

VANDERLEI ZARNICINSKI

**UM ESTUDO SOBRE O USO DA CODIFICAÇÃO
DE CANAL PARA OTIMIZAÇÃO DO CONSUMO
DE ENERGIA EM REDES DE SENSOES**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito parcial para obtenção do título de Mestre em Informática.

CURITIBA

2007

VANDERLEI ZARNICINSKI

**UM ESTUDO SOBRE O USO DA CODIFICAÇÃO
DE CANAL PARA OTIMIZAÇÃO DO CONSUMO
DE ENERGIA EM REDES DE SENSORES**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração:
Redes de Computadores

Orientador: Prof. Dr. Marcelo Pellenz

CURITIBA

2007

Zarnicinski, Vanderlei

Um estudo sobre o uso da codificação de canal para otimização do consumo de energia em redes de sensores. Curitiba, 2007. 60 p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Codificação de Canal 2. Redes de Sensores 3. Controle de Erros 4. Redes Sem Fio 5. Consumo de Energia 6. Redes Ad-hoc I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática.

Aos meus pais, por toda uma vida de esforço e perseverança, o que me permitiu chegar até aqui.

À minha esposa e aos meus filhos, pelo apoio e compreensão nos momentos de ausência.

Agradecimentos

A todos meus colegas de curso, de quem sempre tive apoio e companheirismo.

A todos funcionários da PUC-PR, em especial os da secretaria do programa e da biblioteca, pela atenção e cordialidade.

Aos professores, com os quais pude absorver novos conhecimentos e idéias, transmitidos com dedicação em suas aulas.

Ao meu orientador, Prof. Dr. Marcelo Pellenz, pela dedicação e paciência em indicar e corrigir o rumo que resultou neste trabalho.

À minha família, pelo incentivo permanente e momentos de fraterna amizade.

À todos aqueles que me precederam no estudo e cujos trabalhos pude consultar para embasamento deste.

À PUC-PR pela oportunidade concedida.

Sumário

Agradecimentos	v
Sumário.....	vi
Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Abreviaturas e Siglas	x
Resumo.....	xi
Abstract.....	xii
Introdução.....	1
1.1. Desafio.....	1
1.2. Motivação	3
1.3. Proposta	5
1.4. Justificativa	6
1.5. Organização	6
Modelagem do Canal de Rádio	7
2.1. Introdução	7
2.2. Perda de Percurso.....	8
2.3. Relação Sinal-Ruído.....	10
2.4. Métricas de Desempenho.....	11
2.5. Conclusão	12
Estratégias de Controle de Erros	14
3.1. Introdução	14
3.2. Codificação Convolutional	15
3.3. Decodificação convolutional.....	17
3.4. Complexidade de decodificação	20
3.5. Simulação	22

3.6. Conclusão	26
Otimização do Consumo de Energia.....	27
4.1. Introdução	27
4.2. Modelagem do Consumo de Energia Para Processamento	29
4.3. Modelagem do Consumo de Energia Para Transmissão	30
4.4. Consumo Total de Comunicação	31
4.5. Resultados.....	34
4.6. Conclusão	39
Conclusões e Trabalhos Futuros.....	41
5.1. Conclusões.....	41
5.2. Trabalhos Futuros.....	43
Referências Bibliográficas	44

Lista de Figuras

Figura 1.1	: Arquitetura típica de um sensor.....	2
Figura 1.2	: Estrutura de camadas de um sensor.....	3
Figura 1.3	: Sensores utilizados em monitoração de habitat.....	3
Figura 1.4	: Micro-sensor.....	4
Figura 1.5	: Micro-sensor encapsulado com bateria e oscilador.....	4
Figura 2.1	: Tensão rms do envelope do sinal recebido	8
Figura 2.2	: Potência média do sinal recebido, $n = 4$, $\sigma = 4$, $P_t = 0$ dBm.	10
Figura 3.1	: Codificador convolucional (2,1,2,5).....	15
Figura 3.2	: Diagrama de estados para codificador convolucional (2,1,2,5)	17
Figura 3.3	: Diagrama de treliça para o código (2,1,2,5).....	19
Figura 3.4	: Complexidade de decodificação em função da memória do código	21
Figura 3.5	: Modelo de simulação	22
Figura 3.6	: Desempenho teórico e simulado no canal Rayleigh	24
Figura 3.7	: Taxa de erro de bit para dados codificados em canal Rayleigh	25
Figura 3.8	: Taxa de erro de quadro para dados codificados em canal Rayleigh.....	26
Figura 4.1	: Número esperado de transmissões do quadro no canal Rayleigh.....	32
Figura 4.2	: Consumo médio de energia de comunicação para o canal Rayleigh.....	33
Figura 4.3	: Encaminhamento de mensagem em múltiplos saltos	34
Figura 4.4	: Exemplo de rede com rota com múltiplos saltos.....	36
Figura 4.5	: Exemplo de rede com rota interrompida.....	38

Lista de Tabelas

Tabela 3.1	: Sequências geradoras de código convolucional.....	24
Tabela 4.1	: Valores de sinal-ruído em função da distância.....	35
Tabela 4.2	: Valores de sinal-ruído em uma rota com múltiplos saltos.....	37
Tabela 4.3	: Valores de memória de código ajustados aos canais.....	39

Lista de Abreviaturas e Siglas

ARQ	Automatic Repeat-Request
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CRC	Ciclycal Redundancy Check
DARPA	Defense Advanced Research Projects Agency
DSN	Distributed Sensor Network
FEC	Forward Error Correction
FER	Frame Error Rate
MLD	Maximum Likelihood Decoder
RSSF	Rede de Sensores Sem Fio
WSN	Wireless Sensor Network

Resumo

Redes de Sensores Sem Fio (RSSF) são formadas por um conjunto de dispositivos micro-eletrônicos, que monitoram determinado fenômeno físico num ponto ou região de interesse, enviando estas informações para um nó coletor. Os nós sensores estabelecem conexões entre os nós dentro do alcance do sinal do transmissor, promovendo a operação colaborativa da rede. Os dados coletados são repassados para outros nós, até atingirem o seu destino, o que representa inúmeros saltos entre diversos nós. Além do transceptor de rádio, cada nó sensor incorpora um módulo responsável pela coleta da grandeza monitorada, um módulo de processamento no qual é executado o código responsável pela operação do sensor e um módulo de alimentação do conjunto, normalmente baseado em baterias. O consumo de energia é diferente para cada módulo, sendo o transceptor o que apresenta maior demanda, variável de acordo com a potência do sinal transmitido. O canal de rádio entre os nós pode sofrer diversas influências do meio, como ruídos impulsivos aleatórios, interferência e desvanecimento, as quais provocam a degradação do sinal, a limitação no seu alcance e maior dificuldade de recepção da informação. Podem ocorrer então diversas retransmissões e aumentar o consumo de energia no sensor, o que pode levar ao esgotamento prematuro de sua bateria e abreviar a vida útil tanto do sensor como da rede.

Uma das estratégias para reduzir os efeitos desta degradação é codificar os dados a serem transmitidos através de códigos corretores de erro, que possibilitam a recuperação de informações corrompidas durante a transmissão, ainda que isto implique em um maior esforço computacional para codificação e decodificação. Neste trabalho investigamos a relação entre os custos de energia envolvidos na transmissão e no esforço de processamento para decodificação de dados, utilizando códigos convolucionais de diversas complexidades. Como resultado desta investigação, podemos identificar e selecionar a complexidade mais apropriada em cada nó sensor, de maneira a maximizar a vida útil da rede de sensores.

Palavras chave: 1. Rede de Sensores 2. Consumo de Energia 3. Codificação Convolucional 4. Controle de Erros

Abstract

Wireless Sensor Networks (WSN) are a set of tiny electronic devices, which monitor physical phenomena over a specific region and send this data to a sink node. The sensor nodes connect to the other ones in the range of the transmitted signal, establishing a collaborative network operation. The collected data are forwarded to other nodes until the sink is reached, what means several hops among the nodes. Besides the radio transceiver, a sensor node contains an event collecting module, a processing unit, which runs the sensor operation code and a power supply module, usually battery based. The energy consumption is not the same for every modules, being the radio transceiver that normally presents the heaviest demand, although it varies according the transmitted signal power. The radio channel between the nodes may experience several impairments, such as random noise, interference and fading, degrading the received signal and shortening its range, making the data reception difficult. Therefore it may occur many retransmissions and increase the sensor energy consumption, what can drain the battery earlier and reduce the sensor or even the network lifetime.

One of the ways to reduce the channel impairment effects is by coding the data through a error correction code, allowing the corrupted information recovering at reception, although this means more processing power to code and decode. In this work we investigate the transmission and decoding energy consumption tradeoff, employing convolutional codes with several complexities and performances. As result of this investigation, we can identify and choose the most suitable code complexity for each sensor node, in order to maximize the sensor network lifetime.

Keywords: 1. Sensor Networks 2. Energy Consumption 3. Convolutional Coding 4. Error Control.

Capítulo 1

Introdução

1.1. Desafio

O início das pesquisas sobre Redes de Sensores pode ser situado em 1980, com o projeto Distributed Sensor Networks (DSN), patrocinado pela DARPA (*Defense Advanced Research Projects Agency*), dos Estados Unidos da América. O objetivo deste projeto era o desenvolvimento de uma rede colaborativa de sensores de baixo custo, com operação similar ao contemporâneo projeto Arpanet [CHO03]. O desenvolvimento desta rede, no entanto, se limitava à tecnologia de componentes disponíveis, tanto no aspecto de capacidade de processamento como na comunicação.

O contínuo desenvolvimento dos dispositivos micro-eletrônicos e o aumento da sua escala de integração, permitiu nos últimos anos a implementação de sensores com maior capacidade de processamento e comunicação via canal de rádio. Tais sensores permitiram o surgimento de Redes de Sensores Sem Fio (RSSF), com um conjunto de dezenas até milhares de nós operando de forma colaborativa.

O nó sensor se baseia em quatro módulos componentes: sensor do fenômeno monitorado e conversor analógico-digital, unidade de processamento, unidade de comunicação e unidade de alimentação, normalmente baseada em pequenas baterias, comum aos outros três, como mostrado na Figura 1.1. Além destes, eventualmente podem ser incorporados também os módulos de mobilidade, de localização e de geração de energia, como por exemplo conversores foto-voltaicos [AKY02].

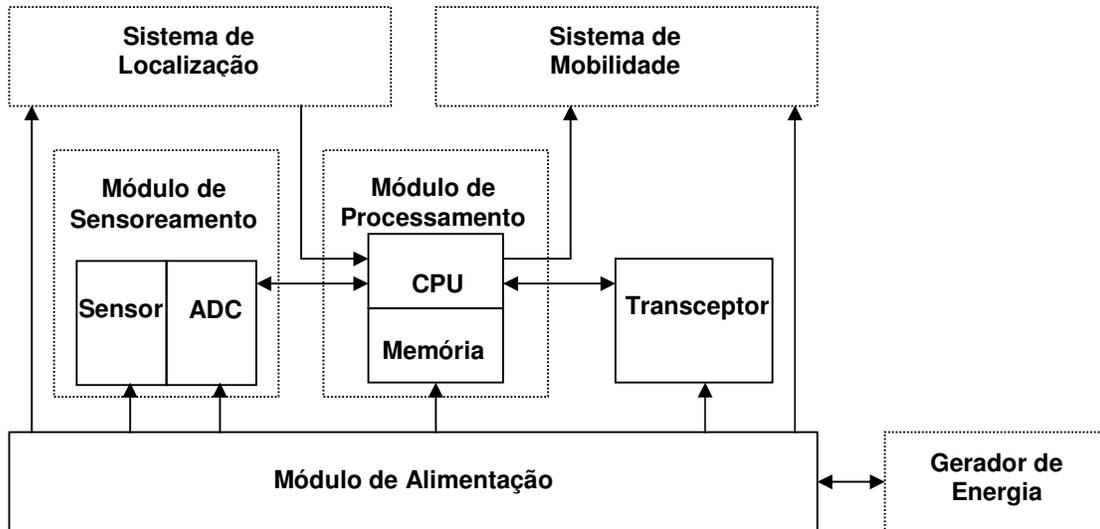


Figura 1.1: Arquitetura típica de um sensor

Nos últimos anos o tema Redes de Sensores Sem Fio vem recebendo atenção crescente, com diversos trabalhos publicados que abordam o conceito de rede de sensores e seus aspectos gerais [AKY02] [KAR03] [CUL04], ou que buscam novas soluções para as várias dificuldades que se apresentam, como controle de acesso ao meio [POL04] [YE03] [RAJ03], controle de topologia [XU03], sincronismo entre os nós [ELS02] [LUC04] e roteamento [HEI00] [GAN04] [BIS05]. Como um dispositivo de rede, o sensor apresenta uma estrutura em camadas nas quais se inserem estas soluções, além de planos transversais comuns a todas [AKY02], conforme mostrado na Figura 1.2.

Nas apresentação destas soluções, uma preocupação sempre ressaltada é a conservação de energia, que se insere no *Plano de Gerenciamento de Energia*, onde se busca a operação mais eficiente da rede em relação ao consumo. O desafio então é focado na elaboração de uma proposta para a otimização do consumo de energia no sensor.

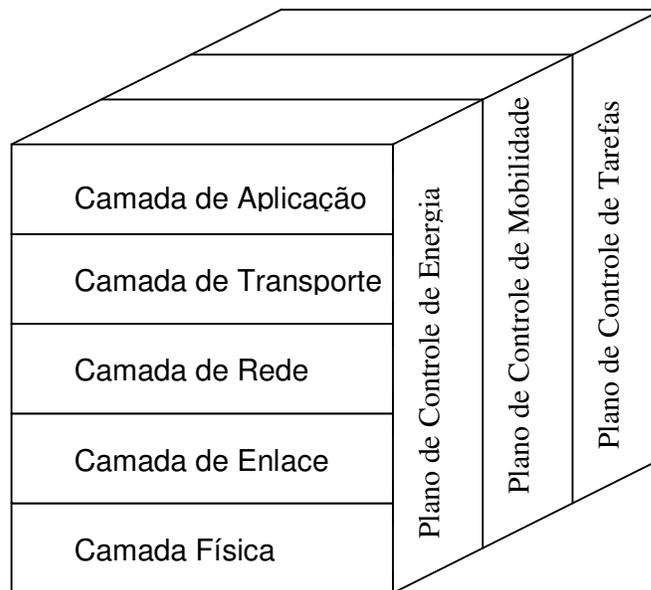


Figura 1.2: Estrutura de camadas de um sensor

1.2. Motivação

Os sensores são normalmente alimentados por baterias e instalados em locais onde não é possível ou não é viável a substituição das mesmas, quando esgotadas. Um exemplo da utilização de sensores para monitoração de habitat é mostrado na Figura 1.3, como relatado em [SZE04]. Micro-sensores ainda podem ser embutidos em alguma estrutura ou “lançados” em ambientes inóspitos [WAR03] conforme mostrado nas Figuras 1.4 e 1.5.



Figura 1.3: Sensores utilizados em monitoração de habitat

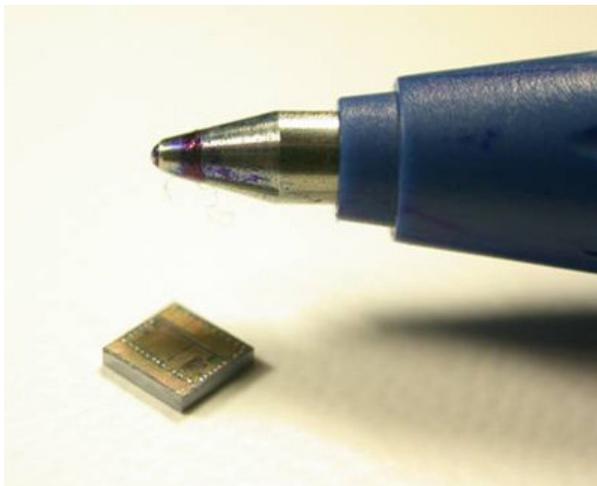


Figura 1.4: Micro-sensor

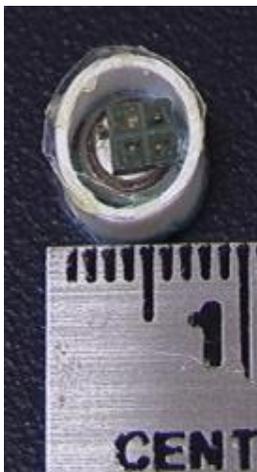


Figura 1.5: Micro-sensor encapsulado com bateria e oscilador

A rede formada por estes dispositivos apresenta uma vida útil limitada, determinada pela forma de operação dos sensores e o decorrente esgotamento da energia disponível na bateria.

Apesar das similaridades de uma rede sem fio *ad-hoc* com a rede de sensores, como a rápida implementação independente de infra-estrutura, a proposta de emprego da rede de sensores é a utilização de muitos dispositivos de baixo custo com o foco da operação nos dados monitorados (*data centric*). Deste modo, além de buscar a melhor confiabilidade dos

dados monitorados, a pesquisa de novas soluções visa também o menor consumo de energia, desde a concepção de aplicações para coleta dos dados [MAD05] e sistema operacional do sensor [HIL00] até o controle do tráfego na rede [HUL03], que possam prolongar sua vida útil. Assim, a otimização do uso da energia fica evidente como preocupação de base para o desenvolvimento de pesquisas em rede de sensores.

1.3. Proposta

As soluções que procuram organizar e prover maior eficiência de operação da rede, como algoritmos de roteamento e controle de acesso ao meio, ainda assim estão sujeitas aos efeitos do ambiente no meio de comunicação do sensor. Qualquer comunicação de rádio pode sofrer degradações decorrente de diversos fatores, principalmente ruído e desvanecimento do sinal [RAP96] [PRO95], o que eleva a taxa de erros na informação transmitida, podendo reduzir ou mesmo anular os ganhos de eficiência.

O simples controle de erros através da detecção e retransmissão de dados corrompidos, aumenta a confiabilidade da informação mas não reduz o consumo de energia. Ao contrário, cada retransmissão implica uma nova operação do transceptor, com consumo adicional de energia para enviar a mesma informação.

A adoção de controle de erros mais eficiente que, além da detecção, utilize uma forma de correção de erros, como a codificação convolucional, pode reduzir o número de retransmissões com a recuperação de dados corrompidos.

Por outro lado, o uso da codificação convolucional aumenta o consumo de energia, tanto pela necessidade de transmissão de dados redundantes quanto pelo esforço computacional necessário para a decodificação. Em situações de degradação do canal, porém, que tendem a aumentar a ocorrência de erros, a redução no consumo devida ao menor número de retransmissões, pela correção de erros, pode ultrapassar o consumo adicional gerado pela codificação convolucional.

Neste trabalho visamos uma forma de otimização do uso da energia do sensor, através da identificação de valores de compromisso entre consumo para transmissão e para decodificação em função das condições do canal.

1.4. Justificativa

O uso de codificação convolucional é largamente empregado em ambientes de comunicações sem fio, apresentando alguns aspectos que propiciam sua utilização, como a facilidade de implementação do codificador, principalmente através de software, e uma estrutura de decodificação bastante simples usando o algoritmo de Viterbi [VIT67].

A eficiência do código convolucional está relacionada à sua taxa de codificação, ou seja, o número de bits gerados para cada bit de informação e à sua memória. Um aspecto importante dos códigos convolucionais é a possibilidade de alteração da taxa do código sem alterar a estrutura do codificador, através da técnica de puncionamento [LIN83] [WIC95].

Outro aspecto interessante da estrutura destes códigos é a variação da eficiência de um código de determinada taxa com a ordem de memória do codificador, o que reflete sua complexidade. Isto abre a possibilidade de adoção de codificação de complexidade e eficiência variável de acordo com as condições do canal. Esta característica é explorada neste trabalho, com a identificação da ordem de memória do código que apresenta a melhor complexidade para determinada condição do canal, ou seja, a complexidade que provoca o menor consumo em conjunto com o consumo de transmissão.

1.5. Organização

Os tópicos envolvidos no desenvolvimento do trabalho, que conduzem à solução proposta, são apresentados em cinco capítulos.

No capítulo 1 são abordados os aspectos gerais de uma rede de sensores e a estrutura típica de um sensor, além do contexto geral do trabalho. No capítulo 2 é apresentado o modelo de canal utilizado e sua formulação, mostrando as características que afetam a comunicação entre os nós da rede. No capítulo 3 são apresentados os mecanismos utilizados para controle de erros e os aspectos que determinam a complexidade da codificação convolucional. Em seguida, no capítulo 4, são abordados os consumos de energia do sensor, tanto para transmissão e recepção, em função dos erros e retransmissões, quanto para decodificação, de acordo com a complexidade do código. É mostrado o valor de compromisso entre transmissão e codificação que minimiza o consumo e sua aplicação em redes de sensores. No capítulo 5 são relatadas as conclusões obtidas no trabalho e as possibilidades de novas pesquisas, em trabalhos futuros, que possam tomar este como base e ampliá-lo, propondo novas soluções e aplicações.

Capítulo 2

Modelagem do Canal de Rádio

2.1. Introdução

Em qualquer processo de comunicação, o sinal recebido sofre a influência do ruído Gaussiano branco aditivo (*AWGN*) e também da atenuação do sinal ao longo do canal de comunicação. Numa comunicação sem fio, através de ondas de rádio, ocorre ainda uma degradação causada pela interferência de multipercurso, decorrente de réplicas do sinal transmitido que se propagam por diversos caminhos, causando interferências construtivas e destrutivas no receptor e gerando flutuações rápidas na intensidade do sinal recebido. Este fenômeno, conhecido como desvanecimento (*fading*), é classicamente modelado de forma estatística seguindo a distribuição de Rayleigh [PRO95] [RAP96]. Estes efeitos do canal de rádio provocam erros nos quadros recebidos, degradando o desempenho da rede de sensores em termos de consumo de energia, uma vez que os quadros recebidos com erro podem ser retransmitidos.

A probabilidade de erro de bit no receptor depende essencialmente de três fatores. O primeiro é a potência média do sinal recebido, determinada pelo modelo de perda de percurso. O segundo é a perda interna no módulo receptor, devido principalmente ao nível de ruído térmico gerado pelo hardware do sensor. Este fator, em conjunto com a potência média do sinal recebido e com a taxa de transmissão, irá definir a relação sinal-ruído média no receptor. O terceiro fator é a condição momentânea do canal de rádio, definida como ganho do canal, durante a transmissão do quadro, que segue a distribuição de Rayleigh. A função densidade de probabilidade Rayleigh da amplitude do sinal r é dada por:

$$p(r) = \frac{r}{\sigma^2} \cdot \exp\left(-\frac{r^2}{2 \cdot \sigma^2}\right) \quad (0 \leq r \leq \infty) \quad (1)$$

onde σ representa o valor *rms* da tensão e σ^2 a potência média do sinal recebido, antes de sua detecção [RAP96], como mostra o gráfico na Figura 2.1.

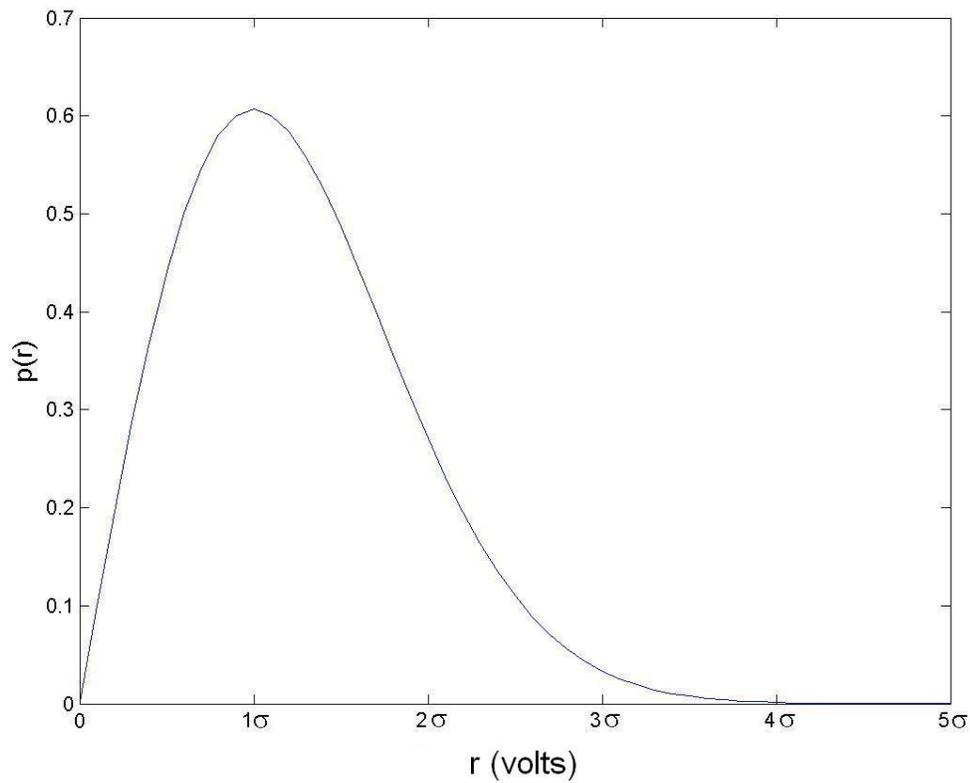


Figura 2.1: Tensão rms do envelope do sinal recebido

2.2. Perda de Percurso

Devido aos efeitos do canal, a potência média do sinal recebido é muito menor que a potência do sinal na saída do transmissor, variando em função da distância que separa transmissor e receptor e da frequência do sinal, podendo ser calculada por [RAP96]:

$$P_r(d) = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4 \cdot \pi)^2 \cdot d^2 \cdot L} \quad (2)$$

onde:

- $P_r(d)$: Potência média recebida em função da distância, em Watts.
 P_t : Potência transmitida, em Watts.
 G_t : Ganho da antena do transmissor.
 G_r : Ganho da antena do receptor.
 λ : Comprimento de onda do sinal, em metros.
 d : Distância entre transmissor e receptor, em metros.
 L : Perdas não relativas à propagação.

A equação (2), conhecida como *equação de Friis para espaço livre*, mostra que a potência do sinal recebido decai com o quadrado da distância da separação entre transmissor e receptor. Esta equação é válida para situações onde há uma linha de visada sem obstáculos entre eles, como no caso de enlaces de microondas e sistemas de comunicação via satélite. Para outras situações, no entanto, o modelo de perda de percurso, em dB, para distâncias arbitrárias é dado pela equação log-normal [RAP96]:

$$PL(d) \text{ dB} = PL(d_0) + n \cdot 10 \cdot \log_{10} \left(\frac{d}{d_0} \right) + \chi_\sigma \quad (3)$$

O parâmetro $PL(d_0)$ é a perda de percurso em uma distância de referência d_0 , a qual pode ser obtida analiticamente através da equação (2) ou empiricamente através de medições [ZUN04] e n é o expoente de perda relativo ao ambiente, que indica a taxa de perda de acordo com a distância, com valor situado normalmente entre 2 e 6 [RAP96]. O parâmetro χ_σ é uma variável aleatória com distribuição gaussiana, com média zero e desvio-padrão σ , que representa o efeito do somreamento do sinal, ou seja, variações no valor da perda de percurso devido às diferenças do ambiente para uma mesma distância entre transmissor e receptor.

A potência do sinal recebido em função da distância, d , entre transmissor e receptor, em dBm, é dada por:

$$P_{rx}(d) = P_{tx} - PL(d) \quad (4)$$

onde P_{rx} é a potência de transmissão, em dBm, e $PL(d)$ é a perda de percurso, em dB, como mostrado na Figura 2.2 [ZUN04]:

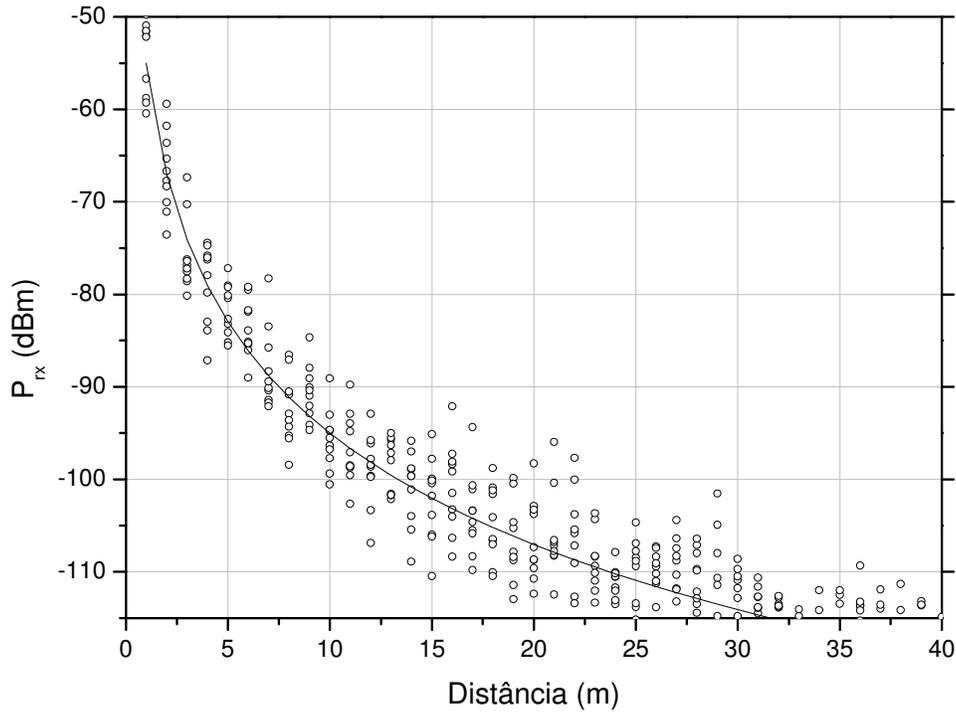


Figura 2.2: Potência média do sinal recebido, $n = 4$, $\sigma = 4$, $P_t = 0$ dBm.

2.3. Relação Sinal-Ruído

O ruído térmico, devido à agitação térmica de partículas no ambiente, é percebido na entrada do receptor e seu valor, em Watts, pode ser calculado por [PRO95] [RAP96]:

$$N = k \cdot T_0 \cdot B \quad (5)$$

onde:

$k = 1,380 \cdot 10^{-23}$: Constante de Boltzmann, em Joules/Kelvin.

T_0 : Temperatura ambiente, em Kelvin.

B : Banda equivalente de ruído do receptor, em Hz.

Além do ruído térmico, ocorrem outras perdas internas no receptor, devido ao ruído gerado pelos seus componentes, a qual é definida como *figura de ruído* F , que é uma característica intrínseca do receptor. Para que haja uma recepção satisfatória, a potência do sinal recebido deve suplantar o ruído térmico e as perdas internas ao receptor. A potência total do ruído, que é referenciado como *noise floor*, é definida como [RAP96]:

$$P_n = F \cdot k \cdot T_0 \cdot B \quad (6)$$

A razão entre a potência do sinal recebido, P_{rx} , e a potência total de ruído, P_n , é definida como relação sinal-ruído (*SNR – Signal-to-Noise Ratio*):

$$SNR = \frac{P_{rx}}{P_n} \quad (7)$$

Para modulações digitais binárias, como por exemplo BPSK e BFSK, a relação sinal-ruído geralmente é expressa em função da energia de um bit, E_b , e da densidade espectral de potência de ruído, N_0 [ZUN04] [PRO95]:

$$\frac{E_b}{N_0} = SNR \cdot \frac{B}{R_b} \quad (8)$$

Esta relação depende não somente da banda equivalente de ruído do receptor, B , em Hertz, como da taxa de transmissão digital, R_b , em bits por segundo.

2.4. Métricas de Desempenho

O sinal recebido sofre os efeitos do canal de comunicação e apresenta uma potência atenuada em relação ao sinal transmitido devido à perda de percurso, conforme equação (3), o que piora a relação sinal-ruído, conforme a equação (7). A probabilidade de erro teórica, em um canal que apresenta desvanecimento segundo o modelo de Rayleigh, depende da relação sinal-ruído e do tipo de modulação utilizado. Para modulação binária por fase (*BPSK – Binary Phase Shift Keying*) em canal Rayleigh, a qual foi adotado como modelo

de canal nas simulações de comunicação desenvolvidas neste trabalho, a probabilidade de erro de bit (*BER – Bit Error Rate*) é dada por [PRO95]:

$$BER = \frac{1}{2} \cdot \left(1 - \sqrt{\frac{\bar{\gamma}}{1 + \bar{\gamma}}} \right) \quad (9)$$

onde $\bar{\gamma}$ é a SNR média.

Como a ocorrência de pelo menos um bit com erro no quadro transmitido provoca a rejeição de todo o quadro, a taxa de erro de quadros (*FER – Frame Error Rate*) pode ser determinada analiticamente a partir da taxa de erro de bit:

$$FER = 1 - (1 - BER)^f \quad (10)$$

onde f é tamanho do quadro, em bits, que é transmitido no canal.

Normalmente os quadros transmitidos em redes de sensores são pequenos, da ordem de algumas dezenas de bytes, podendo no entanto ter seu tamanho programável [LEV07]. Quadros muito pequenos, no entanto, apesar de apresentarem menor probabilidade de ocorrência de erros para uma dada taxa de erro de bit, conforme equação (10), diminuem a eficiência da rede em termos de consumo de energia, devido tanto à maior sobrecarga das informações de controle quanto à energia consumida durante o tempo de estabilização do transmissor (*start-up*) antes da transmissão efetiva [SAN03].

2.5. Conclusão

Neste capítulo apresentamos o modelo de um canal de comunicação, que descreve os efeitos do ambiente sobre o sinal eletromagnético em uma comunicação via ondas de rádio, como a utilizada em redes de sensores sem fio.

A alteração do sinal recebido em função destes efeitos, principalmente o desvanecimento e o ruído, limita o alcance da comunicação em função da distância que separa o transmissor e o receptor, decorrente da perda de percurso. Além disto, aumenta a incidência de erros pela redução da relação sinal-ruído, percebido no aumento da taxa de erro de bit durante a recuperação da informação no receptor. O aumento da taxa de erro, por sua vez,

reduz a eficiência da rede em termos de energia, devido às repetidas operações do rádio para a retransmissão de quadros que apresentam informações corrompidas.

No capítulo seguinte são tratadas as estratégias que visam tornar os dados transmitidos mais robustos aos efeitos do canal e aumentar a sua confiabilidade na recepção, através da adoção de códigos corretores de erro.

Capítulo 3

Estratégias de Controle de Erros

3.1. Introdução

Uma das estratégias mais utilizadas para controle de erros é através do protocolo ARQ (*Automatic Repeat Request*), que existe há mais de quarenta anos, o qual implementa alguma codificação, geralmente com códigos do tipo CRC [WIC95], para detecção de dados corrompidos por erro e solicita sua retransmissão. Apesar de aumentar a confiabilidade dos dados recebidos, a adoção somente desta estratégia, também conhecida como ARQ *puro*, em redes de sensores não melhora a eficiência em termos de uso de energia, uma vez que podem ser necessárias diversas retransmissões para entregar a mesma informação, gerando um consumo adicional de energia do sensor.

Um aperfeiçoamento do controle de erros são os protocolos ARQ-híbridos, que além da detecção de erros incorporam também um código corretor de erros (*FEC – Forward Error Correction*), os quais podem se adaptar às condições mutáveis do canal, conhecidos como ARQ-híbrido Tipo II [WIC95]. O mecanismo utilizado neste controle de erros é aumentar a redundância do código corretor de erros, quando ocorre uma requisição de retransmissão devido a um erro detectado.

Com base neste conceito, a proposta deste trabalho é utilizar também um mecanismo que possa ser alterado face às condições do canal, utilizando, no entanto, codificação convolucional para correção de erros, a qual pode apresentar desempenhos distintos com a variação da ordem de memória do codificador.

3.2. Codificação Convolutacional

Os códigos convolucionais surgiram em 1955 como uma alternativa aos códigos de bloco, diferenciando-se na forma de codificação. Nestes, o quadro de informação precisa ser segmentado em blocos de tamanho k bits, aos quais são adicionados bits de redundância, formando um quadro codificado de tamanho n bits. Nos códigos convolucionais, o codificador apresenta uma memória e o quadro de informação é codificado num fluxo contínuo, sendo que a saída do codificador depende não apenas da informação atual na sua entrada como das informações anteriores, até a extensão da memória do codificador.

Os códigos convolucionais são representados pelos parâmetros $(n,k,m,dfree)$, onde n representa o número de bits na saída do codificador para k bits de entrada, m é a ordem de memória do codificador e $dfree$ é a distância livre do código, que vai definir a mínima distância de Hamming que pode ser obtida entre duas seqüências de bits codificados. Outro fator importante na representação do código é a sua taxa R , definida como:

$$R = \frac{k}{n} \quad (11)$$

A estrutura de um codificador convolutacional é baseada no arranjo entre registradores, que formam sua memória, e somadores binários que determinam a informação codificada na saída, em função da informação de entrada e do conteúdo da memória [LIN83]. Na Figura 3.1 temos um exemplo de um codificador com $(n,k,m,dfree) = (2,1,2,5)$.

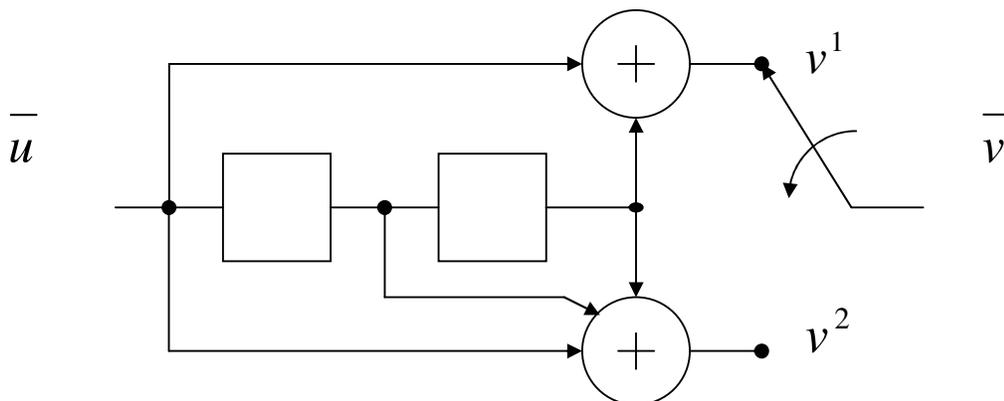


Figura 3.1: Codificador convolutacional (2,1,2,5)

Uma determinada seqüência de informação \bar{u} que entra no codificador bit a bit, produz v^n saídas de acordo com a taxa do código e do arranjo entre os registradores e os somadores binários, as quais são multiplexadas para formar a seqüência codificada \bar{v} . A operação de codificação pode ser representada por:

$$v^1 = u \oplus g^1 \quad (12)$$

$$v^2 = u \oplus g^2 \quad (13)$$

onde \oplus representa a operação de convolução discreta [WIG88] e g^n as seqüências geradoras, as quais podem ser obtidas pela resposta ao impulso do codificador, ou seja, inserindo uma seqüência com apenas o primeiro bit setado e observando as seqüências de saída, as quais são limitadas pela memória e apresentam um tamanho $m+1$. Este valor é conhecido como *constraint length* e representa o número máximo de bits numa seqüência de saída, que são afetados por um único bit de entrada [WIC95].

Um dos fatores determinantes do desempenho do código é sua distância livre, *dfree*. Códigos com a mesma taxa R , no entanto, podem apresentar valores distintos para distância livre, de modo que se procura adotar códigos que apresentam o valor máximo de *dfree* [WIC95] [LIN83].

A codificação convolucional pode ainda ser representada por um polinômio gerador e a operação de codificação pode ser obtida pela multiplicação de polinômios. Para o codificador da Figura 3.1, o polinômio gerador é dado por:

$$G(D) = [1 + D^2 \quad 1 + D + D^2] \quad (14)$$

onde o grau de cada termo do polinômio gerador representa o número de atrasos que o bit recebe nos registradores antes de ser somado.

O codificador convolucional sempre inicia com seus registradores zerados e retorna a este estado após a codificação, ou seja, após o último bit da seqüência de informação recebe ainda m deslocamentos. Para uma seqüência de entrada \bar{u} , com comprimento $k \cdot L$ bits, temos uma seqüência codificada \bar{v} na saída com tamanho $n \cdot (L + m)$ bits.

Uma vez que o codificador convolucional é um circuito seqüencial, sua operação pode ser representada como um diagrama de estados [LIN83]. Cada estado mostra o conteúdo dos registradores e a transição entre estados mostra a saída produzida em função da entrada. Para o codificador mostrado na Figura 3.1 são possíveis os seguintes estados e o correspondente diagrama de estados (Figura 3.2):

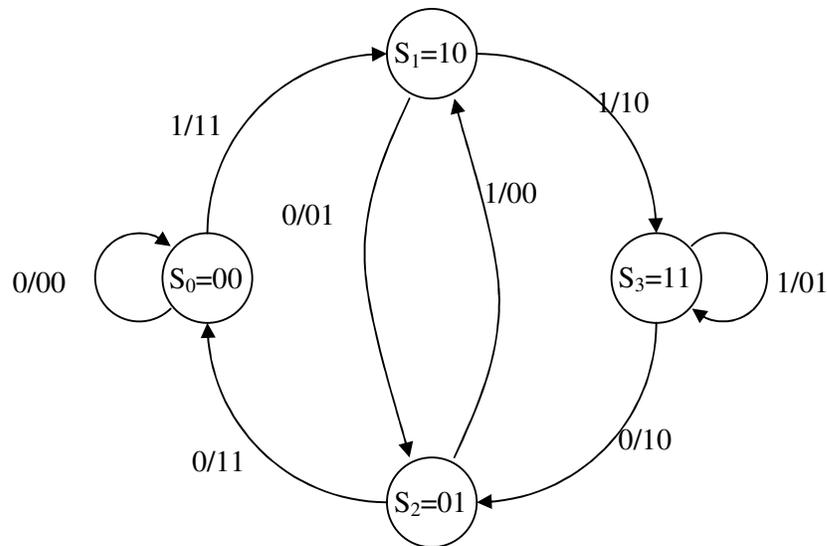


Figura 3.2: Diagrama de estados para codificador convolucional (2,1,2,5)

O diagrama de estados pode então ser visto como um *grafo ponderado orientado*, com os pesos atribuídos a cada ramo definidos pela saída do codificador, o que facilita a análise do código [WIC95].

3.3. Decodificação dos Códigos Convolucionais

Após o surgimento dos códigos convolucionais, foram propostos os primeiros algoritmos de decodificação, conhecidos como algoritmos seqüenciais. Entre estes está o Algoritmo de Fano, o qual aperfeiçoou as propostas anteriores e melhorou o desempenho da decodificação e o Algoritmo ZJ ou Stack. Em 1967 foi proposta por Viterbi [VIT67] uma nova abordagem de decodificação, a qual foi posteriormente apontada como sendo uma

solução para o problema de se encontrar o menor caminho em um grafo ponderado e um algoritmo de máxima verossimilhança para decodificação convolucional (MLD – Maximum Likelihood Decoder) [WIC95] [LIN83].

Os algoritmos seqüenciais se baseiam numa estrutura em árvore para realizar a decodificação, com os ramos sendo divididos a partir da raiz, de acordo com o valor esperado da saída do codificador em função do bit de entrada que o produziu. À medida que recebe a seqüência codificada, o algoritmo computa uma métrica entre a seqüência recebida e a esperada e decide qual ramo seguir, podendo retornar e tomar outro caminho se a métrica ultrapassar determinado valor limite estabelecido. Quando chega ao final da estrutura (folhas), onde não há mais divisão em ramos, que correspondem ao zeramento do codificador, a decodificação é concluída. Estes algoritmos se apresentam menos eficientes que o Algoritmo de Viterbi para baixos valores de comprimento de restrição (*constraint length*), além da dificuldade de gerenciamento de memória no caso do Stack. No entanto, se apresentam como uma boa opção para códigos com valor elevado de memória, uma vez que sua complexidade aumenta linearmente com a memória, enquanto o Algoritmo de Viterbi aumenta exponencialmente [WIC95].

O Algoritmo de Viterbi é o mais utilizado atualmente, principalmente em telecomunicações, havendo circuitos integrados dedicados à decodificação com este algoritmo e alguns transceptores de rádio que o incorporam. Dada uma seqüência de dados codificados, este algoritmo avalia cada palavra-código dentro da mesma e compara com a saída esperada do codificador, computando métricas que possibilitam identificar qual é a seqüência enviada mais provável. Para isto, o algoritmo se baseia no diagrama de estados do codificador, identificando as possíveis transições entre estados. Assim, é possível criar um novo diagrama que representa vários diagramas de estado, separados em unidades de tempo, conhecido como diagrama de treliça, como mostrado na Figura 3.3.

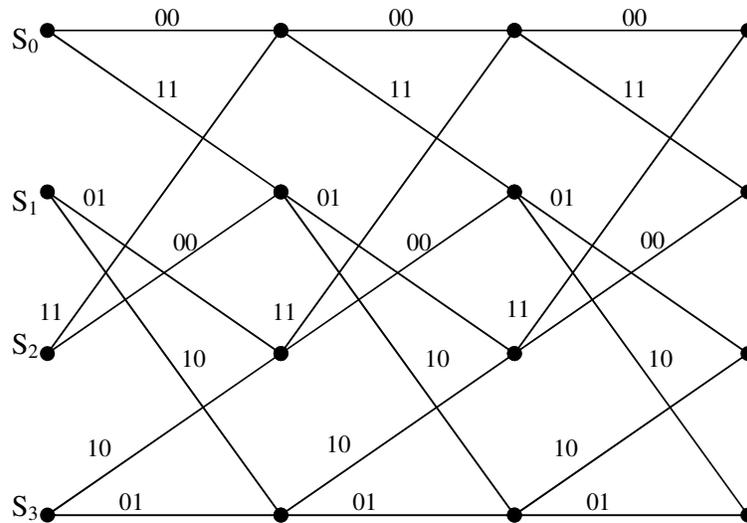


Figura 3.3: Diagrama de treliça para o código (2,1,2,5)

Neste diagrama são representadas réplicas do diagrama de estados do codificador e as possíveis transições entre os estados (S_i, S_{i+1}), nas quais é mostrado a saída do codificador para determinada transição. Como o codificador sempre inicia e termina no estado S_0 , o processo de decodificação nas primeiras m transições não apresentam todos os estados, assim como as m últimas, quando o decodificador retorna ao estado S_0 . As diversas transições que o decodificador realiza dentro da treliça formam um *caminho completo*, o qual é percorrido de forma reversa ao final da decodificação para reproduzir a informação enviada.

Para estabelecer este caminho na treliça, o algoritmo calcula métricas parciais, que podem ser a distância de Hamming entre os dados recebidos e a saída prevista do codificador, para todos os caminhos que entram em cada estado, e armazena aquele que apresenta a maior métrica (*survivor*) a partir da transição m , ou seja, quando entra na parte completa da treliça. Na transição seguinte novamente é calculada uma métrica para todos os caminhos que entram em cada estado e somadas às métricas armazenadas da transição anterior, sendo armazenado o caminho que apresenta a maior métrica acumulada e os demais eliminados. Este processo é repetido até o final da treliça, quando o decodificar percorre a treliça em sentido inverso (*traceback*) através dos caminhos armazenados, reproduzindo a seqüência de informação enviada, de forma invertida. Se dois caminhos apresentam a mesma métrica em determinado estado, o caminho sobrevivente é decidido de forma aleatória. O *Algoritmo de Viterbi* pode então ser formulado:

Passo 1: A partir da transição m calcule a métrica parcial para o caminho entrando em cada estado;

Armazene o caminho e sua métrica para cada estado.

Passo 2: Avance uma transição;

Calcule a métrica parcial para todos os caminhos entrando em cada estado e adicione à métrica armazenada do caminho sobrevivente da transição anterior;

Para cada estado, armazene o caminho com a maior métrica junto com sua métrica e elimine os demais.

Passo 3: Se não chegou ao final da treliça, retorne ao passo 2.

Devido a sua simplicidade este algoritmo apresenta facilidade de implementação, principalmente através de software. Como a treliça é uma repetição de módulos idênticos, que representam os estados do codificador, esta implementação pode ser realizada como uma coleção de estruturas similares, que calculam a métrica e selecionam o melhor caminho (ACS-Add-Compare-Select) [WIC95].

3.4. Complexidade de decodificação

Após um transiente inicial de m transições, o diagrama de treliça do decodificador apresenta uma estrutura que pode se repetir infinitamente, com a repetição de módulos idênticos que representam os estados do codificador e suas transições. Cada módulo da treliça apresenta 2^m estados iniciais e 2^m estados finais, sendo que cada estado inicial se conecta diretamente a 2^k estados finais. Desta forma, cada módulo da treliça apresenta 2^{k+m} conexões entre estados, de comprimento $n \cdot 2^{k+m}$ bits codificados. Como cada módulo da treliça representa a resposta do codificador à k bits de entrada, a complexidade da treliça pode ser definida, em símbolos por bit codificado, como [MEL96]:

$$C = \frac{n}{k} \cdot 2^{m+k} \quad (15)$$

Utilizando o *Algoritmo de Viterbi (MLD)* para decodificação, o esforço para calcular a métrica, selecionar e armazenar o caminho sobrevivente em cada módulo da treliça, é proporcional ao comprimento da conexão entre os estados, de forma que a complexidade apresentada na equação (15) reflete o esforço computacional do algoritmo por bit codificado. Este esforço, de acordo com a implementação do algoritmo, determinará o número de instruções necessárias para a unidade de processamento decodificar um quadro e conseqüentemente a quantidade de energia consumida pelo processador no sensor.

A complexidade cresce exponencialmente em função da memória do código, como pode ser visto na Figura 3.4, com os valores normalizados para oito símbolos, que é a menor complexidade obtida com $n=2$, $k=1$ e $m=1$, conforme a equação (15):

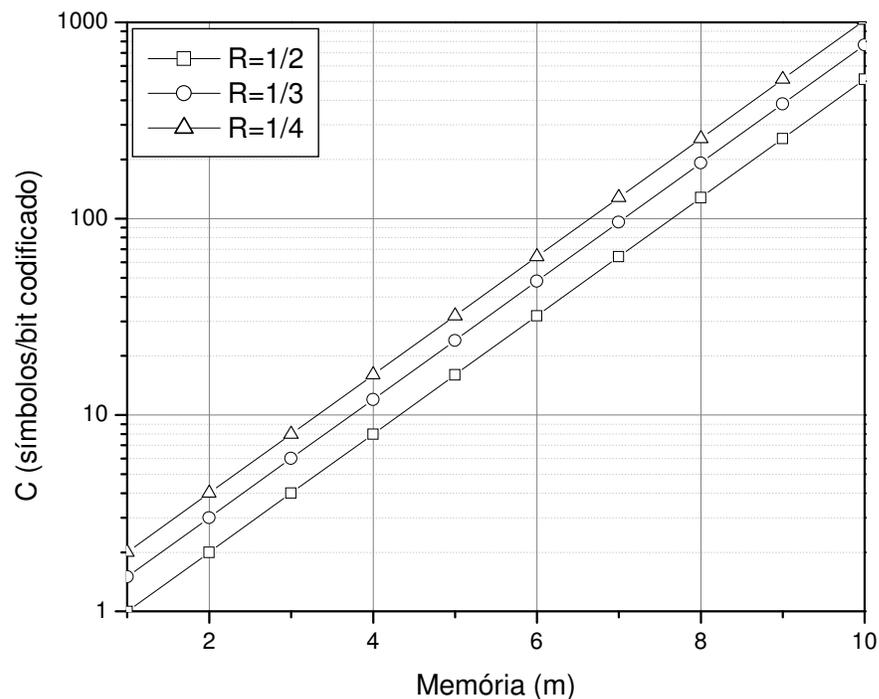


Figura 3.4: Complexidade de decodificação em função da memória do código

O gráfico mostra que a complexidade cresce inversamente com a taxa do código, o que também ocorre com sua capacidade de correção. Outras taxas diferentes de $R=1/n$ podem ser obtidas com outros arranjos do codificador, que realize a multiplexação dos bits tanto na sua entrada como na saída. No entanto, devido a maior facilidade, normalmente é

adotada a técnica de punção, com a qual a partir de um código com taxa $R=1/n$ podem ser obtidas outras taxas, como por exemplo $R=2/3$ [WIC95] [LIN83]. O uso desta técnica requer um esforço computacional adicional, tanto para se *puncionar* os bits na saída do codificador como para recomposição do quadro punçionado na decodificação.

O aumento do desempenho da codificação convolucional, com o aumento da ordem de memória, deve reduzir a taxa de erro de bit (BER) em um canal com erros. Deste modo é necessário estabelecer uma relação entre a complexidade e a taxa de erros, o que permite estimar tanto o número de retransmissões como o esforço de processamento para decodificação.

3.5. Simulação

Com o objetivo de avaliar o desempenho da codificação convolucional e seu impacto na correção de erros, foram realizadas diversas simulações de transmissão de dados codificados, com código de taxa $R=1/2$ e vários valores de memória, sobre um canal AWGN com desvanecimento Rayleigh, de acordo com o modelo de simulação mostrado na Figura 3.5:

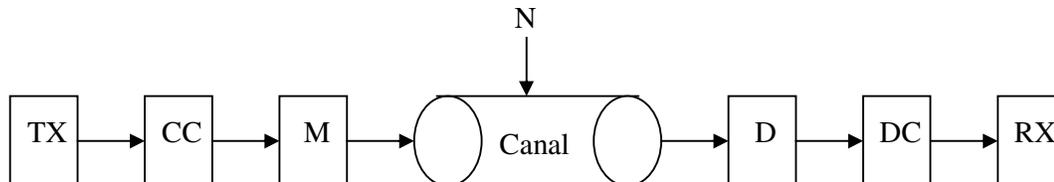


Figura 3.5: Modelo de simulação

onde:

TX:	Gerador aleatório de bits com distribuição uniforme
CC:	Codificador convolucional com diversas complexidades e taxa $R=1/2$
M:	Modulador BPSK
N:	Efeitos de degradação do canal (ruído gaussiano e desvanecimento)
D:	Demodulador
DC:	Decodificador convolucional
RX:	Contagem de erros

A simulação utilizou a modulação binária por fase (*BPSK*), de forma que a informação após a codificação foi mapeada na constelação do sinal analógico com os valores $\pm \sqrt{E_b}$, de acordo com o valor do bit [RAP96]. Os efeitos de degradação inseridos no canal, seguiram a seguinte equação [PRO95]:

$$r(t) = \alpha \cdot s(t) + n(t) \quad (16)$$

onde:

$r(t)$: sinal recebido

$s(t)$ sinal enviado

$n(t)$: efeitos do ruído aditivo com distribuição gaussiana

α : efeitos do desvanecimento do canal, o qual segue a distribuição de Rayleigh, conforme a Figura 2.1.

O tempo de duração média do desvanecimento (*fading*), em segundos, foi calculado de acordo com a equação [RAP96]:

$$\bar{\tau} = \frac{e^{\rho^2} - 1}{\rho \cdot fm \cdot \sqrt{2 \cdot \pi}} \quad (17)$$

onde:

$\bar{\tau}$: Duração média do desvanecimento

ρ : Valor normalizado do nível do sinal R em relação ao valor rms do envelope

fm : Desvio de frequência doppler

Para esta simulação foram adotados os valores $\rho = 0,01$ e $fm = 8$ Hz.

O simulador foi desenvolvido em linguagem C++, com o qual foram executadas diversas simulações, com variação tanto da potência do sinal transmitido como da memória do código convolucional. Para a codificação foram utilizadas seqüências geradoras de código que apresentam o máximo valor de $dfree$, conforme a tabela 3.1 [LIN83].

m	g^1	g^2	$dfree$
1	110	100	3
2	111	101	5
3	1111	1011	6
4	11001	10111	7
5	111101	101101	8
6	1100111	1011101	10
7	11001011	10111101	10

Tabela 3.1 : Sequências geradoras de código convolucional

Para verificar a validade da simulação dos efeitos do canal foi calculado o desempenho teórico de um canal AWGN com desvanecimento Rayleigh, de acordo com a equação (9), e realizadas simulações de diversas transmissões com dados sem codificação, como mostrado na Figura 3.6.

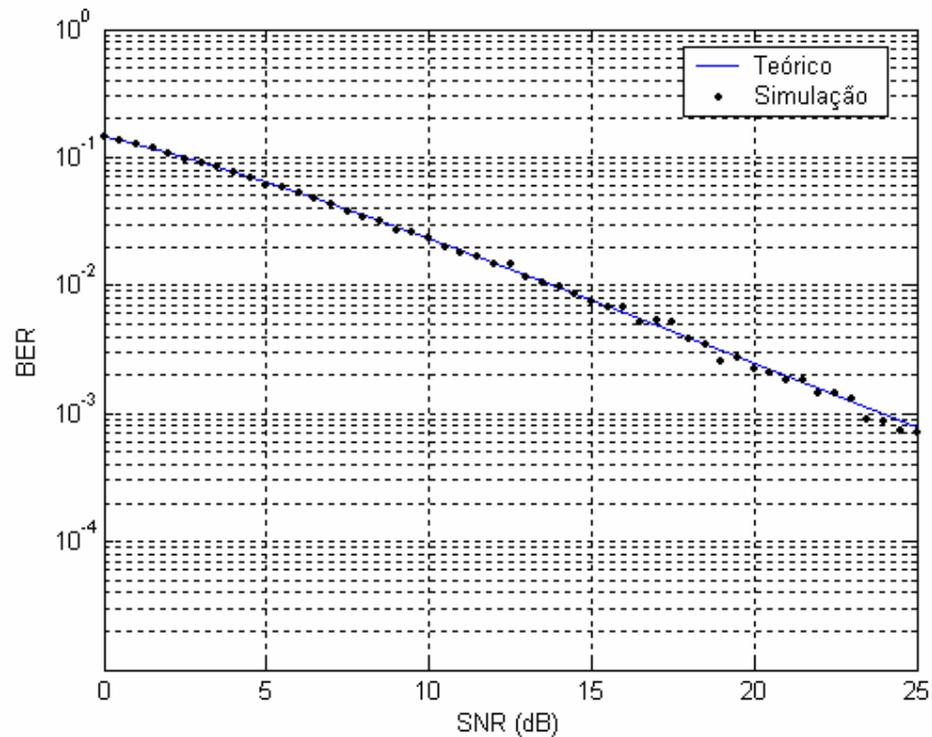


Figura 3.6: Desempenho teórico e simulado no canal Rayleigh

Na simulação de desempenho da codificação convolucional, foram realizadas simulações de transmissões com dados codificados e computados os valores, tanto de erro de bit como de erro de quadro, como mostrado nas Figura 3.7 e Figura 3.8, respectivamente, adotando um pacote de 288 bits. Como esperado, à medida que se aumenta os valores de potência e de memória do código, mais confiável torna-se a comunicação, com a redução da taxa de erros. Numa rede de sensores, no entanto, onde o consumo de energia é uma severa restrição, o incremento de qualquer destes parâmetros implica em maior consumo, tanto pelo módulo de rádio para transmissão, com maior potência, como pelo processador por maior esforço para decodificação.

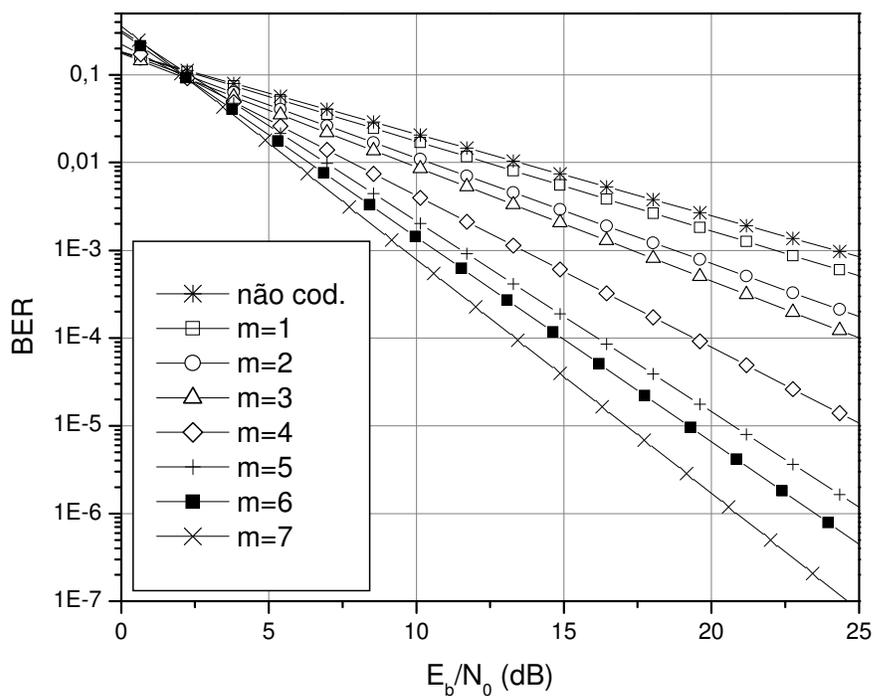


Figura 3.7: Taxa de erro de bit para dados codificados em canal Rayleigh

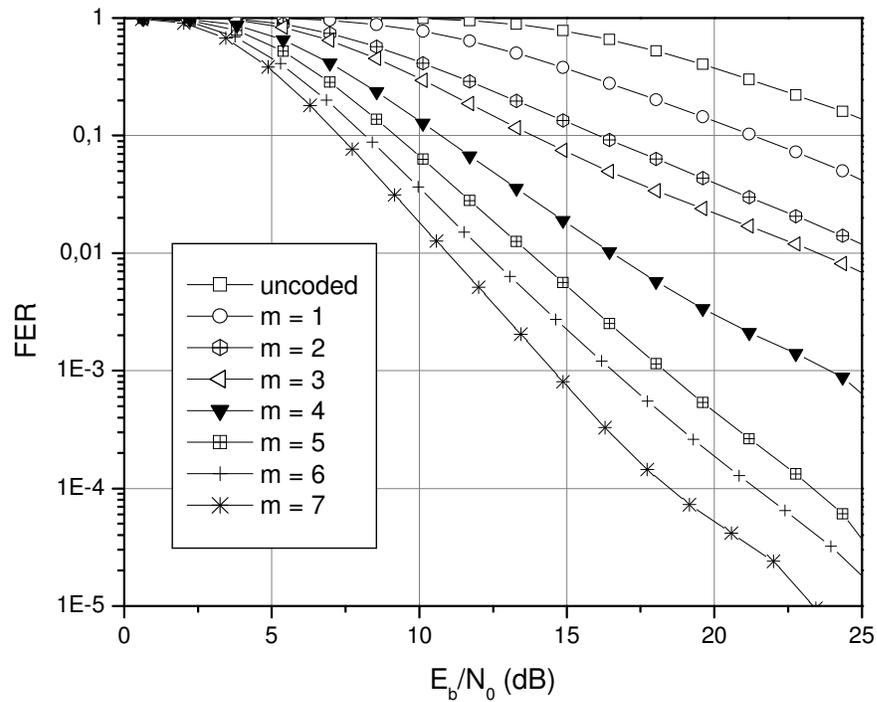


Figura 3.8: Taxa de erro de quadro para dados codificados em canal Rayleigh

3.6. Conclusão

Neste capítulo foram apresentados os aspectos de controle de erro utilizando a codificação convolucional. Foi mostrado a estrutura do codificador e do decodificador convolucional, utilizando o decodificador de máxima verossimilhança, o qual é adotado há bastante tempo e largamente utilizado em controle de erro em comunicações.

Em função do tamanho da memória do código foi mostrado que aumenta exponencialmente a complexidade do decodificador, que se reflete em maior esforço computacional para manipular os símbolos gerados. Esta possibilidade, entretanto, de aumentar a memória do código e conseqüentemente seu desempenho, se mostra adequada para implementação de uma estratégia de controle de erros como o ARQ-híbrido tipo II, com alteração da eficiência do código em função da situação do canal.

Os resultados da simulação de transmissão, em um canal com desvanecimento Rayleigh, mostram que a confiabilidade da comunicação melhora proporcionalmente à memória do código e à potência de transmissão. A alteração destes parâmetros, porém, altera o perfil de consumo de energia numa rede de sensores, havendo a necessidade de um valor de equilíbrio entre ambos, o que será tratado no próximo capítulo.

Capítulo 4

Otimização do Consumo de Energia

4.1. Introdução

A energia em Redes de Sensores Sem Fio é seguramente o fator que impõe as maiores restrições à implementação de novas soluções e novos usos dos sensores. A alimentação dos sensores normalmente é realizada através de baterias, que em função inclusive de suas reduzidas dimensões apresentam capacidade limitada de carga, restringindo a vida útil do sensor, aliado ao fato que em geral os sensores são instalados ou dispostos em locais onde não é possível ou viável a substituição de baterias.

As redes de sensores trabalham de forma colaborativa e orientadas aos dados monitorados (*data centric*), com um determinado nó, que eventualmente pode estar conectado a algum tipo de infra-estrutura, operando como ponto coletor de dados ou sumidouro (*sink*), para o qual são direcionadas todas as mensagens. Como a rede é composta de inúmeros sensores espalhados em uma região, as mensagens com os dados coletados trafegam até o coletor através de inúmeros saltos entre os diversos nós, que encaminham as mensagens recebidas. Desta forma, o esgotamento de energia em um nó pode eventualmente isolar uma parte significativa da rede, devido à perda de encaminhamento de mensagens.

Os fabricantes de dispositivos micro-eletrônicos, que são adotados em sensores, principalmente processador e transceptor, além de disponibilizarem produtos com o menor consumo possível geralmente, os provêm com algum mecanismo de controle de energia, sendo o mais comum a possibilidade de deixar o dispositivo em estado inativo (*idle state*) ou modo dormente (*sleep mode*), nos quais o consumo de energia é significativamente reduzido, pelo desligamento de alguns módulos internos, como por exemplo osciladores. Isto acarreta,

porém, uma outra preocupação em termos de consumo: quando retorna ao modo de operação normal, o dispositivo demanda algum tempo para estabilização da operação dos módulos desligados, com energia sendo consumida sem que o dispositivo esteja efetivamente operando. Alguns estudos sobre sensores focam neste problema, como a otimização do tamanho do quadro a ser transmitido, reduzindo o impacto de consumo durante o tempo de retorno à operação normal (*start-up*) [SAN03].

Outras propostas procuram tirar vantagem desta característica dos dispositivos, principalmente aquelas que tratam do controle de acesso ao meio [POL04] [RAJ03], que buscam deixar o sensor em modo de operação normal o menor tempo possível, ou que tratam de roteamento [HEI00] e que procuram tornar o encaminhamento de mensagens o mais eficiente possível e reduzir o número de operações indevidas do sensor.

Os dispositivos que formam o sensor, no entanto, apresentam uma assimetria no consumo de energia, principalmente o transceptor e o processador. O consumo do processador está diretamente relacionado à sua frequência de operação, sendo que para os modelos utilizados em sensores este valor não é muito elevado, normalmente não ultrapassando 20 MHz, o que por outro lado limita o desempenho de processamento, o que também não causa grande impacto visto que o processamento não é intensivo no sensor. O consumo do rádio, por outro lado, não apresenta muitas possibilidades de redução devido ao consumo do amplificador que propaga o sinal modulado. Como a potência do sinal determina o alcance do mesmo, conforme a equação (3), a sua redução pode inviabilizar a comunicação. As plataformas de rádio utilizadas em sensores normalmente apresentam um limite inferior de potência de transmissão e conseqüentemente de consumo de energia, o que em geral é maior que o consumo do processador, em termos de energia por bit.

Com esta assimetria, uma estratégia na preservação de energia é procurar deslocar a carga de operação do transceptor para o processador quando se tornar adequado. Neste trabalho, esta estratégia é proposta através codificação dos dados com códigos de correção de erros com complexidade variável, que objetiva carregar o processador com o esforço de processamento para decodificação e reduzir as operações do rádio para eventuais retransmissões de dados com erro, quando as condições do canal se tornam adversas.

4.2. Modelagem do Consumo de Energia Para Processamento

De acordo com a equação (15) a complexidade do código determinará o esforço necessário de processamento e conseqüentemente o número de instruções necessárias para o processador decodificar um quadro, refletindo na quantidade de energia consumida.

O *Algoritmo de Viterbi*, pela sua extensiva utilização desde que foi proposto e pela sua simplicidade, pode ser implementado em hardware com componentes discretos ou em dispositivo dedicado, ou através de software, como já desenvolvido em algumas plataformas de microprocessadores [HEN02]. De acordo com a avaliação de desempenho desta implementação, para microprocessadores adotados em alguns sensores [ATM07] [CRO07], é possível estimar o esforço de três ciclos de instrução para decodificar um símbolo. Deste modo, o consumo de energia para decodificar um bit, em joules/bit, pode ser definido como:

$$E_{dec} = \frac{3 \cdot C \cdot P_{proc}}{I} \quad (18)$$

onde:

- C : Complexidade do código, conforme equação (15)
- P_{proc} : Potência dissipada pelo processador em estado ativo, em Watts
- I : Capacidade de processamento, em instruções/segundo/Hertz

O consumo para decodificação de um quadro recebido dependerá do tamanho do quadro, o que é equacionado, em joules/quadro, como:

$$E_{dq} = E_{dec} \cdot f \cdot R \quad (19)$$

onde:

- f : tamanho do quadro, em bits
- R : taxa do código

Deste modo, o consumo de energia para decodificação de um código convolucional de determinada complexidade depende do desempenho do processador, em termos de potência dissipada e capacidade de processamento. Os processadores normalmente utilizados em sensores apresentam baixo consumo mas também pequena capacidade de processamento, atingindo geralmente a taxa de 1MIPS/MHz [ATM07].

4.3. Modelagem do Consumo de Energia Para Transmissão

A informação coletada no sensor, que será transmitida pela rede, será modulada e amplificada pelo módulo transmissor de rádio, antes de ser propagada no meio. No lado do receptor, um sinal percebido é por sua vez amplificado e demodulado. Este estágio do sensor é o que apresenta o maior consumo de energia, devido à necessidade de amplificação do sinal modulado a ser transmitido, sendo comum a operação do transmissor somente na presença de dados a serem transmitidos, permanecendo inativo o maior período. Por outro lado, o módulo receptor precisa ficar ativo para escutar o meio e receber informações disponibilizadas no meio. A energia consumida na comunicação entre dois nós da rede, em joules/bit, é dada por:

$$E_{trans} = \frac{P_{tx} + P_{rx}}{R_b} \quad (20)$$

onde:

P_{tx} : Potência dissipada pelo transmissor em estado ativo, em watts

P_{rx} : Potência dissipada pelo receptor em estado ativo, em watts

R_b : Taxa de transmissão, em bits/segundo

A potência dissipada para recepção normalmente apresenta um valor fixo, a qual está relacionada à sensibilidade do receptor. O transmissor, por outro lado, apresenta a possibilidade de programação da potência de transmissão dentro de uma faixa [CHI07], o que permite um maior controle sobre a energia consumida na transmissão.

4.4. Consumo Total de Comunicação

A utilização de protocolos ARQ causa a retransmissão de um quadro quando detectado algum erro. Em uma rede de sensores, contudo, a retransmissão automática simplesmente, ou ARQ puro, pode gerar um elevado consumo de energia quando o canal apresenta degradação severa. A adoção de protocolos ARQ-híbridos por outro lado, além do mecanismo de retransmissão automática procura reforçar os dados a serem transmitidos contra os efeitos do canal com o uso de um código corretor de erros (*FEC – Forward Error Correction*), ocorrendo retransmissão somente quando a parte *FEC* do protocolo não tem sucesso na correção de erros introduzidos no canal. A ocorrência de retransmissões diminui a vazão da rede (*throughput*), uma vez que a mesma informação é re-enviada num período que novas informações poderiam ser transmitidas. Esta diminuição na vazão pode ser definida como o número esperado de transmissões necessárias até que o quadro seja aceito pelo receptor, o que é dado por [WIC95]:

$$E[Tr] = \frac{1}{1 - FER} \quad (21)$$

onde *FER* é a probabilidade de erro de quadro (*Frame Error Rate*).

É possível assim estimar o número de transmissões do quadro até sua aceitação no receptor em função da potência do sinal recebido, para dados sem codificação e dados codificados com diversas complexidades do código convolucional, de acordo com os resultados da simulação apresentados na seção 3.5, conforme mostrado na Figura 4.1.

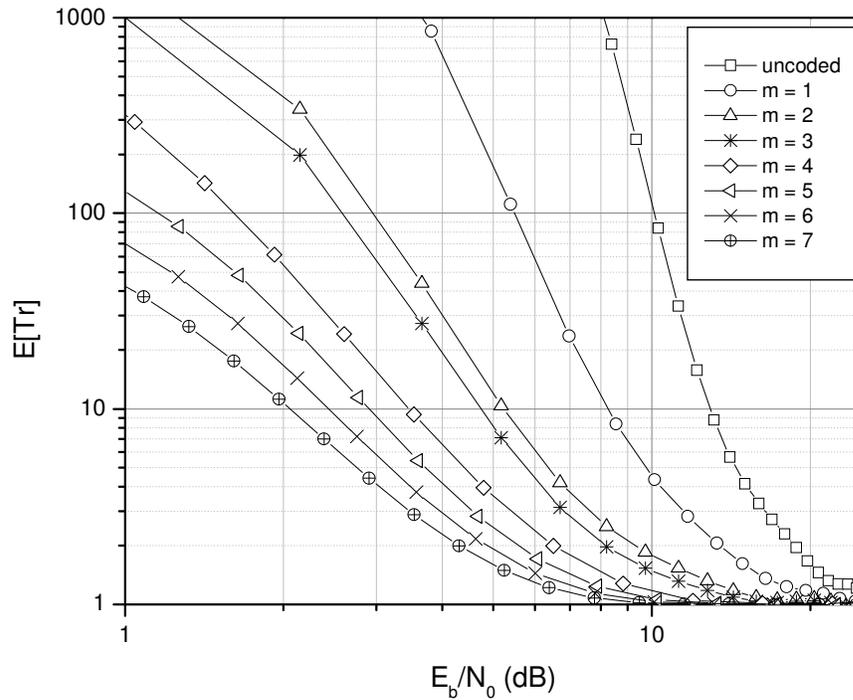


Figura 4.1: Número esperado de transmissões do quadro no canal Rayleigh

Cada retransmissão do quadro requer energia adicional dos sensores envolvidos na comunicação, de forma que a energia total média consumida para a transmissão do quadro, E_T , em joules, será determinada pelo número esperado de transmissões, conforme a equação (21), e pela energia necessária para transmitir e receber um quadro, o que é pode ser equacionado como:

$$E_T = E[Tr] \cdot E_{trans} \cdot f \quad (22)$$

onde f é o tamanho do quadro enviado.

Com base nos dados de dispositivos utilizados em um plataforma comercial de sensor [CHI07] [ATM07] [CRO07], foi calculado o consumo total de energia envolvido na comunicação, ou seja, a energia consumida para transmissão e para decodificação de um quadro, em canal com desvanecimento Rayleigh, adotando como parâmetros uma potência de transmissão $P_t = 0$ dBm e taxa de transmissão $R_b = 19200$ bps, para diferentes valores de

relação sinal-ruído e diversas complexidades de código, com os valores resultantes normalizados para o valor de consumo de um quadro, conforme mostrado na Figura 4.2.

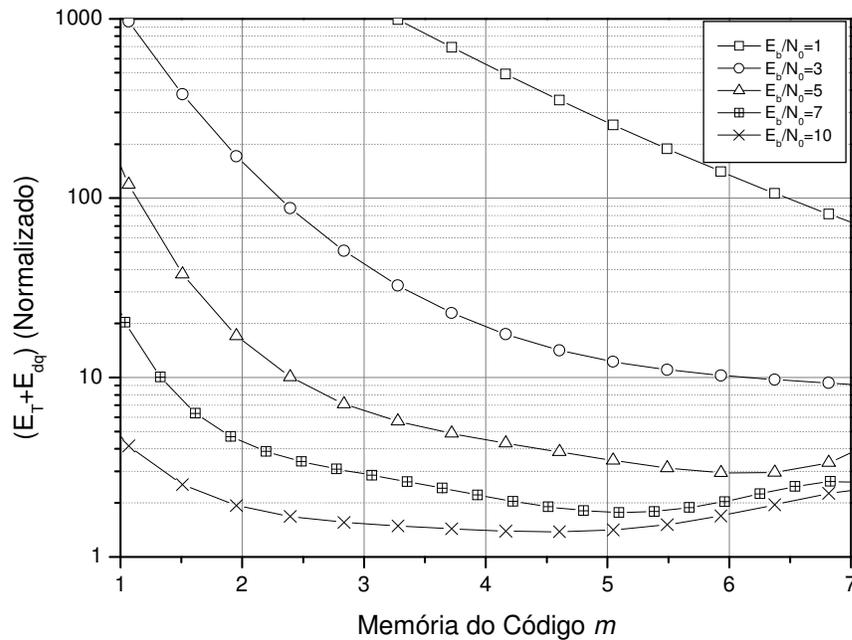


Figura 4.2: Consumo médio de energia de comunicação para o canal Rayleigh

A medida que aumenta o valor da relação sinal ruído, ou seja, a degradação do canal é menos severa ou a potência de transmissão é elevada, o número esperado de transmissões converge para a unidade, para os diversos valores da ordem de memória do código, conforme visto na Figura 4.1. Assim há uma limitação inferior no consumo de energia de transmissão. Por outro lado, a energia consumida na decodificação aumenta exponencialmente com a ordem de memória do código, conforme equações (15), (18), (19) e Figura 3.4. Deste modo, para valores elevados de relação sinal-ruído, o aumento da complexidade do código provocará um aumento no consumo de energia, que é o oposto do que se busca, conforme visto na Figura 4.2, a qual mostra pontos de empenamento nas curvas, onde ocorre o valor ótimo de complexidade e o consumo de energia é minimizado. Para valores baixos de relação sinal-ruído, entretanto, o valor ótimo da ordem de memória ocorre para ordens de memória elevadas, conforme a tendência das curvas superiores na Figura 4.2.

Isto mostra que a adoção uniforme da maior complexidade de código possível em todos os nós da rede não é adequada em termos de consumo de energia, sendo necessário a escolha otimizada da complexidade em cada nó, de acordo com a situação do canal enfrentada. Desta forma, nesta estratégia de controle de erros adaptativa às condições do canal, a escolha da melhor complexidade em cada nó determinará o seu perfil de consumo de energia e conseqüentemente a vida útil da rede.

4.5. Resultados

O consumo total de energia na comunicação entre dois nós reflete a transmissão de dados num único salto (*single hop*). O encaminhamento de mensagens numa rede, entretanto, normalmente ocorre em múltiplos saltos (*multiple hop*), gerando diversas rotas ao longo das quais os nós podem apresentar diversos perfis de consumo, conforme uma região de uma determinada rede mostrado na Figura 4.3.

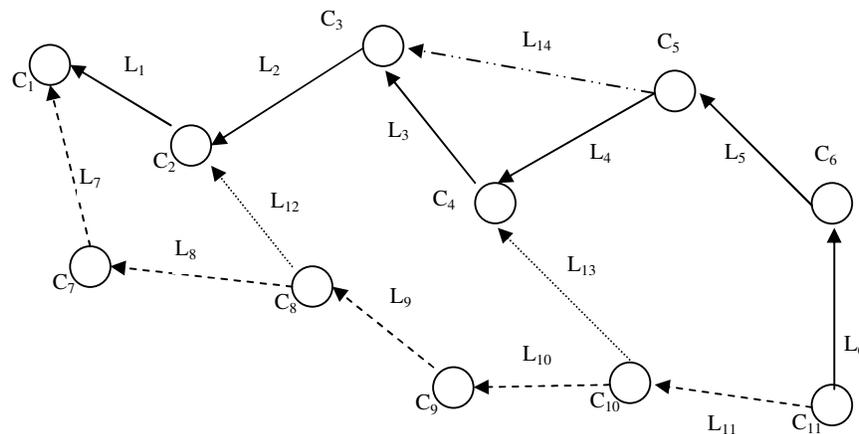


Figura 4.3: Encaminhamento de mensagem em múltiplos saltos

Na escolha da complexidade deve ser avaliada a situação de cada nó na rota para adequar o código que minimize o consumo esperado para encaminhamento da mensagem desde a fonte dos dados até o receptor. Nesta avaliação de consumo eventualmente pode ser vantajosa a eliminação de um salto intermediário. Adotando como rota estabelecida aquela ilustrada em linha cheia, entre os nós C1 e C11, da Figura 4.3, com a eliminação do salto intermediário através de $L_3 - L_4$, a rota se altera com a nova ligação L_{14} . O valor da relação

sinal-ruído em cada nó receptor pode ser determinada, conhecendo-se a distância entre transmissor e receptor, conforme as equações (3), (4) e (8). Adotando $n=4$, a potência de ruído de -115 dBm e as seguintes distâncias entre os nós, podemos obter os respectivos valores de E_b/N_0 no receptor, mostrados na Tabela 4.1.

	L_3	L_4	L_{14}
$d(m)$	15	17	18
E_b/N_0	≈ 10	≈ 8	≈ 5

Tabela 4.1: Valores de sinal-ruído em função da distância

Neste exemplo de rede, supondo que os nós C_4 e C_5 utilizem codificação com memória $m=1$, através de L_3 e L_4 , podemos obter o valor de consumo aproximado, normalizado para um quadro, de acordo com a Figura 4.2, conforme a equação:

$$(E_T + E_{dq})_{m=1}^{C_4} + (E_T + E_{dq})_{m=1}^{C_5} \approx 24,5 \quad (23)$$

Se configurarmos o nó C_5 para utilizar codificação com memória $m=2$, podemos obter, de acordo com a Figura 4.2 e a Tabela 4.1, o valor de consumo para transmissão entre os nós C_5 e C_3 , através da conexão L_{14} , normalizado para um quadro, de acordo com a equação:

$$(E_T + E_{dq})_{m=2}^{C_5} \approx 19 \quad (24)$$

Portanto, a solução de aumentar a memória do código do nó C_5 para $m=2$ produzirá uma redução na energia esperada de comunicação, uma vez que o somatório da energia utilizada nos canais L_3 e L_4 , para $m=1$, é maior que a energia utilizada no canal L_{14} , para $m=2$, conforme a Figura 4.2. Na avaliação desta situação, deve ser satisfeita a seguinte equação:

$$\frac{(E_T + E_{dq})_{m=1}}{(E_T + E_{dq})_{m=2}} \geq 1 \quad (24)$$

De modo similar, pode ser avaliada a rota que apresenta o menor consumo, ou seja, a menor somatória dos consumos dos canais, com eventuais alterações de memória do código. Por outro lado, a busca de melhor confiabilidade com a adoção da maior codificação em todos os nós se mostra prejudicial, pelo aspecto de consumo excessivo de energia.

Na rede mostrada na Figura 4.4 está exemplificada uma rota com onze saltos, L_1 até L_{11} , entre nós distribuídos de aleatoriamente em uma área. Os valores das distâncias entre os nós e da relação sinal-ruído para cada canal, de acordo com as equações (3), (4), (7) e (8), para uma potência de transmissão em cada nó, $P_t = 0$ dBm, são apresentados na Tabela 4.2.

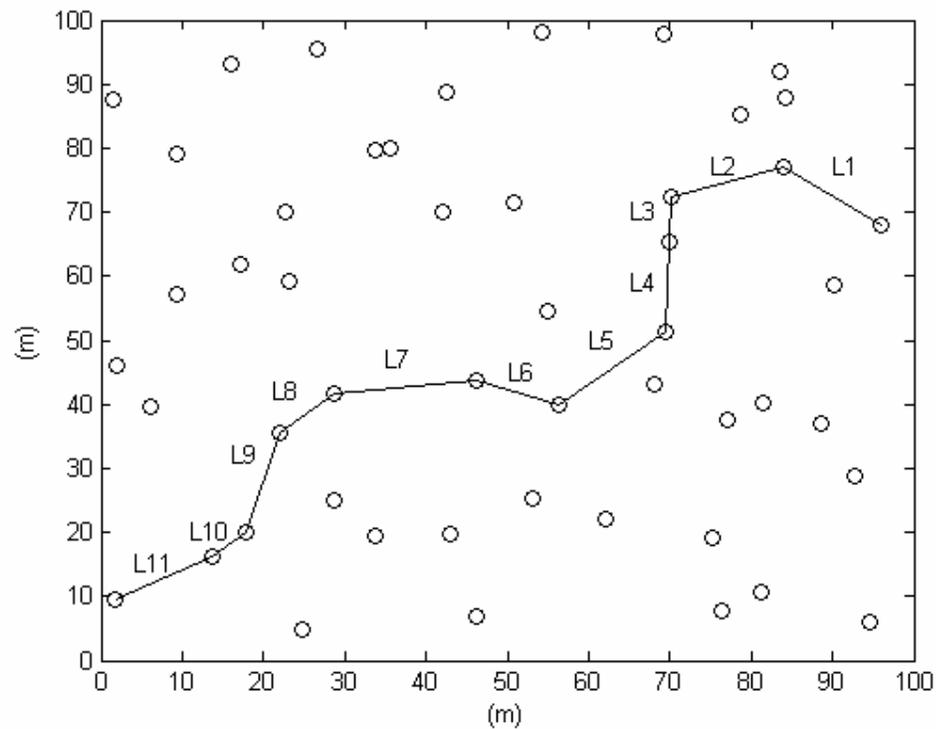


Figura 4.4: Exemplo de rede com rota com múltiplos saltos

<i>Canal</i>	<i>d (m)</i>	$\frac{E_b}{N_0}$
L_1	15,13	≈ 10
L_2	14,56	≈ 10
L_3	6,90	≈ 30
L_4	13,86	≈ 10
L_5	17,60	≈ 5
L_6	10,77	≈ 18
L_7	17,64	≈ 5
L_8	9,12	≈ 23
L_9	15,95	≈ 7
L_{10}	5,70	≈ 36
L_{11}	13,84	≈ 10

Tabela 4.2: Valores de sinal-ruído em uma rota com múltiplos saltos

Percebe-se que os canais L_5 e L_7 apresentam a pior condição do canal, em função da distância. Com todos os sensores operando com a maior complexidade, neste caso $m = 7$, a comunicação será muito confiável, mas a vida útil será abreviada, em função do maior consumo de energia, conforme a equação (25) e a Figura 4.2.

$$\sum_{i=1}^{11} (E_T + E_{dq})_{m=7}^{L_i} \approx 28 \quad (25)$$

No outro extremo, se não for adotada nenhuma codificação, a comunicação através desta rota não é viável, uma vez que, de acordo a Figura 4.1, não é possível a comunicação através dos canais L_5 e L_7 , conforme mostrado na Figura 4.5.

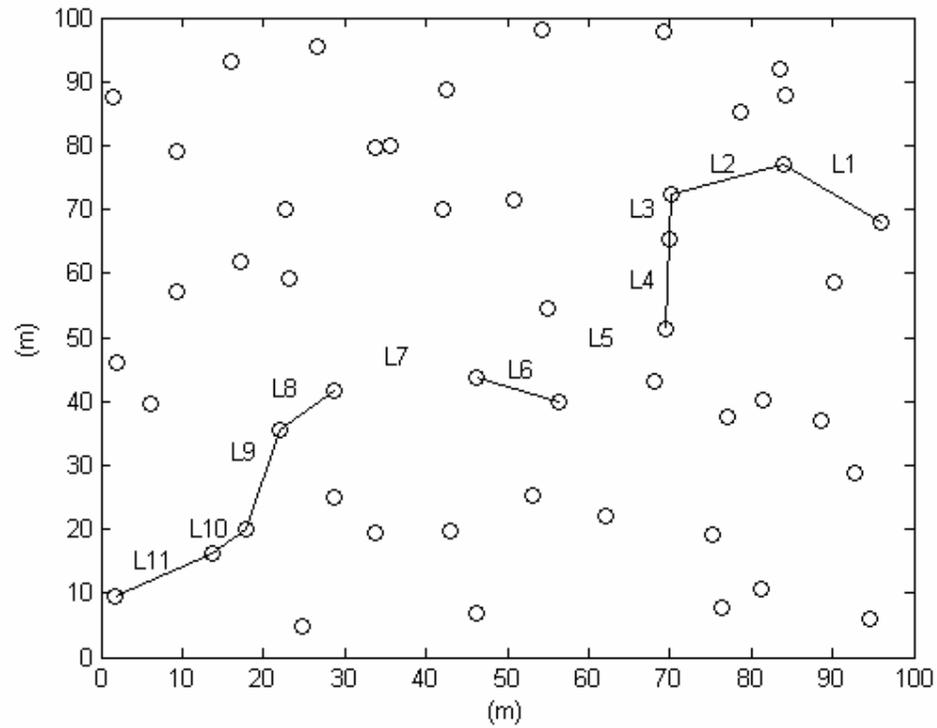


Figura 4.5: Exemplo de rede com rota interrompida

Para se adequar a rota ao melhor perfil de consumo de energia, devem ser ajustados os sensores envolvidos em cada salto para a melhor complexidade, como mostra a Tabela 4.3, minimizando o consumo total na rota.

<i>Canal</i>	<i>d (m)</i>	$\frac{E_b}{N_0}$	<i>m ajustado</i>
L_1	15,13	≈ 10	4
L_2	14,56	≈ 10	4
L_3	6,90	≈ 30	sem codificação
L_4	13,86	≈ 10	4
L_5	17,60	≈ 5	6
L_6	10,77	≈ 18	1
L_7	17,64	≈ 5	6
L_8	9,12	≈ 23	sem codificação
L_9	15,95	≈ 7	5
L_{10}	5,70	≈ 36	sem codificação
L_{11}	13,84	≈ 10	4

Tabela 4.3: Valores de memória de código ajustados aos canais

Com este ajuste a vida útil da rota é aumentada, uma vez que, com a escolha do melhor perfil em cada salto, o consumo total envolvido na rota é reduzido para

$$\sum_{i=1}^{11} (E_T + E_{dq})_{m=ajustado}^{L_i} \approx 16 \quad (26)$$

4.6. Conclusão

Em virtude dos diferentes perfis de consumo de energia dos dispositivos que compõem um sensor, especialmente o transceptor e o processador, a simples retransmissão de quadros com erro aumenta o consumo de energia da rede.

Por outro lado, como foi demonstrado neste capítulo, a adoção de uma estratégia de controle de erro através de protocolo ARQ-híbrido, que possibilite adequar a complexidade do código de correção de erro em cada nó, em função do perfil de sua conexão com os outros

nós, pode reduzir o número de retransmissões e também o consumo de energia, apresentando um valor ótimo de complexidade para cada situação de canal.

De modo similar, o ajuste de rota dentro de uma rede, em função da complexidade do código pode também reduzir o consumo de energia, tanto pela eliminação de saltos múltiplos no encaminhamento de mensagens quanto pela redução do consumo total na rota.

Desta forma, o ajuste de operação da rede, com adoção de codificação convolucional para correção de erros, e com foco na escolha da complexidade adequada, torna a rede mais eficiente em relação ao consumo de energia e prolonga sua vida útil.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1. Conclusões

Este trabalho apresentou uma proposta de otimização de consumo de energia em Redes de Sensores Sem Fio, baseado em controle e correção de erros através de protocolo ARQ-híbrido, com utilização de códigos convolucionais na sua parte FEC.

As Redes de Sensores sem Fio utilizam transceptores de rádio para estabelecer os canais de comunicação entre os nós. Estes canais, contudo, podem sofrer diversas influências do ambiente, principalmente ruídos e interferências de cópias do sinal transmitido, que se propagam por diversos caminhos, causando flutuações na potência do sinal recebido e provocando erros na informação.

Para manter a confiabilidade das informações, na ocorrência de erros ocorrem retransmissões dos dados corrompidos, o que demanda consumo adicional de energia do sensor.

Normalmente, os sensores são constituídos de dispositivos micro-eletrônicos de baixo consumo e são alimentados com baterias de pequena capacidade. A vida útil do sensor é limitada pela carga da bateria, uma vez que a rede de sensores pode ser instalada em locais inóspitos ou de difícil acesso, onde a manutenção e reposição de baterias esgotadas não é possível ou não é viável, como, por exemplo, no caso de rede auto-organizável, na qual os sensores podem ser “lançados” em uma região para monitorar algum fenômeno de interesse. Este cenário, no entanto, ainda é de difícil realização devido aos custos proibitivos dos sensores atuais. Com o avanço das pesquisas e desenvolvimento de novos dispositivos, isto pode se tornar realidade num futuro não muito distante.

Entre outras restrições existentes ao uso atual das redes de sensores, como alcance limitado dos rádios, baixa velocidade de transmissão e custo, a limitação de energia nos sensores se mostra como um grande desafio. Em função disto, a grande maioria das pesquisas sobre o assunto tem como embasamento a preocupação com a utilização da energia disponível, pois o uso indiscriminado pode abreviar muito a vida útil do sensor.

Diversas propostas vem sendo apresentadas com foco na preservação de energia, entre as quais podemos citar:

- Controle de acesso ao meio, com o objetivo de minimizar o número de operações do sensor, como os protocolos BMAC [POL04], SMAC [YE_03] e TRAMA [RAJ03].
- Roteamento, que busca otimizar o encaminhamento de mensagens na rede, procurando otimizar a operação colaborativa com o menor número de nós envolvidos no estabelecimento de rotas, como os protocolos LEACH [HEI00] e ExOR [BIS05].

Além destes, outros trabalhos buscam otimizar a operação do sensor em função das condições do canal, buscando estratégias de controle de transmissão [SEA04] ou codificação dos dados [DES03].

Neste trabalho é estabelecido o objetivo geral de melhoria do uso de energia do sensor, visando prolongar a vida útil da rede. Para tanto, a solução proposta é embasada no uso de codificação convolucional para o controle de erros, devido principalmente à simplicidade de codificação e também de decodificação, através do *Algoritmo de Viterbi*, o qual já é muito utilizado em outras comunicações sem fio. Outro fator que contribui para esta proposta é a identificação que o consumo de energia exigido pelo processador, para realizar a decodificação, fica abaixo do consumo exigido pelo transceptor para realizar retransmissões de dados com erro.

5.2. Trabalhos Futuros

A solução aqui proposta, contudo, evidencia a oportunidade de continuação da pesquisa e aperfeiçoamento da solução, como por exemplo:

- Desenvolvimento de protocolo específico de roteamento focado em codificação.
- Adaptação das soluções já propostas de controle de acesso ao meio e de roteamento, para operação orientada também ao controle de erros.
- Estudo comparativo de novas plataformas de hardware.
- Avaliação empírica da solução em uma rede experimental.
- Estudo comparativo de novas técnicas de codificação, como por exemplo codificação concatenada.
- Continuação do estudo com codificação convolucional, com avaliação de outras taxas de código e técnicas de puncionamento.

Além destes possíveis estudos voltados para redes de sensores, eventualmente outros podem ser desenvolvidos para a adaptação e aperfeiçoamento desta solução em outras redes *ad-hoc*

Referências Bibliográficas

- [AKY02] Akyildiz, Ian F., Su, W., Sankarasubramanian, Y., Cayirci, E. “*A Survey on Sensor Networks*”, IEEE Communications Magazine, August 2002, p. 102-114.
- [ATM07] Atmel Corporation.
http://www.atmel.com/dyn/products/product_card.asp?part_id=2018. Disponível em 02/2007.
- [BIS05] Biswas, S. and Morris, R. “*ExOR: Opportunistic Multi-Hop Routing for Wireless Networks*”. SIGCOMM’05. Philadelphia, USA. August, 21-26 2005.
- [CHI07] Chipcon Products from Texas Instruments
http://www.chipcon.com/index.cfm?kat_id=2&subkat_id=12&dok_id=14. Disponível em 02/2007
- [CHO03] Chong, C.Y. and Kumar, S.P. “*Sensor Networks: Evolution, Opportunities, and Challenges*”. Proceedings of the IEEE, vol. 91, No. 8, August, 2003.p. 1247-1256.
- [CRO07] Crossbow Technology, Inc.
<http://www.xbow.com/Products/productsdetails.aspx?sid=62>. Disponível em 02/2007
- [CUL04] Culler, D. E., Hong, W. “*Wireless Sensor Networks*”. Communications of the ACM.ACM Press. New York. June 2004.
- [DES03] Desset, C. and Fort, A. “*Selection of Channel Coding for Low-Power Wireless Systems*”. The 57th IEEE Semiannual Vehicular Technology Conference (VTC-Spring). Leuven, Belgium. 2003.
- [ELS02] Elson, J., Girod, L., Estrin, D. “*Fine-Grained Network Time Synchronization using Reference Broadcasts*”. Proceedings of the Fifth Symposium on Operating Systems Design and Implementantion. Boston, USA. December, 2002
- [GAN04] Gan, L. Liu, J., Jin, X. “*Agent-Based, Energy Efficient Routing in Sensor Networks*”. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1. New York. July, 19-23. 2004. p.472-479.

- [HEI00] Heinzelman, W.B. and Chandrakasan, A. “*Energy-efficient Communication Protocol for Wireless Microsensor Networks*”. Proceedings of 33rd Hawaii International Conference on Systems Sciences. Maui, Hawaii, January, 2000.
- [HEN02] Hendrix, H. “*Texas Instruments Application Report SPRA071A*”. January 2002.
<http://focus.ti.com/docs/apps/catalog/resources/appnoteabstract.jhtml?abstractName=spra071a>. Disponível em 02/2007
- [HIL00] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K. “*System Architecture Directions for Networked Sensors*”. ACM SIGPLAN Notices. Vol 35. New York. November, 2000. p. 93-104.
- [HUL03] Hull, B., Jamieson, K., Balakrishnan, H. “*Bandwidth Management in Wireless Sensor Networks*”. Massachusetts Institute of Technology. MIT Laboratory for Computer Science. April 8, 2003.
- [KAR03] Karl, H. and Willig, A. “*A short survey of wireless sensor networks*”. Technical University Berlin. Telecommunication Networks Group. TKN Technical Report TKN-03-018. Berlin. October, 2003.
- [KAR04] Karvonen, H., Shelby, Z., Pomalaza-Raez, C. “*Coding for Energy Efficient Wireless Embedded Networks*”. International Workshop on Wireless Ad-Hoc Networks-IWWAN. Oulu, Finland, June, 2004.
- [LEV06] Levis, Phillip. “TinyOS 2.0 Documentation”. October, 2006.
<http://www.tinyos.net/tinyos-2.x/doc/html/tep111.html>. Disponível em 02/2007
- [LIN83] Lin, S. and Costello, D. “*Error control coding: fundamentals and applications.*” Englewood Cliffs: Prentice-Hall, 1983.
- [LUC04] Lucarelli, D. and Wang, I.J. “*Decentralized Synchronization Protocols with Nearest Neighbor Communication*”. Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (Sensys’04). Baltimore, USA. November 3-5, 2004.
- [MAD05] Madden, S., Franklin, M.J., Hellerstein, J., Hong, W. “*TinyDB: An Acquisitional Query Processing System for Sensor Networks*”. ACM Transactions on Database Systems (TODS). Vol 30. New York. March, 2005. p. 122-173.
- [MEL96] McEliece, R.J. and Lin, W. “*The Trellis Complexity of Convolutional Codes*”. IEEE Transactions On Information Theory, vol. 42, n. 6, November 1996.

- [POL04] Polastre, J., Hill, J., Culler, D. “*Versatile Low Power Media Access for Wireless Sensor Networks*”. Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (Sensys’04). Baltimore, USA. November 3-5, 2004.
- [PRO95] Proakis, John G., “*Digital Communications*”. Third Edition. McGraw-Hill International Editions. Singapore, 1995.
- [RAJ03] Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J.J. “*Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks*”. Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (Sensys’03). Los Angeles, USA. November 5-7, 2003.’
- [RAP96] Rappaport, T.S. “*Wireless communications: principles and practice*”. Upper Saddle River: Prentice Hall PTR, 1996.
- [SAN03] Sankarasubramaniam, Y., Akyildiz, I.F., McLaughlin, S.W. “*Energy Efficiency based Packet Size Optimization in Wireless Networks*”. Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.
- [SEA04] Seada, K., Zuniga, M., Helmy, A., Krishnamachari, B. “*Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks*”. Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (Sensys’04). Baltimore, USA. November 3-5, 2004.
- [SZE04] Szewczyk, R., Polastre, J., Mainwaring, A., Culler, D. “*Lessons From A Sensor Network Expedition*”. Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN04) . Berlin. January 19-21, 2004. p 307-322.
- [VIT67] Viterbi, A.J., “*Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*”. IEEE Transactions On Information Theory, vol. IT-13, pp. 260-269, April 1967.
- [WAR03] Warnake, B.A. “*Ultra-Low Energy Architectures and Circuits for Cubic Millimeter Distributed Wireless Sensor Networks*”. PhD Dissertation, UC Berkeley, May 2003. <http://www-bsac.eecs.berkeley.edu/archive/users/warneke-brett/pubs/index.html#Publications>. Disponível em 02/2007.
- [WIC95] Wicker, S. B. “*Error control systems for digital communication and storage*”. Upper Saddle River: Prentice Hall, 1995.
- [WIG88] Wiggert, D. “*Codes for Error Control and Synchronization*”. Norwood: Artech House. 1988.

- [XU03] Xu, Y., Bien, S., Mori, Y., Heidemann, J., Estrin, D. “*Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks*”. Center for Embedded Networked Computing, University of California. Technical Report 6. Los Angeles, 2003.
- [YE03] Ye, W. and Heidemann, J. “*Medium Access Control in Wireless Sensor Networks*”. Technical Report ISI-TR-580, USC/Information Sciences Institute, October, 2003.
- [ZUN04] Zuniga, M. and Krishnamachari, B. “*Analyzing the Transitional Region in Low Power Wireless Links*”. IEEE Secon04. B. Santa Clara, California, October 4-7, 2004.