

# Um Algoritmo Ciente da Correlação Espaço-Temporal e Consumo de Energia para Coleta de Dados em Redes de Sensores sem Fio

Leandro A. Villas<sup>1,3</sup>, Daniel L. Guidoni<sup>1</sup>, Regina B. Araujo<sup>2</sup>,  
Azzedine Boukerche<sup>3</sup> e Antonio A. F. Loureiro<sup>1</sup>

<sup>1</sup>UWL - Laboratório de Pesquisa – Universidade Federal de Minas Gerais

<sup>2</sup>WINDIS - Laboratório de Pesquisa – Universidade Federal de São Carlos

<sup>3</sup>PARADISE Research Laboratory – SITE, University of Ottawa

{leandro, guidoni, loureiro}@dcc.ufmg.br, regina@dc.ufscar.br

{boukerch, leandro}@site.uottawa.ca

**Abstract.** *Dense wireless sensor networks (WSNs) will be increasingly deployed in different classes of applications for accurate monitoring. Due to their high density of nodes, it is very likely that information that is both spatially and temporally correlated can be detected by several nearby nodes when these are sensing an event. The data correlation can be exploited to save energy, a key aspect on these networks. However, current proposals usually result in high delays and outdated data arriving at the sink node. In this work, we go further and propose a new algorithm, called Efficient Data Collection Aware of Spatio-Temporal Correlation (EAST), which uses shortest routes for forwarding the gathered data toward the sink node and fully exploit both spatial and temporal correlations to perform data collection in WSNs. Simulation results clearly indicate that our proposal can sense an event with a high accuracy of more than 99.7% while still saving the residual energy of the nodes in more than 14 times when compared to the accurate data collection strategy reported in the literature.*

**Resumo.** *Redes de sensores densas serão cada vez mais implantadas em diferentes classes de aplicações para o monitoramento preciso. Devido à alta densidade de nós, é muito provável que as informações espacialmente e temporalmente correlacionadas podem ser detectadas por vários nós próximos, quando estes estão sentindo um evento. A correlação dos dados pode ser explorada para economizar energia, um aspecto fundamental nessas redes. No entanto, as propostas atuais geralmente resultam em grandes atrasos e entrega de dados desatualizados. Neste trabalho, nós vamos mais longe e propusemos um novo algoritmo, chamado Efficient Data Collection Aware of Spatio-Temporal Correlation (EAST), qual utiliza rotas mais curtas para a transmissão de dados recolhidos em direção ao nó Sink e explora plenamente tanto correlações espaciais e temporais para realizar coleta de dados em RSSFs. Resultados de simulação indicam claramente que a nossa proposta pode sentir um evento com uma alta precisão de mais de 99,7% enquanto ainda economiza a energia residual dos nós em mais de 14 vezes quando comparado com a estratégia de coleta de dados precisos na literatura.*

## 1. Introdução

Rede de Sensor Sem Fio (RSSF) [Akyildiz et al. 2002, Romer and Mattern 2004, Boukerche 2008, Villas et al. 2010] pode ser definida como uma rede cooperativa composta por nós sensores equipados com uma interface sem fio, dispositivos de sensoriamento, processador, memória e energia. Os objetivos de uma RSSF são: monitorar fenômenos físicos e transmitir os dados sensoreados para um nó *Sink* usando normalmente comunicação *multihop*. O nó *Sink* será responsável por processar todos os dados recebidos de vários nós fontes e reporta-los para uma instalação de monitoramento.

Uma das principais limitações de RSSFs é a utilização de bateria pelos nós sensores, o que faz esse tipo de rede ser altamente restritiva em relação ao consumo de energia. A comunicação de dados é a tarefa que mais consome energia em um nó sensor. Dessa forma, algoritmos e protocolos projetados para RSSF devem considerar o consumo de energia em sua concepção. Para economizar energia, a maioria dos protocolos evita o envio imediato dos dados sensoreados, para então processá-los e enviá-los em menor quantidade [Nakamura et al. 2006, Fan et al. 2006, Villas et al. 2009, Villas et al. 2010]. Entretanto, essa estratégia pode resultar em uma entrega de informações desatualizadas, limitando assim a sua utilização.

Neste trabalho nós vamos além de atrasar o envio dos dados sensoreados para processar os dados e reduzir a quantidade de comunicações. Para isso, este trabalho explora a correlação espaço-temporal dos dados coletados com o objetivo de reduzir o consumo de energia e prover uma entrega de informação em tempo hábil sobre a área de monitoramento. O algoritmo proposto nesse trabalho evolui trabalhos da literatura [Min and Chung 2010, Zhang et al. 2010, Villas et al. 2011], nos quais mostraram que em várias aplicações de RSSF, dados coletados por nós vizinhos tendem ser correlacionados no tempo e espaço. Dados correlacionados no tempo possuem uma pequena mudança do padrão da leitura atual comparada às leituras anteriores. Dados correlacionados no espaço possuem uma pequena mudança no padrão das leituras em nós próximos geograficamente. Algumas soluções da literatura exploram a correlação espacial [Akyildiz et al. 2004, Le et al. 2008, Liu et al. 2007b, Yoon and Shahabi 2005, Yuan and Chen 2009] e temporal [Min and Chung 2010, Pham et al. 2010]. Entretanto, as soluções introduzem atrasos e entregam dados desatualizados para o nó *Sink* [Deligiannakis and Kotidis 2008, Pham et al. 2008, Pham et al. 2010].

O algoritmo proposto nesse trabalho, chamado de EAST (Efficient Data Collection Aware of Spatio-Temporal Correlation), os nós sensores que detectam um evento são agrupados considerando uma correlação espacial entre eles, enquanto o nó líder e os nós representativos do grupo consideram a correlação temporal para o envio de dados coletados. O nó líder do grupo transmite para o *Sink* um valor que representa todas as leituras feitas pelos nós representativos do grupo, que são um subconjunto dos nós que está sensoreando o evento.

Um dos aspectos principais da solução proposta é que ela pode ser dinamicamente ajustada para melhor representar as características do evento sensoreado e da energia residual dos nós, que são comumente desconsideradas pelas soluções da literatura. Assim, o algoritmo EAST possui como características (i) balanceamento da energia consumida pelos nós sensores, (ii) redução do consumo de energia eliminando notificações redundantes e (iii) ajuste dinâmico da quantidade de nós representativos de um grupo baseado

nas características do evento e requisitos de precisão da aplicação. De forma a prover um entendimento do comportamento do algoritmo proposto em diferentes cenários, será apresentado um conjunto de experimentos no qual o algoritmo proposto é comparado com soluções da literatura. Resultados obtidos mostram que o algoritmo proposto possui uma alta precisão dos dados coletados com um balanceamento da energia consumida e pequeno atraso na entrega dos dados.

Este trabalho está organizado da seguinte forma. A próxima Seção apresenta os trabalhos relacionados. A Seção 3 descreve o algoritmo EAST. A Seção 4 apresenta os resultados de simulação. Finalmente, a Seção 5 apresenta as conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

A correlação espacial dos dados coletados pelos nós existe quando esses nós estão geograficamente próximos. Nesse caso, ao invés de todos os nós notificarem os dados sensorados para o *Sink*, apenas um subconjunto representativo desses nós devem reportar seus dados. Dados coletados em diferentes intervalos de tempo podem ser correlacionados temporalmente se existir uma pequena variação entre eles. Por exemplo, uma amostragem de temperatura realizada a cada minuto de um dia pode produzir dados temporalmente correlacionados, ou seja, valores de temperatura similares em uma determinada janela de tempo. Vários algoritmos da literatura exploram a correlação espacial e temporal na coleta de dados [Akyildiz et al. 2004, Le et al. 2008, Liu et al. 2007a, Deligiannakis and Kotidis 2008, Min and Chung 2010, Zhang et al. 2010, Villas et al. 2011]. A seguir, serão investigado as soluções parecidas com a solução proposta nesse trabalho.

Akyildiz et al. [Akyildiz et al. 2004] verificaram a relação entre confiança da detecção de um evento e a correlação espacial dos nós que detectaram o evento em questão. A solução proposta estima a quantidade de nós sensores (nós representativos) necessária para reportar o evento detectado com o objetivo de se ter uma informação confiável sobre o evento. Embora a solução proposta possua um baixo consumo de energia, ela não considera a energia residual na escolha dos nós representativos. Se um nó representativo possui esse papel por um longo período de tempo, seu consumo de energia é maior devido a quantidade de dados transmitidos comparado aos demais nós que detectaram o evento.

Liu et al. [Liu et al. 2007a] propuseram um algoritmo chamado *Energy-Efficient Data Collection framework - (EEDC)*, qual explora a correlação espacial dos dados coletados. Primeiramente, todos os dados coletados são enviados para o *Sink*. De posse de todos os dados, o *Sink* executa o algoritmo EEDC para encontrar grupos de nós com dados correlacionados espacialmente. A principal desvantagem do algoritmo EEDC é a consideração que todos os nós estão a um salto do *Sink*. Essa suposição não é prática em sistemas distribuídos, dificultando soluções em RSSFs para uma rede com grande quantidade de nós. Outra desvantagem é o algoritmo para encontrar grupos de nós, que é executado de maneira centralizada no *Sink*. Por causa disso, todos os dados da rede devem ser reportados para o *Sink*, que irá armazená-los e processá-los.

Pham et al. [Pham et al. 2008, Pham et al. 2010] propuseram uma solução que combina a utilização de correlação espacial e temporal, chamada de *Spatiotemporal Clus-*

*tering and Compressing Schemes - (SCCS)*. O algoritmo SCCS é dividido em duas etapas. Na primeira, os nós que detectam o evento são agrupados com o objetivo de realizar correlação espacial. Na segunda etapa, os nós representativos armazenam suas leituras em um *buffer* e quando este estiver cheio, o nó executa o algoritmo de divisão e conquista (DCA) para encontrar dados representativos. Considerando todos os dados no *buffer*, o algoritmo DCA cria uma reta que passa pelo primeiro e último elementos. Para cada dado do *buffer*, o DCA calcula a distância entre esse dado e a reta criada. Se a distância for menor do que um erro pré-definido, o dado não é considerado. Se a distância for maior, a reta inicialmente criada é dividida em dois segmentos de reta e o processo é reiniciado e todos os dados do *buffer* são novamente considerados. O algoritmo pára quando nenhuma reta é subdividida. Quando isso acontece, apenas os pontos finais de cada segmento de reta são enviados para o *Sink*. Como mencionado anteriormente, a desvantagem dessa abordagem é que o nó tem que esperar até que o *buffer* esteja cheio para executar o algoritmo DCA.

### 3. Algoritmo Proposto para Coleta de Dados em RSSFs

Nessa seção, os modelos de correlação espacial e temporal utilizados na solução proposta são definidos. Além disso, é apresentado o algoritmo proposto, chamado *Efficient Data Collection Aware of Spatio-Temporal Correlation* (EAST).

#### 3.1. Modelo de Correlação Espacial

Como descrito anteriormente, nós espacialmente próximos tendem a sensorar valores semelhantes. No entanto, essa proximidade ( $c$ ) depende dos requisitos da aplicação e características do evento. Algumas aplicações são mais críticas e são menos tolerantes a discrepâncias nos valores sensorados sobre o fenômeno observado, exigindo que nós próximos reportem os dados sensorados (região de correlação é menor). Enquanto outras aplicações podem ser mais tolerantes a discrepâncias nos valores sensorados, não exigindo que nós próximos reportem os dados sensorados (região de correlação é maior).

**Definição 3.1 (região de correlação)** Nós definimos uma região como região de correlação, onde os valores sensorados pelos nós sensores são considerados semelhantes para a aplicação e, portanto, uma única leitura dentro dessa região é o suficiente para representá-la. O tamanho da região de correlação varia de aplicação para aplicação e de evento para evento. Assim, o tamanho da região de correlação está diretamente relacionado com a aplicação.

#### 3.2. Modelo de Correlação Temporal

Tipicamente, a leitura feita pelos sensores no ambiente é periódica. Conseqüentemente, os dados sensorados constituem uma série temporal. Devido a natureza do fenômeno físico, há uma correlação temporal significativa entre cada observação consecutiva de um nó sensor, e os dados recolhidos são geralmente semelhantes em um curto período de tempo. Assim, nestes casos, os nós sensores não precisam transmitir suas leituras se a leitura atual está dentro de um limiar aceitável em relação à última leitura reportada.

**Definição 3.2 (correlação temporal)** Cada nó fonte mantém a última leitura reportada ( $R_{old}$ ). Quando a leitura atual ( $R_{new}$ ) está disponível,  $R_{new}$  é comparada com  $R_{old}$ .  $R_{new}$  de um nó fonte é reportada se um dado limiar é maior que a tolerância na coerência temporal ( $tct$ ), isto é,  $\left(\frac{|R_{new}-R_{old}|}{R_{old}}\right) \times 100 > tct$ , onde  $tct$  é a porcentagem de tolerância na coerência temporal. Caso contrário o valor  $R_{new}$  é suprimido.

### 3.3. Visão Geral do Algoritmo EAST

A idéia principal do algoritmo proposto é gerenciar o consumo de energia dos nós que detectam eventos eliminando notificações redundantes. O algoritmo proposto considera os seguintes papéis para realizar o roteamento dos dados (veja Figura 1):

- *Nó Membro*: Um nó que esta detectando o evento.
- *Nó Representativo*: Um nó que esta detectando o evento e reportando os dados coletados para um o nó coordenador representando todos os nós da região de correlação. Além disso, o nó executa o mecanismo de correlação temporal.
- *Nó Coordenador*: Um nó responsável por reunir todos os dados coletados pelos nós representativos. Processa os dados recebidos e envia o resultado para o *sink*.
- *Nó Retransmissor*: Um nó que encaminha os dados para o nó *Sink*.
- *Nó Sink*: O gateway entre a RSSF e a instalação de monitoramento.

O algoritmo proposto usa rotas mais curtas (em distância euclidiana) em dois diferentes níveis para encaminhar os dados coletados para o nó *Sink*. No primeiro nível, nós representativos usam rotas mais curtas para encaminhar os dados coletados para os nós coordenadores. No segundo nível, os nós coordenadores usam rotas mais curtas para encaminhar dados para o nó *Sink*. A Figura 1 mostra dois exemplos de estruturas de roteamento criadas pelo EAST (O campo cinza indica a área do evento, as células representam as regiões de correlação e a linha pontilhada vermelha mostra a rota mais curta).

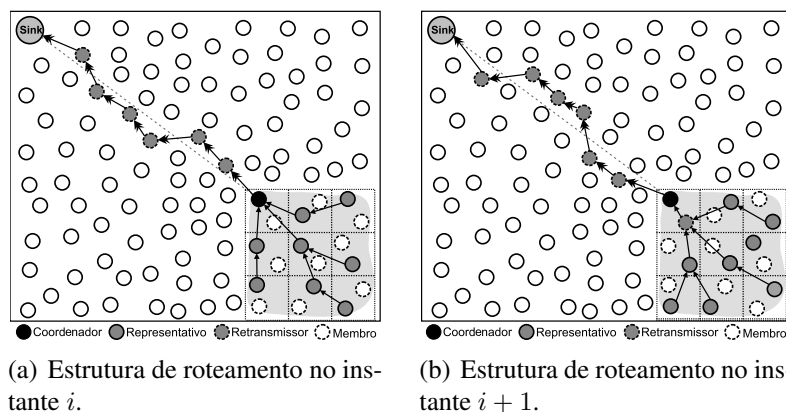
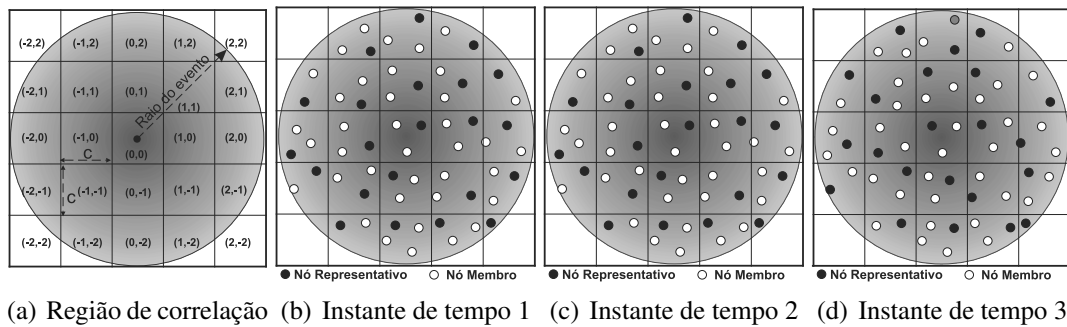


Figura 1. Exemplos de estruturas de roteamento criadas pelo EAST.

O principal objetivo do algoritmo proposto é reduzir o consumo de energia na coleta de dados enquanto preserva a precisão dos dados. Para alcançar esse objetivo, o EAST pode alterar dinamicamente o tamanho da região de correlação e o valor da tolerância na coerência temporal de acordo com as características do evento/aplicação. Para isso, a área do evento é dividida em células, como mostra a Figura 2. Cada célula define uma região de correlação e os nós dentro de cada célula são assumidos ser espacialmente correlacionados. Assim, somente um nó dentro de cada célula reporta os dados sensorados e este nó é considerado o nó representativo da região de correlação. As células são consideradas independentes. Assim, a troca do nó representativo em uma célula não exige nenhuma reconfiguração. O revezamento de nós representativos em cada célula é feito para equilibrar o consumo de energia de nós correlacionados espacialmente. Além disso, cada nó



**Figura 2. Mecanismo de correlação espacial aplicado na área do evento**

representativo executa o algoritmo de correlação temporal proposto localmente sem se comunicar com seus vizinhos.

O algoritmo EAST é realizado em 3 fases. Em todas as comunicações realizadas os pacotes têm um campo chamado nível de energia que armazena o nível de energia do nó que transmitiu o pacote. Ao receber essas transmissões, os nós vizinhos atualizam na tabela de roteamento o nível de energia do nó que transmitiu o pacote. Na Fase 1, apresentada na Seção 3.4, nós sensores armazenam a posição do *Sink* e a posição dos nós vizinhos. Na Fase 2, apresentada na Seção 3.5, três diferentes ações são executadas: formação de cluster, eleição do coordenador e a divisão da área do evento em regiões de correlação. Finalmente, na Fase 3, apresentada na Seção 3.6, nós representativos são escolhidos, o mecanismo de correlação temporal é aplicado e os dados são transmitidos.

### 3.4. Descoberta da posição dos nós vizinhos e do Sink

Esta é a primeira fase do Algoritmo EAST. Como mostra o Algoritmo 1, o nó *Sink* inicia essa fase transmitindo a Mensagem de Configuração. Esta mensagem contém três campos: ID, CoordSender e CoordSink, onde ID é a identificação do nó que retransmitiu a mensagem de configuração, CoordSender é a posição geográfica  $(x_n, y_n)$  do nó que retransmitiu a mensagem de configuração, e CoordSink é a posição geográfica  $(x_s, y_s)$  do *Sink*.

Nessa fase, nós sensores armazenam informações como identificação, posição geográfica dos vizinhos e do *Sink* em uma tabela de vizinhos *neighborhood*. Estas informações são usadas no encaminhamento dos dados coletados para o *Sink*.

### 3.5. Formação de Cluster, Eleição do Coordenador e Divisão da área do Evento em Regiões de Correlação

A segunda fase do algoritmo EAST inicia sempre que um evento acontece. Assim, quando os nós detectam um evento no ambiente monitorado, esses nós iniciam o algoritmo de eleição do coordenador e se candidatam a nó coordenador do *cluster*. Esse processo é descrito no Algoritmo 2. Na eleição do coordenador do grupo, todos os nós são elegíveis (Linhas 3 e 4 do Algoritmo 2). Porém, é eleito apenas o nó que tem a menor distância para o *Sink*. Quando há empate, isto é, dois ou mais nós concorrentes têm a mesma distância para o *Sink*, apenas o nó com maior nível de energia continua elegível. Ao final do algoritmo de eleição de coordenador só existe um coordenador no grupo. Todos os

**Algoritmo 1:** Descoberta da posição dos nós vizinhos e do Sink

---

```

1  Nó Sink transmite a Mensagem de Configuração;
   //  $R_u$  é o conjunto de nós que recebeu a Mensagem de Configuração
2  para cada  $u \in R_u$  faça
3      $neighborhood(u).ID \leftarrow ID_{Mensagem}$ ;
4      $neighborhood(u).CoordSender \leftarrow CoordSender_{Mensagem}$ ;
5      $neighborhood(u).CoordSink \leftarrow CoordSink_{Mensagem}$ ;
6     se  $Nó(u)$  não transmitiu sua posição então
7         //  $Nó u$  atualiza o valor de  $ID$  na Mensagem de Configuração
            $ID_{Mensagem} \leftarrow ID_u$ ;
           //  $Nó u$  atualiza o valor de  $CoordSender$  na Mensagem de Configuração
8          $CoordSender_{Mensagem} \leftarrow Coordinates_u$ ;
9          $Nó u$  transmite a Mensagem de Configuração com os novos valores;
10    fim se
11 fim para

```

---

outros nós que detectaram o mesmo evento são nós com o papel de nós membro (em cada notificação, um subconjunto de nós membro será escolhido como nós representativos).

**Algoritmo 2:** Formação do *Cluster* e Eleição do Coordenador

---

```

1  Entrada:  $S$  // Conjunto de nós que detectou o evento
2  Saída:  $u$  // Um nó do conjunto  $S$  é eleito como coordenador do grupo
3  para cada  $u \in S$  faça
4      $papel_u \leftarrow coordenador$ ;
       //  $Nó u$  transmite a Mensagem de Configuração de Cluster
5     Anúncio de detecção de evento;
       //  $N_u$  é o conjunto de vizinhos do nó  $u$ 
6     para cada  $w \in N_u$  faça
7         se  $DistanciaParaSink(u) > DistanciaParaSink(w)$  então
8              $papel_u \leftarrow colaborador$ ;
9              $Nó u$  retransmite a mensagem recebida do nó  $w$ ;
10        fim se
11        senão se  $DistanciaParaSink(u) = DistanciaParaSink(w) \wedge$ 
            $NivelEnergia(u) < NivelEnergia(w)$  então
12            //  $Nó u$  troca o seu papel para colaborador
            $papel_u \leftarrow colaborador$ ;
13             $Nó u$  retransmite a Mensagem de Configuração de Cluster recebida do nó  $w$ ;
14        fim se
15    fim para
16 fim para

```

---

Após o agrupamento dos nós que detectam o mesmo evento, o mecanismo de correlação espacial é executado. A Figura 2(a) ilustra o mecanismo de correlação espacial proposto. Por uma questão de simplificação, o formato do evento considerado é um círculo, mas a solução proposta funciona para qualquer formato de evento. A região do evento é decomposta em  $\left(\frac{2r_e}{c}\right)^2$  regiões de correlação. A Figura 2(a) mostra um exemplo em qual a região do evento é decomposta em 25 regiões de correlação. Cada região de correlação é representada por um par ordenado  $(x_c, y_c)$ . Cada nó calcula o valor de  $x_c$  e  $y_c$  da região de correlação em que pertence. Para esse cálculo é necessário as coordenadas  $(x_n, y_n)$  do nó, as coordenadas  $(x_e, y_e)$  do centro do evento e o tamanho da região de correlação ( $c$ ). Para calcular os valores  $x_c$  e  $y_c$  da região de correlação em que o nó pertence, os passos mostrados no Algoritmo 3 são executados.

O algoritmo EAST seleciona um único nó representativo em cada região de correlação de dimensão  $c^2$  para cada notificação. As Figuras 2(b), 2(c), e 2(d) mostram

---

**Algoritmo 3:** Cálculo dos valores  $x_c$  e  $y_c$  da região de correlação em qual o nó pertence
 

---

```

1 Entrada:  $S$  // Conjunto de nós que detectou o evento,  $(x_n, y_n)$  // Coordenadas do
  nó,  $(x_e, y_e)$  // Coordenadas do centro do evento e  $(c)$  // Tamanho da região de
  correlação
2 Saida:  $x_c$  e  $y_c$  // valores de  $x_c$  e  $y_c$  no par ordenado  $(x_c, y_c)$  da região de correlação
  em qual o nó pertence
3 para cada  $n \in S$  faça
4    $x_c \leftarrow 0$  //  $x_c$  do nó  $n$  é inicializado como região de correlação central
5    $y_c \leftarrow 0$  //  $y_c$  do nó  $n$  é inicializado como região de correlação central
  // Nó  $n$  verifica se ele pertence a uma região de correlação à direita da região de correlação
  central
6   se  $\frac{(x_n - x_e)}{(\frac{c}{2})} > 1$  então
  // Nó  $n$  calcula o valor  $x_c$  e  $y_c$  da região de correlação que ele pertence
7      $x_c \leftarrow \lfloor \frac{(x_n - x_e) - (\frac{c}{2})}{c} \rfloor + 1$ 
8      $y_c \leftarrow \lfloor \frac{(y_n - y_e) - (\frac{c}{2})}{c} \rfloor + 1$ 
9   fim se
  // Nó  $n$  verifica se ele pertence a uma região de correlação à esquerda da região de correlação
  central
10  senão se  $\frac{(x_n - x_e)}{(\frac{c}{2})} < -1$  então
  // Nó  $n$  calcula o valor  $x_c$  e  $y_c$  da região de correlação que ele pertence
11     $x_c \leftarrow \lfloor \frac{(x_n - x_e) + (\frac{c}{2})}{c} \rfloor - 1$ 
12     $y_c \leftarrow \lfloor \frac{(y_n - y_e) + (\frac{c}{2})}{c} \rfloor - 1$ 
13  fim se
14 fim para

```

---

nós representativos em tempos diferente na região de correlação. Os nós representativos no conjunto de nós membros são os nós que tem o maior nível de energia entre os nós pertencente a mesma região de correlação. Isso garante a distribuição do consumo de energia na rede.

### 3.6. Transmissão dos Dados Coletados

Após o nó calcular a região de correlação em que pertence, a cada leitura realizada, o nó verifica se é um nó representativo e também se o limiar é maior que a tolerância na coerência temporal (Linha 3 do Algoritmo 4).

---

**Algoritmo 4:** Exploração da correlação temporal e transmitindo os dados
 

---

```

1 Entrada:  $S$  // Conjunto de nós que detectou o evento,  $R_{new}$  // É a leitura atual,
   $R_{old}$  // É a ultima leitura notificada e  $tct$  // é a tolerância na coerência
  temporal
2 para cada  $n \in S$  faça
3   se  $\text{pap}_{el}_n = \text{Representativo e } (\frac{|(R_{new} - R_{old})|}{R_{old}}) \times 100 > tct$  então
4      $R_{new} \leftarrow$  valor sensorado;
5     Envia  $R_{new}$  para o coordenador;
6   fim se
7   se  $\text{pap}_{el}_n = \text{Membro}$  então
8     Encaminha  $R_{new}$  para o coordenador;
9   fim se
10  se  $\text{pap}_{el}_n = \text{Coordenador}$  então
11    Processa a mensagem  $R_{new}$  recebida;
12    Encaminha o resultado para o nó Sink;
13  fim se
14  se  $\text{pap}_{el}_n = \text{Retransmissor}$  então
15    Encaminha a mensagem  $R_{new}$  recebida para o nó Sink;
16  fim se
17 fim para

```

---



Se ambas as condições são satisfeitas, então o valor sensorado ( $R_{rew}$ ) é enviado para o coordenador do grupo. O coordenador do grupo envia os dados coletados para o *Sink* usando o menor caminho. O menor caminho é calculado com base nas informações obtidas durante a Fase 1. Com base na sua posição e na posição do *Sink*, o nó coordenador cria um segmento de reta que o liga ao *Sink*. Quando a transmissão de dados é realizada, os nós mais próximos do segmento de reta e do ponto final deste segmento de reta serão escolhidos para transmitir os dados. A Figura 1 mostra um segmento de reta entre o coordenador e o nó *Sink*, bem como os nós retransmissores. A avaliação do nosso algoritmo é apresentada na próxima seção.

#### 4. Avaliação de Desempenho do Algoritmo Proposto

Nessa seção, nós avaliamos o desempenho do algoritmo proposto EAST. Nós também comparamos o desempenho com dois outros conhecidos algoritmos de roteamento:

- *Spatiotemporal Clustering and Compressing Schemes - SCCS* (brevemente descrito na Seção 2).
- Estratégia para coleta de dados precisos, qual é a solução ótima em termos de precisão. Nesta solução, todos os nós reportam as leituras sensoradas para o nó *Sink*.

A principal proposta da comparação é avaliação do algoritmo proposto considerando as seguintes métricas:

- *Número de notificações* enviadas pelos nós que detectam evento.
- *Número de leituras reportadas* pelos nós que detectam evento.
- *Leituras por pacote de dados*: Número médio de leituras dentro de cada pacote (é a taxa de leitura reportada e notificações).
- *Consumo de energia*: A quantidade de energia consumida por sensores que detectam evento.
- *Precisão dos dados*: A precisão da informação do fenômeno observado comparado a informação original.
- *Atraso na notificação*: Tempo para entregar os dados coletados.

##### 4.1. Metodologia

A avaliação é realizada através de simulações utilizando o simulador SinalGo v.0.75.3 [Sinalgo 2008]. Cada simulação foi replicada 33 vezes. Em todos os resultados, as curvas representam os valores médios, enquanto que as barras de erros representam o intervalo de confiança de 95%. Os parâmetros de simulação são apresentados na Tabela 1. Os eventos ocorrem em posições aleatórias e os nós sensores são depositados de forma aleatória.

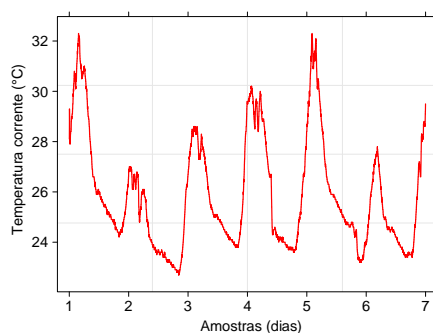
Em todos os casos avaliados, avaliamos nosso algoritmo EAST explorando apenas a correlação temporal. Nós também avaliamos o nosso algoritmo (EAST-15, EAST-30 e EAST-45) explorando tanto correlação temporal e correlação espacial (regiões de correlação de tamanhos 15, 30 e 45). Para o algoritmo SCCS, consideramos *buffers* com diferentes capacidades de armazenamento (25, 50, 100 e 200 leituras).

**Tabela 1. Parâmetros de Simulação**

Parâmetros	Valores
Nó Sink	1 (canto superior esquerdo)
# de nós	1024
# de eventos	1
Densidade (número médio de vizinhos)	25
Diâmetro do evento (m)	(50, 100, 150, 200)
Região de Correlação (c)	(15, 30, 45)
Tolerância na coerência temporal	(0.5, 1, 2, 3, 4)
Tamanho do <i>buffer</i> (leituras)	(25, 50, 100, 200)
Duração do evento (horas)	(1 to 10)
Taxa de notificação (por minuto)	1
Raio de comunicação (m)	80
Duração da simulação (dias)	7
Campo sensorado (m)	900 × 900

## 4.2. Modelo de Evento

Para o nosso modelo de evento, foi utilizado um conjunto de dados de temperatura do ambiente (graus Celsius) de uma semana. Os dados foram coletados da floresta amazônica no Brasil no intervalo de 1 minuto. As amostras são mostrados na Figura 3 [Horacio 2011].



**Figura 3. Dados coletados da floresta Amazônica.**

### 4.2.1. Notificações e Leituras Reportadas

Para essa análise, os parâmetros tolerância na coerência temporal, tamanho do *buffer* e região de correlação (apresentada na Tabela 1) foram todos variados para avaliar o impacto sobre o número de leituras que podem ser eliminadas quando a correlação espaço-temporal é explorada.

A Figura 4(a) mostra que quando a tolerância na coerência temporal aumenta, o número de notificações realizadas pelo EAST diminui enquanto o número de notificações realizadas pelo SCCS permanece fixo uma vez que as leituras são transmitidas somente quando o *buffer* estiver cheio. Conseqüentemente, no SCCS, o número de notificações depende do tamanho do *buffer* e não da tolerância na coerência temporal.

Na Figura 4(b), nós podemos ver que o número de leituras reportadas pelo algoritmo proposto EAST é similar ao número de notificações apresentado na Figura 4(a). Isso ocorre porque sempre que a diferença da leitura atual e a última leitura reportada são maiores que a tolerância na coerência temporal, os dados são notificados. Note que na

maioria dos casos o algoritmo proposto EAST reporta menos leituras do que o algoritmo SCCS. Mas para pequenos valores de tolerância na coerência temporal, o SCCS realiza menos notificações devido ao uso de um *buffer*, em qual cada notificação pode conter mais do que uma leitura.

A Figura 4(c) mostra o número médio de leituras dentro de cada pacote de dados transmitido. O algoritmo EAST, envia somente uma leitura dentro de cada pacote de dados transmitido. Mas o número de leituras dentro do pacote de dados no algoritmo SCCS depende tanto da capacidade do *buffer* quanto da tolerância na coerência temporal.

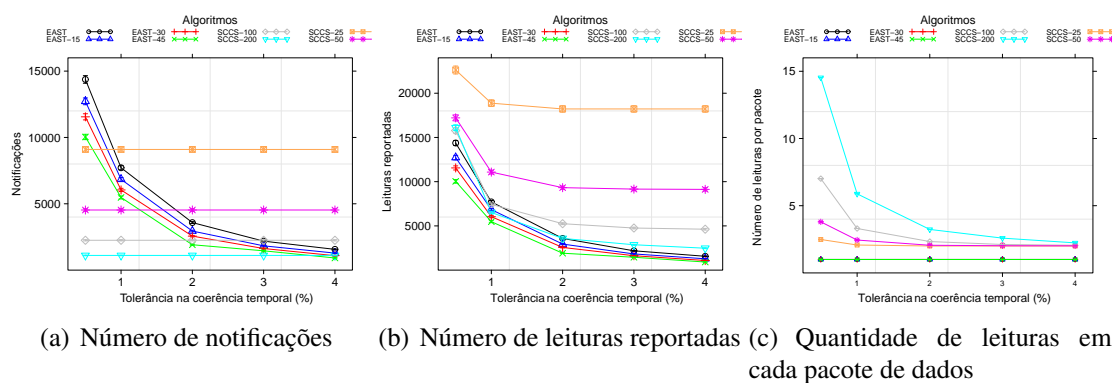


Figura 4. Notificações  $\times$  Leituras.

#### 4.2.2. Consumo de Energia e Precisão dos Dados

Para essa análise, os parâmetros tolerância na coerência temporal, tamanho do *buffer* e tamanho da região de correlação (apresentados na Tabela 1) foram todos variados para avaliar o impacto no consumo de energia e a precisão dos dados.

Como mostrado na Figura 4(a), quando a tolerância na coerência temporal aumenta, o número de notificações realizadas pelo EAST diminui enquanto o número de notificações no SCCS permanece fixo, qual tem um impacto na precisão dos dados (veja a Figura 5(b)). Conseqüentemente, o consumo de energia também diminui no algoritmo proposto EAST (veja a Figura 5(a)). Além disso, quando o tamanho da região de correlação aumenta (15, 30 e 45), o consumo de energia também diminui, uma vez que o número de nós reportando as leituras é menor.

Os resultados para a estratégia de coleta de dados precisos não foram mostrados no gráfico por causa de seu consumo de energia muito alto, o que é devido ao fato de que todas as leituras são reportadas para garantir a precisão dos dados de 100%. Na média, a estratégia de coleta de dados precisos consome  $10J$ , isto é, 14 vezes mais do que os algoritmos SCCS e EAST. Para os cenários avaliados, a menor precisão observada no algoritmo proposto EAST foi de 98.7% e o consumo de energia foi menor que  $0.2J$ , enquanto a estratégia de coleta de dados precisos consome  $10J$ , qual indica as vantagens de usar nossas técnicas de correlação espaço-temporal.

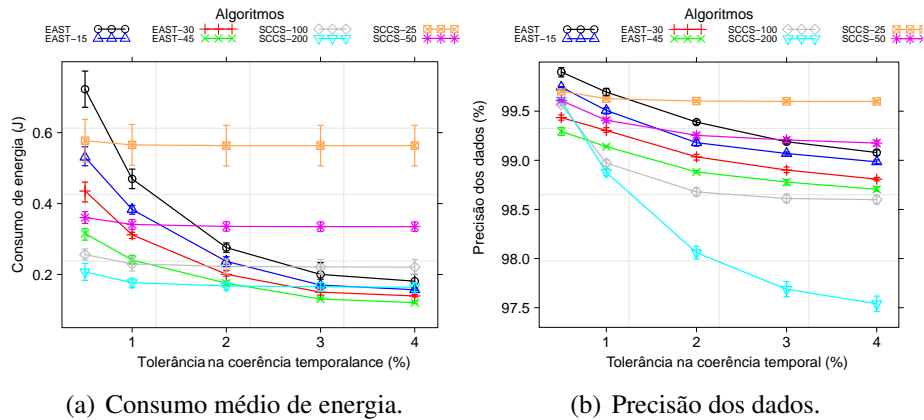


Figura 5. Consumo de energia  $\times$  Precisão dos dados.

#### 4.2.3. Atraso Médio na Notificação das Leituras

Nessa Seção, nós avaliamos o atraso médio na notificação das leituras para cenários de baixa taxa de notificação e cenários com alta taxa de notificação, quais são mostrados nas Figuras 6(a) e 6(b), respectivamente. Para esta análise, a taxa de notificação foi variada para avaliar o atraso médio em relatar as leituras. Como mencionado anteriormente, o algoritmo SCCS explora o uso de um *buffer* para armazenar as leituras e, então, as leituras são processadas e o mecanismo de correlação temporal é aplicado. No entanto, o uso de um *buffer* tem algumas desvantagens. A principal desvantagem desta técnica é o atraso da notificação de cada dado coletado.

As Figuras 6(a) e 6(b) mostram que o atraso para notificar os dados sensorados é muito alto no SCCS. Como esperado, quanto maior o tamanho do *buffer* ou a taxa de notificação, maior será o atraso para notificar os dados sensorados. Por exemplo, para cenários onde o SCCS usa *buffer* de tamanho 100 para leituras e uma notificação por segundo (veja a Figura 6(b)), o nó envia os valores somente depois de 100 leituras, o que implica em um atraso médio próximo a 50 segundos. Assim, o nó *Sink* será notificado sobre o evento com leituras antigas, o que não é aceitável por várias aplicações RSSFs.

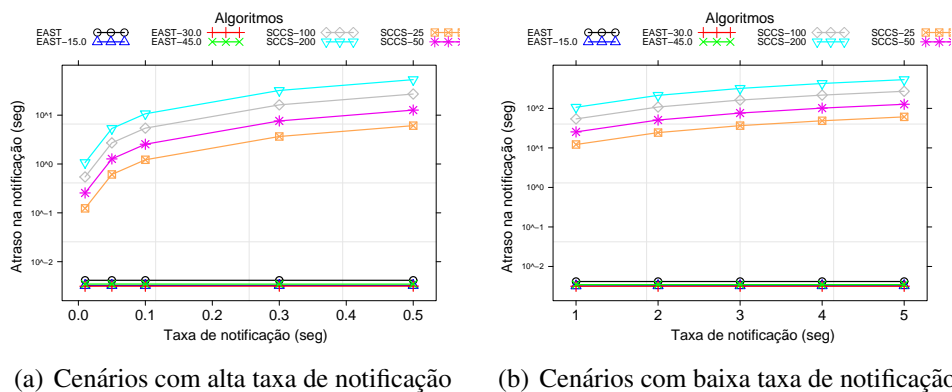


Figura 6. Atraso médio na notificação das leituras.

## 5. Conclusões e Trabalhos Futuros

Na literatura atual de algoritmos que exploram a correlação espacial e/ou temporal, a maioria das abordagens propostas não considera o nível de energia dos nós na seleção dos nós representativos e as que consideram o nível de energia dos nós têm um alto custo de controle,. No entanto, elas não exploram a correlação espaço-temporal eficientemente. Neste trabalho, propomos um algoritmo ciente de energia para coleta de dados em RSSFs que tira proveito das correlações espacial e temporal dos dados. Além disso, o algoritmo mantém as notificações em tempo real com dados precisos e atualizados para o nó *Sink*.

Os resultados obtidos mostram que usando o EAST para coletar dados, as informações sobre o evento podem ser detectadas com alta precisão e poupando energia residual dos nós quando comparado com a abordagem clássica (estratégia de coleta de dados precisos) e o algoritmo SCCC.

Como trabalhos futuros pretendem-se considerar não apenas a última leitura, mas também as leituras anteriores na região de correlação para melhorar a precisão dos dados sobre o fenômeno observado. Para atingir esse objetivo, nós representativos pode-se estimar os valores de sua região de correlação, tendo em conta o tempo e as leituras recentes. Além disso, temos a intenção de considerar regiões de correlação com diferentes tamanhos para o mesmo evento para explorar ainda mais o dinamismo do evento.

## Referências

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.
- Akyildiz, I. F., Vuran, M. C., and Özgür B. Akan (2004). On exploiting spatial and temporal correlation in wireless sensor networks. In *In Proceedings of WiOpt 2004: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 71–80.
- Boukerche, A. (2008). *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press.
- Deligiannakis, A. and Kotidis, Y. (2008). Geosensor networks. In Nittel, S., Labrinidis, A., and Stefanidis, A., editors, *Book chapter: Exploiting Spatio-temporal Correlations for Data Processing in Sensor Networks*, pages 45–65. Springer-Verlag.
- Fan, K.-W., Liu, S., and Sinha, P. (2006). On the potential of structure-free data aggregation in sensor networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12.
- Horacio (2011). Data collected of Amazon forest in Brazil, last visited in May, 2011.
- Le, T. D., Pham, N. D., and Choo, H. (2008). Towards a distributed clustering scheme based on spatial correlation in wsns. In *International Conference on Wireless Communications and Mobile Computing Conference*, pages 529–534.
- Liu, C., Wu, K., and Pei, J. (2007a). An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. *IEEE Transactions on Parallel and Distributed Systems*, 18(7):1010–1023.
- Liu, L., Member, S., and Yu, P. S. (2007b). Asap: An adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2007:1766–1783.

- Min, J.-K. and Chung, C.-W. (2010). Edges: Efficient data gathering in sensor networks using temporal and spatial correlations. *J. Syst. Softw.*, 83:271–282.
- Nakamura, E. F., de Oliveira, H. A. B. F., Pontello, L. F., and Loureiro, A. A. F. (2006). On demand role assignment for event-detection in sensor networks. In *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 941–947, Washington, DC, USA. IEEE Computer Society.
- Pham, N. D., Le, T. D., Park, K., and Choo, H. (2008). Enhance exploring temporal correlation for data collection in wsns. In *Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future*, RIVF '08, pages 204–208. IEEE.
- Pham, N. D., Le, T. D., Park, K., and Choo, H. (2010). Sccs: Spatiotemporal clustering and compressing schemes for efficient data collection applications in wsns. *International Journal of Communication Systems*, 23:1311–1333.
- Romer, K. and Mattern, F. (2004). The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61.
- Sinalgo (2008). Simulator for network algorithms. Distributed Computing Group - ETH-Zurich, last visited in October, 2008.
- Villas, L. A., Boukerche, A., Araujo, R. B., and Loureiro, A. A. (2009). A reliable and data aggregation aware routing protocol for wireless sensor networks. In *MSWiM '09: Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 245–252, New York, NY, USA. ACM.
- Villas, L. A., Boukerche, A., de Oliveira, H. A., de Araujo, R. B., and Loureiro, A. A. (2011). A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks. *Ad Hoc Networks*, (0):–.
- Villas, L. A., Guidoni, D. L., Araújo, R. B., Boukerche, A., and Loureiro, A. A. (2010). A scalable and dynamic data aggregation aware routing protocol for wireless sensor networks. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, MSWIM '10, pages 110–117, New York, NY, USA. ACM.
- Yoon, S. and Shahabi, C. (2005). Exploiting spatial correlation towards an energy efficient clustered aggregation technique (cag) [wireless sensor network applications]. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 5, pages 3307 – 3313 Vol. 5.
- Yuan, J. and Chen, H. (2009). The optimized clustering technique based on spatial-correlation in wireless sensor networks. In *IEEE Youth Conference on Information, Computing and Telecommunication, 2009.*, pages 411–414.
- Zhang, Y., Meratnia, N., and Havinga, P. (2010). Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 12(2):159–170.