

Uma modelagem mais realista do tempo de descoberta de vizinhos para mecanismos de *duty cycle* assíncrono baseados em escalonamento

Beatriz Trabbold
Departamento de Engenharia de
Telecomunicações
Universidade Federal
Fluminense
Niterói, Brasil
beatriztrabbold@id.uff.br

Ricardo C. Carrano
Departamento de Engenharia de
Telecomunicações
Universidade Federal
Fluminense
Niterói, Brasil
ricardocarrano@id.uff.br

Diego Passos
Instituto Superior de Engenharia
de Lisboa, IPL
Lisboa, Portugal
Departamento de Ciência da
Computação
Universidade Federal
Fluminense
Niterói, Brasil
dpassos@ic.uff.br

Cledson de Oliveira
Departamento de Engenharia de
Telecomunicações
Universidade Federal
Fluminense
Niterói, Brasil
cledsons@id.uff.br

Resumo— O ciclo de trabalho é um mecanismo fundamental para redes sem fio com dispositivos operados com bateria, como as redes de sensores sem fio. Devido à sua importância, este tipo de mecanismo é empregado em diversos protocolos de acesso ao meio. Nos ciclos de trabalho assíncronos baseados em escalonamento, os nós ativam e desativam suas interfaces de rádio de acordo com um escalonamento de *slots* preconcebido que garante alguma sobreposição de tempo de atividade entre dois nós, independente da diferença entre seus relógios internos, o que torna a comunicação possível. Este artigo apresenta um novo modelo para o Tempo de Descoberta de Vizinho (ou NDT, do inglês *Neighbor Discovery Time*). Diferente dos modelos anteriores, este trabalho modela o cenário onde as bordas dos *slots* dos escalonamentos de nós vizinhos podem não estar perfeitamente alinhadas – o caso mais comum na prática.

Palavras Chave — *duty cycle*, redes sem fio, tempo de descoberta do vizinho

I. INTRODUÇÃO

A eficiência energética é fundamental para a maioria das redes *ad hoc*. Em redes de sensores sem fio (RSSFs), que são redes que possuem diversos nós sensores distribuídos com o objetivo de adquirir dados do ambiente e/ou detectar a ocorrência de eventos, por exemplo, os nós tipicamente operam sob severas restrições energéticas, tendo, muitas vezes, sua vida útil limitada à duração de sua bateria [1]. Como o rádio costuma ser responsável pela maior porcentagem de gasto energético do sistema, o ciclo de trabalho de sua interface pode ser considerado uma necessidade [2].

O objetivo principal do ciclo de trabalho é reduzir tempo da escuta ociosa, que ocorre quando um rádio espera em vão por um quadro. Manter o rádio ligado apenas quando necessário é um desafio, visto que não é possível determinar, *a priori*, quando um quadro será recebido. Quando o ciclo de trabalho é utilizado, existe a necessidade de um mecanismo que garanta que os nós, enquanto ativos, encontrem seus vizinhos também ativos para que surja uma oportunidade de encontro. Esta coordenação pode ser alcançada com a utilização de mecanismos específicos síncronos ou assíncronos. Os mecanismos de ciclo de trabalho assíncronos,

foco deste artigo, possuem *overhead* de comunicação menor se comparados aos mecanismos síncronos [2].

Em particular, os esquemas assíncronos baseados em escalonamento dependem de um escalonamento para despertar os nós com propriedades particulares. Para que a comunicação seja possível, tal escalonamento deve garantir que quaisquer dois nós terão momentos de sobreposição com seus rádios ativos, independentemente de eventuais diferenças (*offsets*) entre seus respectivos relógios locais. Uma desvantagem do ciclo de trabalho assíncrono baseado em escalonamento em relação a outras classes de solução é o aumento da latência na comunicação, tendo em vista a necessidade de aguardar que o próximo nó se torne ativo – algo condicionado ao escalonamento – para que a comunicação seja efetivada. Assim, modelos precisos para prever a latência de comunicação sob esses escalonamentos são ferramentas fundamentais para a operação das redes de sensores, de forma a garantir que os requisitos da aplicação particular sejam alcançados.

Neste artigo, discutimos as limitações e imprecisões dos modelos atualmente adotados na literatura para este propósito. Mostramos, particularmente, que tais modelos não levam em consideração o fenômeno de escorregamento de relógio, não modelando, por exemplo, a falta de alinhamento das bordas dos *slots* dos escalonamentos que geralmente ocorre na prática. Para lidar com isso, propomos uma modelagem alternativa que leva este aspecto em consideração. Através de simulações, mostramos que esta modelagem fornece previsões significativamente mais precisas em comparação aos modelos tradicionais.

O restante deste trabalho está organizado da seguinte forma. A Seção II apresenta uma revisão de conceitos teóricos importantes para o restante do trabalho. A Seção III apresenta o histórico de pesquisas relacionadas que serviram como base para esse artigo. A Seção IV explica o modelo proposto. A Seção V traz os resultados de simulações do modelo e, por fim, são apresentadas nossas considerações finais e ideias para trabalhos futuros.

II. FUNDAMENTAÇÃO TEÓRICA

Existem três classes de modelos de ciclo de trabalho [2]: síncronos, onde os nós mantêm uma referência de tempo em comum; assíncronos, quando esta referência não é necessária; ou híbridos, caso em que a rede é dividida em *clusters* que são sincronizados internamente, enquanto a comunicação entre os *clusters* é realizada de forma assíncrona.

A sincronização de nós em uma rede de sensores sem fio pode ser considerada difícil e cara, exigindo *hardware* extra, altas capacidades de processamento ou até mesmo a adição de tráfego de controle frequente, que consumiria ainda mais energia [3]. Nos sistemas assíncronos, os nós dividem o tempo em ciclos, subdivididos em *slots* (ativos ou inativos), de acordo com um escalonamento selecionado, onde cada ciclo é uma repetição do anterior, como ilustrado na Figura 1.

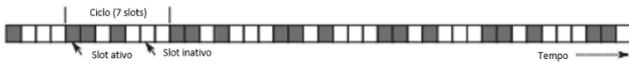


Figura 1 - Exemplo de escalonamento com ciclos de 7 slots, onde três destes slots estão ativos. Adaptado de [3].

A. Mecanismos assíncronos de ciclo de trabalho

Mecanismos assíncronos de ciclo de trabalho diferem entre si pela forma como buscam reduzir a latência e a escuta ociosa. Em [4] é introduzido o mecanismo de amostragem de preâmbulo, onde os nós dormem de forma assíncrona e acordam periodicamente para verificar se existe alguma atividade no canal. Se um preâmbulo, cuja duração é maior do que o tempo de atividade e de sono juntos, for identificado durante o período de verificação, o nó permanece ativo aguardando o quadro a ser recebido. Caso contrário, o nó volta a dormir.

Em [5] é apresentado o mecanismo baseado em iniciativa do receptor. Nesta categoria, ao invés de sinalizar que existem dados para serem transferidos, o remetente espera por um *beacon* do receptor informando que ele está ativo e transmite o quadro apenas depois que este *beacon* é identificado. Este método substitui o preâmbulo longo por *beacons* periódicos.

A ativação sob demanda [6] é um mecanismo baseado na ideia de um rádio despertador, que é um rádio de baixa potência que escuta um sinal de despertar e envia uma mensagem para a CPU que, por sua vez, ativa o rádio primário.

No ciclo de trabalho aleatório [7], a ideia é que, em implantações densas, os nós possam dormir e acordar aleatoriamente, uma vez que há uma grande probabilidade de que existam nós ativos para comunicação a qualquer momento.

Por fim, os mecanismos baseados em escalonamento, que são estudados neste artigo, são baseados na concepção de horários de despertar (ativos) e de sono (inativos), que devem sempre garantir a sobreposição de *slots* ativos entre vizinhos, o que permite a comunicação.

B. NDT – Neighbor Discovery Time

Dados dois nós (A e B) operando sob um esquema de ciclo de trabalho assíncrono, o tempo de descoberta do vizinho (ou NDT) é definido como o número de *slots* de tempo decorridos entre o momento em que o nó A decide enviar um quadro para

um nó vizinho B e o momento em que o nó B recebe o quadro enviado com sucesso [3].

C. Block Designs

Dentre os vários métodos existentes na literatura para geração de escalonamentos para ciclo de trabalho assíncrono, os *Block Designs* [8] são amplamente considerados os de melhor desempenho em termos de NDT. Genericamente, *Block Designs* são esquemas combinatórios, *i.e.*, formas de se selecionar ou combinar elementos de um determinado conjunto respeitando-se determinadas restrições. Dado um conjunto finito V de v elementos e inteiros $k, \lambda \geq 1$, um *Block Design* representado por $\{v, k, \lambda\}$ terá v blocos (subconjuntos de V) de k elementos e as seguintes propriedades [8]:

- 1) Todo e qualquer elemento de V ocorre exatamente em k blocos;
- 2) Quaisquer dois blocos diferentes terão exatamente λ elementos em comum.

Em aplicações de ciclo de trabalho assíncrono, os blocos do *Block Design* correspondem aos escalonamentos cíclicos dos nós, sendo cada elemento do bloco um *slot* de tempo ativo.

Como exemplo, suponha-se um *Block Design* $\{7, 3, 1\}$, que consiste em sete blocos com três elementos cada. Isso é ilustrado na Figura 2, que mostra ainda dois nós (A e B) operando em dois blocos diferentes ($\{1,2,4\}$ e $\{0,4,5\}$) com uma interseção no *slot* 4, o que permite a comunicação. Note que um bloco corresponde a um escalonamento e os demais blocos de um mesmo *Block Design* são apenas rotações do mesmo escalonamento. Com isso, se dois nós operam sob um mesmo escalonamento derivado de um mesmo bloco de um *Block Design* com qualquer *offset* relativo diferente de 0 entre seus relógios internos, isso pode ser visto como cada nó operando sob escalonamentos derivados de blocos distintos do *Block Design* [3]. Nós com um *offset* entre seus relógios internos diferente de zero têm um número fixo (λ) de *slots* ativos comuns em cada ciclo. Esta propriedade garante que quaisquer dois nós apresentem *slots* ativos em comum, mesmo sob condição de assincronismo. O ciclo de trabalho em um *Block Design* é dado por k/v , o que corresponde a 43% no caso deste exemplo, embora existam *Block Designs* com ciclos de trabalho arbitrariamente baixos.

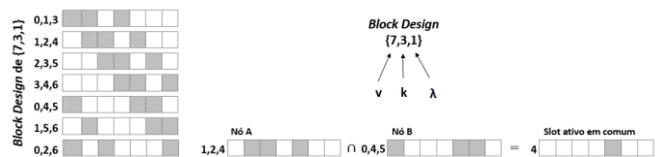


Figura 2 - Exemplo de Block Design onde ocorre interseção de slots. Adaptado de [3].

Até o momento, a literatura sobre redes considerou principalmente, se não exclusivamente, uma categoria particular de *Block Design* chamada de Plano Projetivo. Um Plano projetivo é um *Block Design* que assume a forma $\{s^2+s+1, s+1, 1\}$, onde o parâmetro s é chamado de ordem. Conjectura-se que só existem planos projetivos para valores de s que são potências de primos [3]. Além disso, pode-se constatar que dois nós operando sob um design $\{v, k, \lambda\}$ terão sobreposição de *slots* de tempo ativos, mesmo que suas bordas de slot não estejam alinhadas [8].

A Tabela 1 apresenta alguns exemplos de *Block Designs*, fornecendo os ciclos de trabalho resultantes e a lista de slots ativos.

Tabela 1- Exemplos de Block Design com seu respectivo ciclo de trabalho e slots ativos.

Design	Ciclo de Trabalho (%)	Slots ativos
{7, 3, 1}	42,86	0, 1, 3
{183, 14, 1}	7,65	0, 12, 19, 20, 22, 43, 60, 71, 76, 85, 89, 115, 121, 168
{9507, 98, 1}	1,03	0, 1, 3, 37, 52, 191, 308, 332, 433, 914, 919, 984, 1093, 1155, 1231, 1238, 1600, 1678, 1723, 1732, 1755, 1773, 1826, 1930, 1938, 2099, 2116, 2141, 2457, 2712, 2859, 3058, 3187, 3466, 3524, 3655, 3675, 3748, 4139, 4145, 4183, 4297, 4301, 4518, 4528, 4600, 4720, 4777, 4964, 5043, 5054, 5176, 5268, 5329, 5356, 5496, 5526, 5601, 5617, 5851, 6151, 6173, 6491, 6539, 6759, 6778, 6792, 6878, 7021, 7163, 7226, 7290, 7490, 7650, 7747, 7860, 7941, 8028, 8056, 8154, 8304, 8339, 8370, 8438, 8450, 8505, 8534, 8574, 8797, 9005, 9048, 9094, 9107, 9133, 9154, 9270, 9326, 9400

D. Escorregamento de Relógio

Tipicamente em redes de sensores não se pode contar com um relógio central globalmente acordado. Ao contrário, cada nó utiliza seu próprio relógio físico para ditar seu escalonamento. O escorregamento de relógio é um fenômeno no qual o relógio de um nó dessincroniza gradualmente de um hipotético referencial. Todos os relógios estão sujeitos ao escorregamento e o fenômeno afeta nós diferentes de maneira aleatória, causando eventuais divergências, a menos que sejam resincronizados [9].

A maioria dos nós em redes de sensores sem fio utiliza cristais ou osciladores cuja taxa máxima de escorregamento é de 30 ppm [10], o que corresponde a um erro de 30 μ segundos a cada 1 segundo. Ao contrário dos protocolos com ciclos de trabalho síncronos, que utilizam mensagens de sincronização para ajustar os relógios dessincronizados, o escorregamento do relógio em protocolos que possuem ciclo de trabalho assíncrono causa estimativa imprecisa do momento de despertar do receptor, o que pode alterar a relação de *slots* ativos coincidentes entre dois nós. O escorregamento também torna bastante improvável que as bordas dos *slots* dos escalonamentos de dois nós diferentes estejam alinhadas, sendo mais provável que o início do *slot* de um nó coincida com um período intermediário do *slot* do outro nó.

III. TRABALHOS RELACIONADOS

Modelos para a predição do desempenho de ciclos de trabalho assíncronos baseados em escalonamento têm sido usados com frequência na literatura, particularmente em trabalhos que propõem métodos de construção de escalonamentos. Por exemplo, em [8], trabalho que propõe *Block Designs* como escalonamentos, os autores utilizaram um modelo simples baseado em comprimento do escalonamento. Em outras palavras, o NDT entre dois nós era aproximado pelo número de *slots* do escalonamento avaliado. Sob esta perspectiva, *Block Designs* são ótimos [8].

Em [11], foi apresentado um modelo mais preciso para determinar o NDT. Ele incorporava a probabilidade de perda de pacotes, mostrando que o NDT aumenta exponencialmente à medida que a qualidade do enlace se degrada. Além disso, este modelo era baseado na análise do caso médio quando considerados todos os possíveis *offsets* entre os dois nós e possibilidades de *slot* inicial. Uma limitação deste modelo, no entanto, era assumir apenas *offsets* inteiros, *i.e.*, que as bordas dos *slots* de nós vizinhos estão sempre alinhadas.

Com base no modelo proposto em [11], expressões fechadas aproximadas para o NDT de vários métodos de ciclo de trabalho assíncrono baseados em escalonamento são apresentados em [3]. Mais especificamente, são apresentadas expressões para o *Block Design*, o Grid, o Torus e o Disco. As expressões são derivadas para um cenário com dois nós (A e B) operando sob o mesmo escalonamento cíclico, formado por um certo número de *slots* e levando em consideração a falta de sincronismo entre os nós. Assume-se que, em cada *slot* ativo, A e B acordam, transmitem um *beacon*, e aguardam *beacons* de seus vizinhos até o final do *slot*. As expressões, portanto, estimam o número esperado de *slots* que decorrerão até que um nó escute o *beacon* do vizinho. Por se basear no modelo de [11], as expressões não levam em conta possíveis desalinhamentos entre os *slots* de A e B.

Em [14] é apresentado um estudo sobre a descoberta assíncrona de nós vizinhos em dispositivos móveis que funcionam com ciclo de trabalho. Ele mostra que escalonamentos com *slots* de tamanho fixo podem levar a um desperdício significativo de recursos e, para resolver este problema, é proposto o uso *slots* de duração arbitrária, onde o tempo é contínuo e um nó pode ficar acordado ou adormecido em qualquer ponto do tempo. Como os *slots* não têm mais duração fixa, modelos como o proposto em [11] já não se aplicam ao estudo deste tipo de escalonamento.

IV. MODELO PROPOSTO

Ao contrário dos modelos propostos anteriormente, o modelo apresentado neste artigo considera um dos aspectos do escorregamento do relógio, o desalinhamento dos *slots*, o que torna seus resultados mais precisos.

Assim como em [11], no modelo aqui apresentado, são considerados dois nós (A e B) operando sob o mesmo escalonamento cíclico, formado por um certo número de *slots*. Assume-se que os *slots* sejam longos o suficiente para a transmissão e recepção de ao menos um *beacon* e que os nós possam apresentar um *offset* aleatório entre seus escalonamentos. Uma vez que A decide enviar um quadro para seu vizinho, ele tentará transmitir um *beacon* uma vez a cada um de seus *slots* ativos, até que alcance o sucesso (assumimos que A é notificado em caso de sucesso, por exemplo, através de um *ack*). Uma transmissão só é bem-sucedida caso A e B estejam em *slots* ativos simultaneamente, o que é chamado de *oportunidade de encontro*. Além disso, assume-se que o enlace sem fio é caracterizado por uma probabilidade de recepção de quadro p . Essa probabilidade engloba fatores como a potência de transmissão de A, ruído/interferência ambiental, largura de banda, modulação, codificação e eventuais colisões. Assim, mesmo em uma oportunidade de encontro, uma tentativa de transmissão pode falhar com probabilidade $1 - p$ [3].

A proposta aqui apresentada se diferencia do modelo proposto em [11], pois considera a possibilidade de

desalinhamento entre os *slots*. Isso é feito através da análise dos escalonamentos em uma unidade de *subslots* – ou fatias – ao invés de *slots*. O número de *subslots* de cada *slot* é um parâmetro do modelo proposto e pode variar. Na Seção V deste artigo, são apresentados resultados de experimentos realizados com divisões de 2, 4, 32 e 64 *subslots*.

O modelo assume que, no início de cada *slot* ativo (ou seja, no seu primeiro *subslot*), um nó transmite um *beacon* e passa o resto do seu tempo ativo aguardando por *beacons* dos nós vizinhos. Também se assume que, ao ouvir um preâmbulo válido, um nó permanece com seu rádio ligado para receber o quadro correspondente até o final, independentemente de seu escalonamento. No modelo aqui proposto, considera-se que só há uma oportunidade de encontro entre dois nós se (i) ambos estiverem em *subslots* ativos (para terem seus rádios ligados simultaneamente) e (ii) se um deles estiver no primeiro *subslot* de seu respectivo *slot* (para que transmita um *beacon*). Ainda assim, como em [11], o modelo considera que existe uma probabilidade p de sucesso na recepção de um *beacon* enviado de A para B ou de B para A. Desta forma, mesmo que haja uma oportunidade de encontro em determinado momento, o encontro efetivo de A e B pode não ocorrer com probabilidade $(1-p)$. Caso p seja zero, o NDT tende ao infinito, pois a troca de informações não é possível pelo enlace.

Na Figura 3, é apresentado um exemplo de funcionamento do modelo proposto para o escalonamento dado pelo *Block Design* $\{7, 3, 1\}$, com cada *slot* dividido em dois *subslots*. No exemplo, há um *offset* entre os relógios internos de A e B de 1,5 *slot* (o que equivale a 3 *subslots*). Os quadrados com borda tracejada indicam o primeiro *subslot* de um *slot* – portanto, onde ocorrem as transmissões de *beacon* pelos nós. A terceira linha mostra em quais *subslots* ocorrem as oportunidades de encontro. Como pode ser visto, a transmissão ocorre a cada *subslot* ativo de A que corresponda ao primeiro *subslot* de um *slot* de B – ou vice-versa. Neste exemplo, a primeira oportunidade de encontro ocorre em um *subslot* em que o nó B inicia a transmissão do seu *beacon*, pois trata-se do primeiro *subslot* de um *slot* ativo. Por sua vez, A está no segundo *subslot* de um *slot* ativo. Já na segunda oportunidade, A inicia a transmissão de um *beacon*, pois está no primeiro *subslot* de um de seus *slots* ativos, enquanto o *subslot* ativo de B não é o primeiro do *slot* correspondente.

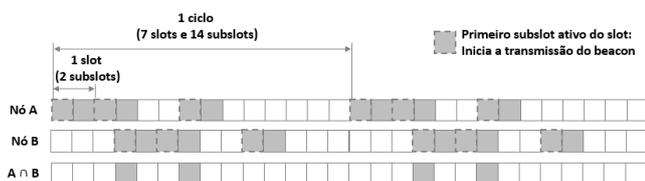


Figura 3 - Exemplo de funcionamento do modelo proposto para o escalonamento com o *Block Design* $\{7, 3, 1\}$, divisão em 2 *subslots* e *offset* de 1,5 *slot*.

A. Características do modelo

A Figura 4 apresenta um segundo exemplo do modelo proposto, mas agora considerando um *offset* de 2 *slots* (4 *subslots*). É possível verificar que, neste caso, é prevista uma única oportunidade de encontro por ciclo. Este valor é consistente com o que se espera de um escalonamento baseado em *Block Designs* (particularmente, planos projetivos), dado que estes garantem exatamente um *slot* coincidente por ciclo para qualquer *offset* que não seja múltiplo de v .

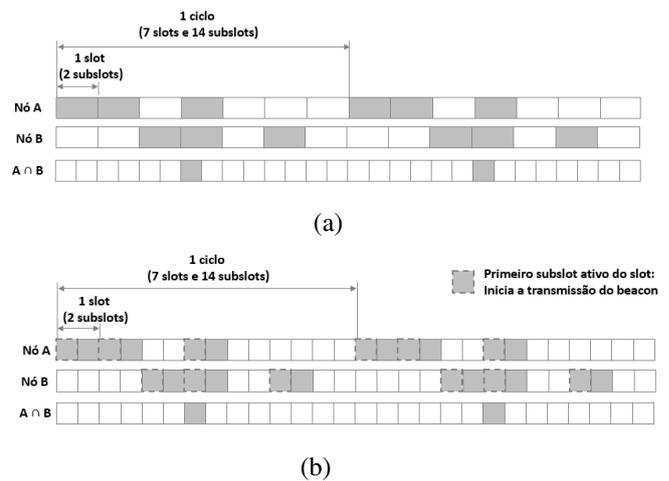


Figura 4 - Exemplo de funcionamento do modelo proposto para o escalonamento com o *Block Design* $\{7, 3, 1\}$, divisão em 2 *subslots* e *offset* de 2 *slots*: (a) visualização sem quebra de *subslots*, (b) visualização com quebra de *subslots*.

A título de comparação, a Figura 5 ilustra o modelo proposto em [11] considerando um *offset* de 2 *slots*. O modelo de [11] não considera a subdivisão dos *slots* em *subslots* e, portanto, assume que os *slots* de A e B estão sempre alinhados. Conforme ilustrado na figura, o número de oportunidades de encontro previstas por este modelo é o mesmo do caso da Figura 4: uma oportunidade por ciclo.

Por outro lado, no exemplo da Figura 3 foi considerado um *offset* de 1,5 *slot*. Este valor não-inteiro de *offset* significa que os *slots* de A e B não se encontram perfeitamente alinhados: ao contrário, os *slots* de A começam exatamente no meio dos *slots* de B. Conforme mostrado na figura, para este caso, o modelo proposto prevê duas oportunidades de encontro a cada ciclo do escalonamento, ao invés de uma única, como no caso de um *offset* inteiro. Vale ressaltar que a duração do *slot* é um parâmetro variável e cabe ao projetista da rede fazer este dimensionamento para que seja possível enviar um *beacon*.

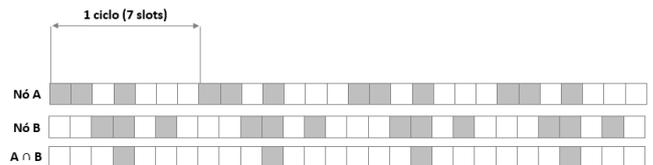


Figura 5 - Exemplo de funcionamento do modelo proposto em [2] para o escalonamento com *Block Design* de $\{7, 3, 1\}$ e *offset* de 2 *slots*.

De fato, um *offset* não inteiro x pode ser entendido como uma superposição entre os *offsets* $\lfloor x \rfloor$ e $\lceil x \rceil$: se considerarmos a interseção entre os *slots* ativos de A e B, há sobreposição parcial dos *slots* que seriam comuns para o *offset* $\lfloor x \rfloor$ e também dos *slots* que seriam comuns para o *offset* $\lceil x \rceil$. Esta característica dá origem a uma quantidade maior de oportunidades de encontro (para *Block Designs*, o dobro) justamente quando os *slots* de A e B não estão perfeitamente alinhados. Note, ainda, que este é o caso mais provável.

Desta forma, conclui-se que, ao não levar em conta a possibilidade de *slots* desalinhados, os modelos existentes na literatura subestimam a frequência de ocorrência de oportunidades de encontro e, portanto, superestimam o NDT. Ao contrário, o modelo proposto neste trabalho fornece uma

previsão mais precisa do NDT por contabilizar também estes casos.

Além da possibilidade de desalinhamento de *slots*, outro aspecto que não é levado em conta em modelos de NDT na literatura é o efeito do atraso de propagação. Se, em uma dada rede, o atraso de propagação dos enlaces for da ordem de um ou mais *slots*, então a hipótese de que as oportunidades de encontro ocorrem quando A e B têm seus rádios ligados simultaneamente se torna falsa – já que o sinal correspondente ao *beacon* do transmissor só chegaria ao receptor depois de um tempo correspondente a um ou mais *slots*.

Embora o atraso de propagação não seja um parâmetro explícito do modelo proposto neste artigo, seu efeito é implicitamente capturado. Isso ocorre porque o efeito do atraso de propagação pode ser contabilizado no próprio *offset* entre os nós. Por exemplo, se o *offset* entre os relógios internos de A e B é de 5 *slots* e o atraso de propagação é de 2 *slots*, então, do ponto de vista das oportunidades de encontro, o cenário se torna equivalente ao caso de um *offset* de 7 *slots* com um atraso de propagação nulo. Embora um raciocínio similar possa ser feito para os modelos existentes na literatura, a capacidade do modelo aqui proposto de lidar com *offsets* não-inteiros o torna também mais adequado para capturar os efeitos do atraso de propagação (que, em geral, não será um múltiplo inteiro da duração dos *slots*).

Por fim, deve-se notar que a ideia de subdividir os *slots* em *subslots* tem como objetivo aproximar a modelagem do funcionamento dos escalonamentos no mundo real – onde o tempo é contínuo. Quanto maior o número de *subslots* nos quais cada *slot* é fatiado, maior a resolução do modelo em representar este tempo contínuo. Desta forma, acredita-se que, quanto maior o número de fatias utilizado, mais realístico e preciso será o modelo. No limite, quando o número de fatias tende a infinito, o modelo seria uma representação fiel em termos deste aspecto do sistema.

V. ANÁLISE NUMÉRICA DO MODELO

Para analisar o modelo proposto, foi desenvolvido um código em Python¹ que simula a comunicação de dois nós seguindo um determinado escalonamento e retorna medidas de NDT máximo, NDT médio e NDT mínimo. Como entrada, o simulador recebe o escalonamento a ser analisado, a probabilidade de sucesso do enlace simulado, o número de repetições desejadas e o número de fatias de cada *slot*. São, então, realizadas tantas repetições quanto solicitadas de simulações do processo de encontro entre dois nós A e B. Em cada repetição, o simulador sorteia aleatoriamente *slots* iniciais para A e B e simula os escalonamentos e transmissões de *beacons* — considerando a probabilidade de sucesso p especificada — até que haja um sucesso. Quando isto ocorre, o número de *slots* simulados para aquela repetição é armazenado para que, posteriormente, estatísticas sejam calculadas.

Foram realizadas simulações com os *Block Designs* {7, 3, 1} e {183, 14, 1}. As simulações realizadas variaram o número de *subslots* por *slot* e a probabilidade de sucesso do enlace (p). Foram considerados cenários com 1, 2, 4, 16, 32 e 64 *subslots* por *slot* e $p \in \{0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 1,0\}$. Para cada combinação de parâmetros, foram realizadas 5.000 repetições.

A primeira análise realizada buscou verificar como o NDT médio se comportava ao variar a probabilidade de sucesso na transmissão p para diferentes quantidades de *subslots*. Como pode ser visto na Figura 6, independente do escalonamento escolhido, o comportamento das curvas é semelhante. Para $p \ll 1$, à medida que o número de *subslots* por *slot* aumenta, o NDT tende a cair pela metade em relação ao caso de uma fatia. Isso é consistente com a observação feita na Seção IV de que o número de oportunidades de encontro aumenta – em particular, dobra – para *offsets* não inteiros: com mais fatias, a chance de *offsets* não-inteiros cresce e o NDT cai. Em particular, para valores mais baixos de p , são tipicamente necessárias várias tentativas de transmissão de *beacon* até o sucesso, causando uma influência direta entre o número de oportunidades de encontro por ciclo e o NDT. Por outro lado, para p próximo de 1, o NDT cai com o aumento do número de fatias, porém não pela metade. Isso ocorre porque o tempo até a primeira oportunidade de encontro se torna mais relevante e este pode não ser diretamente relacionado ao número de oportunidades por ciclo.

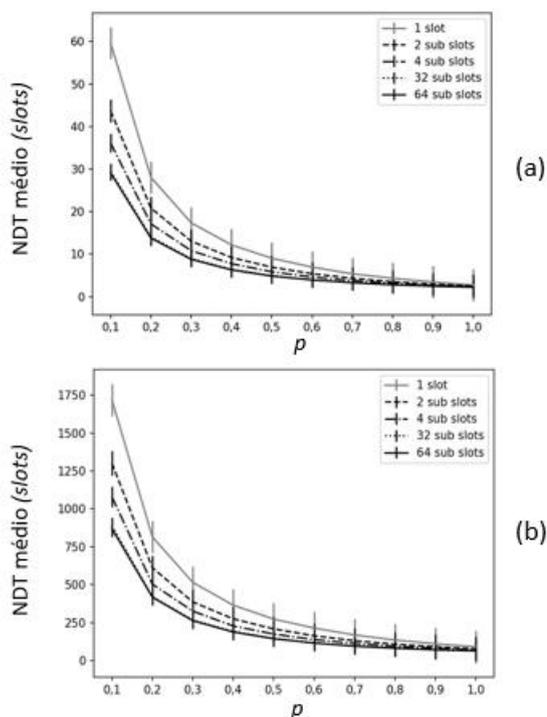


Figura 6 - NDT médio pela probabilidade de sucesso (p) para os Block Designs de (a) {7, 3, 1} e (b) {183, 14, 1}.

Para entender melhor o efeito do número de fatias no modelo, a segunda análise realizada buscou relacionar este parâmetro com o NDT médio para os escalonamentos {7, 3, 1} e {183, 14, 1}, como mostrado na Figura 7. Para esta análise foi considerada uma probabilidade de sucesso de 0,1. Observa-se que o comportamento qualitativo das curvas é idêntico para os dois escalonamentos estudados, apesar da diferença na escala. Além disso, essa análise confirma o que o valor do NDT médio cai com o aumento do número de fatias — aproximadamente a metade do valor previsto para 1 fatia quando o número de fatias tende ao infinito. Através destas

¹ O código pode ser encontrado em <https://github.com/beatriztrabb/NDT>

observações, conjecturou-se que, para valores baixos de p , o NDT pode ser aproximado pela função:

$$g(f) = (f + 1) \times \frac{K}{(2 \times f)},$$

onde f é o número de fatias e K é o valor do NDT médio para uma fatia. Esta função também está plotada nos gráficos da Figura 7 para comparação.

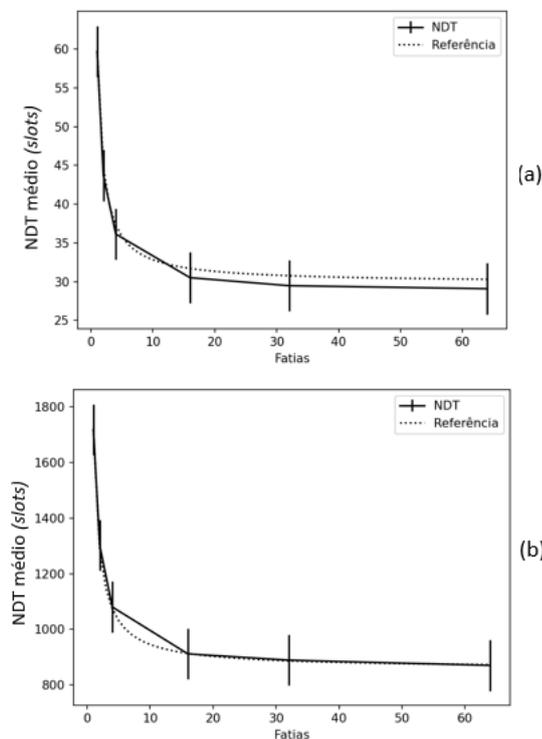


Figura 7 - NDT médio pela quantidade de fatias para os Block Designs (a) $\{7, 3, 1\}$ e (b) $\{183, 14, 1\}$.

Nota-se que a aproximação funciona bem, particularmente para escalonamentos mais longos. Este resultado é relevante, pois o modelo aqui proposto é equivalente ao de [11] quando se considera uma única fatia, e são conhecidas na literatura expressões fechadas para o NDT médio para diversos tipos de escalonamento baseadas no modelo de [11]. Logo, pode-se adaptar estas expressões para o modelo proposto neste artigo para os casos em que $p \ll 1$ através da função $g(\cdot)$.

CONCLUSÃO

Ao considerar o desalinhamento de *slots* para o cálculo do NDT, o modelo proposto neste artigo fornece uma previsão mais precisa que a dos modelos existentes na literatura. Os modelos anteriores, ao não levarem em conta a possibilidade de *slots* desalinhados, subestimam a frequência de ocorrência de oportunidades de encontro e, portanto, superestimam o NDT. Além disso, os modelos de NDT na literatura não consideram o efeito do atraso de propagação e, embora este parâmetro não esteja explícito no modelo proposto neste artigo, seu efeito é implicitamente capturado, sendo contabilizado no próprio *offset* entre os nós.

Através das simulações realizadas para validação do modelo, foi possível concluir que, independentemente do escalonamento analisado, para um valor de probabilidade de sucesso $p \ll 1$ e um número de *subslots* por *slot* tendendo ao infinito, o NDT tende a cair pela metade em relação ao caso

de uma fatia. Para p próximo de um, o NDT também cai, mas não de forma tão acentuada.

No futuro, pretende-se investigar melhor a relação de queda do NDT com o aumento do número de fatias para o caso de p próximo a 1. O objetivo final é obter expressões fechadas para o NDT médio de diversos tipos de escalonamento, talvez baseados nas expressões já disponíveis na literatura para o caso de 1 fatia. Vale ressaltar que o escorregamento de relógio tem outros efeitos além do não alinhamento das bordas dos *slots*, como a possibilidade de os *slots* de cada nó terem durações ligeiramente diferentes. Entretanto, estes outros aspectos não são considerados no nosso modelo atual e serão objeto de estudo em trabalhos futuros.

REFERENCIAS

- [1] Zeadally, Sherali et al. Design architectures for energy harvesting in the Internet of Things. *Renewable and Sustainable Energy Reviews*, v. 128, p. 109901, 2020.
- [2] Anastasi, Giuseppe, et al. "Energy conservation in wireless sensor networks: A survey." *Ad hoc networks* 7.3 (2009): 537-568.
- [3] Carrano, Ricardo C., et al. "A comprehensive analysis on the use of schedule-based asynchronous duty cycling in wireless sensor networks." *Ad Hoc Networks* 16 (2014): 142-164.
- [4] Amin, Farhan, et al. "An Overview of Medium Access Control and Radio Duty Cycling Protocols for Internet of Things." *Electronics* 11.23 (2022): 3873.
- [5] Sun, Yanjun, Omer Gurewitz, and David B. Johnson. "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks." *Proceedings of the 6th ACM conference on Embedded network sensor systems*. 2008.
- [6] Gupta, Sunita, Sakar Gupta, and Dinesh Goyal. "Wireless sensor network in IoT and performance optimization." *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)* 15.1 (2022): 14-22.
- [7] Tripathi, Yogesh, Arun Prakash, and Rajeev Tripathi. "A novel slot scheduling technique for duty-cycle based data transmission for wireless sensor network." *Digital Communications and Networks* 8.3 (2022): 351-358.
- [8] Zheng, Rong, Jennifer C. Hou, and Lui Sha. "Asynchronous wakeup for ad hoc networks." *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. 2003
- [9] Phan, Linh-An, et al. "Performance analysis of time synchronization protocols in wireless sensor networks." *Sensors* 19.13 (2019): 3020.
- [10] Pak, Wooguil, et al. "W-MAC: supporting ultra low duty cycle in wireless sensor networks." *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*. IEEE, 2008.
- [11] Carrano, Ricardo C., et al. "Neighbor discovery time in schedule-based asynchronous duty cycling." *2012 IFIP Wireless Days*. IEEE, 2012.
- [12] Gao, Demin, et al. "Neighbor discovery based on cross-technology communication for mobile applications." *IEEE Transactions on Vehicular Technology* 69.10 (2020): 11179-11191.
- [13] Kim, Song Min, and Tian He. "Freebee: Cross-technology communication via free side-channel." *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 2015.
- [14] Chen, Sixia, et al. "Asynchronous neighbor discovery on duty-cycled mobile devices: Models and schedules." *IEEE Transactions on Wireless Communications* 19.8 (2020): 5204-5217.