Avaliação do Problema de Reordenamento de Pacotes e das Políticas de Escalonamento em Protocolos Multicaminhos

Paulo Lenz Jr., Matheus D. M. da Silva, Aldri Santos, Michele Nogueira

[1]Núcleo de Redes sem Fio e Redes Avançadas (NR2)
Universidade Federal do Paraná (UFPR)

{pljunior,mdms14,aldri,michele}@inf.ufpr.br

Abstract. *Quality of Service (QoS) and Quality of Experience (QoE) are fundamental to many Internet applications such as streaming services, online games and virtual reality. However, traditional transport-layer protocols are unsatisfactory in these issues due to limitations imposed on their conception. The MPQUIC (MultiPath QUIC) and MPTCP (Multipath TCP) protocols increase transmission performance due to the use of multiple network interfaces available on the devices, allowing to employ multiple and diverse end-to-end paths simultaneously or as redundancy. However, because of the network heterogeneity, multipath end-to-end communication results in challenges, such as out-of-order packet delivery, that reduced its benefits. Hence, we have evaluated the performance of MPQUIC and MPTCP in heterogeneous scenarios through several simulations. Results demonstrate that MPQUIC deals better with packet retransmissions and random losses, however, it is not immune to the out-of-order delivery problem. This work contributes to a better understanding of this problem in multipath transport-layer protocols, promoting future advances.*

Resumo. *A Qualidade de Serviço (QoS) e Qualidade da Experiência (QoE) são fundamentais para muitos aplicativos da Internet, como serviços de streaming, jogos online e realidade virtual. No entanto, os protocolos tradicionais da camada de transporte são insatisfatórios nessas questões devido às limitações impostas em sua concepção. Os protocolos MPQUIC (MultiPath QUIC) e MPTCP (Multipath TCP) aumentam o desempenho da transmissão devido ao uso de múltiplas interfaces de rede disponíveis nos dispositivos, permitindo empregar múltiplos caminhos e diversidade na transmissão fim-a-fim simultaneamente ou de forma redundante. No entanto, devido à heterogeneidade da rede, a comunicação multicaminhos fim-a-fim resulta em desafios, como a entrega de pacotes fora de ordem, que reduz seus benefícios. Por essa razão, avaliamos o desempenho do MPQUIC e MPTCP em cenários heterogêneos através de diversas simulações. Os resultados mostram que o MPQUIC lida melhor com retransmissões de pacotes e perdas aleatórias, mas não está imune ao problema de entrega fora de ordem. Este trabalho contribui para uma melhor compreensão do problema em protocolos multicaminhos, promovendo avanços futuros.*

1. Introdução

As aplicações que demandam acesso à Internet com alta vazão e baixa latência estão cada vez mais presentes na Internet. Os jogos *online*, a realidade virtual, as aplicações médicas e os serviços de *streaming*, como os oferecidos pela Netflix, Amazon Prime e HBO

GO, são alguns exemplos que exigem maior Qualidade de Serviço (QoS) e Qualidade de Experiência (QoE). Com a popularização de dispositivos com diferentes tecnologias de comunicação sem fio (*Multihomed*), que provêm o acesso simultâneo a múltiplas redes, a comunicação fim-a-fim por diferentes caminhos possui o objetivo de oferecer uma maior vazão e menor latência.

Devido às limitações dos protocolos de transporte mais usados na Internet hoje em dia, o TCP e o UDP, diversas abordagens que visam ao uso de múltiplas interfaces de rede simultaneamente foram propostas na literatura. Tais protocolos de transporte são chamados protocolos multicaminhos e aumentam o desempenho da transmissão fim-a-fim razão do uso dos diversos caminhos disponíveis de forma a agregar recursos e, para isto, dependem do algoritmo de controle de congestionamento (CC) e do escalonador de pacotes [Paasch et al. 2014]. O CC tem por objetivo ser amigável com os outros protocolos de modo a transportar dados de maneira justa sem prejudicar outros fluxos ou monopolizar o canal. O escalonador por sua vez deve analisar o estado dos caminhos disponíveis e alocar os dados de acordo com o tamanho máximo da janela de congestionamento.

Dentre os protocolos multicaminhos já padronizados pelo IETF (*Internet Engineering Task Force*), o protocolo Multicaminhos TCP (MPTCP) tem como base o TCP e possibilita a transmissão de dados através de múltiplos caminhos. O Multicaminhos QUIC (MPQUIC) é uma evolução do protocolo QUIC de caminhos únicos desenvolvido pela *Google*. Este protocolo utiliza o UDP para transmissão de dados e permite trabalhar em conjunto com o ambiente legado. Isto diminui a latência das conexões por se abster do *3-way-handshake*. Além disso ele usa a multiplexação de fluxos, tornando os fluxos independentes. Apesar de apresentarem melhor desempenho quando comparados com suas versões de caminho único, já analisadas em [De Coninck and Bonaventure 2017], ambos possuem muitos desafios relacionados à heterogeneidade dos caminhos [Alheid et al. 2016], ao princípio de justiça (*fairness*) [Becke et al. 2012], ao consumo de energia [Kaup et al. 2015], ao custo monetário de transmissão [Secci et al. 2014], à segurança [Pearce and Zeadally 2015], entre outros. Os caminhos heterogêneos degradam o desempenho da transmissão multicaminhos devido à perda e à chegada de pacotes fora de ordem (e.g., problema do reordenamento). O problema do reordenamento aumenta o atraso e reduz a vazão da transmissão multicaminhos, e afeta, principalmente, as aplicações sensíveis ao atraso [Yedugundla et al. 2016].

Desta forma, este trabalho apresenta uma investigação do problema de reordenamento com as atuais políticas de escalonamento dos pricipais protocolos multicaminhos em cenários heterogêneos. A fim de verificar o comportamento do MPTCP e MPQUIC, simulamos diversos cenários com diferentes características e cargas de trabalho. A avaliação demonstra que o número de retransmissões é bastante similar para ambos os protocolos, uma vez que retransmissões implicam em perdas ou atraso na entrega dos pacotes, e faz os segmentos aguardarem reordenação para serem entregues à aplicação. Avaliamos também o número de perdas e o tempo total da conexão. As simulações demonstram que o protocolo MPQUIC lida melhor com as perdas e retransmissões, além de concluir a transferência de dados de forma mais rápida do que o MPTCP, possivelmente por se abster do estabelecimento de conexões, uma vez que está fundamentado no protocolo UDP.

O restante do trabalho está organizado como segue. A Seção 2 apresenta brevemente os conceitos pertinentes ao protocolo QUIC, MPQUIC e MPTCP. A Seção 3 detalha a avaliação realizada. A Seção 4 apresenta as principais estratégias empregadas junto aos métodos de escalonamento para lidar com o problema de reordenamento de pacotes. Por fim, a Seção 5 apresenta as conclusões e direções futuras.

2. Os Protocolos MPTCP, QUIC e MPQUIC

Esta seção apresenta os conceitos e as definições dos protocolos MPTCP, QUIC e MPQUIC. Estes conceitos são necessários para a compreensão destes protocolos. Inicialmente apresentamos o protocolo MPTCP, pela proximidade com o protocolo TCP. Assim como o protocolo MPTCP avança o protocolo TCP mantendo grande parte de suas características, salientamos que grande parte dos atributos do protocolo MPQUIC são herdadas do protocolo QUIC. Desta forma, como o protocolo QUIC não é tão conhecido quanto o TCP, uma descrição do protocolo QUIC antecede a explicação do protocolo MPQUIC.

Os protocolos da camada de transporte gerenciam a comunicação fim-a-fim entre processos de aplicações junto aos *hosts* finais. Dentre eles, os protocolos TCP e UDP são os mais utilizados na comunicação entre cliente e servidor na Internet. O protocolo TCP fornece entrega confiável, ordenada e com verificação de erros. Apesar da sua confiabilidade, ele pode ser inadequado para algumas aplicações que não necessitam do controle exercido por ele. As aplicações que não exigem um serviço de fluxo de dados confiável podem usar o protocolo UDP, adequado para operações em que a verificação e a correção de erros não são necessárias ou são executadas na aplicação. O protocolo MPTCP é baseado no protocolo TCP e o protocolo MPQUIC é baseado no protocolo UDP, porém o MPQUIC aplica mecanismos para garantir a entrega confiável.

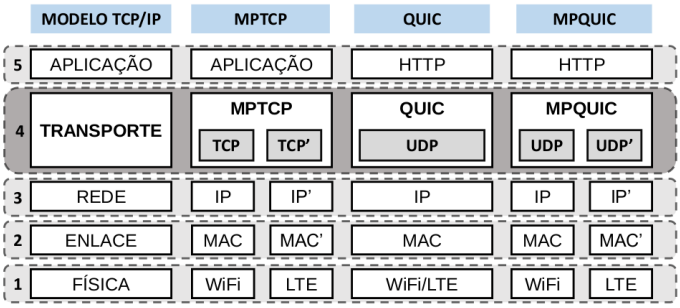


Figura 1. Disposição do protocolo MPTCP, QUIC e MPQUIC no Modelo TCP/IP

Até o momento, os protocolos QUIC e MPQUIC suportam apenas o tráfego web em conjunto com o protocolo da camada de aplicação HTTP, como pode ser observado na Figura 1. Contudo, um de seus objetivos é se tornarem protocolos de transporte generalizados, para múltiplas aplicações. Outro fator importante é o fato desses protocolos (MPTCP, QUIC e MPQUIC) utilizarem o TCP e o UDP sem alterar seus modelos de referências, mantendo-os em consonância com o ambiente legado. De forma simplificada, o MPTCP e o MPQUIC gerenciam o uso das interfaces, podendo iniciar várias conexões com um *host* específico. Para isso, eles administram a utilização de múltiplos fluxos independentes, de forma que um sub-fluxo (*TCP* ou *UDP*) pertença a mesma conexão que outro sub-fluxo (*TCP'* ou *UDP'*). Entretanto, cada um possui controle ao nível de conexão e de fluxo. Consequentemente, as camada inferiores possuem diferentes atributos

para cada canal de comunicação. No nível da camada de rede, cada interface possui um endereço IP diferente (*IP, IP'*). O mesmo ocorre para os endereços de hardware (*MAC, MAC'*), que identificam cada interface utilizada para o tráfego de dados no nível de enlace. Por fim, a comunicação via WiFi e LTE são exemplos das tecnologias utilizadas para propagar os dados, conforme ilustra a Figura 1.

2.1. O protocolo multicaminhos TCP

O protocolo MPTCP possibilita a transmissão de dados através de múltiplos caminhos [Bagnulo 2011], [Silva et al. 2018]. Um dos principais objetivos do MPTCP envolve obter um desempenho superior ao oferecido por um único fluxo TCP, em termos de melhorar a vazão e reduzir a latência. O protocolo MPTCP também possibilita aumentar a resiliência da comunicação, utilizando os caminhos de forma redundante para persistir a conexão em caso de falhas. Uma comunicação MPTCP provê a troca de dados bidirecional entre dois nós comunicando assim como o TCP padrão, não requerendo qualquer mudança na aplicação. Ele também possibilita que os *hosts* finais utilizem diferentes caminhos e endereços para transmitir pacotes pertencentes a uma mesma conexão.

Uma conexão MPTCP inicia de modo similar a uma conexão TCP, utilizando o *3-way handshake*. A Figura 2 ilustra o estabelecimento de uma conexão MPTCP entre cliente e servidor partir dos caminhos C1 e S1, respectivamente. Após a configuração inicial, os *hosts* trocam informações sobre os endereços adicionais e um novo *3-way handshake* é realizado para adicionar um sub-fluxo com os endereços C2-S2. O MPTCP ainda permite que outros sub-fluxos sejam criados, por exemplo, combinando os endereços C1-S2 e C2-B1. Ele considera cada fluxo do TCP como um de seus sub-fluxos, mas que transportam no cabeçalho do TCP um tipo de campo *Option* específico. Todas as operações do MPTCP são sinalizadas através deste campo. Para controlar o envio de pacotes através de diferentes caminhos o MPTCP possui dois níveis de reconhecimento, por sub-fluxo (*SSN - Subflow Sequence Number*) e por conexão (*DSN - Data Sequence Number*), como pode ser observado na Figura 3. Os reconhecimentos SSN, utilizados no TCP, são usados para confirmar o recebimento dos segmentos em cada sub-fluxo independente do DSN.

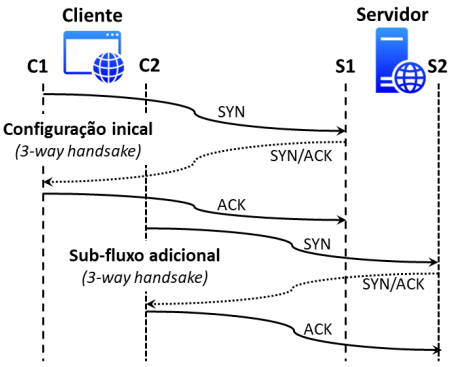


Figura 2. Conexão MPTCP

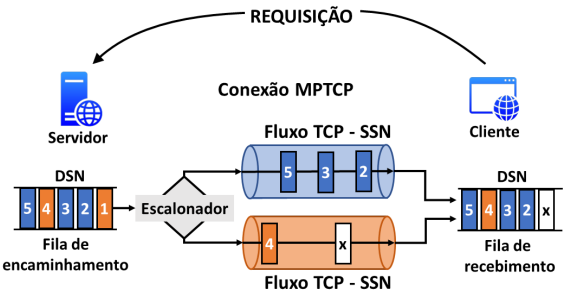


Figura 3. Bloqueio HOL no MPTCP

Apesar dos benefícios que o MPTCP oferece, ele também herda os problemas que o TCP oferece para a conexão fim-a-fim como o bloqueio da fila de recebimento (HOL - *Head-of-line blocking*). O bloqueio HOL ocorre devido ao esgotamento da fila de recebimento em face da entrega de pacotes fora de ordem. Isto causa intermitência no

recebimento dos dados e atrasos na entrega ordenada dos dados à camada de aplicação, como demonstra a Figura 3. Para compensar a heterogeneidade dos caminhos e evitar os problemas com o bloqueio HOL, o protocolo MPTCP usa o mecanismo de retransmissão e penalização [Paasch et al. 2014] , [Possati et al. 2018]. Este mecanismo reencaminha o segmento que possa estar causando o problema do bloqueio HOL no fluxo que tenha espaço na janela de congestionamento. O objetivo é tornar mais rápida a recuperação das situações de bloqueio, compensando a diferença entre os RTT dos caminhos. Contudo, a redução da janela de um fluxo com RTT alto acaba reduzindo e limitando a capacidade de envio e, consequentemente, diminui o desempenho geral da transmissão multicaminhos.

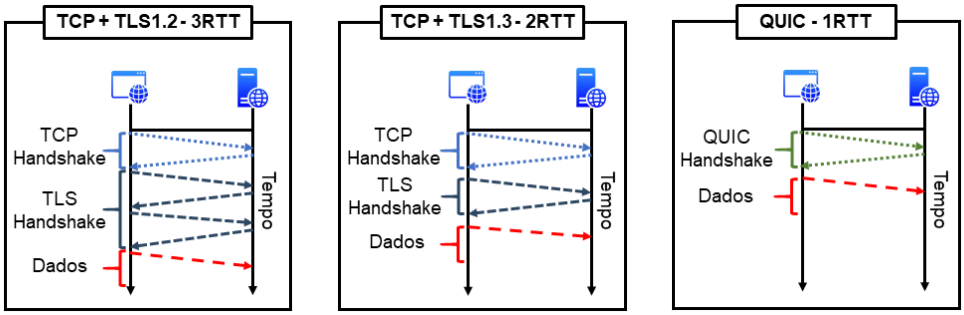
2.2. QUIC

O protocolo QUIC (*Quick UDP Internet Connection*) apresenta um novo conceito de protocolo de transporte. Baseado em técnicas e experiências obtidas com o TCP, SPDY, SCTP, TLS e HTTP/2, o protocolo QUIC utiliza o protocolo UDP em sua essência. Como ele impõe criptografia em todos os seus pacotes, pode ser comparado ao conjunto *TCP + TLS + HTTP/2*. A fim de evitar problemas com o ambiente legado, o QUIC encapsula seus dados em pacotes UDP e aplica mecanismos próprios de controle de conexão e fluxo. Isto facilita a sua implementação, pois não exige alterações no núcleo dos sistemas operacionais ou componentes intermediários (*middleboxes*) à conexão fim-a-fim, e.g., comutadores, *proxies* e *firewalls*. Atualizar ou modificar protocolos consolidados da pilha TCP/IP é uma tarefa difícil e pode levar anos para serem aceitas e distribuídas para os usuários finais. A título de exemplo, o protocolo IP em sua versão 6 levou cerca de 20 anos no processo de desenvolvimento até sua implementação [Dhamdhere et al. 2012].

O pacote QUIC e seu reconhecimento (ACK) contém informações que ajudam o controle do congestionamento e a recuperação de perdas. Cada pacote QUIC é composto de um cabeçalho público (não criptografado) com informações dos *hosts* e da conexão, e dos dados (*payload*) que são sempre criptografados. Esses pacotes transportam um novo número de sequência, incluindo aqueles que transportam dados retransmitidos. Isso elimina a necessidade de um mecanismo separado para distinguir confirmações de retransmissões das transmissões originais, evitando o problema de ambiguidade de retransmissão do TCP. Os reconhecimentos QUIC também codificam explicitamente o atraso entre o recebimento de um pacote e seu reconhecimento sendo enviado e, juntamente com os números de pacote com aumento monotônico, isso permite o cálculo preciso do tempo de ida e volta da rede (RTT). Os quadros ACK do QUIC suportam vários blocos ACK [J. Iyengar and M. Thomson 2018], portanto, o QUIC é mais resiliente à reordenação do que o TCP com suporte ao SACK (sistema de reconhecimento seletivo de pacotes, mensagens ou segmentos), além de poder manter mais bytes em trânsito quando há reordenamento ou perda.

Podemos mensurar os benefícios do QUIC em utilizar o UDP, em vez do TCP, comparando o processo de estabelecimento de uma conexão. Por exemplo, o TCP inicia com a negociação de parâmetros congruentes entre cliente e servidor e esse processo, conhecido como *3-way-handshake*, está presente em todas as conexões desse protocolo. O *handshake* do TCP geralmente leva 1 RTT (tempo de ida e volta) para ser concluído. A fim de aumentar a segurança das transmissões, o protocolo de segurança da camada de transporte (TLS) é empregado para criptografar as mensagens. O TLS também possui um processo de *handshake* para troca de chaves e certificados. Na sua versão mais utilizada

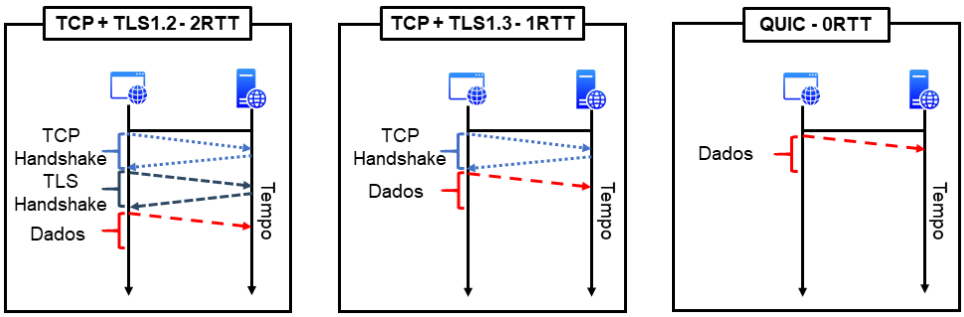
(versão 1.2), ele leva 2 RTTs para efetuar o *handshake* nos casos em que o cliente e servidor não tenham se comunicado anteriormente, totalizando 3 RTTs até o início da transmissão efetiva dos dados, como demonstra a Figura 4(a). A versão 1.3, versão mais atualizada do TLS, diminui seu processo de *handshake* para 1 RTT, além do processo do TCP, totalizando 2 RTTs como ilustra a Figura 4(b). Para o protocolo QUIC, que também criptografa os dados da transmissão e efetua a troca de certificados, o processo de *handshake* leva apenas 1 RTT nos casos em que a comunicação não tenha acontecido previamente, conforme ilustra a Figura 4(c).



(a) TLS v1.2 (b) TLS v1.3 (c) QUIC

Figura 4. Estabelecimento conexão sem comunicação prévia cliente e servidor

Após o estabelecimento da primeira conexão entre cliente/servidor, as chaves e os certificados trocados são armazenados para serem reutilizados. Dessa forma, os protocolos que aumentam a segurança da transmissão diminuem o tempo necessário para estabelecer uma conexão que já foi realizada previamente, desde que essas chaves e certificados não tenham expirado. Assim, como ilustra a Figura 5(a), o TLS na versão 1.2 diminui 1 RTT para estabelecer a conexão a partir da segunda vez que os *hosts* se comunicam. A versão 1.3 necessita apenas do *handshake* do TCP até iniciar a transmissão das requisições do cliente, como mostra a Figura 5(b). O protocolo MPQUIC diferencia-se dos demais por iniciar a transmissão sem a necessidade de um *handshake*, enviando as requisições do cliente no primeiro contato com o servidor, como demonstra a Figura 5(c). O Google estima que 75% das conexões utilizem o 0-RTT *handshake*.



(a) TLS v1.2 (b) TLS v1.3 (c) QUIC

Figura 5. Estabelecimento conexão com comunicação prévia cliente e servidor

2.3. Multicaminhos QUIC

O procolo MPQUIC compartilha todos os benefícios que o QUIC insere na transmissão fim-a-fim com a vantagem de utilizar as múltiplas interfaces de rede disponíveis. Dentre elas, a multiplexação completa de solicitações e respostas, que possibilita a divisão de mensagens em quadros independentes para a transmissão, como pode ser observado na Figura 6. Esta característica reduz os efeitos causados pelo problema de bloqueio HOL bem conhecido para as conexões baseadas no TCP. Por exemplo, se o pacote 1 do caminho 1 for perdido durante a sua transmissão, ele não afetará os outros fluxos compostos pelos pacotes $4, 5$ e $6, 7$. Com a possibilidade de utilizar vários caminhos e a independência dos segmentos que a multiplexação insere, mais informações podem estar em trânsito com menor probabilidade de erros e atrasos causados por perdas de pacotes.

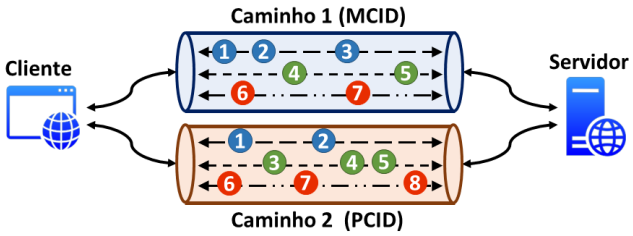


Figura 6. Exemplo de conexão MPQUIC

A administração de múltiplos caminhos exige a adição de novos campos ao cabeçalho público dos pacotes MPQUIC. Durante o estabelecimento de uma conexão, o protocolo MPQUIC executa uma verificação dos caminhos disponíveis entre o cliente/servidor. O caminho pelo qual a conexão é iniciada recebe a identificação MCID (*Master Connection ID*), que é também utilizada para identificar uma conexão. Uma vez que haja apenas um caminho disponível, a conexão seguirá como os padrões do QUIC. Caso haja mais de um endereço utilizável para a transmissão, os *hosts* trocam informações e negociam a adição destes caminhos. Cada caminho adicional recebe então um identificador PCID (*Path Connection ID*) que distingue cada caminho utilizado. O MCID e o PCID compõem a tupla que o protocolo utiliza para gerenciar a conexão em conjunto com o número de sequência do pacote (*PN - Packet Number*), individual para cada caminho.

2.4. Algoritmos de controle de congestionamento e o escalonador

O algoritmo de controle de congestionamento (CC) e o escalonador de segmentos são componentes herdados do MPTCP e adaptados para o MPQUIC. Um dos objetivos do algoritmo de controle de congestionamento é garantir o compartilhamento justo da largura de banda (princípio de *fairness*) e o uso eficiente dos caminhos [De Coninck and Bonaventure 2017]. De forma geral, a extensão desenvolvida para o MPQUIC é mais simples e mais limpa do que a extensão para o MPTCP, e consequentemente, mais fácil de ser implementada. Graças ao suporte para múltiplos fluxos, o MPQUIC não precisa especificar um novo tipo de número de sequência em contraste com o MPTCP. O MPQUIC também não precisa especificar mecanismos para detectar ou reagir à interferência de *middleboxes*, uma vez que todos os dados são criptografados e autenticados. Isso também reduz a possibilidade de ataque a uma conexão MPQUIC comparado com uma conexão MPTCP, cuja segurança depende de chaves trocadas em texto puro durante o aperto de mão inicial (*handshake*).

O algoritmo de controle de congestionamento OLIA (*Opportunistic Linked Increases Algorithm*) [Khalili et al. 2013] é o padrão utilizado tanto para o MPTCP e quanto para o MPQUIC neste trabalho e possui o objetivo de obter melhorias no princípio da eficiência (*Pareto Optimal*). Este princípio estabelece que é impossível aumentar a vazão de uma conexão sem reduzir a vazão de outra ou aumentar o custo de congestionamento. Baseado nesta premissa, o OLIA fornece simultaneamente capacidade de resposta e balanceamento de congestionamento ao adaptar sua janela, que aumenta em função do número de *bytes* transmitidos desde a última perda.

O escalonador é um dos componentes principais de qualquer arquitetura de agregação de banda [Ramaboli et al. 2012]. Para o MPQUIC, o escalonador é um dos responsáveis por alcançar um bom desempenho, dadas as várias restrições dos serviços, requisitos dos usuários e a variação das condições dos caminhos [Paasch et al. 2014]. O escalonador distribui os segmentos através de um ou mais caminhos, com o objetivo que sejam transmitidos, recebidos e entregues de forma confiável e ordenada à aplicação de destino. Os fluxos de dados da aplicação são armazenados na fila de envio e conforme a lógica da política de escalonamento implementada, os pacotes são retirados desta fila e inseridos na fila dos caminhos disponíveis. Atualmente, há duas opções de políticas de escalonamento inseridas no código do MPQUIC: *Round-Robin* (RR) e *Lowest-RTT-First* (Low-RTT). O escalonador RR distribui os dados de modo circular entre os caminhos disponíveis sem critérios de seleção. A política RR em geral não é utilizada por não caracterizar os caminhos e pelo baixo desempenho comparado a outras políticas [Li et al. 2016]. O escalonador Low-RTT seleciona e aloca os dados nos caminhos com menor RTT.

3. Metodologia de Avaliação e Resultados

A construção dos cenários de avaliação está fundamentada nos fatores e parâmetros dos trabalhos de [Paasch et al. 2013] e [De Coninck and Bonaventure 2017]. A Tabela 3 apresenta os valores usados para cada fator. Cada cenário utiliza uma variação dos valores de atraso, fila, largura de banda e taxa de perda para cada caminho de forma independente. Foram construídos 50 cenários distintos com o intuito de simular as diferentes características de redes heterogêneas. Os cenários foram para ambos os protocolos e 3 repetições para cada conjunto. Para a realização dos experimentos, utilizamos um Notebook Dell Inspiron 15 com processador de 1,6 GHz e quatro núcleos, e 8 GB de memória RAM. O Notebook está configurado com o *Ubuntu-16.04 LTS* e com o emulador de redes *Mininet* na versão 2.2.1. O MPTCP foi instalado no *kernel* do *Ubuntu*. A Figura 7 ilustra a representação do cenário dos experimentos composto por dois *hosts* finais (cliente e servidor) e dois comutadores de rede. Com o emulador *Mininet* e a implementação do MPQUIC desenvolvida na linguagem *Go Lang* disponível no *Github* [Clemente 2016].

Fator	Menor Capacidade		Maior Capacidade	
	Min	Max	Min	Max
Capacidade [Mbps]	0,1	100	0,1	100
Atraso (RTT) [ms]	50	100	50	400
Atraso de fila [ms]	0	100	0	2000
Perdas [%]	0	2,5	0	2,5

Tabela 1. Fatores dos caminhos

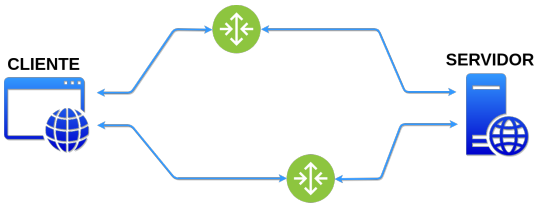


Figura 7. Cenário de teste

Os experimentos simulam a transferência de um arquivo entre um cliente e um servidor. Para avaliarmos o desempenho do escalonador de pacotes *Low-RTT* em cenários e situações mais próximas da realidade, variamos a carga de trabalho em três valores de 300 KB, 1 MB e 20 MB. A primeira carga utilizada simula um tráfego curto, com um arquivo de *300 KB*, referenciado na literatura como *mice flow*. Para a segunda, o arquivo contém *1 MB* e representa conexões mais duradouras comparadas com a primeira. Por fim, a terceira carga de trabalho possui *20 MB* e representa conexões longas, também chamadas de *elephant flow*, com um fluxo contínuo e extenso (em número de *bytes*).

Como métricas para a avaliação, utilizamos a taxa de pacotes retransmitidos ($R = r/s, R \in Emissor$), onde r representa o número total de pacotes retransmitidos e s número total de pacotes enviados observados no emissor. A taxa de pacotes perdidos também é calculada com a simples equação ($L = s - a, s \in Emissor, s \in Recpetor$), onde a representa a quantidade de pacotes recebidos. Por fim, o tempo total da comunicação, isto é, o tempo que compreende o início do processo de *handshake* até o recebimento do ultimo *byte* da transmissão de dados. Note que as retransmissões ocorrem sempre que um pacote é perdido ou após o recebimento de *ACKs* duplicados, isto implica que o segmento aguardará em *buffer* até que o pacote seja retransmitido ou o segmento será descartado após o fim de um temporizador. A Figura 8 ilustra o processo quando há perdas durante a transmissão. Neste exemplo, o pacote 3 é perdido durante a transmissão dos dados. Todos os dados subsequentes que pertença a este segmento são armazenados em *buffer* e aguarda a retransmissão do pacote perdido para que a entrega à aplicação ocorra em ordem.

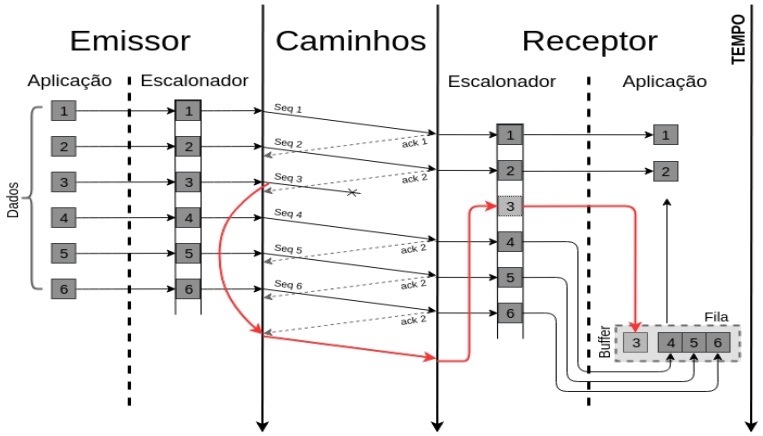
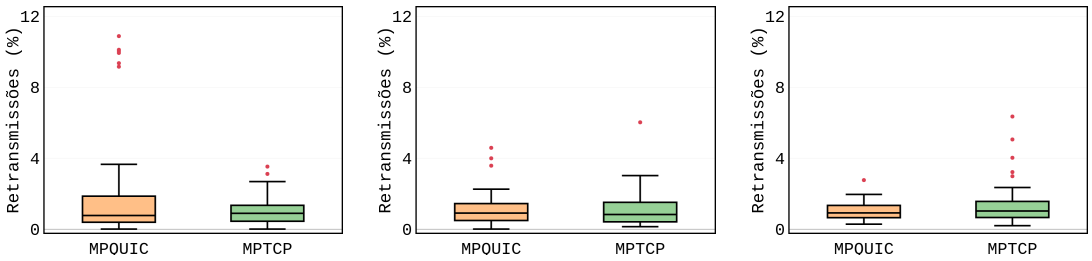


Figura 8. Problema de Reordenamento

A Figura 9(a) ilustra a taxa de retransmissões decorrentes da transferência do arquivo de *300KB* . A distribuição dos dados para o MPTCP apresenta menor variação comparado com o MPQUIC. Também é possível notar uma maior quantidade de pontos discrepantes do MPQUIC comparado com MPTCP, com a máxima em torno de 11%. Contudo, a mediana das amostras avaliadas é menor para o MPQUIC e representa 0,76%, enquanto que para o MPTCP este valor fica em torno de 0,88% das amostras de retransmissão. Nesta ocasião, como a quantidade de dados é relativamente pequena, por vezes o cliente não utiliza o recurso de multicaminhos e realiza toda a transferência por um caminho somente. Esta adversidade foi observada para ambos os protocolos. Alguns trabalhos na literatura estudam este fenômeno e defendem o uso de apenas uma interface de rede

para estes casos, com a justificativa de diminuir o uso de energia, recursos de rede e/ou monetário para os usuários.

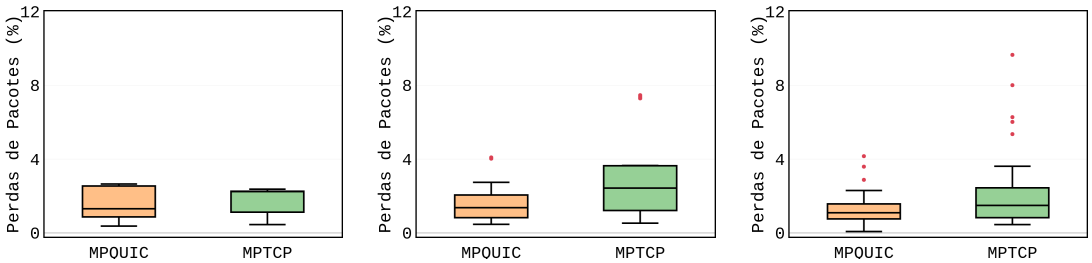


(a) 300 KB (b) 1 MB (c) 20 MB

Figura 9. Retransmissão de pacotes em diferentes cargas de trabalho

A transmissão do arquivo de *1 MB*, como mostra a Figura 9(b), apresenta uma menor distribuição dos dados para o MPQUIC comparado com a carga de trabalho anterior. Contudo, o MPTCP apresenta o mesmo comportamento da carga de *300KB*, com a exceção de um ponto discrepante, o qual representa 6% de pacotes retransmitidos. Para a carga de trabalho de *20MB* , como pode ser visto na Figura 9(c), o MPTCP sofre mais com as retransmissões em relação ao MPQUIC, bem como uma maior incidência de pontos discrepantes. Apesar da baixa taxa de retransmissão observada na avaliação, podemos verificar que o protocolo MPQUIC é moderadamente melhor em lidar com este fenômeno nos cenários em que a carga de trabalho é mais extensa em relação ao MPTCP.

Apesar de estabelecer uma variação de 0 a 2,5% para a taxa de perda aleatória das simulações, este fator pode variar de acordo com outros elementos da rede, como o atraso, tamanho de fila, entre outros. Assim, a Figura 10(a) exibe a variação de perdas sofridas durante a transmissão do arquivo de *300KB* . Ainda que o MPQUIC demonstre uma maior distribuição dos dados com a miníma em 0,37% e a máxima em 2,65%, a mediana se encontra em 1,31% e a média representa 1,51% dos dados observados. Em relação ao MPTCP, a mínima e a máxima representam 0,45% e 2,37%, respectivamente. Contudo, a mediana apresenta 2,26% e a média expõe 1,72% da taxa de pacotes perdidos.



(a) 300 KB (b) 1 MB (c) 20 MB

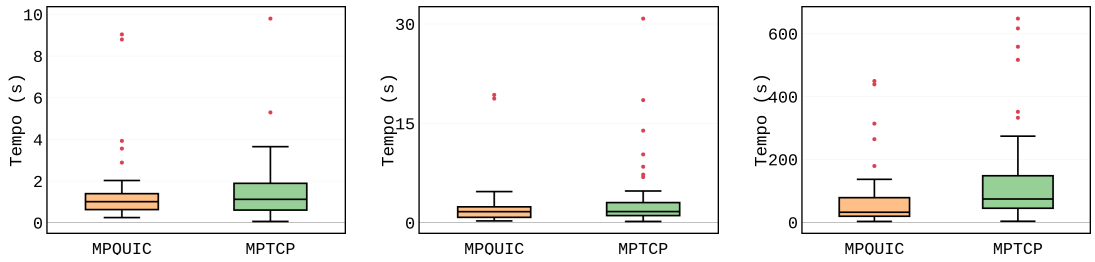
Figura 10. Perdas de pacotes em diferentes cargas de trabalho

Com as cargas de trabalho de *1MB* e *20MB*, o protocolo MPQUIC se sobressai em relação ao MPTCP. De acordo com a Figura 10(b) e 10(c), podemos observar que os resultados obtidos apresentam maior uniformidade e menor ocorrência de pontos discrepantes quando comparado com o MPTCP. No experimento com a carga de *1MB*, o MPQUIC per-

deu 80% menos pacotes em média comparado com o MPTCP. Já com a carga de *20MB*, o MPTCP apresenta 65,6% a mais na média de pacotes perdidos em relação ao MPQUIC.

O MPQUIC leva menos tempo em média na entrega dos dados. Isto se deve a redução da latência durante o estabelecimento da conexão e ao sistema de reconhecimento por pacotes, que oferece informações mais ricas e atualizadas sobre os estados dos *links*. O que colabora com o CC e com o escalonador de pacotes na tomada de decisão. Na Figura 11(a) é possível observar que o MPQUIC mantém a distribuição dos dados observados mais consistentes, entretanto ele apresenta maior quantidade de pontos discrepantes em relação ao MPTCP para a transmissão da carga de *300KB* . Contudo, o MPTCP apresenta o maior valor observado de 9,78 segundos.

Com a carga de trabalho de *1 MB*, o comportamento de ambos os protocolos é bastante similar. Em média, o MPQUIC levou 2,42 segundos e a mediana representa 1,66 segundos. Novamente o MPTCP apresenta a maior variação observada e alcança 30,8 segundos para a transferência do arquivo. Este resultado reflete também a maior capacidade de lidar com as perdas de pacotes analisadas anteriormente, uma vez que com a perda de pacote, a janela de congestionamento é reduzida, diminuindo a quantidade de dados que podem ser transmitidos durante um período de tempo administrado pelo CC.



(a) 300 KB (b) 1 MB (c) 20 MB

Figura 11. Tempo total da conexão

Por fim, o tempo que os protocolos levam para transferir *20MB* evidencia um melhor desempenho na transmissão de dados do MPQUIC, isso pode ser observado na Figura 11(c). Nota-se que a distribuição dos dados referente ao protocolo é menor, comparado com o MPTCP. Isto denota que dentre as simulações realizadas, o MPQUIC apresenta resultados mais consistentes. Em média, o MPQUIC leva 69 segundos para a transferência e sua mediana denota 32 segundos. O MPTCP, em média, precisa de 74,4 segundos para concluir a transferência. Outro ponto consiste nas diferenças entre os valores máximos obtidos, indicando que o MPTCP leva em torno de 200 segundos a mais para concluir toda transferência da carga utilizada, 44,32% mais que o pior caso do MPQUIC.

4. Trabalhos Relacionados

A primeira versão do MPQUIC surgiu em 2017 por [De Coninck and Bonaventure 2017]. Assim, há poucos trabalhos na literatura que abordam o protocolo, e muito menos trabalhos que tratam das políticas de escalonamento para o MPQUIC. Desta forma, nossa referência inicial são os trabalhos sobre escalonamento para o MPTCP, pois em sua atual implementação o MPQUIC herda os mecanismos de controle de congestionamento e escalonamento do MPTCP, tornando-se possível compreender os principais desafios abordados na tarefa de escalonar dados em multicaminhos.

A implementação do escalonador utilizado para o MPTCP e MPQUIC emprega o atraso dos caminhos (RTT) para classificá-los, selecioná-los e alocar os pacotes por eles [Li et al. 2016]. Entretanto, alocar os pacotes pelos caminhos baseados no RTT de cada caminho pode gerar o problema de reordenamento dos pacotes [Garcia-Saavedra et al. 2017]. O atraso de entrega em cada caminho é geralmente incerto e varia ao longo do tempo [Paasch et al. 2014, Albaladejo et al. 2016] devido às filas, retransmissões e perdas de pacotes, políticas do algoritmo de congestionamento e outros [Garcia-Saavedra et al. 2017]. Além das políticas padrões citadas previamente, outras propostas de escalonadores foram desenvolvidas pela comunidade. No trabalho de [Kuhn et al. 2014], os autores apresentam o *Delay-Aware Packet Scheduling* (DAPS) a fim de superar o bloqueio de HOL devido à heterogeneidade dos caminhos. Eles derivaram uma regra geral para o tamanho do *buffer* para o MPTCP. Para isso, o escalonador decide o caminho a enviar cada pacote com base no atraso de envio e na janela de congestionamento individualmente, a fim de prever quando um fluxo enviado por um caminho chegará no destino para enviar outro fluxo esperando que cheguem em sequência.

O trabalho de [Lim et al. 2017] apresenta o escalonador ECF (*Earliest Completion First*), que usa o comprimento do buffer de envio para estimar o tempo total de fluxo (*Flow Complete Time* - FCT) para cada caminho. Se o uso do caminho mais lento aumentar muito o FCT, ele aguardará pelo caminho mais rápido. O ECF toma decisões de escalonamento baseando-se no tamanho da janela de congestionamento e no RTT para evitar períodos de transmissão ociosos e assim alcançar uma maior taxa de transferência agregada. Da mesma forma que o Low-RTT, o ECF prioriza o caminho mais rápido em termos de RTT. Além disso, uma vez que o caminho mais rápido é bloqueado após seu preenchimento pelo controle de congestionamento, ele avalia se é realmente benéfico enviar em um caminho mais lento. Em seguida, ele pode decidir não enviar pelo caminho mais lento, a fim de aguardar a recuperação do caminho mais rápido. O ECF baseia essa decisão na quantidade de dados que aguardam o envio, juntamente com o RTT do caminho e as estimativas de capacidade. Ele não leva em conta o atraso de uma via, portanto, não é capaz de obter uma chegada precisa e ordenada.

[Le and Bui 2018] lida com o problema de reordenamento de pacotes do MPTCP usando um algoritmo de escalonamento baseado no atraso de encaminhamento (FDPS). A ideia principal é que o remetente distribui pacotes através de múltiplos caminhos de acordo com o atraso estimado e a diferença de vazão. Devido à assimetria de latência na Internet, é difícil obter um bom atraso unidirecional. Em geral, as estimativas de atraso consideram as medições do RTT, como feito pelo TCP, assumindo cada atraso, ida e volta, como $RTT/2$. O que nem sempre é aplicável, devido às flutuações do atraso no tempo de ida ser diferente do tempo de volta. Para isso, ele propõe uma metodologia de sincronização de relógio, a fim de se obter o atraso apenas de ida do caminho. Baseado no *timestamp* de pacotes enviados por dois caminhos, o autor calcula as diferenças entre as estatísticas estimadas para se obter o atraso apenas do envio de cada caminho.

5. Conclusão

Este trabalho avaliou o desempenho de transmissões de dados dos protocolos multicaminhos MPTCP e MPQUIC sobre diversos cenários heterogêneos. Diante das diferentes características dos caminhos disponíveis, o MPQUIC apresentou uma pequena vantagem no tempo de transmissão de cargas de trabalho menores. Entretanto, com cargas volu-

mosas, seus benefícios para a transmissão tornam-se mais evidentes, utilizando cerca de 44,32% a menos de tempo que o MPTCP para transmitir *20MB* de dados no pior caso observado. Isto se deve principalmente à redução da latência ao utilizar o protocolo UDP e por se abster do processo de *3-way-handshake* na maioria de suas comunicações. O MPTCP, pelo contrário, exige que todas as interações sejam iniciadas com a negociação de parâmetros da conexão antes de encaminhar as requisições efetivamente ao destinatário. De forma geral, o MPQUIC também lida melhor com as retransmissões e perdas de pacotes durante a comunicação. Graças ao seu sistema de reconhecimento por pacotes e não por blocos como é feito pelo padrão do MPTCP, herdado do mecanismo SACK (*Selective ACKs*) do protocolo TCP. O escalonador de pacotes, em conjunto com o algoritmo de controle de congestionamento, exercem um importante papel na comunicação multicaminhos fim-a-fim. Ainda que o MPQUIC apresente melhores resultados, estes mecanismos são heranças adaptadas do MPTCP, o qual possui diferenças fundamentais quando comparado ao MPQUIC. O desenvolvimento de algoritmos de controle de congestionamento e de escalonadores de pacotes voltados para as características do MPQUIC pode beneficiar ainda mais os usuários e contribuir no alcance de melhores resultados.

Agradecimentos

Os autores agradecem o apoio da UFPR e auxílio financeiro do CNPq e da CAPES, processo # 99999.000404/2016-00.

Referências

[Albaladejo et al. 2016] Albaladejo, M. B., Leith, D. J., and Manzoni, P. (2016). Measurement-based modelling of lte performance in dublin city. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*, pages 1–6. IEEE.

[Alheid et al. 2016] Alheid, A., Doufexi, A., and Kaleshi, D. (2016). A study on mptcp for tolerating packet reordering and path heterogeneity in wireless networks. In *Wireless Days (WD), 2016*, pages 1–7. IEEE.

[Bagnulo 2011] Bagnulo, M. (2011). Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6181, IETF.

[Becke et al. 2012] Becke, M., Dreibholz, T., Adhari, H., and Rathgeb, E. P. (2012). On the fairness of transport protocols in a multi-path environment. In *Communications (ICC), 2012 IEEE International Conference on*, pages 2666–2672. IEEE.

[Clemente 2016] Clemente, L. (2016). A quic implementation in pure go. `https://github.com/lucas-clemente/quic-go`. Acessado em 21/05/2018.

[De Coninck and Bonaventure 2017] De Coninck, Q. and Bonaventure, O. (2017). Multipath quic: Design and evaluation. In *13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT 2017). http://multipath-quic.org*.

[Dhamdhere et al. 2012] Dhamdhere, A., Luckie, M., Huffaker, B., Elmokashfi, A., Aben, E., et al. (2012). Measuring the deployment of ipv6: topology, routing and performance. In *Proceedings of the 2012 Internet Measurement Conference*, pages 537–550. ACM.

[Garcia-Saavedra et al. 2017] Garcia-Saavedra, A., Karzand, M., and Leith, D. J. (2017). Low delay random linear coding and scheduling over multiple interfaces. *IEEE Transactions on Mobile Computing*, 16(11):3100–3114.

[J. Iyengar and M. Thomson 2018] J. Iyengar, E. and M. Thomson, E. (2018). QUIC: A UDP-Based Multiplexed and Secure Transport. Technical report, IETF.

[Kaup et al. 2015] Kaup, F., Wichtlhuber, M., Rado, S., and Hausheer, D. (2015). Can multipath tcp save energy? a measuring and modeling study of mptcp energy consumption. In *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*, pages 442–445. IEEE.

[Khalili et al. 2013] Khalili, R., Gast, N., Popovic, M., et al. (2013). Opportunistic linked-increases congestion control algorithm for mptcp.

[Kuhn et al. 2014] Kuhn, N., Lochin, E., Mifdaoui, A., Sarwar, G., Mehani, O., and Boreli, R. (2014). Daps: Intelligent delay-aware packet scheduling for multipath transport. In *Communications (ICC), 2014 IEEE International Conference on*, pages 1222–1227. IEEE.

[Le and Bui 2018] Le, T.-A. and Bui, L. X. (2018). Forward delay-based packet scheduling algorithm for multipath tcp. *Mobile Networks and Applications*, 23(1):4–12.

[Li et al. 2016] Li, M., Lukyanenko, A., Ou, Z., Yla-Jaaski, A., Tarkoma, S., Coudron, M., and Secci, S. (2016). Multipath Transmission for the Internet: A Survey. *IEEE Communications Surveys & Tutorials*.

[Lim et al. 2017] Lim, Y.-s., Nahum, E. M., Towsley, D., and Gibbens, R. J. (2017). Ecf: An mptcp path scheduler to manage heterogeneous paths. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 147–159. ACM.

[Paasch et al. 2014] Paasch, C., Ferlin, S., Alay, O., and Bonaventure, O. (2014). Experimental evaluation of multipath tcp schedulers. In *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, pages 27–32. ACM.

[Paasch et al. 2013] Paasch, C., Khalili, R., and Bonaventure, O. (2013). On the benefits of applying experimental design to improve multipath tcp. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pages 393–398. ACM.

[Pearce and Zeadally 2015] Pearce, C. and Zeadally, S. (2015). Ancillary impacts of multipath tcp on current and future network security. *IEEE Internet Computing*, 19(5):58–65.

[Possati et al. 2018] Possati, D., Silva, B., Santos, A., and Nogueira, M. (2018). Mitigação de ataque low-rate dos no protocolo mptcp. In *Workshop de Gerência e Operações de Rede (WGRS) - SBRC 2018*.

[Ramaboli et al. 2012] Ramaboli, A. L., Falowo, O. E., and Chan, A. H. (2012). Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues. *Journal of Network and Computer Applications*, 35(6):1674–1690.

[Secci et al. 2014] Secci, S., Pujolle, G., Nguyen, T. M. T., and Nguyen, S. C. (2014). Performance–cost trade-off strategic evaluation of multipath tcp communications. *IEEE Transactions on Network and Service Management*, 11(2):250–263.

[Silva et al. 2018] Silva, B., Steuck, I., Santos, A., and Nogueira, M. (2018). Escalonador de pacotes para transmissão por multi-caminhos não-correlacionados em redes heterogêneas sem fio. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2018)*.

[Yedugundla et al. 2016] Yedugundla, K., Ferlin, S., Dreibholz, T., Alay, Ö., Kuhn, N., Hurtig, P., and Brunstrom, A. (2016). Is multi-path transport suitable for latency sensitive traffic? *Computer Networks*, 105:1–21.