

Controle de Erros para Redes de Sensores Bluetooth

João H. Kleinschmidt, Walter C. Borelli e Marcelo E. Pellenz

Resumo – Este artigo investiga a eficiência de diferentes esquemas de controle de erros em redes de sensores sem fio com tecnologia Bluetooth. São analisadas a confiabilidade e o consumo de energia dos pacotes Bluetooth, que usam diferentes técnicas de controle de erros, como retransmissão e codificação de canal. O pacote AUX1 é utilizado para estratégias específicas de codificação e são propostas técnicas adaptativas baseadas no número de saltos da rede. O canal sem fio é modelado com desvanecimento Rayleigh. Os resultados obtidos podem servir como parâmetros para o tipo de pacote a ser usado numa aplicação de sensoriamento.

Palavras-Chave – Bluetooth, controle de erros, redes de sensores, consumo de energia.

Abstract – This paper studies different error control schemes in wireless sensor networks with Bluetooth technology. It is analyzed the reliability and the energy consumption of Bluetooth packets, that have different error control techniques, such as retransmission and channel coding. The AUX1 packet is utilized for custom coding and also are proposed adaptive techniques based on the number of hops of the network. The wireless channel is modeled with Rayleigh fading. The results obtained may be used as references to determine the packet type in a sensor application.

Index terms – Bluetooth, error control, sensor networks, energy consumption.

I. INTRODUÇÃO

Os avanços nas comunicações sem fio e na eletrônica digital têm permitido o desenvolvimento de sensores de baixo custo e baixa potência. Estes dispositivos, que possuem componentes para sensoriamento, processamento de dados e comunicação, podem formar um tipo especial de redes ad hoc, as redes de sensores [1]. Algumas aplicações destas redes são o monitoramento ambiental em locais como florestas e oceanos, teste de qualidade de produtos num ambiente industrial, entre outras. Os protocolos e algoritmos destas redes (formação, roteamento, gerenciamento, etc) devem ter capacidade de auto-organização. Em geral, as redes de sensores diferem das redes ad hoc em alguns aspectos [1]: o número de nós sensores pode ser muito grande; os sensores são mais suscetíveis à falhas; a topologia da rede pode mudar rapidamente; os nós sensores são limitados em capacidade computacional, memória e principalmente, energia. O protocolo de acesso ao meio (MAC) é responsável pela criação da infra-estrutura da rede e pelo compartilhamento dos recursos de comunicação entre os nós. Várias tecnologias existentes vêm sendo testadas em redes de sensores e outras são desenvolvidas especificamente para este fim.

João Henrique Kleinschmidt e Walter C. Borelli, Departamento de Telemática, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas-SP, E-mails: {joaohk, borelli}@dt.fee.unicamp.br. Marcelo E. Pellenz, Programa de Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba-PR. E-mail: marcelo@ppgia.pucpr.br

O padrão Bluetooth [2] é uma tecnologia para redes sem fio de comunicação pessoal de curta distância (WPANs). Os dispositivos Bluetooth operam na faixa de frequência ISM (*Industrial, Scientific and Medical*) de 2,4 GHz utilizando a técnica de espalhamento espectral com saltos de frequência. O canal é dividido em quadros de tempo de 625 μ s. O protocolo MAC do Bluetooth foi projetado para facilitar a formação de redes ad hoc. Esta característica torna a tecnologia Bluetooth atrativa para aplicações em redes de sensores, assim como seu baixo custo, suporte a comunicação com múltiplos saltos (*multihop*), processo de descoberta de dispositivos e modos de economia de energia. Os dispositivos podem se comunicar formando uma rede de até oito nós, chamada de *piconet*. Numa *piconet*, um dispositivo assume a função de mestre e os outros de escravos. Dispositivos em diferentes *piconets* podem se comunicar usando uma estrutura chamada *scatternet*. Em [3] e [4] foram implementadas redes de sensores usando Bluetooth como protocolo de acesso ao meio. Uma das questões importantes é a formação e roteamento das *scatternets* para tornar possível a comunicação com múltiplos saltos. Embora vários protocolos tenham sido propostos na literatura, poucos consideram essa questão para aplicações de sensores, com requisitos rígidos de energia. Algumas propostas foram apresentadas em [5], [6], [7] e [8].

Os canais sem fio podem ter taxas de erro altas devido à interferência e ao desvanecimento gerado pelo multipercorso, levando ao desperdício de energia. Para melhorar a confiabilidade dos dados enviados no canal sem fio, várias técnicas podem ser empregadas, como o uso de estratégias de retransmissão (ARQ), códigos corretores de erro (FEC) ou o controle da potência de transmissão. Neste artigo analisamos o desempenho dos pacotes de dados Bluetooth visando a característica mais importante em redes de sensores: o baixo consumo de energia. Alguns trabalhos analisam o desempenho [9], mas sempre considerando a taxa de transmissão de dados, que é um fator secundário para as redes de sensores, que tipicamente possuem uma baixa taxa de dados. Os pacotes Bluetooth usados na transmissão de dados possuem diferentes tipos de correção de erros, desde retransmissão ou código de Hamming e são apresentados na Seção II. Também é considerada a proposta de codificação sugerida em [10] e é proposto um novo esquema adaptativo de controle de erros que muda o tipo de pacote de acordo com o número de saltos (*hops*) na rede. Na Seção III é apresentado o modelo de simulação utilizado. Os resultados obtidos são descritos na Seção IV e a Seção V faz as considerações finais.

II. ESTRATÉGIAS DE CONTROLE DE ERROS DO PADRÃO BLUETOOTH

A especificação Bluetooth [2] define sete pacotes de dados (assíncronos). Cada pacote consiste de três campos: o código de acesso (72 bits), cabeçalho (54 bits) e o *payload* (0-

2745 bits). O código de acesso é usado para sincronização e o cabeçalho possui informações como tipo de pacote, controle de fluxo e confirmação de pacotes. O código de acesso é robusto a erros, pois as palavras de sincronização têm uma distância de Hamming grande ($d_{min} = 14$). O cabeçalho contém um código de repetição (3,1) para verificação de erro. O payload carrega os bytes de dados, que variam de acordo com o pacote. A maioria dos pacotes é protegida por um esquema de retransmissão ARQ *stop-and-wait* baseado num código CRC (*Cyclic Redundancy Check*). A unidade receptora indica no próximo pacote de retorno se houve erro ou não. Alguns pacotes possuem os dados protegidos por um código de Hamming de taxa 2/3 (15,10). Este código corrige todos os padrões de erros simples e detecta todos os erros duplos numa palavra código. A Tabela 1 mostra essas informações para cada tipo de pacote assíncrono.

TABELA I. TIPOS DE PACOTES ASSÍNCRONOS.

Pacote	Quadros de tempo	Dados (bytes)	FEC	CRC e ARQ
DM1	1	0-17	Sim	Sim
DH1	1	0-27	Não	Sim
DM3	3	0-121	Sim	Sim
DH3	3	0-183	Não	Sim
DM5	5	0-224	Sim	Sim
DH5	5	0-339	Não	Sim
AUX1	1	0-29	Não	Não

Um pacote enviado não é aceito pelo receptor quando qualquer um dos cinco eventos acontece: (A) o destino falha ao sincronizar com o código de acesso do pacote enviado; (B) o cabeçalho do pacote enviado está corrompido (depois que o código de repetição é decodificado); (C) os dados do pacote enviado estão corrompidos depois que o código de Hamming é decodificado, causando a falha na verificação de CRC; (D) a fonte é incapaz de sincronizar com o código de acesso do pacote de retorno e (E) o cabeçalho do pacote de retorno está corrompido. Em [9], as probabilidades destes eventos são deduzidas e foram adaptadas para serem usadas neste artigo.

A sincronização é feita correlacionando a saída do demodulador com uma cópia gravada do código de acesso. Um pacote é sincronizado se a saída do correlator excedeu um determinado limiar T . O quadro é sincronizado se pelo menos T dos 72 bits do código de acesso forem demodulados corretamente ($T = 65$ neste trabalho). A sincronização com o pacote enviado ocorre se não há mais que $(72 - T)$ erros no código de acesso recebido e sua probabilidade é dada por:

$$P[\overline{A}] = \sum_{k=0}^{72-T} \binom{72}{k} \cdot [p(\gamma_f)]^k \cdot [1 - p(\gamma_f)]^{72-k}, \quad (1)$$

onde $p(\gamma_f)$ é a probabilidade de erro de símbolo do canal direto em função da relação sinal-ruído (SNR) média recebida $\bar{\gamma}$. Como o pacote de retorno também tem um código de acesso de 72 bits, a probabilidade para o evento D é calculada da mesma forma,

$$P[\overline{D}] = \sum_{k=0}^{72-T} \binom{72}{k} \cdot [p(\gamma_r)]^k \cdot [1 - p(\gamma_r)]^{72-k}, \quad (2)$$

onde $p(\gamma_r)$ é a probabilidade de erro de símbolo do canal reverso. O canal direto é usado para envio dos pacotes de dados e o de retorno para indicar o sucesso ou insucesso do recebimento do pacote (para transmissão unidirecional). O evento B ou E ocorre se qualquer das oito triplas do código de repetição (3,1) for decodificada incorretamente,

$$P[\overline{B}] = \{3p(\gamma_f)[1 - p(\gamma_f)]^2 + [1 - p(\gamma_f)]^3\}^{18} \quad (3)$$

$$P[\overline{E}] = \{3p(\gamma_r)[1 - p(\gamma_r)]^2 + [1 - p(\gamma_r)]^3\}^{18} \quad (4)$$

O erro mais provável é causado pelo evento C. Para os pacotes DHx ocorre quando qualquer um dos bits de dados for recebido com erro:

$$P[\overline{C}] = [1 - p(\gamma_f)]^b, \quad (5)$$

onde b é o tamanho do payload do pacote em bits. Para os pacotes DMx os dados são protegidos por um código de Hamming (15,10), onde B é o número de blocos de 10 bits que entram no codificador. A probabilidade do evento C para os pacotes DMx é:

$$P[\overline{C}] = [15p(\gamma_f)[1 - p(\gamma_f)]^{14} + [1 - p(\gamma_f)]^{15}]^B \quad (6)$$

Bluetooth utiliza modulação GFSK com o produto tempo-largura de faixa $BT = 0,5$ e índice de modulação h entre 0,28 e 0,35. A probabilidade de erro de símbolo $p(\gamma)$ para a modulação GFSK deve ser aplicada nas equações (1) a (6) e é dada por [12]:

$$p(\gamma) = Q_1(a, b) - \frac{1}{2} e^{(a^2+b^2)/2} I_0(ab) \quad (7)$$

Sendo que $Q_1(a, b)$ é a função Q-Marcum, I_0 é a função de Bessel modificada de primeira ordem e a e b são constantes que dependem da relação sinal-ruído e da correlação [12]. Assim, a probabilidade de erro de pacote do canal direto, PER_f e reverso, PER_r , podem ser definidas [13] por:

$$PER_f = 1 - \int_0^\infty f(\gamma_f) P[\overline{A}] P[\overline{B}] P[\overline{C}] d\gamma_f \quad (8)$$

$$PER_r = 1 - \int_0^\infty f(\gamma_r) P[\overline{D}] P[\overline{E}] d\gamma_r \quad (9)$$

onde $f(\gamma_f)$ e $f(\gamma_r)$ são as funções de densidade de probabilidade do canal direto e reverso, respectivamente.

Além dos pacotes definidos pela especificação com seus respectivos controles de erro, pode-se implementar codificação customizada utilizando os pacotes AUX1 [10]. Usando este pacote, o esquema ARQ é desativado e o dispositivo Bluetooth entrega os bits recebidos independente se eles estão corretos ou não. Enquanto os seis pacotes assíncronos padrão mantêm um enlace confiável com atraso aleatório (que se aproxima de infinito para baixos valores da relação sinal-ruído), o pacote AUX1 provê um enlace não confiável com atraso de um quadro de tempo. Os demais pacotes não permitem que sejam aplicados outros códigos corretores de erro, a não ser que fossem feitas modificações na especificação [2]. Em [10] é proposta a utilização de códigos BCH juntamente com o código CRC para detecção de erros. Neste caso, o ARQ deve ser implementado na camada de aplicação, pois é desativado no dispositivo Bluetooth. A codificação é então implementada transportando no payload um código BCH (232, k). A entrada no

codificador BCH consiste dos dados da fonte e dos 2 bytes do CRC. Logo, o número de bits de dados é $K=k-16$. Para o pacote ser aceito, os eventos A, B e C não devem ocorrer. Apenas a probabilidade de decodificação correta do payload é diferente, e é dada pela equação (10) [10]. Neste artigo foi considerado um código binário BCH (232,156), com capacidade de corrigir até $t=10$ erros.

$$P[\overline{C}] = \sum_{k=0}^t \binom{232}{k} \cdot [p(\gamma_f)]^k \cdot [1 - p(\gamma_f)]^{232-k} \quad (10)$$

Além da nova abordagem utilizando o código BCH, é proposta uma outra modificação (HAM) no pacote AUX1: codificar o payload com o mesmo código de Hamming (15,10) usado nos pacotes DMx, porém sem o uso de CRC. Embora isto possa ter um custo na confiabilidade de entrega dos pacotes, em termos de consumo de energia pode ser bastante útil, pois não há necessidade do envio de um pacote de retorno para indicar o sucesso de uma transmissão. A Tabela 2 mostra as informações de controle de erros para estes dois novos tipos de pacotes.

TABELA II. TIPOS DE PACOTES COM CONTROLE DE ERROS CUSTOMIZADO.

Pacote	Quadros de tempo	Dados (bytes)	FEC	CRC e ARQ
HAM	1	0-18	Hamming (15,10)	Não
BCH	1	0-17	BCH (232,156)	Sim

Nem sempre usar o mesmo esquema de controle de erros para toda a rede pode ser a melhor solução. Se em algumas situações não é um grande problema que um pacote se perca, em outras a perda de um pacote pode não ser tolerada. O uso de um esquema adaptativo pode ser implementado para verificar a importância de um pacote e escolher um código eficiente para aquele pacote em particular. No caso do Bluetooth, mudar o tipo de correção de erros significa mudar o tipo de pacote a ser transmitido. Para utilizar um esquema adaptativo numa rede de sensores, onde a questão mais importante é reduzir o consumo de energia, foi usada uma abordagem semelhante à proposta em [11] e que será adaptada usando os diferentes pacotes Bluetooth.

Para determinar a importância de um pacote, e conseqüentemente, o tipo de controle de erros a ser aplicado, é usado o princípio de múltiplos saltos de uma rede de sensores, como mostra a Figura 1. O nó sensor envia um pacote de dados, contendo a informação de temperatura de um ambiente, por exemplo, para o nó coletor. O coletor é o responsável em receber os dados de toda a rede. Porém, antes do pacote chegar ao nó coletor, deve passar por outros nós, que podem ser outros sensores ou algum outro tipo de nó com capacidade de roteamento. Se o pacote se perder no primeiro salto, apenas a energia para enviar o pacote do sensor ao nó 1 foi perdida. Se o pacote tiver erros depois que o nó 4 envia ao nó coletor, foi gasta mais energia para enviar o pacote pela rede. Logo, um pacote é mais importante quanto maior o número de saltos que tiver atravessado na rede até o seu destino, no coletor. Então diferentes pacotes Bluetooth podem ser aplicados ao longo do caminho.

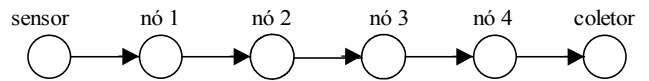


Figura 1. Exemplo de uma rede de sensores com múltiplos saltos.

III. MODELO DE SIMULAÇÃO

Para avaliar o consumo de energia dos pacotes foram feitas simulações usando o software Matlab, que são descritas a seguir. O exemplo de rede considerado foi o da Figura 1, onde um sensor deve enviar dados ao nó coletor. Esta estrutura pode ser considerada apenas como um de vários caminhos da rede de sensores. Está sendo assumido que a scatternet Bluetooth foi formada e que as rotas também estão definidas, usando métodos como os propostos em [5], [6], [7] e [8]. O tráfego é gerado pelo nó sensor, que o envia ao nó 1, e assim sucessivamente, até atingir o coletor. O canal sem fio foi modelado usando o desvanecimento Rayleigh, cuja função densidade de probabilidade é dada por:

$$f(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right), \quad \text{para } \gamma \geq 0 \quad (11)$$

sendo $\bar{\gamma}$ a SNR média recebida e γ a SNR instantânea.

A equação (11) é aplicada em (8) e (9) para calcular as probabilidades de erro de cada pacote. Estas probabilidades são dadas em função da relação sinal-ruído. Quando um nó recebe um pacote, é verificado se aconteceram erros na recepção. Se não houver, o pacote é enviado ao próximo nó. Nos pacotes com ARQ, é ainda enviado um pacote ao nó transmissor indicando o sucesso da transmissão (um pacote Bluetooth do tipo NULL). Se há erro, ou o pacote é descartado (no caso de pacotes sem ARQ) ou enviado ao transmissor um pacote NULL indicando o insucesso da transmissão, fazendo com que o pacote seja reenviado. É importante notar que o pacote NULL usado para indicar o sucesso ou não de uma transmissão também pode sofrer erros, embora não transporte dados (possui apenas os campos de código de acesso e o cabeçalho).

Para simular o consumo de energia foi utilizado um modelo simplificado. Como não está sendo considerado um hardware específico, o consumo de energia é expresso apenas em termos normalizados. As energias consideradas são as gastas para a transmissão e recepção de pacotes, que são as tarefas que mais consomem energia. A energia de processamento gasta para codificar ou decodificar um pacote foi desprezada. A codificação é uma tarefa geralmente mais simples e com um consumo menor que a decodificação, que é mais complexa [11] [14]. Considerando o mesmo modelo de [11], na qual a recepção de um determinado número de bits consome aproximadamente 75% da energia gasta para transmitir a mesma quantidade de bits, a energia consumida total E é dada por:

$$E = n_{bits} * T * 1 + n_{bits} * R * 0,75 + n_{ack} * T_a * 1 + n_{ack} * R_a * 0,75 \quad (12)$$

sendo n_{bits} o número total de bits de um pacote (código de acesso, cabeçalho e payload), T o número total de pacotes transmitidos (incluindo as retransmissões), R o número total de pacotes recebidos (incluindo as retransmissões), n_{ack} o número total de bits do pacote de retorno (NULL) e T_a e R_a o

número de pacotes de retorno transmitidos e recebidos, respectivamente.

A confiabilidade é dada pela porcentagem dos 10000 pacotes enviados que chegam corretos no coletor. Sendo n_{pac} o número total de pacotes enviados pelo sensor e n_{erro} o número de pacotes que chegam com erro no nó coletor, a confiabilidade C é dada por:

$$C = ((n_{pac} - n_{erro}) / n_{pac}) * 100 \quad (13)$$

Para o controle adaptativo de erros, cada pacote deve manter um contador do número de saltos que o pacote teve na rede. Isto pode ser implementado como um campo no payload do pacote. Foram usados dois esquemas adaptativos diferentes: ADP1 e ADP2. Um pacote com menor controle de erros é usado no primeiro salto e pacotes com codificação mais potente nos demais saltos. Para o caso de dois saltos, o esquema adaptativo 1 (ADP1) usa AUX1 na metade das transmissões do primeiro salto e BCH no segundo salto. No esquema ADP2 foi usado o pacote AUX1 no primeiro salto e DH1 no segundo salto. Para cinco saltos, o ADP1 usa o pacote AUX1 no primeiro salto e BCH nos demais saltos. Para ADP2, no primeiro salto usa-se AUX1, no segundo HAM e nos demais saltos BCH.

Na simulação são enviados 10000 pacotes do sensor para o coletor, para quatro diferentes condições de canal. Na primeira simulação é usado um ambiente livre de erros. Nos outros três casos, o canal tem desvanecimento Rayleigh com 30dB, 20 dB ou 10 dB de relação sinal-ruído média recebida. Foram considerados que o tamanho dos dados a serem transmitidos são de 17 ou 32 bytes. Estes dados podem ser transmitidos em intervalos regulares, e podem ser a temperatura de um ambiente ou outra variável que possa ser transmitida com poucos bytes. O valor de 17 bytes foi escolhido por ser o número máximo de dados que os pacotes DM1 e BCH podem transmitir. Para transmitir 32 bytes de informação, devem ser usados dois pacotes para DM1, DH1, AUX1, HAM e BCH. Porém, enquanto que o primeiro pacote contém sua capacidade máxima de dados, o segundo contém apenas o número de bytes que falta para completar os 32 bytes de dados. Nas simulações com 17 bytes, não são usados os pacotes DM3, DH3, DM5 e DH5, pois estes teriam um comportamento idêntico a DM1 e DH1. Com 32 bytes os pacotes DM3 e DH3 são usados e precisam de apenas um pacote para enviar os dados. Os pacotes DM5 e DH5 também não são usados, pois teriam um comportamento igual ao DM3 e DH3.

IV. RESULTADOS

As Figuras 2 a 12 mostram os resultados obtidos de consumo de energia e confiabilidade de entrega dos pacotes. Nos pacotes com retransmissão está sendo considerado que o código CRC detecta qualquer tipo de erro, solicitando então a retransmissão do pacote. É por esta razão que estes pacotes sempre atingem 100% de confiabilidade, ao custo de uma maior energia consumida. Os resultados são médias obtidas com várias simulações.

Para um canal sem erros (Figuras 2 e 9), o pacote AUX1 sempre tem vantagem em relação aos outros, independente do

número de saltos ou tamanho dos dados, pois atinge os mesmos 100% de confiabilidade com o menor custo de energia. Nestas condições, o pior pacote é o DM1, pois a redundância inserida pelo código de Hamming e o pacote de retorno acrescentam bits a serem transmitidos sem necessidade. No entanto, um canal sem erros não ocorre na prática. No caso de um canal Rayleigh com 30 dB de relação sinal-ruído (Figuras 3, 6 e 10) já temos um canal que introduz erros nos bits transmitidos. Novamente o pacote AUX1 tem o menor consumo, mas a confiabilidade diminui. Mesmo não sendo de 100%, ainda é muito grande, geralmente acima de 90%. Esta perda nem sempre é um grande problema, pois em muitas aplicações vários sensores coletam informações redundantes de um ambiente. Se alguma informação se perde, os sensores restantes ainda podem fornecer medidas confiáveis. O pacote HAM tem uma confiabilidade um pouco maior, porém com um gasto de energia também mais alto. O esquema adaptativo ADP2 teve uma boa eficiência para o caso de 2 saltos (Figuras 3 e 10), pois consome apenas um pouco mais de energia e atinge quase 100% de confiabilidade.

Em canais com 20 dB de relação sinal-ruído (Figuras 4, 7 e 11), várias mudanças podem ser percebidas. Como o canal está mais suscetível a erros, os pacotes com controle de erros começam a apresentar uma melhora em seu desempenho. O pacote AUX1 ainda atinge quase 80% de acerto para 2 saltos e 17 bytes (Figura 4), mas com 5 saltos (Figura 7) a confiabilidade cai bastante, pois a probabilidade de erro é acumulada a cada salto. Os esquemas adaptativos ADP1 e ADP2 apresentam um bom ganho de desempenho nestas condições, pois o uso de controle de erros mais eficientes contém a acumulação da probabilidade de erro. Os pacotes com retransmissão tem um aumento significativo no consumo de energia se comparado a 30 dB. Outro dado importante é que nos pacotes com 32 bytes de dados (Figuras 9, 10, 11 e 12) o pacote DH3 tem alta confiabilidade e um consumo pouco maior que os pacotes sem retransmissão. Isto ocorre porque enquanto os outros precisam de dois pacotes para enviar os dados (adicionando mais um campo de cabeçalho e código de acesso), o pacote DH3 (e DM3) coloca toda a informação em apenas um pacote. Já se pode notar que aumentando o número de bytes de dados, estes pacotes sempre terão vantagem em relação aos demais. Para redes de sensores com taxa de dados mais alta, os pacotes com codificação customizada e os esquemas adaptativos não terão boa eficiência, pois precisam de mais pacotes para enviar a mesma quantidade de dados.

Quando temos um canal em que a relação sinal-ruído recebida ficar em apenas 10 dB (Figuras 5, 8 e 12), o canal terá muitos erros e o consumo de energia sobe drasticamente para os pacotes com ARQ, devido ao grande número de retransmissões. Nas mesmas condições, os pacotes sem ARQ têm sua confiabilidade praticamente reduzida a zero. De todos os pacotes com ARQ, o BCH é o que tem o menor consumo de energia para pacotes com 17 bytes (Figuras 5 e 8) e para 32 bytes (Figura 12) é o pacote DM3. Como esperado, a codificação apresenta melhores resultados quando as condições de canal são piores. Os esquemas adaptativos ADP1 e ADP2 conseguem uma pequena melhora no consumo de energia, mas com a confiabilidade não muito alta.

Fica claro nos resultados obtidos que para cada condição de canal um determinado pacote apresenta um melhor desempenho e um método para estimar a relação sinal-ruído seria bastante útil. No entanto, isto requer que o receptor tenha esta capacidade, estimando a SNR para cada pacote recebido. Cada aplicação também pode ter requisitos diferentes e se a confiabilidade exigida não for muito alta, grandes economias de energia podem ser feitas. O número de saltos também afeta diretamente a escolha do pacote. Para o caso específico do Bluetooth, dados maiores que 32 bytes praticamente não se beneficiariam da economia de energia que o pacote AUX1 e seus variantes podem proporcionar, deixando a escolha do pacote basicamente entre DM3 e DH3 ou DM5 e DH5.

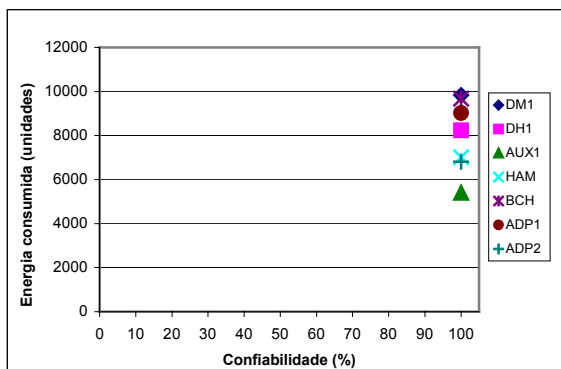


Figura 2. Canal sem erros, 17 bytes, 2 saltos.

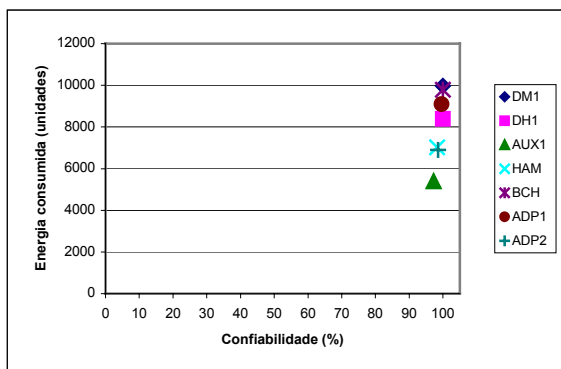


Figura 3. Canal Rayleigh 30 dB, 17 bytes, 2 saltos.

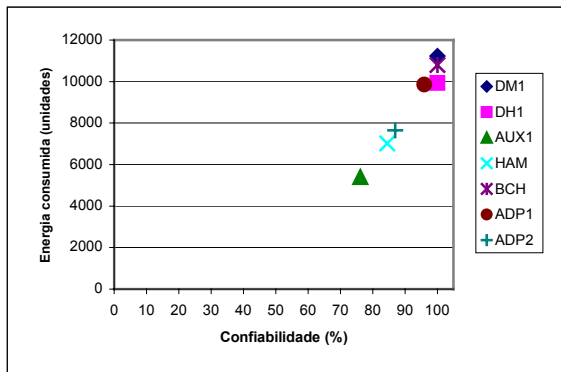


Figura 4. Canal Rayleigh 20 dB, 17 bytes, 2 saltos.

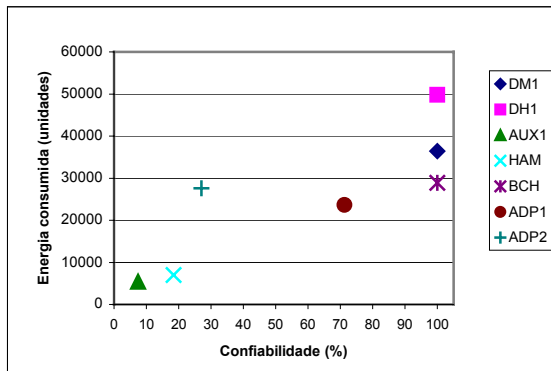


Figura 5. Canal Rayleigh 10 dB, 17 bytes, 2 saltos.

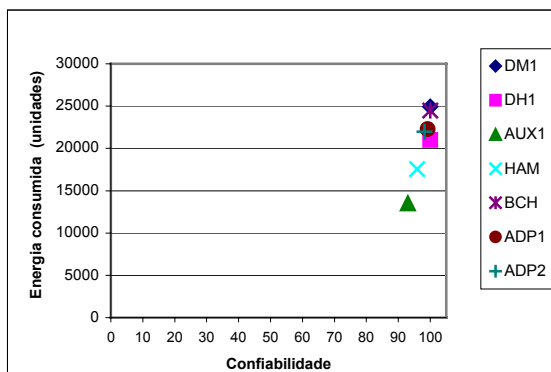


Figura 6. Canal Rayleigh 30 dB, 17 bytes, 5 saltos.

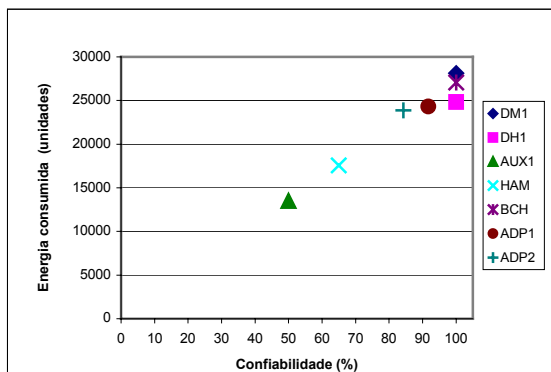


Figura 7. Canal Rayleigh 20 dB, 17 bytes, 5 saltos.

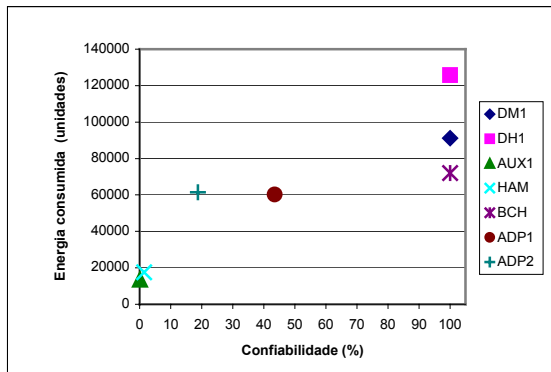


Figura 8. Canal Rayleigh 10 dB, 17 bytes, 5 saltos.

V. CONCLUSÃO

Neste artigo foi abordado o problema de consumo de energia em redes de sensores Bluetooth. Foram analisadas diferentes técnicas de controle de erro, baseadas em retransmissão e códigos corretores de erros. O pacote AUX1 pode ser útil em redes de sensores por não utilizar retransmissão e permitir que estratégias específicas de correção de erros sejam empregadas, como foi proposto com códigos de Hamming e BCH. Também foram propostos esquemas adaptativos, ADP1 e ADP2, que mudam a técnica utilizada para os erros de acordo com o número de saltos na rede. Outras técnicas adaptativas similares podem ser propostas, sempre tentando diminuir o consumo de energia em condições de canal ruins, já que quando o canal está relativamente livre de erros o pacote sem retransmissão tem um baixo consumo e alta confiabilidade. Embora a energia gasta para codificar e decodificar não tenha sido considerada, os pacotes AUX1 e os esquemas adaptativos podem apresentar benefícios adicionais, pois como possuem menos codificação, gastarão menos energia. Os resultados obtidos podem dar indicações do tipo de pacote a ser usado numa aplicação de sensoriamento, de acordo com a confiabilidade desejada.

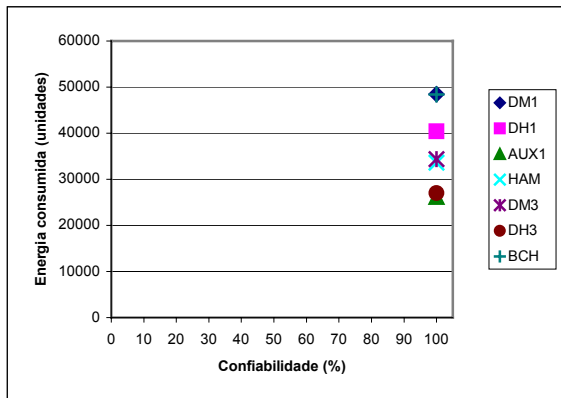


Figura 9. Canal sem erros, 32 bytes, 2 saltos.

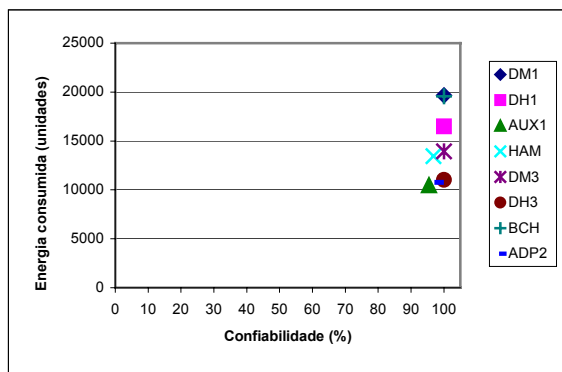


Figura 10. Canal Rayleigh 30 dB, 32 bytes, 2 saltos.

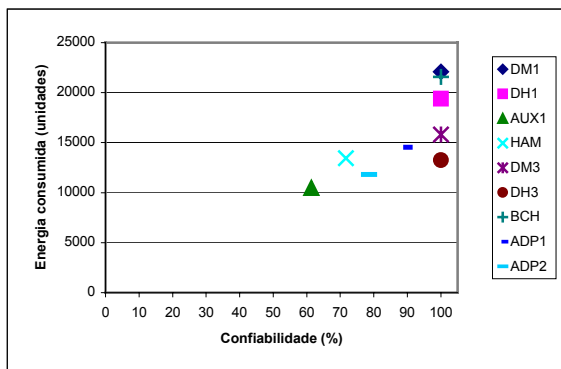


Figura 11. Canal Rayleigh 20 dB, 32 bytes, 2 saltos.

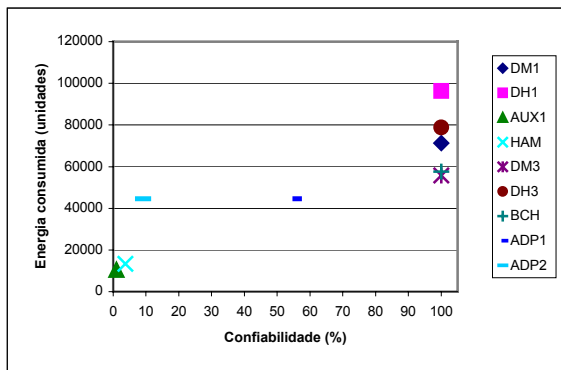


Figura 12. Canal Rayleigh 10 dB, 32 bytes, 2 saltos.

REFERÊNCIAS

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam e E. Cayirci, "A survey on sensor networks", *IEEE Communications Magazine*, pp. 102-114, Agosto 2002.
- [2] Bluetooth SIG, "Specifications of the Bluetooth system", *Core Version 1.2*, Novembro 2003. <http://www.bluetooth.com>
- [3] O. Kasten e M. Langheinrich, "First experiences with Bluetooth in the smart-its distributed sensor network", Workshop on Ubiquitous Computing and Communications, Barcelona, Spain, Setembro 2001.
- [4] M. Leopold, M. D. Dydenborg e P. Bonnet, "Bluetooth and sensor networks: a reality check", 1st ACM Conference on Sensor Systems, Los Angeles, CA, USA, Novembro 2003.
- [5] V. Mehta e M. El Zarki, "Bluetooth based sensor network for civil infrastructure health monitoring", *Wireless Networks*, Kluwer Academic Publishers, vol. 10, pp. 401-412, Julho 2004.
- [6] S. Saginbekov e I. Korpeoglu, "An energy efficient scatternet formation algorithm for Bluetooth-based sensor networks", 2nd European Workshop on Wireless Sensor Networks, Istanbul, Turkey, Fevereiro 2005.
- [7] H. Mathias, D. Jan e T. Dirk, "Energy-efficient data collection for Bluetooth-based sensor networks", IEEE Instrumentation and Measurement Technology Conference, Italy, Maio 2004.
- [8] J. H. Kleinschmidt, M. E. Pellenz e L. A. P. Lima Jr., "Uma aplicação de redes de sensores usando Bluetooth", XXI Simpósio Brasileiro de Telecomunicações, Belém, Brasil, Setembro 2004.
- [9] M. C. Valenti, M. Robert e J. H. Reed, "On the throughput of Bluetooth data transmissions", IEEE Wireless Communications and Networking Conference, Orlando, USA, Março 2002.
- [10] M. C. Valenti e M. Robert, "Custom coding, adaptive rate control and distributed detection for Bluetooth", Proc. IEEE Vehicular Technology Conference, Vancouver, BC, Setembro 2002.
- [11] J. Meer, M. Nijdam and M. Bijl, "Adaptive error control in a wireless sensor network using packet importance valuation", Hardware/software co-design, Enschede, Netherlands, Maio 2003.
- [12] J. Proakis, Digital Communications, New York, NY: McGraw-Hill, 4th edition, 2001.
- [13] J. H. Kleinschmidt, M. E. Pellenz and L. A. P. Lima Jr., "Evaluating and improving Bluetooth piconet performance over Nakagami-*m* fading channels", Proc. of the Ninth IEEE International Symposium on Computers and Communications, Alexandria, Egypt, Julho 2004.
- [14] C. Desset e A. Fort, "Selection of channel coding for low-power wireless systems", Proc. IEEE Vehicular Technology Conference, Jeju, Korea, Abril 2003.