

# Um Protocolo de Roteamento Sensível ao Tempo para Redes de Sensores Sem Fio

João B. Borges Neto, Pedro F. Ribeiro Neto, Rossana M. C. Andrade

**Resumo**—Avanços nas Redes de Sensores Sem Fio (RSSF) permitem sua utilização para diversos fins. Missões espaciais e explorações subaquáticas são exemplos de ambientes desafiadores que se beneficiam com sua utilização. Aplicações nestes ambientes compartilham a necessidade de operação sem intervenção humana, por serem ambientes dinâmicos ou inacessíveis. Além das limitações físicas, utilizar as RSSF como fonte de dados nestes ambientes requer a capacidade de monitoramento em tempo real, considerando suas imprevisibilidades de mobilidade e quantidade de nós. Neste artigo propomos o PROST, Protocolo de Roteamento Sensível ao Tempo, permitindo que as RSSF possam ser utilizadas para monitoramento de ambientes desafiadores.

**Palavras-Chave**—Redes de Sensores Sem Fio, Protocolos de Roteamento, Sistemas de Tempo Real, Computação Móvel.

**Abstract**—Advances and researches in Wireless Sensor Networks (WSN) allowed their use for several purposes. Unmanned space missions, interplanetary and underwater explorations are examples of critical and dangerous environments that benefit with their use. Applications in these environments share the need for operation without human intervention, by being inaccessible or dynamical environments. Besides physical limitations, using WSN as source of data in challenging environments requires the ability to real-time monitoring, considering its unpredictability of mobility and amount of nodes. In this paper we propose PROST, a time-aware routing protocol to enable WSN for real-time monitoring of challenging environments.

**Keywords**—Wireless Sensor Networks, Routing Protocols, Real-Time Systems, Mobile Computing.

## I. INTRODUÇÃO

Os avanços nas pesquisas em Redes de Sensores Sem Fio (RSSF) têm permitido a sua utilização para os mais diversos fins. Missões espaciais não tripuladas, explorações subaquáticas e interplanetárias são exemplos de ambientes críticos e perigosos que podem se beneficiar com sua utilização. As aplicações para estes ambientes compartilham a necessidade de operação sem intervenção humana, tanto por serem ambientes inacessíveis para o homem, quanto pelo dinamismo imposto por sua própria natureza. Esta necessidade de autonomia e eficiência no cumprimento de suas tarefas, sob diferentes condições do ambiente, insere novos desafios a serem considerados pelas RSSF.

Além das tradicionais limitações físicas que os sensores possuem, a utilização de RSSF como uma fonte coletora de

dados para ambientes desafiadores requer a capacidade de suporte à observação em tempo real do ambiente monitorado, considerando as imprevisibilidades da rede, com relação tanto à mobilidade quanto à quantidade de nós e dados coletados.

Neste artigo é proposto o PROST, um Protocolo de Roteamento Sensível ao Tempo, que tem como objetivo permitir que as RSSF possam ser utilizadas para monitoramento de ambientes desafiadores, que possuam a necessidade de leituras com restrições de tempo real, mesmo sob condições críticas de operação. Para analisar a eficiência do PROST, resultados de simulação da sua implementação são apresentados.

## II. MODELAGEM E FORMULAÇÃO DO PROBLEMA

Para o correto funcionamento das RSSF em ambientes desafiadores, em termos de autonomia e eficiência, é necessário considerar três grandes desafios: restrições de tempo real, mobilidade e limitações dos nós sensores. Diversas soluções existem para os problemas citados [1]–[3], porém, não é fácil encontrar soluções destes problemas conjuntamente.

Para considerar todos estes problemas, em conjunto, propomos a utilização de uma abordagem sensível ao tempo para RSSF, que efetua o monitoramento constante das condições das rotas, por meio da estimativa do atraso dos nós vizinhos. As condições assumidas nos estudos de caso para o protocolo proposto e os cenários que podem se beneficiar com sua aplicação são modelados da seguinte forma:

- 1) Sendo  $n_i$  um  $i$ -ésimo nó em uma RSSF e  $N$  o número de todos os nós dessa rede, cada  $n_i$  nó pode se comunicar apenas com seus vizinhos por transmissões sem fio.
- 2) O conjunto  $G_i$  representa o grupo dos vizinhos do nó  $n_i$ .  $n_j \in G_i$  se  $d_{ij} < r$ , onde  $d_{ij}$  é a distância entre os nós  $n_i$  e  $n_j$ , e  $r$  é o raio do alcance de  $n_i$ .
- 3) Cada grupo consiste numa subrede que pode ser interconectada a outra se  $n_j \in G_i$  e  $n_j \in G_k$ . Nesses casos,  $n_j$  pode atuar como um *gateway* entre as duas subredes.
- 4) A topologia da rede é formada pelo conjunto das subredes interconectadas em uma estrutura hierárquica de árvore com o nó  $n_0$  como o sorvedouro.  $p_i$  representa a profundidade, em saltos, do nó  $n_i$  para o sorvedouro.
- 5) Todos os nós podem sensoriar as condições físicas do ambiente e coletar dados dele. Todos os dados coletados são enviados ao mesmo destino, o nó sorvedouro  $n_0$ .
- 6) Cada nó  $n_i$  reporta seus dados a seu pai, um nó  $n_j$  imediatamente superior, onde  $p_j < p_i$  e  $p_j = p_i - 1$ . Ao receber um dado, se o nó  $n_i$  não é o sorvedouro, ele o encaminhará a seu próprio pai. O conjunto dos nós que atuam neste encaminhamento consiste na rota do nó  $n_i$ .

João B. Borges Neto e Rossana M. C. Andrade, Grupo de Redes de Computadores, Engenharia de Software e Sistemas, Programa de Pós-Graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, Brasil. E-mails: joaoborges.rossana@great.ufc.br.

Pedro F. Ribeiro Neto, Grupo de Engenharia de Software, Departamento de Informática, Universidade do Estado do Rio Grande do Norte, Mossoró, Brasil. E-mail: pedrofernandes@uern.br.

Este trabalho foi parcialmente financiado pela FUNCAP.

- 7) O nó sorvedouro  $n_0$  é responsável por disseminar a informação do atraso requerido  $\delta_{MAX}$  para validar os dados coletados. Um dado é válido para sua aplicação se  $\delta_{i0} \leq \delta_{MAX}$ , onde  $\delta_{i0}$  é o atraso fim-a-fim desde sua coleta pelo nó  $n_i$  até sua entrega ao nó  $n_0$ .
- 8) As rotas podem ser perdidas pela mobilidade dos nós, congestionamento do meio ou fim da energia dos nós. Falhas por ataques maliciosos não são consideradas neste trabalho.

### III. FONTES DE ATRASO E INCERTEZAS

Para melhor lidar com o atraso nas transmissões em RSSF é necessário analisar quais as principais fontes de atraso existentes nas comunicações sem fio. Baseado neste conhecimento, é possível identificar os fenômenos que causam incertezas nos cálculos dos atrasos e, assim, efetuar as correções necessárias para incrementar a precisão destes cálculos.

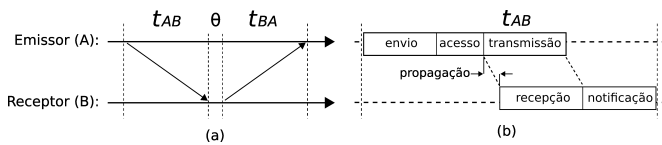


Fig. 1. (a)RTT Entre os Nós. (b)Fontes de Atraso Detalhadas. [4]

A Figura 1(a) ilustra o atraso existente na transmissão sem fio entre dois nós, um emissor (A) e um receptor (B), como no cálculo de RTT (*Round Trip Time*). Enquanto na Figura 1(b) é detalhada a transmissão de A para B ( $t_{AB}$ ). De acordo com [4], as principais fontes de atraso nesta transmissão são:

- 1) *tempo de envio* ( $t_{eA}$ ): Montagem do pacote pelo nó A.
- 2) *tempo de acesso* ( $t_{aA}$ ): Espera por canal livre.
- 3) *tempo de transmissão* ( $t_{tA}$ ): Conversão dos bits.
- 4) *tempo de propagação* ( $t_p$ ): Propagação no meio sem fio.
- 5) *tempo de recepção* ( $t_{rB}$ ): Recuperação dos bits por B.
- 6) *tempo de notificação* ( $t_{nB}$ ): Notificação no destino.

A Tabela I sumariza estas fontes de atraso, suas dependências e classificação de acordo com seu determinismo.

TABELA I

SUMMÁRIO DAS FONTES DE ATRASO. ADAPTADO DE [4]

Fonte	Dependência	Determinismo	Variação
Envio	Sobrecarga do processador	Baixo	10-100ms
Acesso	Congestionamento do meio sem fio	Alto	10-500ms
Transmissão	Vel. do rádio e tamanho do pacote	Determinístico	10-20ms
Propagação	Distância dos nós	Determinístico	$< 1 \mu s$
Recepção	Vel. do rádio e tamanho do pacote	Determinístico	10-20ms
Notificação	Sobrecarga do processador	Baixo	10-100ms

### IV. MECANISMOS DE ECONOMIA DE ENERGIA

#### A. Redução do Impacto das Incertezas

As incertezas presentes na comunicação sem fio são fatores prejudiciais ao correto funcionamento das tarefas que baseiam-se em transmissões de dados, principalmente para aquelas

com restrições de tempo real. Portanto, para alcançar maior precisão nos cálculos de tempo, é necessário reduzir o impacto que as incertezas causam nos processos de transmissão. Consequentemente, cálculos de tempo mais precisos possibilitarão maior otimização destes processos, alcançando níveis maiores de economia de energia.

Como visto na Tabela I, as fontes de atraso determinísticas não inserem incertezas nas transmissões, pois seu comportamento é sempre conhecido. Os tempos de *transmissão*, *recepção* e *propagação* são determinísticos e não requerem maiores preocupações.

Para reduzir o impacto das incertezas e garantir a precisão necessária na obtenção dos valores de tempo, são necessárias as seguintes premissas:

1) *Marcação do Tempo nas Camadas Inferiores*: Segundo [5], a marcação dos valores de tempo dos pacotes nas camadas mais baixas da pilha permite a eliminação das fontes de atraso existentes nos tempos de *envio*, *acesso* e *notificação*.

2) *Compensação do Erro dos Relógios*: Outra forma de compensar os erros oriundos das incertezas na obtenção dos valores de tempo é não utilizar os valores exatos obtidos por meio dos cálculos, mas sim, estimar seus valores com base em valores anteriores, que proporciona maior estabilidade nos resultados. Este mecanismo também é proposto por [6] e [4].

3) *Tamanho Uniforme de Pacotes*: Os cálculos de tempo efetuados para as comunicações de tempo real, como por exemplo a tarefa de sincronização de tempo e de cálculo do atraso das rotas, são efetuados por meio da troca de pacotes entre os nós. Portanto, para que se obtenha a precisão necessária nesses cálculos, é importante que o tamanho dos pacotes trocados por esses processos possuam o mesmo tamanho. Pacotes de mesmo tamanho possibilitam que a diferença entre os atrasos durante *transmissão* e *recepção* sejam praticamente nulos, já que ambos dependem somente dos tamanhos dos pacotes transmitidos, bem como do tempo total entre ida e volta do pacote.

#### B. Estimativa do Atraso das Rotas

Com o objetivo de permitir o monitoramento constante do atraso das rotas vizinhas a um nó sem degradar a energia limitada dos nós sensores e, conforme discutido na seção anterior, reduzir o impacto que as incertezas presentes na transmissão sem fio insere no cálculo dos atrasos, é definido um mecanismo capaz de efetuar estimativas simples do atraso dessas rotas.

A estimativa de atraso proposta é baseada na relação entre os tempos de *transmissão* e *recepção* de um pacote. Cada nó estimará o tempo que leva para enviar um pacote para seus vizinhos, baseado no tempo que os pacotes levam para chegar até ele, apenas com informações prévias de tempo que possui. Para isso, é assumido que as seguintes premissas são satisfeitas: os pacotes são uniformes, o *hardware* dos nós são homogêneos e os nós são sincronizados a uma mesma referência de tempo.

Utilizando como exemplo a comunicação da Figura 1, estimaremos o tempo da transmissão de A para B ( $t_{AB}$ ) conforme segue:

- Satisfeitas as premissas de que os pacotes e o *hardware* dos nós são homogêneos, assumimos que os atrasos determinísticos na *transmissão* ( $t_{tA}$  e  $t_{tB}$ ) e na *recepção* ( $t_{rA}$  e  $t_{rB}$ ) são equivalentes para  $t_{AB}$  e  $t_{BA}$ . E, sendo o atraso de *propagação* ( $t_p$ ) também determinístico, assumimos que a soma destes atrasos determinísticos são iguais, representada por  $t_D$ , como visto nas Equações 1 e 2.

$$t_{AB} = t_{eA} + t_{aA} + \underbrace{t_{tA} + t_p + t_{rB}}_{t_D} + t_{nB} \quad (1)$$

$$t_{BA} = t_{eB} + t_{aB} + \underbrace{t_{tB} + t_p + t_{rA}}_{t_D} + t_{nA} \quad (2)$$

- Como visto na Equação 3, também assumimos que os atrasos de *envio* ( $t_{eA}$  e  $t_{eB}$ ) e *recepção* ( $t_{nA}$  e  $t_{nB}$ ) são todos equivalentes, devido às operações idênticas realizadas pelos nós, como coletar e disseminar os dados.

$$t_{eA} \equiv t_{eB} \equiv t_{nA} \equiv t_{nB} \quad (3)$$

- Assim, sendo essas equivalências satisfeitas, podemos substituir a soma dos atrasos do *envio* e *notificação* da Equação 1 por uma nova equação que usa apenas um dos parâmetros. A Equação 4 corresponde a esta substituição, permitindo a utilização apenas dos parâmetros de A.

$$t_{eA} + t_{nB} \equiv 2t_{eA} \quad (4)$$

- De volta à Equação 1, o estimador do atraso na transmissão de A para B ( $\hat{t}_{AB}$ ) pode ser definido apenas utilizando as informações de um dos nós, como visto na Equação 5.

$$\hat{t}_{AB} = t_D + 2t_{eA} + t_{aA} \quad (5)$$

Uma vez modelada a equação da estimativa do atraso para o envio de um pacote (Equação 5), resta apenas tratar dos parâmetros aleatórios nela contidos, provenientes dos atrasos não-determinísticos. Embora o atraso do tempo de *envio* ( $t_{eA}$ ) seja não-determinístico, sua variação é baixa, não sendo necessário muita preocupação com ele. Já para o tempo de *acesso* ao meio ( $t_{aA}$ ) do nó A, é necessário conhecer o fenômeno aleatório que gera seu comportamento. Para isso, efetuamos simulações na ferramenta computacional NS-2 [7], utilizando o protocolo IEEE 802.15.4 [8] como o protocolo das camadas física e de enlace, e analisamos os tempos coletados durante o período da simulação, em diferentes momentos, sob diferentes topologias e configurações.

O tempo de *acesso* ao meio comportou-se bem imprevisível nas simulações. A Figura 2(a) corresponde ao histograma das amostras obtidas sobreposto com o gráfico da função densidade de probabilidade (PDF) ideal para uma distribuição exponencial e na Figura 2(b) está a função de distribuição empírica (EDF) das amostras, sobreposto a ela está a função de distribuição acumulada (CDF) ideal para uma distribuição exponencial. Isto implica, então, que o fenômeno segue uma distribuição de probabilidade exponencial, com média igual a 2.5ms, variância igual a  $9.9\mu s$  e  $\lambda$  igual a 395.8.

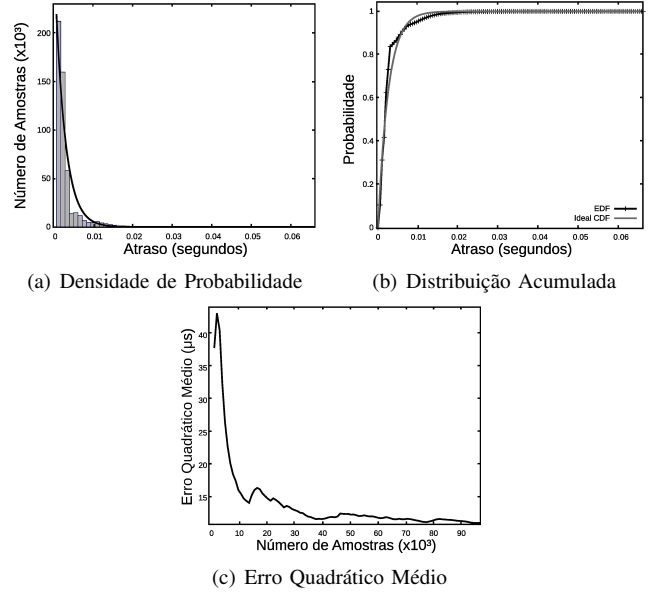


Fig. 2. Análise de Variável Aleatória  $t_{aA}$

De volta à Equação 5, optamos por substituir as variáveis aleatórias (tempos de *envio* e de *acesso*) pelo valor correspondente de suas médias, obtidas em tempo de execução. A Equação 6 corresponde a esse novo estimador, onde  $n$  equivale ao número de amostras obtidas por cada nó sensor.

$$\hat{t}_{AB} = t_D + \left( \frac{2}{n} \sum_{i=1}^n t_{eA_i} \right) + \left( \frac{1}{n} \sum_{i=1}^n t_{aA_i} \right) \quad (6)$$

Apesar da média não ser um ótimo estimador, a utilizamos em nossos experimentos, já que o coeficiente de correlação das amostras é muito baixo, equivalente a 0.021. Com o mesmo cenário da simulação anterior, analisamos o erro quadrático médio (MSE - *Mean Square Error*) obtido da diferença entre o valor de atraso estimado e o real atraso para o envio dos pacotes, que pode ser visto na Figura 2(c). Notamos que o erro do estimador diminui à medida que o número de amostras aumenta, tornando seu uso satisfatório para nossos propósitos, que tem como seu objetivo aumentar a quantidade de dados trafegados na rede.

## V. PROST

O principal objetivo deste artigo é a descrição de um protocolo de roteamento que permita que as RSSF sejam capazes de entregar os dados coletados pelos nós sensores ao seu destino, atendendo às restrições de tempo impostas pelas aplicações que utilizarão esses dados. O protocolo também deve garantir que as imprevisibilidades da rede, como a mobilidade e a variação na quantidade de nós sensores, não prejudiquem a qualidade temporal destas entregas de dados.

### A. Descrição Funcional

O protocolo PROST define 3 processos operacionais, que constituem-se por um conjunto de atividades efetuadas por um nó sensor, com o intuito de garantir a sua autonomia

em diferentes condições da rede. Os processos operacionais definidos pelo PROST são:

- Descoberta de Rotas;
- Monitoramento de Rotas; e
- Manutenção de Rotas.

1) *Descoberta de Rotas*: O processo de descoberta de rotas é definido como um processo composto pela transmissão de 3 pacotes (*three-way handshake*), efetuado por cada nó comunicando-se com os nós superiores na hierarquia, sendo um pacote utilizado pela criação da hierarquia e os dois restantes utilizados pela sincronização de tempo.

A atividade de criação da hierarquia ocorre de forma recursiva, iniciada pelo nó sorvedouro, por meio do envio de um pacote de *descoberta de rotas* (Figura 3(a)), seguido pela atividade de sincronização de tempo (Figura 3(b)).

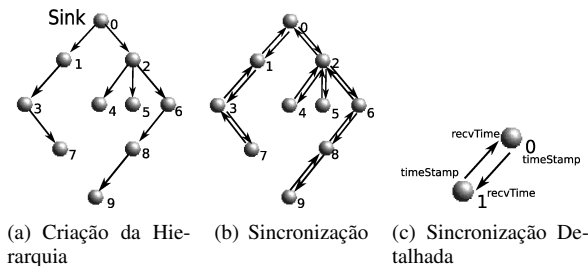


Fig. 3. Processo de Descoberta de Rotas do PROST

A Figura 3(c) detalha a atividade de sincronização de tempo efetuada pelos nós sensores. O algoritmo de sincronização de tempo dos nós usado pelo PROST é baseado na abordagem emissor-receptor definido por [5]. Como visto na figura, o nó 1 inicia a sincronização por meio do envio de um pacote de *sincronização de tempo* para o nó 0, seu pai na hierarquia. Enquanto o pacote é enviado, o nó 1 efetua a marcação do seu tempo de envio do pacote na camada mais inferior possível. No exemplo da figura ele é ilustrado como a variável *timeStamp*, essa informação vai junta ao pacote de sincronização no campo *Tempo Decorrido*. Ao receber o pacote, o nó 0 marcará o tempo desta recepção também na camada mais inferior possível, ilustrado na figura como a variável *recvTime*.

A diferença entre o tempo de recepção e envio do pacote de sincronização corresponde ao tempo decorrido ( $td$ ) para esta comunicação, visto na Equação 7.

$$td = recvTime - timeStamp \quad (7)$$

Após efetuar o cálculo do  $td$ , o nó 0 também enviará um pacote de *sincronização de tempo*, como resposta da sincronização para o nó 1. São efetuados os mesmos procedimentos de marcação do tempo de envio e recepção deste pacote nas camadas inferiores, por ambos os nós, mas diferentemente do pacote inicial, neste segundo haverá a informação do *tempo decorrido remoto*, calculado pelo nó 0, e a informação do *atraso requerido* pela aplicação.

Com ambos os tempos decorridos calculados pelo nó 0 remotamente e pelo seu tempo decorrido calculado após a recepção do pacote de resposta de sincronização, o nó 1 poderá calcular o atraso ( $\delta$ ) desta transmissão e a diferença entre o seu

relógio ( $\Delta$ ) e o relógio do nó 0. Esses valores são calculados conforme a Equação 8.

$$\delta = \frac{(td_{remoto} + td)}{2}, \quad \Delta = \frac{(td_{remoto} - td)}{2} \quad (8)$$

Onde  $td$  equivale ao tempo decorrido e  $td_{remoto}$  ao tempo decorrido remoto obtido do pacote de sincronização.

Caso o valor do atraso desta rota satisfaça o valor do atraso requerido, informado pelo sorvedouro, o nó irá considerar esta como uma rota válida e, tendo definido este nó superior como seu nó pai, mudará seu estado para *ativo*. Agora este nó é capaz de enviar todos os dados que ele coletar para seu pai. Por outro lado, se o atraso da rota não pode satisfazer o atraso requerido, esta rota será considerada inválida, e o nó não será capaz de enviar pacotes, todos os dados coletados serão descartados, para economizar energia, e o nó mudará seu estado para *restrito*.

Após a sincronização de tempo, os nós 1 e 2 irão transmitir um pacote de descoberta de rotas para os seus vizinhos e o processo continuará recursivamente até que todos os outros nós alcançáveis da rede possam participar da hierarquia.

2) *Monitoramento de Rotas*: Após a execução do processo de descoberta de rotas, a topologia da rede é criada, em hierarquia em forma de árvore, de acordo com a métrica de atraso informado pelo sorvedouro e todos os nós estão sincronizados com a referência do relógio do nó sorvedouro. A partir deste momento, é iniciado o processo de monitoramento de rotas, um processo pró-ativo que consiste de três atividades: atualização das informações de vizinhos, inserção de novos nós na topologia e detecção de rotas inválidas.

Para a primeira atividade, o PROST utiliza *pacotes de carona*, embutidos nos pacotes de dados transmitidos pelos nós, em seu *payload*. Este pacote contém a informação do tempo decorrido, tal como definido por [9], e do ID do nó de destino deste pacote.

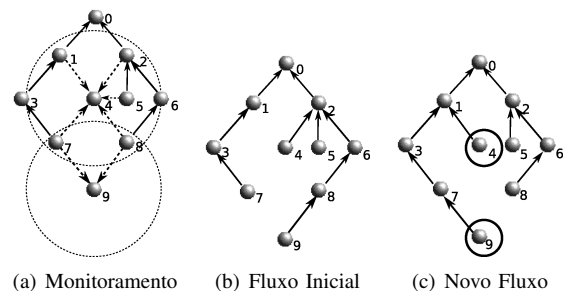


Fig. 4. Processos de Monitoramento e Manutenção

Tomando como exemplo a Figura 4(a), os nós 4 e 9 estão monitorando os atrasos de seus vizinhos, que estão se comunicando em seu raio de alcance, linhas tracejadas, por meio da recuperação do pacote de carona contido no cabeçalho dos pacotes de dados trafegados, linhas cheias.

Para ser possível que os pacotes de dados, que contêm as informações de carona, sejam lidos pelos nós em monitoramento, todos os pacotes de dados são transmitidos para o endereço de *broadcast*, de acordo com o sistema de endereçamento utilizado. O PROST utiliza o campo de

*destino* contido no pacote de carona para efetuar o controle do fluxo dos dados. Isso permite a execução da segunda tarefa, de inserir na topologia novos nós ociosos, que não tenham recebido o primeiro pacote de descoberta de rotas.

As informações dos nós vizinhos obtidos pelo processo de monitoramento serão utilizadas para efetuar a estimativa do atraso que os nós levam para enviar seus dados aos vizinhos. Com a estimativa do atraso de cada rota, um nó pode realizar a sua terceira missão, e detectar se uma rota é válida ou não. Se a rota para o seu pai se tornou inválida, o nó poderá eficientemente alterar seu pai para um outro nó que possua uma rota com atraso válido. Mas essa tarefa de alterar a sua rota é responsabilidade do próximo processo operacional, da manutenção de rotas.

3) *Manutenção de Rotas*: A manutenção de rotas é um processo reativo ativado pelo processo de monitoramento de rotas, quando ele detecta a necessidade de adaptação da rota devido a mudanças ocorridas nas condições da rede. Uma vez ativado, este processo buscará reparar a rota inválida por uma nova rota válida.

Uma rota válida para o PROST depende tanto do atraso identificado para esta rota quanto da profundidade de seu nó vizinho. Para evitar erros com rotas cíclicas, quando um nó é o pai e o filho de um outro nó, o PROST usa um algoritmo baseado na definição de [3]. A principal diferença é que no PROST cada nó é responsável pela decisão de escolher o seu novo pai, e não o pai que decidirá adotar um nó órfão, tal como definido em [3].

Para ser considerado um nó pai válido para o PROST, um nó que é um pai em potencial para uma nova rota deve satisfazer as seguintes condições:

- O nó não é meu filho;
- O atraso requerido é válido; e
- A profundidade do nó atende às seguintes exigências:
  - Sua profundidade é menor ou igual à minha; ou
  - Sua profundidade é igual à do meu filho.

Se os nós 4 e 9, como no exemplo da Figura 4, encontrarem uma nova rota válida, eles mudarão as informações de seu nó pai de acordo com as informações já monitoradas. Essa rápida alternância entre as rotas dos nós, do fluxo de dados inicial visto na Figura 4(b), para o novo fluxo de dados visto na Figura 4(c), torna possível a adaptação eficiente das rotas para suportar as mudanças nas condições da rede.

## VI. RESULTADOS DA SIMULAÇÃO

O protocolo PROST foi implementado e simulado na ferramenta computacional NS-2, *Network Simulator*, versão 2.33 [7]. Os parâmetros utilizados nos cenários da simulação são os seguintes:

- *Aplicação*: Tráfego CBR
- *Protocolo (MAC/PHY)*: IEEE 802.15.4
- *Tamanho dos Cenários*: 100m x 100m
- *Raio de Transmissão*: 15m
- *Energia*: 10J (Joules)
- *Modelo de Mobilidade*: Random Waypoint
- *Velocidade dos Nós*: 10 m/s
- *Tempo de Simulação*: 200s

O PROST foi comparado com resultados dos protocolos SWIFT [3] e AODV [10], simulados nos mesmos cenários. Analisamos o impacto de 4 condições da rede sobre a taxa de entrega dos dados pelos protocolos e a taxa de sucesso nas entregas atendendo às restrições de tempo:

- *Escalabilidade* - Variando o número de nós: 10-100 [50]
- *Mobilidade* - Variando o tempo de pausa: 0-200s [100s]
- *Congestionamento* - Var. interv. de envios: 0.1-1.0s [0.5s]
- *Tempo Real* - Variando atraso requerido: 5-50ms [20ms]

Os valores padrão estão destacados entre colchetes acima.

Todos os nós são móveis, incluindo o sorvedouro, e os nós comuns são responsáveis por coletar dados, representados pelos envios do tráfego CBR, e enviados ao sorvedouro. Os valores dos gráficos são valores médios de obtidos de 100 amostras de simulação, com 95% de intervalo de confiança. O posicionamento inicial dos nós é definido conforme uma distribuição de probabilidade uniforme.

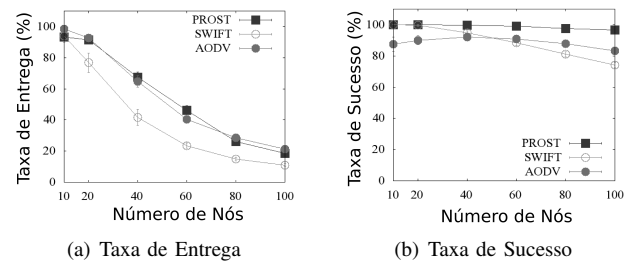


Fig. 5. Impacto do Número de Nós (Escalabilidade)

A Figura 5(a) mostra o impacto da escalabilidade da rede sobre a taxa de entrega dos nós. Conforme o número de nós aumenta, a taxa de entrega de todos os protocolos diminui. Porém, como mostrado na Figura 5(b), a taxa de sucesso dos pacotes, entregues de acordo com as suas restrições de tempo, do PROST permanece quase inalterada, com cerca de 100% de sucesso.

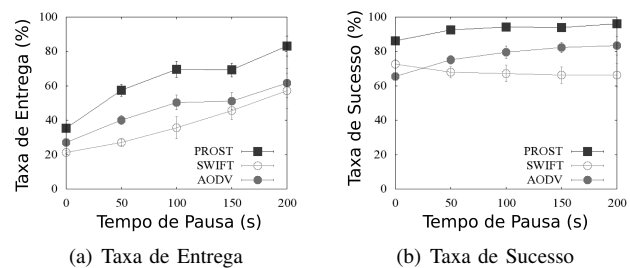


Fig. 6. Impacto do Tempo de Pausa (Mobilidade)

Como visto na Figura 6(a), a taxa de entrega dos protocolos é afetada pela mobilidade dos nós. Zero de tempo pausa significa que os nós se deslocam durante todo o tempo da simulação, enquanto que para 200 de tempo de pausa eles permanecem imóveis todo o tempo. O PROST teve melhor desempenho na entrega dos dados sob condições de mobilidade, devido às adaptações mais eficientes para as mudanças na topologia da rede. Como visto na Figura 6(b), verificamos que o PROST foi o melhor nas entregas atendendo ao prazo de tempo, com taxa de sucesso de quase 90%.

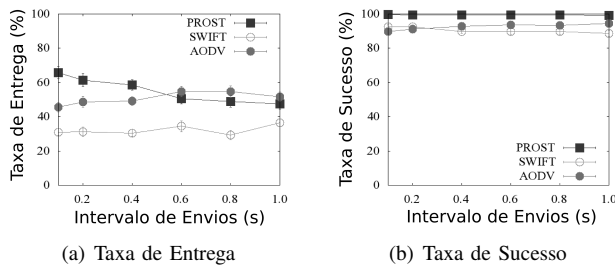


Fig. 7. Impacto do Intervalo de Envios (Congestionamento)

Na Figura 7(a) analisamos o impacto do congestionamento. O intervalo entre envios de 0.1s significa o envio de 10 pacotes por segundo, enquanto que 1.0s significa o envio de um pacote por segundo. Enquanto a taxa de entrega dos outros protocolos é quase inalterada, o PROST tem um comportamento decrescente. Sua eficiência possui forte dependência no número de pacotes de carona recebidos, possibilitando maior precisão de seu estimador (Seção IV-B), e o PROST pode se adaptar de forma mais eficiente. Todavia, como pode ser visto na Figura 7(b), embora sua sensibilidade ao tempo diminui, a taxa de sucesso das entregas permanece inalterada e mais eficiente.

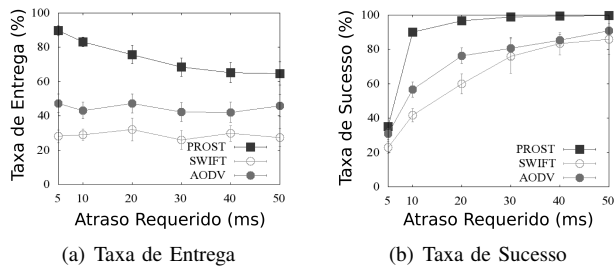


Fig. 8. Impacto do Prazo de Entrega (Restrições de Tempo Real)

A Figura 8(a) mostra o impacto de diferentes restrições de tempo real sobre a taxa de entrega dos protocolos. Ao contrário dos outros protocolos, o PROST é muito influenciado pelo atraso requerido. Enquanto o prazo do atraso é mais crítico, mais agilmente o PROST será na manutenção das rotas. Para prazos mais curtos, o processo de monitoramento detectará mais rotas inválidas, e acionará a manutenção de rotas mais vezes, fazendo com que o suporte à mobilidade seja mais eficiente, como pode ser visto na Figura 8(a). Para a taxa de sucesso dos protocolos, Figura 8(b), À medida que esse prazo é relaxado, a taxa de sucesso dos protocolos aumenta, com o PROST sendo mais eficiente, mesmo para prazos mais curtos.

O consumo de energia é visto na Figura 9, verificamos que enquanto os protocolos AODV e SWIFT têm comportamentos semelhantes no consumo de energia, os processos utilizados pelo PROST aumentam o consumo de energia dos nós e diminui a vida útil da rede. No entanto, este consumo não está longe dos outros protocolos e, como visto nas outras análises, os benefícios do monitoramento constante do PROST justificam sua utilização para os cenários onde restrições de tempo real e suporte à mobilidade são fatores cruciais para o seu funcionamento.

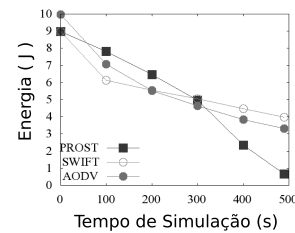


Fig. 9. Consumo de Energia

## VII. CONCLUSÕES E TRABALHOS FUTUROS

Neste artigo apresentamos o PROST, um protocolo de roteamento sensível ao tempo que permite que as RSSF possam ser utilizadas como uma fonte de dados em tempo real em ambientes desafiadores. Por utilizar técnicas para agilmente se adaptar às mudanças da rede, tal como monitoramento constante e eficaz das rotas, permite a sua utilização para suportar tanto a mobilidade como a variabilidade no número de nós da rede. Comprovamos por meio de simulações a eficiência e a viabilidade do PROST para as RSSF que têm como principal exigência as restrições de tempo real na entrega dos dados coletados. Como trabalhos futuros, propomos a implementação do PROST em nós reais, sendo avaliado em cenários reais e comparado com protocolos de tempo real.

## REFERÊNCIAS

- [1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*. Washington, DC, USA: IEEE Computer Society, 2000, p. 8020.
- [2] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "Speed: A stateless protocol for real-time communication in sensor networks," in *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2003, p. 46.
- [3] S. Kim, V. Vasireddy, and K. Harfoush, "Scalable coordination for sensor networks in challenging environments," in *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2007, pp. 214–221.
- [4] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi, "The flooding time synchronization protocol," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 39–49.
- [5] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 138–149.
- [6] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *SIGOPS Oper. Syst.*, 2002, pp. 147–163.
- [7] NS-2, "Network simulator. available at <http://nsnam.isi.edu/nsnam/>." May 2009.
- [8] IEEE, "Ieee specific requirements part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, 2006.
- [9] B. Kusy, P. Dutta, P. Levis, M. Maroti, A. Ledeczi, and D. Culler, "Elapsed time on arrival: a simple and versatile primitive for canonical time synchronisation services," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 1, no. 4, pp. 239–251, 2006.
- [10] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Second IEEE Workshop on Mobile Computer Systems and Applications*, 1999, p. 90.