

Modelo Distribuído para Otimização do Armazenamento em Redes de Sensores Sem Fio

Yakov Nae e Lee Luan Ling

Resumo — *As redes de sensores sem fio têm uma grande capacidade de armazenamento agregada, que não pode ser utilizada inteiramente de forma eficiente, pois cada sensor tem baixa capacidade de armazenamento. Logo esta capacidade é subjetiva, devido as limitações de cada sensor individualmente. Neste artigo, desenvolvemos um modelo distribuído de armazenamento do sistema. O objetivo do modelo é contornar as limitações dos sensores individuais. Neste modelo, sensores constroem cadeias distribuídas de armazenamento sob demanda (“on-demand” Distributed Storage Chains ou DSC) que localizam armazenamento que não são utilizados dos sensores disponíveis. As cadeias são construídas via estratégia do passeio determinístico acima da área geográfica. Esses passeios lembram o comportamento de passeios aleatórios [8]. Com esse modelo, poderemos aprimorar eficientemente a utilização do armazenamento do sistema. Além disso, como este modelo é determinístico, não é necessário manter informações sobre a estrutura dessas cadeias.*

Palavras chaves – *Algoritmos Distribuídos, Expanders, Geo-Roteamento, Passeio Aleatório, Diagramas de Voronoi*

Abstract — *While wireless sensor network systems host a vast capacity of storage on the aggregate, that capacity cannot be used entirely. Since each sensor has a poor storage capacity, the system is subjected to constraints derived from every individual sensor. In this paper, we suggest a distributed model for storage aggregation whose goal is to limit those individual storage constraints. On our model, sensors construct “on-demand” distributed storage chains (DSC) that locate non-utilized storage based on available sensors resources. Those chains are constructed via a deterministic walk strategy over the geographic field that reminds the behavior of random walks [8]. Using this model, we managed to efficiently increase the system’s storage utilization. Moreover, since this model is deterministic, no information is maintained about the structure of those chains.*

Keywords – *Distributed algorithms, Expanders, Geo-Routing, Random walk, Voronoi diagrams*

I. INTRODUÇÃO

Redes de sensores (também chamadas SensorNets) são sistemas sensoriais distribuídos que são compostos de um grande número de dispositivos autônomos que gravam atividades em diferente regiões [7,10]. Estes dispositivos gravam informações detalhadas do ambiente em que estão inseridos para estudar um fenômeno físico em campo, através de medidas numéricas. Como a quantidade de transistores em um chip de custo mínimo dobra a cada um ou dois anos [6] (Lei do Moore), os pesquisadores aplicam este tipo de tecnologia de forma a permitir que a ciência de computação chegue a níveis cada vez mais altos. Assim, estas redes fornecem soluções “estado da arte” para monitorar observações

extensivas, por exemplo: atividade sísmica, clima, vigilância, radioatividade, monitoramento de habitat, etc.

Enquanto a quantidade de recursos de cada sensor individual é pobre, o sistema como todo tem uma vasta quantidade de recursos agregados [6]. Então, apesar de o sistema como um todo parecer robusto, ele está sujeito às limitações de armazenamento de cada sensor individual. Desde que é inviável implantar novos sensores (em diversos cenários) ou prever a quantidade de atividades que cada região pode produzir, muitos trabalhos foram realizados no estudo da exploração das capacidades totais do sistema. Apesar de já haver muitos trabalhos anteriores sobre agregações de recursos (como dados, energia, computação, etc.), a maioria dos estudos usam a característica do recurso a ser agregado para implementar uma solução. Em outras palavras, em muitas pesquisas a agregação de recursos depende do cenário ou características do modelo. Muito pouco se sabe sobre a própria agregação do recurso. Apresentamos um novo conceito que agrega armazenamento sem ter que pressupor previamente o fenômeno.

Muitos trabalhos tratam sobre a otimização da energia. Eles consideram o consumo da energia como um problema local (de cada sensor individual – por exemplo quando ele está ativo) ou como um problema global da rede como um todo. Um exemplo para o último caso é a pesquisa sobre buracos de energia (*energy holes*) [9]. Este trabalho tenta solucionar o problema do alto consumo de energia em sensores que estão colocados perto da estação “*sink*” (estação base). Estes sensores que sofrem tráfego intenso de dados até a sua total exaustão de energia são chamados buracos de energia. Isso porque eles formam buracos de cobertura que são inaceitáveis numa aplicação de redes de sensores. Entretanto, este trabalho é baseado nas características do *many-to-one* SensorNet onde todos os sensores são obrigados a se comunicar com uma estação base. De acordo com esta característica, os sensores que estão perto do “*sink*” absorvem uma grande carga e conseqüentemente correm risco de perderem sua energia inteiro quase simultaneamente. Entretanto, este estudo não oferece solução para buracos aleatórios de energia que possam estar longe do “*sink*”.

A idéia de construir redes P2P baseadas numa decomposição dinâmica de um espaço contínuo para células, foi apresentada em Naor et. al. [2]. Eles propuseram uma técnica para divisão de um intervalo de uma dimensão para células não sobrepostas. Neste estudo, cada célula representou um processo. Assim eles usaram uma abordagem *Discreta-Contínua* “onde a rede atual é uma discretização de um gráfico contínuo para células”. Eles demonstraram a sua abordagem para aplicação DHT (*Distributed Hash Table* – Tabela de Hash Distribuída). Uma abordagem similar foi apresentada

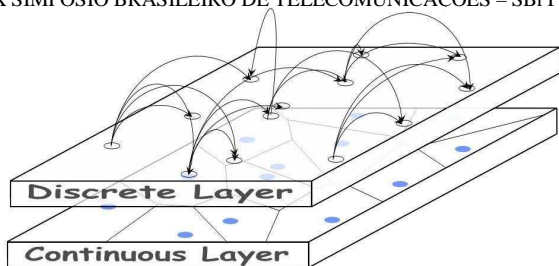


Figura 1. A abordagem discreta contínua para duas dimensões

para construir GHT (*Geographic Hash Table* – Tabela de Hash Geográfica) onde o espaço geográfico contínuo é associado a um conjunto discreto de *home-node* (as localizações dos sensores). Ratnasamy et, al. [3] sugeriram um modelo de agregação dos dados que trata as capacidades das comunicações e armazenamento juntos. Dados com o mesmo nome geral (por exemplo, observações dos elefantes) são gravados no mesmo sensor. Então, consultas sobre um tipo de dado seriam comunicadas para um único sensor responsável, ao invés de pesquisar a rede inteira. Neste modelo, os sensores comunicam-se entre si, para entregar eventos de mesmo nome geral para a localização *hash*. Diferentemente da agregação do armazenamento, a agregação de dados é baseada num tipo de atividades que poderia ser armazenada. Assim, GHT, DHT e buracos de energia são similares, mas não são comparáveis com o trabalho apresentado nesse artigo.

Focamos na questão de atividades sobrecarregadas que foram geradas por uma região, cujo sensor responsável já tinha *esgotado* sua capacidade de armazenamento. Quando a rede de sensores é implantada em um campo alvo perfeitamente balanceado, eventos ocorrem uniformemente pelas diferentes regiões. Assim, a utilização do armazenamento é maximizada uma vez que os sensores se esgotam simultaneamente suas capacidades de armazenamento. Entretanto, se o campo alvo não estiver balanceado, o sistema teria rapidamente se tornar ineficiente por causa dos buracos de cobertura. Neste caso, o armazenamento inteiro não terá sido utilizado. Mais ainda assim, o sistema ainda teria capacidade de armazenamento não utilizada. De modo geral, nesta pesquisa desenvolvemos um protocolo que constrói enlaces entre sensores com suas capacidades de armazenamento esgotadas e sensores disponíveis que contenham reservas de armazenamento que não foram utilizadas totalmente. Estes enlaces representam associações entre sensores que compartilham as suas capacidades de armazenamento.

O restante deste artigo está organizado da seguinte forma. Na Seção II descrevemos o modelo de rede a ser utilizado. Na seção III é apresentado o protocolo desenvolvido para o modelo de rede proposto. Na Seção IV, apresentamos as simulações e resultados obtidos e finalmente as conclusões são apresentadas na Seção V.

II. MODELO DE REDE E AS CAMADAS

A. Modelo de Rede

Modelamos a rede de sensores como um conjunto S de n sensores: s_1, s_2, \dots, s_n , localizados na quadrado unitário $[0, 1]^2$. Seja $\{x(s_i), y(s_i)\}$ as coordenadas do sensor s_i , e assumimos que os sensores sabem suas localizações (usando GPS ou alguma técnica do localização). Além disso, assumimos que não existem duas localizações geográficas iguais para nenhum par

dos sensores. Modelamos o modelo de comunicações como gráfico de disk unitário (*unit disk graph*) [11], onde cada par de sensores pode se comunicar, se e somente se a distância Euclidiana entre eles for menor que a *faixa de transmissão* r . Inclusive, assumimos que esse parâmetro r é suficientemente grande para garantir que não exista componentes isolados.

Quando qualquer sensor $s_i \in S$ esgota sua capacidade de armazenamento, ele armazena seus atividades adicionais num outro sensor disponível $l_1(s_i) \in S$. Se e quando $l_1(s_i)$ também esgota sua capacidade, s_i aloca (através $l_1(s_i)$) outro sensor disponível $l_2(s_i) \in S$ e assim por diante. Definimos esta cadeia de sensores, que é originada no sensor s_i : $\omega(s_i) = (s_i, l_1(s_i), l_2(s_i), \dots, l_n(s_i))$ e a chamamos DSC (*Distributed Storage Chain* – Cadeia Distribuída de Armazenamento).

Descrevemos o nosso protocolo, usando um conceito bastante conhecido da área de geometria computacional, o *Diagrama de Voronoi* [5]. O diagrama de *Voronoi* para um conjunto de nós S num espaço, compõe um mosaico deste espaço em células (regiões) *Voronoi* não sobrepostas. Anotamos $Vor(s)$, $s \in S$, para descrever uma região do nó s , onde todos os pontos dentro de $Vor(s)$ são mais próximos de s do que qualquer outro nó em S . Consideramos apenas os pontos dentro do quadrado unitário $[0, 1]^2$. Usamos a *distância Euclidiana* como medida de distância dentre cada par de nós.

B. Camada de Rede – Roteamento geográfico

Assim como em GHT [3] construímos o protocolo em cima de outro protocolo chamado GPSR (*Greedy Perimeter Stateless Routing* – Roteamento Guloso Perímetro Sem-Estado) [4]. O protocolo GPSR é um protocolo de georoteamento (*geo-routing*) que oferece instruções de roteamento baseadas apenas na localização geográficas dos sensores em campo. Este tipo de roteamento é local, sem memória e *unicast* que se encaixa as necessidades da DSC.

O protocolo GPSR tem dois modos de roteamento: O avanço guloso (*Greedy forwarding*) e o avanço de perímetro (*Perimeter forwarding*). No avanço guloso, os pacotes são progressivamente roteados cada vez mais perto de seus destinos. Assim, em cada salto (*hop*) os pacotes são roteados em direção ao vizinho que estiver mais próximo do seu destino. Em outras palavras, quando um sensor recebe um pacote de dados, ele lança este pacote para o vizinho que pode atingir o destino desejado. Já o modo avanço guloso (*Greedy forwarding*), falha quando o pacote de dados chega a um nó cujo vizinho não consegue passá-lo adiante em direção ao destino (*mínimo local*). Neste caso, nós evitamos a mínima local, usando o modo de avanço de perímetro. Este modo vai rotar o pacote pelas faces do mínimo local, usando a regra da mão direita. Quando um pacote alcança um sensor mais próximo ao destino (ainda mais próximo do que o sensor que iniciou o modo de roteamento perímetro), o GPSR volta ao modo de avanço guloso.

Como veremos abaixo, a característica continua da DSC exige que os pacotes sejam roteados não só entre dois sensores, mas também entre duas localizações geográficas, ou seja, entre duas localizações contidas em regiões de sensores. Fazemos isso, usando a mesma solução provada por Ratnasamy et. al. [3]. Quando a mensagem é endereçada a um destino $d = \{x, y\}$ que não é uma localidade específica de um sensor $d \notin S$, dizemos que a mensagem foi roteada na região de um determinado sensor $d \in V(s_i)$. Eles mostraram [3] que se

um pacote não pode alcançar o destino final, ele vai parar num sensor s_i que esteja no caminho que chamamos de *Home Node*. Este sensor é o sensor que é mais próximo do destino [3,4]. Em particular, o endereço de destino $\{x,y\}$ é localizado dentro da célula do *Voronoi* daquele *Home Node*.

C. Rede Overlay – Gabber-Galil Expander

Os sensores com suas capacidades de armazenamento esgotadas montam uma *rede overlay* via uma construção *sob demanda* das cadeias DSC. Esta *rede overlay* começa como um gráfico vazio que cresce constantemente. Os enlaces são adicionados dinamicamente quando algum sensor ou sua cadeia se esgotam suas capacidades de armazenamento. Esta *rede overlay* é baseada em “*explicit expander construction*” que foi apresentada originalmente por Gabber e Galil [1]. Mostramos que esta *rede* pode ser construída distribuídamente e manter as características do gráfico tipo *expander*: um gráfico altamente conectado com o número linear de enlaces. Além disso, este gráfico tem diâmetro curto, muitos caminhos entre os pares de nós, podem ser coberto rapidamente por um passeio aleatório (*Random Walk*) e obviamente gráficos tipo *expander* tem uma ótima expansão. Para este gráfico tipo *expander* em particular, como apresentado no Teorema 1, qualquer conjunto menor que $n/2$ nós, tem um conjunto de vizinhos de primeiro grau (primeiro salto), sem incluir a si mesmo cujo tamanho é pelo menos uma fração constante do conjunto original.

Introduziremos o gráfico usado: Seja $G(V, \epsilon)$ um gráfico contínuo dentro do quadrado unitário onde $\{x, y\} \in V = [0, 1]^2$ é um conjunto contínuo de nós. $\epsilon \subset V \times V$ denota o conjunto de enlaces em G . Usamos a transformada Gabber-Galil [1] (na verdade usamos apenas uma parte dela) para relacionar cada ponto em V com dois enlaces Gabber Galil. De acordo com suas direções chamamos estes enlaces de enlace Leste e Norte.

Definição 1. Transformada Gabber-Galil para um ponto $\{x,y\}$ em $[0, 1]^2$ gera os dois enlaces a seguir:

$$GG(x, y) = \begin{cases} \text{Leste} : \{x + y, y\} \pmod 1 \\ \text{Norte} : \{x, x + y\} \pmod 1 \end{cases} \quad (1)$$

Observe que os enlaces acima são cíclicos por definição do módulo 1. Um enlace $(\{x,y\}, \{x', y'\})$ está em ϵ *iff* $\{x',y'\}$ encontra o enlace Leste ou Norte do $\{x, y\}$. A motivação básica para aplicar esta tipo de gráfico na área alvo é explicada no Teorema 1.

Teorema 1. Para cada conjunto A dos nós em V que satisfazem $\mu(A) \leq 0.5$, quando $\mu(A)$ representa a área de A :

$$\mu((GG_{\text{Leste}}(A) \cup GG_{\text{Norte}}(A)) \setminus A) \geq \frac{2 - \sqrt{3}}{2} \mu(A) \quad (2)$$

De acordo com Teorema 1 demonstrado em [1], qualquer conjunto de sensores cujas áreas reunidas somam uma área $A \leq 0.5$, pode alcançar uma grande área em seu primeiro salto, usando a transformada Gabber-Galil. Em outras palavras, qualquer conjunto de tamanho A tem uma grande vizinhança formada por vários outros sensores da *rede overlay*. Consideramos que com o uso desta transformada (Definição 1), os sensores com suas capacidades de armazenamento esgotadas sejam facilmente capazes de localizar reservas

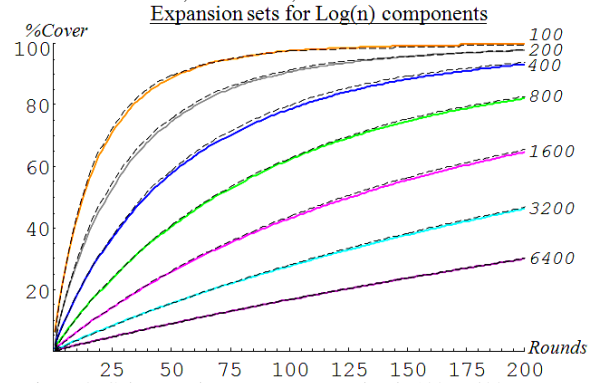


Figura 2. Cobertura da expansão para redes de 100 - 6400 sensores

potenciais de armazenamento.

De acordo com a nossa definição, a transformada define um conjunto infinito de enlaces. De fato, cada ponto no quadrado unitário contínuo é relacionado com dois enlaces. A fim de relacionar o conjunto contínuo de nós V e enlaces ϵ com o conjunto discreto de sensores S , definimos uma versão 2-D da abordagem contínua discreta [2]. Segundo esta abordagem, a *rede atual* é uma discretização do gráfico contínuo baseado numa decomposição da área alvo em células, onde cada sensor é responsável por uma célula. Duas células são conectadas se contiverem pontos adjacentes no gráfico contínuo. Em nosso caso, os sensores dividem a área alvo em células *Voronoi* (regiões) e dizemos que cada sensor é *home node* para todos os pontos em sua região *Voronoi*.

Definição 2. Para cada ponto $p \in [0, 1]^2$ e um conjunto dos sensores S localizados em $[0, 1]^2$, $D(p)$ denota o *Home node* de p . $D(p)$ é o sensor único $s \in S$ que satisfaz $p \in \text{Vor}(s)$.

Agora formalmente, definimos a *rede overlay* como um gráfico discreto $G(V, E)$. Seja $V = S$ denota o conjunto dos sensores que estão localizados no quadrado unitário. Seja E um conjunto discreto dos enlaces. G descreve a *rede overlay* que é um produto do nosso protocolo (seção III). A Definição 3 (e fig. 1.) demonstra a relação entre os conjuntos dos enlaces E do gráfico discreto e dos enlaces ϵ do gráfico contínuo:

Definição 3. De acordo com a abordagem contínua discreta: Um *enlace unidirecional* (s_i, s_j) está em E *iff* :

$$\exists p, p' \text{ s.t. } (p, p') \in \epsilon \wedge D(p) = s_i \wedge D(p') = s_j \quad (3)$$

ou em outras palavras, um enlace em E é direcionado de um ponto na região do s_i para outro ponto na região de s_j

III. DESCRIÇÃO DO PROTOCOLO

Em geral, podemos descrever este protocolo como uma simples abstração que ocorre em cada sensor: *Se possível, armazena localmente, senão armazena adiante*. Este protocolo implementa a instrução de armazenar adiante via construção da DSC. Determinamos os enlaces usando um *search function* que localiza os sensores disponíveis. Por esta função ser determinística, ela pode ser usado tanto para localizar sensores disponíveis, como para reconstruir a DSC e recuperar os dados da cadeia. Em geral, toda vez que um sensor com sua capacidade de armazenamento esgotada desejar armazenar uma nova atividade, ele se dirige ao último enlace de sua própria cadeia (EOC – *End Of Chain*). Observe que sensor em

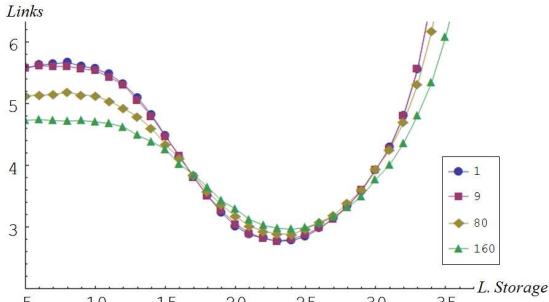


Figura 3. Número de enlaces STORE para diferentes conj. do Hs (SM=80%)

seu estado inicial mantém DSC com comprimento zero, ou seja, seu EOC é o próprio sensor. Quando ele esgota sua capacidade de armazenamento, ele adiciona novo enlace ao seu DSC usando o *search function*.

A. Cadeias de Armazenamento Distribuídas (DSC)

O *search function* é definido pela a transformada Gabber – Galil (Definição 1). Uma vez que a rede *overlay* é uma camada discreta acima do gráfico tipo *expander* (Figura 1), assumimos que a rede *overlay* é também um *expander*. Recentemente foi demonstrado [8] que muitos passeios aleatórios exploram (cobrem) os gráficos tipos *expander* muito eficientemente. Especificamente, k passeios aleatórios cobrem o gráfico k vezes mais rápido que um passeio aleatório único. Entretanto, em nosso caso usamos um passeio determinístico que pode ser facilmente revertido quando se quiser recuperar os dados. Começamos por definir os passeios contínuos e discretos:

Definição 4. Passeio Gabber–Galil Contínuo do comprimento $k+1$ que começa no sensor $s=g^0$ é um caminho $\omega(s) = (g^0, g^1, g^2, \dots, g^k)$ onde $g^j = \{x_j, y_j\}$ e para $1 \leq j \leq k$, $(g^{j-1}, g^j) \in \mathcal{E}$. Falamos que $\omega(s)$ é uma *cadeia* na camada contínua. De acordo com a abordagem Discreta–Contínua, a forma discreta do $\omega(s)$ é a cadeia: $W(s) = (s, D(g^1), D(g^2), \dots, D(g^k))$.

Definição 5. DSC Discreto que começa no sensor s , é a forma discreta do passeio contínuo, onde para cada passeio j , $1 \leq j \leq n$, $g^j = GG^{j-1}(g^{j-1})$:

$$GG^j(x, y) = \begin{cases} GG_{East}(x, y) & \text{if } j = 0 \text{ mod } 2 \\ GG_{North}(x, y) & \text{if } j = 1 \text{ mod } 2 \end{cases} \quad (4)$$

Em outras palavras, a cadeia contínua é construída pela seleção dos enlaces Leste e Norte alternadamente. Observe que a cadeia contínua do sensor s , $\omega(s) = (g_0(s), g_1(s), g_2(s), \dots)$ pode ser calculada recursivamente. Ou seja, $g^{j+1}(s) = GG^j(g^j(s))$. Então, cada sensor pode calcular localmente o locação j da sua cadeia contínua.

B. Especificações do Protocolo

O protocolo que constrói DSC consiste de três tipos de pacotes: PUT, ACK e GET: PUT é usado para armazenar um evento que originalmente foi gerado por um sensor gerador com sua capacidade de armazenamento esgotada. Pacotes tipo PUT são enviados de um gerador para seu EOC. Quando também o sensor EOC tem sua capacidade de armazenamento esgotada, ele consegue calcular localmente o próximo enlace da cadeia do sensor gerador. Note que para isso o sensor precisa somente da posição do gerador e do comprimento da cadeia. Pacotes ACK são enviados sempre que PUT for armazenado com sucesso. Eles são enviados para informar que

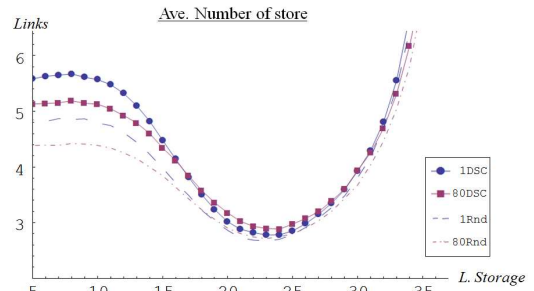


Figura 4. DSC vs. Seleção aleatória (SM=80%)

o evento foi armazenado e atualizar o comprimento da cadeia (localização do EOC). Pacotes GET são enviados através das cadeias DSC para recuperar a informação que foi armazenada num sensor com sua capacidade esgotada.

IV. RESULTADOS DAS SIMULAÇÕES

Queremos avaliar a qualidade da DSC, respeitando os procedimentos que armazenam e/ ou recuperam os dados. Comparamos DSC com o modelo de passeio aleatório, onde localizações, e conseqüentemente seus *home nodes*, são selecionados aleatoriamente. Conduzimos as simulações no *Wolfram Mathematica 7*. Usamos topologias das redes que são baseadas numa implantação uniforme aleatória de n sensores no quadrado unitário.

A. Experimento 1: Conjuntos expandidos via DSC

Apresenta-se a comparação entre a *search function* sugerida e o passeio aleatório (rXY). Para avaliar as propriedades de expansão da DSC fizemos a seguinte simulação. Selecionamos o componente primário $S_0 \subset S$ onde selecionamos, $s_i \in S$ $0 \leq i < k$, numa probabilidade uniforme de $S = \{s_1, s_2, s_3, \dots, s_n\}$ onde $|S_0| = k$. Assim, cada sensor $s_i \in S_0$ pode iniciar a sua própria DSC. A cada passo j , $0 \leq j < 200$ anotamos que S_j seja o conjunto de todos os sensores que foram visitados (cobertos) por pelo menos uma das k cadeias.

A Figura 2 apresenta a taxa de crescimento de $|S_j|$ (em % de número total de sensores) com passo j para topologias da rede de tamanho 100-6400 sensores com componentes primários de $(\log_2(n))$ sensores. Escolhemos demonstrar tamanho de componente relativamente pequeno que gere 80% das atividades do sistema. Para avaliar os resultados, comparamos o nosso passeio determinístico com um passeio aleatório, que é conhecido por ter uma cobertura rápida [8]. A cada passo, rXY escolhe-se um ponto aleatório do quadrado unitário. Note que no rXY a probabilidade de se selecionar um sensor s , não é exatamente uniforme. Ela é proporcional à área de $Vor(s)$. O problema das células de *Voronoi* com tamanho diferente já foi também discutido em [2] e parece inerente a esta abordagem. O melhor índice de cobertura que podemos desejar é do rXY . rXY está representado na Figura 2 pelas linhas pontilhadas. Figura 2 demonstra os resultados da *search function* determinística que é baseada em DSC. A figura mostra claramente que o DSC quase coincide com os resultados do rXY para todos os tamanhos de redes. Assim, afirmamos que a transformada pode ser considerada uma *search function* determinística ideal, na qual cada sensor conta apenas com a informação de sua localização. Apesar dos resultados que apresentamos aqui serem para $k = \log_2(n)$, nossos experimentos mostram que o comportamento da DSCs

é próximo daquele de rXY para todos os valores de k .

Nossa análise avalia o desempenho da DSC e não do GPSR. Entretanto, é preciso manter em mente que cada passo da rede *overlay* pode ser composta por mais de um salto (de acordo com GPSR). O GPSR gera um volume constante de pacotes *overhead* (*beacons*) por nó. Independente do tamanho do sistema, esse overhead é constante e tem magnitude muito menor que a DSC sugerida.

B. Experimento 2: Armazenando e recuperando dados

Neste conjunto de experimentos, avaliamos a eficiência da DSC. Propusemos avaliar duas métricas básicas que são STORE e RETRIEVE (ARMAZENAMENTO e RECUPERAÇÃO). STORE refere-se ao número de pacotes que precisamos mandar via a rede *overlay* (usamos pacotes PUT). RETRIEVE é o número de saltos que o pacote de dados deve enviar para fazer a recuperação. Focamos a nossa simulação numa rede de 1600 sensores. Cada sensor tem um armazenamento local de $M=100$ unidades. Simulamos a geração de atividades até 80% da capacidade total da rede (SM – *System Memory*). O modelo de geração das atividades é o seguinte: Seja $S_{HS}=\{s^1, s^2, s^3, \dots, s^k\}$ um conjunto Hs do tamanho k , cada atividade será gerada com probabilidade $p_{HS}=0.8$ num sensor $s_i \in S_{HS}$ (uniformemente). A mesma atividade, com probabilidade $1-p_{HS}=0.2$, será gerada por um sensor s_j que não está em Hs : $s_j \in S/S_{HS}$ (uniformemente). Assim, há uma grande probabilidade para os sensores $s_i \in S_{HS}$ terem que construir DSC. Fixamos um TTL (*Time To Live*) de 16 saltos (*hops*) para evitar que pacotes tipo PUT sejam roteados infinitamente sem achar um *host* disponível. Os resultados mostram que apenas 0.05% dos pacotes do total enviado foram descartados.

Defina-se uma quantidade de armazenamento local. Ou seja, uma quantidade das unidades de memória que podem ser usadas somente para atividades que forem geradas nos sensores. Variamos esta quantidade local (*local storage*) entre 0–40 unidades.

Figuras 3 e 4 mostram os resultados das simulações. Figura 3. mostra 4 tipos diferentes do conjunto de Hs (1, 9, 80, 160 sensores). Figura 4. Mostra uma comparação entre DSC e uma seleção dos *hosts* aleatoriamente. As curvas das duas figuras mostram o custo de gravação das atividades, ou seja, o número médio de pacotes PUT. Note que os custos de recuperação são menores porque ignoramos as entradas nas cadeias DSC que não contém dados. Em outras palavras, os números dos pacotes tipos GET são bem menores do que os dos tipos PUT. No entanto, o comportamento das curvas é igual, e por esta razão, mostramos somente as curvas do custo para pacotes tipo PUT. (que são mais caros).

Reportamos que para até 50% de capacidade total do sistema, os custos são menores porque os sensores com suas capacidades de armazenamento esgotadas conseguiram encontrar com sensores disponíveis rapidamente. Nas Figuras 3 e 4, mostramos os custos para utilização de 80% da capacidade total do sistema, quando o custo de armazenamento é maior (mais difícil de achar sensores disponíveis), observamos que as curvas têm comportamento exponencial para *local storage* maiores do que 25 unidades. Isso é esperado, pois menos armazenamento global será oferecido. Ou seja, o conjunto S/S_{HS} oferece menos espaço de armazenamento para os sensores $s_i \in S_{HS}$. Todas as curvas têm

um ponto de mínimo para capacidade local entre 20–25 unidades. Então, se podemos caracterizar a área alvo, podemos otimizar os custos de armazenamento global. Note que o custo nessa faixa é de cerca de três enlacs, que é relativamente pequeno para melhorar a utilização da capacidade de armazenamento. Na Figura 4. compara-se o comportamento do *Random walk* (rXY) com a do DSC. Usamos conjuntos de sensores Hs de tamanho 1 e 80. Note que a diferença entre as curvas é pequena, sendo ainda menor na faixa de interesse.

V. CONCLUSÕES

Nesse artigo mostramos um modelo que maximiza a utilização de armazenamento nas redes dos sensores. Em outras palavras, sensores que produzem mais atividades que suas capacidades, maximizam a utilização do sistema quando gravaram atividades nos sensores *host* disponíveis. Por outro lado, os sensores que produzem menos atividades, maximizam também a utilização do sistema, quando gravam atividades dos sensores com suas capacidades de armazenamento esgotadas. Então, o sistema fica mais robusto uma vez que se consegue prolongar sua eficácia, quando maximiza-se seu armazenamento total.

Mostramos que podemos usar a DSC como um *search function* efetiva. Este função consegue uma cobertura da rede da mesma maneira como em um passeio aleatório que é e conhecido por ter uma cobertura *rápida* [8]. A grande vantagem do nosso novo modelo é de ter a mesma forma de cobertura do passeio aleatório, além de ser determinístico. Em outras palavras, a reconstrução das cadeias DSC pode ser prevista sem necessidades de uso de tabelas de roteamento.

Redes de sensores que implementam DSC podem operar sob condições de *unfairness* entre os sensores. Nesse artigo, mostramos uma condição de *unfairness* quando um conjunto de sensores gera 80% das atividades. Ademais, nossas simulações mostram que a DSC pode tolerar também, até mesmo um único sensor que gera 80% da capacidade total do sistema.

REFERENCES

- [1] Gabber O. and Galil Z. : "Explicit constructions of linear size super-concentrators", Tel Aviv University, 1981
- [2] Naor M. and Wieder U. : "Novel Architectures for P2P Applications: The continuous-discrete approach (CDA)", Weizmann institute, 2003
- [3] Karp B., Estrin D., Yin L., Yu F., Rantnasamy S., Shenker S. and Govinadan R. : "Data centric storage in SensorNets with GHT, A Geographic Hash Table", MONET, 2003
- [4] Karp B. and Kung H. T. : "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", Harvard, 2000
- [5] M. de Berg : "Computational Geometry", Springer, 2000
- [6] Culler D., Estrin D., and Srivastava M. : "Overview of Sensor Networks", IEEE, 2004
- [7] Estrin D., Govindan R., Hiedemann J. and Kumar S. : "Next century challenges: scalable coordination in sensor network", USC, 1999
- [8] N. Alon, C. Avin, M. Koucký, G. Kozma, Z. Lotker, and M. R. Tuttle, "Many random walks are faster than one", in SPAA, 2008, pp. 119–128.
- [9] X. Wu, G. Chen, and S. K. Das, "Avoiding Energy Holes in Wireless Sensor Networks with Non-uniform Node Distribution", in IEEE transactions on parallel and distributed systems, vol. 19, no. 5, 2008
- [10] A. Dimakis, A. Sarwate, and M. Wainwright, "Geographic gossip", in Proc. Information Processing in Sensor Networks, 2006, pp. 69–76
- [11] B. Clark, C. Colbourn, and D. Johnson, "Unit disk graphs," Discrete Math., vol. 86, pp. 165–177, 1990