

# Economia de Energia em Redes de Sensores Sem Fio Através de Compressão Distribuída de Dados Utilizando Transformadas Wavelet

Alexandre Ciancio e Antonio Ortega

**Resumo**—Este trabalho aborda o problema do consumo de energia em redes de sensores sem fio, onde cada sensor tem potência limitada e adquire dados que devem ser transmitidos a um nó central. No cenário proposto, cada sensor pode escolher entre diferentes algoritmos de codificação, atingindo níveis diferentes de compressão. Os algoritmos considerados são baseados numa versão distribuída da transformada wavelet e exploram o fluxo natural através da rede para agregar dados. Demonstramos que através da seleção ótima de algoritmos é possível reduzir significativamente a energia consumida na rede comparada ao caso em que nenhum tipo de codificação é utilizada.

**Palavras-Chave**—Redes de Sensores Sem Fio, Wavelets Distribuídas, Algoritmos Distribuídos.

**Abstract**—We address the problem of energy consumption reduction for wireless sensor networks, where each of the sensors has limited power and acquires data that should be transmitted to a central node. In our scenario, sensors have a choice of different coding schemes to achieve varying levels of compression. The compression algorithms considered are based on a distributed version of the wavelet transform, and exploit the natural data flow in the network to aggregate data. We demonstrate that by optimizing the coding algorithm selection the overall energy consumption can be significantly reduced compared to the case where no encoding is performed.

**Keywords**—Wireless Sensor Networks, Distributed Wavelets, Distributed Algorithms.

## I. INTRODUÇÃO

Redes de Sensores Sem Fio oferecem mobilidade e versatilidade para uma variedade de aplicações, tais como detecção e acompanhamento de objetos, monitoramento ambiental e controle de tráfego [1]. Com o barateamento progressivo da tecnologia em sensores, aplicações que envolvem redes com grande número de nós se tornaram cada vez mais comuns. Entretanto, o seu principal desafio está no fato de frequentemente dependerem de baterias como fonte de energia, e a limitação de seu consumo se torna essencial para garantir sua durabilidade.

Quando os dados são adquiridos em múltiplas fontes correlacionadas, sua combinação através de métodos de compressão dentro da própria rede pode oferecer uma forma mais eficiente de representá-los. Uma vez que grande parte da energia consumida nos sensores é destinada à transmissão de dados, a redução da quantidade de informação que deve ser transmitida através da rede pode levar a uma redução significativa no consumo de energia. Pesquisas anteriores já propuseram diversos métodos para Codificação de Fontes Distribuídas (CFD) como forma de descorrelacionar dados. Enquanto algumas dependem de troca de informações entre sensores numa vizinhança para a computação de versões distribuídas de transformadas, como Karhunen-Loève [6] e Wavelets [10], outras propõem métodos que não requerem comunicação adicional entre nós, tal como codificação Slepian-Wolf para redes [5], [9]. Em geral, técnicas de CFD enfrentam um *trade-off* entre i) processamento em cada sensor para

Alexandre Ciancio - Laboratório de Processamento de Sinais, COPPE, Universidade Federal do Rio de Janeiro, Brasil. Antonio Ortega - Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, USA. E-mails: ciancio@lps.ufrj.br, ortega@sipi.usc.edu. Este trabalho foi parcialmente financiado pelo CNPq.

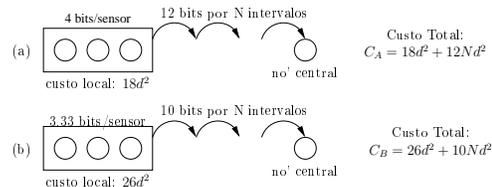


Fig. 1. (a) Método “A”: um sistema de codificação simples é usado; 12 bits são enviados ao nó central. (b) Método “B”: um sistema que consome mais energia localmente é usado, mas uma melhor compressão é obtida; 10 bits são enviados ao nó central. O custo da transmissão de  $k$  bits através de uma distância  $d$  foi computado como  $kd^2$ .

compressão de dados e ii) transporte de informação (bits) para o nó central. Este *trade-off* também foi abordado em outras pesquisas. Por exemplo, [8] fornece uma análise a respeito das regiões da rede que devem favorecer compressão sobre roteamento de caminho-mais-curto baseado no impacto da correlação espacial das medidas.

Entretanto, apesar de pesquisas anteriores haverem proposto inúmeros métodos para descorrelacionar dados na rede e/ou analisar individualmente sua performance, nenhum método conhecido pelos autores tratou do problema da atribuição ótima de algoritmos a diferentes sensores da rede tal que o consumo global de energia seja minimizado, quando métodos diferentes estão disponíveis. Uma vez que o *trade-off* entre distorção e consumo de energia depende de outros fatores tais como topologia da rede e características do meio, métodos de codificação diferentes podem oferecer melhores resultados para diferentes regiões da rede. Esses métodos podem consistir desde sistemas de codificação simples, tal como DPCM, até métodos mais complexos, como Transformadas Wavelet com diferentes níveis de decomposição.

Para melhor ilustrar como a topologia da rede pode influenciar a performance de um dado esquema de codificação, considere um simples exemplo, mostrado na Fig. 1. Um grupo de três sensores alinhados e igualmente espaçados (de uma distância  $d$  entre eles) está separado do nó central por  $N$  intervalos, e deve escolher entre dois modos de codificação para descorrelacionar seus dados. O método “A” é limitado em termos de performance, mas requer apenas um pequeno custo local (que leva em conta transmissões adicionais e computações locais) para o grupo. O método “B” é mais “caro” localmente em termos do consumo de energia, mas alcança uma melhor compressão, i.e., requer menos bits para representar os dados dos três sensores. É fácil mostrar que, neste exemplo, se os três sensores estão distantes a mais de quatro intervalos do nó central ( $N > 4$ ), o método “B” terá um menor consumo global de energia, apesar do maior custo local. Em geral, espera-se que sensores que estejam mais afastados do nó central se beneficiem de métodos que necessitem de menos bits para representar os dados para um dado nível de distorção, enquanto sensores mais próximos do nó central devem utilizar esquemas que ofereçam um menor custo local.

Uma conclusão semelhante foi obtida em [5], onde os dados são modelados e a análise da performance da codificação é baseada na

entropia dos dados. Uma contribuição significativa deste trabalho é que i) nenhum modelo a respeito da estatística dos dados é assumido (o algoritmo só necessita de dados para treinamento) e ii) o método compara diretamente a performance de sistemas de codificação reais (ao invés de considerar entropia) e oferece estimativas mais realistas para o consumo de energia na rede.

Em [2], [3] introduzimos algoritmos distribuídos para compressão de dados para redes sem fio que levavam em conta o consumo de energia na rede, e exploravam a redundância *espacial* entre os dados medidos. Em nossos trabalhos anteriores, introduzimos a abordagem de coeficientes parciais baseados na fatorização em passos de *lifting* [11] para a transformada wavelet. Nosso objetivo era gerar os coeficientes da transformada em cada sensor, às custas de poucas transmissões “locais”, i.e., transmissões de dados entre sensores vizinhos, necessárias para computar os coeficientes da transformada, uma vez que a transformada opera como uma filtragem através dos sensores. Se os dados originais são suficientemente correlacionados no espaço, após a quantização, os coeficientes da transformada wavelet podem representar as medidas originais utilizando um número menor de bits, e o consumo global de energia é reduzido através da redução da quantidade de informação que deve ser transmitida. O método do cálculo de coeficientes parciais [3] permite às operações da transformada a serem causais, no sentido de que os dados são processados à medida que são encaminhados ao nó central, de forma que apenas os dados de sensores que já foram atravessados são utilizados no cálculo dos coeficientes. Isso requer o cálculo e quantização de coeficientes aproximados, ou “parciais”, que são transmitidos à frente por alguns sensores antes de serem usados para gerar os coeficientes refinados, ou “finais”, da wavelet. Neste artigo, nos referimos aos coeficientes *parciais* como sendo os coeficientes que foram computados em sensores com dados insuficientes, e que ainda precisam ser refinados em outros sensores até que se tornem coeficientes finais.

Neste trabalho consideramos redes em que cada sensor pode utilizar um entre vários esquemas de compressão, que incluem transformadas wavelet 5/3 (também conhecida como CDF(2,2)) com números diferentes de níveis de decomposição, ou abordagens mais simples, como simples transmissão de dados quantizados (sem compressão). Propomos uma metodologia que permite encontrar, para uma dada topologia da rede, qual dentre os métodos de codificação disponíveis deve ser utilizado em cada um dos sensores de forma a minimizar o consumo global de energia para um determinado nível de distorção na representação dos dados. Nosso método é flexível o suficiente para acomodar qualquer configuração da rede (topologia).

Este artigo é organizado conforme se segue. A Seção II descreve brevemente o Algoritmo de Coeficientes Parciais, e introduz a metodologia para a otimização da rede, descrevendo-a como uma máquina de estados, associando cada modo de operação de um sensor a um estado, e custos de transmissão e processamento a pesos dos ramos ligando estados. O algoritmo proposto é introduzido na Seção III, e as estratégias de roteamento utilizadas nas simulações são discutidas na Seção IV. Os resultados obtidos nas simulações são apresentados na Seção V. A conclusão do artigo encontra-se na Seção VI.

## II. REPRESENTAÇÃO DE DADOS E METODOLOGIA DE OTIMIZAÇÃO

### A. O Algoritmo de Coeficientes Parciais

Neste artigo assumimos que uma rede de sensores adquire dados através da amostragem de um campo bidimensional correlacionado, onde cada ponto de amostragem é representado pela posição física do sensor no campo. Consideramos que a compressão de dados ocorre

ao longo de um caminho ligando um sensor na extremidade da rede (chamado de *folha*) ao nó central, ou *sumidouro* (veja Fig. 2). Uma abordagem heurística é definida para tratar dados na vizinhança de pontos onde dois caminhos se juntam (pontos de junção). Assume-se também que os caminhos ligando as folhas ao sumidouro são conhecidos, o que implica que algum algoritmo de roteamento já foi aplicado à rede. Cada sensor em um caminho é representado por um número  $n$ , começando pela folha. A topologia da rede, e, portanto, a distância entre os sensores também é conhecida, e cada sensor no caminho pode operar usando um entre vários esquemas de codificação pré-definidos. Os esquemas considerados consistem em transformadas wavelet discretas com diferentes níveis de decomposição: com o aumento do número de níveis, o potencial para compressão também aumenta (assumindo que os dados são suficientemente correlacionados), mas ao custo da necessidade de uma maior troca de informações entre sensores vizinhos (uma vez que dados de mais sensores são necessários para o cálculo dos coeficientes da transformada).

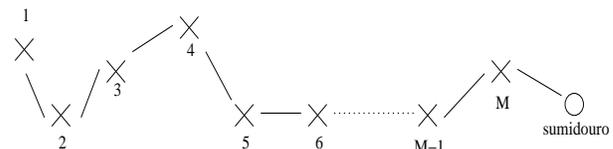


Fig. 2. Caminho ligando folha ao sumidouro.

Uma vez que a transformada wavelet utilizada é criticamente amostrada, o número de coeficientes é igual ao número de sensores. Usando a abordagem de coeficientes parciais [3], o coeficiente da transformada correspondente ao sensor  $n$  é calculado em passos: no sensor  $n$  uma versão aproximada do coeficiente é inicialmente gerada, e se torna um coeficiente final à medida que incorpora dados adicionais provenientes de sensores futuros (i.e., sensores mais próximos do sumidouro). O número de intervalos necessários para que um coeficiente parcial se torne final, bem como as operações computadas em cada sensor dependem da transformada sendo implementada. Referimos o leitor a [3] para maiores detalhes.

### B. Otimização de Caminhos Individuais

Para encontrar o esquema de codificação mais adequado para cada um dos sensores em um caminho, propomos representar este caminho como um grafo (Fig. 3), onde cada nó corresponde a um possível estado (método de codificação) de um sensor, e custos de comunicação e processamento estão associados aos pesos de cada ramo.

Assumimos que os métodos de codificação disponíveis na Fig. 3 correspondem a transformadas wavelet com número de níveis de decomposição progressivamente maiores, e que, à medida que um sensor se aproxima do sumidouro, ele pode apenas optar por continuar no mesmo esquema de codificação que o sensor anterior ou então por operar em um modo de codificação mais simples, mais especificamente pela transformada que use um nível de decomposição a menos que o sensor anterior. Conforme motivado pelo exemplo da Seção I, a intuição por trás desta limitação nas possíveis transições no grafo é que, além de reduzir o número de combinações entre possíveis modos de codificação em cada sensor no caminho, conforme os sensores se aproximam do sumidouro, métodos mais simples de codificação tendem a ser mais eficientes em termos do consumo de energia. Esta idéia também pode ser relacionada a resultados obtidos em [5].

Seguindo esta metodologia, cada estado pode ser completamente descrito por dois parâmetros: 1) a sua posição física no caminho e 2) o esquema de codificação sendo considerado. Todas as operações

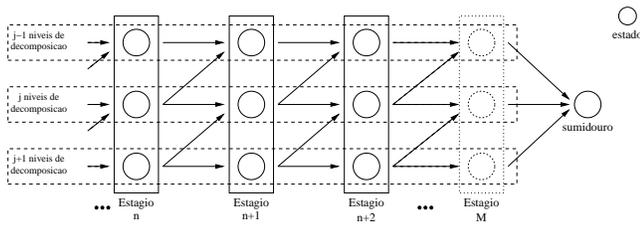


Fig. 3. Transições permitidas para a máquina de estados correspondente à um caminho da rede, contendo  $M$  sensores, e na qual três possíveis algoritmos de codificação podem ser escolhidos.

(e transmissões) que devem ser efetuadas em cada sensor para que seu coeficiente da transformada seja computado podem ser descritas conhecendo-se estes parâmetros. Cada transição (indicada por um ramo) corresponde a uma mudança no esquema de codificação que o próximo sensor passará então a utilizar, e seu peso é definido por uma métrica ou função-custo. A função-custo considerada neste trabalho leva em conta toda a energia consumida com transmissões e operações matemáticas associadas a cada transição entre os possíveis estados presentes na rede.

Além disso, uma importante característica do algoritmo de coeficientes parciais introduzido em [3] é que todos os cálculos em qualquer um dos sensores não requer nenhum tipo de informação de sensores futuros (mais próximos do sumidouro). Esta propriedade nos permite descrever os custos de transições e estados de tal forma que para qualquer estado presente considerado, o custo total até aquele ponto não é afetado pela escolha de estados futuros, tornando possível a escolha da sequência ótima de estados que minimiza o custo total através de técnicas de programação dinâmica, mais especificamente de um algoritmo do caminho-mais-curto. A cada estágio  $n$  de decisão, o algoritmo calcula a melhor transição entrando em cada estado naquela posição através da minimização das métricas de todos os ramos possíveis entrando no estado, e selecionando como sobrevivente o ramo que apresenta a métrica mínima. No último estágio, o caminho que apresentar a menor métrica é selecionado como o caminho ótimo. Uma vez que cada estado descreve o sistema de codificação usado em cada sensor, o caminho ótimo fornece também a seleção ótima de modos de codificação tal que o consumo de energia seja minimizado.

Uma descrição mais detalhada sobre programação dinâmica e sobre os estados e transições utilizados pode ser encontrada em [4].

### C. Pontos de Junção

Em redes genéricas, bidimensionais, caminhos individuais podem fundir-se antes de alcançar o nó central. Extendemos agora a representação unidimensional apresentada anteriormente para o caso geral.

A Fig. 4 ilustra o caso onde dois caminhos se encontram em um ponto de junção, prosseguindo até o sumidouro como um único caminho. Para efeitos de ilustração, e sem perda de generalidade, assumiremos que o mesmo algoritmo de codificação é utilizado nos dois caminhos. As informações capturadas nos Caminhos 1 e 2 podem ser correlacionadas, mas não se espera que sejam idênticas. Logo, os coeficientes da wavelet recebidos pelo sensor  $n$  provenientes dos dois caminhos serão, em geral, diferentes. Assuma inicialmente que as informações de cada caminho são processadas independentemente, i.e., os coeficientes recebidos do Caminho 1 e do Caminho 2 são refinados para incorporar as informações dos sensores  $n$ ,  $n + 1$ , e assim por diante. Uma vez que esses novos coeficientes são gerados a partir de dados comuns aos dois caminhos (provenientes dos sensores  $n$ ,  $n + 1$ , etc.), e os filtros da transformada wavelet têm

memória finita, espera-se que, após um certo número de iterações, por exemplo  $n + k$  (onde  $k$  depende do comprimento do filtro), os novos coeficientes produzidos sejam idênticos, independente se os dados iniciais foram originados do Caminho 1 ou do Caminho 2. Então, a partir deste ponto, os coeficientes de cada caminho podem ser unicamente representados, de forma que apenas um conjunto de coeficientes é necessário para a representação dos dados para os sensores a partir da posição  $n + k$ .

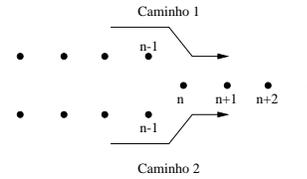


Fig. 4. Múltiplos coeficientes são gerados a partir de pontos de junção.

Além disso, os Caminhos 1 e 2 são em geral correlacionados, e mesmo que esta correlação não seja elevada, o conjunto de coeficientes (um por caminho) gerados nos sensores  $n$  até  $n + k - 1$  serão cada vez mais similares. Isso sugere uma abordagem simples para reduzir ainda mais o custo decorrente dos múltiplos coeficientes gerados nas proximidades de pontos de junção. Em qualquer sensor onde múltiplos coeficientes sejam gerados (devido à junção de dois ou mais caminhos), esse conjunto de coeficientes é codificado utilizando DPCM (*Differential Pulse Code Modulation*), i.e., a diferença entre coeficientes é quantizada e utilizada como representação da sequência de medidas (para um mesmo sensor). Se os valores são próximos, esta diferença é pequena, e pode ser mais eficientemente representada por DPCM. Estes dados adicionais (originados devido à multiplicidade dos coeficientes) são enviados ao nó central sem ser modificados por nós futuros.

## III. ALGORITMO PROPOSTO

O algoritmo proposto assume que o roteamento dos sensores na rede já tenha sido definido previamente (encontra-se em fase de pesquisa um algoritmo que otimiza conjuntamente o roteamento e a seleção de métodos de codificação). Dada a topologia da rede, aplica-se a otimização descrita na Seção II-B (baseada no algoritmo de coeficientes parciais [3]) *independentemente* a cada um dos caminhos conectando cada folha ao nó central, i.e., a representação de dados para alguns sensores será efetuada múltiplas vezes (por exemplo, a seleção para os sensores  $n$ ,  $n + 1$ , etc. na Fig. 4 será considerada duas vezes, correspondendo aos Caminhos 1 e 2). Dessa forma, alguns sensores, por exemplo o sensor  $n$  na Fig. 4, poderão ter atribuídos diferentes esquemas de codificação quando otimizados segundo diferentes caminhos. Quando isto ocorre, atribuímos a este sensor a representação mais simples (i.e. a transformada wavelet com o menor número de níveis de decomposição). Com esta heurística forçamos caminhos que se unem a utilizar, a partir do ponto de junção, o método de codificação que apresenta o menor esforço computacional entre os possíveis candidatos.

Obviamente, nos casos em que todos os caminhos que passam por um determinado sensor atribuem a ele o mesmo esquema de codificação, este será utilizado na compressão de seus dados.

O algoritmo proposto pode ser resumido conforme a seguir:

### Algoritmo 1

#### 1) Otimização:

para cada caminho até o sumidouro:

- encontre o esquema de codificação ótimo para cada sensor;
- se há conflito de esquemas com caminhos anteriores:

-use esquema mais simples;

**fim**

## 2) Codificação:

para cada caminho:

-codifique dados usando esquema de codificação encontrado no Passo 1 para cada sensor;

-se há coeficientes múltiplos, codifique-os usando DPCM;

**fim**

## IV. ROTEAMENTO

Quando o método de otimização proposto é utilizado em uma rede bidimensional, há um custo extra de processamento sempre que dois caminhos diferentes se combinam a caminho do nó central. Em alguns casos particulares, conforme será visto na Seção V, este custo de processamento e transmissão adicional necessário para representar coeficientes múltiplos originados devido à pontos de junção podem fazer com que o sistema se torne ineficiente, de forma que uma topologia (roteamento) com um menor número de pontos de junção possa vir a ser preferível mesmo que possua distâncias de transmissão maiores na média.

Para que o impacto deste custo adicional devido a pontos de junção possa ser melhor avaliado, consideramos uma variedade de topologias, chamadas de *árvores de grau restrito*. Essas árvores são geradas através de uma versão modificada do Algoritmo de Dijkstra para grafos. Considere um grafo  $G(V, E)$  com pesos de ramos  $EW(i, j)$  para os nós  $i, j \in V$  e ramos  $(i, j) \in E$  e grau máximo  $maxDegree$ . Cada nó  $n$  mantém um valor para o peso do caminho-mais-curto até o sumidouro  $WPD(n)$  (com valor inicial =  $\infty$ ) e o seu número de descendentes  $C(n)$  (valor inicial = 0) na árvore que se inicia no sumidouro. Começando com um conjunto  $T$  que inicialmente contém apenas o nó central  $S$ , a cada passo adiciona-se a  $T$  o nó  $p \notin T$  para o qual

- há um nó  $p' \in T$  tal que o ramo  $(p, p') \in E$ ,
- se  $p' \neq S$  então  $C(p') < maxDegree - 1$ , e
- o peso do caminho-mais-curto até o sumidouro  $[= EW(p, p') + WPD(p')]$  é minimizado.

As atualizações são feitas conforme  $WPD(p) = WPD(p') + EW(p, p')$  e  $C(p) = C(p') + 1$ . O algoritmo para quando  $|S| = |V|$  ou quando nenhum nó pode ser adicionado (já que todos os seus vizinhos atingiram a restrição de grau). Este último caso é evitado quando considera-se apenas grafos bem-conectados. Quando  $maxDegree$  é maior que o grau máximo dos nós em  $G$ , o algoritmo se reduz a encontrar o caminho-mais-curto e quando  $maxDegree = 2$ , ele resulta em caminhos longos, sem pontos de junção, exceto no nó central. Nos experimentos que se seguem, gerou-se árvores usando o número máximo de pontos de junção =  $maxDegree - 1$  como parâmetro.

## V. EXPERIMENTOS

Durante os experimentos, utilizou-se dados simulados através de um modelo auto-regressivo (AR) de segunda ordem, além de dados empíricos, provenientes de uma rede de sensores real. Para a obtenção dos dados simulados, gerou-se três processos com  $600 \times 600$  pontos utilizando o modelo AR de segunda ordem com baixa, média e alta correlação entre dados. Os nós foram colocados no grid de dimensão  $600 \times 600$ , e o valores de cada medida da rede correspondia ao valor do processo associado à sua posição no grid.

Os dados reais fazem parte de um subconjunto contendo 19 sensores de uma rede de monitoramento de habitat [7], [12] na Ilha Great Duck. O conjunto de dados utilizado nos experimentos consiste em 200 leituras de temperatura tomadas em cada sensor em 1<sup>o</sup> de agosto de 2003, a intervalos de aproximadamente 5 minutos.

No caso de dados simulados, a localização dos sensores consiste no posicionamento aleatório de 100 sensores no grid. O algoritmo de roteamento utilizou árvores de grau restrito. Para valores suficientemente grandes do do número máximo de junções, a árvore se reduzia árvore de caminho-mais-curto. Os custos de ramos utilizado na otimização da representação da rede foram proporcionais ao quadrado da distância entre os sensores, com a constante de proporcionalidade sendo o número de bits alocados para o nó transmitindo dados em direção ao nó central.

Na Seção V-A analisamos a performance do método proposto para uma dada topologia. Isso nos permite demonstrar que ganhos são atingíveis através da seleção de diferentes modos de operação para diferentes sensores. Então, na Seção V-B, selecionamos o modo de codificação para cada sensor como sendo o obtido com o método de otimização proposto, e comparamos a performance de estratégias diferentes de roteamento, utilizando diferentes limites para o número máximo de pontos de junção em cada caso. Isso nos permite demonstrar que, em certos casos, o roteamento do caminho-mais-curto não é a escolha ótima, uma vez que este método pode apresentar um número indesejável de pontos de junção.

Em todas as representações de rede nas figuras que se seguem, nós marcados com “×” são codificados com uma wavelet 5/3 com um nível de decomposição, enquanto nós marcados com “o” são codificados com uma wavelet 5/3 com dois níveis de decomposição. Transmissão de dados não codificados (i.e., não comprimidos) é representada com o símbolo “◇”, embora nenhum sensor tenha sido atribuído este modo de codificação nos casos simulados.

### A. Performance do Algoritmo Dada para uma Dada Topologia

As Figuras 5–7 ilustram a performance de diferentes algoritmos em termos da distorção na representação dos dados (medida através da relação sinal-ruído, SNR) em função da energia consumida pela rede. Em cada figura representamos o roteamento e a atribuição ótima de métodos de codificação para cada sensor. Os resultados plotados foram obtido através da média entre várias realizações do experimento. Observe que em todos os casos, a rede com a configuração otimizada apresenta performance superior aos casos em que apenas um método de codificação é utilizado em toda a rede. Note, entretanto, que nas Figuras 5–7 os ganhos são modestos. Isso também pode ser visto notando-se que apenas alguns poucos sensores fazem uso de wavelets com 2 níveis de decomposição.

Nas Figs. 5–6 os sensores foram uniformemente distribuídos. Ao contrário, na Fig. 7 eles foram divididos em dois grupos separados por um longo intervalo, e o nó central foi colocado na região com menor densidade de sensores. Observe que o comportamento na Fig. 7 é intuitivamente coerente: no grupo com maior densidade, na região ao alto e à direita do campo, uma wavelet com 2 níveis de decomposição é escolhida para minimizar a taxa global necessária para a representação dos dados. Isto se deve ao fato deste grupo se encontrar distante do nó central, e qualquer aumento na taxa (devido ao uso de um algoritmo menos eficiente) resultaria em um aumento significativo do consumo de energia necessária à transmissão.

As Figuras 8 e 9 ilustram a performance dos métodos para dados reais. Pode ser observado que quando nenhum ponto de junção é permitido (Fig. 8), o caso com a atribuição ótima de algoritmos apresenta uma performance consideravelmente maior que os outros esquemas. Isto confirma nossa expectativa inicial de que pontos de junção tendem a ser caros, do ponto de vista do consumo de energia. Entretanto, como será visto, roteamento do caminho-mais-curto ainda apresenta o melhor resultado global.

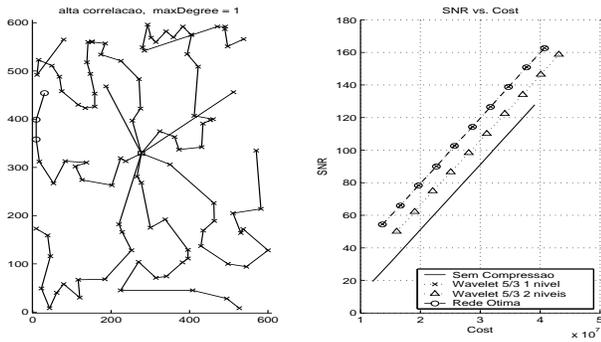


Fig. 5. Performance de uma Rede Aleatória sem Pontos de Junção.

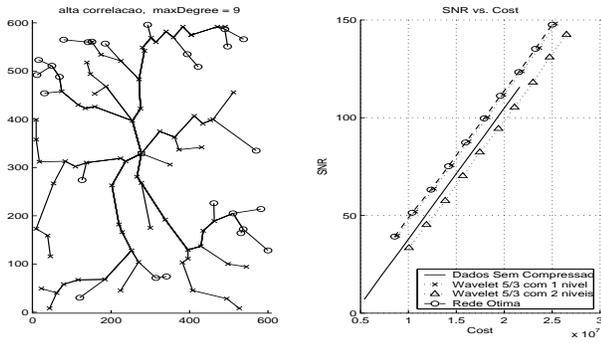


Fig. 6. Performance de uma Rede Aleatória com Roteamento do Caminho-Mais-Curto.

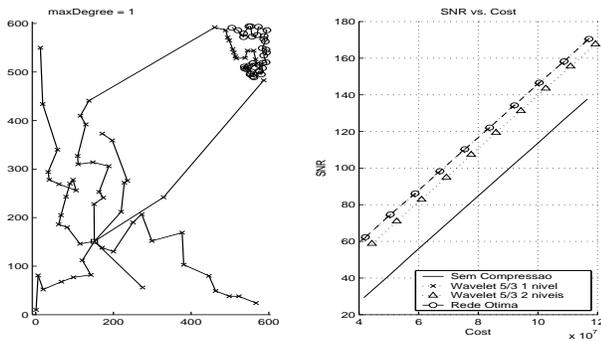


Fig. 7. Performance de uma Rede Segmentada com Roteamento do Caminho-Mais-Curto.

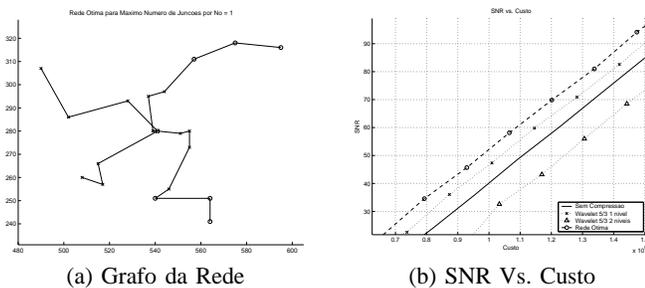


Fig. 8. Performance do Algoritmo para Rede com Dados Reais sem Pontos de Junção.

**B. Comparação entre Roteamentos para Representação Ótima da Rede**

Agora que estabelecemos que a atribuição ótima de algoritmos na rede (i.e., escolha de modos de codificação para cada sensor)

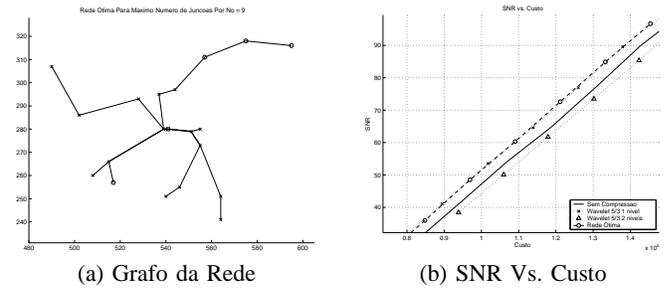


Fig. 9. Performance do Algoritmo para Rede com Dados Reais para Roteamento do Caminho-Mais-Curto.

minimiza o consumo de energia para uma dada topologia, comparamos a performance de diferentes topologias em termos do consumo de energia quando utilizando esta representação ótima. (Figs. 10–12). Em particular, comparamos a performance entre roteamentos nos casos do caminho-mais-curto e de quando não há pontos de junção.

Para os casos onde se utilizou tanto dados simulados (Fig. 10) e empíricos (Fig. 11), podemos observar que o caminho-mais-curto apresenta a melhor performance. Acreditamos que no caso da Fig. 10, devido à maior densidade da rede, o custo adicional devido aos pontos de junção não é tão significativo (devido ao uso de DPCM), e a penalidade por permitir caminhos mais longos (como na configuração onde não há pontos de junção) é muito alta para ser compensada pela redução do número de junções. A Fig. 12 mostra os resultados para a rede com duas regiões com densidades diferentes de sensores. Note que neste caso, ao contrário dos anteriores, o roteamento sem pontos de junção leva a uma melhor performance do que o caminho-mais-curto.

Note também que a inclinação das retas nas Figs. 10–12 é menor para roteamentos sem pontos de junção (i.e., a qualidade se degrada mais lentamente com a diminuição do custo). Atribuímos mais uma vez este comportamento ao custo adicional devido ao maior número de pontos de junção no caso do roteamento do caminho-mais-curto.

Performance da rede com atribuição ótima de algoritmos

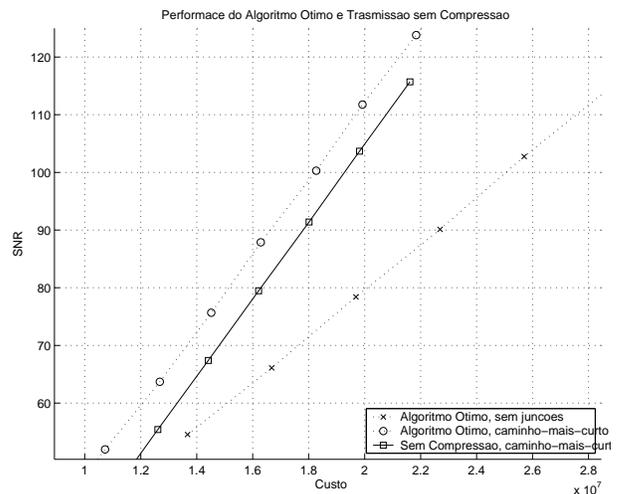


Fig. 10. Performance da rede com atribuição ótima de algoritmos para dados simulados comparado ao caso onde dados são transmitidos sem codificação nos casos de roteamento sem pontos de junção e do caminho-mais-curto.

**VI. CONCLUSÕES**

Neste trabalho propusemos uma metodologia que permite encontrar, para uma rede com uma dada topologia, qual método de

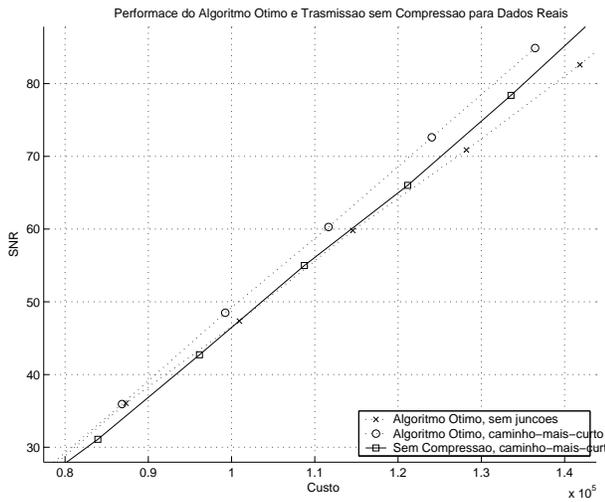


Fig. 11. Performance da rede com atribuição ótima de algoritmos para dados reais comparado ao caso onde dados são transmitidos sem codificação nos casos de roteamento sem pontos de junção e do caminho-mais-curto.

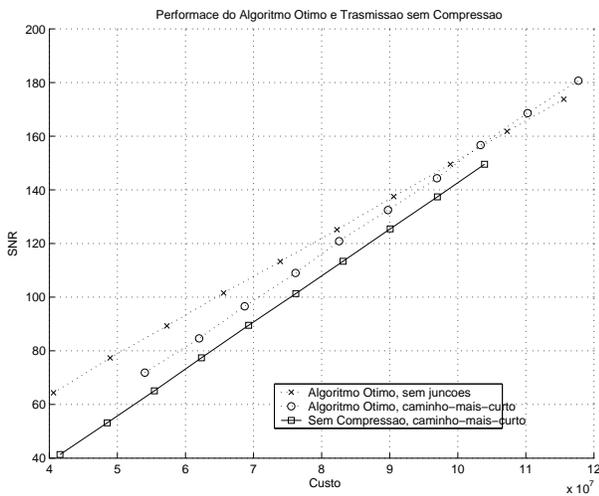


Fig. 12. Performance da rede com atribuição ótima de algoritmos para para a rede segmentada comparado ao caso onde dados são transmitidos sem codificação nos casos de roteamento sem pontos de junção e do caminho-mais-curto.

codificação, dentre um número disponível de métodos, deve ser utilizado em cada sensor da rede de forma a minimizar o consumo global de energia quando comparado ao caso onde nenhum método de codificação é utilizado (simples quantização dos dados). Durante as simulações comparamos diferentes estratégias de roteamento e identificamos aquelas mais eficientes para uma determinada localização de sensores. Verificamos que o método proposto sempre encontra o melhor esquema de codificação para cada nó da rede, e que roteamentos do caminho-mais-curto nem sempre levam ao menor consumo de energia.

Neste cenário, inúmeros tópicos ainda podem ser estudados. Apesar de considerarmos neste trabalho que a topologia da rede era conhecida, a otimização conjunta entre modos de codificação e roteamento seria o próximo passo natural. Além disso, funções-custo mais realistas (que, por exemplo, levem em conta o hardware específico dos sensores) podem ser incorporadas ao algoritmo de otimização, e diferentes regiões da rede podem ser modeladas de formas diferentes para replicar de forma mais eficiente fenômenos encontrados na natureza.

Finalmente, apesar de termos assumido que a comunicação entre sensores ocorre por intervalos (i.e., um sensor transmite seus dados para o próximo, que por sua vez transmite os dados recebidos para o próximo, até que se alcance o nó central), aplicações práticas podem se beneficiar do fato de que transmissões são feitas omnidirecionalmente, levando a uma melhor performance. Por exemplo, coeficientes que já foram completamente refinados não precisam estar limitados ao roteamento especificado para a rede, podendo ser enviados ao nó central através do caminho-mais-curto; ou então, nós que necessitam dos dados de um sensor comum podem adquiri-los através de *broadcasting* em um passo único, também reduzindo o consumo de energia.

## REFERÊNCIAS

- [1] C. Chong and S. P. Kumar. "Sensor networks: Evolution, opportunities, and challenges." *Proceedings of the IEEE*, 91(8):1247–1256, Agosto 2003.
- [2] A. Ciancio and A. Ortega. "A distributed wavelet compression algorithm for wireless sensor networks using lifting." In *International Conference on Acoustics, Speech and Signal Processing - ICASSP04*, Montreal, Canada, Maio 2004.
- [3] A. Ciancio and A. Ortega. "A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting." In *International Conference on Acoustics, Speech and Signal Processing - ICASSP05*, Philadelphia, USA, Março 2005.
- [4] A. Ciancio and A. Ortega. "A dynamic programming approach to distortion-energy optimization for distributed wavelet compression with applications to data gathering in wireless sensor networks." *International Conference on Acoustics, Speech and Signal Processing - ICASSP06*, 2006.
- [5] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. "Networked Slepian-Wolf: Theory and algorithms." *1st European Workshop on Sensor Networks EWSN 2004*, 2004. Berlin
- [6] M. Gastpar, P. Dragotti, and M. Vetterli. "The distributed Karhunen-Loève transform." In *International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin Islands, Dezembro 2002.
- [7] Habitat Monitoring on Great Duck Island. Banco de dados online, localido em <http://www.greatduckisland.net>.
- [8] S. Pattem, B. Krishnamachari, and R. Govindan. "The impact of spatial correlation on routing with compression in wireless sensor networks." *Versão Journal*, Submetido, 2006.
- [9] S. S. Pradhan, J. Kusuma, and K. Ramchandran. "Distributed Compression in a Dense Microsensor Network." *IEEE Signal Processing Magazine*, páginas 51–60, Março 2002.
- [10] S. D. Servetto. "Sensing Lena - massively distributed compression of sensor images". *ICIP - International Conference on Image Compression*, Setembro 2003.
- [11] W. Sweldens, "The lifting scheme: A construction of second generation wavelets." *Technical Report 1995:6, Industrial Mathematics Initiative*, Department of Mathematics, University of South Carolina, ([ftp://ftp.math.sc.edu/pub/imi\\_95/imi95\\_6.ps](ftp://ftp.math.sc.edu/pub/imi_95/imi95_6.ps)), 1995.
- [12] R. Szwedczyk, A. Mainwaring, J. Polastre, and D. Culler. "An analysis of a large scale habitat monitoring application". In *Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Novembro 2004.