CONTRIBUIÇÃO PARA O AUMENTO DA CONFIABILIDADE NA ENTREGA DE PACOTES EM REDES DE SENSORES SEM FIO MULTISSALTOS

FABŔICIO N.DE GODOI[1], GUSTAVO W. DENARDIN[1], CARLOS H. BARRIQUELO[2]

[1] Programa de Pós-Graduação em Engenharia Elétrica e [2] GEDRE Inteligência em Iluminação
[1] Universidade Tecnológica Federal do Paraná, Pato Branco e [2] Universidade Federal de Santa Maria, Santa Maria
E-mails: fabricio.n.godoi@gmail.com, gustavo@utfpr.edu.br, barriquello@gmail.com

Abstract— The Wireless Sensor Network has been used to collect data in the field and forward it to data collectors. However, this networks relies on unreliable communication medium that is susceptible to noise and interference, which results in loss of information through the route to the data aggregator. Therefore, this work describes a new communication method named µNet, which aims to guarantee the delivery of packages in multi-hops networks by using a peer-to-peer transmission acknowledges control messages. Besides that, the µNet is compatible with constrained devices, with limited processors and memories, used in these networks. With this method, it is shown that it is possible to achieve better reliability in noisy environments than with others well-known methods.

Keywords— Wireless Sensor Networks, Reliability, Communication Protocol, Buffers.

Resumo— As Redes de Sensores Sem Fio têm sido utilizadas para coletar informações dos ambientes e enviá-las a um agregador de informação. Entretanto, essas redes utilizam um meio de comunicação que é suscetível a ruídos e interferências, de modo que as informações da rede são perdidas em suas rotas até o agregador de dados. Portando, esse trabalho descreve um método de comunicação, denominado µNet, que possui objetivo de garantir a entrega de informações em rede multissaltos, ao utilizar um sistema de confirmação de transmissão ponto-a-ponto. Além disso, o µNet leva em consideração que os dispositivos utilizados nessas redes possuem capacidades limitadas de processamento e memória, tornando-o compatível. Com esse método é demonstrado que é possível obter maior confiabilidade nas transmissões de informações em ambientes ruidosos do que com outros métodos conhecidos.

Palavras-chave— Redes de Sensores Sem Fio, Confiabilidade, Protocolo de Comunicação, *Buffers*.

1 Introdução

Redes de Sensores Sem Fio (RSSF) é uma tecnologia capaz de recuperar informações dos ambientes no qual está inserido, e enviá-las a uma central de processamento através de um meio de comunicação sem fio, conforme exemplificado pela Figura 1. A infraestrutura de RSSF é composta de elementos de medição, processamento computacional e de comunicação, que possibilitam a observação e o controle de características especificas de seus cenários (Sohraby, Minoli e Znati, 2007). Essa tecnologia tem sido amplamente aplicada com objetivo de aperfeiçoar o uso de recursos (Mahoor, Salmasi e Najafabadi, 2017), (Ullah, Faheem e Kim, 2017), melhorar qualidade de vida (Puvaneshwari e Vijayashaarathi, 2016), melhorar produtividade industrial (Cheng et al., 2016), qualidade ambiental (Torii, Otsuka e Ito, 2016), entre outros.

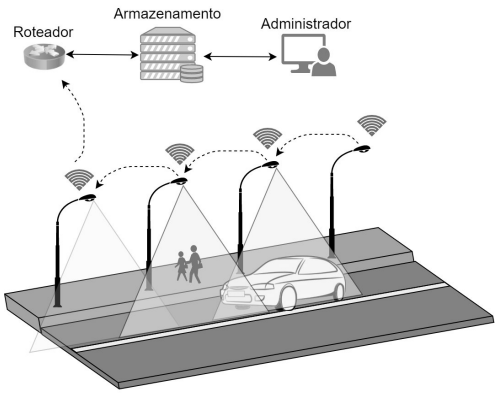

Figura 1. Representação de RSSF para sistemas de iluminação pública.

Aplicações, baseadas em RSSF, para controlar o sistema de iluminação pública têm sido implantadas em Cidades Inteligentes (*Smart Cities*) para reduzir consumo energético em áreas com pouca movimentação. Quando não há detecção de objetos (veículos, pedestres, etc.), as luzes são desligadas evitando desperdício de energia, e permanecendo ligadas somente enquanto algum objeto estiver na área (Mahoor, Salmasi e Najafabadi, 2017; Knobloch e Braunschweig, 2017). Além disso, aplicações para monitorar o consumo energético e controlar sua distribuição, têm sido desenvolvidas com base em redes de comunicação. Essas aplicações, conhecidas como *Smart Grids*, têm utilizado o conceito de RSSF para coletar a medição automática de consumo energético (ARM – *Automatic Meter Read*ing). Para

mensurar todos os componentes da rede elétrica, é necessário instalar sensores em cada consumidor e coletar suas informações por uma rede de comunicação (Ullah, Faheem e Kim, 2017; Hosni e Hamdi, 2016).

Apesar da aplicabilidade, as RSSFs são desenvolvidas utilizando dispositivos com recursos limitados, o que restringe sua funcionalidade e desempenho. Além disso, os dispositivos utilizados para RSSF, em sua maioria, são situados em ambientes ruidosos que diminuem seu desempenho. Esses ambientes frequentemente provêm fontes de ruídos, terra, poeira, umidade, interferência eletromagnética, etc., que causam falhas no sistema de comunicação e modos de operação (Sohraby, Minoli e Znati, 2007; Gungor e Hancke, 2009). Por causa desses fatores, metodologias foram propostas com o intuito de melhorar a confiabilidade das RSSF, como: aprimorar protocolos de rede (Yunus et al., 2011; Masirap et al., 2016); ajustar o buffer da rede (Al-Anbagi, Khanafer e Mouftah, 2013; Omondi et al., 2015); alternar canais de transmissão do rádio (Decker et al., 2008); adicionar redundância ao sistema (Munir, Antoon e Gordon-Ross, 2015), etc. Apesar desses métodos serem eficientes, ainda há falhas nesse tipo de rede, o que indica necessidade de melhorá-las. Portanto, esse trabalho tem como objetivo aprimorar a qualidade de serviço em RSSF, desenvolvendo um protocolo de comunicação que fornece melhor confiabilidade na transmissão de informações fim-a-fim, do que os protocolos tradicionalmente utilizados em RSSFs. Para isso, é criado um novo método de comunicação para dispositivos com recursos limitados (processador, memória, periféricos de comunicação e energia), denominado µNet, que implementa mensagens de confirmação em transmissões ponto-a-ponto.

2 Redes de Sensores Sem Fio

Assim como descrito por Sohraby, Minoli e Znati (2007), a maioria das RSSFs são compostas por centenas ou milhares de dispositivos, para atender os requisitos das aplicações. Portanto, de forma a ser possível implantar os projetos, é necessário que os dispositivos sejam de baixo custo financeiro, o que implica que seus processadores, memórias, periféricos de comunicação e energia disponível para operação são limitados. Além disso, RSSFs são implantadas em ambientes ruidosos e, em sua maioria, sem fontes de energia, dependendo somente de baterias para funcionar. Tais condições acarretam na necessidade de sistemas energeticamente eficientes e robustos a interferências com os recursos disponíveis.

Ao mesmo tempo, RSSF frequentemente utilizam o espectro de rádio de 2,4 GHz para realizar suas comunicações, definida como espectro *Industrial*, *Scientific and Medical* (ISM). A banda ISM é um padrão de uso livre na maioria dos países, o que torna o desenvolvimento de projetos com tecnologia sem fio mais acessível. Consequentemente, isso ocasiona em sobrecarga no espectro de comunicação, conhecido como interferência ISM. Outros padrões de comunicação utilizados em cidades também pertencem ao espectro ISM, como o IEEE 802.11, conhecido como Wi-Fi. O trabalho em (Wagh, More e Kharote, 2015) demonstram os efeitos de dois padrões de comunicação sem fio operando em bandas de comunicação próximas e formas de reduzir suas interferências.

2.1 Estrutura de Rede

Por utilizar dispositivos de recursos limitados, a estrutura de rede para RSSF deve ser alterada para ser compacta e confiável. O protocolo IEEE 802.15.4 (IEEE Computer Society, 2015) é um padrão desenvolvido para dispositivos de baixa capacidade. Ele padroniza as camadas Física e de Controle de Acesso ao Meio (MAC – *Medium Access Control*), disponibilizando topologias de rede estrela, ponto-a-ponto e por agrupamento. São definidas frequências de rádios para cada tipo de aplicação, variando de 470 MHz até 2,4 GHz. Essa topologia de rede utiliza um dispositivo responsável por coletar e coordenar a rede, denominado Coordenador PAN (*Personal Area Network*). Cada dispositivo na rede possui um identificador (ID) único para endereçamento, que é utilizado para realizar as transmissões entre dispositivos.

2.2 Protocolos de Transmissão

Apesar do protocolo TCP (*Transmission Control Protocol*) ser utilizado em sistemas computacionais convencionais como método mais confiável para entrega de dados na rede, seu modelo não é otimizado para RSSF. O TCP é um protocolo baseado em conexão, o qual para enviar e receber dados é necessário estabelecer uma conexão entre emissor e receptor com o método *Three-way Handshake*. Esse método troca mensagens de sincronismo e confirmação entre dispositivos fim-a-fim na rede antes de iniciar a transmissão de dados. Apesar do desempenho desse protocolo ser adequado para conexões de alta velocidade, ele é inapropriado para o padrão IEEE 802.15.4, que opera a uma taxa de comunicação de 250 kbps e com tamanho de pacote de 127 *bytes*, pois causa sobrecarga na rede com os métodos de controle, tornando-o um método de alta latência e com elevado cabeçalho de informação para RSSF (Sohraby, Minoli e Znati, 2007).

Em contrapartida, o protocolo UDP (*User Datagram Protocol*) possui um cabeçalho reduzido e métodos de comunicação de baixa latência. Entretanto, o UDP não fornece métodos de controle para garantir a entrega de informações, tornando-o não confiável. Portanto, novos protocolos baseados

no UDP foram desenvolvidos para torná-lo mais confiável (Yunus et al., 2011; Masirap et al., 2016).

O protocolo *Reliable* UDP (RUDP) (Bova e Krivoruchka, 1999; Mora, 2005) é um exemplo de metodologia que define uma subcamada de controle acima do protocolo UDP. O RUDP utiliza mensagens de confirmação de rede para garantir que a informação tenha sido transmitida corretamente. Ao receber um pacote de informação, o receptor emite um pacote de confirmação ao emissor, juntamente com a informação recebida. Quando o emissor recebe de volta seu pacote de informação, é realizada uma verificação de integridade e, se estiver correto, é emitido uma confirmação para o receptor para a informação ser roteada. Apesar de ser descrito como rascunho pela *Internet Engineering Task Force* (IETF) (Bova e Krivoruchka, 1999), tal metodologia não possuí padronização formal, de modo a não ser comum em aplicações de RSSFs.

Outro protocolo que pode ser utilizado como método de garantir entrega de informações é o CoAP (Shelby, Hartke e Bormann, 2014), que agrega serviços Web e métodos de controle de comunicação, para redes de dispositivos com recursos limitados (processamento, energia, redes, etc.). O CoAP implementa mecanismos de confiabilidade leves, sem recriar todas as funcionalidades do TCP, utilizando mensagens de confirmação fim-a-fim para confirmar se as mensagens foram enviadas, de forma parecida com o RUDP. Além disso, utiliza de espaçamentos exponenciais de espera entre transmissões que não houveram retorno da mensagem de confirmação (Shelby, Hartke e Bormann, 2014).

3 Solução Proposta

De forma a aprimorar a confiabilidade na entrega de pacotes em RSSF, o método proposto, denominado µNet, utiliza mensagens de confirmação em transmissões ponto-a-ponto, conforme a Figura 2. A primeira mensagem de confirmação (ACKR) tem o propósito de notificar o recebimento do pacote (PKT). A segunda mensagem (ACKN) notifica que o pacote foi devidamente alocado no *buffer* do receptor, com garantia de que o pacote que será devidamente encaminhado ao seu destino. A terceira mensagem, da mesma forma que a primeira, notifica ao receptor que a mensagem ACKN foi recebida pelo emissor, podendo liberar o espaço do buffer para armazenar novos dados, finalizando o processo de confirmação de recebimento. Todas as mensagens transmitidas nesse procedimento são efetuadas com limites de tempo e de tentativas, não bloqueando o sistema indefinidamente. Na Figura 3 demonstra-se o fluxo do µNet em cenários baseados em saltos, na qual dois pacotes, X e Y, são enviados à um receptor em comum. Nota-se que o pacote X foi devidamente entregue ao "Sensor 2", entretanto, quando roteado ao "Sensor n", houve falha, pois não há espaço disponível no buffer. Portanto, após aguardar um tempo predefinido para novas tentativas, o pacote X finalmente é entregue ao "Sensor n" na tentativa "m". Caso o número de tentativas atinja seu limite, então o pacote é finalmente descartado.

Os métodos de confirmação situam-se na camada MAC, sendo que a confirmação ACKR é realizada pela subcamada inferior do MAC, utilizando a mensagem existente do padrão IEEE 802.15.4, e a confirmação ACKN é realizada por uma subcamada superior de Link de Dados, de modo similar ao padrão ISO/IEC 8802-2:1998 (LLC – *Logical Link Control*). Desse modo, a solução proposta utiliza a mensagem de confirmação ACKR para garantir que a mensagem foi enviada corretamente ao módulo de comunicação do receptor e a mensagem ACKN para garantir que o pacote foi armazenado no dispositivo e está pronto para ser roteado. Portanto, para realizar o controle das transmissões, foi definida uma sequência de informações a serem adicionadas no cabeçalho MAC. O cabeçalho LLC utilizado nessa rede, demonstrado na Figura 4, foi criado de forma a utilizar somente um *byte* para indicar qual protocolo será responsável pelo processamento nessa subcamada (µNet, UDP, TCP, etc.). Se o Número LLC definido for o µNet, espera-se os quatro *bytes* subsequentes utilizados no controle de transmissão.

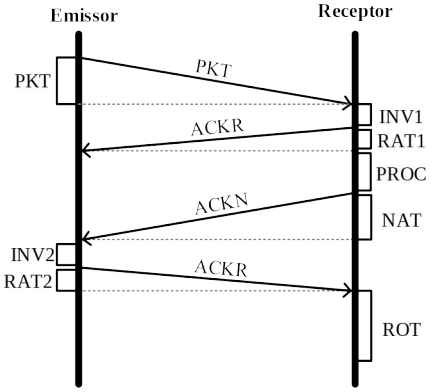

Figura 2. Transmissão ponto-a-ponto utilizando o método µNet.

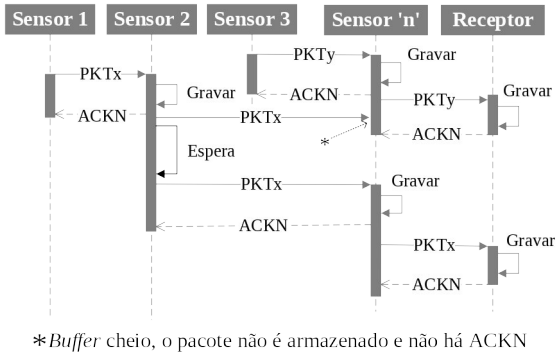

Figura 3. Transmissão de pacotes em cenário multissaltos com o método µNet.

Número LLC (1 *byte*)	Tipo de Pacote (1 *byte*)	Tamanho do Cabeçalho (1 *byte*)	Saltos Máximos (1 *byte*)	Próximo Cabeçalho (1 *byte*)

Figura 4. Cabeçalho LLC µNet.

As informações subsequentes são: Tipos de Pacotes, definidos em *Broadcast*, *Unicast*, Confirmação e *Multicast*; Tamanho de Cabeçalho indica o tamanho total do cabeçalho subsequente em *bytes*; Saltos Máximos é utilizado para limitar a quantidade de saltos que um pacote pode percorrer antes de ser descartado; Próximo Cabeçalho é utilizado para informar o tipo de pacote que está sendo transmitido, aplicação ou controle de rede.

Para realizar o controle de rotas dos dispositivos, foram utilizadas métricas de menor contagem de saltos e maior qualidade de sinal entre dois dispositivos. Cada dispositivo é configurado com um endereço de rede de oito *bytes*, contendo a sequência: endereço da PAN (PANID), endereço de rede (ADDR32) e endereço local do dispositivo (ADDR16). O endereço ADDR32 é utilizado para definir sub-redes na RSSF, enquanto o ADDR16 é o endereço MAC do dispositivo. Além disso, são utilizados dois *buffers*, um para rotas ascendentes (do coordenador para os diapositivos sensores) e outro para rotas descendentes (dos dispositivos sensores para o coordenador).

Para realizar o transporte de informações entre aplicações, é utilizado um *byte* para denotar a porta de origem, um *byte* para porta de destino e um *byte* para denotar o tamanho em *bytes* da informação anexada. Diferente do UDP, que utiliza dois *bytes* para as portas e tamanho do pacote, além de um campo de soma de verificação (Postel, 1980). Como o padrão IEEE 802.15.4 realiza a verificação de integridade do pacote, o campo de soma de verificação foi descartado da camada de transporte. Além disso, por ser uma RSSF com dispositivos de capacidade reduzida, reduziu-se o tamanho dos campos de portas e tamanho de pacote, pois não se espera que uma RSSF utilize mais do que 256 aplicações.

4 Método de Avaliação

A solução proposta foi desenvolvida para o sistema operacional BRTOS (*Brazilian Real-Time Operational System*) (Denardin e Barriquello, 2017) em conjunto com o protocolo IEEE 802.15.4 para controle da camada MAC. O código da solução proposta está disponível no repositório de Denardin e Barriquello (2017) e a versão adaptada para o simulador em Godoi, Denardin e Barriquelo (2017).

Para avaliar o método proposto foram utilizados diversos cenários de simulações, nos quais o sistema operacional Contiki OS foi definido como métrica de comparação, pois é um sistema especializado em RSSF utilizado em diversos estudos acadêmicos. Como base de comparação, o Contiki OS foi configurado com os protocolos mais recentes de controle de RSSFs: IEEE 802.15.4, 6LoWPAN, IPv6, RPL e UDP. Com essa configuração de protocolos, o Contiki OS pode atingir até 60 *bytes* de informação de cabeçalho, o qual é maior do que os 36 *bytes* necessários pelo µNet. Nota-se que o tamanho máximo aceitado pelo padrão IEEE 802.15.4 é de 127 *bytes*. Portanto, o conjunto de protocolos do Contiki OS utiliza até 47% do tamanho máximo, enquanto o µNet utiliza somente 28% para seu cabeçalho.

O Contiki OS foi configurado para utilizar o método de confirmação na camada MAC pelo padrão IEEE 802.15.4 e com rádio sem controle de ciclo de trabalho, deixando-o sempre ativo para enviar e receber informações. O RPL foi configurado com a função objetivo OF0 e o 6LoWPAN com o método HC06. Como o Contiki OS possui um simulador próprio para RSSF, denominado Cooja (Contiki OS Java), tal ferramenta foi utilizada para simular os cenários de comunicação. Apesar dessa ferramenta ser destinada ao Contiki OS, suas propriedades possibilitam a execução códigos genéricos compilados em linguagens C e Assembly. Portanto, para realizar as simulações foi utilizado um dos possíveis dispositivos emuladores fornecidos pela ferramenta, o Wismote com módulo de comunicação sem fio CC2520.

Entre os diversos cenários simulados, foi definido um que se aproxima dos sistemas de iluminação pública, utilizando 49 e 100 dispositivos, conforme demonstrado na Figura 6. Foram definidos cenários ideias, sem simulação de interferência, e outros com interferências variadas. Para controle dos cenários, demonstrado na Figura 6, utilizou-se o controlador UGDM (*Unit Disk Graph Medium*) do Cooja, o qual possibilita configurar o alcance de comunicação (área cinza escura interna) e o alcance de interferência (área cinza clara externa), assim como a taxa de confiabilidade em entrega de dados ponto-a-ponto. Para simular cenários com interferência e validar os métodos de transmissão de dados, foram utilizadas taxas de confiabilidade de transmissão ponto-a-ponto de 100% (sem interferência), 91% e 75%. As redes foram configuradas com um coordenador (dispositivo zero no canto inferior direito, conforme exemplo) e que cada dispositivo envie dados predefinidos de 32 *bytes*, de modo que possam ser armazenados em um único pacote de transmissão (127 *bytes*).

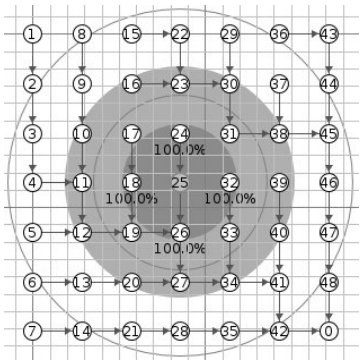

Figura 5. Exemplo de cenário de rede com 48 sensores e um coordenador no simulador Cooja.

Como o Contiki OS e o BRTOS não utilizam confirmações de entrega fim-a-fim, utilizou-se de intervalos entre as transmissões sucessivas de dados, de modo a permitir que a rede tenha tempo de enviar a informação sem interferência da própria rede. Além disso, garantiu-se que os dispositivos não iniciem ao mesmo tempo. O processo de análise de desempenho verifica a confiabilidade fim-a-fim da rede, ao calcular a razão entre a quantidade de pacotes entregues no coordenador pela quantidade total de pacotes gerados na rede. Portanto, cada dispositivo sensor é configurado para enviar 100 pacotes de dados com intervalos entre transmissões pré-definidos. Para avaliar o desempenho de confiabilidade dos métodos de comunicação, os intervalos entre transmissões utilizadas iniciam em no máximo em 16 s até no mínimo de 1 s. Antes de iniciar as transmissões de dados na rede, determinou-se que cada dispositivo deve ter pelo menos uma rota definida ao coordenador e, após todos dispositivos possuírem ao menos uma rota, é aguardado um tempo de espera de estabilização da rede de 15 s, de forma a evitar concorrência de acesso ao meio pelas mensagens de controle. Ao final das transmissões, aguarda-se um tempo para que o buffer da rede se esvazie, evitando finalizar a simulação com pacotes ainda sendo roteados.

O µNet foi configurado com *buffer* de capacidade unitária (um único pacote) e com no máximo de 30 retransmissões com intervalo de espera de 50 ms, gerenciadas pela camada LLC do µNet.

Por fim, validou-se ambos os sistemas operacionais com a adição do protocolo CoAP. Como o CoAP possui métodos de confirmação fim-a-fim para entrega de mensagens, foi utilizado um intervalo de 1 s entre as transmissões, somente para garantir que os sistemas não sobrecarreguem com tentativas de enviar mensagens. O CoAP foi configurado com intervalos entre transmissões e quantidades de tentativas conforme especificado em seu documento (Shelby, Hartke e Bormann, 2014).

5 Análise de Desempenho

Para descobrir os intervalos mínimos de transmissão dos pacotes de dados, foram analisados os tempos de transmissão ponto-a-ponto necessários para enviar uma informação de 32 *bytes* com o BRTOS e com o Contiki OS. O tempo total é calculado pela diferença entre o tempo de inicio de transmissão do pacote pelo emissor e pelo tempo de inicio de retransmissão do mesmo pacote pelo primeiro receptor. Dessa forma, é possível verificar o tempo total que um pacote necessita para ser transmitido por múltiplos saltos. Foram necessários 6,8 ms com o método µNet no BRTOS e 5,59 ms com o conjunto de protocolos no Contiki OS. Portanto, o Contiki OS é 18% mais rápido do que o BRTOS para enviar uma informação ponto-a-ponto, pois o µNet necessita transmitir mais mensagens de confirmação. Com os tempos de transmissão ponto-a-ponto é possível definir o tempo de transmissão para qualquer dispositivo na rede, ao calcular o produto do tempo ponto-a-ponto com a quantidade de saltos necessários até o destinatário.

Esses valores possibilitam determinar um intervalo aproximado de tempo mínimo necessário entre transmissões de dados para a rede. No exemplo citado na Figura 6, o µNet necessita de aproximadamente 82 ms para que o dispositivo mais longe do coordenador, 12 saltos, entregue sua informação ao coordenador, enquanto o Contiki OS necessita de 67ms. Portanto, para que os 48 dispositivos enviem ao menos um pacote de dado sem interferência da própria rede, estimasse um intervalo aproximado mínimo entre transmissões de 4s para o método µNet e de 3,2s para o Contiki OS.

Os resultados de confiabilidade do sistema BRTOS com µNet e Contiki OS com UDP, IPv6, RPL e 6LoWPAN foram separados em três gráficos, por nível de interferência utilizada, conforme as Figuras 6, 7 e 8. As linhas do gráfico denominada B49 e B100 representam as taxas de confiabilidade do BRTOS com µNet para cenários com 49 e 100 dispositivos, respectivamente. As linhas C49 e C100 são os resultados do conjunto de protocolos utilizado no Contiki OS nos cenários com 49 e 100 dispositivos, respectivamente.

Os resultados da Figura 6 demonstram a confiabilidade de entrega de informação fim-a-fim em cenários sem simulação de interferência. É possível notar que tanto no método proposto, quanto o Contiki OS, atingiram confiabilidades elevadas, quanto maior o intervalo entre transmissões. Isso ocorre, pois não ocorram colisão nas transmissões pela concorrência de acesso ao meio. Entretanto, é notável que o Contiki OS manteve-se mais confiável à medida que o intervalo reduz, pois o BRTOS necessita de mais tempo para realizar as transmissões ponto-a-ponto. Portanto, em cenários ideais, o Contiki OS apresenta melhor confiabilidade para sistemas que emitem mais informações do que a rede pode suportar.

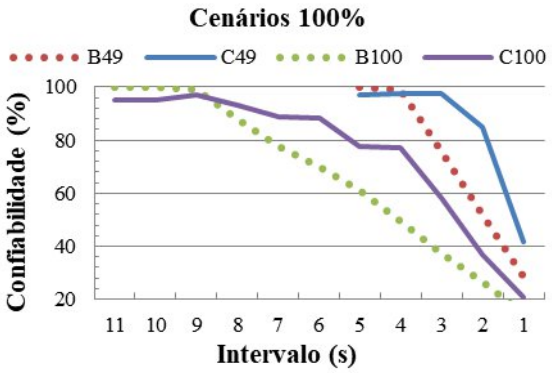

Figura 6. Confiabilidade dos protocolos em cenários com 100% de garantia de entrega ponto-a-ponto.

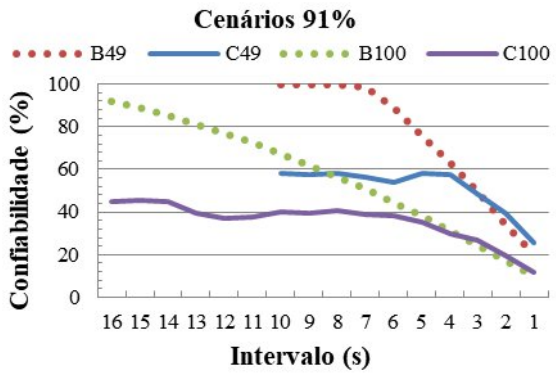

Figura 7. Confiabilidade dos protocolos em cenários com 91% de garantia de entrega ponto-a-ponto.

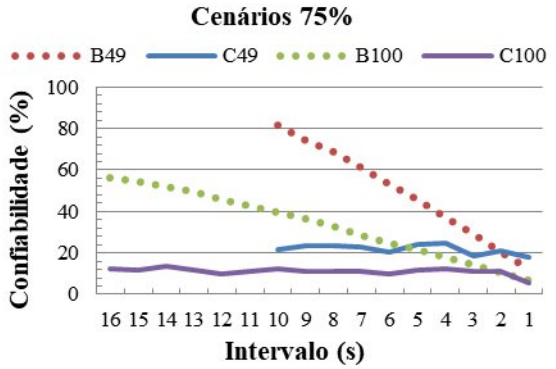

Figura 8. Confiabilidade dos protocolos em cenários com 75% de garantia de entrega ponto-a-ponto.

Os resultados apresentados nas Figuras 7 e 8 demonstram a confiabilidade em ambientes com simulações de interferência, com garantia de entrega de pacotes ponto-a-ponto de 91% e 75%, respectivamente. Nesses cenários é possível notar que o desempenho do BRTOS é muito superior do que o Contiki OS, à medida que o intervalo entre transmissões de dados aumenta, independente da quantidade de dispositivos. Nota-se que o BRTOS tem a capacidade de entregar todos os pacotes criados, se for disponibilizado o tempo necessário. Por outro lado, o Contiki OS não apresenta melhoras significativas quanto maior o intervalo.

As Tabela 1 apresenta as comparações de confiabilidade entre o Contiki OS e BRTOS com CoAP, denominados C-CoAP e B-CoAP, respectivamente. São analisadas as confiabilidades para enviar 100 pacotes de dados e o tempo necessário para enviá-las. Por fim, compara-se o BRTOS com CoAP e sem CoAP (B-µNet), conforme valores demonstrados na Tabela 2.

Tabela 1. Confiabilidade dos sistemas operacionais com CoAP.

Cenários	C-CoAP Confia.	C-CoAP Tempo	B-CoAP Confia.	B-CoAP Tempo
49 100%	98,50%	98 min	90,75%	31 min
49 91%	86,81%	97 min	90,21%	49 min
49 75%	53,52%	97 min	84,77%	70 min
100 100%	99,67%	99 min	75,17%	62 min
100 91%	78,12%	99 min	69,36%	83 min
100 75%	34,61%	99 min	58,14%	86 min

Tabela 2. Comparação BRTOS com e sem CoAP.

Cenários	B-CoAP Confia.	B-CoAP Tempo	B-µNet Confia.	B-µNet Tempo	B-µNet Intervalo
49 100%	90,75%	31 min	98,96%	7 min	4 s
49 91%	90,21%	49 min	99,71%	16 min	10 s
49 75%	84,77%	70 min	81,44%	16 min	10 s
100 100%	75,17%	62 min	99,51%	19 min	11 s
100 91%	69,36%	83 min	81,60%	27 min	16 s
100 75%	58,14%	86 min	52,03%	27 min	16 s

Ao utilizar o CoAP no Contiki OS, observou-se um acréscimo de confiabilidade entre 20% e 30%, ao custo de um tempo muito maior de comunicação. Por outro lado, o BRTOS perdeu confiabilidade em cenários que a interferência simulada era menor, ao custo de um tempo maior de comunicação. Entretanto, o BRTOS com CoAP ainda obteve melhores resultados de confiabilidade com menor tempo de comunicação do que o Contiki OS em cenários com interferência simulada. Verifica-se que apesar de acrescentar mensagens de confirmação fim-a-fim, o BRTOS teve menor desempenho com o CoAP, pois a adição da mensagem de confirmação fim-a-fim sobrecarrega o meio de comunicação. Portanto, se for possível ajustar os intervalos entre transmissão de dados da aplicação, é possível obter maior confiabilidade e tempo menor de comunicação da rede com o BRTOS ao utilizar somente o µNet.

6 Conclusão

Neste artigo foi proposto um novo método de comunicação (µNet) responsável por gerenciar transmissões de pacotes ponto-a-ponto, com o objetivo de melhorar a confiabilidade de entrega de informações em RSSFs. Comparando-o com os métodos adotados pelo Contiki OS, é demonstrado que a solução apresenta aproximadamente 18% de acréscimo no tempo de comunicação ponto-a-ponto, devido a implementação do sistema de confirmação. Por causa disso, o desempenho do Contiki OS foi melhor a medida que o intervalo de transmissões reduzia em cenários ideais. Entretanto, ao utilizar ambas as metodologias em cenários ruidosos, o BRTOS com µNet extrapola a confiabilidade do Contiki OS, sendo capaz de manter a confiabilidade de até 100% em cenários com 91% de garantia de entrega ponto-a-ponto. Portanto, o método proposto é

mais adequado para redes afetadas por fontes de interferências se a taxa de geração de pacotes puder ser ajustada.

Apesar da abordagem proposta acrescentar um atraso nas comunicações ponto-a-ponto, ela apresenta maior confiabilidade em entrega de informações fim-a-fim em cenários ruidosos do que a abordagens utilizadas atualmente. Essa característica torna o método adequado para aplicações em cidades inteligentes, que são sistemas tolerantes a atrasos e inseridos em meios com diversas fontes de interferências, como redes Wi-FI e Bluetooth.

Como trabalhos futuro, planeja-se avaliar o consumo de energia com essa abordagem e o desempenho ao utilizar *buffers* de tamanhos variados. Além disso, validar valores ideais para intervalos entre transmissões e quantidade de retransmissões ponto-a-ponto necessárias.

Agradecimentos

Os autores agradecem à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), à Fundação Araucária, ao FINEP (Financiadora de Estudos e Projetos) e à UTFPR pelo apoio financeiro dessa pesquisa.

Referências Bibliográficas

Al-Anbagi, I., Khanafer, M. e Mouftah, H. T. (2013). "MAC finite buffer impact on the performance of cluster-tree based WSNs". IEEE International Conference on Communications, p. 1485–1490.

Bova, T. e Krivoruchka, T. (1999). "Reliable UDP protocol". https://tools.ietf.org/html/draft-ietf-sigtran-reliable-udp-00, Novembro.

Cheng, B., Zhao, S., Wang, S. e Chen, J. (2016). "Lightweight Mashup Middleware for Coal Mine Safety Monitoring and Control Automation". IEEE Transactions on Automation Science and Engineering, v.14, n. 2, p. 1245–1255.

Decker, E. B., Rajendran, V., Obraczka, K. e Garcia-Luna-Aceves, J. J. (2008). "The Multi-Channel Flow-Aware Medium Access Control Protocol for Wireless Sensor Networks". IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC).

Denardin, G. W. e Barriquello, C. H. (2017). "BRTOS: Brazilian Real-Time Operating System". https://github.com/brtos, Novembro.

Godoi, F. N. de, Denardin, G. W. e Barriquello, C. H.
(2017). "uNet for Cooja". https://www.github.com/fabricio-godoi/unet4cooja, Novembro.

Gungor, V. C. e Hancke, G. P. (2009). "Industrial Wireless Sensor Networks : Challenges, Design Principles, and Technical Approaches". IEEE Transactions on Industrial Electronics, v. 56, n. 10, p. 4258–4265.

Hosni, I. e Hamdi, N. (2016). "Cross layer optimization of end to end delay in WSN for smart grid communications". 2016 International Symposium on Signal, Image, Video and Communications, ISIVC 2016, p. 217–223.

IEEE Computer Society (2015). "IEEE Standard for Low-Rate Wireless Networks, Amendment 2: Ultra-Low Power Physical Layer". The Institute of Electrical and Electronics Engineers, Inc. http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf, Novembro.

Postel, J (1980). User Datagram Protocol. Internet Engineering Task Force. Disponível em: <https://tools.ietf.org/html/rfc768>. Acesso em: 28 mar. 2018.

Knobloch, F. e Braunschweig, N. (2017). "A Traffic-Aware Moving Light System Featuring Optimal Energy Efficiency". IEEE Sensors Journal, v. 17, n. 23, p. 7731-7740.

Mahoor, M., Salmasi, F. R. e Najafabadi, T. A. (2017). "A Hierarchical Smart Street Lighting System With Brute-Force Energy Optimization". IEEE Sensors Journal, v. 17, n. 9, p. 2871–2879.

Masirap, M., Amaran, M. H., Yussoff, Y. M., Rahman, R. A. e Hashim, H. (2016). "Evaluation of Reliable UDP-Based Transport Protocols for Internet of Things (IoT)". IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), p. 200–205.

Mora, O. (2005). "Reliable Transport Layer Protocol in Low Performance 8-bit Microcontrollers". US 2005/0129064 A1. http://www.google.sr/patents/US20050129064, Novembro.

Munir, A., Antoon, J. e Gordon-Ross, A. (2015). "Modeling and Analysis of Fault Detection and Fault Tolerance in Wireless Sensor Networks". ACM Transactions on Embedded Computing Systems, v. 14, n. 1.

Omondi, F. A., Shah, P., Gemikonakli, O. e Ever, E. (2015). "An Analytical Model for Bounded WSNs with Unreliable Cluster Heads and Links". 40th Annual IEEE Conference on Local Computer Networks, p. 201–204.

Puvaneshwari S e Vijayashaarathi S (2016). "Efficient Monitoring system for cardiac patients using Wireless Sensor Networks (WSN)". 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), p. 1558–1561.

Shelby, Z., Hartke, K. e Bormann, C. (2014). "The Constrained Application Protocol (CoAP)". Internet Engineering Task Force (IETF).

Sohraby, K., Minoli, D. e Znati, T. (2007). "Wireless Sensor Networks: Technology, Protocols, and Applications". New Jersey: John Wiley & Sons, Inc.

Torii, Y., Otsuka, T. e Ito, T. (2016). "A diversity sensor connection capability WSN for disaster information gathering system". IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS).

Ullah, R., Faheem, Y. e Kim, B. S. (2017). “Energy and Congestion-Aware Routing Metric for Smart Grid AMI Networks in Smart City”. IEEE Access, p. 13799–13810.

Wagh, S. S., More, A. e Kharote, P. R. (2015). “Performance Evaluation of IEEE 802.15.4 Protocol under Coexistence of WiFi 802.11b”. Procedia Computer Science, v. 57, p. 745–751.

Yunus, F., Ismail, N. N., Ariffin, S. H. S., Shahidan, A. A., Fisal, N. e Yusof, S. K. S. (2011). “Proposed Transport Protocol for Reliable Data Transfer in Wireless Sensor Network (WSN)”. 4th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO).

.