Seleção de Nó Raiz Ótimo para Sincronização Eficiente de Relógios em Redes de Sensores Sem Fio

Tiago C. de S. Xavier[1], **Claudio L. Amorim**[1], **Flavia C. Delicato**[1]

[1]Universidade Federal do Rio de Janeiro (COPPE-UFRJ)
Rio de Janeiro – RJ – Brasil

{tiagocariolano,amorim}@cos.ufrj.br, fdelicato@dcc.ufrj.br

Abstract. *To reduce clock synchronization error in wireless sensor networks, recent works aimed to optimize network topology. Such approaches presented limited performance since the root node, who provides global clock, may remain being located in unfavorable positions after topological optimization. Moreover, in case of node failures, recuperation is a very costly operation. This work presents two proposals to select network optimal root node: the first is a graph-based approach and the second is a distributed algorithm which it is applied to a failure scenario. Experiments show that selecting optimal root node, rather than optimizing topology, decreases synchronization error close to six times less and distributed approach is able to be robust to failures of the root node.*

Resumo. *Para reduzir o erro de sincronização de relógios em redes de sensores sem fio, trabalhos recentes buscaram otimizar a topologia da rede. Essas abordagens apresentaram desempenho limitado pois o nó raiz, o qual fornece o relógio global, pode continuar mal localizado mesmo após a otimização e em caso de falha, a recuperação é muito custosa. Este trabalho propõe duas abordagens para selecionar o nó raiz ótimo da rede: a primeira é baseada em grafos e a segunda é um algoritmo distribuído aplicado em um cenário de falha. Experimentos indicam que, comparada à otimização de topologia, a seleção do raiz ótimo reduz cerca de seis vezes o erro de sincronização e que a abordagem distribuida é capaz de ser resiliente em cenários de falha do nó raiz.*

1. Introdução

Redes de sensores sem fio (RSSFs) têm sido cada vez mais utilizadas como poderosos sistemas de aquisição de dados. Suas características de flexibidade, facilidade de implantação e baixo custo as tornam atraentes para uma ampla gama de domínios de aplicações [Kamilaris and Pitsillides 2016].

Um dos principais problemas no projeto RSSF de larga escala consiste na sincronização temporal dos relógios dos diversos nós. O objetivo da sincronização de relógios é fazer com que todos os nós tenham uma visão comum do tempo. A principal forma de sincronizar relógios de maneira estável e precisa é utilizar um nó como referência e, por meio de troca de mensagens de sincronização, fazer com que todos os nós estimem o valor de relógio deste nó (chamado de raiz) [Maróti et al. 2004, Yıldırım et al. 2018, Cardoso et al. 2017, Huang et al. 2014, Lenzen et al. 2015, Upadhyay et al. 2018]. A principal dificuldade com esta abordagem é que, conforme se aumenta a escala da rede, os nós que estão a mais de um salto de distância do nó raiz apresentam um erro de sincronização maior. Isso ocorre pois o erro de sincronização é cumulativo em função da distância para o raiz. Os autores em

[Lenzen et al. 2008] demonstraram teoricamente que o diâmetro da rede é correlacionado com o erro de sincronização na rede. Em seu estudo eles mostraram que, quanto maior a distância de um nó para o raiz, maior o erro do protocolo.

Para tentar mitigar este problema, trabalhos recentes tentam melhorar a sincronização de relógios explorando propriedades topológicas da rede para atenuar o problema do aumento da escala [Li et al. 2014, Panigrahi and Khilar 2015, Su et al. 2016]. Tais abordagens consistem em mudar virtualmente a topologia da rede ou escolher nós específicos por onde as mensagens irão trafegar. A topologia é modificada definindo *links* de comunicação entre os nós, forçando as mensagens a seguir determinada rota. A escolha estratégica de nós consiste em selecionar nós que iniciem a sincronização ou sejam responsáveis por repassar as mensagens de sincronização. Tanto os *links* quanto os nós são escolhidos baseados em parâmetros críticos do protocolo de sincronização.

O principal parâmetro considerado nesse tipo de abordagem é a distância em *hops* dos nós para o raiz, mas outros como a densidade da rede, a contenção do canal ou uma combinação deles podem ser utilizados. A distância em *hops* para o nó raiz é um dos principais fatores que afetam os protocolos de sincronização, pois quando o nó raiz inicia uma rodada de ressincronização existe um atraso entre o instante em que a mensagem sai do emissor e chega a cada um dos nós vizinhos. Esse atraso é decorrente do tempo de envio e recebimento da mensagem, tempo de transmissão da mensagem no enlace e tempo de propagação da mensagem no meio físico.

As abordagens de otimização topológica para mitigar este problema são limitadas em desempenho, porque mesmo que alguns caminhos das mensagens sejam otimizados ou que alguns nós sejam escolhidos para encaminhar os pacotes de maneira eficiente, o nó raiz pode estar mal localizado (por exemplo quando está em uma região distante dos outros nós da rede). Neste cenário, o erro de sincronização pode se tornar proibitivo para aqueles nós mais distantes do raiz. Além disso, abordagens de otimização de topologia possuem um alto custo em casos de falha de nós, principalmente do raiz. Quando ocorre uma falha, o algoritmo de otimização de topologia deve ser reexecutado e toda a rede reconfigurada, caso se deseje manter o erro de sincronização em níveis aceitáveis.

Visando abordar o problema do erro cumulativo de protocolos de sincronização baseados em nó raiz, apresentamos duas estratégias que exploram propriedades de topologia da rede para atenuar o erro de sincronização conforme se aumenta a distância dos nós para o nó raiz. A primeira estratégia consiste em um algoritmo baseado em grafos que seleciona o nó que minimiza as distâncias para todos os outros da rede. Ao invés de otimizar toda a topologia, a estratégia realiza uma seleção do nó raiz ótimo considerando a distância em *hops*. O algoritmo possui complexidade de tempo polinomial e é executado *offline*, durante o projeto da rede, antes do posicionamento dos nós. A segunda estratégia é um novo algoritmo que resolve o mesmo problema de selecionar o nó ótimo da rede, porém realiza isso de maneira distribuída. Este algoritmo distribuído é aplicado em situações de falha do nó raiz, é executado durante a operação da rede. É importante ressaltar que a estratégia proposta não requer qualquer dispositivo GPS (*Global Positioning System*), em qualquer nó sensor, o qual é um fator importante para a construção de RSSFs de baixo custo.

As principais contribuições deste trabalho são:

- Um algoritmo para seleção do nó raiz ótimo baseado em grafos. Este algoritmo é executado na implantação da rede e serve para encontrar o nó que minimiza as

distâncias para todos os outros da rede e possui complexidade de tempo polinomial, enquanto estratégias como [Li et al. 2014, Panigrahi and Khilar 2015], que fazem otimização topológica, são algoritmos de complexidade exponencial.

- Um algoritmo distribuído para resolver o problema de seleção de nó raiz ótimo em cenários de falhas do nó raiz. Este algoritmo é executado durante a operação dos nós sensores, monitorando por falhas do nó raiz e quando esse evento ocorre, disparando ações para mitigar o problema.

Resultados de simulações demonstraram que nossa abordagem de seleção de nó raiz baseada em grafos supera em quase seis vezes o ganho em relação ao erro de sincronização de uma abordagem que otimiza a topologia da rede. Além disso, experimentos exaustivos demonstraram que a abordagem distribuída é capaz de manter o ganho de erro de sincronização obtido pela abordagem baseada em grafos, mesmo em situações de falhas do nó raiz.

2. Trabalhos Relacionados

Muitas estratégias têm sido desenvolvidas para melhorar a acurácia de sincronização, porém sem considerar à localização dos nós, ou fazendo isso com o auxílio de dispositivos GPS. [Benzaïd et al. 2017] desenvolveram um mecanismo que divide a rede em *clusters* sincronizados e permite aos nós utilizarem mensagens de outros *clusters*, a fim de reduzir o *overhead* de comunicação. [Cardoso et al. 2017] propuseram uma solução híbrida para o problema de localização e de sincronização, em que um veículo aéreo não-tripulado equipado com um GPS é responsável por enviar as mensagens de sincronização aos nós. [Yıldırım et al. 2018] propuseram um protocolo de sincronização de relógios que aplica um mecanismo de correção de *feedback* proporcional e integral nos erros das mensagens de sincronização. [Upadhyay et al. 2018] desenvolveram um método de correção gaussiano para as diferenças de *offsets* entre os relógios dos nós.

Alguns trabalhos procuraram otimizar a topologia da rede, a fim de controlar adequadamente a comunicação envolvida na sincronização. [Li et al. 2014] propuseram uma estratégia para reduzir logicamente o diâmetro da rede a fim de diminuir o impacto do aumento da escala em termos de número de nós. Eles fizeram isso por meio da criação de um *overlay* virtual da rede, de modo que o caminho que uma mensagem passa do raiz até qualquer nó é significativamente reduzido. O problema gerado pelo algoritmo dos autores é NP-Completo. [Panigrahi and Khilar 2015] propuseram uma estratégia de otimização da topologia da rede, em que o algoritmo proposto é aplicado para protocolos de sincronização baseados em consenso. Ele seleciona um subconjunto de nós e seus vizinhos que serão responsáveis por iniciar a sincronização. Este problema de seleção é mapeado para o problema de encontrar o conjunto dominante conectado de um grafo, o qual é NP-Completo, e é resolvido por meio de um novo algoritmo genético. [Su et al. 2016] desenvolveram um esquema que seleciona nós para controle de mensagens de sincronização num ambiente de redes naúticas *adhoc*. Inicialmente a rede é dividida em camadas, de acordo com a distância em *hops* dos nós, e em cada camada, pelo menos um nó previamente selecionado é o responsável por repassar as mensagens de sincronização para os nós regulares. A principal diferença em relação ao presente trabalho é que na abordagem de [Su et al. 2016] é que ocorre uma seleção de nós por camadas, ao invés do nó raiz ótimo da rede, de modo que o problema de má localização do nó raiz não é solucionado.

3. Seleção de Nó Raiz Ótimo

3.1. Modelo de Rede e Definição do Problema

A rede de nós sensores é modelada como um grafo não-direcionado $G = (V, E)$, em que V é um conjunto de $n = |V|$ nós e E é um conjunto de arestas que representam *links* bidirecionais de comunicação. O conjunto de nós dentro de uma região de *broadcast* de qualquer nó $u \in V$ corresponde a seu conjunto de vizinhos e é denotado por $N_u = \{v \in V | (u, v) \wedge (v, u) \in E\}$. Um único nó da rede $r \in V$ é denotado como o nó raiz e é responsável por iniciar a sincronização. O custo de uma aresta conectando dois vértices $u \wedge v \in V$ é dado por $w(u, v) = 1$, que significa que u e v estão na área de cobertura um do outro, e se $w(u, v) = 1$ então $w(v, u) = 1$.

A seguir o problema de selecionar o nó que minimiza as distâncias para todos os outros é modelado como um problema de grafos. O conjunto de caminhos possíveis entre quaisquer dois nós v_0 e v_k, em que $0 \leq k \leq n$, é dado pela função $\pi(v_0, v_k) = \{\langle v_0, v_1, ..., v_k \rangle | \ \forall \ i \in \{1..k\} \Rightarrow (v_{i-1}, v_i) \in E\}$ e a distância em *hops* de um caminho $p \in \pi(v_0, v_k)$ é dada por

$$d(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i) \tag{1}$$

A partir da função acima, $D(u, v)$ é definido como o custo do caminho mais curto entre dois nós u e v da seguinte forma $D(u, v) = \min(\{d(p) \ \forall \ p \in \pi(u, v)\})$. Como o grafo é não-direcionado, tem-se que $D(u, v) = D(v, u)$. A distância do caminho mais longo entre o nó u e todos os outros, a qual corresponde à distância do nó mais distante de u, é definida a partir do conjunto $M_u = \{D(u, v) \ \forall \ v \in V\}$. M_u contém as distâncias de u para cada nó em V. Desse modo, a distância máxima de um nó u é $\max(M_u)$. O nó raiz r é aquele que minimiza as distâncias máximas entre todos os nós. Isso significa que r é o nó que minimiza $\{\max(M_u) \ \forall \ u \in V\}$, logo o problema de encontrar o nó que está mais próximo de todos os outros consiste em descobrir o nó r que torna a seguinte igualdade verdadeira

$$\min(\max(M_r)) = \min_{\forall u \in V}(\max(M_u)) \tag{2}$$

3.2. Algoritmo de Seleção de Nó Raiz Baseado em Grafo

O Algoritmo 1 foi proposto para resolver o problema delineado na Expressão 2. Ele é uma modificação da estratégia de programação dinâmica de Floyd-Warshall para resolver o problema de encontrar os caminhos mais curtos entre todos os pares de um grafo. O algoritmo usa esta estratégia para encontrar o conjunto M_u para cada nó u, calcular a distância dos nós mais distantes de u e identificar o nó que minimiza a distância para cada M_u. Sua entrada é o grafo da rede, bem como o número de vértices e a saída é o nó r.

As linhas de 1-9 correspondem à fase de inicialização. Na matriz $A = A^k_{ij}$, k representa as iterações que executarão operações nessa matriz. A cada iteração, A^k_{ij} contém a distância mínima entre i e j em algum caminho que contém os nós de 1 até k. As linhas de 11-21 fazem três tarefas: computam o caminho mais curto entre cada par de nó, computam a distância máxima de cada nó para todos os outros e calculam a distância mínima das distâncias máximas, bem como o nó r que a possui. Nas linhas 17-18, o algoritmo verifica se é mais curto o caminho passando pelo nó k ($A[i, k] + A[k, j] < A[i, j]$) e se sim atualiza essa distância em $A[i, j]$. Nas linhas 17-21, pra cada nó i, as variáveis M_i e $minmax$ são atualizadas, de modo que $minmax$ é usada para decidir na iteração k, qual

Algoritmo 1 Algoritmo de Seleção de Nó Raiz Ótimo Baseado em Grafos

Entrada: $G = (V, E)$ /* Grafo da rede de sensores */
$n = |V|$ /* Número de vértices*/
Saída: $r \in V$ /* Nó raiz */
1: $A = [n, n]$ /* Matriz auxiliar */
2: $minmax \leftarrow \infty$ /* Resultado do problema minmax */
3: **for** $i \in \{1..n\}$ **do**
4: **for** $j \in \{1..n\}$ **do**
5: **if** $i = j$ **then**
6: $A[i, j] \leftarrow 0$
7: **else if** $(i, j) \in E$ **then**
8: $A[i, j] \leftarrow (i, j)$
9: **else**
10: $A[i, j] \leftarrow \infty$
11: **for** $k \in \{1..n\}$ **do**
12: **for** $i \in \{1..n\}$ **do**
13: $M_i \leftarrow 0$
14: **for** $j \in \{1..n\}$ **do**
15: **if** $A[i, k] + A[k, j] < A[i, j]$ **then**
16: $A[i, j] \leftarrow A[i, k] + A[k, j]$
17: **if** $A[i, j] > M_i$ **and** $A[i, j] \neq \infty$ **then**
18: $M_i \leftarrow A[i, j]$
19: **if** $M_i < minmax$ **then**
20: $minmax \leftarrow M_i$
21: $r \leftarrow i$

é o raiz r até aquele momento. Por causa dos três *loops* aninhados este algoritmo tem complexidade de $O(n^3)$.

3.3. Algoritmo Distribuído de Seleção de Nó Raiz

Em caso de falha do nó raiz seria necessária uma re-execução do algoritmo da Seção 3.2 e uma reconfiguração do nó raiz. Esses passos de re-execução do algoritmo e reconfiguração de propriedades da rede são também necessários em abordagens como [Li et al. 2014, Panigrahi and Khilar 2015, Su et al. 2016]. No entanto, é possível utilizar um algoritmo distribuído para realizar a seleção de nó raiz ótimo, como será descrito a seguir.

Segundo a definição do problema formulada na Seção 3.1, o nó raiz é aquele que minimiza as distâncias para todos os outros. Suponha que todos os nós da rede enviassem uma mensagem para este nó r e que ele guardasse a identificação do nó emissor da mensagem em um conjunto L_r. Depois de algum tempo r receberia n mensagens e $L_r = V$. Se todos os nós realizassem o mesmo, então depois de algum tempo cada nó teria um dos conjuntos $L_{v_0} = L_{v_1} = ... = L_{v_n} = V$. No entanto, como r minimiza a distância em *hops* para os demais, então o conjunto L_r estaria completo, com n elementos antes dos outros. Nesse momento, r saberia que ele é o nó escolhido, então ele poderia enviar uma mensagem anunciando que é o raiz para a rede. A partir desta ideia, o Algoritmo 2 foi concebido para selecionar o nó raiz ótimo da rede. O código é executado em todos os nós da rede ($\forall u \in V$).

Em RSSFs o canal de comunicação entre os nós não é confiável e o tempo de enviar e receber uma mensagem, por *hop*, varia em todos os *links*, logo é necessário tratar estes problemas. Para resolver o primeiro problema pode-se usar um protocolo de

Algoritmo 2 Algoritmo Distribuído de Seleção de Nó Raiz Ótimo

Entrada: $n = |V|$ /* Número de nós*/
Saída: r /*Nó raiz ótimo*/
At Start
  $r \leftarrow \lambda$ /* Nó raiz */
  $L_u \leftarrow \emptyset$ /* Lista de nós verificados */
  $max_u \leftarrow 0$ /* Distância do nó mais distante de u */
  $minmax \leftarrow \infty$ /* Mínimo das distâncias máximas*/
  if DETECTADA FALHA DO NÓ RAIZ **then**
    Broadcast $MSG(SLCT, u, 1)$
Upon Receiving $MSG(SLCT, x, seq_x)$ **of** v
  if $max_u = 0$ **then**
    Broadcast $MSG(SLCT, u, 1)$
  if $x \notin L_u$ **and** $x \neq u$ **then**
    Broadcast $MSG(SLCT, x, seq_x)$
    $L_u \leftarrow L_u \cup \{x\}$
    $max_u \leftarrow \max(seq_x, max_u)$
    if $|L_u| = n - 1$ **then**
      if $max_u < minmax$ **or** $(max_u = minmax$ **and** $u < r)$ **then**
        $r \leftarrow u$
        $minmax \leftarrow max_u$
        Broadcast $MSG(ROOT, u, minmax)$
Upon Receiving $MSG(ROOT, x, max_x)$ **of** v
  if $max_x < minmax$ **or** $(max_x = minmax$ **and** $x < r)$ **then**
    $r \leftarrow x$
    $minmax \leftarrow max_x$
    Broadcast $MSG(ROOT, x, minmax)$

entrega confiável para RSSFs, já para resolver o segundo problema, que é mais difícil, o Algoritmo 2 faz uso de dois tipos de mensagens, $SLCT$ e $ROOT$, que servem para: (i) cada nó descobrir sua distância para cada outro da rede ($SLCT$), (ii) cada nó verificar se algum outro nó da rede ou ele próprio deve ser o raiz ($ROOT$). O item (ii) é uma verificação que resolve o segundo problema acima. A detecção de falha do nó raiz é realizada por meio da configuração de um *timer* em cada nó, de modo que se nenhuma mensagem de sincronização do raiz chegar após um determinado número de *timeouts* o nó considera que o raiz não está mais em operação.

4. Avaliação Experimental da Estratégia de Seleção de Nó Raiz

4.1. Configuração Experimental

Experimentos de simulação foram conduzidos para avaliar a estratégia de seleção de nó raiz contra a abordagem baseada em teoria espectral de grafos que otimiza a topologia da rede [Li et al. 2014]. Esta abordagem foi escolhida para comparação, pois ela tenta otimizar exatamente a distância em *hops* dos nós da rede para o nó de referência, o qual também é o principal objetivo da estratégia de seleção aqui proposta. Os dois parâmetros a avaliados foram:

- Distância em *hops* dos nós em relação ao raiz.
- Erro de sincronização do protocolo de sincronização.

O objetivo foi avaliar esses dois parâmetros e observar seu comportamento conforme se aumenta a escala da rede, tanto em cenários de redes esparsas, quanto para redes saturadas. O simulador utilizado foi o NS-3. O experimento consistiu em configurar uma área quadrada, posicionar os n nós aleatoriamente e executar o algoritmo de seleção de nó

raiz. O FTSP foi o protocolo utilizado tanto na abordagem de seleção de nó raiz quanto na abordagem de otimização da topologia [Li et al. 2014]. O resultado de cada experimento é a média da execução de 100 rodadas independentes. A Tabela 1 descreve todos os parâmetros da simulação e do protocolo de sincronização. Os experimentos foram conduzidos nos seguintes cenários, todos possuindo o FTSP como protocolo de sincronização subjacente:

- BASELINE: Apenas o algoritmo de sincronização FTSP é executado.
- OPTSEL : Estratégia de seleção de nó raiz ótima baseada em grafos descrita na Seção 3.
- TOPOREFINER : Estratégia de otimização de topologia da rede baseada em teoria espectral de grafos para redução da distância em *hops* dos nós da rede para o nó de referência[Li et al. 2014].
- BOTH (OPT+TOPO) : Realiza o refinamento de topologia (TOPOREFINER) e em seguida seleciona o nó raiz ótimo (OPTSEL).

Parâmetro	**Valor**
Número de nós máximo	2000
Área	$250m$ x $250m$, $500m$ x $500m$
Período de amostragem	$18s - 22s$ (Uniformemente distribuído)
Número de repetições	100
Raio de alcance do rádio	$30m$
Modelo de atraso de propagação	RANDOMPROPAGATIONDELAYMODEL
Atraso de mensagem	$0 - 100ms$

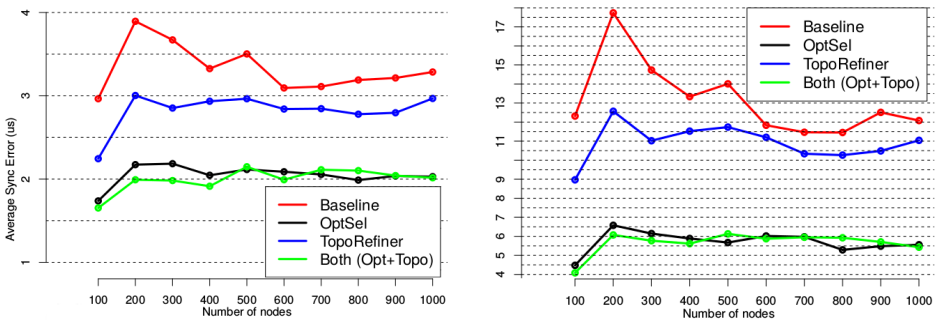
Tabela 1. Parâmetros da simulação e do protocolo de sincronização.

4.2. Erro de Sincronização

A Figura 1 e a Figura 2 apresentam os resultados de erro de sincronização em microssegundos em áreas de 250 x 250 m^2 e 500 x 500 m^2. Na Figura 1, TOPOREFINER foi capaz de melhorar o desempenho médio do protocolo FTSP (BASELINE) em 14%, porém OPTSEL apresentou um resultado ainda melhor de 42%. Com relação ao erro máximo, TOPOREFINER manteve aproximadamente a mesma relação de melhoria, enquanto OPTSEL foi mais do que 2 vezes melhor do que o BASELINE. Na Figura 2, OPTSEL foi pelo menos 4 vezes melhor que o BASELINE e 3 vezes melhor do que TOPOREFINER. Considerando o erro máximo da rede (Figura 2(b)), OPTSEL supera o BASELINE e TOPOREFINER por pelo menos 4 e 5 vezes, respectivamente. Em todos os cenários BOTH (OPT+TOPO) apresentou ganho semelhante ao uso apenas de OPTSEL.

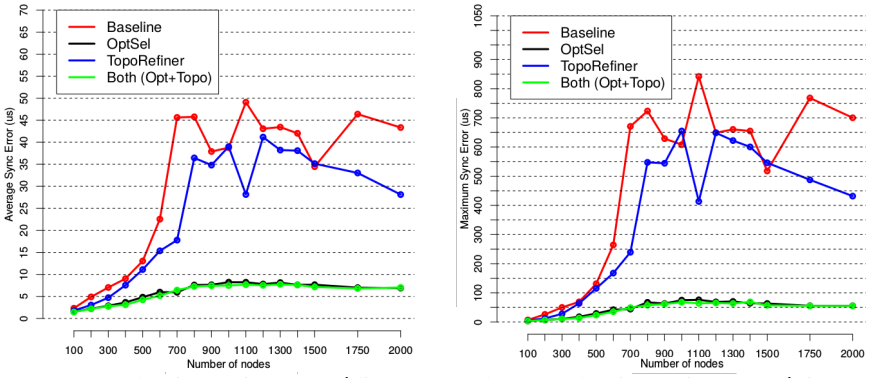
4.3. Distância em *hops*

A fim verificar a hipótese de que a escolha adequada do nó raiz ótimo pode reduzir significativamente sua distância para todos os outros e, consequentemente, o erro de sincronização, a distância de todos os nós para o nó raiz foram computadas e são apresentadas na Figura 3 e na Figura 4. Nessas figuras, é possível observar uma correlação entre a distância dos nós para o raiz e o erro de sincronização, como esperado. Os resultados indicam que a estratégia de seleção é capaz de melhorar o erro de sincronização significativamente e é muito mais efetiva do que a abordagem que realiza uma otimização da topologia da rede.



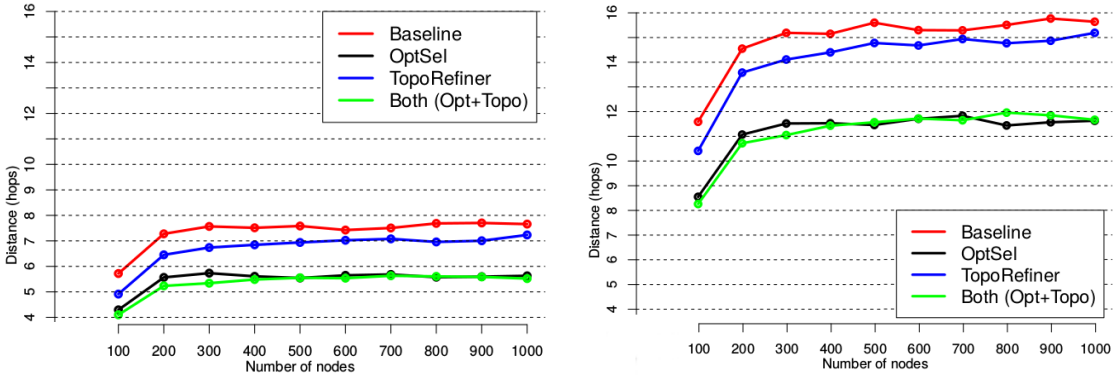
(a) Erro de Sincronização médio (b) Erro de Sincronização máximo

Figura 1. Erro de sincronização em uma área de 250 x 250 metros.



(a) Erro de Sincronização médio (b) Erro de Sincronização máximo

Figura 2. Erro de sincronização em uma área de 500 x 500 metros.

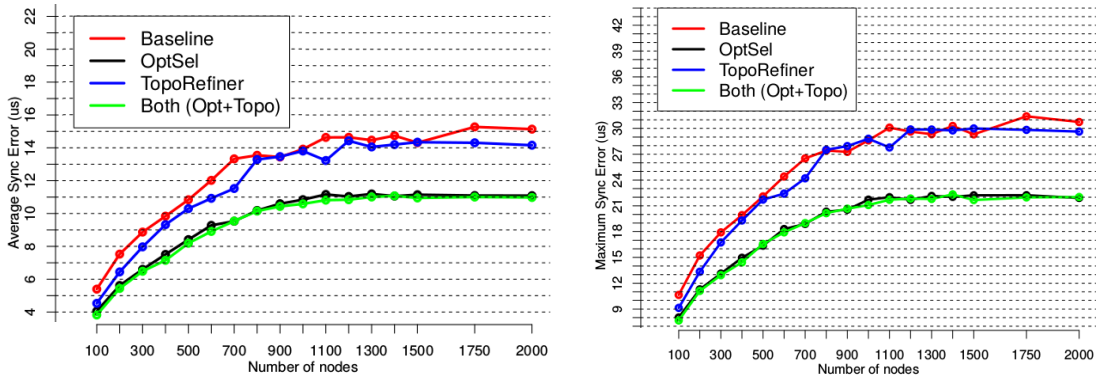


(a) Distância média (b) Distância máxima

Figura 3. Distância em *hops* para o nó raiz.

4.4. Tolerância à falha

Para demonstrar a efetividade do algoritmo distribuído descrito na Seção 3.3 em encontrar o nó raiz ótimo quando este falha, foi realizado um experimento de mais de 10 horas com 1000 nós, cujos resultados são apresentados na Figura 5. O experimento consistiu em: inicialmente executar o algoritmo de seleção baseado em grafos e desligar o nó raiz no instante 3000 segundos (linha vertical da figura). O objetivo foi verificar se a seleção do novo nó raiz ótimo faz com que o erro de sincronização permaneça nos mesmos níveis iniciais dados pelo algoritmo baseado em grafos. Para tal, dois cenários foram confrontados, um com a estratégia de seleção distribuída (OPTSELDIST) e outro sem ela (NOOPTSELDIST) - estratégia do FTSP de escolher o nó com menor ID [Maróti et al. 2004]. Na Figura 3.3, considerando o cenário OPTSELDIST, após a falha em 3000s o erro de sincronização per-



(a) Erro de Sincronização médio (b) Erro de Sincronização máximo

Figura 4. Distância em *hops* para o nó raiz.

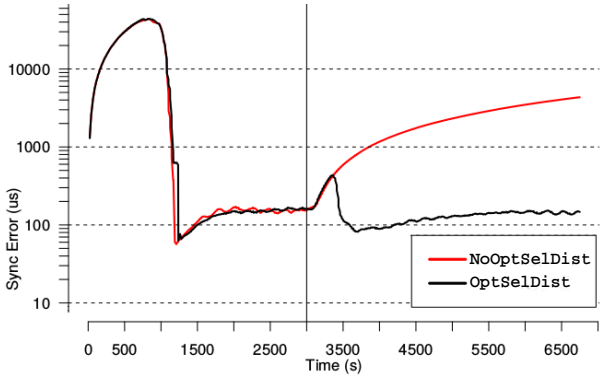


Figura 5. Erro de sincronização com desligamento do nó raiz aos 3000 segundos.

manece aproximadamente constante; sem a seleção de nó raiz ótimo, o erro aumenta. No início (entre 0 e cerca de 1000 s) o alto erro de sincronização se deve aos relógios iniciarem com valores aleatórios e à fase inicial de calibração do protocolo de sincronização subjacente FTSP.

5. Conclusões

Este trabalho apresentou duas propostas de seleção de nó raiz ótimo para reduzir a distância em *hops* dos nós para o raiz. Por meio de resultados experimentais é possível concluir que a abordagem baseada em grafos supera em cerca de seis vezes uma abordagem baseada em otimização topológica e a abordagem distribuída permite uma recuperação eficiente do nó raiz em caso de falha.

Referências

Benzaïd, C., Bagaa, M., and Younis, M. (2017). Efficient clock synchronization for clustered wireless sensor networks. *Ad Hoc Networks*, 56:13 – 27.

Cardoso, C. B., Guidoni, D. L., Kimura, B. Y., and Villas, L. A. (2017). A hybrid solution for 3d location and time synchronization in wsn. In *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '17, pages 105–112, New York, NY, USA. ACM.

Huang, G., Zomaya, A. Y., Delicato, F. C., and Pires, P. F. (2014). Long term and large scale time synchronization in wireless sensor networks. *Computer Communications*, 37(Supplement C):77 – 91.

Kamilaris, A. and Pitsillides, A. (2016). Mobile phone computing and the internet of things: A survey. *IEEE Internet of Things Journal*, 3(6):885–898.

Lenzen, C., Locher, T., and Wattenhofer, R. (2008). Clock synchronization with bounded global and local skew. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 509–518.

Lenzen, C., Sommer, P., and Wattenhofer, R. (2015). Pulsesync: An efficient and scalable clock synchronization protocol. *IEEE/ACM Trans. Netw.*, 23(3):717–727.

Li, X., Ma, Q., Sun, W., Liu, K., and Liu, Y. (2014). Topology shaping for time synchronization in wireless sensor networks. In *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 33–40.

Maróti, M., Kusy, B., Simon, G., and Lédeczi, A. (2004). The flooding time synchronization protocol. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 39–49, New York, NY, USA. ACM.

Panigrahi, N. and Khilar, P. M. (2015). An evolutionary based topological optimization strategy for consensus based clock synchronization protocols in wireless sensor network. *Swarm and Evolutionary Computation*, 22:66–85.

Su, X., Hui, B., and Chang, K. (2016). Multi-hop clock synchronization based on robust reference node selection for ship ad-hoc network. *Journal of Communications and Networks*, 18(1):65–74.

Upadhyay, D., Dubey, A. K., and Thilagam, P. S. (2018). Application of non-linear gaussian regression-based adaptive clock synchronization technique for wireless sensor network in agriculture. *IEEE Sensors Journal*, 18(10):4328–4335.

Yıldırım, K. S., Carli, R., and Schenato, L. (2018). Adaptive proportional 2013;integral clock synchronization in wireless sensor networks. *IEEE Transactions on Control Systems Technology*, 26(2):610–623.