

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

Pedro Olmo Vaz de Melo<sup>1</sup>, Raquel A.F. Mini<sup>\*1</sup>,  
Max do Val Machado<sup>2</sup>, Antonio A.F. Loureiro<sup>2</sup>

<sup>1</sup>Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais  
Belo Horizonte, MG, 30535-901 – Brasil

<sup>2</sup>Departamento de Ciência da Computação – Universidade Federal de Minas Gerais  
Belo Horizonte, MG, 31270-901 – Brasil

pedroolmo@gmail.com, raquelmini@pucminas.br, {maxm,loureiro}@dcc.ufmg.br

**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. One way to reach this goal is to reduce the communication between nodes, once communication is the main responsible for the energy consumption. This work proposes the game theory as a tool for reducing the communication in wireless sensor networks, once it deals with the optimization of a global result from local decisions. A game theoretic model to build an energy map for wireless sensor networks is proposed and analyzed. Simulation results reveal that the use of this model reduces significantly the number of packets required to build the energy map, maintaining its quality.*

**Resumo.** *Um importante desafio no projeto de redes de sensores sem fio é desenvolver técnicas que façam o uso eficiente da sua energia e, assim, estendam o seu tempo de vida. Uma maneira de fazer isso é reduzindo a comunicação entre nós, uma vez que a comunicação é o principal responsável pelo consumo da energia. Este trabalho propõe a teoria dos jogos como ferramenta de redução da comunicação em redes de sensores sem fio, uma vez que a ela lida com a otimização de um resultado global a partir de decisões locais. Um modelo baseado na teoria dos jogos para construir um mapa de energia para redes de sensores sem fio é proposto e analisado. Resultados de simulação revelam que o uso desse modelo reduz significativamente a quantidade de pacotes requeridos para construir o mapa de energia, mantendo a sua qualidade.*

### 1. Introdução

As Redes de Sensores Sem Fio (RSSFs) [Akyildiz et al. 2002] são um ambiente de pesquisa desafiador caracterizado por nós sensores com restrições de memória, processamento e energia. Como esses nós são equipados por baterias com capacidade finita e a sua reposição é raramente possível, o desenvolvimento de protocolos que otimizem o consumo de energia dos nós é o principal tópico de pesquisa em RSSFs. Nesse contexto, a atividade de comunicação de dados é uma das maiores preocupações

---

\*Projeto CNPq número 481750/2004-5.

no projeto dessas redes porque o seu custo de energia é o mais significativo. Segundo [Pottie and Kaiser 2000], 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 [Fudenberg and Tirole 1991] lida com a otimização de resultados globais a partir de tomadas de decisões locais, ela se torna uma abordagem favorável para 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” [Fudenberg and Tirole 1991]. 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.

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. O paradigma “*sensor-centric*” afirma que para maximizar a utilização e a viabilização de 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 [Kannan et al. 2004]. 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.

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 a informação sobre a quantidade de energia disponível nas diversas partes da rede e ele é fundamental para estender o tempo de vida da rede [Mini et al. 2005]. Esse mapa é construído no nó monitor (*sink*) a partir de informações de energia que são enviadas pelos nós sensores. O objetivo deste trabalho é usar a teoria dos jogos para decidir quando realizar a comunicação de dados, que neste caso são informações necessárias para construir o mapa de energia. Nós sensores vão receber um melhor *payoff* enquanto eles economizam 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. Resultados de simulação revelam que o uso da teoria dos jogos reduz o número de pacotes necessários para atualizar o mapa de energia e ainda mantém a sua qualidade. A fim de avaliar a qualidade do mapa, este trabalho analisa o comportamento de um protocolo de disseminação de dados que usa o mapa de energia para rotear pacotes.

O restante deste trabalho está organizado da seguinte maneira. A seção 2 apresenta os trabalhos relacionados. O modelo e os algoritmos que usam teoria dos jogos para construir o mapa de energia se encontram na seção 3. Na seção 4 são analisados os resultados experimentais. Finalmente, na seção 5, são mostradas as conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

A presente seção discute a construção do mapa de energia e problemas das RSSFs que foram resolvidos a partir dos conceitos da teoria dos jogos.

### 2.1. Mapa de Energia

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 o seu nível 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 [Mini et al. 2005], 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, 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 [Mini et al. 2005], esse modelo é eficiente em energia.

Dentre as aplicações do mapa de energia, uma das mais importantes é referente ao seu uso no roteamento. O mapa de energia é usado como parâmetro de entrada para algoritmos de geração de rotas que, a partir dele, selecionam por onde o roteamento deve ocorrer. O algoritmo de disseminação de dados *Trajectory and Energy-based Data Dis-*

*semination* (TEDD) [do Val Machado et al. 2005] usa as informações do mapa de energia para gerar equações de curvas que representam as rotas pelas quais os pacotes de disseminação devem trafegar. As curvas passam por regiões com alta reserva de energia e evitam regiões com baixa reserva de energia. A figura 1 ilustra um exemplo de curvas de roteamento num mapa de energia, sendo que quanto mais escura a cor, menos energia ela representa. Outro algoritmo que usa o mapa de energia é o *Delay and Energy Based Broadcasting* (DEBB) [Guidoni et al. 2006], que a partir dele calcula a energia de corte dos nós que não devem transmitir mensagens de difusão. Um algoritmo de geração de rotas a partir do 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 cada nó, nesse caso, não se faz necessária.

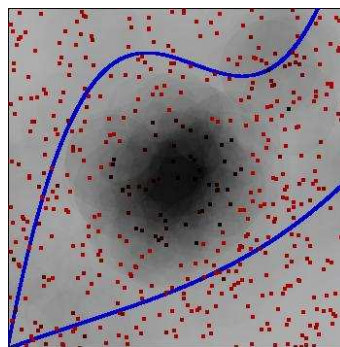


Figura 1. Curvas de roteamento no mapa de energia.

## 2.2. Modelos Baseados em Teoria dos Jogos

Na literatura, várias modelagens de problemas de RSSFs a partir de uma abordagem que usa teoria dos jogos foram propostas. Invariavelmente, o objetivo dessas modelagens é fazer com que o caráter egoísta dos jogadores, que normalmente são modelados como os nós sensores, não prejudique o funcionamento da rede e ainda economize a sua energia.

No problema de roteamento, [Bandyopadhyay and Bandyopadhyay 2005] propôs um mecanismo de punição para nós sensores que não desejam cooperar e encaminhar pacotes de outros nós. Em [Zhu Han and Liu 2005], a fim de impedir que nós sensores não encaminhem pacotes, conceitos de jogos repetidos são introduzidos, permitindo que estratégias sejam definidas a partir de jogadas anteriores, tornando possível efeitos de reputação e reembolso, o que contribui para cooperação do nó sensor. Em [Agah et al. 2004], é proposto um *framework* de segurança para RSSF. O *framework* proposto é basicamente um sistema que seleciona *clusters* da rede para defender. Por fim, em [Felegyhazi et al. 2005], 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 os seus pacotes até um nó monitor, economizando assim a energia da primeira.

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*. Na próxima seção, será apresentado um modelo baseado em teoria dos jogos para construção do mapa de energia em RSSFs.

### 3. Teoria dos Jogos na Construção do Mapa de Energia

#### 3.1. 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 dos nós sensores e é usado como parâmetro de entrada de algoritmos que geram rotas. As rotas geradas precisam ser eficientes em energia e devem usar os nós com as maiores reservas 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 realizado até que o tempo de vida da RSSF expire e é dividido em rodadas. 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.

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 rotar pacotes no processo de roteamento de dados. O valor de  $cEIP$  é menor que o valor de  $cRouter$ , pois  $cEIP$  é o custo de transmitir um pacote e  $cRouter$  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

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

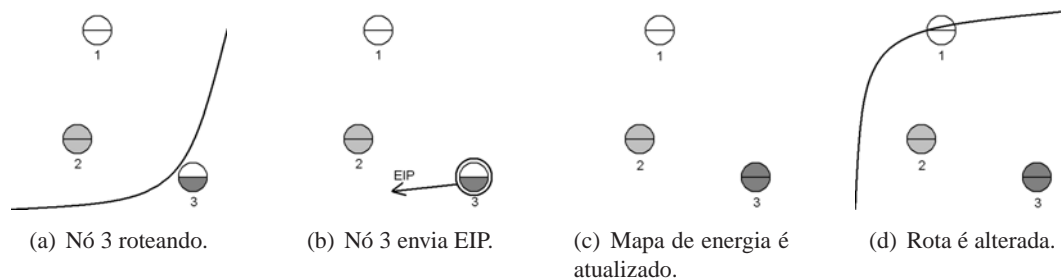
em que  $nEIP_{it}$  é 1 caso o nó sensor  $i$  tenha enviado o seu EIP durante a rodada  $t$  do jogo e 0 caso contrário. O valor de  $nRouter_{it}$  é o número de vezes que o nó sensor  $i$  foi usado para rotar pacotes de dados durante a rodada  $t$  do jogo. Assim, o *payoff* máximo que um jogador pode obter é 0, quando ele não envia o seu EIP e nem participa do processo de roteamento. O objetivo de cada nó sensor no jogo é maximizar o seu *payoff* total.

Depois de definir o jogador, as suas estratégias e as métricas para o seu *payoff*, é preciso definir as condições para a escolha das estratégias. Uma estratégia só deve ser escolhida se ela fornecer ao nó sensor um *payoff* maior. Uma vez que os piores *payoffs* são pagos aos nós que fizeram parte do processo de roteamento, a escolha da estratégia deve ser de forma que os nós sensores evitem participar do processo de roteamento todas as vezes que for possível. Assim, os nós escolherão as suas estratégias considerando as chances que eles têm de serem escolhidos para participar do processo de roteamento.

A escolha dos nós sensores que farão parte do processo de roteamento é feita a partir das informações contidas no mapa de energia. O protocolo de roteamento lê a informação contida neste mapa e constrói rotas que usam os nós mais capazes de rotar pacotes. Um determinado nó sensor é considerado mais capaz de rotar pacotes que outro se o mesmo possuir uma reserva de energia maior. Assim, caso um nó que está sendo usado no processo de roteamento julgue que existam outros nós sensores mais capazes

de rotear pacotes, ele deve enviar o seu EIP para o nó monitor. Esse nó sensor considera que, assim que o nó monitor receber a sua informação e atualizar o mapa de energia, ele não será escolhido para fazer parte do processo de roteamento novamente. Da mesma maneira, um nó que considera que o envio do seu EIP não evitará que o mesmo continue a fazer parte do processo de roteamento não possui incentivos para gastar a sua energia enviando um pacote para o nó monitor.

A figura 2 ilustra o comportamento esperado de um sensor e do protocolo de roteamento quando um nó com baixa reserva de energia é escolhido para fazer parte da rota. Cada nó sensor é ilustrado utilizando duas cores. A cor verificada na parte superior do nó representa a sua quantidade de energia no mapa de energia e a verificada na parte inferior, a sua quantidade real de energia. Quanto mais escura a cor, menor a quantidade de energia representada. Na figura 2-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 2-b. Por sua vez, o nó monitor, como ilustrado na figura 2-c, atualiza o mapa de energia e gera uma nova rota, que não passa mais pelo nó 3 (figura 2-d).



**Figura 2. Comportamento do algoritmo de roteamento que usa as informações do mapa de energia.**

Para verificar a aplicabilidade do modelo descrito acima, o algoritmo *Pred-GT* foi desenvolvido. O algoritmo tem como base o algoritmo para construção do mapa de energia baseado em previsão descrito em [Mini et al. 2005], em que um EIP só deve ser enviado caso a diferença entre a energia real do nó sensor e a energia prevista no mapa de energia for maior que determinado limiar (*threshold*). Além disso, o *Pred-GT* usa os *payoffs* anteriores e as informações de energia dos nós sensores para decidir se eles enviam ou não os seus EIPs. O algoritmo *Pred-GT* é descrito na seção 3.2. Na seção 3.3 será explicado o conceito do equilíbrio de Nash e a sua aplicação no modelo desta seção.

## 3.2. O Algoritmo Pred-GT

### 3.2.1. Algoritmo Base

O algoritmo *Pred-GT* utiliza as informações mantidas pelos nós sensores para decidir se eles enviam ou não os seus EIPs. Um EIP só deve ser enviado se for pra retirar o nó do processo de roteamento. O envio de um EIP só pode retirar um determinado nó do processo de roteamento caso a atualização do mapa de energia a partir desse EIP revele que há outros nós mais capazes de rotear pacotes que ele. Assim, um nó sensor deve executar o algoritmo 1 para decidir se o mesmo envia ou não o seu EIP.

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

---

**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

```

---

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 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 3.2.2 descreve o algoritmo Pred-GT- $\alpha$ , que utiliza informações disseminadas do mapa de energia na tomada de decisão. A seção 3.2.3 descreve o algoritmo Pred-GT- $\beta$ , que utiliza as informações de energia dos nós vizinhos na tomada de decisão. Por fim, a seção 3.2.4 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.

**3.2.2. 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 baseado em cinco atributos que o nó sensor mantém: (1)  $avg$ , a energia média; (2)  $stdDev$ , o desvio padrão da energia de toda a 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. Assim, 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 boa probabilidade (84% dos valores de uma distribuição normal são maiores que a média subtraída do desvio padrão) de haver nós sensores mais aptos a rotear pacotes que ele. 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.

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. Esses valores só poderão ser mantidos pelos nós sensores quando o processo de roteamento for uma disseminação de dados. Assim, após o cálculo, os valores são embutidos nos pacotes que forem disseminados por ele. O valor inicial de *avg* e *cutEne* é 0 e será atualizado apenas por nós que fizerem parte do processo de roteamento.

### 3.2.3. 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 (nós localizados a um *hop* de distância) 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: e, highestNeighEnergy
2:
3: if e < highestNeighEnergy then
4:   return true
5: else
6:   return false
7: end if

```

---

A função *energyMustBeUpdated()* necessita dos valores *e*, que é a energia do nó sensor, e *highestNeighEnergy*, que é o maior valor de energia de algum vizinho do nó sensor. Se *e* for menor que *highestNeighEnergy*, então há outro nó sensor mais capacitado para rotear pacotes e o seu EIP deve ser enviado. O parâmetro de entrada *highestNeighEnergy* é atualizado a partir de mensagens que o nó sensor escuta dos seus vizinhos. Como toda mensagem contém o valor da energia do nó que a encaminhou, o nó sensor atualiza o parâmetro *highestNeighEnergy* toda vez que escuta um valor maior que o armazenado. Além disso, a partir de predição, ele mantém o valor de



*highestNeighEnergy* atualizado, considerando que os parâmetros de consumo de energia do seu vizinho são semelhantes aos seus. É importante ressaltar que o processo de roteamento não necessita ser uma disseminação de dados, uma vez que o nó sensor necessita apenas das informações dos seus vizinhos.

### 3.2.4. Utilizando Níveis de Energia (Pred-GT- $\gamma$ )

A terceira versão do Pred-GT também utiliza informações de energia dos nós sensores vizinhos, com a diferença que a quantidade de energia de um nó sensor é classificada em níveis de energia. O objetivo dessa classificação é evitar que pequenas diferenças de energia entre nós sensores vizinhos os levem a enviar EIPs. Os níveis de energia são definidos a partir de intervalos discretos 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  $iEne$ .

O mapa de energia é então representado não mais pela energia dos nós sensores, mas sim pelos níveis de energia dos mesmos. 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 nas suas proximidades. 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. 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.

---

#### 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

```

---

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

### 3.3. Equilíbrio de Nash na Construção do Mapa de Energia

Na teoria dos jogos, um dos conceitos mais importantes é o equilíbrio de Nash [Fudenberg and Tirole 1991]. 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. No modelo proposto, o equilíbrio de Nash pode ser detectado quando todos os nós usados no processo de roteamento são aqueles com a maior quantidade de energia dentre os seus vizinhos.

Para detectar o equilíbrio de Nash no modelo proposto, deve-se analisar os *payoff* 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. 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  $-cEIP$  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 fazer parte do roteamento novamente e teriam um *payoff* pior, uma vez que  $cEIP < cRouter$ . Quando o *payoff* pago a um nó for  $-cRouter$ , existem dois casos. Quando a sua energia for maior ou equivalente a dos seus vizinhos, o seu *payoff* é o melhor possível, pois se ele tivesse optado por alterar a estratégia e enviar o seu EIP, nada aconteceria, uma vez que a energia dele continuaria sendo maior que a dos seus vizinhos e a rota continuaria a passar por ele, com a diferença que o seu *payoff* passaria de  $-cRouter$  para  $-cEIP - cRouter$ . Quando a sua energia for menor que a dos seus vizinhos, provavelmente a sua informação de energia está desatualizada no mapa de energia e, por isso, ele foi usado como roteador. Se ele tivesse enviado o seu EIP na rodada anterior, ele provavelmente não faria mais parte do processo de roteamento e obteria um *payoff* melhor. Por fim, se um nó sensor recebeu  $-cEIP - cRouter$  como *payoff*, ele poderia ter recebido um *payoff* melhor caso tivesse alterado sua estratégia, ou seja, não tivesse enviado o seu EIP. 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  $-cRouter$ , somente para os nós que fizeram parte do roteamento e que

possuem as suas quantidades de energia maiores ou equivalentes ao dos seus vizinhos. Isso garante que quando o jogo estiver em equilíbrio de Nash, as rotas utilizarão apenas nós com as maiores quantidades de energia dentre os seus vizinhos para rotear pacotes. É uma vez que os nós roteadores são aqueles com maior quantidade 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.

#### 4. Resultados de Simulação

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 300 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  $50 \times 50 m^2$ . Cada nó sensor possui, 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) [do Val Machado et al. 2005]. Os parâmetros de consumo de energia do nó sensor são baseados no Mica2 [Mica2 2004]. Os gastos em energia com o envio e o roteamento do EIP não são considerados.

Os algoritmos propostos para construir o mapa de energia foram implementados no simulador *Network Simulator 2.30* [ns2 2002] e são comparados entre si e com o algoritmo baseado em predição descrito em [Mini et al. 2005] 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 [Jain 1991]. Em todos os experimentos, são usados níveis de confiança de 95%. Além disso, o *T-test* [Jain 1991] com 0.05 de significância é usado para afirmar de maneira significativa que um protocolo é melhor, pior ou semelhante a outro.

A figura 3 ilustra o comportamento do número total de EIPs enviados durante o tempo de simulação. Uma vez que o algoritmo Pred só limita o envio do EIP a partir da diferença da energia do nó sensor e da prevista no mapa, é de se esperar que o número total de EIPs enviados seja significativamente maior que o dos demais algoritmos. Isso mostra que os algoritmos propostos reduzem a comunicação na rede. Nota-se também que à medida que a energia dos nós decaem, a taxa de envio de EIPs a partir do algoritmo Pred aumenta, uma vez que o parâmetro limiar do Pred é definido em termos percentuais. Nos demais algoritmos, o inverso ocorre, pois à medida que nós sensores morrem, menos nós roteam pacotes e assim menos nós possuem incentivos para enviar EIPs. Os algoritmos Pred-GT- $\beta$  e Pred-GT- $\gamma$  ainda enviam menos EIPs que o algoritmo Pred-GT- $\alpha$ , pois eles mantêm informações mais precisas que o Pred-GT- $\alpha$ .

O mapa de energia deve informar quais nós estão aptos a rotear pacotes e quais não estão. A figura 4 mostra o energia média dos nós sensores que rotearam pacotes durante

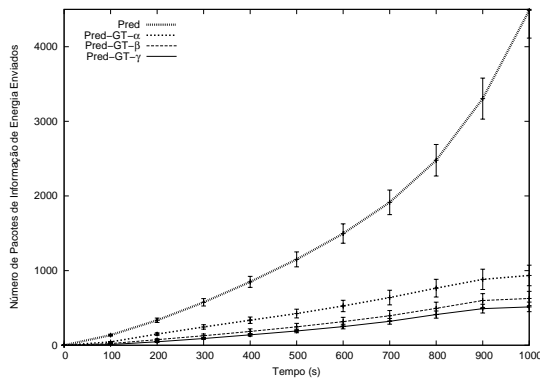


Figura 3. Número de EIPs enviados para o nó monitor.

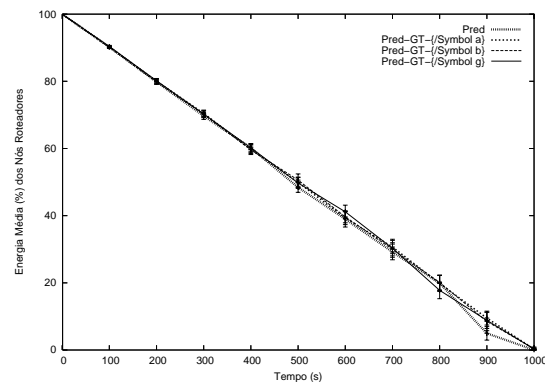


Figura 4. Energia média dos nós roteadores.

o tempo de vida da rede. Como todos os algoritmos analisados apresentam um desempenho similar, pode-se afirmar que 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, ou seja, os algoritmos baseados em teoria dos jogos são capazes de informar quais nós estão aptos a rotear pacotes e quais não estão.

A figura 5 mostra o tempo de total que o jogo esteve no estado de equilíbrio de Nash, ou seja, o tempo que as rotas usaram apenas nós sensores com a maior quantidade de energia dentre os seus vizinhos. O algoritmo Pred-GT- $\gamma$  é o que apresenta maior tempo em equilíbrio de Nash, pois a classificação dos nós segundo níveis de energia faz com que pequenas diferenças de energia sejam desconsideradas. Isso garante uma maior estabilidade, pois um nó tem menos incentivos para enviar EIPs quando existe uma pequena diferença de energia. O algoritmo pred-GT- $\beta$ , por ter acesso a informações mais precisas, apresenta o mesmo desempenho do algoritmo pred-GT- $\alpha$  mesmo enviando menos EIPs.

A figura 6 ilustra o número total de vezes que um nó sensor enviou um EIP e continuou a rotear pacotes na rodada seguinte do jogo. Um EIP que não retira o nó sensor que o enviou do processo do roteamento é considerado um gasto inútil de energia e deve ser evitado sempre que possível. 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.

Por fim, as figuras 7 e 8 mostram, respectivamente, as médias por rodada e cumulativa dos *payoffs* dos nós sensores durante o tempo de simulação. Apesar de não ser baseado em conceitos da teoria dos jogos, pode-se usar o *payoff* no algoritmo Pred como métrica de consumo de energia dos nós sensores, pois quanto mais um nó sensor gasta energia, menor será o seu *payoff*. Isso significa que o consumo de energia da rede é menor nos três algoritmos propostos por este trabalho que no Pred, levando a crer que o modelo de construção do mapa de energia a partir dos conceitos da teoria dos jogos reduz o número de comunicações e economiza a energia da rede, estendendo o seu tempo de vida.

## 5. 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

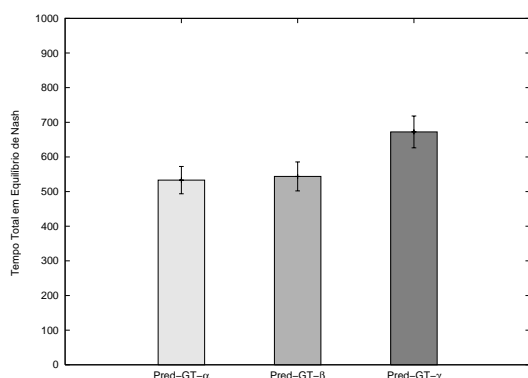


Figura 5. Tempo em equilíbrio de Nash.

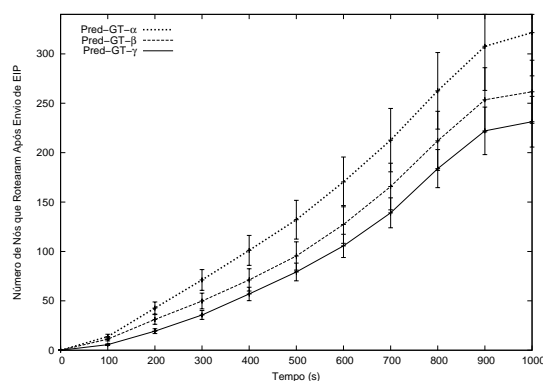


Figura 6. Número de nós enviaram EIPs e continuaram a rotar.

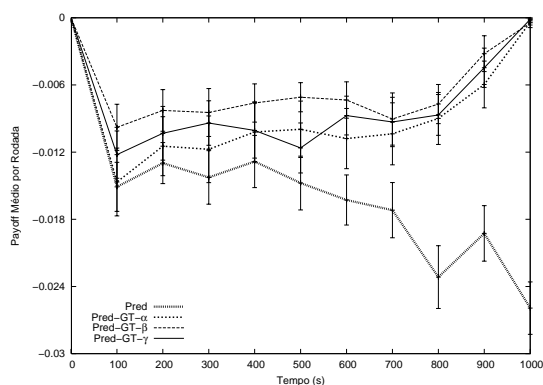


Figura 7. Payoff médio.

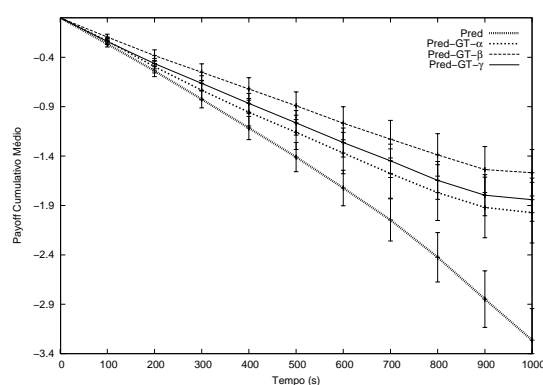


Figura 8. Payoff cumulativo.

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 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 decidir localmente quando deve ou não enviar o seu pacote com as suas informações de energia. Esse modelo tem como objetivo minimizar o número de pacotes de atualização do mapa de energia enviados e maximizar o valor da energia dos nós que foram usados no roteamento. 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) [do Val Machado et al. 2005], que usa o mapa de energia para rotar pacotes.

Planeja-se aplicar o modelo deste trabalho para outras tomadas de decisão em outras aplicações que não sejam algoritmos de roteamento. Além disso, pretende-se estudar o comportamento dos três algoritmos descritos neste trabalho atuando em conjunto em ambientes reais. Por fim, é importante ressaltar que a teoria dos jogos pode ser usada em outros problemas que envolvam tomada de decisão em RSSFs.

## Referências

- Agah, A., Das, S. K., Basu, K., and Asadi, M. (2004). Intrusion detection in sensor networks: A non-cooperative game approach. In *NCA*. IEEE Computer Society.
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422.
- Bandyopadhyay, S. and Bandyopadhyay, S. (2005). A game-theoretic analysis on the conditions of cooperation in a wireless ad hoc network. In *WiOpt*, pages 54–58. IEEE Computer Society.
- do Val Machado, M., Goussevskaia, O., Mini, R. A., Loureiro, A. A., Mateus, G. R., and Nogueira, J. M. (2005). Data dissemination in autonomic wireless sensor networks. In *Special Issue of Autonomic Communication Systems in IEEE Journal on Selected Areas in Communications (JSAC)*.
- Felegyhazi, M., Buttyan, L., and Hubaux, J. P. (2005). Cooperative packet forwarding in multi-domain sensor networks. In *Proceedings of IEEE PerSeNS 2005*, Hawaii, USA.
- Fudenberg, D. and Tirole, J. (1991). *Game Theory*. MIT Press.
- Guidoni, D. L., do Val Machado, M., de Freitas Mini, R. A., and Loureiro, A. A. F. (2006). 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.
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation and modeling*. John Wiley.
- Kannan, R., Sarangi, S., and Iyengar, S. S. (2004). Sensor-centric energy-constrained reliable query routing for wireless sensor networks. *J. Parallel Distrib. Comput.*, 64(7):839–852.
- Mica2 (2004). Mts/mda sensor and data acquisition boards user's manual. [www.xbow.com](http://www.xbow.com).
- Mini, R. A. F., Machado, M., Loureiro, A. A. F., and Nath, B. (2005). Prediction-based energy map for wireless sensor networks. *Ad Hoc Networks Journal*, 3(2):235–253.
- ns2 (2002). The network simulator. [www.isi.edu/nsnam/ns](http://www.isi.edu/nsnam/ns).
- Pottie, G. and Kaiser, W. (2000). Embedding the internet wireless integrated network sensors. In *Communications of the ACM*, volume 43, pages 51–58.
- Zhu Han, C. P. and Liu, K. J. R. (2005). A self-learning repeated game framework for optimizing packet forwarding networks. In *Wireless Communications and Networking Conference*, pages 2131–2136. IEEE.