

# TSCH-LBV: Alocação dinâmica de canais com largura de banda variável em redes TSCH

Guilherme A. S. Milanez<sup>1</sup>, Thiago Oliveira<sup>1</sup>, Marcos A. M. Vieira<sup>2</sup>, Luiz F. M. Vieira<sup>2</sup>, and José A. M. Nacif<sup>1</sup>

<sup>1</sup>Instituto de Ciências Exatas e Tecnológicas – Universidade Federal de Viçosa (UFV).  
Email: {guilherme.aguiar, thiago.o.barbosa, jnacif}@ufv.br

<sup>2</sup>Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)  
Email: {lfvieira, mmvieira}@dcc.ufmg.br

**Abstract**—A transmissão e recepção de pacotes em redes sem fio sofre com problemas como interferência, colisões e presença de gargalos, causando o aumento da latência e diminuição da vazão na rede. A alta demanda por um conjunto de enlaces pode tornar a rede ineficiente e, até mesmo, inoperante. As redes TSCH tentam evitar colisões via escalonamento de canais e tempo, no entanto, ainda há a presença de gargalos e um aprimoramento para os escalonamentos tradicionais deve ser fornecido. A largura de banda dos canais pode ser usada para melhorar a vazão em regiões com gargalos. Dessa forma, neste artigo propomos a alocação dinâmica de canais com largura de banda variável para reduzir gargalos e aumentar a vazão. A alocação de canais e escalonamento das transmissões são feitos utilizando o método de mapeamento de conflitos, onde são identificados seções de congestionamento de pacotes, em que um conjunto de nós compete pelos mesmos recursos de canal de comunicação e tempo de envio, organizando-os a fim de aprimorar a performance da rede. Neste trabalho propomos um procedimento de alocação de canais com largura de banda proporcional à demanda de pacotes por enlace. O algoritmo foi implementado na plataforma Contiki-NG. Os resultados mostram a redução no tempo do *slotframe* e melhoria na entrega de mensagens.

**Index Terms**—Redes de sensores sem fio, TSCH, largura de banda variável.

## I. INTRODUÇÃO

As redes de sensores sem fio (RSSFs) são compostas por dispositivos com sensores embarcados, capazes de monitorar aspectos de um ambiente por meio da coleta e processamento de dados. Devido ao tráfego volumoso, essas redes sofrem com gargalos que aumentam a latência e reduzem a vazão. Esse problema, geralmente provocado por colisões e interferência entre os enlaces, causa um comportamento ineficiente e indesejado. Em aplicações para essas redes, busca-se sempre entregar os dados coletados com confiabilidade e eficiência, como no monitoramento em ambientes de alto risco [1]. Por esse motivo, alguns protocolos buscam reduzir esses conflitos coordenando os enlaces para que não haja sobreposição de tempo e canais de comunicação, diminuindo os gargalos e aumentando a vazão.

Nesse cenário, muitas pesquisas atuais buscam aumentar o desempenho mitigando os gargalos e demais problemas de performance. O padrão IEEE 802.15.4e [2], por exemplo, dispõe de uma configuração de comunicação eficiente baseada em intervalos de tempo, capaz de mitigar os conflitos.

Contudo, mesmo oferecendo um preciso controle sobre a atividade dos nós da rede, o *Time Slotted Channel Hopping* (TSCH), diferentemente de outros protocolos aptos a operar com largura de banda variável, sofre com problemas como aumento de latência e prolongamento do *slotframe* quando o tráfego de pacotes é volumoso. Uma solução seria a adequação do protocolo para operar com canais de largura de banda variável.

Este trabalho busca atenuar os efeitos descritos ao propor um protocolo adicional para o TSCH, cujo principal objetivo consiste na alocação de canais com largura de banda variável. O TSCH - Largura de Banda Variável (TSCH-LBV) utiliza a topologia da rede para organizar os enlaces evitando conflitos e aumenta a vazão selecionando enlaces com maiores requisições para serem operados em canais mais largos. O TSCH-LBV é composto por um algoritmo que identifica os tipos básicos de conflitos de interferência analisando a topologia prévia da rede e construindo uma matriz com os parâmetros de tempo e canais para cada um dos links. O algoritmo foi adaptado para ser executado no ambiente do Contiki-NG, a fim de coletar dados com o simulador COOJA.

As contribuições deste trabalho são: (i) apresentação de um escalonamento que coordena comunicações em canais de largura de banda variável para redes operando com o protocolo TSCH; (ii) implementação de um protocolo adaptado para o sistema operacional Contiki-NG; (iii) resultados que mostram a redução do uso de *slots* de tempo para envio de pacotes; (iv) validação experimental da proposta, provando ser superior em relação à outro algoritmo com mesma proposta de identificação de conflito.

O TSCH-LBV apresenta uma abordagem inédita na literatura para aumentar a vazão e aprimorar a performance da rede. Diferentemente do estado de arte, em que muitos algoritmos de escalonamento para o TSCH utilizam técnicas centralizadas, autônomas ou apenas com detecção de conflitos, o TSCH-LBV aprimora a vazão propondo um escalonamento distribuído adaptado para canais com largura de banda variável. Para isso, a extensão proposta dispõe e seleciona canais com maiores larguras de banda para enlaces cuja requisição é maior.

O restante do trabalho está organizado como segue. A Seção II revisa a base teórica utilizada. A Seção III apresenta

os trabalhos relacionados. Na Seção IV é descrito o protocolo proposto. A Seção V apresenta e discute os resultados. Por fim, a Seção VI conclui o artigo.

## II. REFERENCIAL TEÓRICO

Esta seção apresenta a base teórica utilizada neste trabalho. Portanto, mostra-se pertinente discorrer sobre a motivação para o aumento da largura de banda, descrever os tipos de enlaces empregados, evidenciar o modelo de funcionamento do padrão IEEE 802.15.4, conceituar aspectos sobre os principais tipos de escalonamentos neste padrão e apresentar as definições dos conflitos tratados pelo algoritmo.

### A. Largura de banda variável

Uma das principais motivações para aumentar a largura de banda de determinados canais é intensificar a transmissão e evitar gargalos. A capacidade de um canal  $C$ , que é perturbado por um ruído branco gaussiano, é obtida através da Equação 1. Em que  $B_w$  representa a largura de banda do canal,  $S$  a energia do sinal e  $N$  a energia média do ruído [3]. Percebe-se que é possível melhorar a capacidade do sistema aprimorando tanto a energia do sinal quanto melhorando a largura de banda. O primeiro fator não apresenta grande viabilidade no contexto de RSSFs no que diz respeito ao consumo energético, tornando o aperfeiçoamento do uso da largura de banda como o principal objetivo deste trabalho, a fim de melhorar a performance de transmissão da rede.

$$C \leq B_w \log_2 \left( 1 + \frac{S}{N} \right) \quad (1)$$

### B. Enlaces

Ao longo do trabalho, são apresentados métodos que tratam a organização dos enlaces da rede a fim de torná-la mais eficiente, sendo necessário defini-los de antemão. Os dois tipos de enlaces de comunicação nos quais este trabalho opera, são os links de *unicast* e *broadcast* [4].

Enlaces de *unicast* são aqueles em que a comunicação é dada entre dois nós, ponto a ponto, sendo utilizados no envio de pacotes de dados. Enlaces de *broadcast*, são aqueles nos quais a mensagem não possui apenas um único destinatário, sendo usados para envio de mensagens de manutenção da rede. Neste trabalho os links de *unicast* são usados para envio e recepção de dados. Já os de *broadcast* para, além da manutenção da rede, realizar a descoberta de nós vizinhos.

### C. IEEE 802.15.4

O padrão IEEE 802.15.4, IEEE 802.15.4e para ambientes industriais, constrói uma rede sincronizada de nós utilizando uma política de intervalos de tempo agrupadas em um *slotframe*. Essa estrutura composta por todos os links cadastrados, que se repete periodicamente, abriga informações relevantes na comunicação dos nós como: canal de comunicação, *slot* de tempo, tipo do link, quantidade de intervalos de tempo disponíveis dentre outras [5].

Para que as transmissões possam ser executadas, é necessário, primeiramente, que um novo nó seja associado com

a rede. O processo é sucedido quando o nó folha encontra-se sincronizado com os demais. Este novo componente só poderá ser inserido quando receber um *Beacon* de outro nó já devidamente sincronizado. Um *Beacon*, por sua vez, é uma mensagem que carrega informações relevantes como número absoluto de *slot* (ASN), a sequência de espera de canal e o identificador do *slotframe* usado.

Após a escuta, de acordo com X. Vilajosana et al. [6], o coordenador PAN envia os parâmetros de tempo até o novo nó folha por meio da estrutura de um grafo dirigido acíclico [5]. A atualização do tempo acontece sempre quando os nós pais enviam ACKs, mensagens indicando sucesso na comunicação, ou pacotes de dados. O escalonamento proposto neste trabalho atua na organização dos enlaces já devidamente estabelecidos, sendo a sua criação, responsabilidade de um algoritmo genérico executado por cada nó. Portanto, cada nó constrói a topologia da rede, criando tanto links de *broadcast* quanto de *unicast* para transmissão e recepção de dados.

### D. Escalonamento

Ao serem sincronizados, os nós não são capazes de organizar as transmissões de forma autônoma, sendo necessário incorporar um mecanismo de escalonamento capaz de coordenar as atividades dos nós. O escalonamento utiliza conceitos previamente estabelecidos de intervalos de tempo com salto de canais, como os adotados nos protocolos *Time Division Multiple Access* (TDMA) e *Frequency Division Multiple Access* (FDMA). Ademais, o escalonador sustenta uma boa confiabilidade e maior conservação de energia, melhorando a vida útil da bateria [7].

Existem na literatura, quatro tipos de escalonamento para o TSCH [8]: distribuídos, centralizados, autônomos e híbridos [9]. Na abordagem distribuída, os próprios nós coordenam as atividades. Nas abordagens autônomas, os nós negociam com seus vizinhos as melhores configurações para o escalonamento. Nas centralizadas, um único nó realiza o escalonamento com conhecimento global da rede. Já nos híbridos, combina-se mais de uma técnica.

O TSCH-LBV opera de forma distribuída. Ao obter a topologia por meio de um *snapshot* [10], os nós, operando com o protocolo, constroem seus próprios escalonamentos, utilizando conhecimento global da rede. Isso faz com que os nós não sejam estritamente dependentes de um coordenador, tornando o algoritmo mais escalável para sua implementação e evitando a sobrecarga de sinalização sobre o nó raiz [9].

Apesar dos escalonamentos centralizados serem mais eficientes, em termos de desempenho, que os distribuídos. Muitos trabalhos [11], [12], [13] dispõem apenas de uma análise teórica por trás da melhoria proposta pelas heurísticas e não apresentam uma implementação prática. Ao comparar o TSCH-LBV com o estado da arte de escalonadores centralizados [12], demonstra-se que o algoritmo possui desempenho melhor que os centralizados e, conseqüentemente, mais eficiente que os distribuídos. Ademais, com a abordagem distribuída é possível implementar o escalonamento em um sistema operacional real para RSSFs.

### E. Conflitos

O problema da definição de frequência para os enlaces pode ser tratado de forma semelhante ao problema de coloração de vértices em grafos de interferência [14]. Por esta razão o algoritmo considera a mesma metodologia para o tratamento dos conflitos. Em protocolos baseados na política TDMA, quando um nó está em transmissão, o conjunto de nós que geram conflitos com o operante não podem transmitir. Por conta disso, é de suma importância a projeção da rede com base na topologia para criar um agendamento capaz de evitar os conflitos.

Os dois tipos de conflitos básicos, que o trabalho trata, são os primários e secundários [15]. Para os conflitos primários, é necessário observar se um dado nó está, simultaneamente, atuando como transmissor e receptor ou se este determinado nó é o receptor de múltiplas transmissões. Essa restrição é consequência do caráter *half-duplex* dos rádios equipados nos nodos que operam o padrão IEEE 802.15.4. Já os conflitos secundários são identificados quando um dado nó atuando como receptor se encontra entre uma outra transmissão não destinada a este primeiro, ou seja, o nó em conflito está dentro do raio de alcance de outra transmissão, da qual não participa.

### III. TRABALHOS RELACIONADOS

Alguns trabalhos recentes buscaram melhorar a eficiência das redes operando TSCH por meio de diversos tipos de escalonamentos. Tavares et al. [16] propõem uma extensão para o padrão de rede sem fio *Carrier Sense Multiple Access* (CSMA) a fim de permitir a comunicação por meio de canais com largura de banda variável. Os autores atuam principalmente tentando mitigar o fenômeno do funil no qual o volumoso tráfego perto do sink causa gargalos e lentidão na coleta de dados. Nosso projeto, contudo, implementa um escalonamento que seleciona maiores larguras de bandas para canais em redes TSCH que, por sua vez, é baseado em TDMA.

Recentemente, Ribeiro Jr et al. [17] propuseram a utilização de dois rádios em duas bandas diferentes para transmissão e/ou recepção ao mesmo tempo. Isso equivale a utilizar dois canais ao mesmo tempo, o que pode ser visto como um canal de largura de banda com o dobro do tamanho. O trabalho utiliza CSMA. Diferente deles, nosso trabalho permite utilizar canais com largura de banda variável não estando limitado a dois, e, além disso, com TSCH.

Já Palattella et al. [12], apresentam um algoritmo de escalonamento centralizado denominado *Traffic Aware Scheduling Algorithm* (TASA), cujo princípio baseia-se na análise do grafo da rede por coloração, emparelhamento e na carga de trabalho de cada nó. O nosso projeto, além de utilizar a mesma ideia de mapeamento de conflitos para determinar quais arestas terão prioridade de alocação, apresenta, também, suporte para canais com largura de banda variável, escalonados de forma descentralizada.

O grupo IETF 6TiSCH, propõe um procedimento de negociação, também distribuído, denominado 6top [18] que permite operações em células, suportando a função de escalonamento mínima do 6TiSCH criando enlaces sem critérios

rigorosos, com um tamanho mínimo de largura de banda. O nosso trabalho, entretanto, busca alocar canais com larguras de banda maiores conforme a quantidade de pacotes a serem transmitidos, selecionando enlaces de forma a evitar conflitos.

Em [19] Osman e F. Nabki, melhoram o algoritmo Orchestra [20], ao propor um algoritmo autônomo de alocação dinâmica de *slots* de tempo conforme a necessidade do nó transmissor. O TSCH-LBV, todavia, além de ser de caráter distribuído, aloca canais mais largos, conforme a necessidade da transmissão, e não *slots* de tempo. A alocação dos *slots* no TSCH-LBV é reduzida em virtude do melhor uso dos canais, como é apresentado na Seção V.

Por fim, Kim et al. [21] apresentam um escalonamento para o TSCH, implementado para o Contiki-NG, que, de forma autônoma, efetua a análise e organização dos links. Este trabalho, por sua vez, utiliza análise de links e é implementado para o Contiki-NG. Porém, opera de forma distribuída e em canais de largura de banda variável.

Dado o exposto, existem diversos trabalhos apresentando propostas de escalonamentos eficientes para o protocolo IEEE 802.15.4. Todavia, até então nenhum apresentou um método de escalonamento distribuído que utiliza técnica de mapeamento de conflitos a fim de definir quais enlaces devem ser alocados e qual o tamanho de banda ideal para a transmissão, com base na carga de trabalho.

### IV. PROTOCOLO TSCH-LBV

O TSCH-LBV tem como objetivo estabelecer uma organização de enlaces que, com base na análise da topologia e dos possíveis conflitos, reduza o tempo de transmissão, ao utilizar canais com maiores larguras de banda. Para isso, organiza-se apenas o *slotframe* de enlaces de *unicast*, havendo um *slotframe* apenas para *broadcast* que não é escalonado seguindo métricas rigorosas.

Portanto, todos os nós executam o procedimento da seguinte forma: analisa-se a topologia, escalona-se os links evitando conflitos e seleciona-se canais com maiores larguras de banda para enlaces com maiores demandas. Os canais mais largos são abstraídos ao unir mais de um canal padrão no mesmo *slot* de tempo em frequências diferentes como ilustrado na figura 1. Ao invés de usar três *slots* em um mesmo canal, usa-se apenas um *slot* mas em canais com frequências diferentes que representam um canal com maior largura de banda.

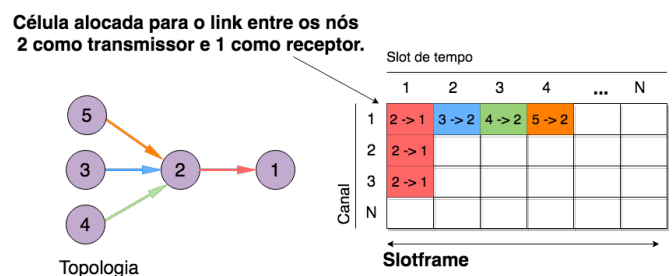


Fig. 1. Topologia e escalonamento do *slotframe*.

### A. Procedimento

O procedimento começa com a execução de um algoritmo, distribuído, de descoberta de vizinhos, para construir a topologia. Em seguida, um *snapshot* [10] da rede é usado a fim de mapeá-la. O *snapshot* é armazenado em um arquivo de forma que cada linha deste arquivo possua dois inteiros que identificam o transmissor e o receptor de um enlace componente. Portanto, a finalidade desse arquivo consiste no manutenção do conhecimento global da rede durante a execução do escalonamento.

A topologia é representada por um grafo direcionado denotado por  $G(V, A)$  para  $V$  representando o conjunto de nós na rede e  $A$  os respectivos enlaces, ou seja, as arestas do grafo.  $G$  é definido como acíclico uma vez que há um sumidouro padrão  $S = 1$ , equivalente ao coordenador PAN da rede, bem como não é permitida a formação de ciclos. Por conta dessa restrição define-se um formato de topologia único para simulações, baseado em árvore, havendo apenas um nó pai para cada nó filho, respeitando a seguinte relação:  $Identificador_{filho} > Identificador_{pai}$ . Portanto o número de arestas é dado por:  $|A| = V - 1$ , tendo o nó sorvedouro apenas como receptor. Após ler a topologia cada um dos nós, individualmente, cria uma matriz de adjacência  $MA$ ,  $n \times n$  sendo  $n$  o número de nós, para dar início ao escalonamento.

Para cada nó, transmissor, é atribuído um inteiro cuja função é representar um pacote virtual, tornando possível pré-simular a transmissão, admitindo que a rede terá o comportamento exato da topologia descrita. Para formular as coordenadas do escalonamento, admite-se que há apenas um pacote para ser entregue por cada nó transmissor, no período descrito pelo *slotframe*. À medida que o algoritmo opera, esses valores são alterados, podendo aumentar ou diminuir, de acordo com a simulação de envio e recepção. O conjunto desses pacotes é utilizado tanto para reorganizar os enlaces de acordo com suas demandas, quanto para controlar o laço gerando a nova disposição de enlaces no *slotframe*.

Para armazenar os conflitos utiliza-se um grafo de conflitos representado por uma matriz  $MC$   $a \times a$  sendo  $a$  o número de arestas em  $G$ , o número de arestas é contabilizado lendo a quantidade de linhas do arquivo da topologia, uma vez que em sua criação não é permitida escrita de enlaces duplicados. Em  $MC$  há o mapeamento dos possíveis conflitos presentes na rede, a partir da disposição fornecida. Cada nó do grafo de conflito representa um enlace da rede, quando quaisquer dois links apresentam conflitos, tanto primários quanto secundários, gera-se, então, um enlace entre os nós equivalentes no grafo. Quando dois desses nós estão ligados, significa que eles estão em conflito e por isso os enlaces representados por esses nós não podem transmitir ao mesmo tempo. O mapeamento dos conflitos é substancial para a definição da disposição final dos enlaces alocados.

O Algoritmo 1 descreve o processo de construção da matriz de enlaces selecionados,  $ML = [m_{i,j}]$ . Para definir esta estrutura, é necessário observar o mapeamento dos conflitos. Primeiramente, analisa-se o conjunto de nós identificando qual

possui maior quantidade de pacotes a serem transmitidos. Assim como no trabalho [12], o Algoritmo 1 observa o problema do emparelhamento em um grafo direcionado a fim de identificar as melhores arestas tendo em vista seus possíveis conflitos. Ou seja, seleciona-se as melhores posições da matriz para os enlaces que, envolvidos com o nó de maior carga, não possuem vértices em comum com os demais. Ao finalizar o processo de identificação, retorna-se uma matriz com as devidas coordenadas contendo os respectivos enlaces que podem ser usados em um mesmo intervalo de tempo.

---

#### Algorithm 1 Geração da matriz de enlaces selecionados

---

```

procedure GERAMATCHING(node)    ▷ Descobre quem
pode transmitir com nó
  Seleciona um nó aleatório para iniciar o matching
   $MLs \leftarrow 0$  ▷ Preenche a matriz do matching com zeros
  for cada nó da rede do
    Preenche lista com os nós que transmitirão nesse slot
    de tempo
    for cada nó na lista do
      Retira os nós que geram conflito com o nó sele-
      cionado
      while Ainda houver nó não selecionado
      na lista do
        for cada nó na lista do
          Seleciona o nó com mais pacotes que ainda está
          na lista
          for cada nó na lista do
            Retira os nós que geram conflito com o nó
            selecionado
            for cada nó na lista do ▷ Verifica se há nós
            não verificados
              Verifica se ainda há nós na lista que não foram
              selecionados
              for cada nó na lista do    ▷ Preenche a matriz
              com os nós selecionados
                 $ML \leftarrow Lista$  ▷ Preenche a matriz com os nós que
                vão transmitir

```

---

Ao término da criação das estruturas, o algoritmo entra em *loop* enquanto ainda houver entregas de pacotes virtuais. Para realizar o envio desses pacotes são implementados dois procedimentos auxiliares, um para elencar a quantidade de pacotes por links de acordo com sua respectiva demanda e outro para relacionar os pacotes já tratados com a matriz de canais alocados, que em caso afirmativo poderá ter o respectivo link anotado e dotado das informações de canal de envio e *slot* de tempo adequado.

À medida que os links com maior demanda de pacote são escalonados para as melhores posições no quadro,  $MC$  deverá ser remanejada em função dos demais links. Para efetuar isso, os procedimentos de análise de emparelhamento são convocados a cada laço. No interior do laço, durante a leitura da  $MC$  junto à análise da matriz de enlaces selecionados, ocorre a seleção da largura de banda.

Conforme os enlaces com maiores demandas forem sendo identificados, canais com maiores larguras devem ser alocados. O Algoritmo 2, demonstra como os canais maiores são determinados. Uma vez que o atual padrão não possui suporte para canais maiores que 2 MHz, o método acopla dois ou mais canais para um único enlace em um mesmo *slot* de tempo. Para isso, incrementam-se os valores das coordenadas concernentes aos canais para enlaces que irão transmitir mais de um pacote. Portanto, a alocação de mais canais em um mesmo instante de tempo, representa a alocação de 1 canal de largura de banda variável.

---

**Algorithm 2** Alocação de canal com maior largura de banda

```

1: procedure AUMENTALARGURA ▷ Selecciona a aresta na
   matriz de adjacência
2:   for cada Aresta em ML do
3:     NumPacotes ← 0
4:     Pacotes ← VetorPacotes[Aresta]
5:     while NumPacotes < Pacotes do           ▷
   Contabiliza todos os pacotes
6:       if Canal == TotalCanais then Break ▷ Não há
   mais canais disponíveis
7:       CanaisAlocados[Canal][Tempo] ← Aresta
8:       Canal ++ ▷ Conta quanto canais padrões
   foram alocados
9:       CanalLBV ← Canal ▷ Canal de largura de
   banda variável representa os N canais alocados
10:      return CanaisAlocados ▷ Retorna a matriz com
   as novas regras

```

---

### B. Implementação no Contiki-NG

O Contiki-NG, sistema operacional para dispositivos que operam em RSSFs, poupa o trabalho da implementação da rede, sendo necessário modificar apenas a camada MAC com o protocolo TSCH. Ademais, opera-se com o UDP na camada de transporte, o RPL-lite na camada de roteamento e com 6LoWPAN na camada de adaptação. O algoritmo de escalonamento entrega uma estrutura contendo as coordenadas (Canal X Tempo) para que os links possam ser reorganizados a partir dessas informações. Para que isso ocorra, os links são identificados por variáveis inteiras pertencentes ao intervalo de  $[1, n]$  adotadas no momento de sua criação e, através de seus identificadores (IDs), são ordenados caso haja uma correspondência, de seu identificador, na matriz gerada pelo escalonamento.

Em cada nó, há um algoritmo embarcado que realiza a descoberta de seus vizinhos, gera os seus respectivos enlaces e convoca o escalonamento. Para a topologia ser criada, primeiramente, são enviadas mensagens de *broadcast* a fim de consolidar um conjunto de  $N$  vizinhos. Um nó  $n_k$  qualquer ao receber uma mensagem adota o transmissor como seu vizinho sem definir o grau de parentesco. No momento em que  $|N| \geq 1$  identifica-se a presença de ao menos um vizinho, sendo permitida a criação de um link de transmissão

*unicast* do nó de maior identificador (filho) para o de menor (pai). O algoritmo de *snapshot*, então, computa um novo link de *unicast*, armazenando os identificadores dos nós em um arquivo .txt único, em um diretório do computador da simulação, identificando a topologia usada no procedimento da Seção IV-A.

Durante a execução, os nós criam links de recepção perfeitamente sincronizados com os de transmissão. Ao possuir conhecimento global da rede, todo nó convoca uma função que irá percorrer o arquivo da topologia e, caso haja reconhecimento do identificador (ID) do nó em atividade com o respectivo na linha de leitura, cria-se um link de recepção cujo transmissor é o outro nó escrito na linha lida. Para que não haja redundâncias, antes da criação de um novo link de recepção percorre-se todo o *slotframe* a fim de verificar se o mesmo enlace já existe. A Figura 2 ilustra como é feita a integração do escalonamento com o *slotframe*, no Contiki-NG, tendo estabelecida a topologia e os links, tanto de transmissão quanto de recepção, a serem escalonados.

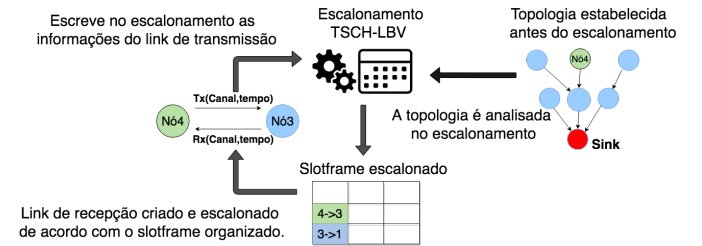


Fig. 2. Esquema de organização dos links.

No ambiente do Contiki-NG, não é possível ter acesso ao *slotframe* total, ou seja, nem mesmo o *sink* terá acesso a todos os links da rede sem ajuda de algum aparato. Dessa forma, cada um dos nós convoca o escalonamento, periodicamente, para organizar a parte do *slotframe* concernente a este nó.

Para escalonar os links, cada um dos nós executa o escalonamento conhecendo a topologia completa da rede. Ao obter uma matriz  $M_l$  de canais alocados, percorre-se a mesma, verificando compatibilidades nos identificadores de seus enlaces. Em caso de compatibilidade, se na matriz for selecionado apenas uma posição  $l = m_{i,j}$  para o link, apenas reescreve-se os parâmetros de canal e tempo. Contudo se  $l = \{m_{i,j}, m_{i+1,j}, \dots, m_{i+n,j}\} \mid i = Canal$ , descrevendo o uso de canais com maiores larguras de banda, armazenase quantos canais extras foram alocados, para aquele enlace específico, a fim de computar a melhora na entrega dos pacotes.

### C. Largura de banda variável

Para escalonar canais com maiores larguras de bandas, é necessário aprimorar a faixa de frequência suportada pelo TSCH a fim de dedicar maiores espectros para a transmissão. Portanto, é fundamental o uso de um rádio apto para realizar tais transmissões. Devido ao trabalho não se concentrar em melhorar a capacidade dos rádios operados, considera-se que o hardware utilizado é capaz de executar transmissões em

canais maiores admitindo o sucesso do aumento da largura nos casos de entregas feitas por enlaces que utilizavam esses canais, respeitando a dinâmica da equação 4.

Conseqüentemente, o escalonamento pode alocar canais com intervalos de frequência superiores a 2 MHz, como exemplificado na Figura 3, em que enlaces com maiores demandas utilizam mais de um canal padrão acoplados representando um canal maior. Isso é concebido pelo escalonamento ao denotar mais de um valor para o índice de deslocamento do canal (*channel offset*), índice fornecido pelo Algoritmo 2, sendo convertido em espectro de frequência com base na Equação 2. Para obter a frequência  $f$  é necessário obter o valor absoluto do *slot*,  $ASN$ , por meio da Equação 3, utilizando o tamanho do *slotframe*,  $S$ , o tamanho de seu ciclo,  $k$ , e o atual *slot*,  $t$ . O *chOF* indica o somatório dos índices dos canais alocados pelo escalonamento e  $n_{ch}$  é a quantidade de canais que podem ser utilizados. Ao serem incrementados, os índices de deslocamento de canais gerados pelo escalonamento são armazenados, a fim de realizar as análises do desempenho do algoritmo, utilizando a Equação 4 como base para comparação.

$$f = F\{(ASN + chOF) \text{ Mod } n_{ch}\} \quad (2)$$

$$ASN = (k * S + t) \quad (3)$$

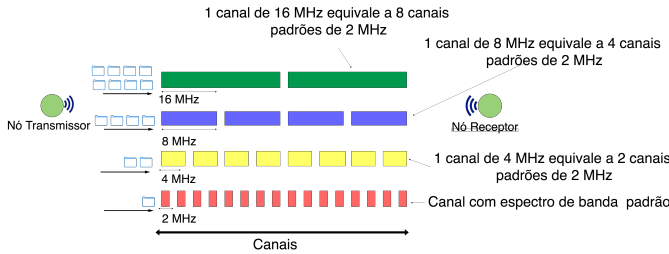


Fig. 3. Número de pacotes transmitidos por largura de banda variável.

#### D. Complexidade

Esta seção apresentará as análises de complexidade das principais etapas de funcionamento do TSCH-LBV. Acerca da construção da topologia, para criar links de transmissão os nós devem percorrer a lista de vizinhos, a fim de definir qual será o seu nó pai, havendo um custo associado, nesta tarefa, de  $O(V)$ , em que  $V$  é o número de vizinhos deste nó. Para a subsequente criação do link de recepção, deve-se percorrer o arquivo de topologia para encontrar qual é o filho do respectivo nó em execução. Havendo, então, um custo associado de  $O(A)$  em que  $A$  é o número de arestas escritas no arquivo da topologia. Desse modo, o custo total, em termos de tempo, dessa etapa é de  $O(V + A)$ , por nó na rede.

Em termos de espaço, existem duas estruturas com custos mais relevantes que são as matrizes de adjacência dos grafos da topologia da rede e de conflitos, ambas possuem custo associado de  $O(A^2)$ , sendo  $A$  o número de arestas nos respectivos grafos.

A complexidade do número de pacotes do algoritmo é de  $\Omega(N)$ , em que  $N$  é o número de nós na rede, haja vista que, um nó precisa mandar mensagens para seu pai e, ao passo que recebe alguma mensagem de algum de seus filhos o nó envia mensagens para seu respectivo pai.

O Algoritmo 1 possui complexidade de tempo de  $\Omega(N^2)$ . A priori, o conjunto dos nós é percorrido em um laço de repetição externo. Internamente, o mesmo conjunto de nós é explorado, diversas vezes a fim de, respectivamente, verificar qual pacote tem maior carga de trabalho, retirar os nós que geram conflitos (identificando os tipos de conflitos) com o nó marcado, verificar se existem nós que não foram computados e preencher a matriz com os selecionados.

Já no Algoritmo 2 para cada enlace,  $L$ , percorre-se a lista de pacotes,  $P$ , a fim contabilizar e alocar canais com maiores larguras de banda de acordo com a demanda. Portanto, o algoritmo tem complexidade de tempo de  $O(L \times P)$ .

## V. RESULTADOS

Os resultados bem como os métodos de análise para aferir o comportamento do TSCH-LBV são descritos nesta seção. Na primeira análise, o algoritmo de escalonamento foi implementado na linguagem C, executado em ambiente Linux, com GNU GCC 10.3, para aferir a relação entre quantidade de *slots* de tempo alocados pelo tamanho da rede. Os outros dados foram colhidos a partir do simulador COOJA<sup>1</sup>, também executado no ambiente Linux. Para tal, implementou-se o algoritmo utilizando a arquitetura do *Contiki-NG* 4.5, para coletar a taxa de entrega de pacotes e a vazão. Todos os testes foram comparados com o TASA [12]. O TASA, além de ser o estado da arte de escalonadores centralizados, o que oferece uma relação do TSCH-LBV com um dos escalonamentos mais bem qualificados para esse tipo de rede, opera com a mesma técnica de conhecimento global da rede. Desconsiderando a obtenção da topologia, a alocação de canais maiores é a única etapa que difere os dois algoritmos em termos de custo. Todas as análises foram feitas por meio de simulações. Mesmo no COOJA, o mote adotado é exclusivo para simulação, não tendo suas especificações condizentes com a realidade.

Acerca do consumo energético, em decorrência de que na literatura [17], constata-se que o uso de dois rádios melhora eficiência energética e que operar em dois canais ou dois rádios, corresponde à duplicar o tamanho da largura de banda para determinado enlace. Denota-se que o TSCH-LBV não oferece prejuízo em termos de consumo energético.

#### A. Análise de slots alocados

Primeiramente, analisa-se a relação de *slots* alocados por tamanho da rede, em um intervalo de [100, 500] nós. Utilizou-se um simulador feito em linguagem C, que simulava o envio e recepção de pacotes com base em uma topologia fornecida. Tendo em vista que a simulação das alocações de *slots* de tempo, independe de fatores provindos do sistema operacional escolhido (*Contiki-NG*), optou-se por performar o algoritmo

<sup>1</sup><https://github.com/lesc-ufv/TSCH-LBV>

fora do COOJA. Para cada tamanho de rede, foram feitas 10 simulações, em cada uma gerou-se um arquivo de topologia aleatória na forma padrão adotada neste trabalho, na qual não há ciclos, em cada par o nó de ID maior é o transmissor e o sink não realiza transmissões. A Figura 4 representa o gráfico com a relação.

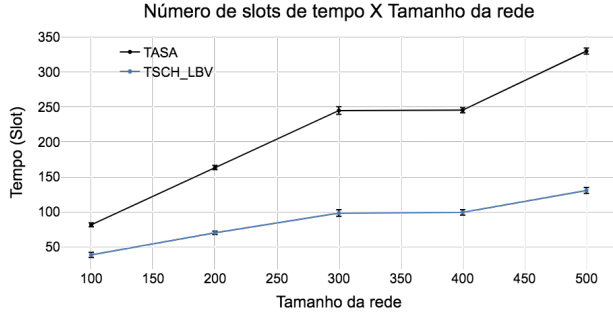


Fig. 4. Relação de *slots* alocados com o tamanho da rede.

O TSCH-LBV apresentou uma melhora expressiva em relação ao TASA, frente a quantidade de *slots* de tempo alocados para redes densas, não ultrapassando o limite de 200 *slots* de tempo para redes de 500 nós. Esse fator é positivo pois indica que mais pacotes podem ser enviados em um período de tempo menor, aumentando a vazão, sem ter que alocar mais links na rede, apenas alargando a banda dos canais utilizados.

#### B. Análises no Contiki-NG

O TSCH-LBV foi experimentado utilizando como padrão do Contiki-NG<sup>2</sup>, além do TSCH, o RPL-Lite e o IPV6. Em sua construção, como foi de suma importância o uso de arquivos, manteve-se a implementação no cooja mote, tendo em vista que a MCU MSP430, comum em grande parte dos outros motes, não é apta para o uso de arquivos. O tamanho de *slot* adotado foi o padrão de 10 ms. Como o TSCH-LBV opera de forma distribuída, todos os nós alocam todas as estruturas do algoritmo e quanto maior a rede maior será o consumo de memória. Durante as análises constatou-se 18 nós como o limite máximo por simulação, a fim prevenir estouros de memória.

Ao todo, para cada tamanho de rede, foram feitas 10 simulações. Em toda simulação, executada no COOJA, altera-se a disposição geográfica dos nós de forma aleatória em uma área simulada equivalente à 10000 m<sup>2</sup>, em dimensões reais, sendo que cada um dos motes tem raio de alcance de 50 m, fator que pode interferir na quantidade de enlaces da rede. Devido ao formato padrão das topologias, descrito na seção IV-A, uma vez localizado próximo apenas de nós com IDs maiores, um dado nó não cria links de transmissão. Existem casos em que um nó é disposto de forma que nenhum dos demais entra em seu raio de alcance e, portanto, não é sincronizado com a rede. Esse fator explica a variabilidade dos dados, nos gráficos das Figuras 6 e 5.

Os testes foram realizados com um modelo de topologia baseado em árvore desbalanceada, em que todos os nós, exceto o coordenador PAN, criavam links de transmissão com nós de IDs inferiores. O intervalo de envio dos pacotes foi de 10 ms, o intervalo de envio de *beacons* e mensagens de *keepAlive* foi de 1 segundo, sem variação. Todas as simulações foram executadas no intervalo de  $(6.0 \pm 1.0) * 10^4$  ms. Em média o intervalo de tempo para a composição da topologia, ficou em torno de 16736.2 ms, com intervalo de confiança de [13330.96, 20141.44] ms. O tamanho do *slotframe*,  $S$ , varia de acordo com o número de nós,  $N$ , obedecendo à relação  $S = N + 2$ .

Para aferir o comportamento do TSCH-LBV, foi necessário realizar uma inferência matemática para calcular a quantidade de pacotes entregues utilizando canais maiores. O padrão IEEE-802.15.4 por definição adota o tamanho da banda de 2 MHz para os canais. Portanto, a cada pacote entregue,  $N_{pkt}$ , por um enlace,  $LP_{n_1 \rightarrow n_2}$ , no qual houve aumento da largura de banda, contabilizou-se o valor correspondente ao número de canais alocados,  $N_{ch}$ , para um único enlace, contando tantos os pacotes enviados, pelo transmissor, quanto os recebidos, pelo receptor, como evidenciado na Equação 4. A contagem, para aferição da taxa de entrega de pacotes e da vazão, foi feita utilizando o registro de *slot* nativo do TSCH no Contiki-NG. Contabilizou-se apenas pacotes de dados provindos de links *unicast*, todos com tamanho de 43 bytes.

$$N_{pkt} = \begin{cases} N_{pkt} \in LP_{n_1 \rightarrow n_2} \times N_{ch} \in LP_{n_1 \rightarrow n_2} \\ \quad \text{if } N_{ch} \in LC_{n_1 \rightarrow n_2} \geq 1 \\ 0 \quad \text{else} \end{cases} \quad (4)$$

A Figura 5 apresenta a relação de entrega de pacotes por tamanho da rede, dada por  $TaxaPacotesEntregues = PacotesRecebidos/PacotesEnviados$ . Denota-se que em 8 dos 14 tamanhos simulados, o TSCH-LBV, apresentou melhora na taxa de entrega de pacotes em relação ao TASA. Essa métrica não apresenta um gradiente expressivo entre os dois algoritmos, uma vez que ambas as variáveis, tanto a quantidade de pacotes enviados quanto a de pacotes recebidos, sofrem aumento à medida que a largura de banda dos canais é acrescida. Mesmo assim, a superioridade do TSCH-LBV, sustenta o argumento de uma confiabilidade melhor por conta dos canais melhorados.

A Figura 6 ilustra o gráfico representando o comportamento da vazão frente ao tamanho da rede. Nota-se que o TSCH-LBV apresenta uma vazão superior ao do TASA em todos os casos simulados. Isso comprova que o aumento do tamanho da banda dos canais pode ser uma boa abordagem a fim de aumentar a vazão. Em todas as simulações o *slotframe* de *broadcast* foi utilizado por todos os nós para manter a associação dos nós na rede e seu tamanho é simétrico ao do *slotframe* de *unicast*. Apesar das mensagens de *Broadcast* não serem contabilizadas, a quantidade de *slots* percorridos, considera o tempo usado para as mesmas, explicando a redução dos valores da vazão para os dois algoritmos.

<sup>2</sup><https://github.com/contiki-ng/contiki-ng>

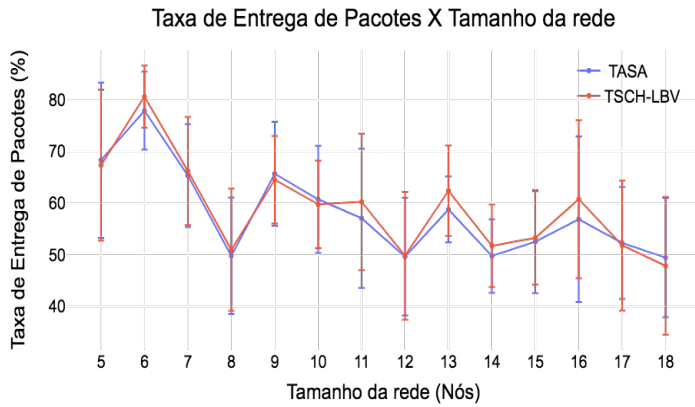


Fig. 5. Taxa de entrega de pacotes por tamanho da rede.

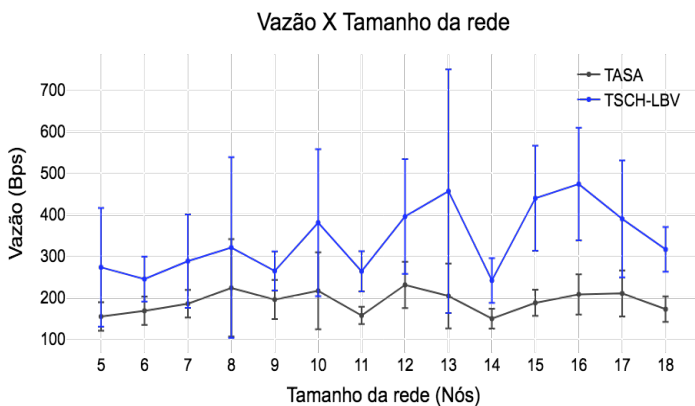


Fig. 6. Vazão por tamanho da rede.

## VI. CONCLUSÃO

Projetou-se, neste trabalho, um escalonamento para canais com largura de banda variável em redes TSCH. Para tal, implementou-se um algoritmo capaz de simular apenas entregas e recepção de pacotes para analisar os *slots* de tempo alocados. Implementou-se, também, um algoritmo integrado à arquitetura do Contiki-NG, capaz de operar em conjunto com outras camadas de uma rede de sensores sem fio e um algoritmo genérico capaz de gerar topologias no padrão de árvore. Denotou-se uma melhora frente ao algoritmo comparado, que adota uma largura de banda padrão para os canais operados no TSCH, ao apresentar um aumento na vazão e diminuição da alocação de *slots* de tempo, por conta do aumento da capacidade de transmissão dos canais operados. Portanto, é possível afirmar que ao operar em canais com largura de banda variável no TSCH, mesmo que em redes menores, é possível obter melhorias na vazão. Como trabalhos futuros, pretende-se aprimorar a vazão, modificar o protocolo a fim de permitir a sua incorporação em redes mais densas utilizando o Contiki-NG e apresentar uma análise acurada acerca do consumo energético.

- [1] R. Lin, Z. Wang, and Y. Sun, "Wireless sensor networks solutions for real time monitoring of nuclear power plant," in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, vol. 4. IEEE, 2004, pp. 3663–3667.
- [2] D. De Guglielmo, S. Brienza, and G. Anastasi, "Ieee 802.15. 4e: A survey," *Computer Communications*, vol. 88, pp. 1–24, 2016.
- [3] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [4] J. Buford, H. Yu, and E. K. Lua, *P2P networking and applications*. Morgan Kaufmann, 2009.
- [5] S. Duquenooy, A. Elsts, B. Al Nahas, and G. Oikonomo, "Tsch and 6tisch for contiki: Challenges, design and evaluation," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2017, pp. 11–18.
- [6] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquenooy, and P. Thubert, "Ietf 6tisch: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 595–615, 2019.
- [7] L. Doherty, J. Simon, and T. Watteyne, "Wireless sensor network challenges and solutions," *Microwave Journal*, vol. 55, no. 8, pp. 22–34, 2012.
- [8] M. Mohamadi and M. R. Senouci, "Scheduling algorithms for ieee 802.15. 4 tsch networks: a survey," in *International Conference on Computer Science and its Applications*. Springer, 2018, pp. 4–13.
- [9] A. Karaagac, I. Moerman, and J. Hoebeke, "Hybrid schedule management in 6tisch networks: The coexistence of determinism and flexibility," *IEEE Access*, vol. 6, pp. 33 941–33 952, 2018.
- [10] A. da Silva, F. Teixeira, R. Lage, L. Ruiz, A. Loureiro, and J. Nogueira, "Using a distributed snapshot algorithm in wireless sensor networks," in *The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003. FTDCS 2003. Proceedings.*, 2003, pp. 31–37.
- [11] M. Ojo and S. Giordano, "An efficient centralized scheduling algorithm in ieee 802.15. 4e tsch networks," in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2016, pp. 1–6.
- [12] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, "Traffic aware scheduling algorithm for reliable low-power multi-hop ieee 802.15. 4e networks," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)*. IEEE, 2012, pp. 327–332.
- [13] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial iot applications using ieee802. 15.4 e tsch," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655–3666, 2013.
- [14] R. Diestel, "Graph theory 3rd ed," *Graduate texts in mathematics*, vol. 173, 2005.
- [15] S. C. Ergen and P. Varaiya, "Tdma scheduling algorithms for wireless sensor networks," *Wireless networks*, vol. 16, no. 4, pp. 985–997, 2010.
- [16] R. C. Tavares, M. Carvalho, E. P. M. C. Júnior, E. d. B. e Silva, M. A. M. Vieira, L. F. M. Vieira, and B. Krishnamachari, "Fwb: Funneling wider bandwidth algorithm for high performance data collection in wireless sensor networks," *Computer Communications*, vol. 148, pp. 136–151, 2019.
- [17] N. d. S. R. Junior, M. A. M. Vieira, L. F. M. Vieira, and O. Gnawali, "Dual radio networks: Are two disjoint paths enough?" *IEEE Internet of Things Magazine*, vol. 4, no. 1, pp. 67–71, 2021.
- [18] F. Righetti, C. Vallati, G. Anastasi, and S. Das, "Performance evaluation the 6top protocol and analysis of its interplay with routing," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2017, pp. 1–6.
- [19] M. Osman and F. Nabki, "Oscar: An optimized scheduling cell allocation algorithm for convergecast in ieee 802.15. 4e tsch networks," *Sensors*, vol. 21, no. 7, p. 2493, 2021.
- [20] S. Duquenooy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled tsch," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 337–350.
- [21] S. Kim, H.-S. Kim, and C. Kim, "Alice: Autonomous link-based cell scheduling for tsch," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, 2019, pp. 121–132.