

## Agrupamento Dinâmico de Sensores Baseado na Similaridade de Leitura de Dados

Fernando Henrique Gielow<sup>1</sup>, Aldri L. dos Santos<sup>1</sup>

<sup>1</sup>NR2 – Departamento de Informática – Universidade Federal do Paraná  
Caixa Postal 19.081 – 81.531-980 – Curitiba – PR – Brasil

{fhgielow,aldri}@inf.ufpr.br

**Abstract.** *Wireless Sensor Networks (WSNs) are an important interface between physical and computational environments, where the logical clustering of sensor nodes is a commonly used technique to organize traffic. Although current clustering protocols treat various kinds of dynamicity on the network, such as mobility or leader rotations, few solutions consider the readings similarity, which would provide benefits in terms of better use of compression techniques and easier reactive detection of anomalous events. This paper proposes a dynamic clustering protocol that handles spatial similarity between nodes readings, called DDFC. Its operation is based on the biological principles of fireflies to ensure distributed synchronization of the cluster's similar readings aggregations, differentiating thus from the classic use of fireflies. Simulations show that DDFC is capable of maintaining the cluster's readings aggregation synchronized, thus clustering nodes dynamically according to their similar readings.*

**Resumo.** *As Redes de Sensores Sem Fio (RSSFs) são uma interface importante entre o ambiente físico e o computacional. Nelas, o agrupamento lógico dos nós sensores é usado para organizar o tráfego da rede. Embora os protocolos de agrupamento atuais tratem várias dinamicidades na rede, como a de mobilidade ou de líderes, são poucas as soluções que consideram a similaridade dinâmica das leituras dos nós, que traria vantagens como o uso mais eficiente de técnicas de compressão e melhor detecção reativa de eventos anômalos. Este trabalho propõe um protocolo de agrupamento dinâmico que trata a relação de similaridade espacial entre as leituras dos nós da RSSF, chamado DDFC. O seu funcionamento é inspirado nos princípios biológicos de vagalumes para garantir a sincronização distribuída da agregação de leituras similares nos agrupamentos, se diferenciando do uso clássico de vagalumes. Simulações demonstram que o protocolo é capaz de manter a agregação de leituras dos agrupamentos lógicos sincronizadas, agrupando dinamicamente os nós de leituras similares.*

### 1. Introdução

Atualmente, é comum no meio urbano o uso de sensores e radares para a detecção de velocidade de veículos, ou mesmo para saber se eles estão indevidamente parados sobre faixas de pedestres. Contudo, tais aplicações ainda são primitivas por usufruírem apenas da interpretação *singular e individual* dos dados, sem o estabelecimento de relações entre eles. As relações mais usuais entre leituras de dados do meio são as relações espaciais e temporais [Yoon and Shahabi 2007]. Para diversas grandezas naturais, como temperatura, umidade e luz, tais medições tem tendência de serem parecidas quando tomadas em

regiões próximas, devido à sua **relação espacial**. Ademais, medições sucessivas em uma localidade individual tem tendência de variar gradualmente, possuindo **relação temporal**.

Ao explorar e analisar os dados de maneira *coletiva*, considerando suas possibilidades de relações, aplicações mais robustas podem ser vislumbradas e desenvolvidas. Em um cenário urbano, por exemplo, atualmente os dados vistos coletivamente possibilitam verificar o tráfego nas ruas para se determinar rotas ótimas, ou mesmo analisar padrões espaciais de temperatura a fim de localizar ilhas de calor [Murty et al. 2008]. Com as leituras de luz durante a noite, seria possível avaliar o perigo que determinadas ruas podem oferecer. Leituras de som permitiriam determinar o nível de poluição sonora em dadas regiões ou mesmo determinar como é a propagação do som no meio.

Embora as Redes de Sensores Sem Fio (RSSFs) sejam uma solução existente há diversos anos, seu uso ainda não alcançou o seu potencial máximo quanto à coleta de dados [Partridge 2011]. As RSSFs servem como uma interface de comunicação entre o meio físico e o meio computacional, formado virtualmente por diversos conjuntos de dados. Assim, elas são interface essencial para os *Cyber-Physical Systems* [Rajkumar et al. 2010] e para o advento da Internet das Coisas [Ma 2011, López et al. 2012].

Uma técnica muito utilizada para a organização lógica dos nós nas RSSFs é o agrupamento destes nós [Banerjee et al. 2011]. Essas formações, denominadas *clusters*, correspondem a grupos lógicos hierárquicos entre os nós. Os agrupamentos são muito utilizados por possibilitar agregação de dados e organizar o tráfego de mensagens na rede [Dechene et al. 2007]. Ademais, utilizando técnicas de agrupamento, a organização lógica que mantém os nós com leituras similares traria vantagens como a possibilidade de agregar os dados de maneira mais eficiente, devido à maior similaridade, além de possibilitar a detecção mais robusta de eventos anômalos [Reis et al. 2007].

Entretanto, pouca pesquisa acerca de protocolos de agrupamento que tratem simultaneamente a correlação e a variação dos dados foi desenvolvida em abordagens relativas até o momento. Os protocolos de agrupamento tem sido propostos com diversos objetivos: Alguns visam **se adequar à dinamicidade da mobilidade** [Islam et al. 2011, Brust et al. 2008], outros tentam até mesmo **recriar formações de agrupamentos por completo** [Guo and Li 2007, Villas et al. 2011]. Contudo, poucos são os que **consideram a similaridade espacial dos dados** [Wu et al. 2008, Yoon and Shahabi 2007] e, menos ainda, os que **suportam a natureza dinâmica dos dados** em uma abordagem também dinâmica de agrupamento [Pham et al. 2010]. Assim, se faz necessária uma maneira distribuída e robusta de manter agrupamentos lógicos de nós sensores com leituras similares.

Este trabalho propõe um protocolo de agrupamento lógico de nós sensores que considera a similaridade das leituras realizadas pelos nós, chamado DDFC (*Dynamic Data-aware Firefly-based Clustering*). O protocolo agrupa os nós de leituras similares, tendo como base de funcionamento os princípios biológicos dos vaga-lumes. O DDFC sincroniza agregações de leituras similares nos agrupamentos, suportando a sua manutenção dinâmica e seu roteamento interno. Simulações mostram a eficiência do DDFC para manter os nós de leituras similares agrupados, assim como eleger líderes adequados.

O artigo está organizado desta forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha o funcionamento do DDFC. A Seção 4 mostra a sua avaliação do desempenho. Finalmente, a Seção 5 apresenta as conclusões e os trabalhos futuros.

## 2. Trabalhos Relacionados

As RSSFs são dinâmicas de diversas maneiras, em questões de topologia, rotas e posicionamento dos nós. Desta forma, os mecanismos de agrupamento devem se adequar, sendo adaptativos e reconfiguráveis. Existe a **dinamicidade pela mobilidade**, que visa manter agrupamentos enquanto nós transitam arbitrariamente pela rede, havendo a necessidade de mecanismos específicos para estes cenários. O protocolo SPRP<sub>G</sub> [Islam et al. 2011] estabelece uma árvore geradora, criando nós líderes e *gateway* a fim de conectar os seus agrupamentos dentro dela. Por sua vez, o protocolo KHOPCA [Brust et al. 2008], inspirado no jogo da vida, opera proativamente através de um conjunto simples de regras que define agrupamentos de  $k$ -saltos até os líderes estabelecidos nesse processo.

Levando em consideração a complexidade e o custo de manter uma estrutura hierárquica de maneira dinâmica e proativa, algumas abordagens optam atender aos requisitos de **dinamicidade por recriação** completa dos agrupamentos, seja ela periódica ou reativa. O mecanismo DCRR [Guo and Li 2007] considera que agrupamentos em uma rede dinâmica são de relevância apenas quando há a detecção de algum evento, sendo que manter a estrutura de agrupamentos de maneira contínua na rede é dispendioso. Da mesma forma, o ESC [Villas et al. 2011] coordena os nós na detecção de um evento relevante para que, com a eleição líderes em células espaciais, não sejam enviadas informações redundantes à estação-base. Entretanto, abordagens que dependam do reagrupamento da rede como um todo introduzem custos em termos de latência e energia.

Destas abordagens, nenhuma oferece suporte apropriado ao agrupamento através da similaridade de dados. Considerando a **similaridade de dados**, e de maneira similar ao DCRR, o CAG [Yoon and Shahabi 2007] cria agrupamentos sob demanda. Seu funcionamento considera consultas realizadas a partir de uma estação-base, que inunda a consulta, tendo como parâmetro um limiar percentual tolerável de diferença entre as leituras dos nós a serem agrupados. Ao final, os nós comuns informam suas leituras aos líderes, que as agregam e enviam à estação-base. Da mesma forma que o CAG, o DACH [Wu et al. 2008] define limiares de diferença quanto à similaridade, visando criar, de maneira centralizada na estação-base, uma hierarquia virtual com vários níveis crescentes de similaridade. Na literatura, o protocolo SCCS [Pham et al. 2010] se destaca ao considerar agrupamentos dinâmicos e reconfiguráveis formados pela similaridade de dados, sem necessitar de inundações tão constantes como o CAG e o DACH.

Contudo, nenhuma destas abordagens oferece suporte apropriado à similaridade de dados. Dos protocolos que consideram dados similares, o protocolo CAG depende de inundações constantes na rede para estabelecer novas hierarquias. O protocolo DACH por sua vez, depende muito da estação-base, que coleta informações da rede inteira para estabelecer a hierarquia. Por fim, embora o SCCS dependa de menos centralização, a sua manutenção não é adequada. Para o SCCS se adequar à variação dos dados, é permitida apenas a operação de quebra de agrupamento e, sem que existam fusões, a estação-base deve disparar novos processos de reagrupamento global.

Assim, um protocolo que atue de maneira mais distribuída e seja capaz de estabelecer agrupamentos lógicos consistentes com a semelhança das leituras de dados dos nós sensores se faz necessário. Os protocolos devem ser capazes de realizar uma espécie de sincronização dinâmica para que os nós possam também ser agrupados ou fragmentados continuamente, sem uma reestruturação completa a partir da estação-base.

### 3. DDFC

Esta seção apresenta o protocolo DDFC (*Dynamic Data-aware Firefly-based Clustering*), para criar e manter agrupamentos lógicos dos nós que possuam leituras de dados similares na RSSF. Através da manutenção de estruturas locais para o **armazenamento da vizinhança**, este protocolo sincroniza localmente a **agregação média de leituras similares** nos nós, possibilitando a determinação precisa de quando um agrupamento de nós deve ser fragmentado ou diferentes agrupamentos devem ser unidos. Uma vez estabelecidos os agrupamentos lógicos, o DDFC define **índices para o roteamento interno** aos agrupamentos, possibilitando que as mensagens dos nós comuns atinjam um líder do agrupamento. No DDFC, os agrupamentos formados podem ser compostos por mais de um líder, devido à extensão espacial das leituras similares. Do mesmo jeito, o roteamento interno nos agrupamentos pode utilizar-se de mais de um salto até atingir um líder.

#### 3.1. Armazenamento de vizinhança

Duas estruturas locais simples de grande importância em cada nó sensor suportam o funcionamento do DDFC, contendo (i) informações sobre as leituras dos vizinhos espaciais e (ii) o conjunto de vizinhos espaciais que satisfazem os *thresholds* de similaridade de dados. A Figura 1 ilustra um pedaço de uma topologia, à esquerda, e as estruturas de dados armazenadas de cada nó. O Nó selecionado possui no total sete vizinhos, dos quais quatro possuem dados similares, que satisfazem os *thresholds* de similaridade de dados.

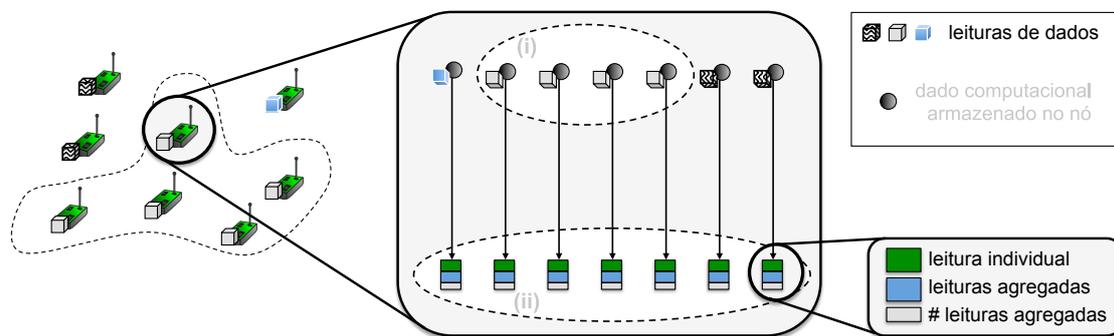


Figura 1. Estruturas de dados que representam a vizinhança dos nós.

Este conjunto de nós de leituras similares é mantido em uma estrutura (i)  $SN_{igh}$ , conjunto indicado por linhas pontilhadas na parte superior central da figura. Além disso, o nó em questão mantém uma estrutura (ii)  $NeighR$ , conjunto indicado por linhas pontilhadas na parte inferior central da figura, que possui informações sobre as leituras de todos os vizinhos espaciais. Tais informações envolvem a leitura individual de cada vizinho e a leitura agregada da vizinhança daquele vizinho, assim como a quantidade de nós cujas leituras que foram agregadas, para se determinar a importância desta agregação.

#### 3.2. Sincronização da agregação de leituras

O DDFC define um componente de sincronização inspirado nos princípios biológicos de vagalumes [Tyrrell et al. 2006], denominado Agente Vagalume, que atende as questões de sincronização de agregação de leituras e estabelecimento de vizinhanças de leituras similares. O Agente Vagalume sincroniza localmente um valor que indica a agregação das leituras do agrupamento do nó em questão. Com este valor, os nós sabem quando

eles devem sair de seu agrupamento, em caso de leituras muito diferentes, e quando agrupamentos vizinhos devem ser unidos devido à leituras muito próximas, satisfazendo o *threshold* de similaridade de dados.

Inicialmente, cada nó da rede faz parte de um agrupamento diferente, e os diversos agrupamentos gradualmente são unidos, de acordo com o *threshold* de similaridade de dados. Após a convergência e formação estável inicial de agrupamentos, estes são mantidos dinamicamente através da sua união e fragmentação. Estas operações resultam da adição ou remoção de arestas, que representam a semelhança de dados entre pares de sensores, referentes à vizinhança de cada nó do agrupamento, como será visto adiante.

O Algoritmo 1 apresenta o funcionamento do Agente Vagalume do DDFC. Periodicamente, cada nó envia em *broadcast* uma mensagem *beacon*, análoga ao piscar de um vagalume, informando seu identificador *ADDR*, sua leitura atual, obtida através da função *getReading()*, a leitura média agregada dos nós com leituras similares em sua vizinhança, obtida através da função *getAverageReading()* e a quantidade de vizinhos com leituras similares na vizinhança (l.1-5). Deve-se notar que o envio periódico destas mensagens insere sempre um tempo aleatório ínfimo para evitar transmissões simultâneas.

---

### Algoritmo 1 Agente Vagalume

---

```

1: procedimento BEACONTIMEREXPIRE
2:   Send(ADDR, getReading(), getAverageReading(), |SNeigh|)
3:   Wait(interval + rnd())
4:   BeaconTimerExpire()
5: fim procedimento
6:
7: procedimento RECEIVEBEACON(src, iR, aR, nR)
8:   NeighR[src] ← {iR, aR, nR}
9:   localAvg ← getAverageReading()
10:  se ( $|iR - localAvg| < CThresh$ ) e ( $|getReading() - aR| < CThresh$ ) então
11:    SNeigh ← SNeigh ∪ {src}
12:  senão
13:    SNeigh ← SNeigh - {src}
14:  fim se
15: fim procedimento
16:
17: procedimento GETAVERAGEREADING
18:   accumulatedReading ← getReading()
19:   numberOfReadings ← 1
20:  para cada  $v \in SNeigh$  faça
21:    temp ← NeighR[v].aR * NeighR[v].nR
22:    accumulatedReading ← accumulatedReading + temp
23:    numberOfReadings ← numberOfReadings + NeighR[v].nR
24:  fim para cada
25:  retorna ( $accumulatedReading / numberOfReadings$ )
26: fim procedimento

```

---

A função *getAverageReading* (l.17) calcula a média ponderada sincronizada da agregação de leituras na vizinhança local que satisfaz as relações de similaridade de acordo com o *threshold* desejado, ou seja, aqueles vizinhos que pertencem ao mesmo agrupamento do nó em questão. Considerando a leitura do nó atual (l.18-19), a média das leituras informadas pelos nós que fazem parte do mesmo agrupamento é computada

(l.20-24), considerando a leitura  $aR$  agregada de cada um destes nós e o número de leituras  $nR$  agregadas nele como um peso (l.21). Ao fim, obtém-se a média agregada de leituras similares na região do nó (l.25).

Ao receber um *beacon* (l.7), o nó saberá a sua origem  $src$  de envio, a leitura individual  $iR$  do nó origem, a leitura agregada  $aR$  média de sua vizinhança, e a quantidade de nós  $nR$  que foram considerados nesta agregação. Inicialmente, a estrutura  $Neighbor$  é atualizada (l.8) com esta informação, independente de relações de similaridade. A agregação média de leituras na região do nó atual (l.9) é considerada para verificar se as leituras do nó atual e do nó origem  $src$  satisfazem a similaridade de dados de acordo com o *threshold* de diferença de leituras  $CThresh$  (l.10). A estrutura  $SNeighbor$  é atualizada, incluindo a origem  $src$  caso este *threshold* seja satisfeito, ou removendo o nó  $src$ , caso contrário.

A função de similaridade utilizada no Algoritmo 1 consiste de duas partes: **(i)**  $|iR - localAvg| < CThresh$  e **(ii)**  $|getReading() - aR| < CThresh$ , que correspondem basicamente à mesma verificação de similaridade, porém com referências diferentes. A parte **(i)** verifica se a leitura  $iR$  recebida do nó vizinhos  $src$  satisfaz o *threshold* de diferença  $CThresh$  com relação ao agrupamento do nó atual. Já a parte **(ii)** verifica se a leitura atual  $getReading()$  do nó local satisfaz o *threshold* de diferença  $CThresh$  com relação ao agrupamento do nó  $src$ . Estas relações são expressas pela Equação 1, onde  $X$  e  $Y$  representam respectivamente a leitura do nó atual e a leitura com a qual ela está sendo comparada, a fim de satisfazer a diferença  $CThresh$ .

$$\left| Y - \frac{X + \sum_{v \in SNeighbor} (Neighbor[v].aR * Neighbor[v].nR)}{1 + \sum_{v \in SNeighbor} (Neighbor[v].nR)} \right| < CThresh \quad (1)$$

A Figura 2 ilustra um exemplo de funcionamento do Agente Vagalume, mostrando a sincronização da agregação de leituras de cada agrupamento e consequentes relações de similaridade de leituras. As arestas pontilhadas indicam vizinhos puramente espaciais, enquanto as sólidas indicam vizinhos que satisfazem a relação de similaridade de dados. As caixas ao lado de cada nó correspondem à estrutura vista na Figura 1, indicando, de cima para baixo, a leitura individual daquele nó, a leitura agregada sincronizada sua e de seus vizinhos, e a quantidade de leituras agregadas. Cada instante  $T$  é separado por um envio de *beacon* de cada nó. No instante inicial  $T1$ , as leituras agregadas de cada nó correspondem à sua leitura inicial, pois ainda não houve nenhuma troca de mensagens.

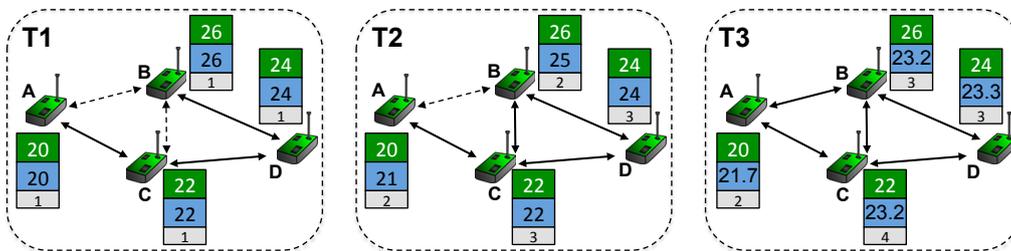


Figura 2. Funcionamento do agrupamento baseado em vagalumes.

Assim, considerando  $CThresh = 3.0$ , as arestas  $((B, D), (D, C), (C, A))$  satisfazem a Equação 1 e estabelecem relações de similaridade no estado  $T1$ . Então, os nós atualizam suas leituras agregadas  $aR_{Tn}$  de acordo com as leituras do instante  $aR_{Tn-1}$  anterior, como elaborado no Algoritmo 1. No instante  $T2$ ,  $aR_{T2}(A) = \frac{20+1*22}{1+1}$ ,  $aR_{T2}(B) =$

$\frac{26+1*24}{1+1}$ ,  $aR_{T_2}(C) = \frac{22+1*20+1*24}{1+1+1}$ ,  $aR_{T_2}(D) = \frac{24+1*22+1*26}{1+1+1}$ . Neste instante, a aresta de similaridade  $(B, C)$  passa a existir. No instante  $T_3$ , as leituras agregadas são atualizadas novamente,  $aR_{T_3}(A) = \frac{21+3*22}{1+3}$ ,  $aR_{T_3}(B) = \frac{25+3*24+3*22}{1+3+3}$ ,  $aR_{T_3}(C) = \frac{22+2*21+2*25+3*24}{1+2+2+3}$ ,  $aR_{T_3}(D) = \frac{24+2*25+3*22}{1+2+3}$ . Neste instante, a aresta de similaridade  $(A, B)$  passa a existir.

Com o Agente Vagalume operando desta maneira, cada nó possuirá sua estrutura *SNeigh* atualizada com a troca de *beacons*. Tal estrutura indica quais nós na vizinhança do nó atual são vistos como membros do mesmo agrupamento. Assim, como cada nó sabe quais vizinhos fazem parte do mesmo agrupamento, o agrupamento global de um nó corresponde ao conjunto formado pela união daquele nó com a união de cada um dos nós nas estruturas *SNeigh*, tal que essa operação é realizada recursivamente para cada nó da *SNeigh*. Indutivamente, se um nó  $A$  pertence ao agrupamento de um nó  $B$  e  $B$  pertence ao agrupamento de um nó  $C$ , então  $A$  também pertence ao agrupamento de  $C$ .

Contudo, esta visão global dos agrupamentos completos não é mantida localmente em cada um dos nós pois a coerência do estado reportado com a formação lógica real não poderia ser garantida. Tal operação causaria grande sobrecarga devido à quantidade de mensagens de atualização e manutenção que a dinamicidade do meio e das formações lógicas demandaria. Desta forma, embora os agrupamentos lógicos existam de maneira global, eles não são visíveis por completo em nível de nó. Entretanto, conhecendo as vizinhanças locais que pertencem aos seus agrupamentos, os nós podem organizar uma hierarquia distribuída baseada em líderes. Com estas estruturas definidas pelo Agente Vagalume, o Agente de Indexação, definido a seguir, opera respeitando a similaridade de dados, e garante que os nós consigam enviar mensagens aos líderes de seus agrupamentos.

### 3.3. Indexação local das rotas até líderes

O DDFC elege os líderes e estabelece rotas dos nós comuns até os líderes mais próximos através da componente denominada Agente de Indexação, considerando as relações de similaridade de leituras que o Agente Vagalume estabeleceu. O Agente de Indexação usa um esquema de pontuação baseado no sistema de regras proposto no KHOPCA [Brust et al. 2008], devido à sua flexibilidade e abordagem adaptativa. Através destas regras, cada nó atualiza sua pontuação auto atribuída de acordo com a pontuação de seus vizinhos que pertençam ao mesmo agrupamento - informação enviada em *piggyback* na mesma mensagem *beacon* utilizada pelo Agente Vagalume. É definido como parâmetro uma pontuação máxima  $MaxK$  que determina também a distância máxima até um líder. Os nós que possuem pontuação equivalente à esta pontuação máxima  $MaxK$  são determinados líderes, enquanto os demais nós utilizam sua pontuação como meio de determinar o próximo salto no roteamento até o líder mais próximo.

No início, todos os nós possuem uma pontuação individual  $pts = 0$ . Considerando  $MaxK$  como sendo o parâmetro da pontuação máxima e a lista  $SN^1$  de vizinhos pertencentes ao mesmo agrupamento, as regras utilizadas para a atualização dinâmica dos pontos são definidas de acordo com a Equação 2, baseada nas regras propostas pelo KHOPCA. A primeira condição das equações visa manter uma diferença máxima de 1 entre a pontuação dos nós adjacentes. A segunda regra define um nó como líder, maximizando seu  $pts$  para  $MaxK$ , caso seus vizinhos tenham pontuação mínima. A terceira regra visa diminuir a pontuação de um nó caso ele tenha pontuação maior que seus vizinhos

<sup>1</sup>Mesma lista *SNeigh*, porém abreviada por questões de espaço.

e não seja líder, a fim de manter a diferença máxima de 1 entre as pontuações adjacentes. Por fim, a quarta regra visa eliminar a existência de líderes, nós com  $pts = MaxK$ , adjacentes. Tais regras compõem o Sistema de Regras do Agente de Indexação.

$$pts(n) \leftarrow \begin{cases} \max(pts(SN(n))) - 1, & \text{se } pts(m) > pts(n), \forall m \in SN(n), \\ MaxK, & \text{se } pts(m) = 0, \forall m \in SN(n), \\ pts(n) - 1, & \text{se } pts(n) \neq MaxK \text{ e } pts(n) > pts(m), \\ & \forall m \in SN(n), \\ pts(n) - 1, & \text{se } pts(n) = MaxK \text{ e } \exists m \in SN(n) \text{ tal que} \\ & pts(m) = MaxK \text{ e } ( (|SN(m)| > |SN(n)|) \\ & \text{ou } (|SN(m)| = |SN(n)| \text{ e } m > n)). \end{cases} \quad (2)$$

Este sistema de regras, embora baseado no Jogo da Vida e nas regras propostas pelo protocolo KHOPCA [Brust et al. 2008], foi estendido para se adaptar melhor aos quesitos de dinamicidade do meio, possuindo melhor estabilidade. O Agente de Indexação dá mais prioridade no quesito pontuação àqueles nós que possuem mais vizinhos com leituras similares - prioridade expressa na quarta regra do Sistema de Regras. Desta forma, a estabilidade dos líderes é maior.

Estas regras são aplicadas periodicamente em cada nó, na mesma ordem que elas foram apresentadas, em uma sequência ordenada da primeira até a quarta. Entretanto, a cada período no qual elas são verificadas, apenas uma delas pode ser aplicada. Ou seja, se a primeira regra for verificada e aplicada, as demais não são sequer verificadas. Da mesma maneira, se uma regra não satisfizer as condições para que ela seja aplicada, são verificadas em sequência cada uma das sucessoras, parando a verificação das demais assim que a primeira delas for aplicada com sucesso. Tal operação é diferente do KHOPCA, que aplica mais de uma regra, em ordem e de maneira indeterminada em cada intervalo de tempo, produzindo resultados mais instáveis.

A Figura 3 ilustra como tais regras podem ser aplicadas a partir de uma topologia inicial, supondo um valor de  $MaxK = 3$ . As arestas sólidas entre cada par de nós indicam uma relação de similaridade de leituras entre eles e, desta maneira, eles se consideram vizinhos, de acordo com a estrutura  $SNeigh$ . No instante  $T1$ , todos os nós possuem a pontuação mínima  $pts = 0$ . No estado  $T2$ , dado que a verificação das regras não possui requisitos de sincronia, neste caso os nós  $B$ ,  $C$  e  $D$  aplicam as regras primeiro, maximizando seus pontos através da regra 2. Como os nós  $A$  e  $E$  realizam a verificação depois, seus vizinhos já possuem  $pts = MaxK$  e, desta forma, eles aplicam a regra 1. No instante  $T3$ , existem 3 nós adjacentes com  $pts = MaxK$ . Logo, os nós  $B$  e  $D$  aplicam a

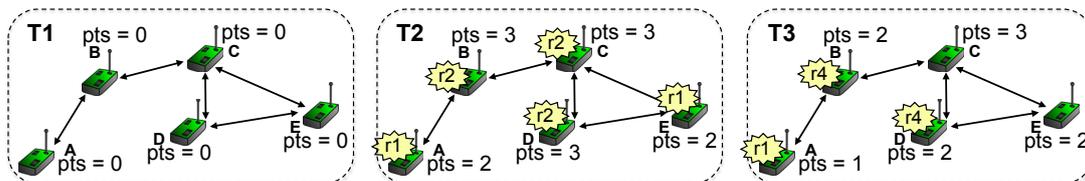


Figura 3. Aplicação das regras do Agente de Indexação.

regra 4, pois o nó  $C$  possui mais vizinhos similares. Novamente, o nó  $A$  aplica a regra 1, mantendo a diferença máxima de pontos entre nós adjacentes como 1.

A estrutura apresentada na figura é mantida dinamicamente com a variação das leituras e da topologia. Com tal estrutura, os nós que satisfazem  $pts = MaxK$  são considerados os líderes. Os nós comuns podem rotear os seus dados para o seu líder mais próximo ao selecionar como próximo salto um nó pertencente à estrutura  $SNeigh$ , cuja pontuação seja maior do que a sua atual. Assim, como os líderes são aqueles com mais pontos e as regras estabelecem uma progressão de pontos na direção aos líderes, garante-se que ele é atingido no final do roteamento da mensagem de um nó comum.

#### 4. Avaliação de Desempenho

Para a avaliação de desempenho do protocolo de agrupamento DDFC, ele foi implementado no simulador NS-3, versão 3.14.1. O cenário de avaliação cria uma situação realística de monitoramento de ambiente, a fim de se determinar a eficiência dos agrupamentos lógicos estabelecidos e verificar as relações de similaridade de dados e a qualidade das escolhas de líderes. Este cenário foi baseado nas leituras de umidade coletadas pelo laboratório Intel, da universidade de Berkeley, disponibilizados em [Berkeley 2012]. Considerando um cenário urbano, assume-se que os nós não tem problemas com escassez de energia, sendo obtida de redes elétricas existentes, como de postes [Furlaneto et al. 2012].

O cenário é composto de 54 nós, que operaram por 1200s. Como os dados de leituras obtidas eram de um ambiente pequeno, este foi aumentado em uma escala de  $15x$ , obtendo-se uma área retangular de  $630m$  por  $480m$ , visto que no cenário original um raio de transmissão padrão seria capaz de cobrir a área toda, comprometendo os resultados obtidos. Nesta escala, foi definido um raio de transmissão de  $100m$ , possibilitando uma avaliação que ainda possui dados com relações de similaridade.

Três parâmetros de funcionamento do DDFC são variados nas simulações: (i)  $CThresh$ , que indica a relação de similaridade entre os dados, (ii)  $int$ , que indica a duração do intervalo fixo de atuação do Controlador de *Beacons*, e (iii)  $MaxK$ , que indica a distância máxima até um nó líder, tal que a quantidade máxima de saltos é  $MaxK + 1$ . Além disso, é considerado tanto a operação do Agente de Indexação do protocolo DDFC com as suas regras propostas, baseadas no KHOPCA, quanto com as regras originais do KHOPCA, sendo este segundo caso de funcionamento denominado DDFC-K<sup>2</sup>. Note que o DDFC não é comparado diretamente com o KHOPCA, pois este não considera questões de similaridade de dados. Assim, o DDFC é completamente distinto do KHOPCA, fato evidente na componente principal Agente Vagalume, sendo apenas as regras do Agente de Indexação parcialmente baseadas nas do KHOPCA.

Considerando tais parâmetros, são definidas as métricas: **número de líderes**, **número de agrupamentos**, **número de nós solitários**, **duração dos líderes**, **amplitude média das leituras dos agrupamentos** e **inconsistência de rotas**. Estas métricas determinam o comportamento do protocolo com relação à variação das métricas, e o desempenho com relação à dinamicidade do protocolo para se adaptar às variações de leituras, ao *overhead* gerado e à qualidade das rotas até os líderes.

---

<sup>2</sup>Quando o KHOPCA for referenciado nos resultados, não se trata da sua implementação, mas sim do DDFC utilizando as regras originais do KHOPCA no seu Agente de Indexação.

O número de líderes, de agrupamentos e de nós solitários são avaliados não só por questões de desempenho, mas também a fim de se adequar ao propósito da aplicação. A duração dos líderes expressa quanto tempo um líder se manteve até ter sua pontuação reduzida, dado que durações médias maiores indicam que nós apropriados foram selecionados. Note que questões de energia são desconsideradas - a melhor aptidão destes nós como líder é independente de energia. A amplitude das leituras dos agrupamentos expressa a diferença média entre as maiores e menores leituras nos agrupamentos, e é importante para se determinar a corretude do comportamento da sincronização da agregação de leituras do Agente Vagalume. Por fim, a inconsistência de rotas corresponde à quantidade média de nós que não conseguem atingir seu líder dada a configuração da pontuação na rede em dado momento discreto, devendo ser minimizada.

Os resultados apresentados em seguida foram obtidos a partir de 35 simulações realizadas para cada combinação de parâmetros. Com isso, os gráficos apresentam os resultados com intervalos de confiança de 95%, indicados por barras verticais.

#### 4.1. Agrupamentos formados

A Figura 4 apresenta gráficos que avaliam a influência do parâmetro  $CThresh$  no número de líderes eleitos, de agrupamentos formados, e de nós solitários - isto é, cujo agrupamento é formado apenas por ele só. À esquerda, nota-se que quanto maior o  $CThresh$ , menor o número de líderes na rede. Ademais, o  $MaxK$  gera a mesma influência, sendo mais evidente entre os patamares de  $MaxK = 1$  e  $MaxK = 2$ . Isso acontece pois quanto maior for o parâmetro  $CThresh$ , menos agrupamentos existirão, como visto no gráfico central, pois nós com leituras mais distantes serão agrupados. O  $MaxK$  atua de acordo com as regras propostas no Agente de Indexação, sendo que quanto maior for ele, menor é o número de líderes. Entretanto, nota-se que a diferença é acentuada apenas para  $MaxK = 2$ , quando  $MaxK = 3$  não houve uma diferença significativa no número de líderes, indicando que mesmo havendo possibilidade de nós comuns utilizarem mais saltos para atingir um líder, são poucos os nós que de fato utilizam o número máximo de saltos. Por fim, o número de nós solitários tende a diminuir na medida que o parâmetro  $CThresh$  aumenta, pois com um valor maior para este parâmetro, nós com leituras mais divergentes podem ser agrupados com mais facilidade.

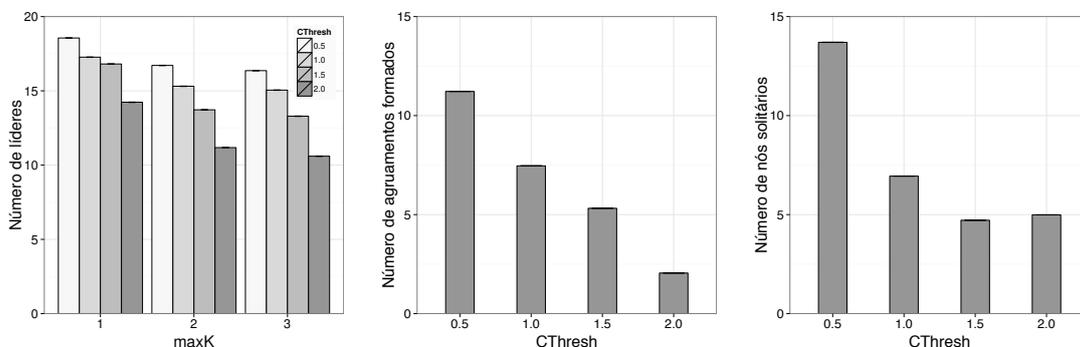


Figura 4. Relação do número de líderes, agrupamentos e nós solitários.

Nestes gráficos, o intervalo de confiança apresentado indica que em 95% das vezes, serão obtidos dados quase idênticos aos apresentados nos gráficos. Esta precisão se deve ao fato de o cenário ser estático, isto é, não possuir mudanças quanto à posição

dos nós e quanto às suas leituras. Entretanto, a operação do protocolo DDFC não é determinística, ou seja, depende de fatores de aleatoriedade. Por isso, tal intervalo de confiança mínimo indica que o protocolo opera de maneira controlada e estável.

#### 4.2. Similaridade de leituras dos nós agrupados

Para determinar se o Agente Vagalume foi capaz de agrupar nós com leituras similares, a métrica amplitude é utilizada, correspondendo à diferença entre a menor e maior leituras dentro de um agrupamento. Os gráficos na Figura 5 mostram a amplitude dos agrupamentos com relação aos parâmetros avaliados. Percebe-se que ela aumenta proporcionalmente ao  $CThresh$ , mas se mantém sempre inferior à  $2 * CThresh$ . Isso mostra que o Agente Vagalume foi capaz de agrupar nós com leituras similares, pois dado um valor médio  $m$ , um agrupamento aceitaria novos nós no intervalo  $[m - CThresh, m + CThresh]$ , cuja amplitude é exatamente  $2 * CThresh$ . Nota-se que o parâmetro  $MaxK$  não exerce grande influência na amplitude. Embora ele influencie no número de líderes, como visto na Figura 4, o número de agrupamentos se mantém o mesmo, visto que ele depende apenas da relação de similaridade de leituras entre os nós e do parâmetro  $CThresh$ .

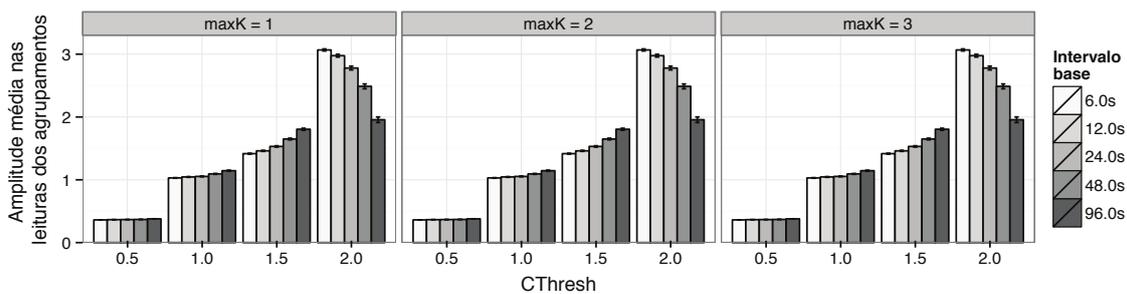


Figura 5. Similaridade de leituras dentro do mesmo agrupamento.

Por fim, quanto maior o parâmetro  $int$ , intervalo base da operação do Controlador de *Beacons*, maior é a amplitude das leituras nos agrupamentos. Isso ocorre pois com intervalos de operação muito grandes, os nós demoram mais para trocar *beacons*. Ademais, percebe-se que para  $CThresh = 2.0$ , o comportamento da variação de amplitude de acordo com o  $int$  é anômalo. Embora este comportamento não possa ser inteiramente justificável, isto acontece pois com  $CThresh = 2.0$  a rede opera com apenas 2 agrupamentos, como visto na Figura 4. A fragmentação da rede em apenas dois agrupamentos lógicos é anômala devido ao tamanho imenso que eles atingem. Isso mostra que o parâmetro  $CThresh$  deve ser ajustado de maneira sensível às leituras coletadas, sendo que  $CThresh = 2.0$  é uma valoração inadequada para a operação no cenário descrito.

#### 4.3. Duração dos líderes e inconsistência das rotas

A Figura 6 apresenta um conjunto de histogramas que avalia a relação entre o número de líderes e a sua duração, determinada em quantidades de turnos, dado que um turno representa um intervalo de tempo de 10s. O conjunto de histogramas é apresentado em um quadro que varia horizontalmente o parâmetro  $int$  e verticalmente o parâmetro  $CThresh$ .

De imediato, nota-se que em todos os casos a maior concentração de nós se encontra na duração de 120 turnos, que neste caso corresponde ao tempo de vida inteiro da rede. O uso das regras modificadas no DDFC acarretou em uma duração de nós líderes

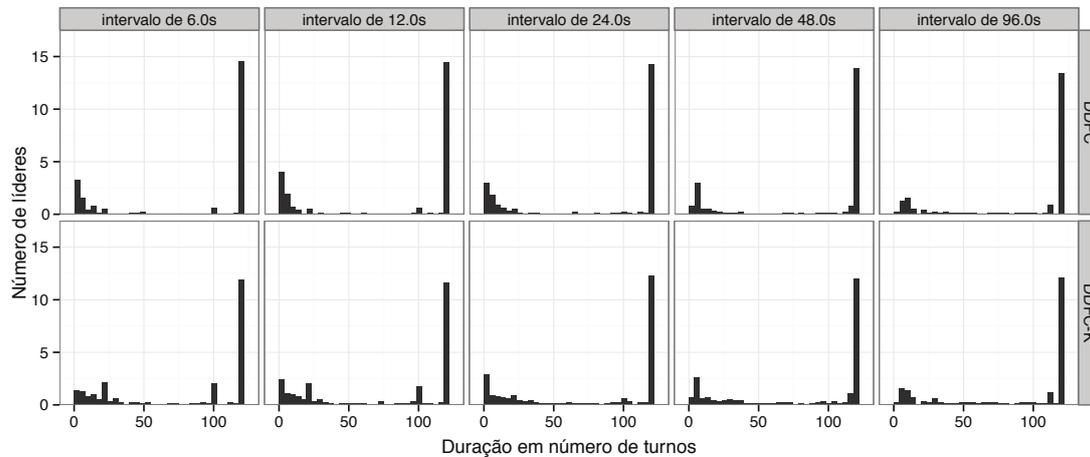


Figura 6. Duração dos líderes em número de turnos de 10s.

ainda maior quando comparado ao uso das regras originais do KHOPCA no DDFC-K, que apresentou os nós um pouco mais distribuídos nas baixas durações. Esta maior duração no DDFC resulta do sistema de pontuação do Agente de Indexação, indicando que as regras utilizadas determinam nós capazes de se manter estáveis como líderes, mesmo com a dinamicidade das leituras. Isso se deve em especial à quarta regra utilizada pelo Agente de Indexação, que dá prioridade como líder àqueles nós que possuem maior vizinhança com leituras comuns.

Os gráficos na Figura 7 mostram o valor da inconsistência média acumulada das rotas nos cenários que obedecem os parâmetros indicados - isto é, a média de rotas inválidas no tempo total de simulação. Confirma-se que quanto maior o *int*, maior a quantidade de rotas inválidas e que o parâmetro  $CThresh = 0.5$  é muito pequeno e, assim, inapropriado para este cenário. Este gráfico considera também o parâmetro  $MaxK$ . Percebe-se que, quanto maior o  $MaxK$ , maior a quantidade de rotas inválidas, devido ao fato das maiores distâncias possíveis entre nó comum e líder aumentarem. Quando comparado ao DDFC-K, as modificações utilizadas no DDFC diminuem o número de rotas inválidas na rede, devido à maior estabilidade dos nós líderes.

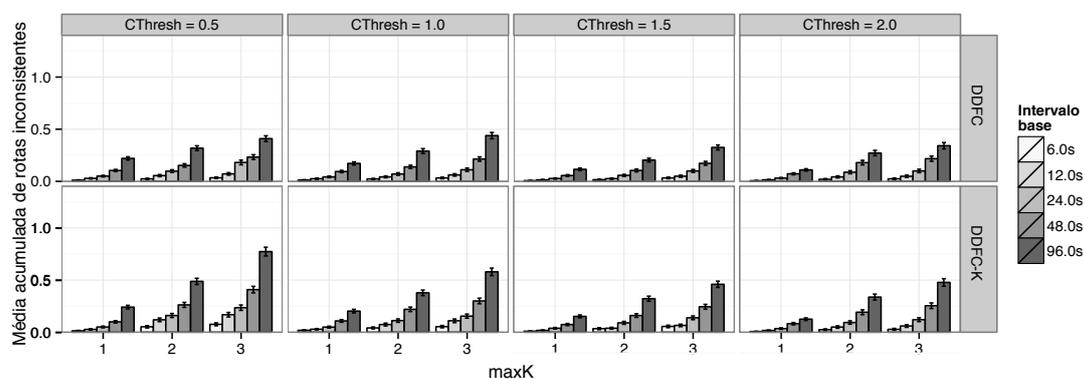


Figura 7. Inconsistência acumulada das rotas.

#### 4.4. Análise crítica dos parâmetros

Com os resultados apresentados, pôde-se estabelecer diversas relações entre os parâmetros e as métricas avaliadas. O *CThresh* influencia diretamente no número de agrupamentos formados, pois é ele que estabelece o critério de similaridade de leituras que deve ser satisfeito para que os nós se agrupem. Da mesma forma que um valor menor para o *CThresh* aumenta o número de agrupamentos formados, o número de líderes eleitos também é afetado, visto que ele tem relação com o número de agrupamentos. Assim, com menos agrupamentos formados, os nós da rede normalmente ficam mais próximos dos líderes, utilizando menos saltos para alcançá-los.

O Parâmetro *MaxK*, assim como o *CThresh*, influencia no número de líderes estabelecidos na rede, mas de maneira mais branda. Ademais, embora o aumento neste parâmetro possibilite que existam nós mais distantes de líderes na rede, a maior parte dos nós ainda assim se encontra próximo de líderes, sendo que de  $MaxK = 2$  para  $MaxK = 3$  já não houve grande mudança. Por outro lado, notou-se um aumento considerável no número de rotas inconsistentes à medida que o *MaxK* aumentou.

O intervalo de atuação *int* do Controlador de *Beacons* afeta diretamente a amplitude das leituras dos agrupamentos lógicos, sendo que intervalos menores diminuem a amplitude dos agrupamentos. Do mesmo modo, a inconsistência das rotas é inversamente proporcional ao parâmetro *int*. Esses comportamentos se devem ao número de *beacons* enviados, que aumenta para intervalos de atuação menores e, assim, possibilita uma maior frequência de atualização dos agrupamentos.

Mostrou-se que, independente da combinação de parâmetros, a amplitude dos agrupamentos sempre é inferior a  $2 * CThresh$ . Isso indica que a abordagem de sincronização *coletiva* de agregações de leituras nos nós foi bem sucedida. Através desta, os nós foram agrupados logicamente e dinamicamente, de acordo com a similaridade entre as suas leituras. Por fim, verificou-se que a mudança proposta nas regras originais do KHOPCA, utilizada pelo Agente de Indexação, foi capaz de estabelecer líderes mais estáveis, melhorando assim a consistência das rotas internas dos agrupamentos.

### 5. Conclusão

Agrupamentos de nós com leituras similares possibilitam melhor uso de técnicas de agregação, bem como detecção mais robusta de eventos anômalos. Baseando-se nisso e bioinspirado em vagalumes, o protocolo DDFC utiliza mensagens *beacon* periódicas para manter sincronizada a agregação das leituras dos nós de cada agrupamento. Com isso, definem-se de modo dinâmico nós vizinhos que satisfazem a similaridade de leituras desejável para o agrupamento, possibilitando a fragmentação e a união de agrupamentos lógicos. Considerando os vizinhos de leituras similares definidos pelo agente vagalume, um esquema de indexação mantém a indexação interna aos agrupamentos, estabelecendo rotas até os líderes mais próximos.

O DDFC foi avaliado com leituras reais, obtidas do laboratório Intel, de Berkeley. Simulações demonstram que o DDFC mantém os nós agrupados de maneira dinâmica através da agregação sincronizada das leituras dos agrupamentos, satisfazendo sempre o limiar de similaridade definido. Trabalhos futuros envolvem o controle adaptativo do intervalo entre as mensagens *beacon* enviadas e a comparação do protocolo DDFC com o SCCS, que também cria agrupamentos baseados na similaridade de dados.

## Referências

- Banerjee, J., Mitra, S. K., Ghosh, P., and Naskar, M. K. (2011). Memory based message efficient clustering (MMEC) for enhancement of lifetime in wireless sensor networks using a node deployment protocol. In *ICCCS '11: Proceedings of the 2011 International Conference on Communication, Computing & Security*. ACM.
- Berkeley (2012). <http://db.csail.mit.edu/labdata/labdata.html>.
- Brust, M. R., Frey, H., and Rothkugel, S. (2008). Dynamic multi-hop clustering for mobile hybrid wireless networks. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication - ICUIMC '08*, page 130, New York, New York, USA. ACM Press.
- Dechene, D. J., Jardali, A. E., Luccini, M., and Sauer, A. (2007). A survey of clustering algorithms for wireless sensor networks. *Computer Communications*, pages 2826–2841.
- Furlaneto, S., Santos, A., and Hara, C. (2012). An Efficient Data Acquisition Model for Urban Sensor Networks. *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 113–120.
- Guo, B. and Li, Z. (2007). A dynamic-clustering reactive routing algorithm for wireless sensor networks. *Wireless Networks*, 15(4):423–430.
- Islam, M., Abdullah, S., Wada, K., Uchida, J., and Chen, W. (2011). An efficient routing protocol on a Dynamic Cluster-based Sensor Network. In *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 161–165.
- López, T. S., Ranasinghe, D. C., Harrison, M., and Mcfarlane, D. (2012). Adding sense to the Internet of Things. *Personal and Ubiquitous Computing*, 16(3).
- Ma, H.-D. (2011). Internet of Things: Objectives and Scientific Challenges. *Journal of Computer Science and Technology*.
- Murty, R. N., Mainland, G., Rose, I., Chowdhury, A. R., Gosain, A., Bers, J., and Welsh, M. (2008). CitySense: An Urban-Scale Wireless Sensor Network and Testbed. In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 583–588.
- Partridge, C. (2011). Realizing the future of wireless data communications. *Communications of the ACM*, 54(9).
- 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(11):1311–1333.
- Rajkumar, R., Lee, I., Sha, L., and Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pages 731–736. ACM.
- Reis, I. A., Câmara, G., Assunção, R., and Monteiro, A. M. V. (2007). Data-aware clustering for geosensor networks data collection. *Anais XIII Simpósio Brasileiro de Sensoriamento Remoto*, pages 6059–6066.
- Tyrrell, A., Auer, G., and Bettstetter, C. (2006). Fireflies as role models for synchronization in ad hoc networks. In *BIONETICS '06: Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems*. ACM.
- Villas, L., Guidoni, D., and Araujo, R. (2011). Explorando a correlacao espacial na coleta de dados em redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2011)*, 29:411–424.
- Wu, X., Wang, P., Wang, W., and Shi, B. (2008). *Data-aware clustering hierarchy for wireless sensor networks*. Springer-Verlag.
- Yoon, S. and Shahabi, C. (2007). The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. *Transactions on Sensor Networks*, 3(1).