Reduzindo a latência de comunicação em múltiplos saltos dos mecanismos de *duty cycle* assíncrono baseados em *schedule* através de sincronização de baixa resolução

André R. C. Saraiva[1], **Diego Passos**[1], **Ricardo C. Carrano** [2] **e Célio V. N. Albuquerque** [1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
[2]Departamento de Engenharia de Telecomunicações – Universidade Federal Fluminense (UFF)
Av. Gal. Milton Tavares de Souza, s/nº – 24.210-346 – São Domingos – Niterói – RJ – Brasil

{andresaraiva,dpassos,celio}@ic.uff.br

carrano@midiacom.uff.br

Abstract. *In Wireless Sensor Networks, nodes typically employ batteries that cannot be recharged or replaced. Hence, optimizing energy consumption is a major concern. Among the several solutions already proposed, schedule-based asynchronous duty cycle methods are the simplest because they do not require mechanisms, protocols or specific hardware for clock synchronization between nodes. These solutions, however, result in high latency for multi-hop communication. In this work, we show how existing asynchronous mechanisms can benefit from a low level of synchronism, with resolution of slots. Assuming this possibility, we show, using numerical simulations, that the use of specific offsets between clocks of neighboring nodes according to their distances to the sink node drastically reduces latency.*

Resumo. *Nas Redes de Sensores Sem Fio, nós tipicamente operam por baterias não recarregáveis ou substituíveis. Logo, a otimização do consumo energético é uma das principais preocupações. Dentre as várias soluções já propostas, os métodos de duty cycle assíncronos baseados em schedule se mostram os mais simples por não demandarem mecanismos, protocolos ou hardwares específicos para sincronização de relógio entre os nós. Estes métodos, no entanto, resultam em alta latência para comunicação de múltiplos saltos. Neste trabalho, demonstramos como os mecanismos assíncronos já existentes podem se beneficiar de um baixo nível de sincronismo, em resolução de slots. Assumindo esta possibilidade, mostramos com simulações numéricas que o uso de offsets específicos entre os relógios de nós vizinhos, de acordo com suas distâncias ao nó sorvedouro, reduz drasticamente a latência.*

1. Introdução

Em um mundo globalizado, com ênfase na sustentabilidade, a eficiência energética dos equipamentos eletrônicos é de suma importância. Em particular, as redes sem fio de múltiplos saltos, como as Redes de Sensores Sem Fio (RSSF), apresentam severas limitações energéticas por comumente dependerem de fontes de energia portáteis. Além das limitações energéticas, a troca da fonte de energia de nós sensores, na maioria das vezes, não é trivial ou mesmo possível, fazendo com que o gerenciamento de energia seja vital para operação adequada da RSSF. O *duty cycling* da interface de rádio,

que é um dos componentes responsáveis pela drenagem de energia, se torna necessário [Anastasi et al. 2009, Bachir et al. 2010], alternando períodos de atividade e inatividade.

Entretanto, fazer com que todos os elementos de uma RSSF mantenham de forma coordenada seus períodos de atividade e inatividade a fim de garantir que um nó sempre encontre um nó vizinho para transmitir seus dados com sucesso requer mecanismos síncronos, com um relógio de tempo comum compartilhado por todos os nós, ou assíncronos, onde um relógio comum não se faz necessário.

Mecanismos síncronos comumente requerem *hardware* adicional, geram um maior tráfego de controle e resultam em maiores custos [Carrano et al. 2014a]. Em contrapartida, mecanismos assíncronos não demandam *hardware* especializado ou a adição de grande volume de tráfego de controle, se mostrando, portanto, uma alternativa interessante no contexto de redes de sensores. Porém, segundo [Ye et al. 2002, Lu et al. 2004], os mecanismos assíncronos apresentam uma latência excessiva ao longo de caminhos múltiplos saltos.

Mesmo assim, diversas propostas assíncronas baseadas em *schedules* podem ser encontradas na literatura, como *Block Design* [Zheng et al. 2003], *Grid* [Jiang et al. 2005], *Torus* [Tseng et al. 2003] e *Disco* [Dutta and Culler 2008]. No entanto, estas propostas não abordam a questão da elevada latência fim-a-fim.

O problema de *sleep waiting* e o problema de *data forwarding interruption* [Lu et al. 2004] são exemplos de fenômenos que contribuem para o aumento da latência, uma vez que um transmissor deverá aguardar certo tempo até que ele e o receptor pretendido fiquem, ambos, com seus rádios ligados simultaneamente (o que constitui uma oportunidade de encontro). Estes fenômenos fazem com que o tempo de descoberta de vizinho (ou NDT, do inglês *Neighbor Discovery Time*) seja elevado.

Embora não seja o escopo deste trabalho apresentar um protocolo de sincronização, neste artigo, é estudada a possibilidade de introduzir um certo grau de sincronismo de baixa resolução em mecanismos assíncronos, sem que haja a necessidade de *hardware* adicional ou aumento significativo de tráfego de controle, mas proporcionando uma redução representativa do NDT. Para tanto, considera-se a possibilidade de existência de um protocolo de sincronização dos nós em resolução de *slots* que permita que nós vizinhos sigam escalonamentos com *offsets* específicos (ao invés dos *offsets* aleatórios resultantes da natural dessincronização dos relógios dos nós). Através do uso de simulações numéricas, este trabalho demonstra que a utilização de certos valores específicos de *offset* reduz consideravelmente a latência da comunicação em múltiplos saltos.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta uma revisão bibliográfica sobre alguns mecanismos de *duty cycle* síncronos e assíncronos. A Seção 3 mostra como o uso de uma sincronização de baixa resolução pode reduzir a latência da comunicação em múltiplos saltos. A Seção 4 avalia os benefícios da sincronização proposta através de simulações numéricas. Por fim, a Seção 5 apresenta considerações finais e ideias para trabalhos futuros.

2. Revisão Bibliográfica

Mecanismos de *duty cycling* de rádio têm como objetivo reduzir o tempo de escuta ociosa (*i.e.*, intervalos em que o rádio se encontra ligado sem que haja transmissões ou recepções

de quadros de interesse) dos nós de uma rede sem fio, reduzindo assim o consumo de energia e aumentando a vida útil da rede. Na literatura da área, define-se o *duty cycle* de um nó como o percentual de tempo que este mantém seu rádio ligado [Carrano et al. 2014b]. Considerando os requisitos e capacidades atuais dos nós sensores, *duty cycles* inferiores a 1% são desejáveis [Carrano et al. 2014b]. Em contrapartida, a realização de *duty cycling* do rádio não é trivial, já que algum nível de coordenação é necessário para garantir que quaisquer dois nós vizinhos ainda tenham oportunidades de comunicação (*i.e.*, que exista algum período suficientemente longo em que os nós tenham seus rádios simultaneamente ligados). Ademais, a redução do *duty cycle* dos nós de uma rede geralmente é associada a um aumento da latência de comunicação. Por estes motivos, a comunidade acadêmica tem realizado esforços para obter soluções viáveis com *duty cycles* baixos e, idealmente, baixo *overhead* de controle [Zheng et al. 2003].

2.1. Principais Mecanismos Síncronos

Como forma de sincronizar o período de atividade dos rádios, reduzindo a latência, a escuta ociosa e a perda de pacotes, os esquemas síncronos introduzem um relógio comum entre todos os nós da rede. O estabelecimento deste relógio comum demanda a utilização permanente de protocolos de sincronização de relógio ou a adição de *hardware* especializado para manter a sincronização com o grau de precisão necessário. Estes mecanismos podem ser classificados em duas famílias: os esquemas estritamente síncronos ou baseados em *Rendezvous* [Sivrikaya and Yener 2004] e os esquemas baseados em escalonamento [Anastasi et al. 2009].

Os esquemas estritamente síncronos são os de mais fácil compreensão, uma vez que todos os nós deverão ter seus rádios ligados ou desligados ao mesmo tempo. No entanto, em múltiplos saltos este esquema torna-se complexo, já que erros de sincronização entre os nós são comuns e tendem a se acumular ao longo dos saltos, dificultando a sincronização da rede como um todo e, muitas vezes, demandando elementos adicionais de *hardware* e certa quantidade de mensagens de controle [Sivrikaya and Yener 2004].

Exemplos de esquemas estritamente síncronos encontrados na literatura são o RT-Link [Rowe et al. 2008], que acrescenta um GPS para sincronização, e TRAMA [Rajendran et al. 2006], que tem como princípio a eliminação das colisões e a suspensão dos nós que não participam da comunicação naquele instante de tempo. Ambos utilizam o protocolo de acesso múltiplo TDMA, do inglês *Time Division Medium Access*, evitando colisões entre pacotes, transmissões desnecessárias e escutas ociosas, e consequentemente desperdício de bateria, características estas que estão entre as principais vantagens destes esquemas.

Outras propostas são os esquemas baseados em escalonamento, que visam resolver o problema de interrupção de encaminhamento de dados (do inglês *Data Forwarding Interruption Problem*) – que segundo [Lu et al. 2004] ocorre quando o percurso de um pacote pela rede é interrompido devido ao próximo nó no caminho tornar seu rádio inativo – e reduzir o tempo de espera por inatividade, uma vez que se pretende manter um sincronismo para que um nó não fique ativo prematuramente, aguardando até que seu nó vizinho se ative. Estas soluções utilizam uma topologia em árvore, cuja raiz é o nó sorvedouro e os demais nós são ativados em instantes de tempo determinados de acordo com sua profundidade na árvore.

Diversos esquemas baseados em escalonamento foram propostos na literatura, entre eles o SPEED-MAC [Choi et al. 2010], que introduziu um período de sinalização para notificação de eventos e preparo dos nós para transmitirem os dados recebidos de comunicações prévias, e o CUPID [Kruger et al. 2010], que propõe um esquema bidirecional para o tráfego na árvore, apontando não só a existência de fluxo dos sensores para o nó sorvedouro, mas também um fluxo de configuração do nó sorvedouro para os sensores.

Uma dificuldade dos esquemas baseados em escalonamento é que, mesmo sendo menos restritivos em relação ao nível de sincronização dos nós como um todo (basta garantir um nível de sincronização entre nós vizinhos), ainda é necessária uma sincronização de relógio com alta resolução, o que vem associado à adição de *hardware* especializado e/ou alto *overhead* de sincronização. Além disso, no caso da entrada de novos nós na rede, estes deverão estar previamente sincronizados com o restante da rede (solução mais complexa) ou deverão entrar em um estado inicial de sincronização no qual seus rádios permanecerão constantemente ligados, induzindo um alto consumo energético até a sincronização.

2.2. Principais Mecanismos Assíncronos

Diferentemente dos mecanismos síncronos, os mecanismo assíncronos são menos complexos, em termos de adição de *hardware*, mensagens de controle e custo. Por esta razão, a tendência ao assincronismo tem ganhado espaço na comunidade científica, em especial aos mecanismos baseados em escalonamento. Porém, problemas com a latência e escuta ociosa ainda continuam como uma preocupação.

O mecanismo de amostragem de preâmbulo [Polastre et al. 2004] faz com que cada nó fique inativo de forma assíncrona, ativando-se periodicamente para escutar o meio. Se, ao escutar o meio, o nó detecta um preâmbulo, este permanece com seu rádio ligado para receber o quadro completo.

Já [Sun et al. 2008] classifica alguns mecanismos baseados na iniciativa do receptor. Nestes métodos, o transmissor aguarda um sinal do receptor, indicando que este está apto a receber o dado. Neste caso, a demanda energética é mais alta para o transmissor, que deve aguardar com seu rádio ligado até que o receptor acorde.

O *On-demand wakeup* [Schurgers et al. 2002] assume a existência de uma segunda interface de rádio de baixo consumo que permanece sempre ligada através da qual o nó receptor pode ser avisado de que deve ligar sua interface principal para a transmissão de dados.

O *duty cycling* aleatório [Paruchuri et al. 2004] adequa-se a RSSF densas. Os nós ligam e desligam seus rádios aleatoriamente, sem coordenação prévia. Quando um nó deseja transmitir um quadro, ele o faz independentemente de quais outros nós estejam ativos naquele momento. Se a rede for densa o suficiente, há alta probabilidade de que pelo menos um receptor receba o quadro transmitido. O método, portanto, assume que as transmissões na camada de enlace não são endereçadas a um próximo salto específico.

Finalmente, existem os mecanismos baseados em *schedules*, tais como *Block Design, Grid, Torus* e *Disco*, que são caracterizados por dividir o tempo em ciclos, por sua vez subdivididos em *slots* ativos e inativos. O grande diferencial destas propostas é que os padrões de *slots* ativos e inativos que compõem um ciclo são especialmente projetados

com o intuito de garantir ao menos uma sobreposição de *slots* ativos por ciclo entre quaisquer dois nós, independentemente do *offset* relativo entre seus respectivos relógios locais, uma característica conhecida na literatura como fechamento para rotação (do inglês, *rotation closure*) [Carrano et al. 2014a]. Ao contrário dos demais mecanismos assíncronos discutidos até aqui, estes mecanismos não assumem a existência de um segundo rádio de baixo consumo, nem requerem a escuta ao meio por um sinal ou uma grande densidade de nós.

Um *Block Design* [Zheng et al. 2003] $\{v, k, \lambda\}$ é uma entidade matemática composta por um conjunto V de v elementos inteiros, juntamente com uma família de v subconjuntos de V (chamados blocos), cada um com exatamente k elementos, tal que a interseção entre quaisquer dois blocos diferentes tenha exatamente λ elementos. No caso particular em que λ=1, os *Block Designs* recebem o nome de plano projetivo.

Um bloco de um *Block Design* pode ser mapeado para um padrão de *slots* ativos e inativos da seguinte forma. Cada *slot* do padrão corresponde a um elemento em V. Para cada elemento em V, se este pertence ao bloco, então ativa-se o *slot* correspondente no padrão. Caso contrário, o *slot* correspondente será inativo. A Figura 1 apresenta um exemplo com dois nós seguindo padrões baseados em blocos diferentes do *Block Design* $\{7,3,1\}$ onde temos V=$\{1,2,3,4,5,6,7\}$, k = 3, resultando nos seguintes blocos: $\{$[1,2,4], [2,3,5], [3,4,6], [4,5,7], [5,6,1], [6,7,2], [7,1,3]$\}$.

Note que os padrões gerados por cada bloco são simplesmente rotações dos padrões de outros blocos do mesmo *Block Design*. Logo, se dois nós de uma rede seguem um padrão resultante de um mesmo bloco, porém possuem seus relógios dessincronizados, isto é equivalente a cada nó utilizar um bloco diferente do mesmo *Block Design* sob uma base de tempo global. Com isso, os *Block Designs* garantem a existência de ao menos λ *slots* ativos coincidentes por ciclo, independentemente do *offset* relativo entre os relógios dos nós.

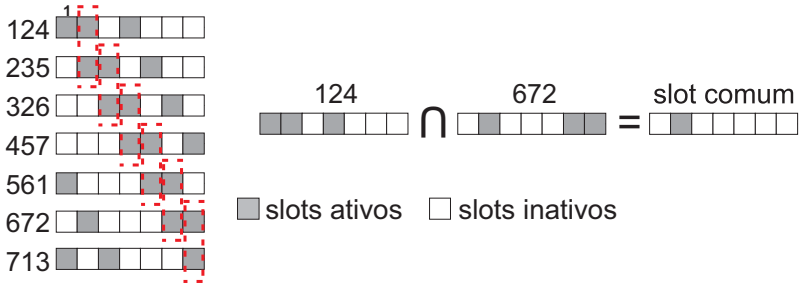


Figura 1. Mapeamento de cada bloco do *Block Design* $\{7, 3, 1\}$ para um padrão de *slots* ativos e inativos. À direita, ilustra-se o fechamento para rotação, com a interseção entre padrões de dois blocos distintos resultando em exatamente um *slot* em comum.

O *duty cycle* para um *Block Design* é dado por $\frac{k}{v}$. Em particular, é possível demonstrar que planos projetivos constituem os padrões de menor *duty cycle* com fechamento para rotação para um dado tamanho de ciclo [Maekawa 1985]. No entanto, o número de *Block Designs* conhecidos é limitado. Em particular, não são conhecidos planos projetivos com *duty cycles* menores que 1,03% [Link et al. 2011].

Em [Carrano et al. 2014a], é derivada uma expressão que aproxima o NDT (*i.e.*, o

tempo esperado até a comunicação efetiva entre dois nós vizinhos) para um *Block Design* $\{v, k, \lambda\}$ em função de p, a probabilidade de entrega de quadros do enlace:

$$E[NDT] = \frac{v+1}{p(\lambda+1)} - \frac{(v+1)(1-p)^{\lambda} - (\lambda+1)}{(\lambda+1)[(1-p)^{\lambda} - 1]} \quad (1)$$

O *Grid* [Jiang et al. 2005] é um dos mecanismos baseados em sistemas de quórum. O *Grid* define uma lei de formação de padrões de *slots* ativos e inativos que, assim como os *Block Designs*, garante o fechamento para rotação. A Figura 2 mostra um exemplo de um padrão gerado por um *Grid* 5x5. Cada nó seleciona uma linha e uma coluna em uma matriz n x n. Todos os *slots* da linha e da coluna selecionadas serão ativos, enquanto os demais *slots* serão inativos. O padrão resultante é obtido através da linearização da matriz, *i.e.*, o padrão é formado pela sequência de *slots* da primeira linha, seguidos dos *slots* da segunda e assim sucessivamente, resultando em um ciclo de n^2 *slots*. Mesmo o método funcionando com a escolha de linhas e colunas diferentes por cada nó, na prática comumente todos os nós escolhem as mesmas linha e coluna, resultando no mesmo padrão linearizado, embora, possivelmente, em *offsets* diferentes por conta da falta de sincronização, conforme ilustrado na Figura 2 (A e B seguem o mesmo padrão, mas com um *offset* relativo de um *slot*).

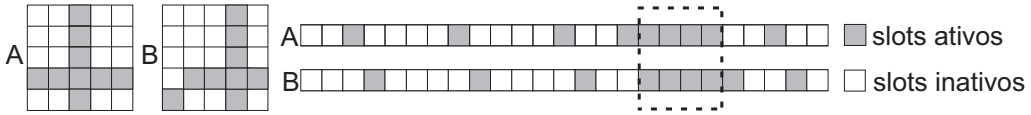


Figura 2. Dois nós A e B operando de acordo com o padrão gerado por um *Grid* de ordem 5, com um *offset* relativo de 1 *slot*. O padrão garante ao menos, dois *slots* ativos comuns em cada ciclo, independentemente do *offset* entre os nós.

O *duty cycle* no *Grid* é dado por $\frac{2n-1}{n^2}$, onde n é a ordem da matriz. A Equação 2, extraída de [Carrano et al. 2014a], fornece uma estimativa do NDT para um *Grid* de ordem n na comunicação por um enlace com probabilidade de entrega de quadros p:

$$E[NDT] = \frac{(3-p)n^2}{6p} \quad (2)$$

O *Torus* [Jiang et al. 2005], outro método da família dos sistemas de quórum, é uma proposta de melhoria do *Grid* em termos de *duty cycle*. Embora sua lei de formação de padrões seja similar à do *Grid*, ao invés de se ativarem todos os *slots* da linha selecionada, são ativados apenas metade mais um destes, resultando em um padrão de comprimento n^2, mas com menos *slots* ativos em comparação ao *Grid*. A Figura 3 exemplifica essa lei de formação, mostrando os *slots* ativos coincidentes para dois nós com *offset* relativo de um *slot*.

O *duty cycle* de um padrão resultante de um *Torus* de ordem n é dado por $\frac{3}{2n}$, para n par, ou $\frac{3n-1}{2n^2}$, para valores de n ímpares. Já o NDT estimado, segundo [Carrano et al. 2014a], é dado por:

$$E[NDT] = \frac{(2-p)n^2}{2p} \tag{3}$$

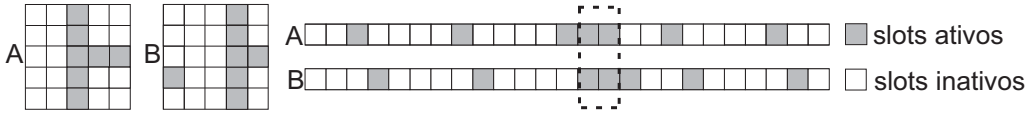


Figura 3. Dois nós A e B operando de acordo com o padrão gerado por um *Torus* de ordem 5, com um *offset* relativo de 1 *slot*. O padrão garante pelo menos um *slot* ativo em comum, independentemente do *offset* relativo entre os nós.

O *Disco* [Dutta and Culler 2008], um mecanismo baseado em números primos, garante que se os *slots* ativos de dois nós A e B são múltiplos de q_1 ou de q_2 (os números primos que são parâmetros do método), sempre ocorrerá uma sobreposição de *slots* ativos, independentemente dos *offsets* relativos de seus relógios locais.

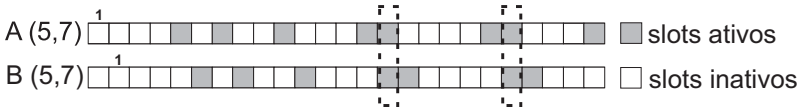


Figura 4. Dois nós A e B operando com o padrão gerado pelo mesmo Disco ($\{5, 7\}$) e com um *offset* relativo de 1 *slot* . O padrão garante a existência de *slots* ativos comuns independentemente do *offset* relativo entre os nós.

Segundo [Carrano et al. 2014a], o NDT resultante da utilização do *Disco* com os parâmetros q_1 e q_2 é aproximadamente dado por:

$$E[NDT] = \frac{q_1 q_2 (p^2 - 3p + 3)}{3p(2-p)} \tag{4}$$

3. Sincronização de *Slots* em Múltiplos Saltos

Como citado anteriormente, não é o escopo deste trabalho apresentar um protocolo de sincronização, mas analisar a possibilidade de ganho de desempenho na introdução de certo grau de sincronismo de baixa resolução em mecanismos assíncronos. A proposta se baseia na possibilidade de escalonar o uso do rádio de acordo com a distância de cada nó ao sorvedouro (em número de saltos) e não aleatoriamente, com base no momento em que o mesmo é ligado. Isso pode ser feito, por exemplo, através de adição da informação do *slot* de tempo atual de um nó quando este transmite seus pacotes de controle relativos ao próprio protocolo de roteamento. Ao receber um destes pacotes de um dado vizinho, o nó verificaria se este vizinho é, de acordo com as informações atuais de roteamento, seu próximo salto em direção ao sorvedouro. Em caso afirmativo, o nó alteraria seu *slot* de tempo atual para o *slot* atual do vizinho somado, possivelmente, de algum *offset* configurável. O *offset* irá estabelecer o deslocamento entre o padrão no qual um nó deverá operar em relação aos seus vizinhos. A distância em que cada nó se encontra do sorvedouro ou a própria árvore de menores caminhos entre os sensores e o sorvedouro são tipicamente obtidas dos protocolos de roteamento.

Neste trabalho, assumimos que um *slot* seja longo o suficiente para a transmissão de um quadro (incluindo a transmissão de um eventual *ACK* da camada de enlace, bem como outros tempos associados ao procedimento de acesso ao meio). Esta suposição é razoável, já que quanto menores os *slots*, menor é o comprimento de um ciclo do padrão de *duty cycle* o que, em uma escala de tempo absoluta, reduz a latência de comunicação. Uma consequência desta hipótese é que a perfeita sincronização dos padrões dos nós ao longo de um caminho de múltiplos saltos (*i.e.*, o uso de um *offset* = 0) não é uma boa escolha porque, ao receber um pacote para encaminhamento, um nó intermediário terá que esperar pelo próximo *slot* ativo em seu padrão, o que pode demorar a ocorrer. Na Figura 5, do lado esquerdo é mostrado um exemplo de um caminho de múltiplos saltos no qual todos os nós têm seus padrões de *duty cycle* perfeitamente alinhados (*i.e.*, *offset* = 0). Se o nó C estiver no *slot* quatro no momento da sua transmissão bem sucedida para o nó B, B precisará aguardar vários *slots* até que exista uma oportunidade de transmissão para A (*i.e.*, para que ambos estejam simultaneamente com seus rádios ligados). Por outro lado, se a transmissão bem sucedida de C para B se der no primeiro *slot* do padrão, B poderá realizar uma tentativa de encaminhamento do pacote logo no *slot* subsequente, já que este será um *slot* ativo para ambos B e C. Mesmo neste caso, no entanto, se existisse um terceiro salto neste caminho, (*e.g.*, um nó entre A e o sorvedouro), o nó A necessariamente teria que aguardar certa quantidade de *slots* para obter uma nova oportunidade de transmissão. Por fim, é importante notar que o *Block Design* $\{7, 3, 1\}$ possui quase 50% dos *slots* ativos, o que reduz o número de *slots* até a próxima oportunidade de transmissão. Se o padrão usado fosse o $\{9507, 98, 1\}$, com aproximadamente 1% dos *slots* ativos (um *duty cycle* muito mais adequado às aplicações atuais), a possibilidade de que um nó intermediário receba um pacote e possua uma oportunidade de transmissão logo no *slot* seguinte se torna bem mais baixa.

Ainda na Figura 5, do lado direito, é apresentado um esquema de escalonamento alternativo, no qual, ao invés de todos os nós operarem sob padrões perfeitamente alinhados, um *offset* de 1 *slot* é aplicado entre os padrões de nós vizinhos. O nó C, o mais distante do sorvedouro, ativa seus *slots* 1, 2 e 4 (de acordo com o *Block Design*$\{7, 3, 1\}$). Já o nó B, próximo salto de C em direção ao sorvedouro, tem seu padrão atrasado em um *slot* em relação a C, resultando no padrão $[1 + 1, 2 + 1, 4 + 1] = [2, 3, 5]$. Da mesma forma, o nó A precisa de um *offset* relativo a B de um *slot*, resultando no padrão $[2 + 1, 3 + 1, 5 + 1] = [3, 4, 6]$.

Este escalonamento garante que, quando um nó intermediário recebe um pacote para encaminhamento, este terá uma oportunidade de realizá-lo logo no *slot* subsequente. Se os enlaces da rede possuem alta probabilidade de entrega de quadros, este esquema mitiga o problema da interrupção do encaminhamento, reduzindo a latência de comunicação fim-a-fim.

Embora a Figura 5 ilustre o uso do *offset* = 1 para um padrão específico, a mesma análise pode ser realizada para qualquer padrão de *duty cycle* com fechamento para rotação. Qualquer padrão que apresente fechamento para rotação necessariamente possui dois *slots* consecutivos ativos (do contrário, dois nós com *offset* relativo igual a 1 nunca teriam *slots* ativos coincidentes). Logo, através da aplicação do *offset* = 1 entre vizinhos, sempre que um nó intermediário B recebe um pacote para encaminhamento, seu próximo *slot* será, necessariamente ativo. Como o próximo nó no caminho, digamos A, por sua

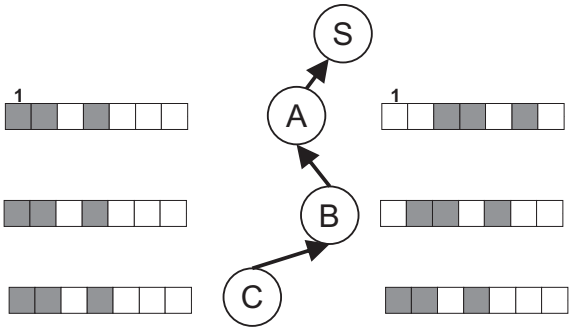


Figura 5. Exemplo de caminho de múltiplos saltos no qual cada nó opera segundo um padrão de *duty cycle* baseado no *Block Design* $\{7, 3, 1\}$. O nó S é o sorvedouro da rede. São ilustradas duas possíveis sincronizações: os padrões à esquerda possuem *offset* = 0, enquanto os padrões à direita têm um *offset* relativo de 1 *slot* entre vizinhos.

vez, também opera com um *offset* = 1 em relação a B, este também terá seu *slot* ativo. Este argumento pode ser facilmente estendido para um número arbitrário de saltos.

Em um caso extremo, se todos os enlaces do caminho têm probabilidade de entrega de quadros igual a 1, a latência fim-a-fim é dada pelo NDT do primeiro salto, mais um *slot* para cada salto subsequente. Em comparação, no caso de *offsets* aleatórios (*i.e.*, de uma rede sem qualquer sincronismo), a latência fim-a-fim se aproxima do produto do NDT pelo número de saltos do caminho.

Na prática, enlaces sem fio são sempre susceptíveis a algum nível de perda, tornando a expectativa de enlaces com a probabilidade de entrega de quadros igual a 1 irreal. Entretanto, como será mostrado nos resultados da nossa avaliação, mesmo para valores realísticos de probabilidade de entrega de bons enlaces sem fio, o uso do *offset* = 1 resulta em grande redução na latência fim-a-fim.

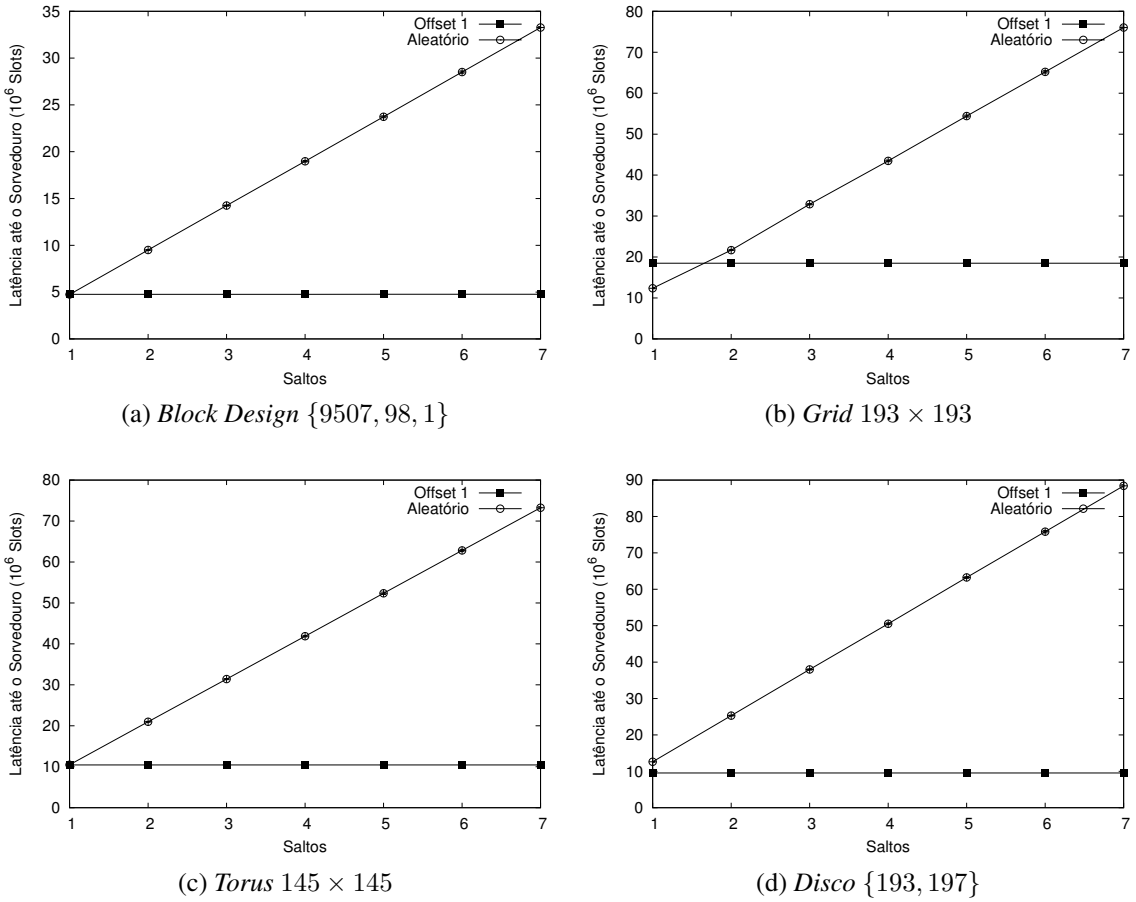
Por fim, é importante destacar as diferenças entre a presente proposta e mecanismos síncronos baseados em escalonamento, como o SPEED-MAC e o CUPID. Assim como estes dois mecanismos, nossa proposta também associa o escalonamento dos nós a suas distâncias em relação ao sorvedouro. No entanto, ao contrário do que ocorre nos mecanismos síncronos, o uso de esquemas assíncronos, como *Block Design*, *Grid*, *Torus* e *Disco*, simplifica a entrada de novos nós na rede, sem a necessidade de um sincronismo de relógio prévio ou de uma fase preliminar em que os novos nós fiquem com seus rádios permanentemente ligados. Um novo nó, ao entrar na rede, pode simplesmente seguir o padrão de *duty cycle* assíncrono, economizando energia e, ainda assim, com a garantia de que conseguirá se comunicar com seus vizinhos. Após algum tempo, através da troca de mensagens de controle com seus vizinhos, o nó realizará sua sincronização com os demais membros da rede, otimizando seu desempenho em relação a latência fim-a-fim.

4. Avaliação

Para avaliar a análise teórica apresentada na seção anterior, realizamos simulações numéricas comparando o funcionamento dos mecanismos assíncronos baseados em *schedules* com *offsets* aleatórios, resultantes da falta de sincronização de relógio entre os nós da rede, e o *offset* = 1 entre os vizinhos. Em mecanismos assíncronos baseados em *sche-*

dules, tais comparações se dão tradicionalmente em termos de métricas como consumo de energia e latência [Kandhalu et al. 2010, Link et al. 2011].

Foram realizadas simulações considerando comunicações de 1 a 7 saltos e diversos valores de probabilidade de entrega de quadros (variando de 0,05 a 1, com incrementos de 0,05). Para cada conjunto de parâmetros, as simulações foram repetidas 20000 vezes, permitindo a estimativa do NDT médio e o cálculo do intervalo de confiança de 95%. Embora nos resultados que se seguem os intervalos de confiança tenham sido plotados para todos os pontos de todos os gráficos, seus valores foram percentualmente muito baixos, dificultando sua visualização nas figuras. Foram comparados o *Block Design* $\{9507, 98, 1\}$, *Grid* 193x193, *Torus* 145x145 e *Disco* $\{193, 197\}$, todos resultando em *duty cycle* aproximado de 1,03%. O simulador utilizado foi o mesmo em [Carrano et al. 2013], estendido com a capacidade de simular comunicações em múltiplos saltos.



(a) *Block Design* $\{9507, 98, 1\}$ (b) *Grid* 193×193

(c) *Torus* 145×145 (d) *Disco* $\{193, 197\}$

Figura 6. Comparação da latência de comunicação obtida com *offset* aleatório e igual a 1 por padrões baseados em *Block Design*, *Grid*, *Torus* e *Disco* variando-se o número de saltos do caminho com p=1.

A Figura 6 mostra, para cada mecanismo de *duty cycle*, a latência de comunicação fim-a-fim como uma função do número de saltos do caminho para padrões com *duty cycles* de aproximadamente 1,03% e $p = 1$. Os gráficos mostram que o uso do *offset* = 1 resultou em menores latências para todas as comunicações de pelo menos dois saltos

em todos os quatro mecanismos avaliados. Tanto para o *offset* aleatório, quanto para o *offset* = 1, a latência fim-a-fim aumenta linearmente com o aumento do número de saltos. No entanto, a taxa de aumento é bem mais baixa para o *offset* = 1 (para este cenário, um aumento de apenas um *slot* por salto, conforme explicado na Seção 3). Para comunicações de um único salto, o *offset* = 1 resulta em latência superior em alguns dos mecanismos, já que embutido no caso aleatório está o *offset* = 0, que garante um número maior de oportunidades de encontro por ciclo. No entanto, à medida que os caminhos se tornam mais longos, a diferença percentual de latência entre os dois métodos se torna cada vez mais pronunciada.

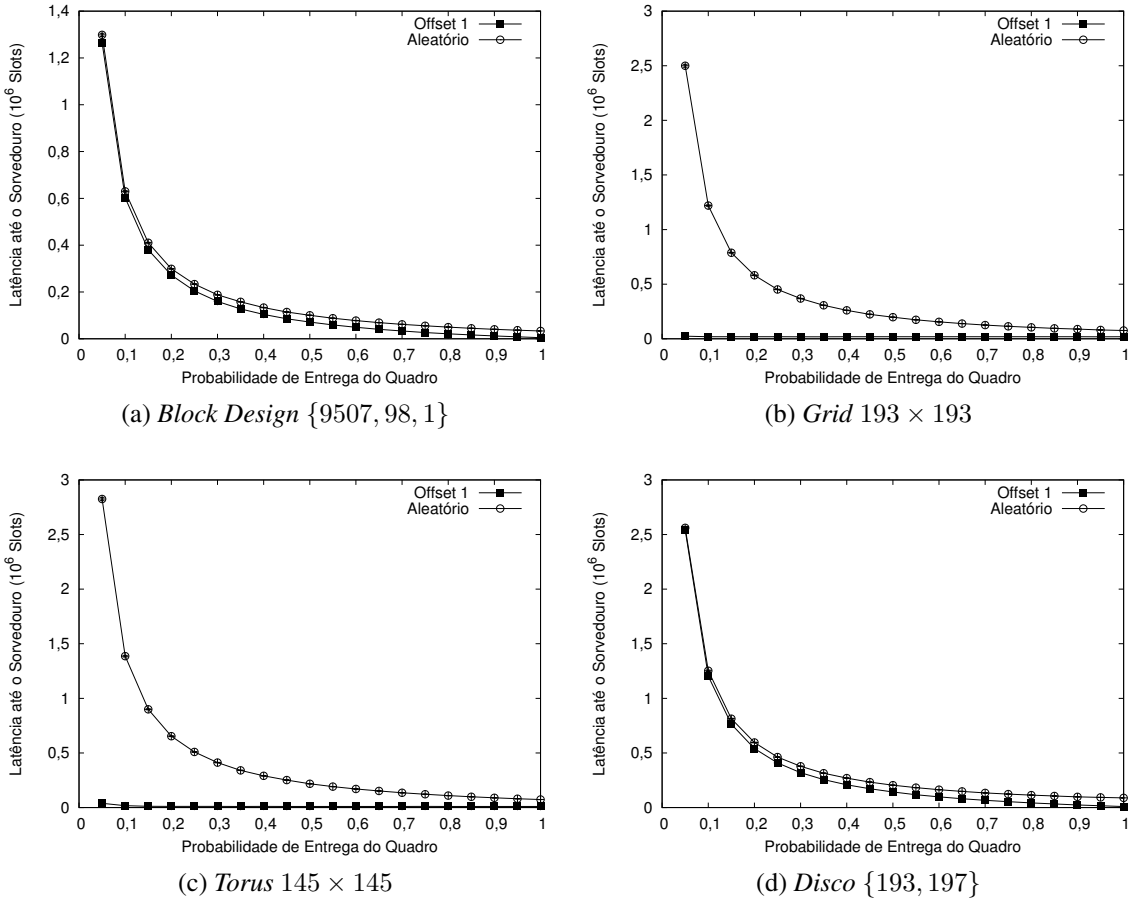
Já na Figura 7, são apresentados os resultados de latência fim-a-fim em função da probabilidade de entrega de quadros p, para uma comunicação em 7 saltos. É interessante notar que o *Grid* e o *Torus* apresentam uma latência quase linear para o *offset* = 1, uma vez que os dois métodos selecionam *slots* em linha (*i.e*, que se transformam em uma grande sequência de *slots* ativos no padrão linearizado). Quando a entrega de um quadro é feita com sucesso em um salto, o *slot* seguinte também será uma oportunidade de encontro no próximo salto. Embora isso seja válido para os quatro mecanismos estudados aqui (porque todos apresentam ao menos dois *slots* ativos consecutivos) no caso do *Grid* e do *Torus* com *offset* = 1 haverá até $n - 1$ oportunidades de encontro consecutivas. Ainda que a primeira tentativa não seja bem sucedida, os nós podem tentar novamente várias vezes em pouco tempo, mantendo a latência baixa mesmo quando p diminui. Este resultado é particularmente relevante porque a literatura acerca destes métodos consistentemente mostra que o *Block Design* é superior aos demais métodos em termos de NDT. No entanto, como ilustrado nestes resultados, no caso da latência fim-a-fim, *Grid* e *Torus* se mostraram a melhor opção para redes com enlaces de qualidade baixa, desde que o *offset* = 1 seja garantido entre os nós vizinhos. Já o *Block Design* e o *Disco* apresentam uma latência aumentando exponencialmente à medida que p cai, de forma similar ao que ocorre com o *offset* aleatório. Ainda assim, mesmo para estes mecanismos, o uso do *offset* = 1 resultou em redução na latência fim-a-fim para todos os valores de probabilidade avaliados.

5. Conclusão

Este artigo apresenta uma proposta para redução da latência de comunicação em múltiplos saltos dos mecanismos de *duty cycle* assíncrono baseados em *schedule*. Para tanto, consideramos a possibilidade de existência de um protocolo de sincronização dos nós em resolução de *slots* que permita que nós vizinhos sigam escalonamentos com *offsets* específicos. Este nível de sincronização pode ser obtido com baixo *overhead*, por exemplo, através da simples adição de um campo numérico nos pacotes de controle de um protocolo de roteamento informando o *slot* de tempo em que o nó atual se encontra. Com base nestas informações e no conhecimento do próximo salto em direção ao sorvedouro, nós podem alinhar seus padrões de *duty cycle*, possivelmente adicionando um *offset* entre eles.

Mostramos que o uso de um *offset* = 1 proporciona uma redução representativa da latência de comunicação em múltiplos saltos, ao mesmo tempo em que garante que novos nós podem ser facilmente adicionados à rede, sem a necessidade de um processo de sincronização prévia ou de uma fase inicial de sincronização na qual o novo nó é obrigado a permanecer com seu rádio ligado todo o tempo, resultando em desperdício de energia. Através de simulações numéricas, mostramos que, de fato, a introdução do *offset*

= 1 entre nós vizinhos resulta em reduções representativas de latência fim-a-fim. Estes ganhos se tornam ainda mais expressivos à medida que o número de saltos aumenta. Outro resultado interessante é o fato dos mecanismos *Grid* e *Torus* manterem latências quase lineares em relação à probabilidade de entrega de quadros dos enlaces na comunicação de múltiplos saltos com o uso do *offset* = 1, resultando em latências bem mais baixas que o *Block Design*, por exemplo (um mecanismo conhecidamente superior em termos de NDT [Carrano et al. 2014a]).



(a) *Block Design* $\{9507, 98, 1\}$ (b) *Grid* 193×193

(c) *Torus* 145×145 (d) *Disco* $\{193, 197\}$

Figura 7. Comparação da latência de comunicação para 7 saltos obtida com *offset* aleatório e igual a 1 por padrões baseados em *Block Design*, *Grid*, *Torus* e *Disco* variando-se a probabilidade de entrega do quadro (p).

Embora a avaliação tenha sido discutida sob a forma de simulações numéricas, futuramente pretende-se implementar este protocolo de sincronização de baixa resolução com escalonamento de *offsets* para corroborar os resultados obtidos e para permitir a avaliação do impacto do *overhead* de sincronização na comunicação de rede e no consumo energético dos nós. Também pretende-se aprofundar mais simulações e experimentos com outros valores fixos de *offset* entre nós vizinhos.

Referências

Anastasi, G., Conti, M., Di Francesco, M., and Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3):537–568.

Bachir, A., Dohler, M., Watteyne, T., and Leung, K. K. (2010). MAC essentials for wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 12(2):222–248.

Carrano, R. C., Passos, D., Magalhães, L. C., and Albuquerque, C. V. (2013). Nested block designs: Flexible and efficient schedule-based asynchronous duty cycling. *Computer Networks*, 57(17):3316–3326.

Carrano, R. C., Passos, D., Magalhães, L. C., and Albuquerque, C. V. (2014a). A comprehensive analysis on the use of schedule-based asynchronous duty cycling in wireless sensor networks. *Ad Hoc Networks*, 16:142–164.

Carrano, R. C., Passos, D., Magalhães, L. C., and Albuquerque, C. V. (2014b). Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 16(1):181–194.

Choi, L., Lee, S. H., and Jun, J.-A. (2010). SPEED-MAC: Speedy and energy efficient data delivery MAC protocol for real-time sensor network applications. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–6. IEEE.

Dutta, P. and Culler, D. (2008). Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 71–84. ACM.

Jiang, J.-R., Tseng, Y.-C., Hsu, C.-S., and Lai, T.-H. (2005). Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks. *Mobile Networks and Applications*, 10(1-2):169–181.

Kandhalu, A., Lakshmanan, K., and Rajkumar, R. R. (2010). U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 350–361. ACM.

Kruger, D., Pfisterer, D., and Fischer, S. (2010). CUPID-communication pattern informed duty cycling in sensor networks. In *2010 Fifth International Conference on Systems and Networks Communications*, pages 70–75. IEEE.

Link, J. Á. B., Wollgarten, C., Schupp, S., and Wehrle, K. (2011). Perfect difference sets for neighbor discovery: energy efficient and fair. In *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, page 5. ACM.

Lu, G., Krishnamachari, B., and Raghavendra, C. S. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 224. IEEE.

Maekawa, M. (1985). An algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computer Systems (TOCS)*, 3(2):145–159.

Paruchuri, V., Basavaraju, S., Durresi, A., Kannan, R., and Iyengar, S. S. (2004). Random asynchronous wakeup protocol for sensor networks. In *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pages 710–717. IEEE.

Polastre, J., Hill, J., and Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM.

Rajendran, V., Obraczka, K., and Garcia-Luna-Aceves, J. J. (2006). Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78.

Rowe, A., Mangharam, R., and Rajkumar, R. (2008). RT-link: A global time-synchronized link protocol for sensor networks. *Ad Hoc Networks*, 6(8):1201–1220.

Schurgers, C., Tsiatsis, V., Ganeriwal, S., and Srivastava, M. (2002). Optimizing sensor networks in the energy-latency-density design space. *IEEE transactions on mobile computing*, 1(1):70–80.

Sivrikaya, F. and Yener, B. (2004). Time synchronization in sensor networks: a survey. *IEEE network*, 18(4):45–50.

Sun, Y., Gurewitz, O., and Johnson, D. B. (2008). RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 1–14. ACM.

Tseng, Y.-C., Hsu, C.-S., and Hsieh, T.-Y. (2003). Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. *Computer Networks*, 43(3):317–337.

Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576. IEEE.

Zheng, R., Hou, J. C., and Sha, L. (2003). Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 35–45. ACM.