados, uma vez que o custo no envio não é compensado pela utilidade que eles terão para o nó monitor. A partir de um modelo que seja capaz de julgar quais pacotes são relevantes e quais não são, a energia que usualmente é gasta para o envio e encaminhamento de informações não relevantes será economizada e a vida útil da rede será prolongada.

O objetivo deste trabalho é aplicar os conceitos da teoria dos jogos na tomada de decisão dos nós sensores de uma RSSF. As RSSFs serão modeladas como um jogo que segue os princípios ditados pela teoria dos jogos, na qual nós sensores, que serão os jogadores, alcançarão um maior benefício à medida que eles conseguirem economizar mais energia. Para tal, eles devem decidir localmente sobre quais ações eles devem realizar. Essa ação, que é a transmissão ou não de uma informação, deve ser realizada visando a economia de energia do nó sensor. Assim, à medida que ele economiza sua energia e consegue ainda contribuir para o funcionamento da rede, ele prolonga sua vida útil e, conseqüentemente, a da rede também.

Para avaliar a aplicabilidade dos conceitos da teoria dos jogos na solução de problemas das RSSFs, este trabalho propõe um modelo de construção e atualização do mapa de energia de uma rede de sensores sem fio. O mapa de energia corresponde à informação sobre a quantidade de energia disponível nas diversas partes da rede e ele é fundamental para estender o tempo de vida da mesma [MMLN05]. Esse mapa é construído no nó

monitor a partir de informações de energia que são enviadas pelos nós sensores. O mapa de energia construído será usado por algoritmos de geração de rotas para definir os nós sensores mais capazes de rotear pacotes. O objetivo deste trabalho é usar a teoria dos jogos para decidir quando realizar a comunicação de dados, que neste caso significa o envio de informações necessárias para construir o mapa de energia. A tomada de decisão é feita a partir de três algoritmos propostos: Pred-GT- $\alpha$ , Pred-GT- $\beta$  e Pred-GT- $\gamma$ . Nós sensores irão receber um melhor *payoff* quando eles economizarem energia. Quando um nó sensor decidir não enviar pacotes que ele julga desnecessários, a sua energia será economizada e o seu tempo de vida prolongado.

O presente trabalho apresenta resultados de simulação em que várias disseminações de dados são realizadas pelo nó monitor a partir de informações contidas no mapa de energia. O mapa de energia é construído a partir do algoritmo *Prediction-based* [MMLN05] e dos algoritmos baseados em teoria dos jogos propostos neste trabalho. A qualidade do mapa de energia é medida a partir do custo na sua construção e da qualidade das rotas geradas a partir dele. Resultados de simulação revelam que o uso da teoria dos jogos reduz o número de pacotes necessários para atualizar o mapa de energia mantendo a qualidade das rotas geradas. Isso mostra que a teoria dos jogos reduz o número de comunicações realizadas na rede, economizando sua energia e estendendo seu tempo de vida.

### 1.3 Organização do Trabalho

Este trabalho está organizado da seguinte maneira. O capítulo 2 apresenta os trabalhos relacionados. Os principais trabalhos apresentados são: comunicação de dados em RSSFs, mapa de energia e teoria dos jogos em RSSFs. O capítulo 3 apresenta os conceitos básicos da teoria dos jogos. O capítulo 4 apresenta o modelo proposto para construção do mapa de energia a partir dos conceitos da teoria dos jogos e os três algoritmos propostos para a tomada de decisão dos nós sensores. O capítulo 5 apresenta os resultados de simulação em que várias disseminações de dados são realizadas pelo nó monitor para toda a rede. Nas simulações, os algoritmos propostos são comparados com o algoritmo *Prediction-Based*. Finalmente, o capítulo 6 apresenta as conclusões e trabalhos futuros.

## Capítulo 2

# Trabalhos Relacionados

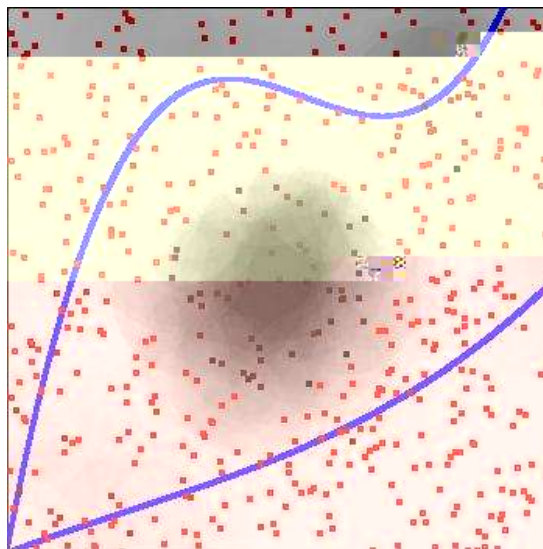
Este trabalho avalia o uso da teoria dos jogos na comunicação de dados em RSSFs e propõe, como estudo de caso, um algoritmo de construção e atualização do mapa de energia para ser usado no roteamento. É preciso então discutir as formas de comunicação de dados existentes nas RSSFs assim como os trabalhos existentes na literatura que usaram os conceitos da teoria dos jogos para solucionar problemas das RSSFs. Para o entendimento do estudo de caso, é necessário discorrer sobre a construção e a atualização do mapa de energia e como ele pode auxiliar no roteamento de dados. Assim, o presente capítulo discute a comunicação de dados em RSSFs, a construção do mapa de energia e problemas das RSSFs que foram resolvidos a partir dos conceitos da teoria dos jogos.

### 2.1 Comunicação de Dados em RSSFs

A comunicação de dados em RSSFs, sob o ponto de vista das entidades de comunicação, pode ser dividida em três casos [dVM05]: dos nós sensores para o nó monitor, entre os nós sensores e do nó monitor para os nós sensores. A comunicação de dados feita dos nós sensores para o nó monitor, também chamada de coleta de dados, ocorre, por exemplo, quando os nós sensores transmitem informações que eles possuem do ambiente para o nó monitor. Ela pode ocorrer também quando os nós sensores desejam transmitir alguma informação de controle para o nó monitor como, por exemplo, sua informação de energia para construção do mapa de energia. A comunicação entre os nós sensores ocorre quando eles cooperam entre si na realização de alguma tarefa específica. Os nós sensores podem, por exemplo, calcular a média da temperatura de determinada região antes de enviar os dados para o nó monitor. Isso pode economizar significativamente o número de comunicações na rede. Além disso, nós sensores podem comunicar entre si para trocar informações sobre a posição de determinado objeto em movimento, a fim de enviar para

o nó monitor uma informação mais precisa. Por fim, a comunicação feita do nó monitor para os nós sensores é chamada de disseminação de dados. O processo de disseminação de dados é de suma importância para o funcionamento correto e eficiente da rede. Ele ocorre, por exemplo, quando o nó monitor deseja alterar o modo de operação dos nós sensores ou quando deseja requisitar alguma nova informação do ambiente para todos ou para determinados nós sensores.

Dentre os algoritmos de disseminação de dados, o *Trajectory and Energy-based Data Dissemination* (TEDD) [dVMGM<sup>+</sup>05] é um exemplo de algoritmo que usa as informações contidas no mapa de energia. O TEDD usa o mapa de energia para gerar equações matemáticas de curvas para setores da topologia. As curvas representam as rotas pelas quais os pacotes de disseminação devem trafegar e cada equação de curva gerada define o roteamento em um setor da topologia. Assim, cada setor da topologia contém uma curva de roteamento gerada pelo TEDD. As curvas passam por regiões com alta reserva de energia e evitam regiões com baixa reserva de energia. A figura 2.1 ilustra um exemplo de curvas de roteamento geradas pelo TEDD em um mapa de energia. Neste exemplo, o mapa de energia é representado como uma imagem em tons de cinza em que as áreas claras representam regiões com mais energia enquanto as áreas escuras representam regiões com pouca energia.



**Figura 2.1:** Curvas de roteamento no mapa de energia.

As equações das curvas geradas pelo TEDD são armazenadas nos pacotes de disseminação que são propagados pelo nó monitor. Os nós sensores que recebem os pacotes de disseminação, por sua vez, decidem localmente se propagam ou não o pacote recebido. Para propagar o pacote, o nó sensor precisa estar dentro do setor da curva e a uma

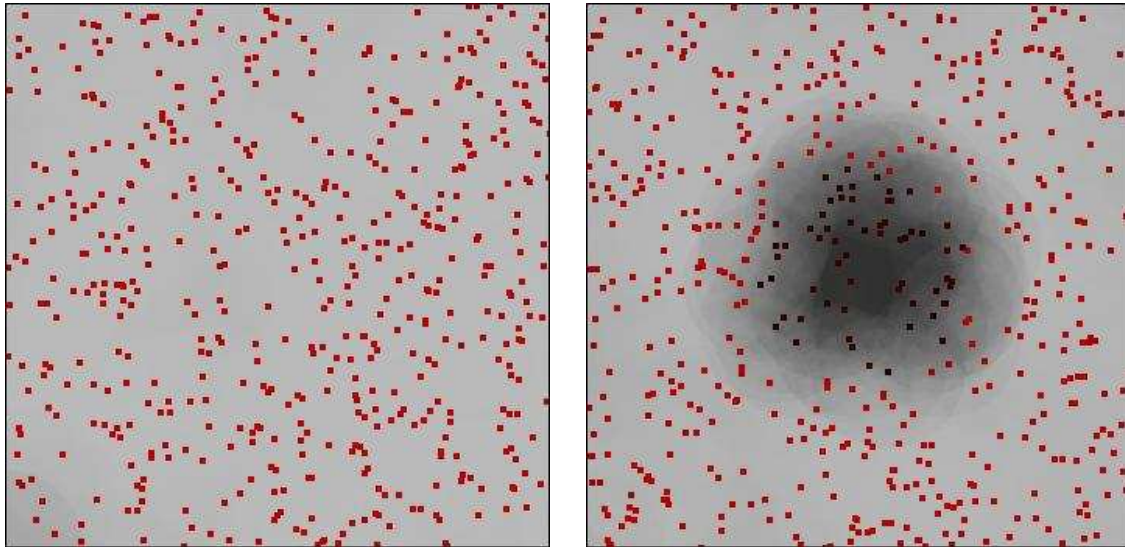
distância menor que o raio de comunicação de um ponto de referência definido pela curva. Se essas condições forem satisfeitas, o nó sensor calcula um tempo de espera para enviar o pacote recebido, que é proporcional à sua distância ao ponto de referência definido pela curva. Se, enquanto espera, o nó sensor não receber o pacote que está prestes a enviar, então ele envia o pacote. Caso contrário, ele o descarta.

Outro algoritmo de disseminação que usa o mapa de energia é o *Delay and Energy Based Broadcasting* (DEBB) [GdVMdFML06], que a partir dele calcula a energia de corte dos nós que não devem transmitir mensagens de difusão. É importante ressaltar que um algoritmo de roteamento que usa o mapa de energia necessita apenas da informação de quais nós podem ser usados para rotear e quais não podem. A informação exata da quantidade de energia de todos os nós, nesse caso, não se faz necessária. A próxima seção discute a construção e atualização do mapa de energia.

## 2.2 Mapa de Energia

A informação sobre a energia restante em cada parte da rede é chamada de *mapa de energia* e pode ajudar a prolongar o tempo de vida da rede. A figura 2.2 mostra exemplos de mapas de energia. A figura 2.2-a ilustra um mapa de energia com distribuição homogênea de energia e a figura 2.2-b ilustra um mapa de energia que possui uma região com baixa reserva de energia no centro da topologia. Pode-se representar o mapa de energia de uma RSSF como uma imagem em tons de cinza, em que as áreas claras representam regiões com mais energia, e regiões com pouca energia são representadas por áreas escuras. A partir do mapa de energia, um usuário deve ser capaz de determinar se alguma parte da rede está próxima de sofrer alguma falha de sistema em um futuro próximo devido ao esgotamento de sua energia [ZGE02]. O conhecimento de áreas com baixa reserva de energia pode ajudar na deposição incremental de sensores, uma vez que sensores adicionais podem ser colocados seletivamente nessas regiões escassas de recursos. A escolha da melhor localização para o nó monitor pode ser feita também baseada no mapa de energia. O nó monitor é um nó especial responsável por coletar informação dos nós sensores. Sabe-se que nós próximos ao nó monitor provavelmente vão gastar mais energia, uma vez que eles são usados frequentemente para entregar pacotes para o nó monitor. Conseqüentemente, se o nó monitor for movido para áreas com mais energia disponível, pode-se prolongar o tempo de vida da rede. Como mencionado na seção anterior, algoritmos de roteamento podem fazer um melhor uso das reservas de energia se eles escolherem rotas que passam por nós com mais energia residual. Outras possíveis aplicações que podem tirar vantagens do mapa de energia são algoritmos de reconfiguração, processamento de requisições e fusão

de dados. Na verdade, é difícil de imaginar uma aplicação e/ou um algoritmo que não precise usar o mapa de energia [Min04].



(a) Distribuição homogênea de energia.

(b) Uma região com baixa reserva de energia.

**Figura 2.2:** Exemplos de mapa de energia de uma rede.

O desafio de construir um mapa de energia é fazer com que a energia gasta para construí-lo seja menor que a energia economizada pelo seu uso. Uma solução ingênua para construir este mapa é programar cada nó sensor para enviar periodicamente a sua quantidade de energia para o nó monitor. Como uma rede de sensores frequentemente possui muitos nós com limitados recursos, a quantidade de energia gasta por esse algoritmo é proibitiva. Por essa razão, técnicas mais eficientes em energia precisam ser desenvolvidas a fim de coletar a informação da energia disponível em cada parte da rede.

Em [MMLN05], propõe-se um algoritmo baseado em predição para construir o mapa de energia para RSSFs. O algoritmo usa um modelo probabilístico baseado em cadeias de Markov. O nó monitor recebe dos nós sensores as suas quantidades de energia atuais e os parâmetros de um modelo que descreve como o nó, provavelmente, consumirá a sua energia no futuro. Dessa maneira, o nó monitor é capaz de prever a quantidade de energia disponível em cada nó sensor. Além disso, cada nó sensor calcula a quantidade de energia prevista pelo nó monitor e, quando a diferença entre a energia prevista e a sua energia real for maior que um limiar pré-definido (*threshold*), o nó sensor envia um pacote para o nó monitor com informações sobre a sua energia atual e os parâmetros do modelo do seu consumo de energia. Conforme apresentado em [MMLN05], esse modelo é eficiente em energia.

Na próxima seção serão mostrados problemas das RSSFs que foram resolvidos a partir

de abordagens que usam os conceitos da teoria dos jogos. Na seção 2.3.2 são mostradas soluções para o roteamento em RSSFs a partir de conceitos da teoria dos jogos. Na seção 2.3.3 é mostrada uma solução para o controle de acesso ao meio a partir de conceitos da teoria dos jogos. Na seção 2.3.4 são mostrados modelos baseados na teoria dos jogos para solução de problemas na segurança em RSSFs e quando duas RSSFs de proprietários diferentes forem depositadas na mesma região geográfica. Por fim, a seção 2.3.5 apresenta conclusões sobre o uso dos conceitos da teoria dos jogos na solução de problemas de RSSFs.

## 2.3 Teoria dos Jogos em RSSFs

### 2.3.1 Introdução

A economia de energia por parte dos nós sensores é um fator primordial para o funcionamento eficiente das RSSFs. Assim, uma vez que a comunicação de dados é a maior responsável pelo consumo de energia nas RSSFs, a troca de mensagem entre nós sensores deve ser evitada sempre que possível. Isso torna ainda mais latente o caráter distribuído das RSSFs, tornando a tomada de decisão local uma escolha preferencial sempre que esta for possível. Uma tomada de decisão, quando realizada a partir de informações armazenadas localmente, evita comunicações de dados na rede e economiza a energia da mesma. Como a teoria dos jogos lida com otimização de resultados globais a partir de tomadas de decisões locais, ela se torna uma abordagem favorável na solução de problemas das RSSFs.

Na teoria dos jogos, os agentes são caracterizados por um comportamento egoísta e racional, com o objetivo de maximizar seus respectivos *payoffs* independentemente das conseqüências que suas ações irão trazer para os outros. Usando mecanismos fornecidos pela teoria dos jogos, é possível alterar o resultado da interação desses agentes a fim de beneficiar o sistema como um todo. Na literatura, várias modelagens de problemas de RSSFs a partir de uma abordagem que usa teoria dos jogos foram propostas. A seguir, será discutido como a teoria dos jogos tem sido usada na solução de alguns problemas presentes em RSSFs.

### 2.3.2 Teoria dos Jogos em Protocolos de Roteamento

Um importante desafio no processo de roteamento de pacotes em RSSFs usando uma abordagem baseada em teoria dos jogos é fazer com que os nós sensores encaminhem pacotes de outros nós, considerando que eles são modelados como jogadores egoístas e racionais. Se o único benefício do nó sensor em um jogo que envolve rotear é a economia



de energia, então nenhum nó terá incentivo para encaminhar pacotes e a rede não irá funcionar. Por essa razão é necessário que o modelo contenha um sistema que estimule os nós sensores a encaminharem pacotes, mantendo o funcionamento da rede.

Em [BB05], foi proposto um modelo baseado em um mecanismo de punição para nós sensores que não desejam cooperar e encaminhar pacotes de outros nós. O modelo considera que o *payoff* do nó sensor, além de ser em função da economia de energia, é também em função do sucesso que ele tem no envio do seu pacote para o nó monitor. O mecanismo de punição considera que, se um nó não cooperar deliberadamente, com a única intenção de preservar a sua energia, ele é punido. Isso é feito com a retirada do nó sensor que não cooperou do jogo por um número determinado de intervalos, de modo que nenhum pacote gerado por esse nó seja encaminhado por outros nós. Foi verificado que esse mecanismo de punição não se revela totalmente eficiente, uma vez que a cooperação ainda não se mostra como a estratégia preferencial dos nós sensores. Para solucionar esse problema, outra regra foi adicionada ao mecanismo de punição: um coeficiente de desconto permanente que pune o *payoff* do jogador todas as vezes que ele é excluído do jogo. Com a adição desse coeficiente, nenhum nó é estimulado a optar pela estratégia de não cooperar, encaminhando assim os pacotes que lhes são recebidos.

Outra estratégia para tratar roteamento a partir de uma abordagem baseada em teoria dos jogos é descrita em [ZHL05]. A fim de impedir que os nós sensores não encaminhem pacotes, conceitos de jogos repetidos são introduzidos, permitindo que as estratégias sejam definidas a partir das jogadas anteriores. Isso torna possível a aplicação de efeitos de reputação e reembolso [FT91], que contribuem para cooperação do nó sensor. Nessa abordagem, o *payoff* do nó sensor, além de ser em função da economia de energia, também é em função do sucesso que o mesmo tem no envio do seu pacote para o nó monitor.

No modelo baseado em jogos repetidos, nós sensores alteram suas estratégias, que são baseadas em probabilidades de encaminhamento de pacotes, a cada rodada, procurando sempre otimizar seus *payoffs*. A cooperação (probabilidade de encaminhar pacotes é maior que zero) é a estratégia inicial do nó e, no caso de algum nó optar pela não cooperação, isso vai ser difundido e todos os nós que cooperaram anteriormente e detectaram isso irão também optar pela não cooperação. O período de não cooperação irá durar por um determinado número de rodadas, até que o benefício inicial do nó que não cooperou seja compensado. Apesar dessa solução [ZHL05] ser similar ao mecanismo de punição descrito em [BB05], o nó que não cooperou causa a parada temporária do funcionamento da rede. Dada essa razão, essa estratégia estimula a cooperação, uma vez que o funcionamento da rede provê um maior *payoff* comparado ao cenário que a rede não funciona.

Ambos os modelos descritos acima lidam com o problema de encaminhamento de

pacotes em RSSFs, no qual um nó sensor precisa decidir localmente se envia ou não um pacote para um nó vizinho. Quando o roteamento é feito a partir de uma árvore de coleta de dados, um importante desafio é a própria construção da árvore. Cada nó sensor precisa decidir localmente qual nó será seu pai e quais serão seus filhos na árvore. Critérios diferentes podem ser usados, tais como confiabilidade, energia e latência.

Em [NSK05], o critério escolhido para geração da árvore de coleta de dados foi a largura de banda que um pai oferece para seus dependentes na árvore. Basicamente, o algoritmo é iniciado com uma inundação no nó monitor. Cada nó sensor é marcado com um nível, definido pela sua distância ao nó monitor. Em seguida, uma árvore balanceada é construída escolhendo um nó pai de nível  $i$  pelos seus filhos de nível  $i+1$ . Os filhos escolhem um pai de acordo com a largura de banda disponível, que depende do número possível de filhos. A largura oferecida pelos pais varia enquanto os filhos vão sendo escolhidos. Um nó filho pode se desconectar de um nó pai caso ele considere que a associação a outro nó pai seja mais benéfica para ele. O processo termina quando nenhum nó filho tem a intenção de conectar-se a outro nó pai. É interessante pontuar que o critério principal dos jogadores desse modelo não é a economia de energia. Apesar disso, devido à construção da árvore balanceada, essa solução gera economia de energia em uma RSSF.

Semelhante ao problema da montagem de uma árvore de coleta de dados, o desafio da construção de rotas em uma RSSF é fazer com que um nó sensor escolha localmente um vizinho que julgue apto a fazer parte da rota. Em [SKI02], os autores propõem um modelo que otimiza a confiabilidade do tráfego de informações em uma RSSF minimizando o custo dessa comunicação, ou seja, economizando a energia da rede e maximizando seu tempo de vida. A modelagem de construção de rotas em uma RSSF é feita de maneira que os nós escolham um vizinho que julgue mais confiável para fazer parte da rota, ou seja, aquele vizinho que o nó julgue que tem menos chances de falhar durante a transmissão de dados.

### 2.3.3 Teoria dos Jogos no Controle do Acesso ao Meio

Além do roteamento, o problema do controle de acesso ao meio também foi modelado a partir da teoria dos jogos. Em [MW01] e [MW03], foi elaborada uma proposta para modificar o comportamento dos terminais no protocolo *slotted Aloha*. A modelagem define que os jogadores são os nós sensores que disputam um mesmo canal de comunicação e suas estratégias sejam escolhas de probabilidades de transmissão no canal.

Para que os jogadores atinjam o equilíbrio e consigam acessar o canal de maneira estável, foi proposto que as probabilidades de transmissão variem unicamente com o número de jogadores, sendo assim estratégias Markovianas. Então, é de competência dos

jogadores escolher uma probabilidade que seja ótima para  $n$  jogadores. Uma estratégia em equilíbrio deve ser um vetor de probabilidades de transmissão  $p_n = (p_1, p_2, p_3, \dots)$  tal que  $p_n$  maximize o *payoff* de cada jogador quando for dado que existem  $n$  jogadores e que cada novo jogador que entrar no jogo usará também  $p$  para determinar a sua probabilidade de transmissão.

Esse modelo é difícil de se aplicar, uma vez que os nós precisam saber o valor de  $n$ . Em uma situação real, isso é inviável e os autores sugerem que esse número seja obtido através de um cálculo de previsão por Markov, por exemplo. É afirmado pelos autores que a modelagem é eficiente apenas quando o valor de  $n$  puder ser estimado de forma precisa.

### 2.3.4 Teoria dos Jogos em Outras Aplicações

Além de modelar os problemas das RSSFs tendo os nós sensores como jogadores, pode-se modelá-los de outras maneiras. Em [ADBA04], é proposto um *framework* de segurança para RSSF. O *framework* proposto é baseado em um IDS (*Intrusion Detection System*), que é basicamente um sistema que seleciona *clusters* da rede para defender. Como os recursos de uma RSSF são escassos, o IDS não deve escolher mais de um *cluster* para defender. Devido a isso, ele deve escolher o *cluster* mais suscetível à ataques. Se o *cluster* atacado for diferente daquele que o IDS escolheu para defender, então o ataque obteve sucesso.

O modelo proposto é um jogo de dois jogadores: o IDS, que visa proteger a rede, e o atacante, que visa invadir a rede. A estratégia do IDS é basicamente escolher um *cluster* para defender e as estratégias do atacante são duas: escolher um *cluster* para atacar ou aguardar. O jogo é modelado construindo uma matriz de *payoffs* para cada jogador, considerando: (i) o valor da sessão atacada/protegida na RSSF, (ii) o valor de se perder um determinado *cluster*, (iii) o custo para se defender um determinado *cluster* e (iv) o número de nós em um determinado *cluster*.

Em [FBH05], foi modelado um jogo para lidar com a possível cooperação de diferentes RSSFs em uma mesma região. A idéia principal é fazer com que uma rede utilize os nós sensores da outra para encaminhar seus pacotes até um nó monitor, economizando assim a energia da primeira. O problema é que a rede que recebeu os pacotes pode simplesmente descartá-los, fazendo com que eles não cheguem ao nó monitor destino. Os jogadores são modelados como os responsáveis pelo comportamento da rede, podendo ser os nós monitores, por exemplo. A estratégia dos jogadores é definir o comportamento dos nós sensores, que pode ser cooperar e encaminhar sempre os pacotes de outras redes ou não

cooperar, não encaminhando nunca pacotes de outras redes.

Em simulações realizadas por esse modelo, dois equilíbrios foram encontrados. O primeiro deles, denominado equilíbrio cooperativo, acontece quando todas as redes cooperam. O segundo, denominado equilíbrio não cooperativo, acontece quando nenhuma das redes decide cooperar com as demais. O equilíbrio cooperativo, ao contrário do não cooperativo, trás benefícios para as redes, fazendo com que elas economizem energia e prolonguem seus respectivos tempos de vida.

### 2.3.5 Conclusões

Como visto nesta seção, existem diversas possibilidades de aplicar teoria dos jogos na solução de problemas de RSSFs. O desafio na modelagem desses problemas é a definição dos jogadores, as suas estratégias e funções que mapeiem os seus *payoffs*. No próximo capítulo, serão apresentados os conceitos mais importantes da teoria dos jogos e maneiras para solucionar um problema modelado a partir dos mesmos.

# Capítulo 3

## Teoria dos Jogos

### 3.1 Introdução

Teoria dos jogos é o estudo formal de conflitos e cooperação entre agentes quando a ação dos mesmos é independente [TvS01]. A força da teoria dos jogos é a metodologia que ela provê para se analisar e estruturar problemas de decisão estratégica. Os conceitos da teoria dos jogos são amplamente usados na análise de decisões tomadas por agentes como empresas disputando uma fatia do mercado ou nações em guerra. Na televisão, programas se utilizam de jogos clássicos da teoria dos jogos nas suas composições como, por exemplo, o Programa Sete e Meio do SBT. Nas últimas décadas a teoria dos jogos passou a ser utilizada também para estudar o comportamento de agentes computacionais, ou seja, seres que não podem agir de maneira diferente da qual foram programados para agir.

Quando utilizada para estudar o comportamento de agentes computacionais, a teoria dos jogos lida primariamente com problemas relacionados à otimização distribuída, ou seja, otimização do resultado da interação de agentes que tomam decisões independentemente de algum controle central. Esse problema é a generalização de diversos problemas encontrados nas redes de computadores e, dentre eles, o controle de tráfego [KP99, YMR00, AB02], o balanceamento de carga [STZ04], o roteamento [LA00, CDR03, RT02], o *peering* [CP04] e o projeto de sistemas P2P [BAS03] possuem na literatura soluções modeladas a partir dos conceitos da teoria dos jogos.

A modelagem de um problema a partir da teoria dos jogos é feita pela definição de um jogo, dos participantes desse jogo e das ações permitidas, preferências e reações de cada jogador. Um dos jogos mais clássicos da teoria dos jogos é “A Batalha dos Sexos”. Esse jogo possui dois jogadores, um homem e uma mulher. O homem e a mulher desejam sair juntos, mas desejam ir a lugares distintos. O homem prefere ir ao jogo de futebol,

enquanto a mulher prefere ir ao cinema. Se eles forem juntos ao jogo de futebol, o homem terá satisfação maior que a mulher, enquanto que, se eles forem juntos ao cinema, a mulher terá satisfação maior que o homem. Caso cada um vá sozinho a seu programa predileto, ambos ficarão insatisfeitos. A ação de cada jogador é justamente escolher a qual lugar ele irá. Sua satisfação vai depender da ação escolhida por ambos.

A medição da satisfação de um jogador é feita numericamente através de um número real que é gerado a partir de uma função que leva em conta a ação de cada outro jogador participante do jogo. Esse valor numérico que cada jogador recebe para determinar sua satisfação é chamado de *payoff* e a função que gera esse número é chamada de *função utilidade*. O conjunto de *payoffs* dos jogadores é chamado de *outcome*.

Em termos matemáticos, pode-se definir um jogo por um conjunto finito de jogadores, representado por  $G = g_1, g_2, \dots, g_n$ . Cada jogador  $g_i \in G$  possui um conjunto finito  $S_i = s_{i1}, s_{i2}, \dots, s_{im_i}$  de ações, denominadas *estratégias puras* do jogador  $g_i$  ( $m_i \geq 2$ ). A notação  $s_{-i}$  é usada para representar todas as estratégias puras dos jogadores, exceto do jogador  $g_i$ . Um vetor  $s = (s_{1j}, s_{2j}, \dots, s_{nj})$ , onde  $s_{ij}$  é uma estratégia pura para o jogador  $g_i \in G$ , é denominado um *perfil de estratégia pura*. O conjunto de todos os perfis de estratégia pura formam o produto cartesiano

$$\prod_{i=1}^n S_i = S_1 \times S_2 \times \dots \times S_n,$$

denominado *espaço de estratégia pura* do jogo. Para cada jogador  $g_i \in G$ , existe uma função utilidade

$$u_i : S \rightarrow \mathfrak{R}$$

$$s \rightarrow u_i(s)$$

que associa o ganho (*payoff*)  $u_i(s)$  do jogador  $g_i$  a cada perfil de estratégia pura  $s \in S$ .

A partir dessas definições pode-se modelar o jogo “A Batalha dos Sexos” descrito anteriormente da seguinte maneira:

$$G = \{homem, mulher\},$$

$$S_{homem} = \{futebol, cinema\},$$

$$S_{mulher} = \{futebol, cinema\},$$

$$S = \{(futebol, futebol), (futebol, cinema), \\ (cinema, futebol), (cinema, cinema)\}$$

As duas funções utilidade  $u_{homem} : S \rightarrow \mathfrak{R}$  e  $u_{mulher} : S \rightarrow \mathfrak{R}$  devem retornar valores reais para cada jogador e, nesse caso, pode-se definir valores arbitrários para elas, de forma que

$$u_{mulher}(futebol, futebol) = 5;$$

$$u_{mulher}(futebol, cinema) = 0;$$

$$u_{mulher}(cinema, futebol) = 0;$$

$$u_{mulher}(cinema, cinema) = 10;$$

$$u_{homem}(futebol, futebol) = 10;$$

$$u_{homem}(futebol, cinema) = 0;$$

$$u_{homem}(cinema, futebol) = 0;$$

$$u_{homem}(cinema, cinema) = 5;$$

Os resultados descritos acima podem ser representados por uma matriz denominada *matriz de payoffs*, que pode ser vista na figura 3.1. Cada quadrante da matriz contém um par de valores que representa um *outcome* possível do jogo. O primeiro valor é o *payoff* dado ao jogador situado ao lado da tabela, que no caso é o homem. O segundo valor é o *payoff* dado ao jogador situado no topo da tabela, que no caso é a mulher. As linhas e as colunas

será mostrada a solução por dominância estrita iterada e, na seção 3.4, será mostrada a solução por *backward induction*. Na seção 3.5 será mostrado o conceito de equilíbrio de Nash e as soluções que podem ser alcançadas por ele.

## 3.2 Taxonomia

A teoria dos jogos é uma teoria matemática criada para se modelar qualquer fenômeno que pode ser observado quando dois ou mais agentes capazes de tomar decisões interagem entre si [BS04]. Como a quantidade e diversidade de fenômenos que possuem essa característica é bastante ampla, as características dos jogos que são modelados também são bem variadas e decisivas para se encontrar a solução do jogo.

Um aspecto crucial na especificação de um jogo envolve a informação que um jogador dispõe quando vai realizar uma ação. Os jogos mais simples, em termos de estrutura lógica, são aqueles em que os agentes possuem a informação completa do que aconteceu no jogo até o momento da sua jogada. Esses jogos são chamados de *jogos de informação perfeita*. Exemplos de jogos de informação perfeita podem ser encontrados em jogos de tabuleiros, como xadrez e damas. Por outro lado, jogos em que os agentes não possuem toda informação sobre as jogadas dos outros jogadores e/ou resultados das mesmas são chamados de *jogos de informação imperfeita*. Jogos de cartas geralmente são jogos de informação imperfeita, como o poker e o buraco.

Outro aspecto importante na especificação de um jogo reside no fato dos jogadores poderem ou não se comunicar antes de realizar jogadas. Jogos nos quais os jogadores podem se comunicar, fazendo acordos de jogadas para que elas produzam um bom resultado para ambos, são chamados de *jogos cooperativos*. O jogo “A Batalha dos Sexos” pode se tornar um jogo cooperativo caso o homem se comunique com a mulher e combinem, por exemplo, de cada semana irem juntos a um dos programas. Jogos que esses acordos não são permitidos são chamados de *jogos não-cooperativos*.

Nos jogos cooperativos, a cooperação é estabelecida porque há um consenso entre os jogadores que essa é a melhor estratégia para todos a longo prazo. Se um jogador quebrar o acordo de cooperação, ele vai receber um melhor *payoff* na rodada corrente do jogo mas vai receber piores *payoffs* nas rodadas seguintes. Assim, todos os jogadores são incentivados a manter a cooperação. Isso acontece porque nenhum jogador sabe quando o jogo se encerrará. Quando o fim do jogo é de conhecimento deles, eles possuem incentivos para quebrar o acordo de cooperação na última rodada do jogo, pois não precisam mais pensar a longo prazo. O problema é que todos sabem que isso vai ocorrer e assim também quebram o acordo na penúltima rodada do jogo, e por saberem disso, também quebram na ante-



penúltima rodada e assim por diante, não acontecendo cooperação em momento algum. Dessa forma, o conhecimento do fim do jogo por parte dos jogadores é determinante no estabelecimento da cooperação entre os jogadores. Jogos que não há conhecimento do fim do jogo são chamados de *jogos infinitos* e jogos que o fim do jogo é sabido são chamados de *jogos finitos*.

Na análise dos resultados, pode-se classificar os jogos como *jogos de soma constante* e *jogos de soma não-constante*. Jogos de soma constante são jogos em que a soma dos *payoffs* dos jogadores é sempre constante, ou seja, jogadores não ganham sem que outros percam. Jogos em que a constante é zero são chamados de *jogos de soma zero*. O poker novamente pode servir de exemplo para esse jogo. Por outro lado, jogos de soma não-constante não fazem nenhuma suposição sobre o valor gerado pela soma dos *payoffs* dos jogadores, que pode variar estocasticamente. O jogo “A Batalha dos Sexos” é um jogo de soma não-constante.

Um jogo também pode ser explicado e modelado segundo a ordem das suas jogadas. Quando as jogadas são simultâneas, ou quando a ordem das jogadas é irrelevante para o resultado final, o jogo é um *jogo na forma normal*. Quando a ordem das jogadas influencia o *payoff* obtido pelos jogadores, então o jogo é um *jogo na forma extensiva*. O jogo na forma normal é normalmente representado por uma matriz de *payoffs* e o jogo na forma extensiva por uma árvore. Existem entretanto jogos que são jogados tanto na forma normal quanto na forma extensiva como, por exemplo, um jogo que analisa a disputa entre duas empresas no mercado.

Por fim, a modelagem e estratégias de solução para o jogo variam de acordo com o número de jogadores participantes. Os jogos mais simples de serem resolvidos são os *jogos de dois jogadores* que, como o próprio nome diz, possuem apenas dois jogadores. A “A Batalha dos Sexos” e o xadrez são jogos de dois jogadores. Jogos que possuem mais de dois jogadores são chamados de *jogos de n-jogadores* e têm como exemplo o poker. A tabela 3.1 apresenta um resumo da taxonomia da teoria dos jogos.

### 3.3 Solução por Dominância Estrita Iterada

A solução por *dominância estrita iterada* é aplicada quando existe uma estratégia que, quando escolhida, resulta *payoffs* inferiores a uma outra estratégia para todas as combinações de estratégias dos seus adversários. Pode-se dizer então que uma estratégia pura  $s_{ik} \in S_i$  do jogador  $g_i \in G$  é *estritamente dominada* pela estratégia  $s_{ik'} \in S_i$  se

$$u_i(s_{ik'}, s_{-i}) > u_i(s_{ik}, s_{-i}),$$

Característica	Opções
Informação disponível	Informação perfeita Informação imperfeita
Cooperação	Jogos cooperativos Jogos não-cooperativos
Número de rodadas	Jogos infinitos Jogos finitos
Distribuição dos <i>payoffs</i>	Soma constante Soma não-constante
Ordem das jogadas	Forma normal Forma extensiva
Número de jogadores	2-jogadores n-jogadores

**Tabela 3.1:** Taxonomia da teoria dos jogos.

para todo  $s_{-i} \in S_{-i}$ . A estratégia  $s_{ik} \in S_i$  é fracamente dominada pela estratégia  $s_{ik'} \in S_i$  se  $u_i(s_{ik'}, s_{-i}) \geq u_i(s_{ik}, s_{-i})$ , para todo  $s_{-i} \in S_{-i}$ .

*Dominância Estrita Iterada* nada mais é que um processo onde se eliminam as estratégias estritamente dominadas.

		$p_2$			
		$s_{21}$	$s_{22}$	$s_{23}$	$s_{24}$
	$s_{11}$	5, 2	2, 6	1, 4	0, 4
	$s_{12}$	0, 0	3, 2	2, 1	1, 1
$p_1$	$s_{13}$	7, 0	2, 2	1, 1	5, 1
	$s_{14}$	9, 5	1, 3	0, 2	4, 8

**Figura 3.2:** Jogo na forma estratégica.

Considere o jogo da figura 3.2. Nesse jogo, a estratégia  $s_{21}$  do jogador  $p_2$  é estritamente dominada pela estratégia  $s_{24}$ . Devido a isso, a coluna que representa a estratégia  $s_{21}$  pode ser eliminada do jogo, uma vez que o jogador  $p_2$  sempre vai optar pela estratégia  $s_{24}$  em função da estratégia  $s_{21}$ . O jogo resultante dessa ação pode ser visto na figura 3.3.

		$p_2$		
		$s_{22}$	$s_{23}$	$s_{24}$
	$s_{11}$	2, 6	1, 4	0, 4
	$s_{12}$	3, 2	2, 1	1, 1
$p_1$	$s_{13}$	2, 2	1, 1	5, 1
	$s_{14}$	1, 3	0, 2	4, 8

**Figura 3.3:** Jogo na forma estratégica depois da eliminação de uma coluna.

O jogador  $p_1$ , por ser um agente racional, tem ciência da modificação da matriz de *payoffs*, pois sabe que o jogador  $p_2$  jamais escolheria a estratégia  $s_{21}$ . Devido a isso, a matriz de *payoffs* resultante gerou para ele duas estratégias estritamente dominadas:  $s_{11}$  e  $s_{14}$ , que são estritamente dominadas pelas estratégias  $s_{12}$  e  $s_{13}$ , respectivamente. Isso permite que as linhas da matriz que representam essas estratégias sejam também eliminadas. Feito isso, o jogador  $p_2$  verifica que pode eliminar da matriz a coluna que representa a estratégia  $s_{23}$ , que é estritamente dominada pela estratégia  $s_{22}$ . A matriz resultante pode ser vista na figura 3.4.

		$p_2$	
		$s_{22}$	$s_{24}$
$p_1$	$s_{12}$	3, 2	1, 1
	$s_{13}$	2, 2	5, 1

Figura 3.4: Jogo na forma estratégica depois da eliminação de duas linhas e duas colunas.

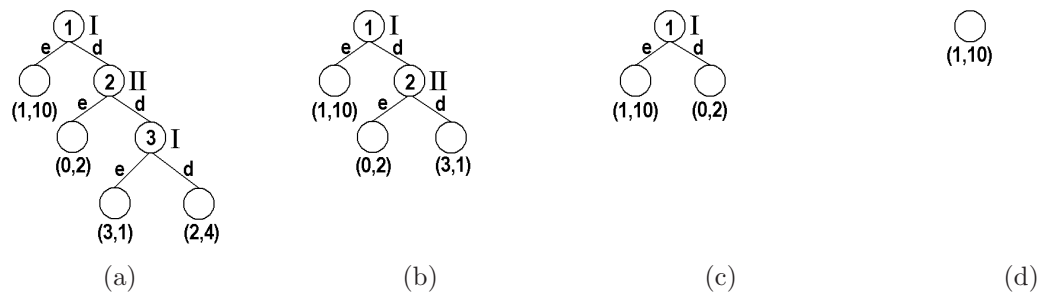
Finalmente, na matriz  $2 \times 2$  resultante, a estratégia  $s_{24}$  é estritamente dominada pela estratégia  $s_{22}$ , fazendo com que a coluna  $s_{24}$  seja excluída da matriz. Na matriz  $2 \times 1$  resultante, a estratégia  $s_{13}$  é estritamente dominada por  $s_{12}$ , eliminando sua linha correspondente da matriz. Depois desse processo de eliminação de células da matriz, o jogo teve como resultado o perfil de estratégia pura  $(s_{12}, s_{22})$ .

É importante ressaltar que essa técnica não gera necessariamente apenas um perfil como resultado. Existem jogos que vários perfis de estratégias são resultados da aplicação dessa técnica. Muitas vezes essa técnica é aplicada antes de outra, afim de diminuir o espaço de estratégias possíveis.

### 3.4 Solução por *Backward Induction*

A solução por *backward induction* deve ser levada em conta apenas em jogos finitos na forma extensiva. Esse tipo de jogo pode ser representado por uma árvore e através dela é possível visualizar todas as jogadas possíveis dos jogadores, além de todos os *outcomes*. A figura 3.5 contém um jogo na forma extensiva representado por uma árvore. Os jogadores  $I$  e  $II$  podem realizar as ações  $d$  e  $e$  alternadamente, sendo que cada nível da árvore representa o momento da ação de um dos jogadores. Os nós internos da árvore são pontos de decisão, também chamados de *sub-jogos*. As folhas representam os possíveis *outcomes*.

A técnica de *backward induction*, também comumente chamada de *algoritmo de Zermelo*, é iniciada nos sub-jogos mais profundos da árvore. No jogo da figura 3.5-a o algoritmo é iniciado no nó 3, no qual o jogador  $I$  pode escolher entre as estratégias  $e$ , que o



**Figura 3.5:** Solução de um jogo na forma extensiva por *backward induction*.

leva a um *payoff* de 3, e *d*, que o leva a um *payoff* de 2. Dadas as opções e o requisito de que um jogador sempre procura maximizar o seu *payoff*, a estratégia escolhida pelo jogador *I* será, inevitavelmente, *e*. Isso significa dizer que sempre que o jogador *I* chegar ao sub-jogo 3 ele irá optar pela estratégia *e*. O jogador *II*, por estar em um jogo de informação perfeita, tem ciência disso e sabe que, sempre que ele optar pela estratégia *d* no sub-jogo 2, ele receberá um *payoff* de 1, uma vez que o jogador *I* sempre irá usar a estratégia *e* no sub-jogo 3. Assim sendo, o jogo da figura 3.5-a, após essa etapa do algoritmo, é transformado no jogo da figura 3.5-b.

O algoritmo prossegue, então, analisando cada sub-jogo restante da árvore, sempre dos mais profundos até a raiz. No sub-jogo 2, o jogador *II* constata por sua vez que receberá um *payoff* de 1 se optar pela estratégia *d* e um *payoff* de 2 se optar por *e*. Isso faz com que o sub-jogo 2 seja eliminado da árvore e substituído pelo nó folha (0, 2), que é o *outcome* do jogo gerado pela escolha do jogador *II*, resultando no jogo da figura 3.5-c. O jogador *I* então tem a opção de, no sub-jogo 1, escolher entre a estratégia *d* e receber o *payoff* de 1 e a estratégia *e* e receber o *payoff* de 0. A solução do jogo é então o *outcome* (1, 10), como mostrado na figura 3.5-d.

Uma observação importante deve ser feita a respeito desse jogo. A solução do jogo é o *outcome* (1, 10), que é atingido sem que o jogador *II* realize qualquer jogada. Ambos os jogadores podem resolver o jogo por *backward induction*, mas o jogador *II* só deve resolver o jogo por *backward induction* se assumir que o jogador *I* é irracional, não optando pela estratégia *e* no sub-jogo 1. Esse fato é chamado de *paradoxo do backward induction*. O jogador *II* só irá realizar alguma jogada caso o jogador *I* faça uma jogada irracional. A possibilidade de um jogador realizar uma jogada irracional é chamada na literatura de *trembling hand*.

## 3.5 Solução por Equilíbrio de Nash

### 3.5.1 Equilíbrio de Nash e Estratégias Puras

Outra maneira para se chegar à solução de um jogo é usar técnicas para se alcançar um equilíbrio de Nash. Um equilíbrio de Nash é caracterizado quando nenhum jogador é capaz de aumentar seu *payoff* mudando de estratégia quando todos seus adversários mantiverem as suas [FT91]. Dessa maneira, um perfil de estratégia

$$s^* = (s_1^*, \dots, s_{i-1}^*, s_i^*, s_{i+1}^*, \dots, s_n^*) \in S$$

é um equilíbrio de Nash se

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_{ij_i}, s_{-i}^*)$$

para todo  $i = 1, \dots, n$  e para todo  $j_i = 1, \dots, m_i$ , com  $m_i \geq 2$ .

Quando os jogadores de um jogo atingem um equilíbrio de Nash, o jogo a que eles pertencem atingiu um estado completamente estável, ou seja, nenhum jogador irá alterar as suas estratégias. É importante ressaltar que o estado estável é assegurado levando em conta o caráter egoísta e racional dos jogadores. Existem várias maneiras para se calcular estratégias de jogadores que levarão a um equilíbrio de Nash e elas variam segundo as características do jogo, podendo ser encontradas em detalhes em [FT91]. É importante ressaltar também que existem jogos que não possuem equilíbrio de Nash para estratégias puras, como o jogo do par ou ímpar, e outros que possuem vários equilíbrios de Nash, como o jogo “A Batalha dos Sexos”, com os perfis  $(futebol, futebol)$  e  $(cinema, cinema)$  sendo soluções que estão em equilíbrio de Nash.

### 3.5.2 Equilíbrio de Nash e Estratégias Mistas

Como foi visto anteriormente, existem jogos, como o par ou ímpar, que não possuem um equilíbrio de Nash para estratégias puras. Uma alternativa para esses jogos possuírem um equilíbrio de Nash é considerar as estratégias de um ponto de vista probabilístico. No jogo do par ou ímpar, por exemplo, pode-se considerar que os jogadores tem 50% chance de escolherem *par* e 50% de chance de escolherem *ímpar*. Uma distribuição de probabilidades sobre o conjunto  $S_i$  de estratégias puras do jogador é chamada de *estratégia mista*  $p$

denominado espaço de estratégia mista. A notação  $p_{-i}$  é usada para representar todas as estratégias mistas dos jogadores, exceto do jogador  $g_i$ .

As distribuições de probabilidades dos jogadores para o jogo do par ou ímpar seriam  $p_1 = (\frac{1}{2}, \frac{1}{2})$  e  $p_2 = (\frac{1}{2}, \frac{1}{2})$  e o perfil de estratégia mista para  $p_1$  e  $p_2$  seria  $p = (p_1, p_2)$ , ou  $p = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ . Cada perfil de estratégia mista gera um *payoff*  $u_i$  para cada jogador  $g_i \in G$  com  $m_i$  estratégias puras, e pode ser definido como:

$$u_i(p) = \sum_{j=1}^m \dots \sum_{j_n=1}^{m_n} \left( \prod_{k=1}^n p_{kj_k} u_i(s_{1j_1}, \dots, s_{nj_n}) \right)$$

Um perfil de estratégia mista

$$p^* = (p_1^*, p_2^*, \dots, p_n^*) \in \Delta = \Delta_{m_1} \times \Delta_{m_2} \dots \times \Delta_{m_n}$$

é um equilíbrio de Nash se

$$u_i(p_i^*, p_{-i}^*) \geq u_i(p, p_{-i}^*)$$

para todo  $p \in \Delta_{m_i}$ , isto é, nenhum jogador sente motivação de trocar sua estratégia mista se os demais jogadores não o fizerem. Jogos não-cooperativos sempre possuem um equilíbrio de Nash para estratégia mistas [Nas50, Nas51].

### 3.5.3 Considerações Finais

O problema de se encontrar um Equilíbrio de Nash já foi descrito como o “problema mais fundamental da computação” na interface de ciência da computação e teoria dos jogos [Pap01]. Apesar de várias décadas de pesquisa, a resolução desse problema ainda é complicada, sua complexidade computacional precisa ainda é desconhecida e novos algoritmos são poucos e esparsos entre si [PNS04].

A grande dificuldade de formalizar a complexidade de se encontrar um Equilíbrio de Nash está na quantidade de características que um jogo pode ter. Essa quantidade de características, mencionadas na seção 3.2, são fundamentais para definir a complexidade de se achar um equilíbrio de Nash. Um jogo de soma constante e de dois jogadores, por exemplo, possui uma ordem de complexidade diferente de um jogo de soma não-constante e de  $n$ -jogadores. Outro fator que deve-se levar em conta é o que se deseja computar em um jogo. A ordem complexidade para se achar um equilíbrio de Nash qualquer é diferente da ordem de complexidade para se achar o equilíbrio de Nash que possui a maior soma dos *payoffs*. Um trabalho formal sobre a complexidade da solução por equilíbrio de Nash pode ser encontrado em [CS02].

Dentre os algoritmos para se achar um equilíbrio de Nash em um jogo, pode-se citar o algoritmo Lemke-Howson [LH64] para jogos de dois jogadores, o *Simplicial Subdivision* [adLTdH87] e o Govidan e Wilson [GW03] para jogos de  $n$ -jogadores. Esses algoritmos são exponenciais no pior caso. Uma breve descrição sobre os algoritmos para se encontrar um equilíbrio de Nash pode ser encontrada em [PNS04].

O próximo capítulo descreve um jogo de informação imperfeita, não cooperativo, infinito, de soma não-constante e de  $n$ -jogadores para construir um mapa de energia para o roteamento em RSSFs. A verificação do equilíbrio de Nash nesse jogo é possível e varia de acordo com o protocolo de roteamento que usa o mapa de energia. O capítulo 6 descreve um método para verificar o equilíbrio de Nash nesse jogo quando o protocolo de roteamento é o TEDD [dVMGM<sup>+</sup>05].

## Capítulo 4

# Teoria dos Jogos na Construção do Mapa de Energia

### 4.1 Modelo

Nesta seção será apresentado o modelo baseado nos conceitos da teoria dos jogos para construir o mapa de energia para protocolos de roteamento, que podem ser de coleta de dados ou de disseminação de dados. O mapa de energia é construído a partir de pacotes com a informação de energia que os nós sensores enviam para o nó monitor e é usado como parâmetro de entrada de algoritmos que geram rotas. As rotas geradas precisam ser eficientes em energia e devem usar apenas nós com alta reserva de energia no roteamento de dados.

Neste modelo, nós sensores são jogadores de um jogo e se comportam de maneira egoísta e racional, de forma que um nó gastará a sua energia apenas se o mesmo considerar que isso lhe trará benefícios. Se o consumo de energia de um nó for a única métrica para calcular o seu *payoff*, então ele gastará a sua energia somente se ele considerar que isso economizará a sua energia no futuro. Assim, as estratégias do nó sensor são (i) enviar a sua informação de energia, e (ii) não enviar a sua informação de energia. O pacote que contém a informação de energia do nó é chamado de Pacote de Informação de Energia (EIP). O jogo é dividido em rodadas e realizado até que o tempo de vida da RSSF expire. Uma rodada do jogo ocorre a cada fluxo de roteamento na rede, podendo esse ser um processo de disseminação de dados ou de coleta de dados. Esse é um jogo de informação imperfeita, não cooperativo, de soma não constante e de n-jogadores.

O cálculo do *payoff* do jogador considera as constantes  $cEIP$ , que é o custo incorrido ao nó sensor por enviar o seu EIP para o nó monitor, e  $cRouter$ , que é o custo incorrido



ao nó sensor por rotear pacotes no processo de roteamento de dados. O valor de  $cEIP$  é o custo de transmitir um pacote e o valor de  $cRouter$  é o custo de receber e transmitir um pacote. O *payoff*  $\pi_{it}$  do nó sensor  $i$  na rodada do jogo  $t$  é definido como

$$\pi_{it} = -(nEIP_{it} \times cEIP + nRouter_{it} \times cRouter)$$

em que  $nEIP_{it}$  é 1 caso o nó sensor  $i$  tenha enviado o seu EIP durante a rodada  $t$  do jogo e 0 caso contrário. O valor de  $nRouter_{it}$  é o número de vezes que o nó sensor  $i$  foi usado para rotear pacotes de dados durante a rodada  $t$  do jogo. Assim, o *payoff* máximo que um jogador pode obter é 0, quando ele não envia o seu EIP e nem participa do processo de roteamento. O *payoff* inicial dos nós sensores é, então, 0, sendo alterado à medida que ele participa do processo de roteamento e envia EIPs. O objetivo

d927(I)0.16303(o)-373.914(06)0.04  
-369.389(d)-0.312447(e)-368.841(r)-0.147034(o)0.245057(t)-0.147034(e)0.0490113765(t)-0.14707(u)17(t21(j)9((n)7m)

mento quando um nó com baixa reserva de energia é escolhido para fazer parte da rota. A parte superior do corte feito no nó representa a sua quantidade de energia no mapa de energia e a parte inferior a sua quantidade real. Na figura 4.1-a, o nó 3 é escolhido para rotear pacotes e, por saber que existem vizinhos com mais energia que ele, envia o seu EIP para o nó monitor, como ilustrado na figura 4.1-b. Por sua vez, o nó monitor, como ilustrado na figura 4.1-c, atualiza o mapa de energia e gera uma nova rota, que não passa mais pelo nó 3 (figura 4.1-d).

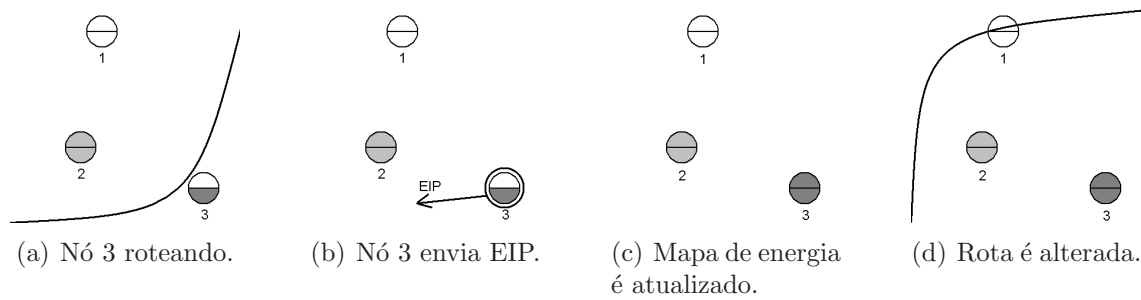


Figura 4.1: Comportamento do algoritmo de roteamento que usa as informações do mapa de energia.

**Algoritmo 1 : Pred-GT**


---

```

1: input:  $\pi_{t-1}$ ,  $cEIP$ 
2:
3: if  $\pi_{t-1} \geq -cEIP$  then
4:   Não envia o pacote
5: else
6:   if  $energyMustBeUpdated()$  then
7:     Envia o pacote
8:   else
9:     Não envia o pacote
10:  end if
11: end if

```

---

O algoritmo 1 utiliza o *payoff*  $\pi_{t-1}$  obtido pelo nó sensor na rodada anterior para decidir se o EIP deve ser enviado ou não. O *payoff*  $\pi_{t-1}$  informa se o nó sensor participou do processo de roteamento e se enviou o seu EIP. Caso ele tenha sido maior ou igual a  $-cEIP$ , significa que ele não participou do processo de roteamento e não precisa enviar o seu EIP. Caso contrário, significa que ele participou do processo de roteamento e foi um daqueles nós que obtiveram os piores *payoffs* da rodada. Então, se o nó julgar que existem nós mais capazes de rotear pacotes que ele, o seu EIP deve ser enviado. Essa decisão, representada pela função  $energyMustBeUpdated()$ , presente no algoritmo 1, pode ser feita de três maneiras, representadas pelos algoritmos Pred-GT- $\alpha$ , Pred-GT- $\beta$  e Pred-GT- $\gamma$ . A seção 4.2.2 descreve o algoritmo Pred-GT- $\alpha$ , que utiliza informações globais disseminadas do mapa de energia na tomada de decisão. A seção 4.2.3 descreve o algoritmo Pred-GT- $\beta$ , que utiliza as informações de energia dos nós sensores vizinhos na tomada de decisão. Por fim, a seção 4.2.4 descreve o algoritmo Pred-GT- $\gamma$ , que utiliza níveis de energia para definir a qualidade para rotear de um nó e, a partir disso, decide se o seu EIP deve ser enviado ou não.

### 4.2.2 Utilizando Informações do Mapa de Energia (Pred-GT- $\alpha$ )

No algoritmo Pred-GT- $\alpha$ , a decisão sobre a existência de nós sensores mais capazes de rotear pacotes é baseada em cinco atributos que o nó sensor mantém: (1)  $avg$ , a energia média da rede; (2)  $stdDev$ , o desvio padrão da energia média da rede; (3)  $cutEne$ , a energia de corte das regiões de baixa energia da rede, ou seja, todos os nós que possuem quantidade de energia menor que  $cutEne$  são nós sensores com baixa reserva de energia; (4)  $e$ , a energia do nó sensor; e (5)  $eAtSink$ , o valor da energia que o nó monitor estima que o nó sensor tem, ou seja, o valor da energia que se encontra no mapa de energia.

É importante ressaltar que os valores de  $avg$ ,  $stdDev$  e  $cutEne$  são calculados pelo

nó monitor a partir das informações contidas no mapa de energia  $e$ , após o cálculo, são embutidos nos pacotes que forem disseminados por ele. Assim, esses valores só poderão ser mantidos pelos nós sensores quando o processo de roteamento for uma disseminação de dados. O valor inicial de  $avg$  e  $cutEne$  é 0 e será atualizado apenas por nós sensores que receberam os pacotes de disseminação. Aqueles nós sensores que não receberam os pacotes de disseminação obtiveram  $payoffs$  maiores ou iguais a  $-cEIP$  e, portanto, não possuem incentivos para enviar seus EIPs e não precisam dos valores  $avg$ ,  $stdDev$  e  $cutEne$  atualizados. A função  $energyMustBeUpdated()$  do algoritmo Pred-GT- $\alpha$  é descrita no algoritmo 2.

---

**Algoritmo 2 :  $energyMustBeUpdated()$  do algoritmo Pred-GT- $\alpha$**

---

```

1: input:  $avg$ ,  $stdDev$ ,  $cutEne$ ,  $e$ ,  $eAtSink$ 
2:
3: if  $e < (avg - stdDev)$  and  $eAtSink > cutEne$  then
4:   return true
5: else
6:   return false
7: end if

```

---

A função  $energyMustBeUpdated()$  usa os valores  $avg$  e  $stdDev$  para conhecer a situação de energia do nó sensor perante toda a rede. O nó sensor deve enviar o seu EIP somente se a sua energia for menor que a energia média da rede subtraída do desvio padrão. Se essa condição for satisfeita, há uma probabilidade aproximada de 0,84 de haver nós sensores mais aptos a rotear pacotes que ele, uma vez que 84% dos valores de uma distribuição normal são maiores que a média subtraída do desvio padrão [SdFC91]. Além disso, se  $eAtSink$  for menor que  $cutEne$ , então o algoritmo de geração de rotas sabe sobre a sua condição de nó de baixa energia e o está usando como roteador por algum motivo como, por exemplo, ele ser o único nó sensor ativo na sua região. Assim, ele não precisa enviar o seu EIP e gastar a sua energia.

### 4.2.3 Utilizando Informações dos Vizinhos (Pred-GT- $\beta$ )

O algoritmo Pred-GT- $\beta$ , ao invés de usar a média da energia de todos os nós da rede, usa apenas as informações de energia dos nós sensores vizinhos na tomada de decisão do envio ou não do EIP. A média geral fornece ao nó sensor uma visão sobre a sua situação de energia perante a rede, mas não lhe informa sobre a sua situação perante a sua localidade na rede, que é o determinante para saber se ele está sendo usado incorretamente como roteador no processo de disseminação de dados. Um nó sensor pode ter uma reserva de energia abaixo da média da rede mas pode ter uma reserva de energia acima da média entre

os nós sensores próximos a ele, o que lhe capacita como melhor roteador da região. Assim, a função *energyMustBeUpdated()* do algoritmo Pred-GT- $\beta$  é descrita no algoritmo 3.

---

**Algoritmo 3 : *energyMustBeUpdated()* do algoritmo Pred-GT- $\beta$**

---

1: **input:**

entre valores de energia, que variam de  $0J$  até a energia inicial  $iEne$  do nó. Por sua vez, os níveis de energia variam de 1 até o número  $nNiveis$  de níveis de energia. Quanto menor for o valor do nível de energia de um nó sensor, menor será a sua energia. Os níveis são definidos por intervalos discretos de energia, sendo o nível 1 definido pelo intervalo de  $0J$  até  $iEne/nNiveis$  e o nível  $nNiveis$  definido pelo intervalo de  $iEne - (iEne/nNiveis)$  até  $iEne$ . Assim, a energia real  $e$  do nó sensor é convertida para um valor  $e_c$ , de modo que:

$$e_c = \left( \frac{e}{\frac{iEne}{nNiveis}} + 1 \right) \times \frac{iEne}{nNiveis}$$

O mapa de energia é então representado não mais pela energia real dos nós sensores, mas pela energia convertida segundo o nível de energia dos mesmos. A tabela 4.1 mostra exemplos de valores de energia convertidos para sistemas com 2 níveis de energia e 10 níveis de energia quando a energia inicial dos nós sensores é  $40J$ . O algoritmo de geração de rotas deve ler esse mapa de energia e selecionar para o roteamento os nós que possuem os maiores níveis de energia para rotear pacotes. Nós sensores que possuem vizinhos com níveis de energia maiores que o dele não devem fazer parte do processo de roteamento e, se estão fazendo, é porque provavelmente a sua informação de energia no mapa de energia está desatualizada. Quando isso acontecer, eles devem enviar o seu EIP.

Energia real	Energia com 2 níveis	Energia com 10 níveis
$40J$	$40J$	$40J$
$35J$	$40J$	$36J$
$30J$	$40J$	$32J$
$25J$	$40J$	$28J$
$20J$	$20J$	$20J$
$15J$	$20J$	$16J$
$10J$	$20J$	$12J$
$5J$	$20J$	$8J$

Tabela 4.1: Relação da energia real de um nó sensor que tem energia inicial de  $40J$  com sua energia convertida para um sistema de 2 níveis de energia e 10 níveis de energia.

O algoritmo 4 descreve a função *energyMustBeUpdated()* para o algoritmo Pred-GT- $\gamma$ . O parâmetro de entrada *eLevel* se refere ao nível de energia do nó sensor e o parâmetro *highestNeighEnergyLevel* se refere ao nível de energia mais alto que algum vizinho seu possui.

É importante ressaltar que a escolha de  $nNiveis$  influencia a precisão do mapa de

---

**Algoritmo 4 : `energyMustBeUpdated()` do algoritmo Pred-GT- $\gamma$** 

---

```

1: input:  $eLevel$ ,  $highestNeighEnergyLevel$ 
2:
3: if  $eLevel < highestNeighEnergyLevel$  then
4:   return true
5: else
6:   return false
7: end if

```

---

energia, o número de EIPs a serem enviados e a garantia de que todos os nós que fizeram parte do roteamento tenham o maior nível de energia dentre os seus vizinhos. Quanto menor for o valor de  $nNiveis$ , maior será a diferença de energia que poderá existir entre nós sensores que forem de níveis de energia diferentes. Dessa maneira, a precisão do mapa de energia diminui conforme diminui o número de níveis de energia. Além disso, quanto maior for o intervalo discreto que define os níveis, mais tempo levará para um nó sensor mudar de nível e enviar um EIP, se for o caso. Por fim, se o intervalo entre a energia inicial e a final de um nível for muito pequeno, menor será a diferença de energia entre nós sensores com níveis de energia diferentes e menos diferença a variação dos níveis fará no processo de geração das rotas.

### 4.3 Equilíbrio de Nash na Construção do Mapa de Energia

Na teoria dos jogos, um dos conceitos mais importantes é o equilíbrio de Nash [FT91]. O equilíbrio de Nash é caracterizado quando nenhum jogador se arrepende da estratégia que optou, ou seja, nenhum jogador conseguiria melhorar o seu *payoff* alterando a sua estratégia quando todos os outros mantiverem as suas. O equilíbrio de Nash não garante a solução ótima sempre, mas garante estabilidade em jogos não-cooperativos, ou seja, jogos nos quais não há qualquer acordo de cooperação entre os jogadores. Além disso, um jogo pode possuir vários ou nenhum equilíbrio de Nash, e a detecção das suas ocorrências nem sempre é possível.

Para detectar o equilíbrio de Nash no modelo proposto, deve-se analisar os *payoffs* de todos os jogadores. O maior *payoff* que um nó sensor pode obter numa rodada do jogo é 0, ou seja, ele não roteou e não enviou o seu EIP. O segundo maior *payoff* que um nó pode obter é  $-cEIP$ , que é pago aos nós que enviaram os seus EIPs e não fizeram parte do roteamento. Em terceiro lugar, tem-se o *payoff* de valor  $-cRouter$ , que é pago aos nós sensores que fizeram parte do roteamento e decidiram não enviar seus EIPs. Por fim, o

pior *payoff* que um nó sensor pode obter numa rodada é  $-cEIP - cRouter$ , que é pago aos nós que decidiram enviar os seus pacotes com as suas informações de energia e que rotearam pacotes.

Para o jogo se encontrar em equilíbrio de Nash, os *payoffs* pagos numa rodada devem ser sempre os melhores possíveis, tanto para os nós sensores que não fizeram parte do roteamento quanto para os que fizeram. Os nós sensores que não fizeram parte do roteamento obtiveram *payoffs* iguais a 0 ou a  $-cEIP$ . Todos os nós sensores que receberam 0 de *payoff* não tinham como obter um *payoff* melhor, uma vez que 0 é o valor máximo que um nó pode receber. Os nós que receberam  $-cEIP$  provavelmente também não tinham como obter um *payoff* melhor, uma vez que se eles enviaram os seus EIPs é porque fizeram parte do roteamento na rodada passada e, caso não enviassem os seus pacotes, iriam provavelmente fazer parte do roteamento novamente e teriam um *payoff* pior, uma vez que  $cEIP < cRouter$ . Há casos de nós sensores que receberam *payoffs* iguais a  $-cEIP$  mas poderiam receber 0, pois iriam deixar de fazer parte do roteamento mesmo se não enviassem seus EIPs. Isso pode acontecer, por exemplo, quando os nós sensores vizinhos ao nó monitor morrem e nenhum processo de disseminação de dados pode mais ser propagado na rede. Assim, todos os nós sensores que enviaram EIPs na rodada que o nó monitor se tornou inalcançável poderiam ter obtido um *payoff* melhor se não tivessem enviado seus EIPs.

Para os nós sensores que fizeram parte do roteamento, os *payoffs* possíveis são  $-cRouter$  e  $-cEIP - cRouter$ . Quando o *payoff* pago a um nó for  $-cRouter$ , existem dois casos. Quando o nó poderia ter evitado de fazer parte do processo de roteamento se tivesse enviado seu EIP e quando não poderia. No primeiro caso, a sua informação de energia provavelmente está desatualizada no mapa de energia e, por isso, ele foi usado como roteador. No segundo caso, se ele tivesse optado por alterar a estratégia e enviar o seu EIP, nada aconteceria, uma vez que provavelmente a sua informação de energia no mapa de energia continuaria sendo alta em relação à sua localidade na rede e a rota continuaria a passar por ele, com a diferença que o seu *payoff* passaria de  $-cRouter$  para  $-cEIP - cRouter$ . Os nós sensores que receberam  $-cEIP - cRouter$  como *payoff* poderiam ter recebido um *payoff* melhor caso tivessem alterado as suas estratégias, ou seja, não tivessem enviado os seus EIPs. Isso acontece com nós sensores que enviaram os seus EIPs acreditando que seriam excluídos do processo de roteamento mas que, por alguma razão, não foram.

Então, para o sistema estar em equilíbrio de Nash, os *payoffs* dos jogadores devem ser exclusivamente 0,  $-cEIP$ , somente para os nós que fizeram parte do roteamento na rodada anterior e deixaram de fazer por terem enviado seus EIPs, e  $-cRouter$ , somente



para os nós que não poderiam evitar de parte do roteamento se tivessem enviado seus EIPs, ou seja, para os nós que possuem alta reserva de energia e foram escolhidos para fazer parte do processo de roteamento. Isso garante que quando o jogo estiver em equilíbrio de Nash, as rotas utilizarão apenas nós com alta reserva de energia para rotear pacotes. E uma vez que os nós roteadores são aqueles com alta reserva de energia, eles não possuem incentivos para enviar EIPs, garantindo ao jogo um estado estável em que nenhum EIP é enviado. Isso reduz significativamente a comunicação entre os nós sensores, economizando a energia da rede e estendendo o seu tempo de vida. Além disso, quando o sistema está em equilíbrio de Nash, todos os nós sensores obtiveram os melhores *payoffs* possíveis a eles. Isso significa que eles consumiram a menor quantidade de energia possível. Quando os nós sensores estão consumindo a menor quantidade de energia possível eles estão estendendo seus tempos de vida e, por consequência, a da rede também. Isso faz do equilíbrio de Nash um estado a ser alcançado pelos algoritmos propostos por este trabalho. O próximo capítulo apresenta os resultados de simulação para os algoritmos propostos neste trabalho.

## 4.4 Conclusões

O presente capítulo apresentou o modelo baseado na teoria dos jogos para construção de atualização do mapa de energia para ser usado na geração de rotas eficientes em energia. Para implementar o modelo, foram propostos três algoritmos: Pred-GT- $\alpha$ , Pred-GT- $\beta$  e Pred-GT- $\gamma$ . Os três algoritmos se distinguem na forma que fazem os nós sensores decidirem se devem ou não enviar os seus EIPs. É desejável que um nó sensor decida pelo envio do EIP somente quando isso for retirá-lo do processo de roteamento. Além disso, é desejável que o sistema alcance o estado de equilíbrio de Nash, em que as rotas geradas usem nós com alta reserva de energia e poucos EIPs sejam enviados, fazendo que a rede economize energia e o seu tempo de vida seja estendido. No próximo capítulo serão apresentados os resultados de simulação para os algoritmos propostos neste capítulo.

## Capítulo 5

# Resultados de Simulação

Este capítulo apresenta os resultados de simulação para os algoritmos propostos no capítulo anterior. A seção 5.1 descreve o cenário e as topologias usadas nas simulações. A seção 5.2 descreve os resultados obtidos em uma topologia com distribuição homogênea da energia inicial dos nós sensores. A seção 5.3 descreve os resultados obtidos em uma topologia com distribuição aleatória da energia inicial dos nós sensores. Os resultados obtidos em uma topologia com uma região com baixa reserva de energia são descritos na seção 5.4. A seção 5.5 apresenta, por fim, os resultados obtidos em uma topologia que forma uma região com baixa reserva de energia em tempo de simulação. A seção 5.6 apresenta as conclusões obtidas a partir das simulações realizadas.

### 5.1 Cenário

Para avaliar o desempenho dos algoritmos para construção do mapa de energia usando teoria dos jogos, foram realizadas simulações que consideram uma rede de sensores com 200 nós homogêneos e estáticos com uma quantidade finita de energia, ou seja, a sua recarga é considerada inviável. Os nós são depositados aleatoriamente, formando uma topologia plana em um campo de sensoriamento de  $35 \times 35 m^2$ . Cada nó sensor tem, em média, 8 vizinhos, possui energia inicial de  $40 J$ , tem raio de comunicação de  $5 m$  e conhece a sua localização. O nó monitor, localizado no canto inferior esquerdo da rede, não possui restrições de energia e conhece a localização de todos os nós sensores da rede, enviando 200 mensagens de disseminação uniformemente distribuídas durante os 1000 segundos de simulação. As rotas são construídas a partir do algoritmo de roteamento em curva *Trajectory and Energy-based Data Dissemination* (TEDD) [dVMGM<sup>+</sup>05]. O modelo de dissipação de energia é baseado no *State-based Energy Dissipation Model* (SEDM) [MMLN05] e os parâmetros de consumo de cada estado do SEDM foram calculados baseados no consumo

do Mica2 [Mic04]. Os valores de  $cEIP$  e  $cRouter$  também são baseados no Mica2, sendo que o valor de  $cEIP$  é 0.1011, que é o valor em Jaules de uma transmissão, e valor de  $cRouter$  é 0.15, que é o valor em Jaules de uma transmissão e de uma recepção. É importante ressaltar que os resultados apresentados neste capítulo não consideram o gasto de energia dos nós sensores com o envio e roteamento de EIPs. A tabela 5.1 apresenta um resumo dos parâmetros de simulação.

Os algoritmos propostos para construir o mapa de energia foram implementados no simulador *Network Simulator 2.30* [ns202] e são comparados entre si e com o algoritmo baseado em predição descrito em [MMLN05] e que será chamado Pred, tendo o parâmetro *threshold* com o valor de 3%, ou seja, 3% é o erro máximo permitido entre o mapa real e o previsto. O algoritmo Pred-GT- $\gamma$  usa 10 níveis de energia. Todos os resultados de simulação correspondem à média aritmética de  $n$  simulações, em que  $n$  é o menor valor que provê a confiança desejada [Jai91]. Em todos os experimentos são usados níveis de confiança de 95%. Além disso, o *T-test* [Jai91] com 0.05 de significância é usado para afirmar de maneira significativa que um protocolo é melhor, pior ou semelhante a outro.

Parâmetro	Valor
Número de nós	200
Energia Inicial	$40J$
Área de Sensoriamento	$35 \times 35m^2$
Raio de Comunicação	$5m$
Tempo de Simulação	$1000s$
Mensagens de Disseminação	200
<i>Threshold</i> da Predição	3%
$cEIP$	0.1011
$cRouter$	0.15

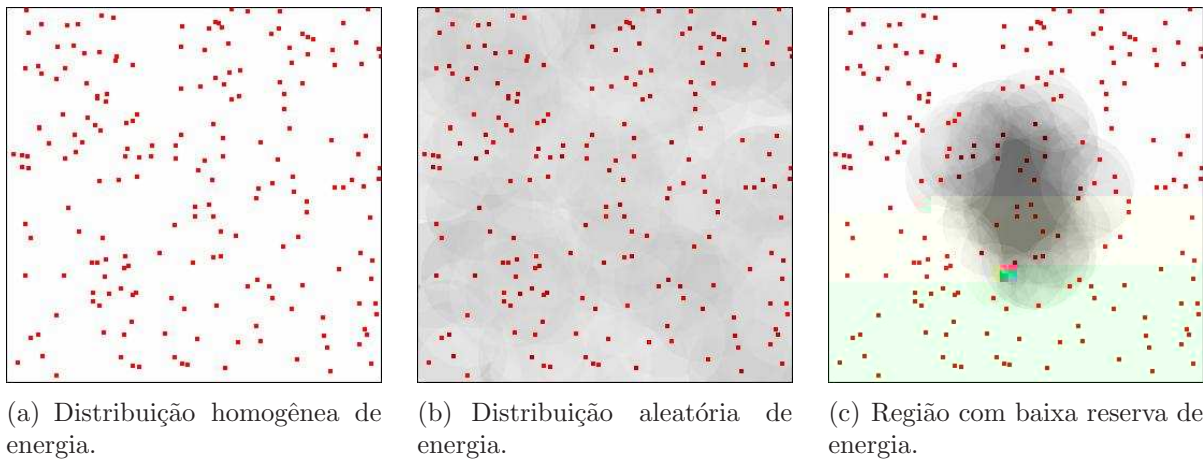
**Tabela 5.1:** Valores padrão usados nas simulações.

Para avaliar a robustez dos algoritmos desenvolvidos neste trabalho, quatro topologias serão avaliadas nas simulações. A primeira topologia é caracterizada por uma distribuição homogênea da energia inicial dos nós sensores, ou seja, todos os nós sensores apresentam suas energias iniciais idênticas ( $40J$ ). Um exemplo dessa topologia é ilustrado na figura 5.1-a.

A segunda topologia da simulação pode ser ilustrada pela figura 5.1-b, em que a energia inicial dos nós sensores varia aleatoriamente, com valores entre  $28J$  e  $40J$ . Essa topologia é usada para avaliar os algoritmos de construção do mapa de energia quando é exigida uma maior sensibilidade do algoritmo de geração de rotas. O algoritmo de geração de

rotas deve ser capaz de usar as informações contidas no mapa de energia para escolher os nós sensores mais capazes de rotear pacotes, ou seja, aqueles que possuem maiores reservas de energia.

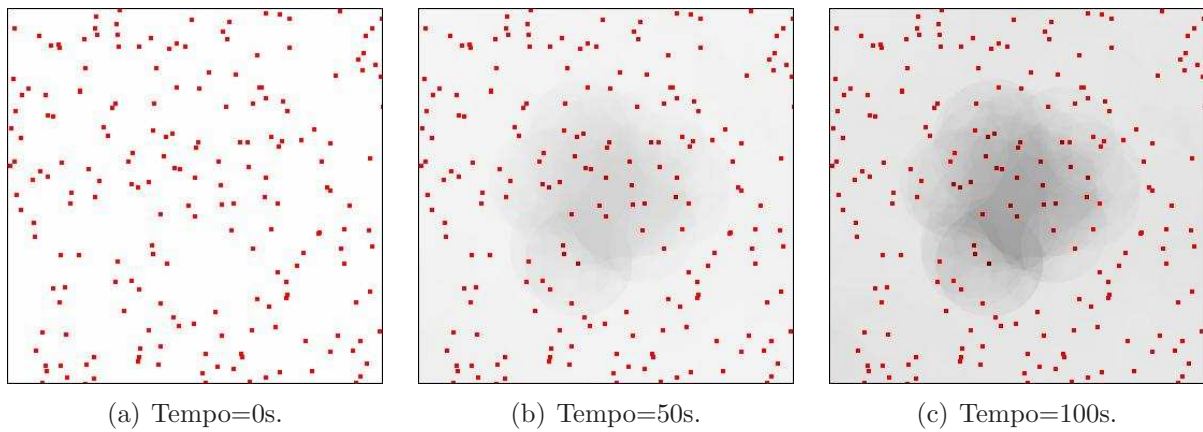
A figura 5.1-c ilustra um mapa de energia que serve de exemplo para a terceira topologia da simulação, que apresenta uma região de baixa reserva de energia localizada no centro da mesma. Essa topologia é usada para avaliar a capacidade do algoritmo TEDD de evitar os nós sensores localizados na região de baixa reserva de energia quando esse usa as informações dos mapas de energia construídos pelos algoritmos avaliados. Além disso, é necessário observar o número de nós sensores localizados dentro da região de baixa energia da rede que enviaram EIPs, uma vez que não é desejável que esses nós sensores gastem suas energias com pacotes de controle, como são caracterizados os EIPs.



**Figura 5.1:** Exemplos de topologias usadas nas simulações.

Por fim, a quarta topologia da simulação é uma topologia com distribuição homogênea de energia mas que, nos 100 segundos iniciais, se transforma numa topologia com uma região de baixa reserva de energia localizada no centro da mesma, como ilustrado na figura 5.2. Essa topologia é usada para avaliar o comportamento dos algoritmos de construção do mapa de energia quando a região com baixa reserva de energia é formada em tempo de simulação. É desejável que os algoritmos de construção do mapa de energia sejam capazes de detectar a formação da região com baixa reserva de energia e forneçam informações suficientes ao algoritmo de geração de rotas de maneira que as rotas evitem passar por essa região.

A seção 5.2 apresenta os resultados na topologia com distribuição homogênea de energia. Os resultados na topologia com uma distribuição aleatória de energia são apresentados na seção 5.3. A seção 5.4 apresenta os resultados na topologia com uma região com baixa reserva de energia e, por fim, a seção 5.5 apresenta os resultados na topologia que



**Figura 5.2:** Topologia que forma uma região com baixa reserva de energia.

forma a região com baixa reserva de energia em tempo de simulação. Por fim, a seção 5.6 apresenta as conclusões obtidas a partir das simulações realizadas.

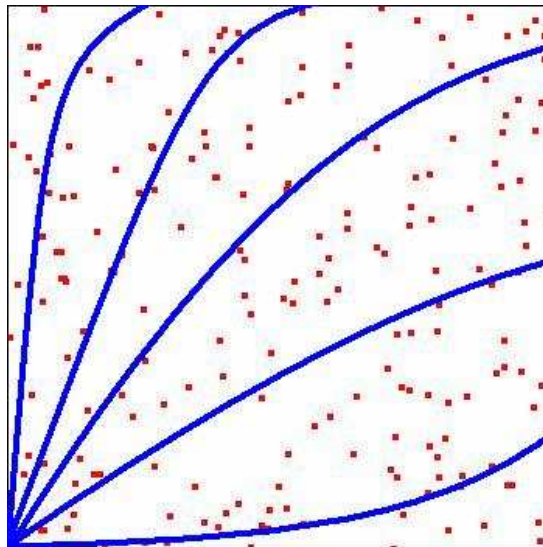


Figura 5.3: Curvas geradas para uma topologia com distribuição homogênea de energia.

menos nós possuem incentivos para enviar EIPs. Se, por exemplo, como ilustrado na figura 5.4, os nós vizinhos ao nó monitor morrerem, os demais nós sensores não receberão nenhum pacote de disseminação e, por isso, não vão enviar nenhum EIP quando estiverem executando os algoritmos de construção do mapa de energia baseados na teoria dos jogos. Por outro lado, quando estiverem executando o algoritmo Pred, os mesmos enviarão EIPs numa taxa cada vez maior. Isso mostra que os algoritmos que usam a teoria dos jogos na tomada de decisão reduzem a comunicação na rede. O algoritmo Pred-GT- $\gamma$  ainda envia menos EIPs que o algoritmo Pred-GT- $\alpha$  e o Pred-GT- $\beta$ , pois nós sensores com o mesmo nível de energia são tratados como nós sensores com a mesma quantidade de energia residual. Como a topologia é homogênea, há pouca diferença de energia entre os nós sensores, fazendo com que o desvio padrão seja pequeno e as condições de envio do EIP para os algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$ , que consideram a energia real dos nós sensores, sejam mais facilmente alcançadas.

Além de construir um mapa de energia usando poucos EIPs, os algoritmos devem construir um mapa de energia com informações relevantes para a geração de rotas eficientes em energia. Não basta construir um mapa de energia usando poucos EIPs se o mesmo não é eficiente para o algoritmo de roteamento que o utiliza. Um mapa de energia é eficiente apenas se o mesmo contiver informações suficientes sobre os nós sensores capazes de rotear pacotes na rede. Um mapa de energia construído e atualizado a um baixo custo de EIPs, como aquele construído pelo algoritmo Pred-GT- $\gamma$ , não terá utilidade se o mesmo não fornecer informações sobre os nós sensores que tem alta reserva de energia e os que não tem. Uma forma de avaliar a qualidade do mapa de energia criado é verificar a energia

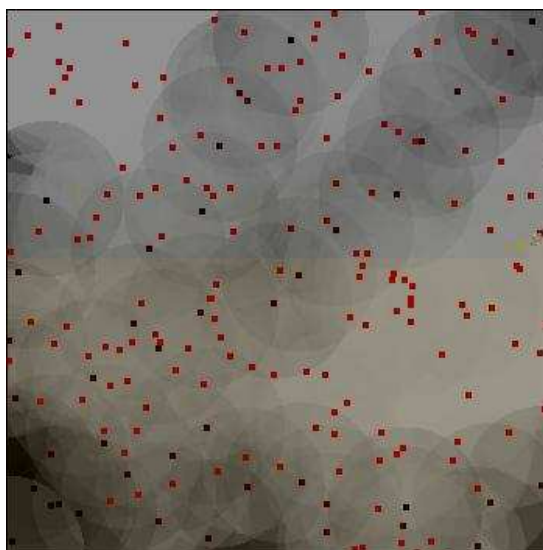


Figura 5.4: Exemplo de um mapa de energia que representa uma topologia em que os nós sensores vizinhos ao nó monitor estão mortos.

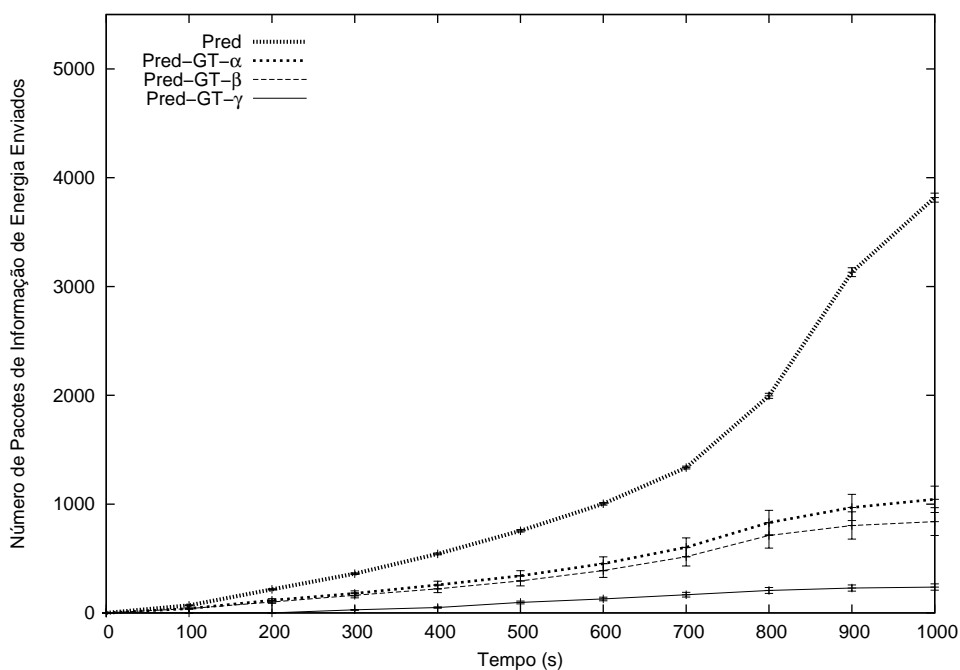
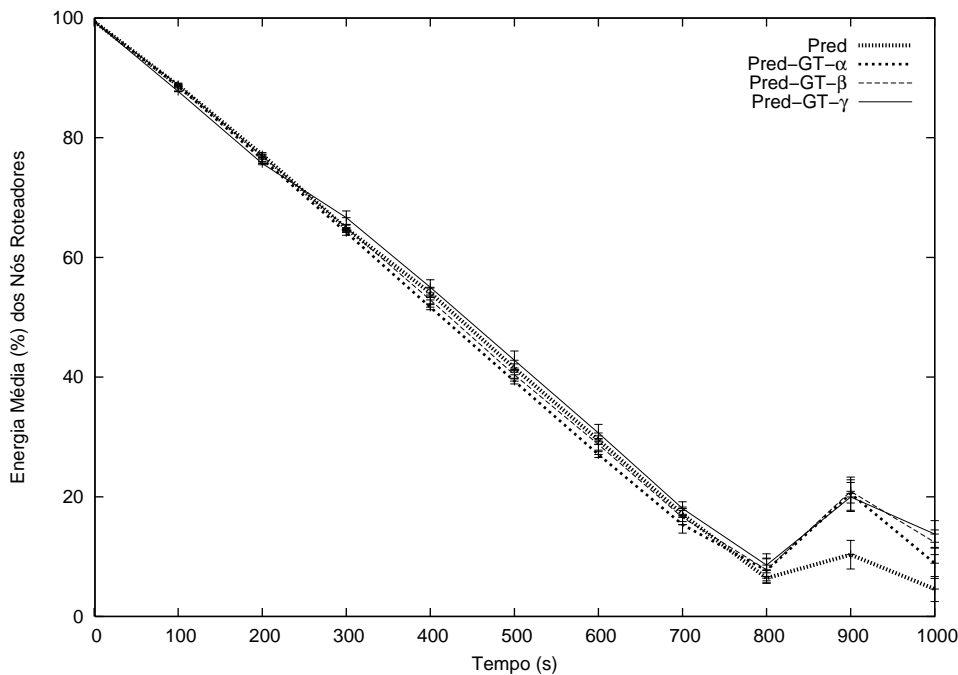


Figura 5.5: Número de EIPs enviados para o nó monitor.

média dos nós sensores que rotearam pacotes. Quanto mais alta for a energia média dos nós sensores que rotearam pacotes, melhor foi a curva gerada pelo algoritmo TEDD.

A figura 5.6 mostra a energia média dos nós sensores que rotearam pacotes durante

sensores possuírem uma distribuição homogênea da energia inicial, os algoritmos propostos neste trabalho são capazes de fornecer ao algoritmo de roteamento informações tão precisas quanto a informação contida no mapa de energia fornecido pelo algoritmo Pred, que produz um mapa de energia com a diferença do real sendo no máximo de 3%. Dessa forma, os algoritmos baseados em teoria dos jogos são capazes de informar quais nós estão aptos a rotear pacotes e quais não estão quando a topologia apresenta uma distribuição homogênea da energia inicial dos nós sensores. O súbito aumento da energia média dos nós roteadores que acontece após os 800 segundos de simulação é devido à morte de nós sensores que faziam parte do roteamento. Uma vez que eles morrem, a energia média dos nós roteadores aumenta, pois as suas baixas reservas de energia que contribuíam para um menor valor da energia média não contribuem mais.



**Figura 5.6:** Energia média dos nós roteadores.

Além de verificar a qualidade dos nós roteadores, é necessário avaliar a quantidade de energia desperdiçada na rede com o envio de EIPs. Um nó sensor só deve enviar seu EIP quando isso for retirar o mesmo do processo de roteamento. Da mesma forma, nós sensores que estão fazendo parte do roteamento não devem enviar seus EIPs se isso não for capaz de retirá-los do processo de roteamento. Uma forma de verificar isso é a

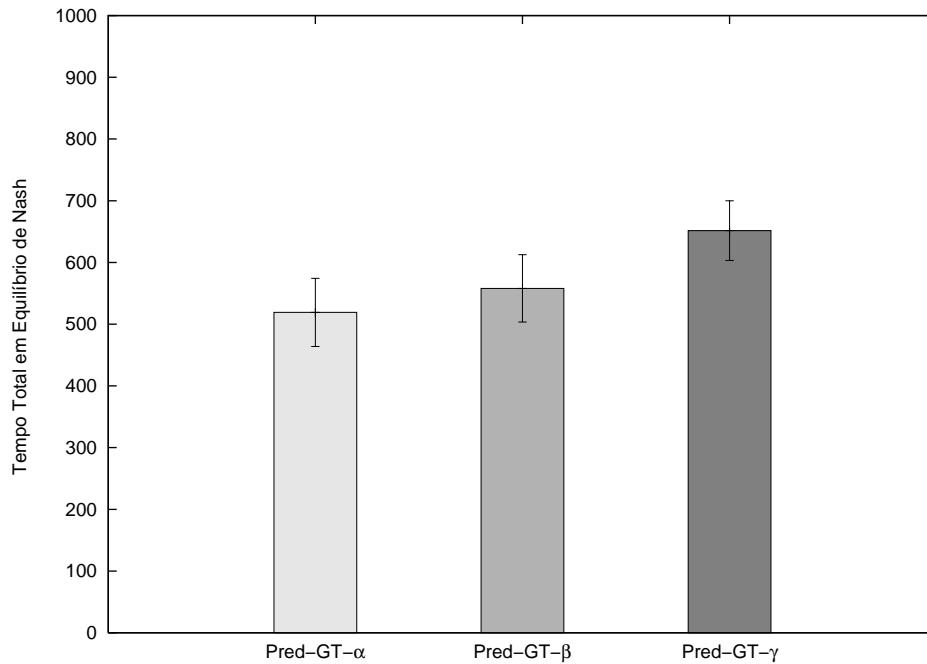


de Nash, todos os EIPs enviados foram capazes de retirar o nó sensor do processo de roteamento. Além disso, nenhum nó sensor que roteou pacotes evitaria de fazer parte do processo de roteamento a partir do envio do seu EIP.

No caso das simulações apresentadas neste capítulo, a verificação do equilíbrio de Nash é feita através de uma nova simulação da rodada para cada nó sensor, alterando a sua estratégia original. Para cada nó sensor que enviou o seu EIP é simulada a mesma rodada com ele não enviando o seu EIP. Da mesma forma, para cada nó sensor que não enviou o seu EIP na rodada, é simulada uma rodada com ele enviando o seu EIP. Se algum nó sensor, nessa nova simulação, obtiver um *payoff* maior que aquele obtido a partir da estratégia original, então o sistema não está em equilíbrio de Nash na rodada corrente do jogo. Os nós sensores que obtiveram *payoff* igual zero não precisam ser testados, uma vez que zero é o melhor *payoff* que um nó sensor pode obter. Além disso, é importante ressaltar que o TEDD divide a topologia em setores, de maneira que nós sensores que pertencem a setores diferentes não fazem parte do mesmo jogo, uma vez que a ação de um não influencia no *payoff* do outro.

A figura 5.7 mostra o tempo total que o jogo esteve no estado de equilíbrio de Nash. Na topologia desta seção, há pouca variação entre a energia dos mesmos durante a simulação, fazendo que o tempo em equilíbrio de Nash dos algoritmos seja semelhante. O algoritmo Pred-GT- $\gamma$  apresenta um tempo em equilíbrio de Nash um pouco maior que os demais, pois a classificação dos nós segundo níveis de energia faz com que pequenas diferenças de energia sejam desconsideradas, tanto na decisão do envio do EIP quanto na geração das rotas a partir do mapa de energia. Isso garante uma maior estabilidade, pois um nó sensor tem menos incentivos para enviar EIPs quando existe uma pequena diferença de energia. O algoritmo Pred-GT- $\beta$ , por sua vez, possui um tempo em equilíbrio de Nash superior ao algoritmo Pred-GT- $\alpha$ , pois a tomada de decisão dos nós sensores utiliza informações locais, que dão aos mesmos a certeza da existência de outros nós sensores mais aptos a rotear nas suas localidades.

Em conjunto com o equilíbrio de Nash, o desperdício de energia da rede também é verificado quando um nó sensor envia continuamente o seu EIP para o nó monitor a cada momento que a sua energia real se torna diferente da sua energia prevista num percentual maior que o *threshold* sem que isso o retire do processo de roteamento. Quando um nó sensor detecta a existência de outros nós sensores com mais energia que ele e que não estão roteando pacotes, ele possui incentivos para enviar o seu EIP, pois ele acredita que o seu envio é capaz de retirá-lo do processo de roteamento. O problema é que nem sempre um EIP é suficiente para retirar um nó sensor do processo de roteamento. No caso do TEDD, que gera equações matemáticas a partir das informações contidas no mapa de

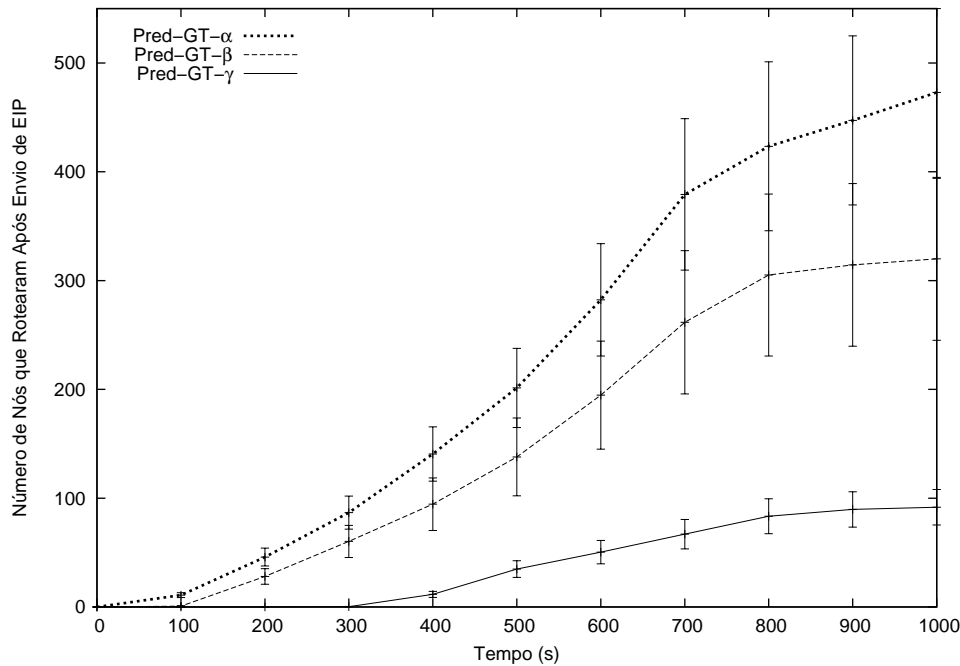


**Figura 5.7:** Tempo em equilíbrio de Nash.

energia, um único EIP enviado pode atualizar uma informação que não é suficiente para o algoritmo gerador de rotas alterar a equação matemática. Isso acontece devido a alta densidade da rede e a qualidade da informação que o nó sensor dispõe para tomar a decisão de envio ou não do seu EIP. Assim, quando um nó sensor que envia o seu EIP não é retirado do processo de roteamento, ele continuará a enviar EIPs e a gastar a sua energia desnecessariamente. É desejável então que o número de EIPs que não retira o nó sensor do processo de roteamento seja o menor possível, pois quanto maior for, maior será a tendência de envio de EIPs dessa natureza e maior será o gasto desnecessário de energia.

A figura 5.8 ilustra o número total de vezes que um nó sensor enviou um EIP e continuou a rotear pacotes nas rodadas seguintes do jogo. O algoritmo Pred-GT- $\alpha$ , por possuir uma informação menos precisa que a dos demais algoritmos, faz os nós sensores enviarem mais EIPs sem utilidade durante o tempo de simulação. Os algoritmos Pred-GT- $\beta$  e Pred-GT- $\gamma$ , por terem acesso a informações locais, dão aos nós sensores a certeza da existência de outros nós sensores mais capazes de rotear pacotes nas suas localidades. Novamente, por desconsiderar pequenas diferenças de energias entre os nós sensores, o algoritmo Pred-GT- $\gamma$  evita que os nós sensores gastem as suas energias nesses casos.

Outra forma de avaliar o gasto de energia dos nós sensores é a partir dos *payoffs* obtidos por eles. O *payoff* é calculado em função das atividades de comunicação presentes no envio



**Figura 5.8:** Número de nós enviaram EIPs e continuaram a rotear.

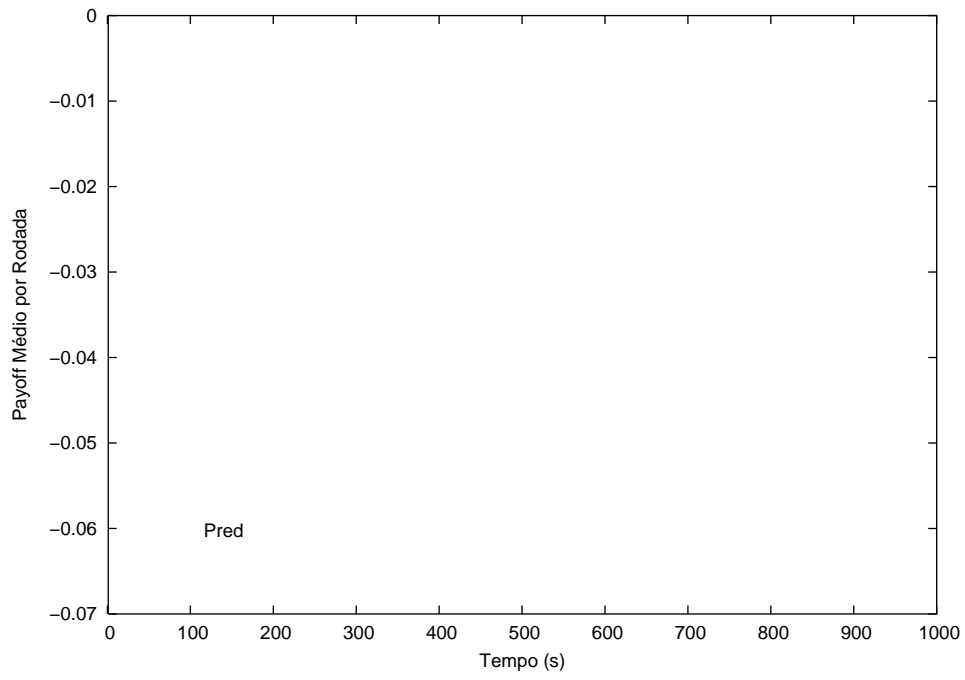
do EIP e na participação no processo de roteamento. Isso permite que ele seja usado como métrica de avaliação também para o algoritmo Pred, apesar do mesmo não ser baseado nos conceitos da teoria dos jogos. Então, quanto menor é o *payoff* dos nós sensores, maior foi o consumo de energia decorrente da comunicação de dados.

As figuras 5.9 e 5.10 mostram, respectivamente, a média dos *payoffs* por rodada e a média cumulativa dos *payoffs* dos nós sensores durante o tempo de simulação. O consumo de energia da rede é menor nos três algoritmos propostos por este trabalho que no Pred. A diferença dos *payoffs* cumulativos dos algoritmos baseados na teoria dos jogos e do Pred vista na figura 5.10 é decorrente principalmente do envio de EIPs, uma vez que o número de disseminações é o mesmo para todos os algoritmos. É importante ressaltar que à medida que o tempo de simulação chega ao fim e alguns nós sensores morrem, o *payoff* dos nós sensores nos algoritmos baseados em teoria dos jogos aumenta, uma vez que nesses algoritmos, quanto menos um nó roteia pacotes, menos incentivos ele possui para enviar EIPs. No algoritmo Pred, apesar de ainda haver gastos com o envio de EIPs, esses são menos frequentes, pois à medida que o roteamento diminui, o consumo de energia também diminui e, com isso, aumenta o tempo para a diferença entre a energia real de um nó sensor e a prevista atingir o *threshold*. A súbita queda e o súbito aumento do *payoff* médio dos nós sensores que estão executando o algoritmo Pred verificados na figura 5.9 é em decorrência da morte de nós sensores a partir dos 800 segundos de simulação. Nos 700

segundos de simulação, a energia desses nós sensores que irão morrer nos 800 segundos de simulação é próxima de 0. Assim, o parâmetro *threshold* é alcançado rapidamente, fazendo que a taxa de envio de EIPs aumente significativamente e o *payoff* dos mesmos, por sua vez, decresça de forma abrupta. Isso influencia na queda súbita do *payoff* médio por rodada dos nós sensores após os 700 segundos de simulação. O aumento súbito ocorre, por sua vez, depois que os nós sensores começam a morrer e param de enviar EIPs a altas taxas, contribuindo para o aumento do *payoff* médio por rodada.

Um caso interessante dessa situação ocorre quando os nós sensores vizinhos ao nó monitor morrem e esse fica inacessível, como ilustrado na figura 5.4. Quando os nós sensores estão executando o Pred como algoritmo de construção e atualização do mapa de energia, mesmo com o nó monitor inacessível, EIPs ainda são enviados e o *payoff* continua a decrescer. Quando estão executando os algoritmos baseados na teoria dos jogos, o *payoff* total da rede será zero, uma vez que nenhum nó sensor está fazendo parte do processo de roteamento e enviando EIPs. Isso impede o gasto da energia da rede com transmissões desnecessárias, economizando a sua energia. Além disso, como o nó monitor está inalcançável, não há razão para a rede continuar funcionando. O ideal é a rede desligar as suas funções e aguardar pela reposição de nós sensores nas regiões sem energia. Assim, sem o uso de um processo de disseminação de dados para alterar o modo de operação dos nós sensores, pode-se configurá-los para desligarem todas as suas funções quando a rede alcançar esse estado, pois eles possuem informações locais suficientes para a detecção do mesmo. Isso mostra que, além da economia no número de EIPs necessários para construção e atualização do mapa de energia, o modelo de construção do mapa de energia a partir dos conceitos da teoria dos jogos pode reduzir o número de comunicações em outras situações, economizando a energia da rede e estendendo o seu tempo de vida.

A figura 5.11 ilustra a evolução do mapa de energia real da rede e daqueles construídos pelos algoritmos avaliados neste capítulo na topologia desta seção. Pode-se concluir que os mapas de energias gerados diferem significativamente do mapa de energia real e do mapa de energia gerado pelo Pred. Isso mostra que um mapa de energia não necessita ser próximo do real para servir de maneira eficiente algoritmos de roteamento que usem as informações contidas nele. A figura 5.6 mostra que a qualidade das rotas geradas é a mesma para os quatro algoritmos avaliados. O algoritmo Pred-GT- $\gamma$ , apesar de ser o algoritmo que apresentou os melhores resultados na topologia com distribuição homogênea de energia, é o que constrói o mapa de energia mais diferente do real.



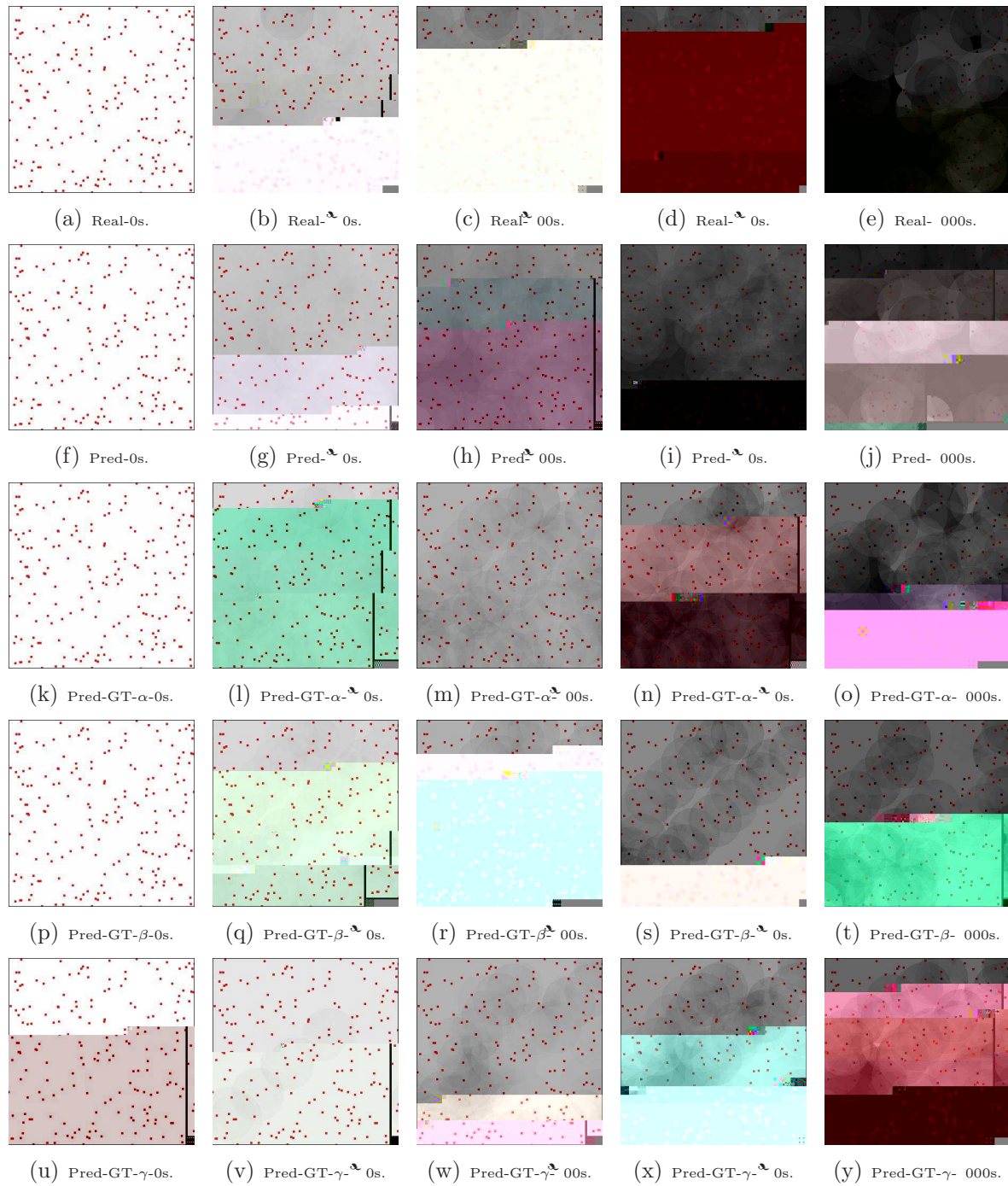


Figura 5.11: A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário com distribuição homogênea de energia.

rotear pacotes entre os nós da rede, dificultando a análise da qualidade das rotas geradas pelo TEDD. Uma vez que os nós sensores possuem energias iniciais distintas entre si, é possível avaliar se as rotas geradas a partir dos mapas de energia construídos pelos algoritmos avaliados neste trabalho usam os nós sensores mais capazes de rotear pacotes. A figura 5.12 ilustra exemplos de curvas geradas para a topologia desta seção.

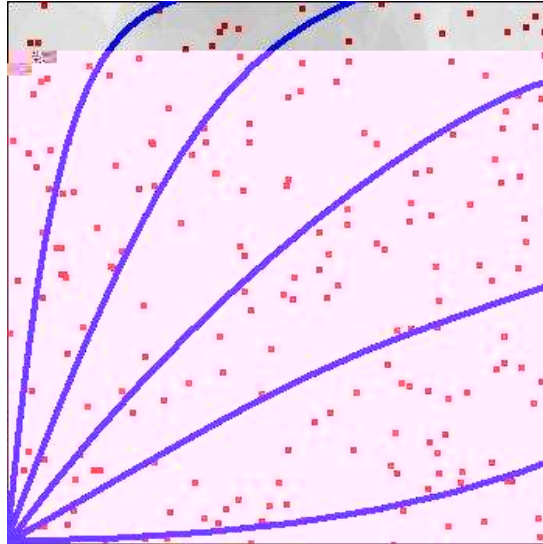
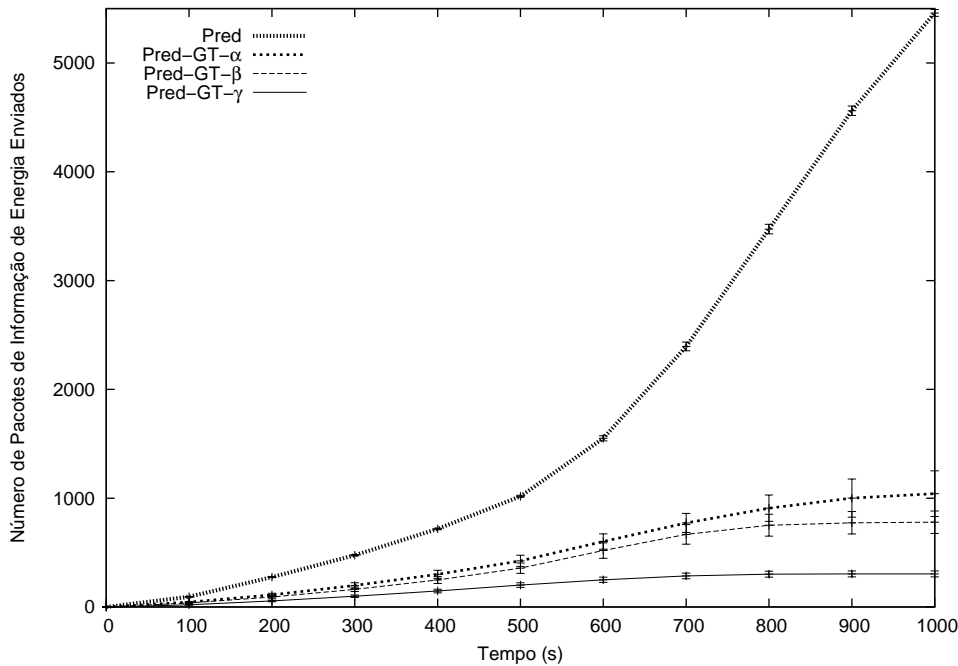


Figura 5.12: Curvas geradas para uma topologia com uma distribuição aleatória das suas energias iniciais.

Assim como foi verificado na seção anterior, o número de EIPs necessários para construção e atualização do mapa de energia na topologia com distribuição aleatória de energia foi significativamente superior com o Pred, como ilustrado na figura 5.13. Novamente, o algoritmo Pred-GT- $\gamma$  apresentou o melhor resultado. Nota-se ainda que a taxa de envio de EIPs do algoritmo Pred-GT- $\gamma$  alcança a estabilidade a partir dos 600s de simulação, ocorrendo antes neste cenário que no cenário da topologia com distribuição homogênea de energia. Isso ocorre porque nesta topologia os nós sensores começam a morrer mais cedo, uma vez que a energia inicial dos mesmos pode variar de  $28J$  até  $40J$ . Quando nós sensores morrem, menos nós sensores recebem os pacotes de disseminação e, por consequência, menos nós sensores roteam pacotes. Como os algoritmos baseados na teoria dos jogos faz os nós sensores enviarem EIPs somente se tiverem roteado pacotes, então a taxa de envio de EIPs decai à medida que cai o roteamento de pacotes de disseminação na rede. Além disso, o número de EIPs enviados pelo Pred é maior na topologia desta seção que na da anterior, uma vez que os nós sensores possuem energias iniciais menores, fazendo que o *threshold* seja alcançado com mais frequência.

Uma vez que o cenário desta seção apresenta uma diferença significativa entre as



**Figura 5.13:** Número de EIPs enviados para o nó monitor.

energias iniciais dos nós sensores, a energia média dos nós roteadores poderá variar mais se os mapas de energias fornecidos pelos algoritmos avaliados não contiverem informações atualizadas sobre quais nós estão aptos a rotear e quais não estão. Mas, a partir da análise da figura 5.14, que mede a energia média dos nós roteadores durante o tempo de simulação, pode-se verificar que não houve variação significativa das energias médias dos nós roteadores para os algoritmos avaliados. Assim como na topologia anterior, os algoritmos baseados na teoria dos jogos, a partir das informações que utilizam em suas tomadas de decisão, conseguem produzir um mapa de energia que gera rotas tão eficientes em energia quanto aquelas geradas pelo algoritmo Pred. Isso mostra que, mesmo em um cenário que apresenta uma variação significativa das energias iniciais dos nós sensores, mapas de energias construídos com menos EIPs podem conter as informações necessárias para a geração de rotas eficientes em energia, desde que os EIPs enviados contêm informações relevantes para a constituição das mesmas.

As figuras 5.15 e 5.16 ilustram, respectivamente, o tempo em equilíbrio de Nash e o número total de nós sensores que enviaram seus EIPs e continuaram a rotear. Pode-se observar que o tempo em equilíbrio de Nash dos algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$  aumentou em relação ao tempo obtido na topologia da seção anterior. Além disso, o número de EIPs sem utilidade enviados por esses algoritmos na topologia desta seção diminuiu em relação à topologia da seção anterior. Esses fatos ocorreram porque a topologia desta



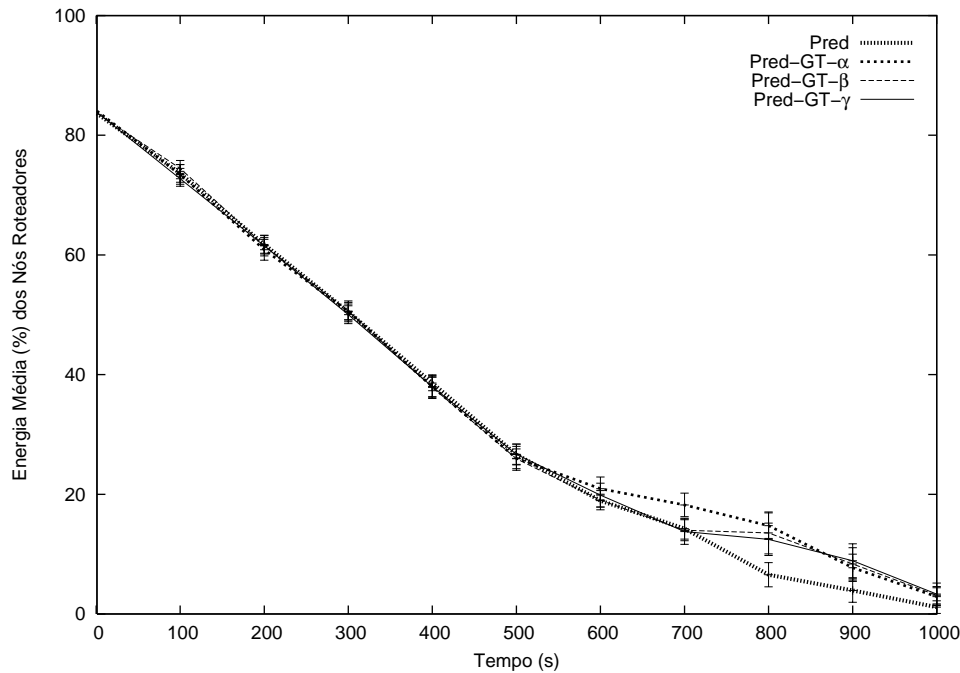


Figura 5.14: Energia média dos nós roteadores.

seção apresenta nós sensores com energias significativamente diferentes, fazendo que as decisões tomadas a partir de pequenas diferenças de energia diminuíssem.

Em relação ao *payoffs* dos nós sensores, pode-se observar a partir das figuras 5.17 e 5.18, que ilustram respectivamente o *payoff* médio por rodada e o *payoff* médio acumulativo dos nós sensores, que a relação entre os valores obtidos pelos algoritmos baseados na teoria dos jogos e os valores obtidos pelo Pred é a mesma da topologia da seção anterior. A única diferença entre os valores é que, na topologia desta seção, os nós sensores morrem antes e assim os *payoffs* dos mesmos se elevam antes que na topologia da seção anterior. Pode-se observar que ao final da simulação, o *payoff* médio dos nós sensores se aproxima de zero, indicando que praticamente não há mais roteamento na rede.

Os resultados encontrados nesta seção não diferem significativamente dos resultados encontrados na seção anterior. As energias médias dos nós roteadores, que poderiam ser diferentes entre si, também se mantiveram semelhantes, da mesma forma que na seção anterior. A diferença mais significativa dos resultados encontrados nesta seção para os resultados da seção anterior é quanto ao tempo que os nós sensores começam a morrer. Enquanto no cenário anterior os nós começavam a morrer por volta dos 800s de simulação, no cenário desta seção os mesmos começavam a morrer por volta dos 600s de simulação. Isso produz um impacto na taxa de envio de EIPs dos algoritmos baseados na teoria dos jogos, que decai após esse tempo e, por conseguinte, aumenta o *payoff* dos mesmos. Além

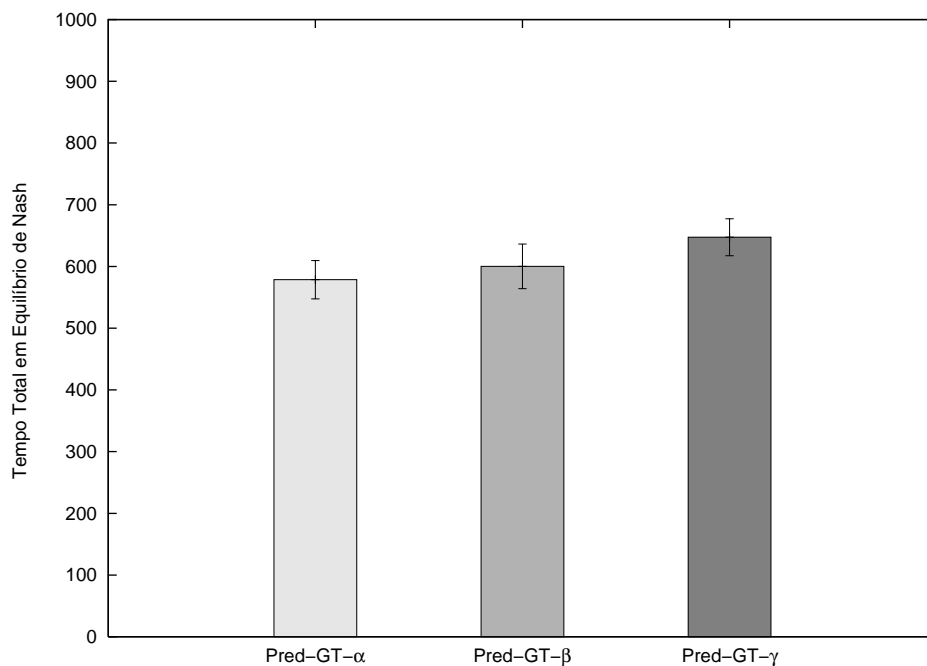


Figura 5.15: Tempo em equilíbrio de Nash.

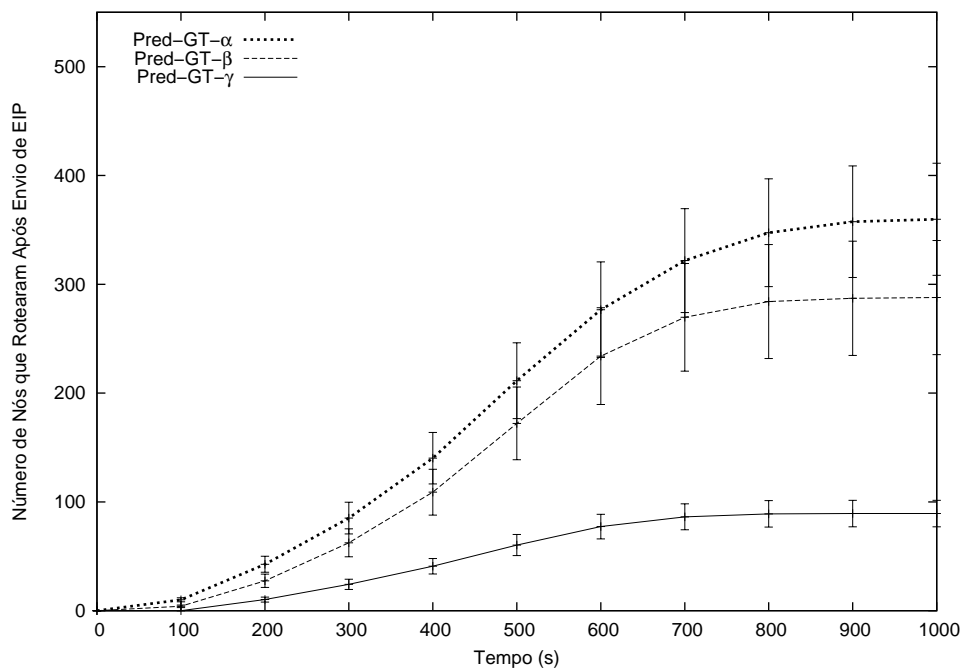


Figura 5.16: Número de nós enviaram EIPs e continuaram a rotar.

disso, na topologia desta seção o Pred enviou mais EIPs que na topologia da seção anterior, uma vez que a energia inicial dos nós sensores se reduziu e, a partir disso, o parâmetro *threshold* foi alcançado mais rapidamente. Por fim, o número de decisões realizadas a

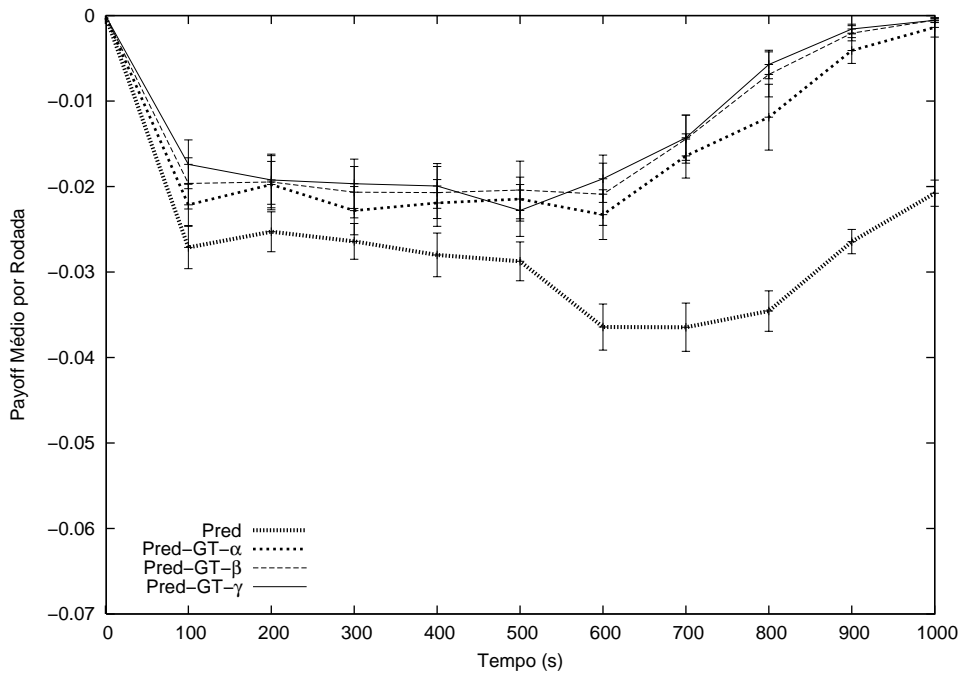


Figura 5.17: *Payoff* médio.

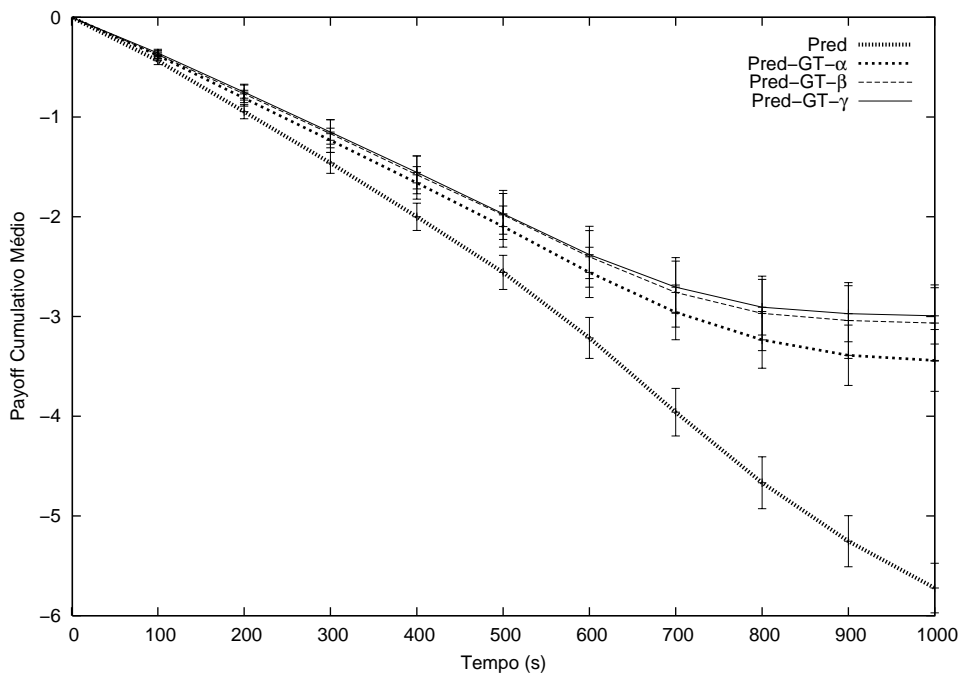


Figura 5.18: *Payoff* cumulativo.

partir de pequenas diferenças de energia diminuiu, fazendo que o tempo em equilíbrio de Nash dos algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$  aumentasse. A tabela 5.2 descreve o resumo dos resultados encontrados na topologia desta seção e na topologia da seção anterior

	Pred		Pred-GT- $\alpha$		Pred-GT- $\beta$		Pred-GT- $\gamma$	
Topologia	Anterior	Corrente	Anterior	Corrente	Anterior	Corrente	Anterior	Corrente
Número de EIPs enviados	4	4, 4, 4	4, 4	4, 4	4, 4	4, 4	4	4, 0
Tempo em Equilíbrio de Nash (s)	-	-	4, 4	4, 4	4, 00	40, 4	4, 4, 0	4, 4
EIPs enviados sem utilidade	-	-	4, 4	4, 4	4, 4	4, 4	4, 4	4, 0
Payoff total	4, 4	4, 4	- 4, 4	- 4, 4	- 4, 4	- 4, 4	- 4, 4	- 4, 4

Tabela 5.2: Comparação entre os resultados obtidos na topologia com distribuição homogênea de energia e a topologia com distribuição aleatória de energia.

A figura 5.19 ilustra a evolução do mapa de energia real da rede e daqueles construídos pelos algoritmos avaliados neste capítulo na topologia desta seção. A diferença entre os mapas de energias gerados pelos algoritmos baseados na teoria dos jogos e os mapas de energias gerados pelo Pred e o real é menor que na topologia da seção anterior. Isso aconteceu porque nesta topologia o roteamento foi mais distribuído, assim como variou mais as energias entre os nós sensores. Desta maneira, uma maior variedade de nós sensores enviaram EIPs e mantiveram o mapa de energia mais próximo do real.

## 5.4 Topologia com uma Região de Baixa Energia

Esta seção apresenta os resultados dos algoritmos baseados na teoria dos jogos e do Pred em uma topologia que possui uma região com baixa reserva de energia no centro da mesma. O mapa de energia é iniciado com a energia real dos nós sensores, já contendo então a informação sobre a região com baixa reserva de energia. Uma vez que o TEDD gera rotas que passam pelos nós sensores com mais energia e evitam aqueles que estão com baixa reserva de energia, é desejável que nenhum nó sensor localizado dentro da região de baixa energia da rede roteie pacotes de disseminação. Para alcançar esse objetivo, o algoritmo TEDD reduz o número de rotas geradas, como pode ser verificado através da figura 5.20, que ilustra exemplos de curvas geradas para a topologia desta seção, e através das figuras 5.3 e 5.12, que ilustram exemplos de curvas geradas para as topologias das seções anteriores. Além disso, da mesma maneira que não devem rotear pacotes de disseminação, os nós sensores localizados dentro da região de baixa energia não devem enviar os seus EIPs pois, como estão com baixa reserva de energia, devem dedicar-se exclusivamente à execução de tarefas mais críticas, como a de sensoriamento.

Apesar de não ser desejável que os nós sensores localizados dentro da região de baixa energia da rede enviem EIPs, os mapas de energia fornecidos ao algoritmo gerador de

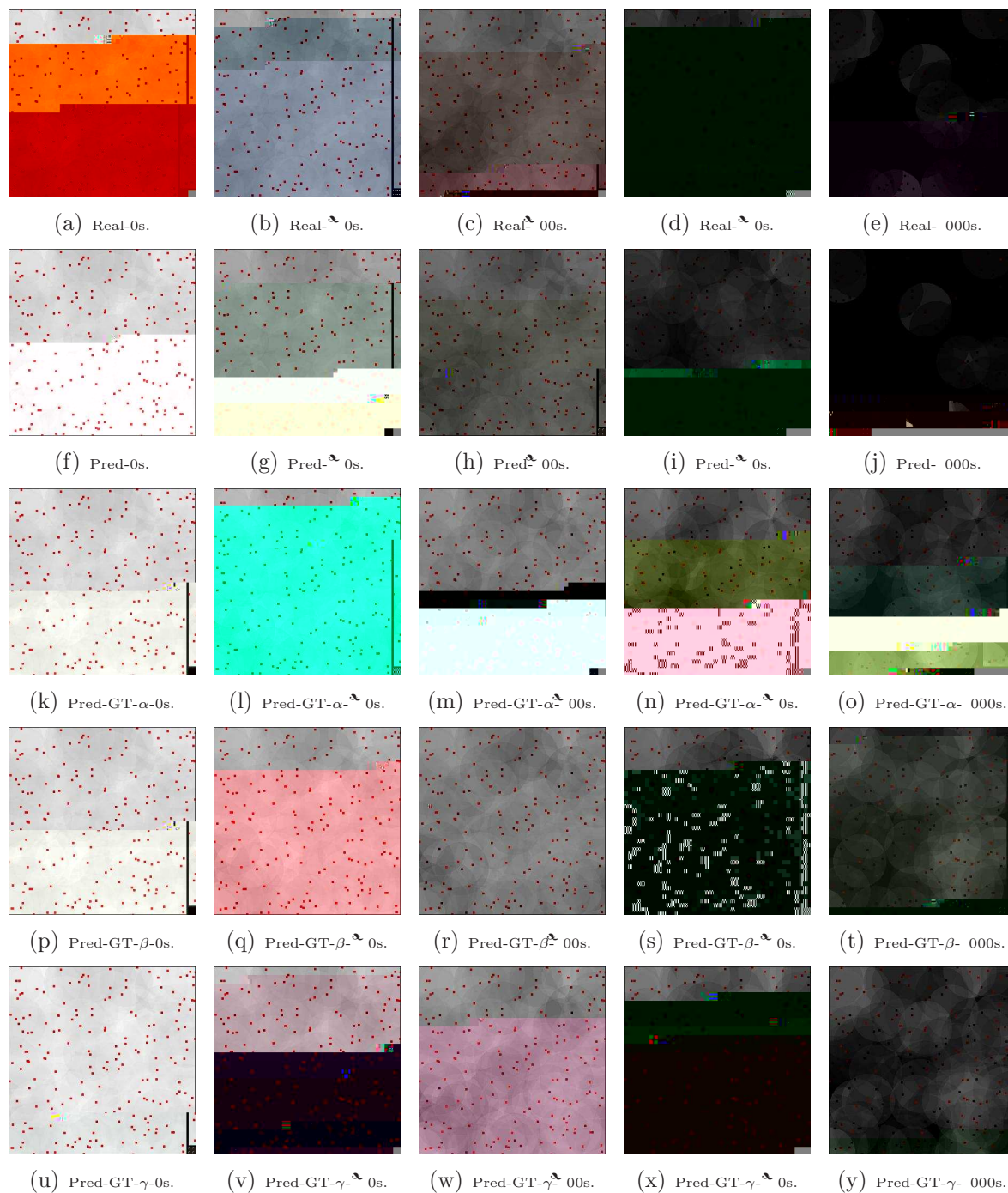
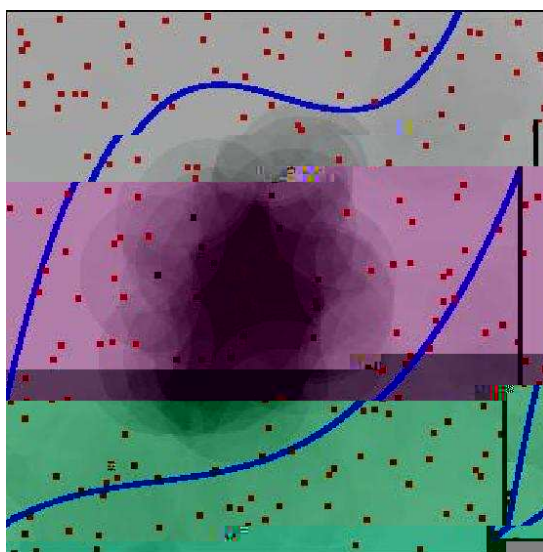


Figura 5.19: A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário com distribuição aleatória de energia.



**Figura 5.20:** Curvas geradas para uma topologia com uma região de baixa energia.

rotas do TEDD devem conter informações suficientes para que os mesmos sejam evitados

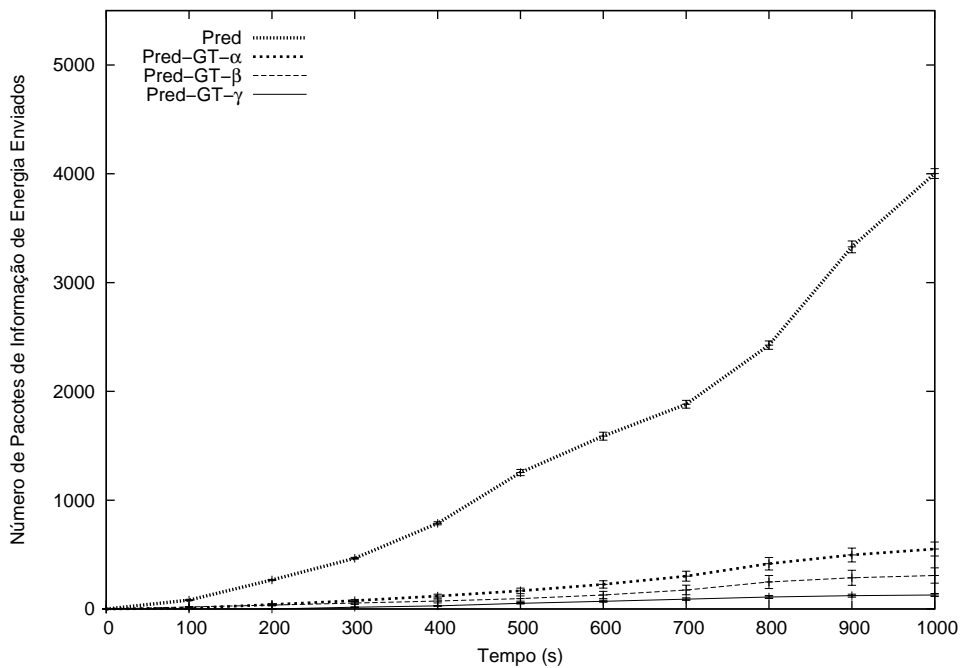


Figura 5.21: Número de EIPs enviados para o nó monitor.

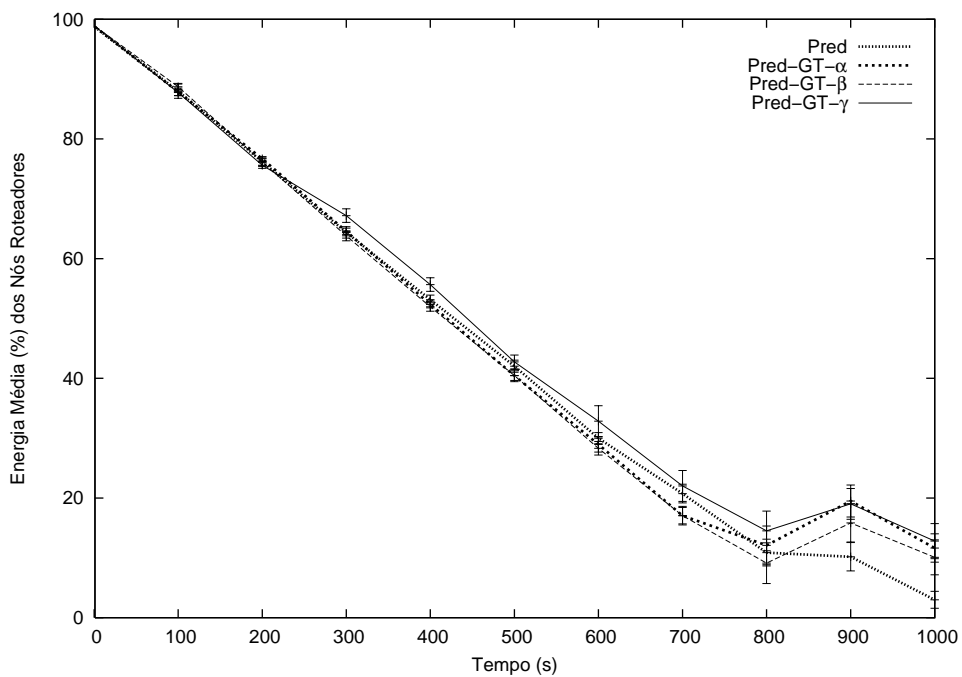
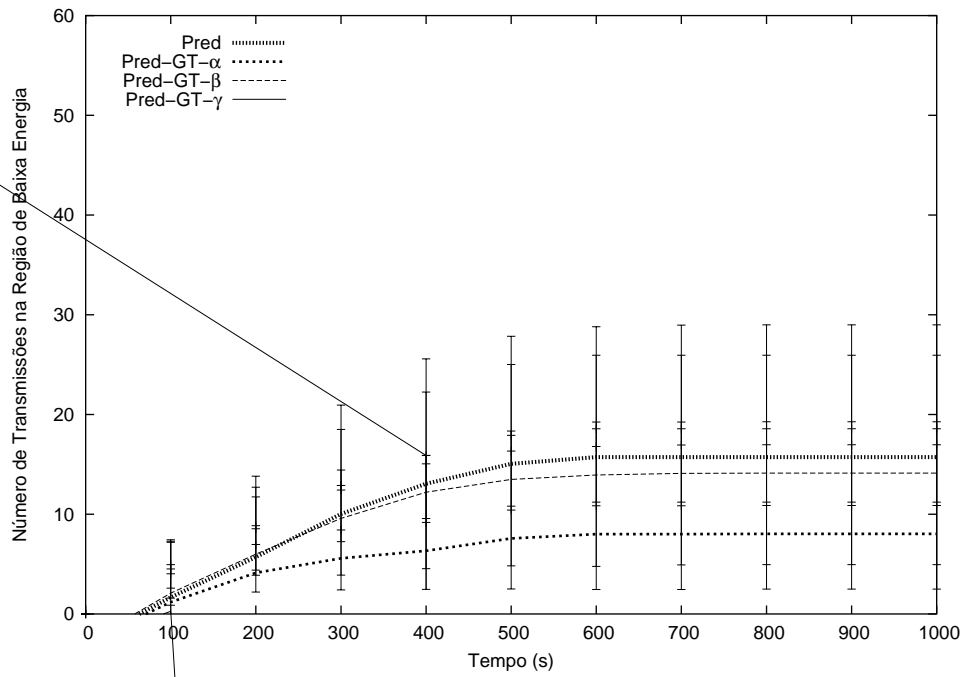
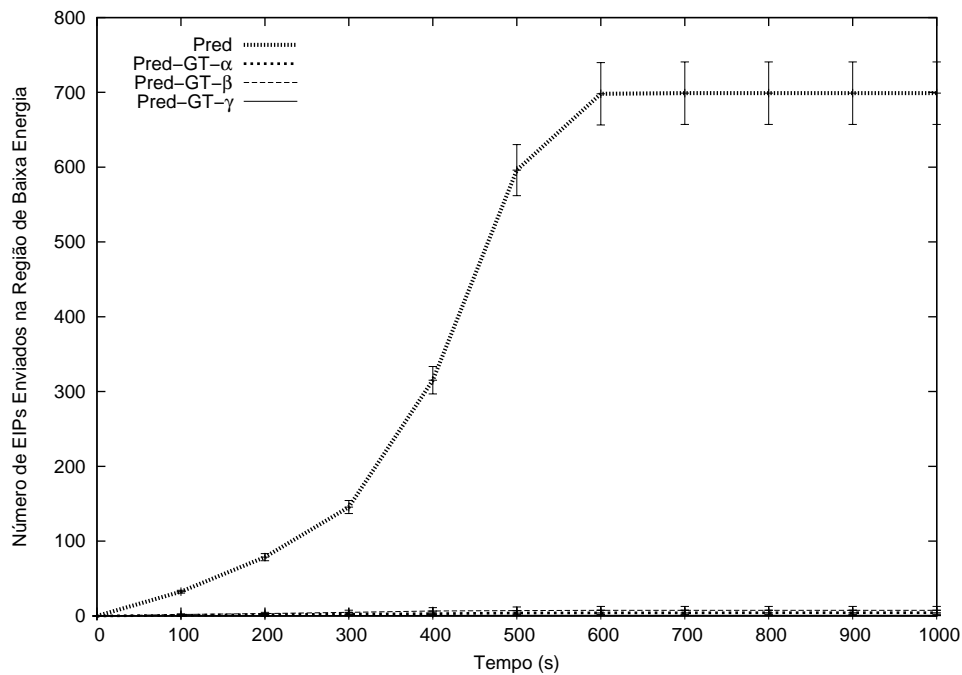


Figura 5.22: Energia média dos nós roteadores.

roteamento. Como mencionado no início desta seção, não é desejável que nós sensores gastem suas reservas de energia com tarefas de controle, como pode ser classificada a tarefa de envio do EIP. Assim, o número de EIPs enviados por nós sensores localizados dentro da







**Figura 5.24:** Número de EIPs enviados dentro da região de baixa energia.

de baixa energia da rede. Por outro lado, o algoritmo Pred condiciona os nós sensores a enviarem os seus EIPs se a diferença entre a energia real dos mesmos e a prevista for maior que o parâmetro *threshold*, fazendo que 100% dos nós sensores localizados na região de baixa energia enviem seus EIPs para o nó monitor, o que não é recomendável.

Para evitar as transmissões dentro da região de baixa energia da rede, o algoritmo TEDD reduz

Devido ao número reduzido de rotas, o número de EIPs enviados para o nó monitor é reduzido. Com isso, também reduz-se a quantidade de nós sensores que não fazem parte do roteamento. Além disso, cresce também o número de nós que fazem parte do roteamento a partir da figura 5.26, que ilustra o tempo de equilíbrio para os algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$ . As diferenças de energia são consideradas pequenas e o tempo total em equilíbrio de Nash é menor que as topologias anteriores. O algoritmo Pred-GT- $\gamma$  não considera as diferenças de energia que desconsidera pequenas diferenças de energia e o equilíbrio de Nash verificado nas topologias anteriores.

Juntamente com o efeito produzido pelo algoritmo TEDD, o efeito produzido pelo algoritmo Pred-GT- $\alpha$  e Pred-GT- $\beta$  é

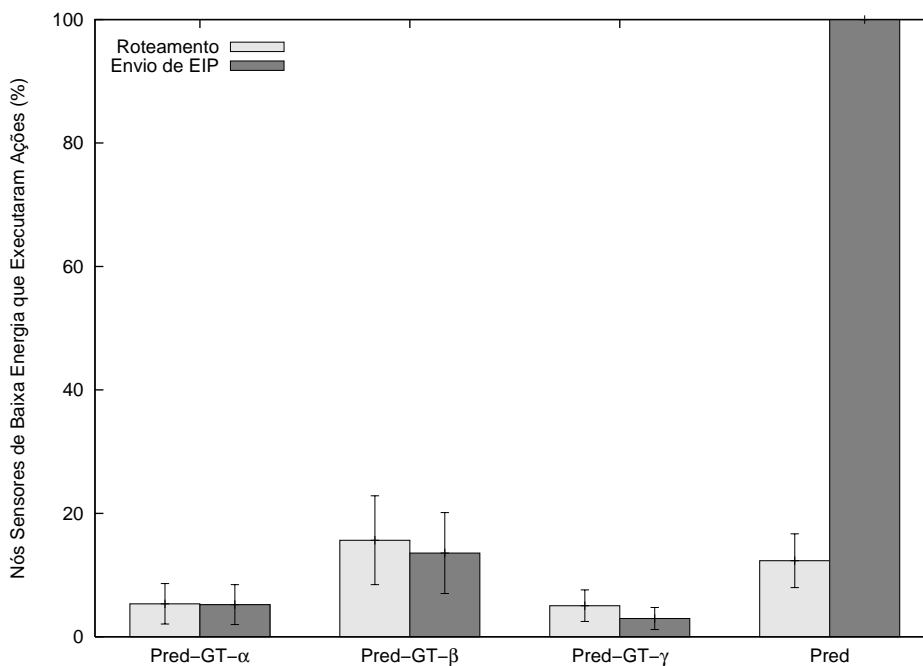


Figura 5.25: Porcentagem de nós sensores localizados na região de baixa energia que executaram ações de roteamento e de envio de EIPs.

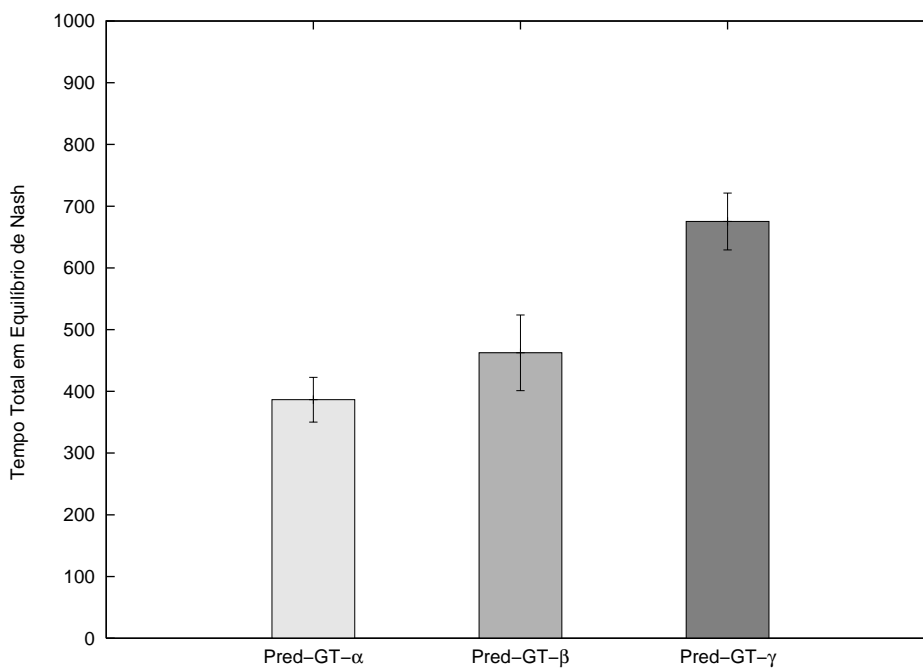
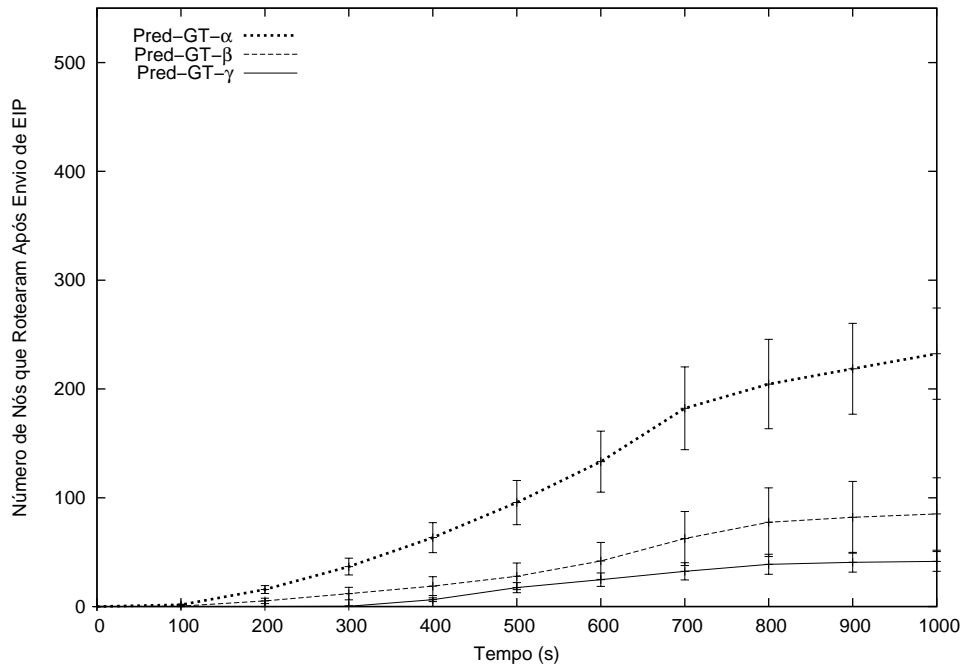


Figura 5.26: Tempo em equilíbrio de Nash.

mento do número de nós sensores aptos a rotear pacotes também altera o número de nós que enviaram os seus EIPs e continuaram a rotear. Por haver mais nós aptos a rotear, as

chances de um EIP alterar a equação matemática gerada pelo TEDD também aumentam. Isso pode ser verificado na figura 5.27, que ilustra o número de nós que enviaram seus EIPs e continuaram a rotear pacotes. Para os três algoritmos analisados, o número de nós que enviaram seus EIPs e continuaram a rotear decaiu significativamente em relação às topologias que não possuem uma região com baixa reserva de energia.



**Figura 5.27:** Número de nós enviaram EIPs e continuaram a rotear.

Por fim, as figuras 5.28 e 5.29 ilustram, respectivamente, o *payoff* médio por rodada e o *payoff* médio acumulativo dos nós sensores durante o tempo de simulação para os quatro algoritmos analisados. Assim como foi verificado para as topologias sem a região de baixa energia, o *payoff* para o algoritmo Pred é significativamente inferior ao *payoff* para os algoritmos baseados na teoria dos jogos. Isso permite concluir que para a topologia com uma região de baixa energia, os algoritmos baseados na teoria dos jogos reduzem o consumo de energia dos nós sensores e prolongam o tempo de vida útil da rede. A tabela 5.3 descreve o resumo dos resultados encontrados na topologia desta seção e na topologia da seção anterior

A figura 5.30 ilustra a evolução do mapa de energia real da rede e daqueles construídos pelos algoritmos avaliados neste capítulo na topologia desta seção. No caso da topologia desta seção, os algoritmos Pred-GT- $\alpha$  e Pred-GT- $\gamma$  foram aqueles que construíram mapas de energias mais diferentes do real. No caso do algoritmo Pred-GT- $\alpha$ , os nós sensores que rotearam pacotes fora da região de baixa energia recebiam pelos pacotes disseminados a

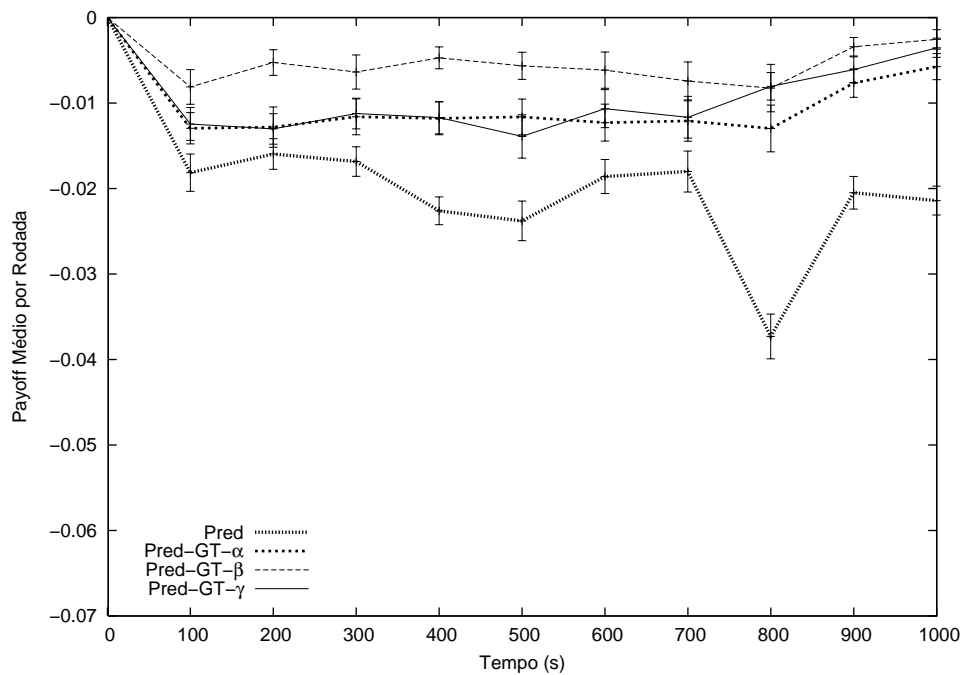


Figura 5.28: *Payoff* médio.

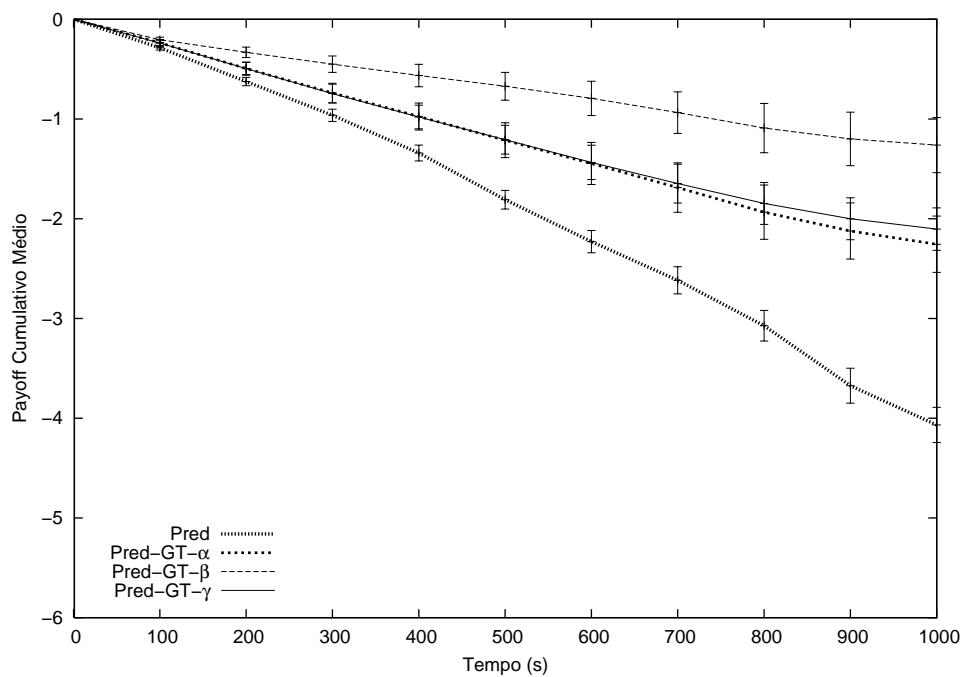


Figura 5.29: *Payoff* cumulativo.

energia de corte da rede. Isso não os incentivava a enviar EIPs, pois esses nós sensores sabiam que estavam entre os nós sensores aptos a rotear na rede. No caso do algoritmo Pred-GT- $\gamma$ , o mapa ficou significativamente diferente do real porque os nós sensores fora



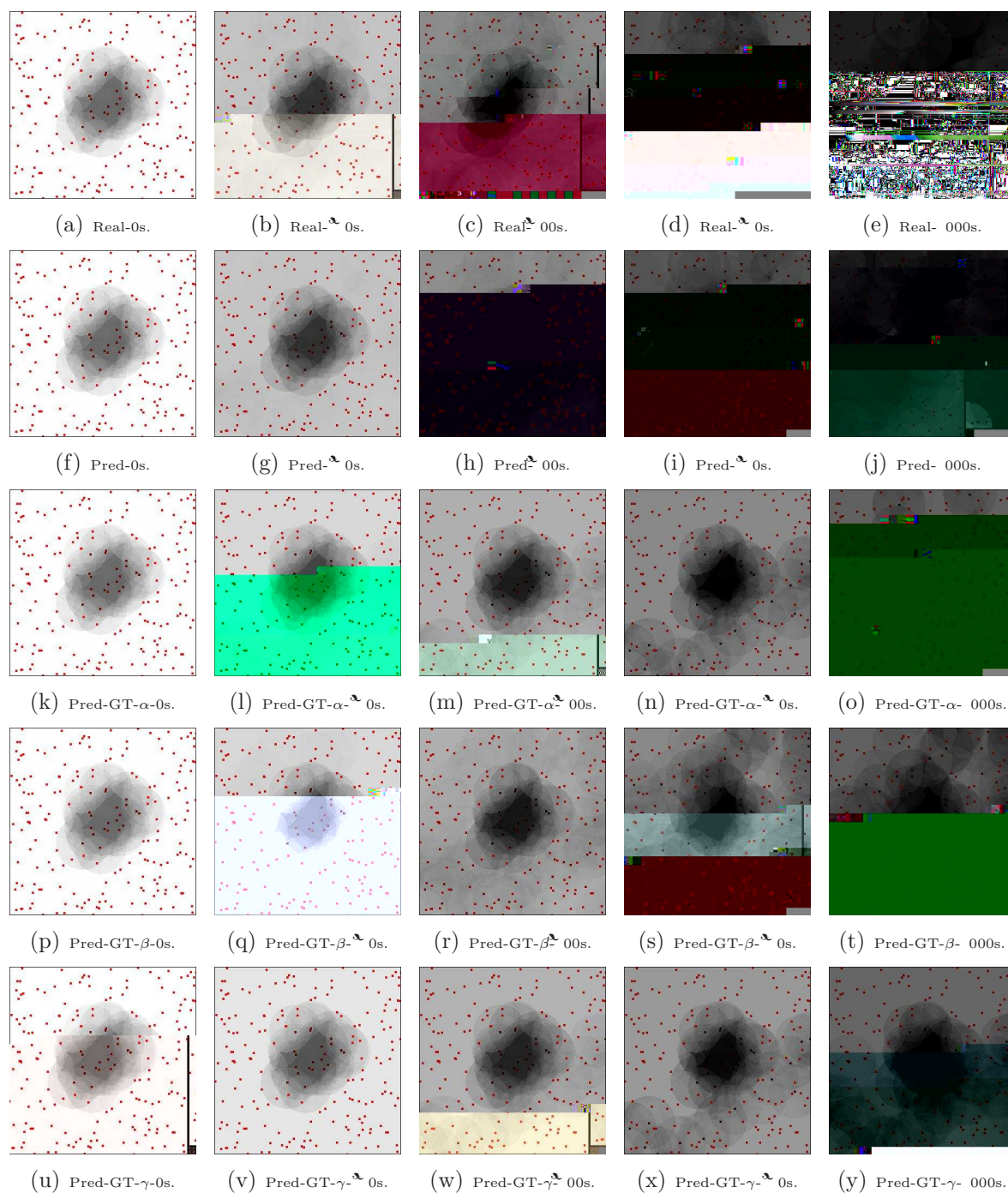


Figura 5.30: A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário com uma região de baixa energia.

possível evitar o roteamento na região de baixa energia da rede à medida que a mesma se forma.

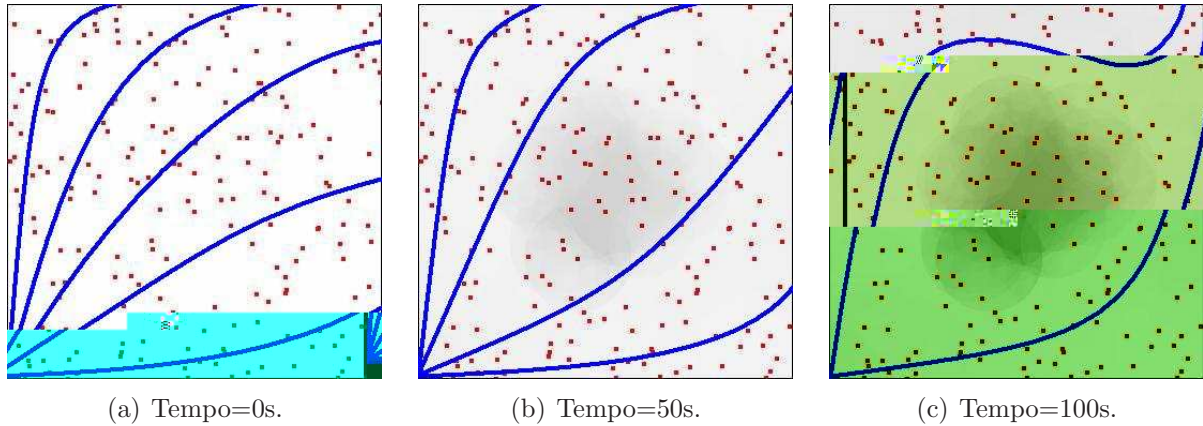


Figura 5.31: Curvas geradas numa topologia que forma uma região com baixa reserva de energia.

Para a topologia que forma a região com baixa reserva de energia em tempo de simulação, a relação do número total de EIPs enviados é a mesma das topologias analisadas anteriormente, como pode ser observado a partir da figura 5.32. O algoritmo Pred envia significativamente mais EIPs que os algoritmos baseados na teoria dos jogos. Através da figura 5.33, pode-se constatar também que a energia média das rotas geradas pelo TEDD é novamente semelhante para os quatro algoritmos analisados, indicando que os algoritmos baseados na teoria dos jogos são capazes de fornecer ao algoritmo de geração de rotas um mapa de energia tão eficiente quanto aquele fornecido pelo Pred.

A equivalência das médias das energias dos nós roteadores demonstra também que as rotas geradas nas simulações dos quatro algoritmos utilizaram um número semelhante de nós sensores situados dentro da região de baixa energia para rotear pacotes. Isso pode ser comprovado pela análise da figura 5.34, que ilustra o número de transmissões realizadas dentro da região de baixa energia. Os quatro algoritmos analisados apresentam resultados semelhantes, indicando que os algoritmos baseados na teoria dos jogos foram capazes de detectar a região com baixa reserva de energia que se formou na rede. O algoritmo Pred-GT- $\gamma$ , por desconsiderar pequenas diferenças de energia, faz com que menos EIPs sejam enviados pelos nós sensores localizados na região com baixa reserva de energia. Isso torna a detecção da região de baixa energia mais lenta, fazendo que mais nós sensores localizados nessa região transmitam pacotes. Além disso, novamente o número de EIPs transmitidos por nós sensores dentro da região com baixa reserva de energia é significativamente inferior nos algoritmos baseados na teoria dos jogos que no Pred, como pode-se verificar a partir da

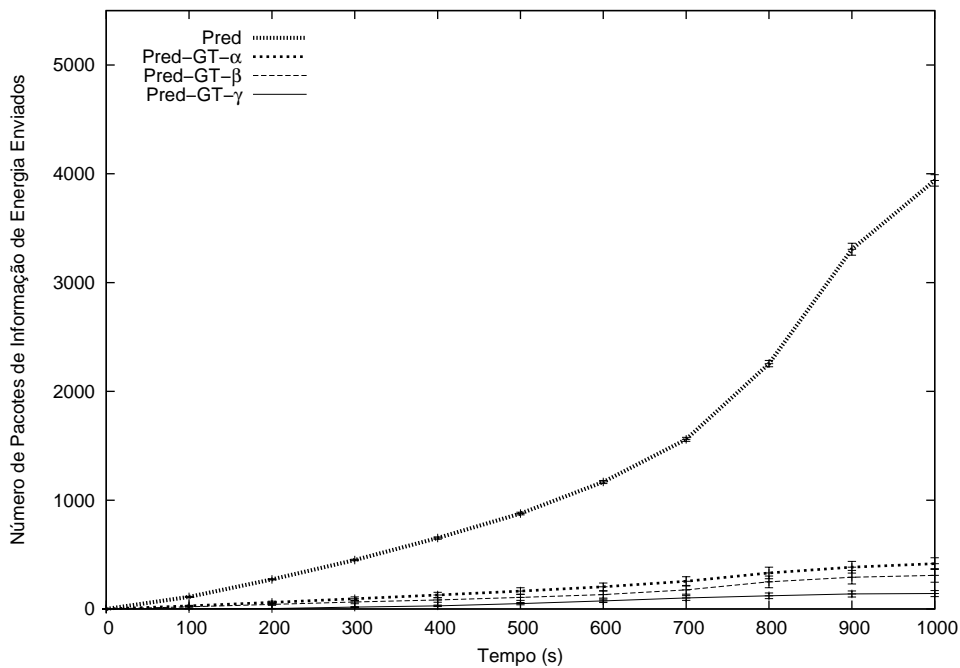


Figura 5.32: Número de EIPs enviados para o nó monitor.

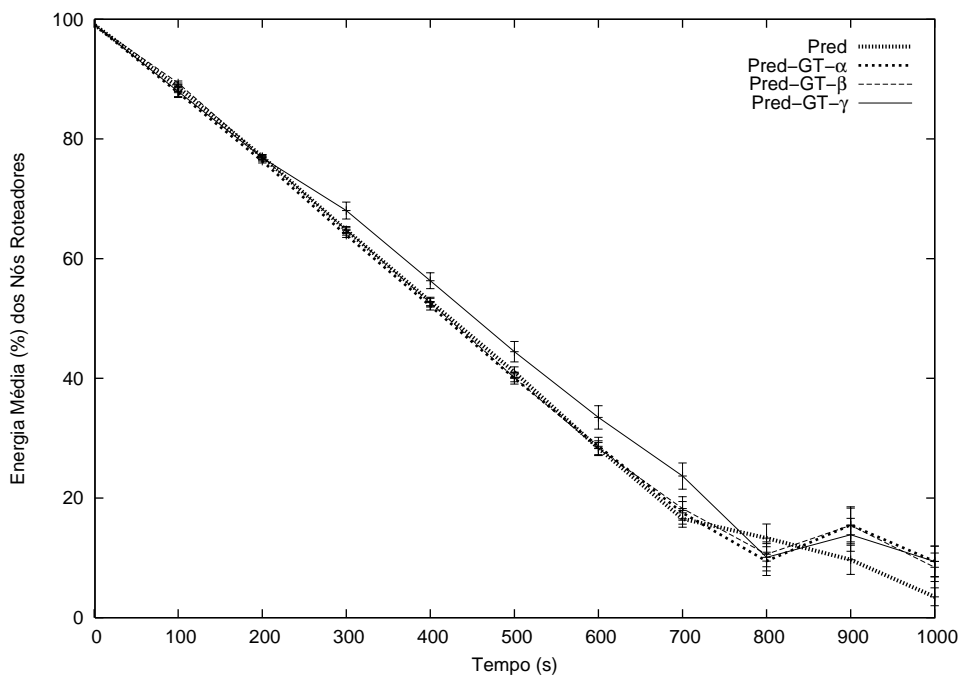
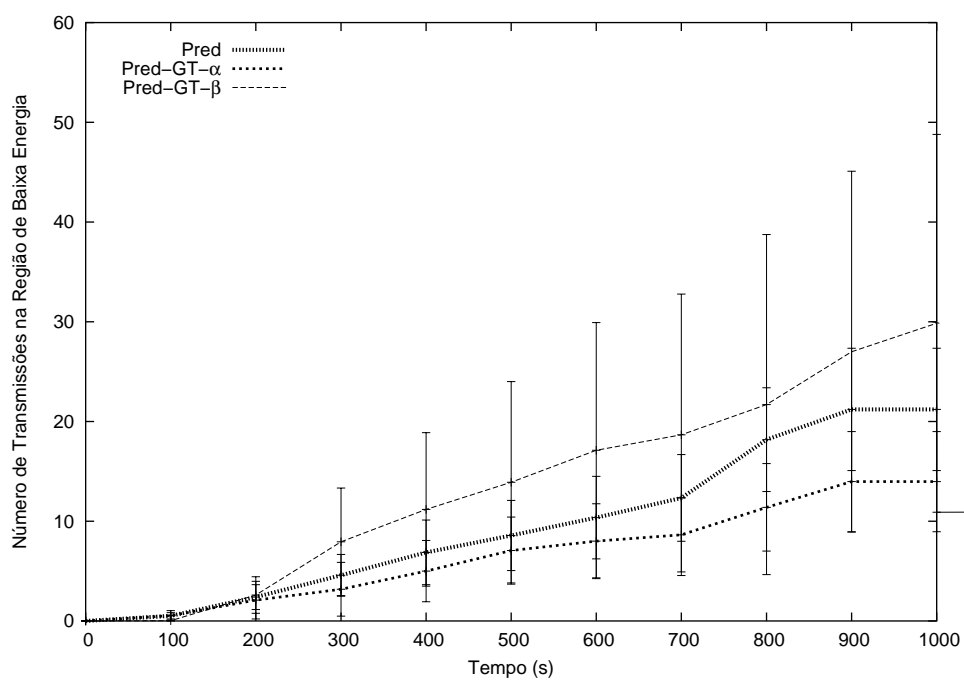


Figura 5.33: Energia média dos nós roteadores.

figura 5.35. É importante ressaltar que um nó sensor só é considerado como pertencente à região com baixa reserva de energia depois da completa formação da mesma, ou seja, depois dos 100 segundos de simulação.





para o algoritmo Pred-GT- $\gamma$  é o maior dentre os algoritmos baseados na teoria dos jogos. Isso acontece devido à dificuldade do mesmo em detectar a região de baixa energia nos primeiros 100 segundos de simulação. Novamente, para o algoritmo Pred-GT- $\alpha$ , a diferença entre a porcentagem de nós localizados na região de baixa energia que rotearam e enviaram EIPs é insignificante, uma vez que esse algoritmo condiciona os nós sensores a enviarem seus EIPs se eles estiverem localizados na região de baixa energia da rede. O algoritmo Pred, por condicionar os nós sensores a enviarem os seus EIPs se a diferença entre a energia real dos mesmos e a prevista for maior que o parâmetro *threshold*, faz novamente que 100% dos nós sensores localizados na região de baixa energia enviem seus EIPs para o nó monitor.

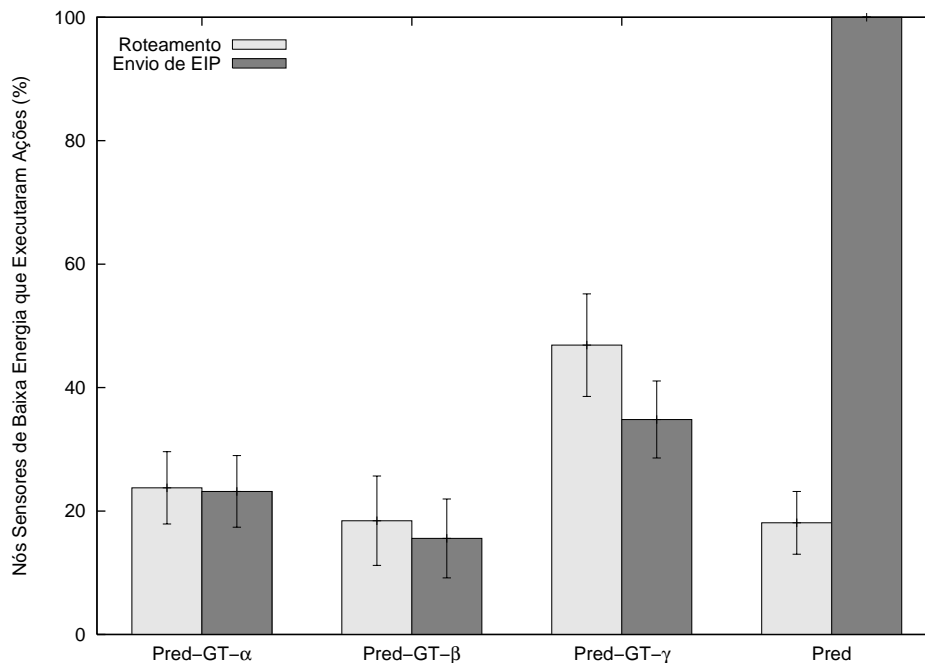


Figura 5.36: Porcentagem de nós sensores localizados na região de baixa energia que executaram ações de roteamento e de envio de EIPs.

As figuras 5.37, 5.38, 5.39 e 5.40 ilustram, respectivamente, o tempo em equilíbrio de Nash, o número total de nós sensores que enviaram seus EIPs e continuaram a rotar, o *payoff* médio por rodada e o *payoff* médio acumulativo dos nós sensores durante o tempo de simulação. Pode-se observar que os resultados foram semelhantes aos resultados observados na topologia da seção anterior, que apresentava uma região com baixa reserva de energia no início da simulação. Isso garante que os algoritmos baseados na teoria dos jogos são capazes de detectar a formação da região com baixa reserva de energia sem um custo de energia significativo adicional. Conclui-se, então, que os algoritmos de

construção e atualização do mapa de energia podem ser usados em topologias que formam rapidamente regiões com baixa reserva de energia. A tabela 5.4 descreve um resumo dos resultados encontrados na topologia desta seção e na topologia da seção anterior

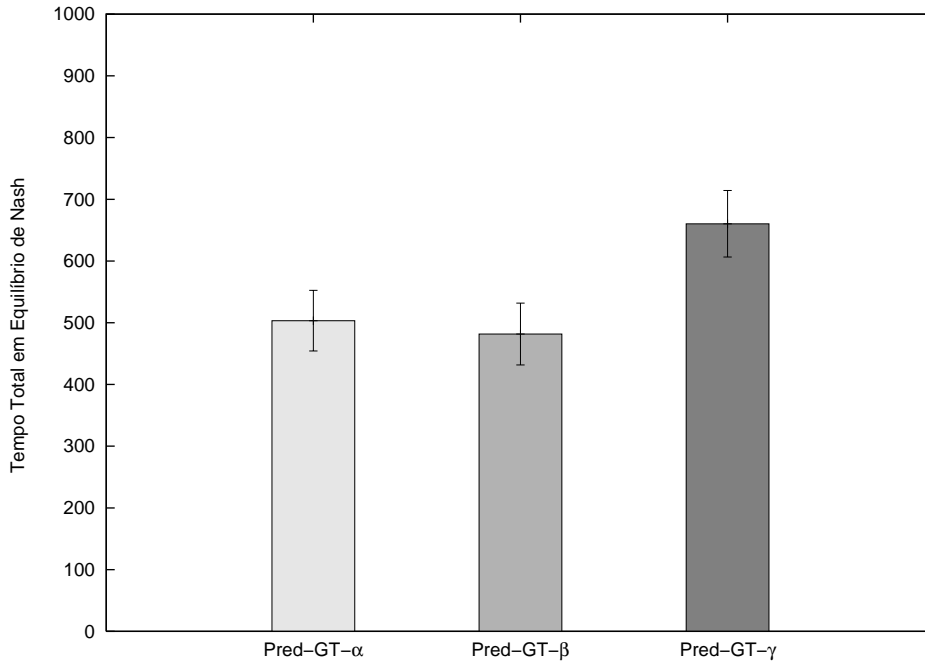


Figura 5.37: Tempo em equilíbrio de Nash.

Topologia	Pred		Pred-GT- $\alpha$		Pred-GT- $\beta$		Pred-GT- $\gamma$	
	Anterior	Corrente	Anterior	Corrente	Anterior	Corrente	Anterior	Corrente
Número de EIPs enviados	4,00	5,5	3,0	4,5	0,5	0,5	5,5	4,5
Tempo em Equilíbrio de Nash (s)	-	-	5	4,5	4,5	4,5	5,5	5,5
EIPs enviados sem utilidade	-	-	4	4,0	2,0	5,5	4,5	5,5
Payoff total	4,0	-5,5	-5,5	-5,0	-5,5	-5,5	-5,0	-4,5
Transmissões na região de baixa energia	5	5	0	5	5,5	5,5	4,5	4,5
EIPs enviados por nós de baixa energia	5,5	4,5	4,5	5,5	5	0	0,5	5,5

Tabela 5.4: Comparação entre os resultados obtidos na com uma região de baixa energia e a topologia que forma uma região de baixa energia.

A figura 5.41 ilustra a evolução do mapa de energia real da rede e daqueles construídos pelos algoritmos avaliados neste capítulo. No caso da topologia desta seção, o algoritmo Pred-GT- $\gamma$  é aquele que apresentou um mapa de energia mais diferente ao real e ao

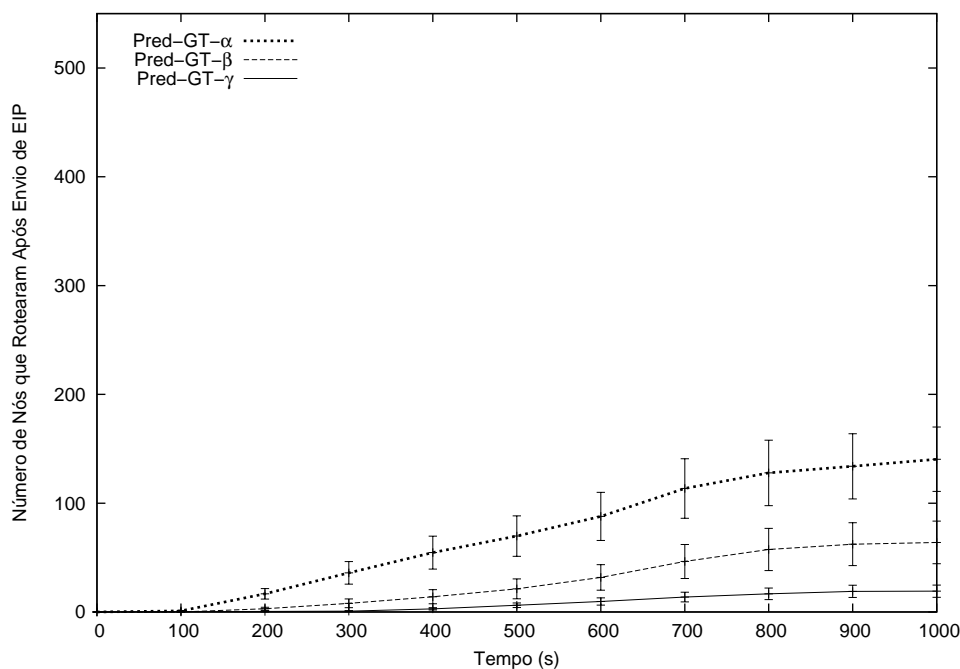


Figura 5.38: Número de nós enviaram EIPs e continuaram a rotear.

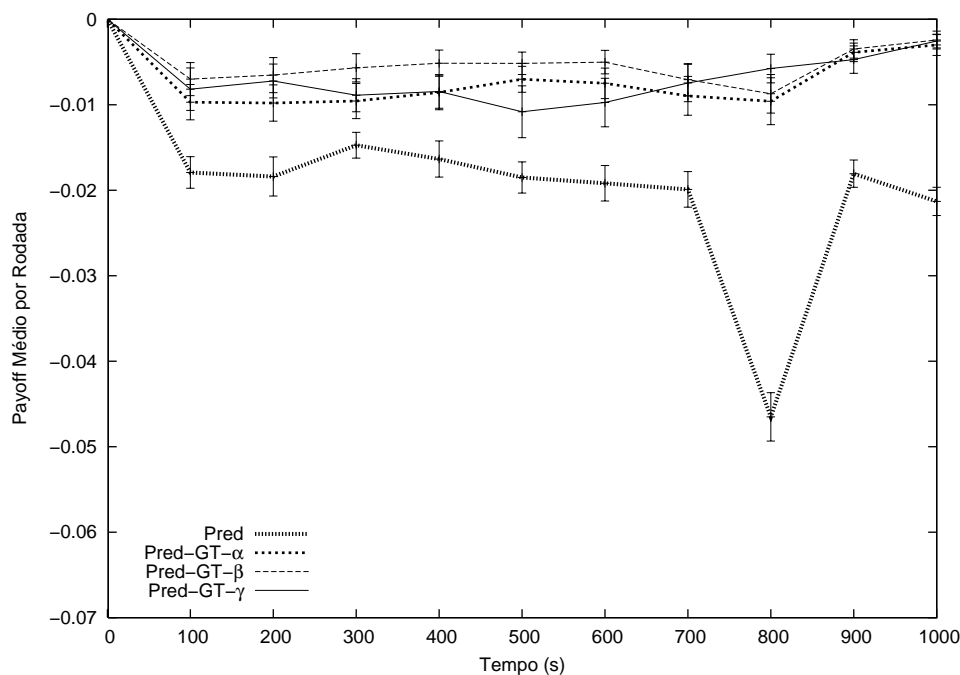


Figura 5.39: *Payoff* médio.

construído pelo Pred nos primeiros 500 segundos de simulação. Isso aconteceu porque todos o algoritmo Pred-GT- $\gamma$  desconsidera pequenas diferenças de energia entre os nós sensores, fazendo que poucos EIPs sejam enviados por nós sensores localizados na região

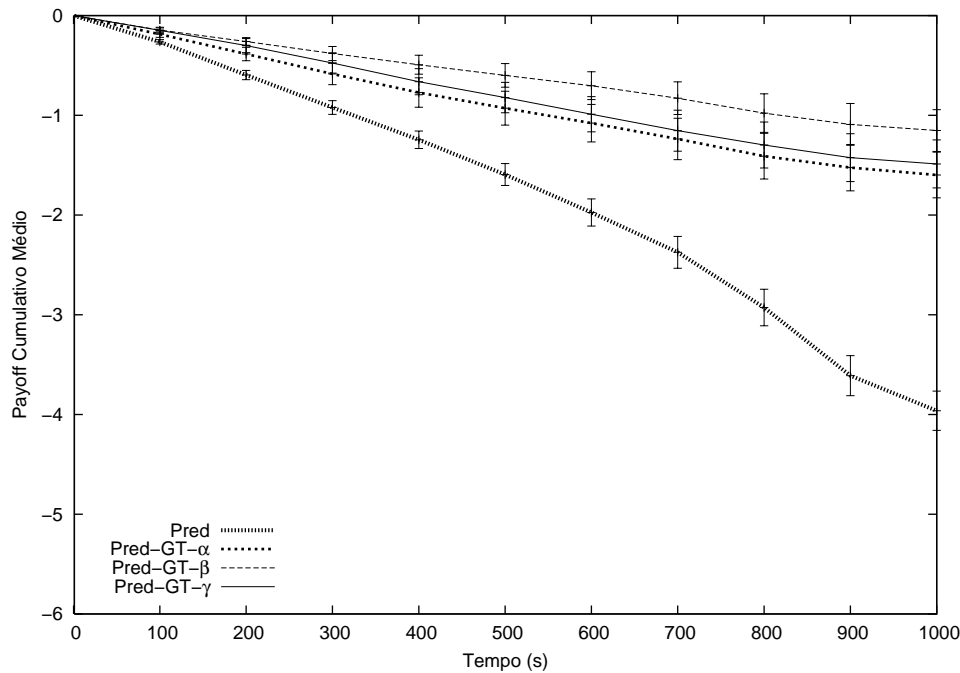


Figura 5.40: *Payoff* cumulativo.

de baixa energia que se formava. Após os 500 segundos de simulação, as diferenças entre as energias dos nós sensores aumentava. Assim, que executavam o algoritmo Pred-GT- $\gamma$  e que roteavam pacotes possuíam mais incentivos para enviar os seus EIPs. Isso fez com que na segunda metade da simulação o mapa de energia construído pelo algoritmo Pred-GT- $\gamma$  se aproximasse dos mapas de energia gerados pelos algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$ .

## 5.6 Conclusões

A partir da análise dos resultados, pode-se concluir que os três algoritmos baseados na teoria dos jogos enviam significativamente menos EIPs que o algoritmo Pred nas quatro topologias analisadas. Isso se deve ao fato dos algoritmos baseados na teoria dos jogos limitarem o envio do EIP aos nós sensores que fizeram parte do processo de roteamento. Além disso, os nós sensores que fizeram parte do roteamento que julgaram não existir nós sensores mais capazes de rotear que eles também não enviam seus EIPs.

Apesar de enviarem significativamente menos EIPs que o Pred, os algoritmos baseados na teoria dos jogos fornecem ao nó monitor mapas de energias com informações suficientes para geração de rotas eficientes em energia. Nas topologias que possuem uma região com baixa reserva de energia, as rotas geradas a partir dos mapas de energia fornecidos pelos algoritmos baseados na teoria dos jogos foram capazes de evitar a região de baixa energia

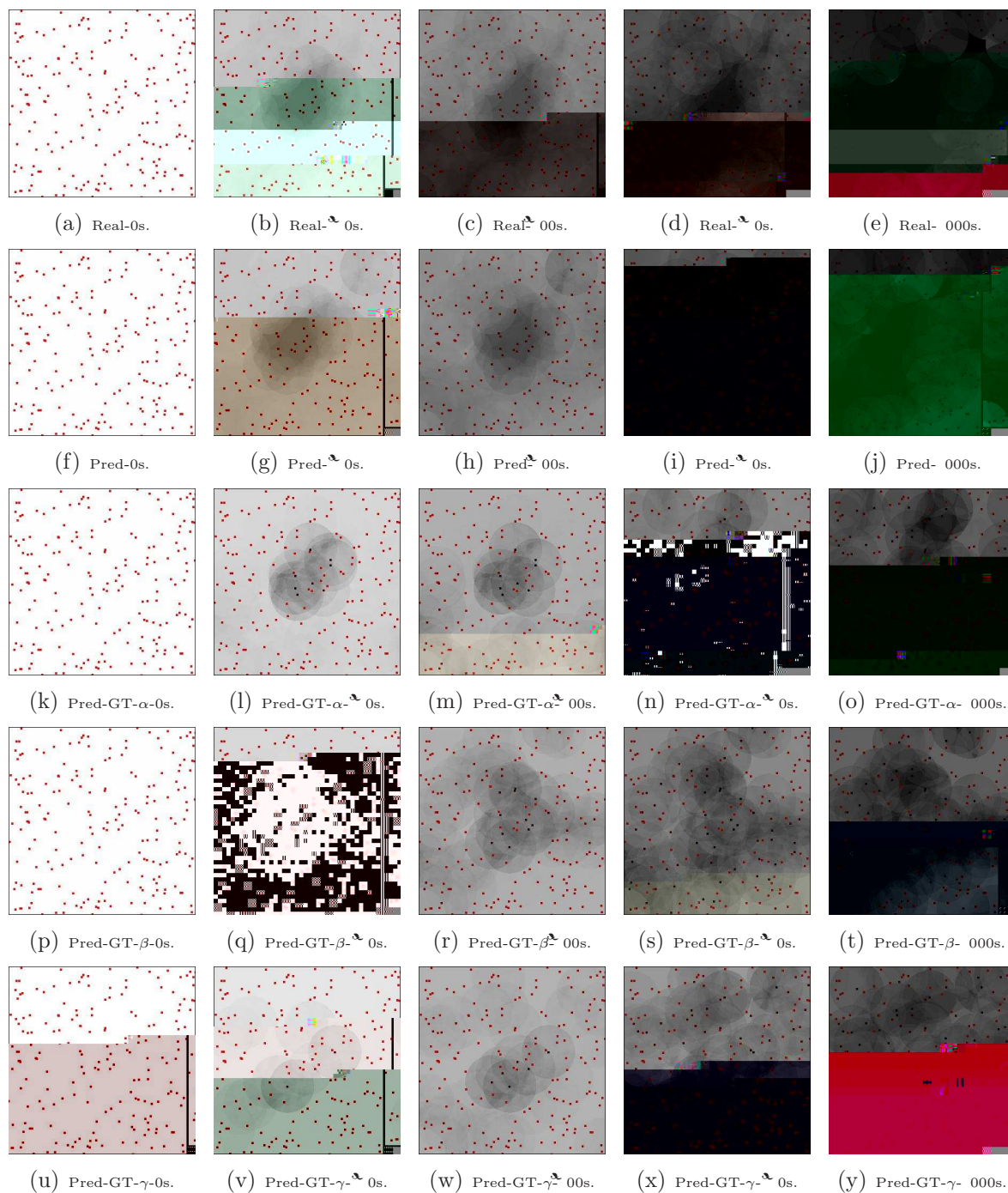


Figura 5.41: A evolução do mapa de energia da rede construído usando cada algoritmo avaliado no cenário que forma uma região de baixa energia na rede em tempo de simulação.

da mesma maneira que as rotas geradas a partir do mapa de energia fornecido pelo algoritmo Pred. É importante ressaltar que as informações contidas nos mapas de energia construídos pelos algoritmos baseados na teoria dos jogos foram suficientes para geração de rotas eficientes em energia, ainda que os mapas de energia sejam significativamente diferentes do real e daquele construído pelo Pred.

Em relação aos três algoritmos baseados na teoria dos jogos, o algoritmo Pred-GT- $\gamma$  foi aquele que apresentou melhores resultados. A classificação da energia dos nós sensores em níveis de energia possibilitou que pequenas diferenças de energia entre os nós sensores fossem desconsideradas no envio do EIP e na geração das rotas. Isso fez com que o número de EIPs enviados pelos nós sensores quando executam o algoritmo Pred-GT- $\gamma$  fosse inferior aos demais algoritmos. Além disso, o tempo em equilíbrio de Nash do algoritmo Pred-GT- $\gamma$  foi sempre maior que o tempo em equilíbrio de Nash dos algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$ , ainda se mantendo constante em todas as topologias. Enquanto isso, o tempo dos algoritmos Pred-GT- $\alpha$  e Pred-GT- $\beta$  se reduziu nas topologias com a região de baixa energia. Por fim, o desempenho dos algoritmos Pred-GT- $\gamma$  e Pred-GT- $\beta$  foi superior ao algoritmo Pred-GT- $\alpha$  no número de EIPs enviados ao nó monitor que não retiraram o nó sensor que o envio do processo de roteamento. Por possuírem informações locais, os algoritmos Pred-GT- $\gamma$  e Pred-GT- $\beta$  dão ao nó sensor uma maior precisão na identificação de nós sensores mais capazes de rotar que ele.

É importante ressaltar que os algoritmos Pred-GT- $\gamma$  e Pred-GT- $\beta$  só devem ser usados quando há comunicação entre nós sensores vizinhos na rede, pois é assim que eles identificam se um nó sensor está sendo usado injustamente como roteador. O algoritmo Pred-GT- $\alpha$ , por sua vez, não necessita que haja comunicação entre nós sensores vizinhos para ser executado, uma vez que o mesmo depende apenas das informações contidas no mapa de energia. No entanto, o algoritmo Pred-GT- $\alpha$  necessita que o algoritmo de roteamento seja um algoritmo de disseminação de dados para que os seus parâmetros sejam atualizados corretamente. É importante ressaltar que, caso não seja possível definir os níveis de energia do algoritmo Pred-GT- $\gamma$ , pode-se definir a diferença de energia entre os nós sensores em termos percentuais.

## Capítulo 6

# Conclusões e Trabalhos Futuros

### 6.1 Conclusões

Este trabalho propôs um modelo baseado nos conceitos da teoria dos jogos para reduzir a comunicação em RSSFs, dado que o custo na comunicação é o principal responsável pelo consumo de energia das mesmas. A teoria dos jogos, por lidar com a otimização de resultados globais a partir de decisões locais, é uma ferramenta apropriada para reduzir a troca de mensagens nas RSSFs. Assim, foi proposto um modelo para construir um mapa de energia para o roteamento em redes de sensores sem fio. O mapa de energia é construído no nó monitor a partir de informações enviadas pelos nós sensores. Entretanto, quando um nó sensor envia a sua informação de energia, ele está executando uma atividade de comunicação de dados. Conseqüentemente, um modelo para construir o mapa de energia precisa enviar um número reduzido de pacotes pois, de outra forma, poderá ser gasta energia suficiente para fazer com que o custo para construir o mapa de energia não compense a economia a partir do seu uso.

Este trabalho utiliza os conceitos da teoria dos jogos para construir um modelo que faça o nó sensor decidir localmente quando deve ou não enviar o seu pacote com as suas informações de energia. A base para a tomada de decisão dos nós sensores é se eles fizeram parte ou não do processo de roteamento. Nós sensores que não fizeram parte do processo de roteamento não devem enviar seus pacotes com as suas informações de energia. Só devem enviar os seus pacotes de atualização do mapa de energia aqueles nós sensores que julgam que isso os retirarão do processo de roteamento. Esse modelo tem como objetivo minimizar o número de pacotes de atualização do mapa de energia enviados. Além disso, o mapa de energia fornecido para o algoritmo de geração de rotas deve conter informações suficientes para que as rotas geradas utilizem apenas nós sensores com alta reserva de energia.



Para avaliar o modelo, foram propostos três algoritmos: Pred-GT- $\alpha$ , Pred-GT- $\beta$  e Pred-GT- $\gamma$ . Resultados de simulação revelaram que os algoritmos propostos reduziram o número de pacotes de informação de energia necessários para construir/atualizar o mapa de energia e ainda mantiveram a sua qualidade. Além disso, os algoritmos Pred-GT- $\beta$  e Pred-GT- $\gamma$ , que usaram informações locais dos nós sensores, obtiveram melhores resultados que o algoritmo Pred-GT- $\alpha$ , que utiliza informações globais da rede. Para avaliar a qualidade do mapa construído, este trabalho analisou o comportamento do algoritmo de roteamento *Trajectory and Energy-based Data Dissemination* (TEDD) [dVMGM<sup>+</sup>05], que usa o mapa de energia para rotear pacotes.

## 6.2 Trabalhos Futuros

Planeja-se estender o modelo de construção do mapa de energia.

## Bibliografia

- [AB02] T. Alpcan and T. Basar. A game-theoretic framework for congestion control in general topology networks. *41st IEEE Conference on Decision and Control*, 2002.
- [ADBA04] Afrand Agah, Sajal K. Das, Kalyan Basu, and Mehran Asadi. Intrusion detection in sensor networks: A non-cooperative game approach. In *NCA*, pages 343–346. IEEE Computer Society, 2004.
- [ADL<sup>+</sup>98] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser, and H. Marcy. Wireless integrated network sensors: Low power systems on a chip. In *European Solid State Circuits Conference*, The Hague, Netherlands, October 1998.
- [adLTdH87] G. an der Laan, A. J. J. Talman, and L. Van der Heyden. Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. *Mathematics of Operations Research*, 12(3):377–398, 1987.
- [ASSC02] Ian F. Akyildiz, W. Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [BAS03] Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. A game theoretic framework for incentives in p2p systems. *CoRR*, cs.GT/0310039, 2003.
- [BB05] Seema Bandyopadhyay and Subhajyoti Bandyopadhyay. A game-theoretic analysis on the conditions of cooperation in a wireless ad hoc network. In *WiOpt*, pages 54–58. IEEE Computer Society, 2005.
- [BS04] Humberto Bortolossi e Polyane Santos e Larissa Barreto Brinda Sartini, Gilmar Garbugio. Uma introdução à teoria dos jogos. *II Bienal da SBM*, 2004.

- [CDR03] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Pricing network edges for heterogeneous selfish users. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 521–530, New York, NY, USA, 2003. ACM Press.
- [CP04] Jacomo Corbo and Thomas Petermann. Selfish peering and routing in the internet. In *Santa Fe Inst. Complex Systems Summer School*, Santa Fe, NM, June 2004.
- [CS02] V. Conitzer and T. Sandholm. Complexity Results about Nash Equilibria. Technical Report CM-CS-02-135, CMU, 2002.
- [dVM05] Max do Val Machado. Disseminação de dados baseada em trajetória e energia para redes de sensores sem fio. Master's thesis, Federal University of Minas Gerais, March 2005.
- [dVMGM<sup>+</sup>05] Max do Val Machado, Olga Goussevskaia, Raquel A.F. Mini, Antonio A.F. Loureiro, Geraldo R. Mateus, and José M. Nogueira. Data dissemination in autonomic wireless sensor networks. In *Special Issue of Autonomic Communication Systems in IEEE Journal on Selected Areas in Communications (JSAC)*, December 2005.
- [EGHK99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *MOBICOM 99*, pages 263–270, USA, 1999.
- [FBH05] M. Felegyhazi, L. Buttyan, and J. P. Hubaux. Cooperative packet forwarding in multi-domain sensor networks. In *Proceedings of IEEE PerSeNS 2005*, Hawaii, USA, March 2005.
- [FT91] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
- [GdVMdFML06] Daniel Ludovico Guidoni, Max do Val Machado, Raquel Aparecida de Freitas Mini, and Antonio A. F. Loureiro. Difusão de dados baseada em atraso e energia para redes de sensores sem fio. In *XXIV Simpósio Brasileiro de Redes de Computadores (SBRC)*, Curitiba, PR, Brasil, 2006.
- [GW03] Srihari Govindan and Robert Wilson. A global newton method to compute nash equilibria. *Journal of Economic Theory*, 110(1):65–86,

2003. available at <http://ideas.repec.org/a/eee/jetheo/v110y2003i1p65-86.html>.
- [IEGH02] Calermek Intanagonwiwat, Deborah Estrin, Ramesh Govindan, and John Heidemann. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, 2002.
- [IGE00] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the sixth annual international conference on Mobile computing and networking*, pages 56–67, Boston, MA USA, 2000.
- [Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley, 1991.
- [KKP99] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In *In Proceedings of MOBICOM*, pages 271–278, Seattle, 1999.
- [KP99] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, Trier, Germany, 4–6 March 1999.
- [KSI04] Rajgopal Kannan, Sudipta Sarangi, and S. Sitharama Iyengar. Sensor-centric energy-constrained reliable query routing for wireless sensor networks. *J. Parallel Distrib. Comput.*, 64(7):839–852, 2004.
- [LA00] Richard J. La and Venkat Anantharam. Optimal routing control: Game theoretic approach. Technical report, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2000.
- [LH64] C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *j-J-SIAM*, 12(2):413–423, June 1964.
- [Mic04] Mica2. Mts/mda sensor and data acquisition boards user’s manual. [www.xbow.com](http://www.xbow.com), 2004.

- [Min04] Raquel A. F. Mini. *Mapa de Energia baseado em Predição para Redes de Sensores Sem Fio*. PhD thesis, Departamento de Ciência da Computação - Universidade Federal de Minas Gerais, Janeiro 2004.
- [MMLN05] Raquel A. F. Mini, Max Machado, Antonio A. F. Loureiro, and Badri Nath. Prediction-based energy map for wireless sensor networks. *Ad Hoc Networks Journal*, 3(2):235–253, 2005.
- [MW01] A.B. MacKenzie and S.B. Wicker. Selfish users in Aloha: A game-theoretic approach. In *Proceedings of the 54th IEEE Vehicular Technology Conference, Atlantic City, NJ, USA*, pages 1354–1357, October 2001.
- [MW03] Allen MacKenzie and Stephen B. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information., 2003.
- [Nas50] John F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.
- [Nas51] John F. Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, September 1951.
- [ns202] ns2. The network simulator. [www.isi.edu/nsnam/ns](http://www.isi.edu/nsnam/ns), 2002.
- [NSK05] M. Singh N. Sadagopan and B. Krishnamachari. Decentralized utility based sensor network design. In *Mobile Networks and Applications Journal (MONET)*, to appear. ACM/Kluwer, 2005.
- [Pap01] Christos Papadimitriou. Algorithms, games, and the internet. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 749–753, New York, NY, USA, 2001. ACM Press.
- [PK00a] G.J. Pottie and W.J. Kaiser. Embedding the internet wireless integrated network sensors. In *Communications of the ACM*, volume 43, pages 51–58, may 2000.
- [PK00b] G.J. Pottie and W.J. Kaiser. Wireless integrated network sensors. In *Communications of the ACM*, volume 43, pages 551–8, may 2000.
- [PNS04] Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a nash equilibrium. In *AAAI*, pages 664–669, 2004.

- [RT02] Tim Roughgarden and Eva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.
- [SdFC91] José Francisco Soares, Alfredo Alves de Farias, and Cibele Comini Cesar. *Introdução à Estatística*. LTC, 1991.
- [SKI02] Sudipta Sarangi, Rajgopal Kannan, and S.S. Iyengar. A simple model of reliable query reporting in sensor networks. Technical Report 2002-06, Department of Economics, Louisiana State University, 2002. available at <http://ideas.repec.org/p/lsu/lsuwpp/2002-06.html>.
- [STZ04] Subhash Suri, Csaba D. Tiş, and Yunhong Zhou. Selfish load balancing and atomic congestion games. In *SPAA '04: Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 188–195, New York, NY, USA, 2004. ACM Press.
- [TvS01] Theodore L. Turocy and Bernhard von Stengel. Game theory. Technical report, Texas A&M University and London School of Economics, 2001.
- [WCE04] Department of Electrical and Computer Engineering, W.A.L.A.N and North Carolina State University Computer Engineering. <http://www.ece.ncsu.edu/wireless/wsn.html>, 2004.
- [YMR00] Haykel Yaiche, Ravi R. Mazumdar, and Catherine Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Trans. Netw.*, 8(5):667–678, 2000.
- [ZGE02] Yonggang Jerry Zhao, Ramesh Govindan, and Deborah Estrin. Residual energy scans for monitoring wireless sensor networks. In *IEEE Wireless Communications and Networking Conference(WCNC'02)*, Orlando, FL, USA, March 2002.
- [ZHL05] Charles Pandana Zhu Han and K. J. Ray Liu. A self-learning repeated game framework for optimizing packet forwarding networks. In *Wireless Communications and Networking Conference*, pages 2131–2136. IEEE, 2005.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)