

Escalonador de Pacotes para Transmissão por Multi-caminhos Não-Correlacionados em Redes Heterogêneas sem Fio

Benevid Felix^{1,2}, Igor Steuck¹, Aldri Santos¹, Michele Nogueira¹

¹Núcleo de Redes Sem-Fio e Redes Avançadas (NR2)
Universidade Federal do Paraná (UFPR)

²Universidade do Estado de Mato Grosso (UNEMAT)

{benevid, islopes, aldri, michele}@inf.ufpr.br

Abstract. *Academic and industrial efforts seek to improve the performance of heterogeneous wireless networks to provide the minimum requirements in terms of latency and throughput for delay-sensitive applications, such as on-line gaming, voice and video streams. Multipath Transmission Control Protocol (MPTCP) is designed to meet this demand. Currently, MPTCP provides a redundant scheduler that replicates data packets in multiple paths to mitigate heterogeneity negative effects. However, shared bottlenecks among multiple paths reduce the redundant scheduler benefits and restrain the MPTCP performance. Thus, this work presents RED (REdundant Diversity scheduling), a scheduler that prioritizes packet replication through uncorrelated paths. RED selects the paths and replicates packets based on the Spearman's correlation coefficient. Results show that RED reduces delay and increases the use of low correlated paths, significantly improving transmission performance.*

Resumo. *Esforços acadêmicos e industriais buscam melhorar o desempenho das redes heterogêneas sem fio para que ofereçam os requisitos mínimos de latência e vazão às aplicações sensíveis ao atraso, tais como jogos online, e fluxos de voz e vídeo. O Protocolo de Controle de Transmissão Multi-caminhos (MPTCP) foi projetado para atender a essa demanda. Ele provê um escalonador redundante, que replica os pacotes de dados em múltiplos caminhos, para contrabalançar os efeitos negativos da heterogeneidade. Entretanto, gargalos compartilhados entre os caminhos fim-a-fim reduzem os benefícios providos pelo escalonador. Desta forma, este trabalho apresenta RED (REdundant Diversity scheduling), um escalonador que prioriza a replicação dos pacotes por caminhos não-correlacionados. RED replica os pacotes nos caminhos com base no coeficiente de correlação de Spearman. Os resultados demonstram que RED prioriza o uso de caminhos com baixa correlação e reduz a sobrecarga na rede com um menor número de pacotes replicados, além de prover uma melhora significativa no desempenho da transmissão.*

1. Introdução

Nos últimos anos, os dispositivos móveis passaram a incorporar múltiplas interfaces de rede sem fio (e.g., WiFi e Celular), permitindo associações simultâneas (*multihoming*) com diferentes Redes Heterogêneas Sem Fio (HetNets). Apesar dos dispositivos *multihoming* serem maioria atualmente, a conexão de dados TCP ainda está associada a uma

única rede de acesso (ou seja, a um único IP). Além disso, a transição entre diferentes redes de acesso interrompe a conexão, comprometendo aplicações sensíveis ao atraso, tais como aplicações de voz sob IP e chamadas de vídeo [Lima et al. 2009]. Estas limitações têm sido superadas com o uso do protocolo MPTCP (*Multipath TCP*), uma extensão do protocolo TCP. O MPTCP permite ao dispositivo móvel associar uma conexão a múltiplas redes de acesso e assim prover múltiplos caminhos entre o nó de origem e o nó de destino.

Embora o MPTCP seja promissor, seu uso sobre HetNets resulta em muitos desafios devido às características heterogêneas dos caminhos (e.g., atraso, taxa de perdas, largura de banda disponível, e outras) atravessados por cada sub-fluxo. A heterogeneidade dos caminhos dificulta a entrega ordenada dos pacotes e isto resulta em maiores atrasos e conseqüentemente uma diminuição na vazão da transmissão. Isto afeta principalmente as aplicações sensíveis ao atraso, como jogos online, transmissões em tempo real (áudio e vídeo) e tráfego Web. O MPTCP implementa mecanismos para reduzir estes efeitos, como os mecanismos de retransmissão e penalização. Contudo, esses mecanismos saturam o sub-fluxo de menor latência e acabam limitando a utilização dos demais sub-fluxos.

O uso redundante dos caminhos tem sido uma das abordagens empregadas para reduzir o impacto negativo causado pela heterogeneidade [Lopez et al. 2015, Wang et al. 2016a, Frommgen et al. 2016, Hunger and Klein 2016]. Essas abordagens têm como elemento crucial um escalonador de pacotes de dados que os replica sobre os caminhos disponíveis (i.e., sub-fluxos). O projeto desse escalonador segue a premissa de que os caminhos possuem uma alta diversidade, ou seja, não compartilham enlaces. Porém, isto não é válido em redes reais, como a Internet, em que as regras de roteamento tradicionais guiam os pacotes de um fluxo TCP por uma mesma rota [Nguyen et al. 2017]. Como o MPTCP utiliza sub-fluxos TCP, as regras de roteamento os consideram como fluxos TCP distintos, podendo ser transmitidos por caminhos compartilhados. Os caminhos compartilhados podem atravessar gargalos, experienciando problemas como o *bufferbloat* [Felix et al. 2017], que limitam os benefícios da redundância. As abordagens multi-camadas, em que os protocolos da camada de rede transmitem os sub-fluxos por caminhos disjuntos, podem prover uma maior diversidade para as transmissões multi-caminhos [Lopez et al. 2016]. Entretanto, isto pode criar um engessamento indesejável entre os protocolos das camadas adjacentes.

Para evitar o engessamento das camadas, são empregadas técnicas estatísticas para analisar os sub-fluxos e assim indicar se eles atravessam caminhos compartilhados [Rubenstein et al. 2002]. Contudo, no cenário de HetNets, isto é um desafio. O estado dos sub-fluxos sofre grandes variações, gerando ruídos na medição de informações, como do RTT (*Round Trip Time*), e provocando muitos falsos positivos [Liao et al. 2011]. Uma alternativa é medir o atraso OWD (*One Way Delay*) para reduzir os ruídos. Porém, a medição do atraso OWD gera mais sobrecarga na rede ao adicionar informações no cabeçalho dos pacotes de dados [Liao et al. 2011]. O MPTCP por padrão adiciona uma sobrecarga de informações de controle no cabeçalho dos pacotes. Isto se agrava com o uso do escalonador redundante, que replica os pacotes e mede o atraso OWD. Desta forma, para identificar os caminhos compartilhados, considera-se o uso de técnicas passivas, as quais se apoiam apenas em informações disponíveis para reduzir a sobrecarga. Contudo, em situações de congestionamento, as amostras dos atrasos observadas nos caminhos podem ter um comportamento monotônico, ou seja, as

variáveis dos caminhos, como RTT, tendem a mudar juntas, mas não necessariamente a uma taxa constante. [Hayes et al. 2014]. Isto reduz a efetividade de técnicas passivas [Rubenstein et al. 2002], que normalmente utilizam o coeficiente de correlação linear de Pearson. Este coeficiente avalia uma relação linear entre duas variáveis contínuas e pode não indicar corretamente a correlação no contexto das HetNets.

Este trabalho propõe RED (do inglês, *REdundant Diversity scheduling*), um novo escalonador redundante para o MPTCP que prioriza a replicação dos pacotes por caminhos não-correlacionados. O RED melhora o escalonador redundante do MPTCP, identificando e controlando os sub-fluxos que atravessam caminhos compartilhados. Para identificar os sub-fluxos não compartilhados, RED toma como base o coeficiente de correlação de Spearman. Desta forma, RED fica menos suscetível às variações da distribuição das amostras em situações de congestionamento para avaliar a correlação entre os caminhos. RED utiliza apenas as informações do RTT para o cálculo da correlação. Os resultados dos experimentos demonstram que RED aumenta os benefícios da redundância, replicando os pacotes para os caminhos com menor correlação, e com isso reduz o atraso e aumenta a vazão. Além disso, ele provê uma maior resiliência para as transmissões multi-caminhos, beneficiando principalmente as aplicações sensíveis ao atraso.

O restante do trabalho está organizado como segue. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve a proposta do escalonador RED. A Seção 4 detalha a metodologia de avaliação. A Seção 5 discute os resultados obtidos. A Seção 6 apresenta as conclusões e direções futuras.

2. Trabalhos Relacionados

Diferentes trabalhos apresentam propostas de escalonadores redundantes. Em [Lopez et al. 2015, Lopez et al. 2016], os autores propuseram uma estratégia de redundância para o MPTCP em que os pacotes de dados são replicados em múltiplos sub-fluxos. A estratégia focou em aumentar a confiabilidade na transmissão de dados em sistemas SCADA (*Supervisory Control and Data Acquisition*) contra falhas de transmissão. Já os escalonadores redundantes como rMPTCP [Hunger and Klein 2016], ReMPTCP [Frommgen et al. 2016] e MPTCP-L [Wang et al. 2016b] replicam os pacotes de dados sobre os caminhos para compensar as flutuações do atraso e reduzir o problema do reordenamento de pacotes. Em uma versão mais recente do rMPTCP [Hunger et al. 2016], os autores propuseram controlar a quantidade de réplicas de pacotes criadas a fim de limitar o número de caminhos utilizados e reduzir a sobrecarga de dados na rede. De um modo geral, os benefícios destes escalonadores redundantes são limitados se os caminhos são compartilhados. Para este fim, em [Lopez et al. 2016] os autores propuseram uma abordagem em que o protocolo MPLS (*Multiprotocol Label Switching*) garante a diversidade de caminhos ao MPTCP, tendo como base o escalonador redundante proposto anteriormente em [Lopez et al. 2015]. Para garantir a diversidade, o protocolo MPLS encaminha os pacotes provenientes dos sub-fluxos MPTCP por caminhos disjuntos. Embora seja eficiente, este tipo de abordagem cria um engessamento indesejável entre as camadas, dificultando sua utilização na Internet.

Para não criar este engessamento, alguns trabalhos buscaram identificar os caminhos compartilhados utilizando apenas informações fim-a-fim (em nível de transporte) sobre os sub-fluxos [Liao et al. 2011, Liu et al. 2011,

Singh and Reddy 2013, Hayes et al. 2014, Hu et al. 2016]. A maioria dos trabalhos tem como [Rubenstein et al. 2002], um dos percursos em identificar gargalos entre fluxos TCP utilizando medições fim-a-fim em nível de transporte. A técnica de Rubenstein *et al.* empregou o coeficiente de correlação de Pearson para determinar a auto-correlação e a correlação cruzada entre dois caminhos. Liao *et al.* [Liao et al. 2011] aplicaram este mesmo coeficiente, contudo, sugeriram um esquema mais leve para a obtenção do atraso de uma via (OWD). [Singh and Reddy 2013] propuseram o protocolo de transporte multi-caminhos MPERT, que utiliza a técnica de [Rubenstein et al. 2002] para detecção de caminhos compartilhados. Os resultados mostraram que o MPERT possui um desempenho melhor que o MPTCP apenas em cenários mais homogêneos. Hayes *et al.* [Hayes et al. 2014] sugeriram que as variações dos atrasos (RTT) em situações de congestionamento não seguem uma distribuição normal e as variáveis dos caminhos podem não ter uma relação linear entre si. Isto tem grande impacto nas técnicas baseadas em [Rubenstein et al. 2002]. Neste sentido, os autores propuseram um método embasado em três medidas estatísticas: disparidade, variabilidade e frequência. A primeira conta o número de medições OWD abaixo ou acima da média; a segunda quantifica o desvio padrão absoluto da média; e a terceira conta e normaliza o número de vezes que uma média de curto prazo (período curto) intercepta uma média de longo prazo (período longo). Estas estatísticas são usadas para agrupar os sub-fluxos com valores similares. [Ferlin et al. 2016] aplicaram esta técnica em um novo algoritmo de controle de congestionamento (CC) para o MPTCP. Os resultados mostraram uma leve melhora no desempenho comparado com outros algoritmos.

O OWD reduz os ruídos do RTT, contudo, as técnicas empregadas em sua obtenção acarretam uma sobrecarga na transmissão ao utilizar, por exemplo, a opção de *timestamp* no cabeçalho dos pacotes [Liao et al. 2011, Hayes et al. 2014]. Os algoritmos redundantes já produzem uma sobrecarga adicional na rede com a replicação e isto só aumentaria com a obtenção do OWD. Além disso, os dados obtidos (RTT) pode não apresentar uma relação linear entre si dada a dificuldade de sincronizar as amostras dos diferentes caminhos. Neste sentido, propõe-se um escalonador redundante que emprega uma técnica de correlação que suporta um comportamento monotônico, ou seja, que não exige amostras sincronizadas.

3. Escalonador Redundante com Seleção de Caminhos Não-Correlacionados

Esta seção descreve o escalonador proposto RED (do inglês, *REdundant Diversity scheduling*) e seu esquema de identificação de caminhos correlacionados, denominado RED *Correlated Path Identification* (CPI). O objetivo do escalonador RED é prover um melhor desempenho às aplicações sensíveis ao atraso em transmissões multi-caminhos nas Het-Nets. Primeiro, apresenta-se o método empregado pelo RED para identificar a correlação entre os caminhos (RED-CPI). Segundo, detalha-se o processo de escalonamento do RED.

3.1. Identificando os caminhos não compartilhados

As políticas de roteamento da Internet tradicionalmente mapeiam cada par de endereços para uma única rota entre os nós de origem (N_s) e destino (N_d) [Rohrer et al. 2014]. Neste caso, o caminho de ida (P_{ij}) e o de volta (P_{ji}) podem ser assimétricos, sendo i e j um endereço IP de N_s e N_d , respectivamente. Contudo, as regras de roteamento não impedem que os caminhos de uma mesma conexão MPTCP compartilhem um ou mais enlaces na

rede [Nguyen et al. 2017]. Conforme ilustra a Figura 1, os nós de origem N_s e de destino N_d possuem dois endereços IP cada. Desta forma, N_s pode estar associado ao IP_{s1} ou ao IP_{s2} , ou seja, $N_s = \{IP_{s1}, IP_{s2}\}$; e N_d pode estar associado a IP_{d1} ou IP_{d2} , ou seja, $N_d = \{IP_{d1}, IP_{d2}\}$. Utilizando o protocolo MPTCP, estes dois nós podem combinar estes endereços e formar uma malha de $N_s \times N_d$ sub-fluxos. Os sub-fluxos, por sua vez, podem ser compreendidos como caminhos virtuais (camada de transporte) identificados pelas combinações $P_{ij} = P_{1,1}, P_{1,2}, P_{2,1}, P_{2,2}$. Os caminhos virtuais são mapeados sobre caminhos reais na rede. Como ilustrado na figura, o par de caminhos $\pi = P_{1,2}, P_{2,1}$ compartilha o nó (roteador) r_3 . Os caminhos compartilhados são indesejáveis pois limitam o desempenho da transmissão multi-caminhos e diminui a resiliência a falhas.

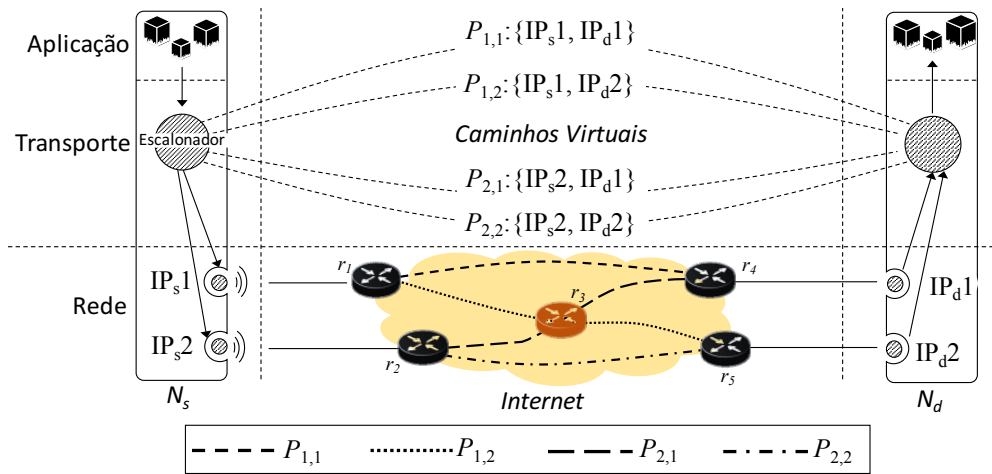


Figura 1. Caminhos reais vs. caminhos virtuais - adaptado de [Liao et al. 2011]

Diferente da camada de rede [Rohrer et al. 2014], as informações acessíveis aos protocolos da camada de transporte, como atraso (ϵ), taxa de perda (l), largura de banda disponível (bw) e outras, não permitem identificar precisamente o número de enlaces compartilhados por uma conexão fim-a-fim entre um dado par de caminhos (π). Por outro lado, os protocolos da camada de rede permitem determinar o número de enlaces compartilhados entre um par de caminhos $\pi = \{P_{ij}, P_{xy}\}$ [Narasimhan et al. 2012]. Os métodos estatísticos, como o proposto neste trabalho, indicam se um par de caminhos é compartilhado através do grau de correlação (c). O grau de correlação pode assumir qualquer valor entre -1 e 1 . Como ilustrado na Figura 2, c indica uma correlação positiva, quando o coeficiente tende a 1 ; nula (sem correlação), quando tende a 0 ; e negativa, quando existe uma relação inversamente proporcional entre as variáveis. Neste último caso o coeficiente tende a -1 .

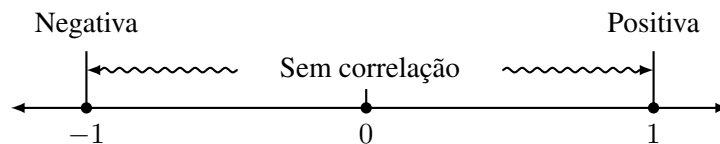


Figura 2. Coeficiente de correlação (c)

RED emprega um método estatístico (RED-CPI) para identificar a correlação entre um par de caminhos $\pi = \{P_{ij}, P_{xy}\}$. O método obtém informações de forma passiva

e similar a [Liao et al. 2011, Singh and Reddy 2013]. Ele utiliza dados do atraso (RTT) para identificar os caminhos compartilhados. Para reduzir os ruídos que afetam o RTT, especialmente nos cenários de HetNets, utiliza-se o atraso suavizado (SRTT, *Smoothed Round Trip Time*) no cálculo do coeficiente de correlação. Assim, dado um par de caminhos $\pi = \{P_{ij}, P_{xy}\}$, os valores de R_{ij} e R_{xy} são os SRTTs dos caminhos P_{ij} e P_{xy} .

Os métodos estatísticos baseados no coeficiente de correlação de Pearson, abordados na Seção 2, assumem que os dados coletados dos caminhos sigam uma distribuição normal e um comportamento linear entre as variáveis R_{ij} e R_{xy} . Contudo, como abordado em [Hayes et al. 2014], em situações de congestionamento as amostras do RTT não seguem uma distribuição normal. Com o objetivo de superar estas limitações, emprega-se o coeficiente de correlação de Spearman (c_s). Este coeficiente avalia a relação monotônica (linear ou não) entre duas variáveis contínuas ou ordinais. Neste tipo de relação as variáveis tendem a mudar juntas, mas não necessariamente a uma taxa constante. Assim, este trabalho considera que coeficiente de Spearman pode indicar com mais precisão a correlação entre os caminhos no cenário das HetNets, dadas as frequentes variações causadas por eventos como a mobilidade dos nós.

RED-CPI calcula a correlação entre pares de caminhos a partir de um conjunto de amostras de R_{ij} e R_{xy} pertencentes aos caminhos P_{ij} e P_{xy} , respectivamente. O coeficiente c_s é calculado para uma amostra de tamanho N em um intervalo de tempo τ . O tamanho da amostra é parametrizado para ambos os caminhos e obedece um esquema de janela deslizante. Conforme ilustra a Figura 3, ambas as janelas R_{ij} e R_{xy} possuem um tamanho N e são atualizadas respectivamente com os valores de cada SRTT, sendo $r_{ij} \in R_{ij}$ e $r_{xy} \in R_{xy}$, lidos a cada instante t . Esta atualização das amostras ocorre de forma independente em cada janela. Isto implica que as amostras não estejam necessariamente sincronizados em função do tempo.

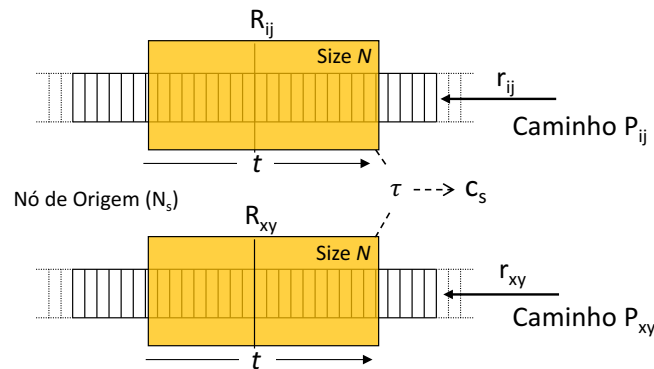


Figura 3. Esquema de janela deslizante

Em [Liao et al. 2011], os autores sugerem que a sincronização das amostras é um passo chave para o cálculo da correlação. No entanto a heterogeneidade dos caminhos torna esta tarefa muito difícil. O RTT é calculado com base no recebimento de um pacote de reconhecimento (ACK) para uma determinada sequência de pacotes. E mesmo que fosse calculado o OWD, eles também seriam diferentes e sofreriam variações em função do estado dos caminhos. Fazer a sincronização seria uma tarefa custosa e limitada pelo número de associações correspondentes com a faixa de tempo determinada para esta sincronização. Neste sentido, RED-CPI não se preocupa com esta sincronização uma vez

emprega o coeficiente de correlação de Spearman, que suporta este tipo de relação (monotônica). Com isto, a relação entre os valores R_{ij} e R_{xy} com a distribuição das amostras têm um menor impacto sobre o cálculo correlação.

O cálculo da correlação entre as amostras das janelas R_{ij} e R_{xy} segue as etapas: (i) classificar e atribuir os valores dos postos para as N amostras de cada caminho; (ii) calcular o coeficiente de correlação entre cada par de caminho P_{ij} e P_{xy} . Na etapa (i), atribui-se postos para cada $r_{ij} \in R_{ij}$ e $r_{xy} \in R_{xy}$, conforme a ordem crescente dos valores. Uma vez que o valor do SRTT se repete inúmeras vezes, a ocorrência de empates entre a classificação de dois valores ou mais valores torna-se quase inevitável. Quando isto ocorre, atribui-se a cada um deles a média dos postos que seriam atribuídos caso o empate não ocorresse. Por exemplo, se os 7 primeiros valores do SRTT forem iguais a 0. Como estes são os valores menores da janela, eles compartilham as classificações 1, 2, 3, 4, 5, 6 e 7 (total de 28) dando a cada valor um índice médio de 4 ($28 \div 7$). Assim, a próxima classificação seria 8, uma vez que as classificações de 1 a 7 já foram atribuídas. A saída desta etapa corresponde aos conjunto de postos K_{ij} e K_{xy} , atribuídos aos valores em R_{ij} e R_{xy} , respectivamente. Na etapa (ii), calcula-se o coeficiente de correlação c_s utilizando os postos $k_{ij} \in K_{ij}$ e $k_{xy} \in K_{xy}$ empregando a Eq. 1. Onde ρ denota o coeficiente de Pearson usual para variáveis classificadas; $cov(K_{ij}, K_{xy})$ é a covariância (Eq. 2); e $\sigma_{K_{ij}}, \sigma_{K_{xy}}$ (Eq. 3.1) se refere ao desvio padrão das variáveis classificadas.

$$c_s = \rho = \frac{cov(K_{ij}, K_{xy})}{\sigma_{K_{ij}} \sigma_{K_{xy}}} \quad (1)$$

$$cov(K_{ij}, K_{xy}) = \frac{1}{n} \left[\sum_{i=1}^n k_{ij} k_{xy} - \frac{1}{n} \left(\sum_{i=1}^n k_{ij} \right) \left(\sum_{i=1}^n k_{xy} \right) \right] \quad (2)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (k_{ij} - \bar{k}_{ij})^2} \quad (3)$$

3.2. RED: Escalonador redundante

O escalonador em uma transmissão multi-caminhos tem como função distribuir os segmentos (pacotes de dados) através de um ou mais sub-fluxos. O MPTCP não possui um escalonador padronizado pelo IETF. Contudo, sua implementação tem uma estrutura modular e é disponibilizada em diferentes sistemas operacionais [Paasch et al. 2013]. Isto permite optar por diferentes escalonadores. Existem três opções de políticas de escalonamento: **Round-Robin** (RR), LRF (**Lowest-RTT-First**) e **Redundante**. O primeiro distribui os dados de modo circular entre os sub-fluxos disponíveis; o segundo distribui os dados primeiramente nos caminhos com menor RTT; e o terceiro replica os dados sobre os caminhos disponíveis. O escalonador redundante provê uma comunicação de alta confiabilidade, por entregar os segmentos de dados de forma redundante através dos múltiplos caminhos disponíveis. Este modelo tem o objetivo de equalizar as diferenças entre os caminhos e suprimir as falhas de forma instantânea.

O escalonador redundante do MPTCP não se preocupa em geral com a diversidade de caminhos. Isto limita o potencial do escalonador redundante no MPTCP, principalmente quando aplicado em redes heterogêneas sem fio, haja visto que essas redes

sofrem com as grandes variações nas características dos seus enlaces. Assim, este trabalho propõe um escalonador redundante que replica os pacotes prioritariamente para os caminhos que apresentem baixa correlação, denominado RED. RED possui duas fases, como ilustra a Figura 4. A primeira fase corresponde ao método RED-CPI, descrito na Subseção 3.1, e tem como função atualizar o grau de correlação c_s entre cada par de caminhos $\{P_{ij}, P_{x,y}\}$. A segunda fase, tal como em outros escalonadores redundantes [Becke 2014], corresponde de designar (alocar) para os caminhos, em um determinado instante de tempo, os segmentos e suas respectivas réplicas. Com RED, a alocação inicia a partir do caminho com menor grau de correlação entre os demais, ou seja, onde $c_s(P_{ij}, P_{x,y}) \rightarrow 0$. Neste sentido, uma réplica é alocada para o caminho com menor grau de correlação com anteriores, ou seja, aqueles caminhos que já receberam a réplica, e que cujo valor de c_s seja menor que um parâmetro (limiar) de corte (φ).

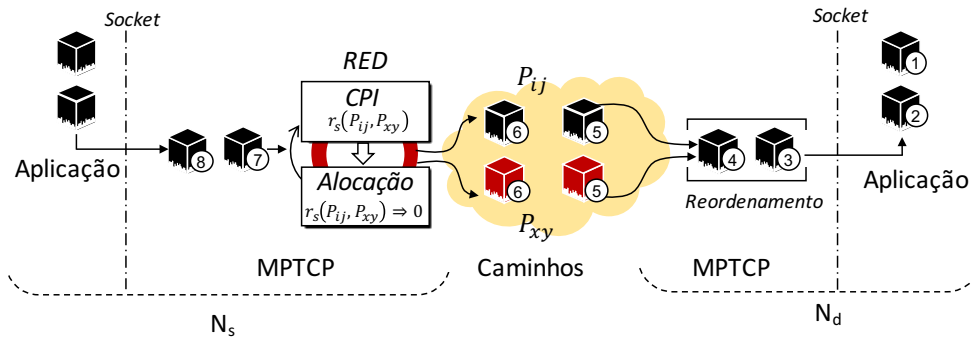


Figura 4. As fases do escalonador RED

4. Avaliação

Esta seção descreve a metodologia de avaliação do RED utilizada nos experimentos, incluindo o cenário de HetNet empregado, os parâmetros e limiares definidos e os detalhes de implementação em kernel Linux.

4.1. Cenários e Parâmetros

Avaliar esta proposta através de experimentos em redes reais é uma tarefa muito difícil, uma vez que os gargalos são desconhecidos e não é possível controlar com precisão os caminhos compartilhados e não compartilhados necessários para testar a eficiência do escalonador RED. Para ter um maior controle dos experimentos, implementou-se o escalonador no módulo do MPTCP [Paasch et al. 2013], versão 0.92, Kernel Linux v.4.4.95+ (Debian 8), e empregou-se o Mininet para emular a topologia de rede necessária. O cenário de avaliação (Figura 5) representa um cenário comum de HetNet, em que um nó móvel (N_s) tem acesso a múltiplas redes sem fio e troca dados a um nó de destino (N_d). N_s estabelece uma conexão multi-caminhos (MPTCP) com N_d . N_s está associado aos pontos de acesso WiFi (r_1) e celular (r_2) e estabelece uma malha de 2×2 sub-fluxos com N_d . Estes atravessam os caminhos virtuais $P_{1,1}$, $P_{1,2}$, $P_{2,1}$, sendo que $P_{1,2}$ e $P_{2,1}$ compartilham o gargalo entre os roteadores r_3 e r_4 (linha vermelha).

A definição dos valores dos parâmetros utilizados teve como referência os trabalhos de [Liao et al. 2011, Ferlin et al. 2016]. Os parâmetros consistem em: porcentagem de perda do enlace (PP_e), largura de banda do enlace (LB_e) e atraso do enlace (A_e), descritos na Tabela 1. Os valores dos parâmetros dos caminhos são definidos com objetivo

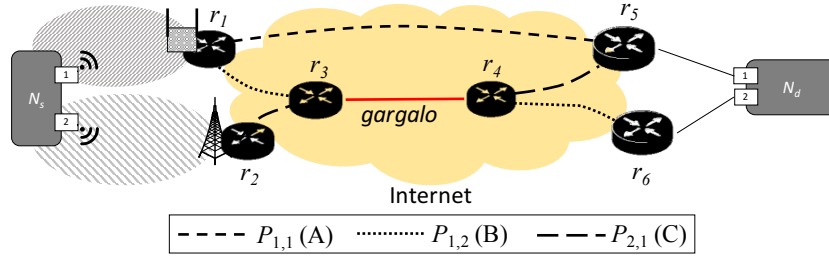


Figura 5. Cenário de avaliação

de que assemelhem-se a um cenário HetNet realístico. Além destes parâmetros, os experimentos foram realizados utilizando um algoritmo de controle de congestionamento desacoplado (TCP Cubic) [Ha et al. 2008] para não limitar o desempenho individual de cada caminho, e o gerenciador de caminhos *Full-mesh* [Paasch et al. 2013], que cria a malha de sub-fluxos conforme a combinação de endereços IP entre N_s e N_d . Em todos os experimentos são utilizados 3 caminhos, chamados de A , B e C para facilitar a compreensão. A é um rótulo usado para $P_{1,1}$, B é usado para $P_{1,2}$, e C para $P_{2,1}$. Os caminhos B e C compartilham o gargalo e sofrem interferência de um tráfego MPTCP. Por outro lado, o caminho A não sofre interferência de tráfego e nem compartilha qualquer enlace. Para efeito de comparação, o método RED-CPI foi comparado com o método MSCM [Liao et al. 2011]. Este método emprega o coeficiente de Pearson e utiliza o OWD como dados de entrada para calcular a correlação entre pares de caminhos.

Parâmetros	$r_1 \iff r_5$	$r_2 \iff r_6$	$r_3 \iff r_4$	outros
PP_e (% pct)	0,001	0,0001	0,001	0,00
LB_e (Mbit/s)	150	100	20	100
A_e (ms)	5	15	40	1

Tabela 1. Parâmetros

4.2. Detalhes de Implementação

O escalonador RED foi implementado no kernel Linux¹. A implementação do MPTCP para o kernel Linux segue o mesmo esquema de implementação do TCP, onde métodos de escalonamento e de controle de congestionamento são implementados como módulos. Assim, o escalonador RED também foi implementado como um módulo. O modelo teórico do escalonador, descrito na Seção 3, precisou ser ajustado em termos de implementação devido às restrições de implementação existentes no kernel Linux. Por exemplo, os módulos e as funções implementadas para o kernel Linux devem evitar de fazer uso de cálculos com ponto flutuante, uma vez que existem arquiteturas que não possuem suporte a este tipo de operação. Visto que o método RED executa algumas divisões durante o cálculo da correlação entre caminhos, optou-se por multiplicar os dividendos por uma potência de 10^3 para que o resultado, truncado, tivesse pouca perda de precisão. Outro consiste na replicação dos pacotes. O modelo define que todos os pacotes devem ser replicados prioritariamente a todos os caminhos que possuam baixa correlação entre si. Apesar de mantida a prioridade, apenas pacotes que se encontram na fila e não foram

¹<https://github.com/ccsc-research/red-scheduler>.

enviados por nenhum caminho são replicados, uma vez que não há sentido em replicar um pacote que já foi enviado com sucesso.

5. Resultados

Esta seção está dividida em duas partes: (i) a primeira apresenta os resultados do método de correlação RED-CPI comparando-o com método MSCM (do Inglês *Multipath Selection for Concurrent Multipath Transfer*); (ii) a segunda parte apresenta os resultados do desempenho do escalonador RED contrastando-o com o escalonador redundante do MPTCP. A escolha do MSCM se deve por ele empregar o coeficiente de correlação de Pearson, o qual sua eficiência depende do comportamento dos dados. Além disso, o MSCM utiliza o atraso de uma via (OWD), o qual possui menos ruídos que o RTT e que, segundo os pesquisadores, favorece o cálculo da correlação [Liao et al. 2011].

5.1. Eficiência do Método de Correlação

Inicialmente foram realizados experimentos com o escalonador redundante do MPTCP emulando o cenário descrito na Seção 4.1 a fim de compreender seu funcionamento. A partir dos traços (i.e., pcap) obtidos nestes experimentos, foram extraídas as informações do RTT e do atraso de uma via (OWD) para cada caminho. Estas informações foram normalizadas em função do tempo relativo de experimento de $T_s = 60s$, e utilizadas como valores de entrada para analisar o comportamento dos métodos RED-CPI e MSCM. Para o cálculo do OWD, utilizado pelo método MSCM, utilizaram-se as informações do RTT coletadas em N_s e N_d para subtrair o atraso de ida e de volta separadamente. Para calcular o RTT suavizado (*Smoothed RTT*), utilizado pelo RED-CPI, empregou-se o cálculo definido em [Paxson et al. 2011]. Para ambos os métodos utilizou-se uma janela deslizante com o tamanho $N = 50$ e um mesmo intervalo de $t = 0.01ms$ para o cálculo da correlação. O intuito desta avaliação foi de medir a eficiência dos métodos em indicar a correlação entre os pares de caminhos a partir das respectivas informações de entrada.

A Figura 6 ilustra os valores do coeficiente de correlação c_s entre cada par de caminhos $(A - B)$, $(A - C)$, $(B - C)$, coletados durante um experimento. O MSCM indica para os pares de caminhos $(A - B)$ e $(A - C)$, os quais são não-compartilhados, uma tendência de correlação negativa. No entanto, para o par de caminhos $(B - C)$, que são compartilhados, os valores possuem uma maior concentração acima de 0 e indicam uma correlação positiva. Os resultados para o RED-CPI possuem uma distribuição mais equilibrada para os pares de caminhos não compartilhados $(A - B)$ e $(A - C)$, tendo 0 como ponto central na distribuição dos dados (mediana). No caso do par de caminhos $(B - C)$, os valores possuem uma concentração muito próxima de 1, indicando, na maioria dos casos, um forte grau de correlação.

O método RED-CPI se mostrou mais eficiente que o MSCM. A Figura 7.a apresenta a taxa de falsos positivos (TFP) para os caminhos não compartilhados conforme um limiar de corte (φ). Para cada φ , a TFP indica o percentual de erro dos métodos em indicar os caminhos não-correlacionados como correlacionados. Neste caso, o RED-CPI apresentou uma menor TFP que o MSCM para $\varphi \leq 0.8$. A Figura 7.b demonstra através de uma curva ROC a variação da sensibilidade (Verdadeiros Positivos) e especificidade (Falsos Positivos) para os diferentes valores de φ . Para o método MSCM, a curva

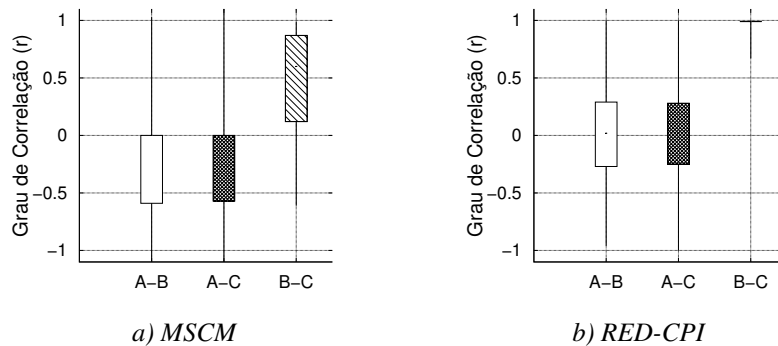


Figura 6. Distribuição dos valores de c_s em um experimento

apresenta uma maior variação de especificidade, indicando um maior número de falsos positivos, como foi apresentado na Figura 7.a. No entanto, o RED-CPI apresenta uma menor variação na taxa de falsos positivos. Isto indica que o método possibilita utilizar um limiar de corte, para separar o grau que indica ou não a correlação entre um par de caminhos, em um intervalo de $0.8 \geq \varphi \leq 0.9$. A partir destes resultados, definiu-se um limiar de $\varphi = 0.8$ na implementação do RED (RED-CPI) em kernel Linux.

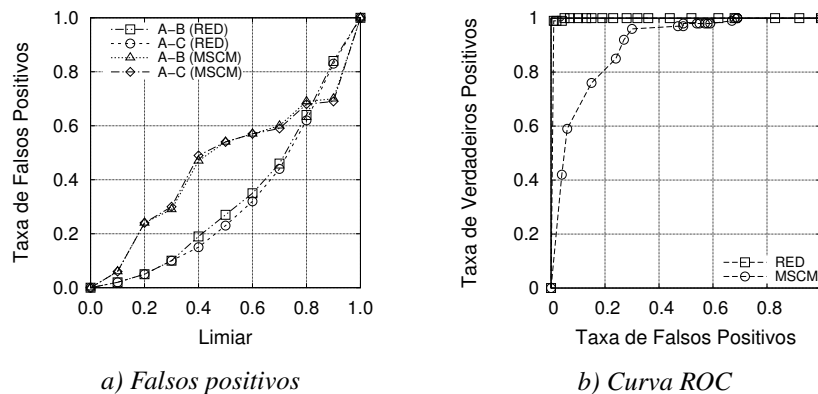


Figura 7. Eficiência do RED vs. MSCM

5.2. Eficiência do Escalonador

Esta subseção apresenta os resultados dos experimentos com o escalonador RED em comparação com o escalonador redundante do MPTCP, aqui denominado apenas MPTCP, para o mesmo cenário descrito na Seção 4. Os gráficos da Figura 8 apresentam os resultados do atraso (RTT) e da vazão de cada caminho A , B e C para os experimentos com ambos escalonadores. Cabe ressaltar que o caminho A não possui gargalos e portanto,

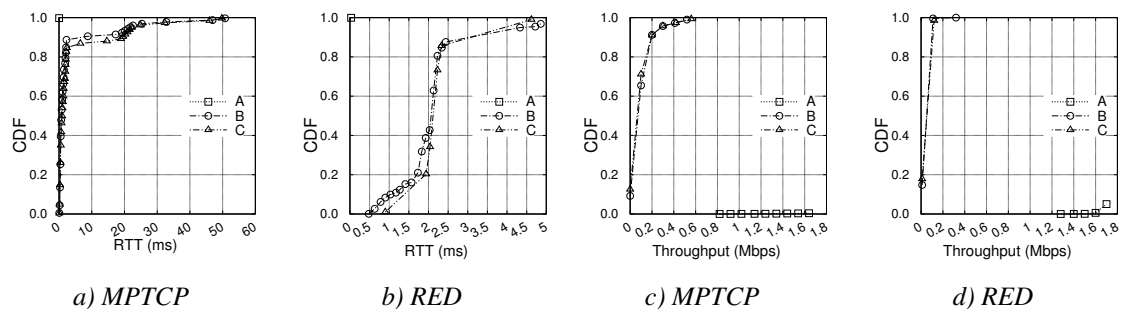


Figura 8. Função de distribuição acumulada (FDA) do atraso e vazão

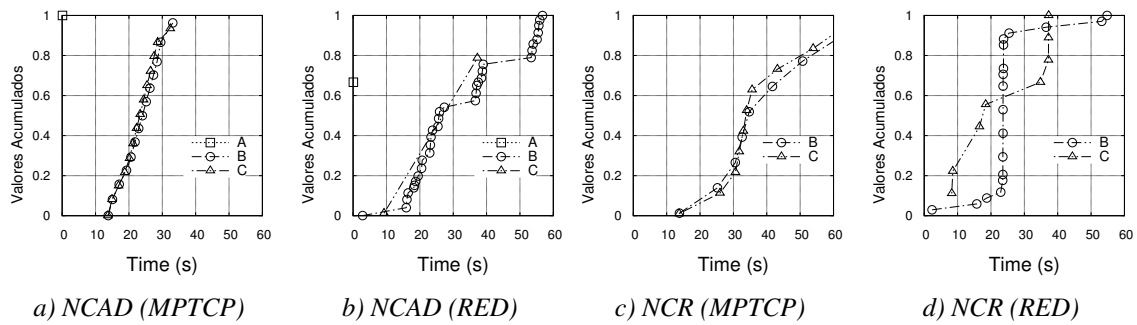


Figura 9. Dados acumulados de retransmissões e ACKs duplicados

não tem uma variação expressiva do RTT nos experimentos. No entanto, o gráfico da Figura 8.b, que corresponde ao RED, apresenta uma redução significativa no atraso máximo alcançado quando comparado aos resultados do MPTCP, vistos na Figura 8.a. Isto ocorre porque o RED prioriza os pares de caminhos com baixa correlação, o que pode ser visto comparando a granularidade dos pontos plotados para os caminhos B e C entre os gráficos da Figura 8.a e da Figura 8.b. A granularidade do caminho B é maior que do caminho A nos experimentos com o RED, e ambos menores que nos experimentos com o MPTCP. No entanto, o RED reduz a vazão dos caminhos B e C, deslocando mais dados para o caminho A, que alcança uma vazão mínima de 0.6 (0.8/1.2) vezes maior que o MPTCP, ilustrado nos gráficos das Figuras 8.c e 8.d, respectivamente.

Além das métricas de atraso e vazão, utiliza-se mais duas métricas: (i) o **número cumulativo de ACKs duplicados (NCAD)**; e o **número cumulativo de retransmissões (NCR)**. O NCAD contabiliza o recebimento de reconhecimentos duplicados (Dup-ACK) para indicar a proporção de segmentos recebidos fora de ordem em N_s . Por outro lado, o NCR contabiliza o número de retransmissões, as quais são ocasionadas por perda de pacotes ou pelo recebimento de Dup-ACKs para pacotes ainda não reconhecidos. Conforme ilustra a Figura 9, o RED possui um menor NCAD, ou seja, causa um menor impacto no reordenamento de pacotes e conseqüentemente um menor NCR. Isto se deve ao fato do RED replicar mais pacotes pelos caminhos com menor grau de correlação, o que otimizando o uso A e reduz o uso dos caminhos B e C, reduzindo assim o NCAD e o NCR.

Com objetivo de avaliar a sobrecarga do RED e do MPTCP na rede, calculou-se o **número médio de réplicas (NMR)** criadas, que se refere ao número médio de segmentos encaminhados (replicados) para mais que um caminho. Executou-se 35 repetições e calculou-se o NMR com um intervalo de confiança de 95%. Conforme ilustra a Figura 10,

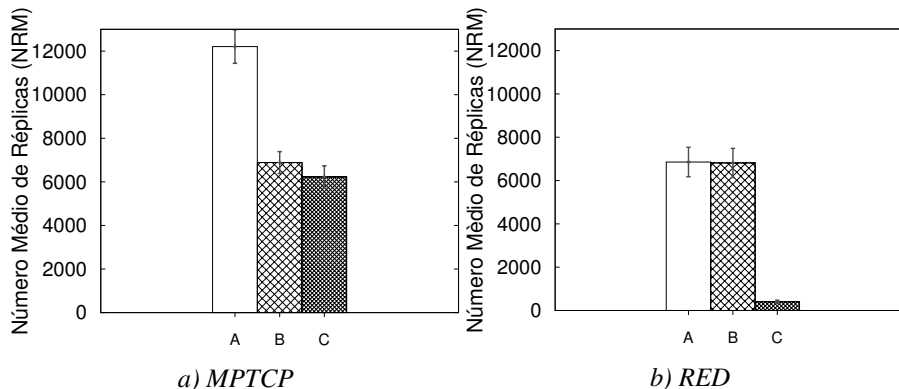


Figura 10. Sobrecarga dos escalonadores

o RED reduz o NMR total em relação ao MPTCP. No entanto, o NMR do caminho C com o RED foi muito inferior ao observado com o MPTCP. Isto ocorre porque o RED prioriza o caminho que possui um menor c_s em relação aos demais. Neste caso, o caminho A , que é não correlacionado aos demais, acaba sendo priorizado para envio de segmentos. Como o RED escolhe o segundo caminho em função do seu grau de correlação com o caminho A , ele recebe um número de réplicas similar. No entanto, o caminho C apenas recebe réplicas se possuir uma baixa correlação com os caminhos A e B . Como geralmente não possui, recebe um pequeno número de réplicas.

6. Conclusões

Este trabalho apresentou RED (*REdundant Diversity Scheduling*), um novo escalonador redundante para o MPTCP que apenas replica os pacotes para os caminhos com menor grau de correlação. Para isto, RED possui um método de análise de correlação de caminhos (RED-CPI) e um abordagem para replicação dos pacotes. Os resultados mostraram que o RED-CPI resulta em uma baixa taxa de falsos positivos na identificação dos caminhos não correlacionados, quando comparado com o método MSCM, um dos métodos representativos da literatura. Além disso, RED-CPI possui uma maior taxa de verdadeiros positivos. Os resultados dos experimentos utilizando a implementação do RED em Kernel Linux confirmam a eficiência do método RED-CPI. Além disso, a abordagem de replicação de pacotes do RED reduz o uso dos caminhos compartilhados e melhora o uso dos demais caminhos. Como resultado, o escalonador RED reduz o atraso e aumenta a vazão mínima dos caminhos não compartilhados. Como trabalhos futuros, pretende-se avaliar a escalabilidade do escalonador.

Referências

- Becke, M. (2014). *Revisiting the IETF Multipath Extensions on Transport Layer*. PhD thesis, University of Duisburg-Essen, Duisburg - Germany.
- Felix, B., Santos, A., Kantarci, B., and Nogueira, M. (2017). CD-ASM: A new queuing paradigm to overcome bufferbloat effects in HetNets. In *IEEE PIMRC*.
- Ferlin, S., Alay, Ö., Dreibholz, T., Hayes, D. A., and Welzl, M. (2016). Revisiting congestion control for multipath TCP with shared bottleneck detection. In *IEEE INFOCOM*, páginas 1–9.
- Frommgen, A., Erbschäuber, T., Buchmann, A., Zimmermann, T., and Wehrle, K. (2016). REMP TCP: Low Latency Multipath TCP. In *IEEE ICC*, páginas 1–7.
- Ha, S., Rhee, I., and Xu, L. (2008). CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS*, 42(5):64–74.
- Hayes, D. A., Ferlin, S., and Welzl, M. (2014). Practical passive shared bottleneck detection using shape summary statistics. In *IEEE LCN*, páginas 150–158.
- Hu, B., Xing, L., Wang, Z., and Liu, N. (2016). MLCS: A Multi-level Correlation Scheduling Algorithm for Multipath Transport. In *IEEE ICOIN*, páginas 166–171.
- Hunger, A. and Klein, P. A. (2016). Equalizing latency peaks using a redundant multipath-TCP scheme. In *IEEE ICOIN*, páginas 184–189.

- Hunger, A., Klein, P. A., and Verbunt, M. H. (2016). Evaluation of the Redundancy-Bandwidth Trade-Off and Jitter Compensation in rMPTCP. In *IEEE IFIP NTMS*, páginas 1–5.
- Liao, J., Wang, J., Li, T., and Zhu, X. (2011). Introducing Multipath Selection for Concurrent Multipath Transfer in the Future Internet. *Elsevier Comp. Networks*, 55(4):1024–1035.
- Lima, M. N., Dos Santos, A. L., and Pujolle, G. (2009). A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 11(1):66–77.
- Liu, Y., Wang, B., Xu, K., and Ma, Z. (2011). Pacc: A path associativity congestion control and throughput model for multi-path tcp. *Elsevier Procedia Comp. Science*, 4:1278–1287.
- Lopez, I., Aguado, M., Pinedo, C., and Jacob, E. (2015). Scada systems in the railway domain: enhancing reliability through redundant multipath tcp. In *IEEE ITSC*, páginas 2305–2310.
- Lopez, I., Aguado, M., Ugarte, D., Mendiola, A., and Higuero, M. (2016). Exploiting redundancy and path diversity for railway signalling resiliency. In *IEEE ICIRT*, páginas 432–439.
- Narasimhan, J., Venkataswami, B. V., Groves, R., and Hoose, P. (2012). Traceflow. Technical report, IETF.
- Nguyen, H. D. D., Phung, C. D., Secci, S., Felix, B., and Nogueira, M. (2017). Can MPTCP secure Internet communications from man-in-the-middle attacks? In *IEEE CNSM*.
- Paasch, C., Barré, S., et al. (2013). Multipath TCP in the Linux kernel. Available from {www.multipath-tcp.org}.}
- Paxson, V., Allman, M., Chu, J., and Sargent, M. (2011). Computing TCP’s retransmission timer. RFC 6298, IETF.
- Rohrer, J. P., Jabbar, A., and Sterbenz, J. P. (2014). Path Diversification for Future Internet End-to-End Resilience and Survivability. *Springer Telecommunication Systems*, 56(1):49–67.
- Rubenstein, D., Kurose, J., and Towsley, D. (2002). Detecting shared congestion of flows via end-to-end measurement. *IEEE/ACM Trans. on Networking (TON)*, 10(3):381–395.
- Singh, A. and Reddy, A. N. (2013). Multi Path PERT. In *IEEE ICCCN*, páginas 1–9.
- Wang, W., Zhou, L., and Sun, Y. (2016a). Improving Multipath TCP for Latency Sensitive Flows in the Cloud. In *IEEE CloudNet*, páginas 45–50.
- Wang, W., Zhou, L., and Sun, Y. (2016b). Improving Multipath TCP for Latency Sensitive Flows in the Cloud. In *IEEE Cloudnet*, páginas 45–50.