

# Um Protocolo de Roteamento Dinâmico para Redes de Comunicação por Luz Visível

Luiz Mendes<sup>1</sup>, Marcos A. M. Vieira<sup>2</sup>, Luiz F. M. Vieira<sup>2</sup>, Alex B. Vieira<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora

<sup>2</sup>Departamento de Ciência da Computação – Universidade Federal de Minas Gerais  
luizmendes@ice.ufjf.br, {mmvieira,lfvieira}@dcc.ufmg.br, alex.borges@ufjf.edu.br

**Resumo.** *Comunicação por Luz Visível (VLC) apresenta diversos desafios como interferência e bloqueios criados por obstáculos. Ao mesmo tempo, vários ambientes oferecem uma rica infraestrutura de fontes de luz, que podem ser utilizadas para auxiliar na comunicação fim-a-fim. Neste trabalho, apresentamos D-VLC, um protocolo de roteamento dinâmico. D-VLC reage às interrupções de comunicação, construindo rotas alternativas na rede de luz visível sem a necessidade de prévio conhecimento da topologia ou até mesmo dos dispositivos que a compõem. O novo protocolo foi implementado e avaliado em um ambiente real utilizando a plataforma embarcada OpenVLC. Os resultados obtidos mostram que, usando o D-VLC, a rede foi capaz de se adaptar a mudanças dinâmicas na comunicação, como sombras e obstáculos, atingindo taxas de até 3.75 Kbps –limitada apenas pela tecnologia VLC adotada– em configurações de múltiplos saltos enquanto apresenta baixa sobrecarga na rede (<1%).*

**Abstract.** *Visible Light Communication (VLC) presents several challenges, such as interference and blockages created by obstacles. At the same time, several environments offer a rich infrastructure of light sources, which can be used to aid end-to-end communication. In this work, we present D-VLC, a dynamic routing protocol. D-VLC reacts to communication interruptions, constructing alternative routes in the network of visible light, without the need for previous knowledge of the topology or even of the devices that comprise it. We implemented and evaluated the new protocol in a real-world environment using the OpenVLC embedded platform. Our results show that, by using D-VLC, the network was able to adapt to dynamic changes in communication such as shadows and obstacles, achieving data rates of up to 3.75 Kbps –limited by the VLC technology used– in multi-hop configuration while presenting low overhead (<1%).*

## 1. Introdução

Comunicação de luz visível (VLC) é um tipo de comunicação que usa a faixa de luz visível do espectro eletromagnético (variando de 380 nm a 750 nm) para transmitir dados [Vieira et al. 2017]. Durante a última década, VLC se tornou muito popular tanto na academia quanto na indústria. Em especial, parte do interesse em uma nova forma de comunicação sem fio se dá pelos desafios atuais, como por exemplo, a lotação do espectro eletromagnético destinada ao Wi-Fi (*spectrum crunch*)<sup>1</sup> [De Vries et al. 2014]. Nesse

---

<sup>1</sup>Will we ever face a wireless “*spectrum crunch*”? - <http://www.bbc.com/future/story/20131014-are-we-headed-for-wireless-chaos>, 2013.

sentido, o uso de luz visível para trocar informações foi considerado para uma ampla gama de aplicações tais como comunicação LED-LED [Matheus et al. 2018], sistemas veiculares [Okada et al. 2009], sistemas de localização [Li et al. 2014] e comunicação subaquática comunicação [Kaushal and Kaddoum 2016].

De forma geral, sistemas VLC apresentam dois componente principais: um transmissor e um receptor. Os diodos emissores de luz (LEDs) são comumente usados como transmissores. Porém, até mesmo a luz natural (i.e., a luz solar) pode ser usada para este fim [Wang et al. 2016]. O transmissor é responsável por modular a intensidade da luz para transmitir dados. No lado do receptor, fotossensores são usados para capturar a luz diretamente. Os LEDs transmitem luz apenas em um campo de visão. Assim, o receptor deve estar alinhado com o transmissor, o que limita a comunicação por luz visível a uma determinada linha de visão entre essas duas entidades.

Existem muitos problemas em relação à adoção de sistemas VLC em ambientes internos. O ambiente dinâmico criado pelas múltiplas fontes de luz e a interferência devido a fatores como sombreamento e obstáculos podem ter um grande impacto no desempenho da comunicação [Xiang et al. 2014]. Por outro lado, o ambiente dinâmico e a infraestrutura de iluminação já existente oferecem oportunidades para protocolos cooperativos de múltiplos saltos. Em suma, tão logo um dispositivo perca a linha de visão durante a comunicação, ele pode recorrer aos demais dispositivos para fazer uma comunicação indireta. Dessa forma, a comunicação fim-a-fim não é interrompida.

Nesse sentido, este trabalho apresenta D-VLC, um protocolo de roteamento dinâmico (*Dynamic Routing Protocol for Visible Light Communication*) com o objetivo de aumentar o desempenho de sistemas VLC em ambientes dinâmicos. D-VLC reage às interrupções de comunicação fim-a-fim, imposta por obstáculos, sombras ou mobilidade, construindo rotas alternativas na rede de luz visível, sem a necessidade de prévio conhecimento da topologia ou até mesmo dos dispositivos que a compõem. Existem trabalhos na literatura que abordam múltiplos saltos em VLC [Kim et al. 2016, Ahmad et al. 2017, Klaver and Zuniga 2015], sendo que se preocupam principalmente em possibilitar comunicação fim-a-fim sem linha de visão. Mais ainda, necessitam do conhecimento da topologia da rede e do uso de retransmissores (*relays*) específicos. O foco principal destes trabalhos é, muitas vezes, aumentar a distância entre os nós. Não lidam com os aspectos dinâmicos da rede; não identificam a quebra da comunicação e não requisitam, dinamicamente, cooperação de seus pares. Em suma, não se adaptam as mudanças do ambiente de comunicação, como o surgimento de sombras e obstáculos.

Nós implementamos o D-VLC no OpenVLC 1.0<sup>2</sup>, uma plataforma de código aberto, atual e bem conhecida, projetada para pesquisas em VLC. Avaliamos o novo protocolo em uma rede dinâmica, onde a comunicação é interrompida por interposição de obstáculos. Nossos resultados mostram que o uso do D-VLC possibilita comunicação fim-a-fim, mesmo em condições em que há obstáculos. Mais ainda, o novo protocolo apresenta baixa sobrecarga. De fato, o protocolo necessita de apenas 1% a mais de recursos, quando comparada a um protocolo que não trata comunicação com múltiplos saltos.

---

<sup>2</sup><http://www.openvlc.org/home.html>

## 2. D-VLC

O D-VLC faz uso do rico ambiente oferecido pela infraestrutura de iluminação existente em ambientes internos. Ele constrói e mantém caminhos dinâmicos para mitigar problemas de comunicação. Mais ainda, o protocolo utiliza recursos de múltiplas camadas (físico-enlace e redes) para oferecer conectividade à Internet.

O D-VLC é inspirado em protocolos de roteamento para MANETs (AODV, DSR). As redes de Comunicação por Luz Visível podem ter características semelhantes às redes Ad-hoc móveis, em especial, por considerar mobilidade, largura de banda limitada, falta de confiabilidade e limitações da camada física. Além desses desafios, obstáculos são fatores importantes em VLC. Eles podem surgir a qualquer momento e assim, interromper o campo de visão entre os dispositivos que estão se comunicando. Nesse sentido, são considerados protocolos reativos para redes Ad-hoc como bloco fundamental na construção de nossa abordagem. A principal ideia é se adaptar, dinamicamente, às mudanças na rede.

Toda vez que um nó deseja se comunicar com outro nó —que não está em sua tabela de roteamento— ele tenta construir uma rota. Se a rota é construída com sucesso, uma ligação bidirecional é criado entre eles. O D-VLC apenas mantém informações de roteamento entre os nós ativos. Ele não necessita ter conhecimento prévio de toda topologia e, portanto, tem uma vantagem considerável em termos de consumo de memória. Cada nó na rede possui uma estrutura de dados que representa uma tabela de rotas. Em nossa abordagem, criamos uma tabela de rotas simples para atender aos requisitos de comunicação dinâmica. A tabela 1 apresenta os campos da tabela de rotas do protocolo.

Campo	Descrição
Endereço de destino	Endereço IP do nó ente fim comunicante
Próximo salto	Endereço IP do vizinho que servirá como relay para a comunicação fim-a-fim
Número de sequência	Valor para assegurar que a rota é fresca e para evitar laços na rota
Tempo de expiração	Valor usado para controlar o estado (status) da roda atual
Contador de saltos	Número de saltos até o nó destino
Status	Status da rota, que pode variar entre “ativo”, “ocioso” e “quebrado”.

**Tabela 1. Parâmetros utilizados para avaliação**

A seguir, detalhamos as principais operações do D-VLC: a construção de rotas; a quebra (encerramento) de rotas e a detecção de rotas quebradas.

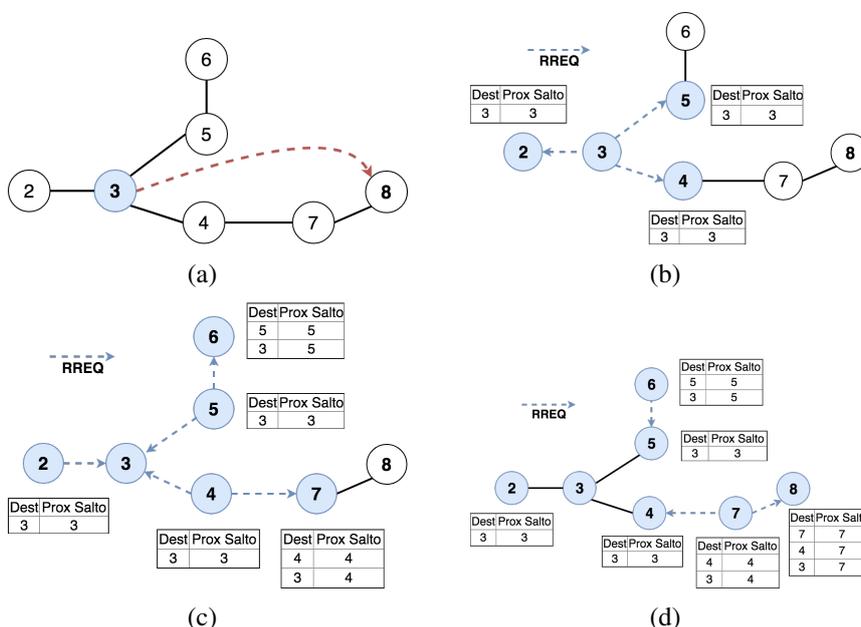
### 2.1. Construção de rotas

Toda vez que um nó recebe um pacote, ele procura o endereço de destino em sua tabela de roteamento. Se não encontrar, o processo de descoberta de rota será iniciado. Este processo envolve a transmissão de uma mensagem *RREQ* (requisição de rota), seguida pela espera por uma mensagem *RREP* (resposta de rota). Essas duas mensagens de roteamento compartilham a mesma estrutura, mas diferem na forma como são enviadas pela rede.

A construção de uma rota começa com um nó de origem transmitindo uma mensagem de solicitação de rota (*RREQ*) para encontrar um caminho válido para o destino. Cada nó intermediário salva uma rota em direção à origem e passa a mensagem adiante, por uma inundação a seus contatos. Quando o destino recebe o *RREQ*, ele cria uma rota completa em direção ao nó de origem, uma vez que o destino conhece, agora, todos os nós intermediários pelos quais a mensagem passou. O nó destino pode então enviar uma

mensagem de resposta de rota (*RREP*), que passa por cada nó presente no caminho exclusivo. Essa mensagem não é inundada na rede. Ela é enviada diretamente a cada nó pertencente a rota. Quando o nó de origem recebe o *RREP*, ele estabelece uma rota para o destino e começa a usá-la para enviar pacotes. Cada nó no caminho atua como nós de retransmissão nesse caso.

A descoberta de rota deve ser controlada para evitar *loops* e rotas obsoletas na rede. Para isso, toda vez que um nó envia uma mensagem *RREQ*, ele aguarda a mensagem *RREP* por um período de tempo. Se o nó falhar ao criar uma rota bidirecional para o destino, ele tentará novamente por um determinado número de vezes. Ao final das tentativas, se uma rota não for descoberta, o nó terá que aguardar outro período até tentar novamente.



**Figura 1. Descoberta de rota onde o nó 3 tenta se comunicar com o nó 8.**

Um exemplo de descoberta de rota é mostrado na figura 1. Para simplificação, supõe-se que cada nó já conhece sua vizinhança. O mecanismo de descoberta de rota inicia (Figura 1-a) quando um nó na rede (nó 3) deseja transmitir para um destino que não está atualmente em sua tabela de roteamento (nó 8). Nesse ponto, o nó 3 gera uma mensagem *RREQ* contendo as seguintes informações: (i) número máximo de saltos, (ii) contador de saltos, que é definido como 0 sempre que uma nova solicitação de rota é criada, (iii) endereço IP de origem, (iv) endereço IP de destino e (v) número de sequência, que é definido como 1 na primeira vez que uma rota é criada. O nó 3 transmite o *RREQ* (figura 1-b). Cada nó adjacente (neste caso, 2, 5 e 4) recebe a mensagem. Em seguida, eles (2, 5 e 4) verificam as informações contidas na mensagem *RREQ* e executam os procedimentos para verificar se a rota de entrada é nova. Se a rota é recente, eles adicionam o nó de origem à sua própria tabela de roteamento, juntamente com as informações contidas na mensagem. Pela figura 1-c, cada um desses nós (i) adiciona seus endereços IP, (ii) atualiza a contagem de saltos, (iii) anexa essas informações à mensagem *RREQ* e a transmite para a rede. É importante notar que, nesse ponto, o nó 3 também recebe o *RREQ*, mas

a mensagem é descartada porque o nó 3 já é a fonte do *RREQ* em questão. Os nós 6 e 7 também recebem o *RREQ* e atualizam suas informações de roteamento. Os nós 6 e 7 atualizam o *RREQ* e o transmitem (figura 1-d). Os nós 5 e 4 recebem a mensagem e a descartam. Finalmente, a mensagem chega ao endereço de destino (nó 8), que atualiza sua tabela de roteamento com o caminho inverso e prepara o *RREP*.

No caminho inverso, a mensagem *RREP* percorre todo o caminho de volta ao nó de origem. Assim, depois que o *RREQ* inunda a rede, todos os nós que a receberam possuem um caminho reverso para o endereço IP de origem. O nó 8, neste caso, após atualizar sua tabela de roteamento, cria um quadro *RREP* com: (i) número máximo de saltos, que é definido como a contagem de saltos fornecida pela mensagem *RREQ*, (ii) contagem de saltos, iniciado com 0, (iii) endereço IP de origem, (iv) endereço IP de destino, (v) número de sequência. Neste ponto, a camada MAC já conhece o endereço IP do próximo salto (que leva ao nó de destino) e atualiza o quadro com o próximo ID de salto. O nó 8 envia o *RREP*, que é então recebido pelo nó 7. A tabela de roteamento de 7 é atualizada com as informações de entrada, o *RREP* é atualizado com a adição do endereço IP e número de sequência e outro *RREP* é enviado. O nó 4 recebe o *RREP*, atualiza sua tabela de roteamento e repete o processo, adicionando o seu endereço IP à mensagem. Finalmente, a mensagem *RREP* chega ao destino. O nó 3 atualiza sua tabela de roteamento com as informações recebidas e cria com sucesso uma rota bidirecional em direção ao destino. Depois que uma rota bidirecional é criada entre dois nós na rede, os pacotes começam a ser encaminhados até chegar ao destino.

## 2.2. Quebra de rotas

A qualquer momento, dado um parâmetro específico, o estado de determinada rota pode ser definido como *quebrado*. Isso acontece se algum nó não conseguir encaminhar o pacote até o destino. Se isso acontecer, o nó gera uma mensagem *RERR* (erro de rota), que percorre o caminho até alcançar o nó de origem.

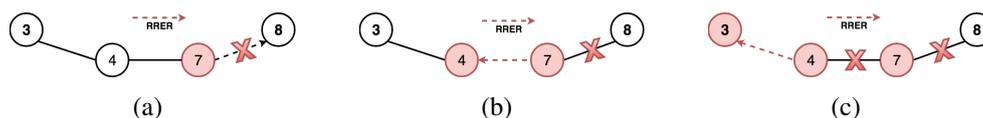


Figura 2. Exemplo de quebra de rota detectada pelo nó 7.

A figura 2 mostra uma quebra de rota, detectada pelo nó 7, enquanto 3 tentava enviar uma mensagem a 8. O nó 7 pode definir que o enlace entre ele e 8 está quebrado por qualquer mecanismo (e.g. temporização ou contagem de mensagens com erro). Ele marca sua rota para 8 como *Quebrado* e gera uma mensagem *RERR*. O nó 4 recebe a mensagem e verifica se o próximo salto para o endereço IP de destino inacessível é o endereço IP de origem contido no *RERR* de entrada. Nesse caso, o nó 4 marca sua rota para o nó 8 como *Quebrado* e atualiza o *RERR*, substituindo o IP anterior por seu próprio endereço IP, diminuindo o número máximo de saltos e mantendo o endereço IP inacessível. Em seguida, ele envia a mensagem de *RERR* para a rede. O nó 3 recebe a mensagem *RERR* e repete o processo feito pelo nó 4. Interrompe a rota para o nó 8, atualiza o *RERR*.

## 2.3. Detecção de rotas quebradas

Um nó deve ser capaz de detectar quando o enlace está quebrado. Sistemas VLC são muito suscetíveis a interferências e, qualquer sombra ou movimento pode bloquear a co-

municação. Para realizar a manutenção da rota, duas abordagens diferentes podem ser tomadas: usar contadores de referência ou usar tempo limite da rota. Contadores de referência dependem de informações baseadas na camada MAC ( número de pacotes perdidos ou errados). A segunda abordagem, apenas quebra a rota a cada intervalo de tempo.

Por restrições de espaço, consideramos apenas a abordagem por contadores de referência (RC). No D-VLC (RC), o protocolo de roteamento está sempre ciente da transmissão da camada MAC, devido ao caráter *cross-layer* do protocolo. Se o enlace for quebrado, eventualmente, o nó que envia o quadro não receberá seu *ACK*. D-VLC recolhe esta informação e toma a decisão de quebrar a rota. Se o *ACK* não for recebido por um determinado período de tempo, o enlace será considerado quebrado e o *RERR* será enviado para o fluxo ascendente. No entanto, se a camada MAC receber um *ACK* após uma certa quantidade de tentativas, a contagem será limpa e iniciada novamente.

### 3. Cenário Considerado e Metodologia de Avaliação

#### 3.1. Cenário Considerado

O protocolo proposto foi implementado e avaliado utilizando o OpenVLC 1.0 como base. O OpenVLC é uma das plataformas mais conhecidas do estado da arte para a realização de pesquisas na área. A versão utilizada é a mais estável e o seu desenvolvimento está ativo. No OpenVLC, a integração com a pilha de protocolos Linux permite a manipulação desde camada física a aplicações, além de validações com ferramentas bem aceitas pela comunidade. De fato, a plataforma permite a implementação e modificação de módulos de *kernel* do sistema operacional, de forma que características das camadas física e de enlace possam ser modificadas e novas funcionalidades adicionadas.

Os dispositivos foram configurados com parâmetros tradicionais para elementos do OpenVLC, conforme apresentados na tabela 2. Esses dispositivos utilizam modulação OOK (On-Off Keying) e CSMA/CA na camada de enlace. A rede considerada não possui nó central, e todos os nós são capazes de se comunicar uns com os outros, desde que possuam linha de visão, e possuem IPs fixos.

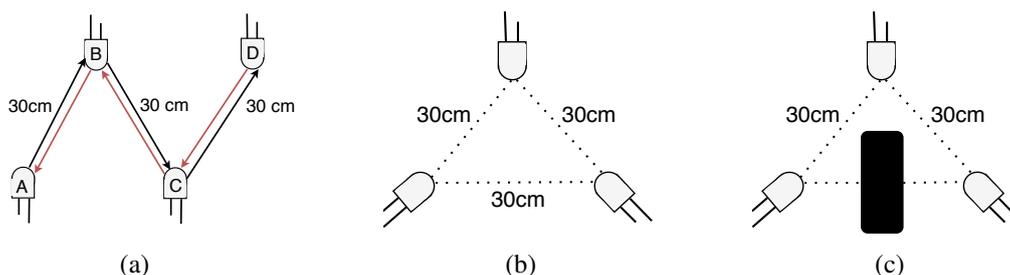
Parâmetro	Valor
Transmissor	LED vermelho (TLCR5800-ND)
Receptor	Fotodiodo (OPT101)
Taxa de amostragem	50 kHz
Tamanho do <i>payload</i> do quadro	255 Bytes
Luz ambiente	120 lx
Modulação	OOK
Protocolo do enlace	CSMA/CA

**Tabela 2. Parâmetros utilizados para avaliação**

Foram considerados dois cenários diferentes: estático e dinâmico. O cenário estático, apresentado na figura 3a, contém 4 nós. Esses nós são capazes de se comunicar apenas com seus vizinhos. O cenário dinâmico, apresentado na figura 3b,c apresenta três nós com visão total entre eles. Nesse cenário, um objeto interrompe a comunicação entre dois nós e assim, avaliamos a dinâmica do protocolo proposto.

#### 3.2. Metodologia de Avaliação

Em ambos os cenários, foi utilizada a ferramenta *iperf3* [Mortimer 2018] para avaliação do protocolo. O *iperf* foi configurado para enviar pacotes UDP de tamanho 0.8 KB. O



**Figura 3. Cenário de experimentação estático (a) e dinâmico (b) e (c).**

cenário estático possui até 4 nós, e o nó A se comunica com o nó D, de forma bidirecional, para que métricas relacionadas ao protocolo de roteamento possam ser avaliadas. No cenário dinâmico, um par de nós iniciam a comunicação, enviando pacotes UDP entre eles, de forma unidirecional (cliente-servidor). Conforme apresentado na figura 3-b, em um determinado momento, interrompemos a comunicação adicionando um obstáculo. O obstáculo interrompe o campo de visão entre os dois nós e aciona o protocolo de roteamento dinâmico, resultando em um cenário no qual um nó extra é usado para encaminhar pacotes para o destino. Para ambos os cenários, os experimentos duram 100 s e são repetidos 5 vezes. No caso do cenário dinâmico, o obstáculo é colocado a cada 30 segundos e, permanece entre os entes comunicantes também por 30 segundos.

Consideramos como principais métricas de interesse a vazão alcançada, o tempo de descoberta de rota e a sobrecarga imposta pelo roteamento. De fato, vazão é a métrica mais utilizada em trabalhos de cooperação MAC em sistemas VLC [Le et al. 2011, Wu 2012, Wang et al. 2014]. Formalmente, definimos essas métricas como segue:

- **Vazão (Kbps):** Relação entre a quantidade de dados corretos que o receptor recebe, por um período de tempo. A vazão não considera assim, pacotes de dados com erros.
- **Tempo de descoberta da rota:** Tempo decorrido entre o envio de uma requisição por criação de rota (*RREQ*) e o recebimento da rota correspondente (*RREP*).
- **Sobrecarga de roteamento:** Relação existente entre a quantidade de mensagens adicionais que os mecanismos de descoberta, manutenção e quebra de rotas impõem ao sistema VLC, e o número total de dados transmitidos durante o experimento. A sobrecarga é calculada através da relação do total de *bytes* enviados em um período de tempo e a porcentagem desses *bytes* que pertencem ao D-VLC.

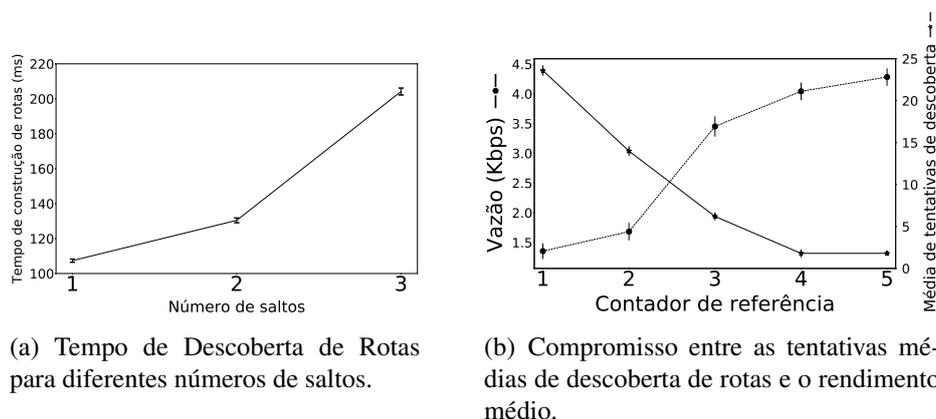
Os resultados apresentados são distribuições ou médias dos dados de cinco repetições de cada experimento, com intervalos de confiança de 95%.

#### 4. Resultados das Avaliações

Nesta seção, apresentamos os resultados das avaliações realizadas, considerando os cenários previamente apresentados. Inicialmente, apresentamos a avaliação em um cenário estático, onde testamos as características básicas do protocolo (seção 4.1). A seguir, mostramos o desempenho do protocolo em um cenário dinâmico (seção 4.2). Por fim, avaliamos a sobrecarga imposta pelo novo protocolo (seção 4.3).

#### 4.1. Avaliação de um cenário estático

Inicialmente, o cenário estático foi utilizado para avaliar o tempo de descoberta de rotas. Essa é uma métrica que não é influenciada pelas diferentes abordagens de manutenção de rota. É influenciada apenas pelas interações do protocolo desenvolvido e, pelo tamanho da rede. Consideramos assim, a configuração apresentada na figura 3-a, onde o nó A inicia um processo de descoberta de rota para outros nós na rede. Variamos os nós destinos e assim, avaliamos a influência da distância (saltos) entre os entes comunicantes. Nesta avaliação, o contador de referência para quebra de rotas não tem influência, uma vez que avaliamos apenas o processo inicial de estabelecimento de rota.



(a) Tempo de Descoberta de Rotas para diferentes números de saltos.

(b) Compromisso entre as tentativas médias de descoberta de rotas e o rendimento médio.

**Figura 4. Resultados obtidos para o cenário estático.**

A figura 4a apresenta o tempo médio de descoberta, e intervalo de confiança, em função da distância entre os entes comunicantes (número de saltos na rede). Como esperado, o tempo para construir rotas aumenta quando os dispositivos VLC estão mais distantes. De fato, a negociação de rota direta entre transmissor e receptor ( $A \rightarrow B$ ) demanda, na média, de 107 ms. Nesse caso, embora o nó de origem (A) tenha o endereço IP de destino (B) em sua tabela vizinhos, ele deve criar uma entrada na sua tabela de roteamento, indicando a direção do vizinho e os parâmetros associados a essa rota.

Na construção de rota entre  $A \rightarrow C$ , o tempo de negociação e descoberta de rota aumenta para, na média, 130 ms. Ao adicionar um salto, o tempo aumenta porque cada nó precisa processar o RREQ recebido, atualizar sua tabela de rotas com as informações recebidas e preparar a mensagem RREP com suas próprias informações anexadas a ela. Esse processo é recursivo e assim, o aumento de tempo fica ainda mais evidente quando um terceiro salto é adicionado no processo (rota  $A \rightarrow D$ ). Enquanto o tempo adicional necessário para descobrir a rota com um nó intermediando origem e destino foi de  $\sim 23$  ms (dois saltos), para dois nós intermediários (três saltos), esse tempo triplicou ( $\sim 70$  ms).

Nós avaliamos a vazão média alcançada pela rede ao se utilizar o protocolo proposto. Nesse sentido, avaliamos como configuração básica as apresentadas na tabela 2 e o cenário apresentado na figura 3-a, onde o nó A envia dados ao nó C por 150 s. Nós variamos o contador de referência utilizado para quebra de rotas e assim, também avaliamos o número de tentativas de descobertas de rotas observado durante o experimento.

A figura 4b mostra o compromisso entre as tentativas médias de descoberta de rotas e o rendimento médio do protocolo, considerando diferentes contadores de referên-

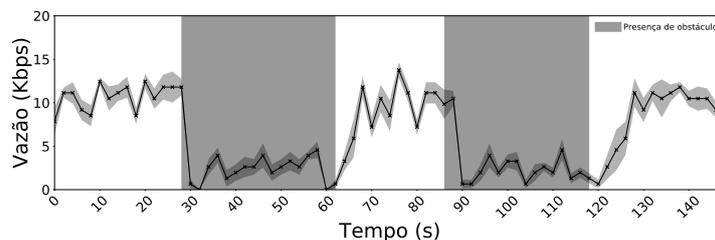
cia(RC). Quando o protocolo é muito sensível e reage ao menor problema, teríamos um número alto de tentativas de descoberta e construção de rota. Amarrado a isso, a vazão média observada durante o experimento seria baixa, uma vez que há grande perda de tempo se estabelecendo rotas (e não transmitindo de fato). De fato, quando o protocolo reage ao primeiro problema percebido camada MAC (i.e. contador de referência RC = 1), observamos mais de 24 tentativas de construção de rota durante nossos experimentos. A vazão também obteve seu pior valor, ficando abaixo de 1,5 Kbps na média.

Assim, postergar a reação do protocolo pode trazer ganhos, como evidenciado na figura 4b. De fato, aumentar o contador de referência dá à camada MAC uma margem de erro para trabalhar. Essa estratégia tem um impacto direto no desempenho, porque menos tentativas de descoberta de rota serão feitas, o que possivelmente bloqueará por menos tempo o protocolo. Nos experimentos realizados, o desempenho da rede se estabiliza para RC = 4, alcançando uma taxa de transferência média de até 4,3 Kbps.

Esse resultado deve ser melhor observado. Aumentar o contador de referência indiscriminadamente irá aumentar o tempo de reação da rede quando um obstáculo se impor entre as fontes comunicantes. Assim, o sistema VLC poderá experimentar longos tempos de espera, frutos de tentativas frustradas de transmissão por uma rota que efetivamente foi quebrada. Assim, a próxima seção avalia o protocolo em um cenário dinâmico, onde o tempo de reação deve ser cuidadosamente escolhido.

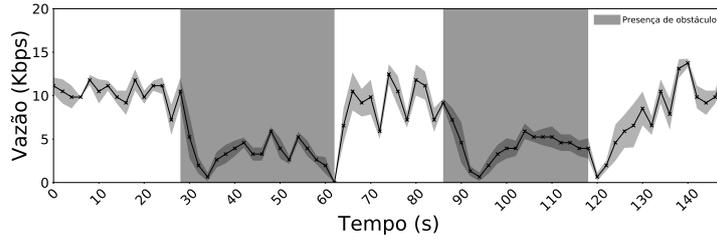
#### 4.2. Cenário dinâmico

Nessa seção, o protocolo proposto será avaliado em um cenário tal qual apresentado nas figuras 3-b e 3-c. Um nó inicia uma comunicação e, em determinado momento, um obstáculo interrompe a comunicação. Como descrevemos anteriormente, os obstáculos são colocados entre dois nós que estão se comunicando entre intervalos de 30 s a 60 s e de 90 s a 120 s. Esses momentos são sinalizados por regiões sombreadas nas figuras 5 e 6. Mais ainda, utilizamos um valor de referência RC=2 e RC=5. O primeiro valor parece ser uma boa razão entre vazão média do sistema e número de tentativas de descoberta de rota, apresentado na seção 4.1. O segundo, foi o valor com melhor desempenho obtido.



**Figura 5. Desempenho da rede utilizando D-VLC (RC = 2).**

A figura 5 apresenta a vazão média do sistema quando o contador de referência RC = 2. Por essa figura, durante a comunicação direta, a vazão média da rede é cerca de 11 Kbps. Como esperado, quando o obstáculo é adicionado ao cenário, a rede apresenta uma taxa de transferência menor. Isso ocorre devido a dois fatores principais: em primeiro lugar, os dados devem ser encaminhados pelo nó intermediário, o que causa um atraso na entrega da mensagem. Em segundo lugar, mais falhas de enlace ocorrem pois a fonte luminosa do intermediário pode interferir na fonte luminosa do nó de origem. Por fim, observamos que tão logo o obstáculo é removido, o desempenho da rede aumenta.



**Figura 6. Desempenho da rede utilizando D-VLC (RC = 5).**

A figura 6 apresenta a vazão, nas mesmas condições de rede discutidas acima. Nesse caso porém,  $RC = 5$ . Os resultados de vazão em momentos de comunicação direta são equivalentes aos resultados apresentados na figura 5. Quando um obstáculo é adicionado, notamos um tempo de reação maior, quando comparado com a figura anterior. Após o período de reação, em conformidade com a subseção 4.1, o cenário com  $RC = 5$  tem melhor desempenho quando comparado com o  $RC=2$ . Nesse caso, o cenário dinâmico com  $RC=5$  teve vazão média de 4,69 Kbps em uma comunicação com 2 saltos.

### 4.3. Análise de Sobrecarga

Nesta seção, derivamos de forma simples as principais fontes de sobrecarga do sistema ao se utilizar o protocolo proposto. Inicialmente, mensagens de rotas apresentam 24 bytes, referentes aos campos apresentados na equação 1.

$$mensagemDeRotaSemCaminho = limiteSalto + contadorSalto + IpOrigem + IpDestino + NumeroSequencia + tamanhoCaminho \quad (1)$$

De forma simples, em um processo onde a descoberta de rotas apresenta *sucesso*, a mensagem de RREQ passa por todos os nós em direção ao destino enquanto a mensagem de resposta RREP passa por um caminho reverso bem definido. A cada nó que a mensagem RREQ passa, ele inunda para todos nós vizinhos. Se consideramos uma topologia da rede bem conectada e com distribuição de grau homogênea, o processo de inundação pode atingir um número total de nós exponencialmente proporcional ao grau médio dos nós da rede. Assim, o tamanho total da mensagem pode crescer, como apresentado na equação 2, onde  $g$  é o grau médio dos nós da rede e  $T$  é a distância média entre origem e destino (profundidade em uma topologia similar a árvore).

$$S(n) = \left( \sum_{i=1}^T mensagemDeRotaSemCaminho + (noAdicional \times (i - 1)) \right) * (g^T + 1) \quad (2)$$

Entretanto, podemos considerar que, na prática, as redes VLC em ambientes internos não serão densas, e sim bem conectadas e homogêneas, devido a infraestrutura de iluminação que este tipo de ambiente oferece. De fato, terão uma topologia similar às apresentadas anteriormente (seção 3. Nesse caso, a inundação não terá um processo exponencial e assim, podemos simplificar a equação 3 como segue, onde a sobrecarga é expressa em função do tamanho do caminho de ida e de volta apenas.

$$S(n) = \left( \sum_{i=1}^n mensagemDeRotaSemCaminho + (noAdicional \times (i - 1)) \right) * 2 \quad (3)$$

Caso a descoberta de caminho não tenha sucesso, a sobrecarga do protocolo ocorre principalmente em função de mensagens RREQ e RREP que se propagam na rede sem atingir o destino final. No pior caso, o erro ocorrerá apenas no último nó do caminho e assim, a mensagem percorrerá  $(T-1)$  nós, onde  $T$  é a profundidade da rede. Mais ainda, o protocolo repete o processo um número  $TENTATIVAS\_MAX$  finito de vezes. Nos experimentos realizados, consideramos  $TENTATIVAS\_MAX=3$ . Assim, a equação 5 sumariza a sobrecarga prevista para caso ocorra falhas.

$$LN(n) = mensagemDeRotaSemCaminho + (noAdicional \times (T - 1)) \quad (4)$$

$$F(n) = (S(n) - LN(n)) \times TENTATIVAS\_MAX \quad (5)$$

Por fim, o processo de manutenção de rota tem custo relativo a uma única mensagem, o RERR, que é enviado em direção à origem do fluxo. Essa mensagem tem 16 bytes. Assim, o custo de manutenção de rota pode ser estimado como apresentado na equação 6:

$$E(n) = rotaRERR \times n \quad (6)$$

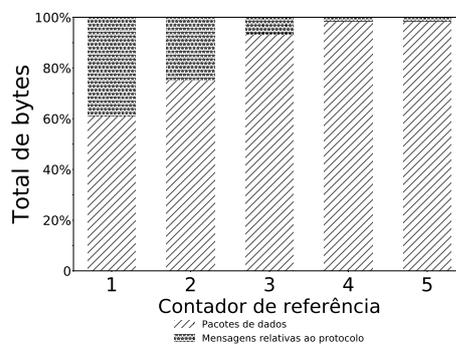
Em suma, a sobrecarga esperada para o protocolo pode ser estimada da seguinte forma:

$$Sobrecarga(n) = S(n) * nSucesso + F(n) \times nFalhas + E(n) \times nManutencao \quad (7)$$

As taxas de falhas ( $nFalhas$ ), sucesso ( $nSucesso$ ) e erros ( $nManutencao$ ) dependem do ambiente avaliado. Esses valores são extraídos da camada de enlace (*firmware* do OpenVLC), através de logs de Kernel do sistema operacional que controla o OpenVLC 1.0. Assim, a seguir, avaliamos a sobrecarga, a partir dos dados obtidos nos cenários experimentados.



(a) Foto de experimento dinâmico com 3 nós. Apesar do obstáculo, D-VLC acha rota que permite os nós se comunicarem.



(b) Sobrecarga do protocolo para um cenário de dois saltos.

**Figura 7. Configuração e resultados obtidos.**

A figura 7a apresenta um *snapshot* da configuração do experimento para o cenário dinâmico, como explicado anteriormente<sup>3</sup>. A figura 7b apresenta os resultados obtidos considerando as equações apresentadas anteriormente. Como é possível observar, quanto maior o valor do parâmetro de referência da camada de enlace (RC), menor a sobrecarga na rede. Por exemplo, para  $RC=1$ , a sobrecarga chega a ser superior a 40%. Nesse caso, o protocolo tem uma super-reação às perdas de dados na rede e, precocemente, interpreta como quebra de rotas.

<sup>3</sup>Video de demonstração do experimento para um cenário dinâmico -<https://photos.app.goo.gl/HnyzMpHK7tGyNS798>, 2018.

Por essa figura, quanto mais o protocolo posterga decisões com relação a quebra de rotas, menores os impactos com relação as mensagens de erro e manutenção de rota. Assim, o impacto ocasionado pela etapa de descoberta de rota é amortizado por uma quantidade maior de dados transmitidos. Assim, a partir de RC=4, observamos uma sobrecarga negligenciável, atingindo apenas 1% de todos os dados observado na rede.

## 5. Trabalhos relacionados

Le *et al.* [Le et al. 2011] propuseram um protocolo MAC cooperativo para sistemas VLC, com cooperação por (*relays*) situados entre o transmissor e o receptor. Esses *relays* atuam quando o enlace atual falha, ou não consegue oferecer banda suficiente para os requisitos de qualidade de serviço. Os autores realizaram avaliações teóricas e simulações e mostraram melhorias significativas de desempenho e confiabilidade para a rede modelada.

[Wu 2012] propõe uma solução *multi-hop* para múltiplos acessos em cenários VLC para ambientes internos, levando em consideração dois desafios principais em tais cenários: (i) linha de visão e (ii) direcionalidade. Com base nesses desafios, o autor oferece duas soluções de rede: *peer-to-peer* e *peer-to-host*. Esses protocolos são equipados com uma construção de roteamento de rede simples, na qual, quando um dispositivo deseja se comunicar com outro, ele tenta construir uma rota verificando seus vizinhos, procurando um nó intermediário para atuar como um relay. Esses protocolos não possuem mecanismos para garantir: (I) a ausência de *loops*, (II) a atualização de rotas; (III) a manutenção de rotas e informações de métricas.

	Tipo	Plataforma	Aplicação	Cooperação	Dinâmico
[Le et al. 2011]	Teoria e Simulação	ns-2 (Simulação)	Ambiente Interno	Sim	Sim
[Chowdhury and Katz 2013]	Simulação	-	Aquático	Não	Não
[Kim et al. 2016]	Prática	Personalizada	Ambiente Interno	Sim	Não
[Cherntanomwong and Namonta 2015]	Prática	Arduino Uno	Ambiente Interno	Sim	Não
[Klaver and Zuniga 2015]	Prática	Genérico / Arduino	Ambiente Interno	Sim	Não
[Schmid et al. 2016]	Prática	Arduino	Ambiente Interno	Sim	Não
Este trabalho	Prática	OpenVLC 1.0	Ambiente Interno	Sim	Sim

**Tabela 3. Estudos relacionados.**

Autores em [Chowdhury and Katz 2013] analisaram comunicação multi-hop usando *relays* para aumentar a conectividade de um sistema de VLC. Os autores desenvolveram um modelo de cobertura para VLC e realizaram simulações, onde avaliaram o impacto da seleção e mobilidade do *relay* na rede. [Schmid et al. 2016] também avaliaram o uso da comunicação baseada em múltiplos saltos para aumentar o alcance da comunicação. Em suas avaliações, com uso de dois nós intermediários, foi possível realizar comunicação por distâncias de até 4 metros entre o transmissor e receptor. sem considerar obstáculos. O aumento da distância de comunicação também é objetivo de pesquisa dos autores em [Cherntanomwong and Namonta 2015], que apresentaram o *Repeater*, um *relay* equipado com transeptores capazes de aumentar o alcance da rede.

A comunicação por luz visível também foi considerada em ambientes aquáticos, onde as ondas ópticas sofrem alta dispersão do sinal e são absorvidas em poucos metros [Vieira 2012]. Nesse sentido, [Kim et al. 2016] apresentam um sistema VLC de múltiplos saltos para aplicações marítimas e o avalia através de simulações, que proporcionaram boa qualidade de link a uma distância de até 5 km, com 4 nós de *relay*. [Ahmad et al. 2017] também desenvolveram um sistema VLC multi-hop subaquá-

tico. Eles avaliaram tal sistema em um cenário real e mediram a potência óptica recebida no cenário de link direto e a compararam ao cenário com múltiplos saltos.

Por fim, plataformas de pesquisa também consideram comunicação por múltiplos saltos. [Klaver and Zuniga 2015] apresentaram uma plataforma VLC baseada em Arduino. Eles exploram o problema de cobertura no sistema VLC usando 20 LEDs como transmissores e 4 fotodiodos como receptores, e assim, fornecem uma cobertura de comunicação com 360° ao seu redor. Os autores exploram a exposição e a direcionalidade dos LEDs para implementar um cenário de múltiplos saltos, no qual mecanismos como o encaminhamento de pacotes e a descoberta de vizinhos são implementados.

Todos os trabalhos acima apontam melhorias na comunicação fim-a-fim com o uso de múltiplos saltos. Porém, destacamos que eles se preocupam, principalmente, em aumentar o raio de alcance de comunicação fim-a-fim. Na sua maioria, necessitam do conhecimento da topologia da rede e do uso de retransmissores (*relays*) específicos. Nenhum deles trata por completo o problema de seleção dinâmica de nós cooperadores, quebra de comunicação e mobilidade. Em suma, não se adaptam as mudanças do ambiente de comunicação, como o surgimento de sombras e obstáculos.

## 6. Conclusão

Neste trabalho, apresentamos o D-VLC: um novo protocolo de roteamento dinâmico *cross-layer* para redes de comunicação com luz visível. D-VLC é baseado em protocolos de roteamento reativo, nos quais as rotas são construídas e mantidas de acordo com as demandas da rede. Nesse protocolo, qualquer nó intermediário pode se tornar parte de uma rota para o destino e encaminhar pacotes até que a rota seja interrompida ou expirada.

O D-VLC foi implementado em uma plataforma real de VLC. Os resultados mostram que, usando D-VLC, a rede toma conhecimento de obstáculos e reage a eles. Mesmo sem conhecimento prévio da topologia de rede, o D-VLC cria caminhos alternativos entre a origem e o destino e, nesse caso, a comunicação não é interrompida, como ocorreria nos sistemas tradicionais existentes. Por fim, os resultados mostram que a sobrecarga relativa ao protocolo é negligenciável (<1%). Mesmo em situações de obstáculos, as taxas de transmissão alcançadas são apenas limitadas pela tecnologia VLC utilizada.

Como principal trabalho futuro, destacamos a análise de cenários com diferentes dinamicidades e a inter-relação com os parâmetros do protocolo. Nessa linha, o protocolo pode ser tornar reativo, não somente aos obstáculos, mas também à taxa de erros encontrada em diversos ambientes.

## Agradecimentos

Capes, CNPq, FAPEMIG, Instituto de Pesquisas Eldorado.

## Referências

- Ahmad, Z., Rajbhandari, S., Salih, O., and Green, R. (2017). Demonstration of a multi-hop underwater visible light communication system. In *IEEE ICTON*.
- Cherntanomwong, P. and Namonta, P. (2015). The repeater system for visible light communication. In *IEEE ICITEE*.

- Chowdhury, H. and Katz, M. (2013). Cooperative multihop connectivity performance in visible light communications. In *Wireless Days (WD), 2013 IFIP*, pages 1–4. IEEE.
- De Vries, J. P., Simić, L., Achtzehn, A., Petrova, M., and Mähönen, P. (2014). The wi-fi “congestion crisis”: Regulatory criteria for assessing spectrum congestion claims. *Telecommunications Policy*, 38(8):838–850.
- Kaushal, H. and Kaddoum, G. (2016). Underwater optical wireless communication. *IEEE Access*, 4:1518–1547.
- Kim, H.-J., Tiwari, S. V., and Chung, Y.-H. (2016). Multi-hop relay-based maritime visible light communication. *Chinese Optics Letters*, 14(5):050607.
- Klaver, L. and Zuniga, M. (2015). Shine: A step towards distributed multi-hop visible light communication. In *IEEE MASS*.
- Le, N.-T., Choi, S., and Jang, Y. M. (2011). Cooperative mac protocol for led-id systems. In *ICT Convergence (ICTC), 2011 International Conference on*, pages 144–150. IEEE.
- Li, L., Hu, P., Peng, C., Shen, G., and Zhao, F. (2014). Epsilon: A visible light based positioning system. In *NSDI*, pages 331–343.
- Matheus, L., Pires, L., Vieira, A., Vieira, L. F. M., Vieira, M. A. M., and Nacif, J. A. (2018). The internet of light: Impact of colors in led-to-led visible light communication systems. *Internet Technology Letters*, 0(0):e78.
- Mortimer, M. (2018). iperf3 documentation. <https://media.readthedocs.org/pdf/iperf3-python/latest/iperf3-python.pdf>.
- Okada, S., Yendo, T., Yamazato, T., Fujii, T., Tanimoto, M., and Kimura, Y. (2009). On-vehicle receiver for distant visible light road-to-vehicle communication. In *IEEE Intelligent Vehicles Symposium*.
- Schmid, S., Richner, T., Mangold, S., and Gross, T. R. (2016). Enlighting: An indoor visible light communication system based on networked light bulbs. In *IEEE SECON*.
- Vieira, A. B., Vieira, L. F., Vieira, M., Freire, J., Matheus, L. M., and Gnawali, O. (2017). Comunicação por luz visível: conceito, aplicações e desafios. In *SBRC - Minicursos*.
- Vieira, L. F. M. (2012). Performance and trade-offs of opportunistic routing in underwater networks. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2911–2915.
- Wang, Q., Giustiniano, D., and Puccinelli, D. (2014). Openvlc: software-defined visible light embedded networks. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, pages 15–20. ACM.
- Wang, Q., Zuniga, M., and Giustiniano, D. (2016). Passive communication with ambient light. In *Proceedings of the 12th International Conference on emerging Networking Experiments and Technologies*, pages 97–104. ACM.
- Wu, Z. (2012). *Free space optical networking with visible light: a multi-hop multi-access solution*. Boston University.
- Xiang, Y., Zhang, M., Kavehrad, M., Chowdhury, M. S., Liu, M., Wu, J., and Tang, X. (2014). Human shadowing effect on indoor visible light communications channel characteristics. *Optical Engineering*, 53(8):086113.