

Universidade do Minho

Comportamento do Protocolo Multipath TCP Sobre Diferentes Tecnologias de Rede

Pedro Filipe Pereira Lopes
(Licenciado)

Dissertação para obter o grau de Mestre em
Redes e Serviços de Comunicações

Trabalho realizado sob a orientação de:

Prof. Paulo Carvalho
Prof. Pedro Sousa

12 de Junho de 2014

To know your Enemy, you must become your Enemy.

Sun Tzu, *The Art of War*

Agradecimentos

Eu gostaria de agradecer à Universidade do Minho pela oportunidade em trabalhar num tema de dissertação bastante interessante no âmbito do mestrado em Redes e Serviços de Comunicações. Especialmente, eu gostaria de agradecer ao meu orientador, Prof. Paulo Carvalho, e co-orientador, Prof. Pedro Sousa, pelo conhecimento teórico e técnico dado ao longo deste trabalho, permitindo-me completar com sucesso este documento.

Eu também gostaria de agradecer à minha família pelo suporte moral e financeiro ao longo de todos estes anos de estudo, permitindo-me hoje apresentar este tema de dissertação.

Abstract

With the constant evolution of the Internet, the services it provides to users have been growing exponentially in the last decade with the introduction of cellular networks of 3rd Generation and most recently, 4th Generation (LTE). However, it is not all good news because with the increase in number of services and users, the requests for network resources increased significantly, and some of these features (in particular, bandwidth) cannot be fully utilized due to restrictions in the TCP/IP protocol stack. These restrictions are especially at the level of the end systems and at the network itself. One of the restrictions is the inability of TCP transport protocol to use multiple paths for sending data between two or more systems over a TCP connection. It was based on this limitation that researchers have developed an extension to TCP, designated by Multipath TCP (MPTCP), in order to allow the use of multiple paths during a single TCP/IP session. This is expected to bring benefits in terms of efficient use of resources within the network and thus improve the user experience through increasing performance and attaining better resilience to network failures.

Many studies have been conducted based on the use of this new extension, ranging from the analysis of core networks in order to demonstrate MPTCP versatility and adaptability in the use of backup links, to the study of mobility cases, focusing on the usefulness of MPTCP before the movement of mobile devices without occur losses of connectivity. However, none of the analysed articles specifically addressed along with MPTCP operation of certain application protocols. Thus, the goal of this work is to study the MPTCP extensions of TCP protocol and, based on two simple network architectures, assess its added value regarding the quality of service (QoS) offered to applications.

The results presented here are meant primarily to demonstrate the potential of MPTCP under the operation of an application protocol, File Transfer Protocol (FTP). This brings positive results in several aspects, such as the significant decrease in transfer time of a given file, good load sharing in the available routes, etc., all compared with the results achieved by TCP under the same conditions. The positive results obtained, suggest that MPTCP brings QoS improvements to the functioning of FTP.

Keywords

MPTCP; quality of service (QoS); applications

Resumo

Com a constante evolução da Internet, os serviços que esta garante aos seus utilizadores têm tido um crescimento exponencial na última década com a introdução das redes celulares de 3ª Geração e, mais recentemente, 4ª Geração (LTE). No entanto, nem tudo são boas notícias porque com o aumento do número de serviços e de utilizadores, os pedidos de recursos à rede aumentaram significativamente, sendo que alguns desses recursos (em particular, largura de banda) não podem ser totalmente utilizados devido a restrições na própria pilha protocolar TCP/IP. Estas restrições encontram-se, sobretudo ao nível dos sistemas finais e da própria rede. Uma das restrições é a incapacidade do protocolo de transporte TCP em utilizar múltiplos caminhos para o envio de dados entre duas ou mais sistemas durante uma comunicação TCP. Foi com base nesta limitação que investigadores da área desenvolveram uma extensão ao protocolo TCP, denominada por Multipath TCP (MPTCP), com o objetivo de permitir a utilização simultânea de vários caminhos durante uma única sessão TCP/IP. Isto permite trazer vantagens ao nível da eficiência na utilização de recursos dentro da rede e assim, melhorar a experiência do utilizador através de um maior desempenho e melhor resiliência a falhas na rede.

Muitos estudos já foram realizados com base na utilização desta nova extensão, desde análises ao seu funcionamento no core de determinadas redes de forma a demonstrar a sua versatilidade e adaptabilidade na utilização das ligações de *backup*, até ao estudo em casos de mobilidade, onde pretendiam obter resultados que demonstrassem a utilidade do MPTCP perante a deslocação de dispositivos móveis sem que ocorresse a perda de conectividade. No entanto, nenhum dos artigos estudados abordava especificamente o funcionamento do MPTCP juntamente com determinados protocolos aplicativos. Assim, neste trabalho o objetivo é estudar/investigar a extensão MPTCP do protocolo de transporte TCP/IP e, com base em duas arquiteturas de rede a planejar, efetuar uma avaliação da sua mais-valia face à qualidade de serviço (QoS) oferecida às aplicações.

Os resultados aqui apresentados, visam, sobretudo, demonstrar as potencialidades do MPTCP no âmbito do funcionamento do protocolo aplicativo, File Transfer Protocol (FTP). Apresentando resultados positivos nos mais diversos aspetos, tais como o decréscimo significativo no tempo de transferência de um determinado ficheiro, a boa divisão da carga pelas rotas disponíveis, etc, tudo isto comparando com os resultados alcançados pelo TCP nas mesmas condições. Os resultados positivos, indicam que o MPTCP traz melhorias de QoS ao funcionamento do FTP.

Palavras Chave

MPTCP; qualidade de serviço; aplicações

Conteúdo

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	2
1.3	Contribuições Originais	3
1.4	Organização da Dissertação	3
2	Análise Teórica do Multipath TCP	5
2.1	Trabalho Relacionado	6
2.2	Introdução ao Multipath TCP	7
2.3	Visão global do modo de operação	8
2.4	Arquitetura	9
2.5	Controlo de congestão	10
2.5.1	Algoritmo OLIA	11
2.6	Considerações sobre a interface aplicacional	12
2.7	Considerações sobre a segurança	13
2.8	Preocupações com as <i>Middleboxes</i>	14
3	Descrição dos Ambientes Simulados	15
3.1	Cenários de teste elaborados	16
3.1.1	Cenário Ethernet - Ethernet	16
3.1.2	Cenário WiFi - Ethernet	17
3.2	Instalação das máquinas	17
3.2.1	Hardware	17
3.2.2	Software	18
3.3	Configuração do endereçamento e rotas	19
3.4	Metodologia	20
3.5	Wireshark	21
4	Discussão dos Resultados	23
4.1	Cenário Ethernet - Ethernet	24
4.1.1	TCP	24
4.1.2	MPTCP com COUPLED	27

4.1.3	MPTCP com OLIA	30
4.1.4	Comparação dos resultados	33
4.2	Cenário WiFi - Ethernet	35
4.2.1	TCP	35
4.2.2	MPTCP com COUPLED	37
4.2.3	MPTCP com OLIA	39
4.2.4	Comparação dos três casos	42
5	Conclusões e Trabalho Futuro	45
	Bibliografia	49
	Apêndice A Wireshark	A-1
A.1	Cenário Ethernet - Ethernet	A-2
A.1.1	TCP	A-2
A.1.2	MPTCP com COUPLED	A-3
A.1.3	MPTCP com OLIA	A-4
A.2	Cenário WiFi - Ethernet	A-5
A.2.1	MPTCP com COUPLED	A-5
A.2.2	MPTCP com OLIA	A-6

Lista de Figuras

2.1	Pilha protocolar com a introdução do MPTCP.	7
2.2	Exemplo do modo de operação de um cenário MPTCP.	8
2.3	Visão atual da Internet.	9
3.1	Cenário Ethernet - Ethernet.	16
3.2	Cenário WiFi - Ethernet.	17
4.1	Tabela de Conversações TCP do cenário Ethernet - Ethernet.	24
4.2	Débito total da conexão TCP do cenário Ethernet - Ethernet.	25
4.3	Número de pacotes transmitidos no cenário Ethernet - Ethernet.	25
4.4	Número total de pacotes perdidos no cenário Ethernet - Ethernet.	26
4.5	Número de pacotes retransmitidos no cenário Ethernet - Ethernet.	26
4.6	Tabela de Conversações TCP com o COUPLED no cenário Ethernet - Ethernet.	27
4.7	Débito total e individual da conexão com o COUPLED no cenário Ethernet - Ethernet.	28
4.8	Número de pacotes transmitidos com o COUPLED no cenário Ethernet - Ethernet.	28
4.9	Número total de pacotes perdidos com o COUPLED no cenário Ethernet - Ethernet.	29
4.10	Número de pacotes retransmitidos com o COUPLED no cenário Ethernet - Ethernet.	29
4.11	Tabela de Conversações TCP com o OLIA no cenário Ethernet - Ethernet.	30
4.12	Débito total e individual da conexão com OLIA no cenário Ethernet - Ethernet.	31
4.13	Número de pacotes transmitidos com OLIA no cenário Ethernet - Ethernet.	31
4.14	Número total de pacotes perdidos com OLIA no cenário Ethernet - Ethernet.	32
4.15	Número de pacotes retransmitidos com OLIA no cenário Ethernet - Ethernet.	32
4.16	Comparação do débito no cenário Ethernet - Ethernet.	34
4.17	Tabela de Conversações TCP no cenário WiFi - Ethernet.	35
4.18	Débito total da conexão no cenário WiFi - Ethernet.	36
4.19	Número de pacotes transmitidos no cenário WiFi - Ethernet.	36
4.20	Tabela de Conversações TCP com o COUPLED no cenário WiFi - Ethernet.	37
4.21	Débito total e individual da conexão com o COUPLED no cenário WiFi - Ethernet.	37
4.22	Número de pacotes transmitidos com o COUPLED no cenário WiFi - Ethernet.	38
4.23	Número total de pacotes perdidos com o COUPLED no cenário WiFi - Ethernet.	39
4.24	Número de pacotes retransmitidos com o COUPLED no cenário WiFi - Ethernet.	39
4.25	Tabela de Conversações TCP com o OLIA no cenário WiFi - Ethernet.	40

4.26 Débito total e individual da conexão com OLIA no cenário WiFi - Ethernet.	40
4.27 Número de pacotes transmitidos com OLIA no cenário WiFi - Ethernet.	41
4.29 Número de pacotes retransmitidos com OLIA no cenário WiFi - Ethernet.	42
4.30 Comparação do débito no cenário WiFi - Ethernet.	43
A.1 Número de ACKs perdidos com o TCP no cenário Ethernet - Ethernet.	A-2
A.2 Número de pacotes desordenados com o TCP no cenário Ethernet - Ethernet.	A-2
A.3 Número de ACKs perdidos com o COUPLED no cenário Ethernet - Ethernet.	A-3
A.4 Número de pacotes desordenados com o COUPLED no cenário Ethernet - Ethernet.	A-3
A.5 Número de ACKs perdidos com o OLIA no cenário Ethernet - Ethernet.	A-4
A.6 Número de pacotes desordenados com o OLIA no cenário Ethernet - Ethernet.	A-4
A.7 Número de ACKs perdidos com o COUPLED no cenário WiFi - Ethernet.	A-5
A.8 Número de pacotes desordenados com o COUPLED no cenário WiFi - Ethernet.	A-5
A.9 Número de ACKs perdidos com o OLIA no cenário WiFi - Ethernet.	A-6
A.10 Número de pacotes desordenados com o OLIA no cenário WiFi - Ethernet.	A-6

Lista de Tabelas

3.1	Tabela de endereçamento da máquina servidor	19
3.2	Tabela de endereçamento do desktop cliente	19
3.3	Tabela de endereçamento da portátil cliente	19
3.4	Tabela de endereçamento da máquina CORE	20
4.1	Tabela de Comparação do Débito no cenário Ethernet - Ethernet	33
4.2	Tabela de Comparação do número de Pacotes no cenário Ethernet - Ethernet	34
4.3	Tabela de Comparação dos RTT no cenário Ethernet - Ethernet	34
4.4	Tabela de Comparação do Débito no cenário WiFi - Ethernet	43
4.5	Tabela de Comparação do número de Pacotes no cenário WiFi - Ethernet	44
4.6	Tabela de Comparação dos RTT no cenário WiFi - Ethernet	44

Abreviações

Application Programming Interface (API)

File Transfer Protocol (FTP)

Internet Protocol (IP)

Internet Engineering Task Force (IETF)

Internet Service Provider (ISP)

Linked-Increases Algorithm (LIA)

Multipath TCP (MPTCP)

Multi-Provider TCP (MTCP)

Multi-path TCP (M/TCP)

Network Address Translation (NAT)

Next-generation Transport (Tng)

Opportunistic Linked Increases Algorithm (OLIA)

Parallel TCP (pTCP)

Quality of Service (QoS)

Round-Trip Time (RTT)

Resilient Overlay Networks (RON)

Stream Control Transmission Protocol (SCTP)

Transmission Control Protocol (TCP)

1

Introdução

Conteúdo

1.1	Motivação	2
1.2	Objetivos	2
1.3	Contribuições Originais	3
1.4	Organização da Dissertação	3

1.1 Motivação

É do conhecimento global que nos últimos anos entramos na era da informação, onde a informação é poder. No que diz respeito à Internet, ela têm tido um crescimento constante em termos de serviços e utilizadores. Por exemplo, sabe-se que os *datacenters* possuem múltiplos caminhos paralelos sendo muitos deles caminhos de *backup* para permitir uma redundância de excelência. No entanto, muitos dos caminhos encontram-se sobre cargas elevadas, o que provoca um elevado congestionamento e com isto uma perda significativa na Qualidade de Serviço entregue. Para resolver este problema e ao mesmo tempo aproveitar-se dos caminhos, os especialistas desenvolveram protocolos de *routing* que levavam em conta algumas métricas de QoS para selecionar os caminhos de menor custo. No entanto, esta solução não era ideal devido a problemas no tempo de resposta e escalabilidade. Portanto, novas pesquisas e soluções são necessárias para trazer um bom desempenho e robustez às aplicações no global. Uma das possibilidades é o aperfeiçoamento do modo de funcionamento do protocolo de transporte TCP. Como se sabe, o TCP é um protocolo que foi desenvolvido num tempo onde a Internet estava a dar os primeiros passos e apesar da sua evolução é um protocolo que utiliza apenas um caminho por cada sessão TCP/IP. Isto levantou a seguinte questão de se o protocolo de transporte TCP pode ser capaz de utilizar múltiplos caminhos em simultâneo para transmitir os dados?

Neste contexto, os investigadores da área das telecomunicações desenvolveram e apresentaram muitas extensões ao protocolo TCP, entre elas encontra-se o *Multipath TCP* (MPTCP) proposto por um grupo de trabalho no *Internet Engineering Task Force* (IETF). Este protocolo distribui os dados de uma única conexão TCP ao longo de múltiplos subfluxos, sendo que cada um pode tomar um caminho diferente através da rede. Com a utilização desta extensão, espera-se que os recursos da rede sejam utilizados de forma mais eficaz, garantindo aos utilizadores uma melhoria significativa na sua experiência graças a um aumento no rendimento e melhorias na resiliência a falhas da rede. Atendendo aos potenciais benefícios do MPTCP, este trabalho foca-se no estudo e teste do MPTCP.

1.2 Objetivos

Esta dissertação, no âmbito do mestrado em redes e serviços de comunicações da Universidade do Minho, têm como objetivo o estudo e familiarização da extensão MPTCP através da sua implementação em duas arquiteturas de rede a planear. Através das experiências a efetuar-se e consequentemente análises aos resultados obtidos, pretende-se demonstrar as potencialidades na aplicação desta extensão nas atuais infraestruturas de rede de modo a assegurar um melhor desempenho nos dois seguintes parâmetros:

1. **Optimização do débito** - Isto vai permitir um melhor uso da largura de banda, o que irá resultar numa entrega mais rápida da informação transmitida.
2. **Balanceamento de carga** - Um benefício no balanceamento de carga vai permitir a redução da carga nas ligações mais congestionadas, movendo a carga para as ligações menos conges-

tionadas.

Como objetivo secundário, pretende-se analisar o desempenho do MPTCP numa vertente mais aplicacional, isto é, obtenção de resultados que demonstrem as vantagens que o uso do *multipath* em detrimento do *singlepath* proporciona a uma determinada aplicação.

1.3 Contribuições Originais

Nesta dissertação de mestrado, encontram-se resultados que visam a aplicação do MPTCP em duas arquiteturas de rede distintas, uma arquitetura homogênea que possui interfaces de rede com características semelhantes e uma arquitetura heterogênea que utiliza duas tecnologias de acesso à rede distintas, Ethernet e WiFi. Em ambas arquiteturas de rede são efetuados testes que visam demonstrar as vantagens introduzidas pelo MPTCP em dois aspetos fundamentais, uma optimização do débito agregado e um balanceamento de carga mais equitativo através da deslocação da carga das ligações mais congestionadas para as ligações menos congestionadas. Todos estes resultados são comparados com o desempenho obtido na aplicação do TCP nessas mesmas arquiteturas de rede, comprovando-se assim que o desempenho do MPTCP nunca deve ser pior que o desempenho do TCP no melhor dos caminhos. Por fim, a obtenção de melhores resultados nestes dois parâmetros proporcionam uma redução de custos nas atuais infraestruturas de rede porque não exige a necessidade de implementar-se novos dispositivos de forma a alcançar-se resultados semelhantes aos alcançados com a aplicação deste tipo de protocolos/extensões que utilizam o *multipath* disponível na rede.

1.4 Organização da Dissertação

O documento desta dissertação está organizada da seguinte maneira: 1 Introdução - onde se mencionaram as motivações inerentes à escolha deste tema, quais são os objetivos a que nos propusemos desenvolver neste tópico de dissertação, as principais contribuições que este documento traz para a área de investigação; 2 Análise Teórica do Multipath TCP - aqui pretende-se apresentar quais os protocolos/extensões desenvolvidos até hoje que dão uso às diversidades de rotas, apresentando as suas principais vantagens e desvantagens face ao MPTCP. Aqui também são analisados todos os detalhes inerentes ao modo de operação do MPTCP, bem como os principais problemas que a sua utilização levanta, indicando também quais as alterações necessárias a introduzir na arquitetura dos sistemas operativos e consequentes aplicações; 3 Descrição dos Ambientes Simulados - trata-se de um capítulo onde se irá explicar o procedimento de testes elaborado em ambos cenários de teste e também o *hardware* e *software* necessário para desenvolver este projeto; 4 Discussão dos Resultados - aqui apresenta-se e analisa-se os resultados alcançados no que diz respeito ao desempenho do MPTCP nos cenários de teste propostos; 5 Conclusões e Trabalhos Futuros - por último, apresenta-se as conclusões finais com base na análise aos resultados alcançados, visando os objetivos propostos nesta dissertação de mestrado.

2

Análise Teórica do Multipath TCP

Conteúdo

2.1 Trabalho Relacionado	6
2.2 Introdução ao Multipath TCP	7
2.3 Visão global do modo de operação	8
2.4 Arquitetura	9
2.5 Controlo de congestão	10
2.6 Considerações sobre a interface aplicacional	12
2.7 Considerações sobre a segurança	13
2.8 Preocupações com as <i>Middleboxes</i>	14

2.1 Trabalho Relacionado

Atualmente, existe um elevado número de novos protocolos de transporte [1–4] que suportam o uso de múltiplas rotas, sendo que todos apresentam soluções para resolver alguns ou mesmo todos os problemas descritos nesta tese de mestrado. Contudo, a maioria destes protocolos não têm como principal foco fazer qualquer tipo de modificações nas aplicações, mas nos mecanismos do protocolo necessários para implementar uma transmissão *multipath*. Como também acontece no MPTCP, os seus objetivos passam pela robustez a falhas nas rotas e na modificação dos termos necessários para o estabelecimento de uma rota. Portanto, apresenta-se apenas três desses protocolos de transporte porque são os mais referenciados nos diversos artigos estudados.

Uma das características mais importantes do *Stream Control Transmission Protocol* (SCTP) é a sua tolerância a falhas na rede, isto é, a transmissão de mensagens sobre múltiplas rotas é suportada, mas apenas para redundância. Portanto, o ideal seria adaptar esta característica na transmissão dos dados pelas diversas rotas simultaneamente [5]. Isto iria proporcionar uma melhor utilização da largura de banda e conseqüente melhor distribuição da carga por todas as rotas disponíveis. No entanto, para isto ser possível, tiveram de aplicar a extensão *Concurrent Multipath Transfer* (CMT) do SCTP [4] para fazer a divisão da carga. Mais detalhadamente, a aplicação desta extensão introduz três modificações no modo de operação do SCTP. Elas são as *Fast Retransmissions*, *appropriate congestion window growth* e *delayed SACKs*. Para uma informação mais detalhada sobre as alterações feitas, aconselha-se a análise aos artigos referenciados anteriormente.

O *parallel TCP* [1] é um cabeçalho que foi inserido depois do cabeçalho TCP e é considerado por muitos uma das propostas mais compreensivas. Ele foi desenvolvido com o intuito de agregar a largura de banda disponibilizada pela grande variedade de tecnologias de acesso à rede sem fios que, atualmente, um dispositivo móvel com múltiplas interfaces têm à sua disposição.

Para o pTCP agregar a largura de banda disponível, ele utiliza a combinação de certos mecanismos, tais como a dissociação do controlo de congestão e fiabilidade, o ajustamento dinâmico da janela, etc. No entanto, este cabeçalho é uma solução complexa, o que traz problemas na sua adaptabilidade aos atuais esquemas de rede.

Por último, aborda-se os protocolos MTCP [2] and M / TCP [3]. Ambos protocolos usam um único número de sequência juntamente com uma tabela de resultados no transmissor, este mantém um estado de congestão e realiza as retransmissões por rota. No entanto, a robustez no MTCP é menor em comparação com o M / TCP porque este utiliza uma única rota para enviar os ACKs. Este protocolo também foi unicamente criado para correr nas *Redundancy Overlay Networks* (RON), o que traz uma redução da sua aplicabilidade e eficiência.

Por outro lado, o que distingue o MPTCP dos restantes protocolos é o facto de ser uma extensão que atualmente está virtualmente normalizada no IETF. Além de ser mais robusto e propôr soluções para determinados problemas que não são referenciados nos outros protocolos estudados. Estes dizem respeito a problemas relacionados com o comportamento das *middleboxes* e de problemas no tamanho do *buffer* de receção. De salientar que estes problemas encontram-se analisados e

explicados nas seguintes secções.

2.2 Introdução ao Multipath TCP

O MPTCP foi desenvolvido com o objetivo de providenciar um serviço TCP que permitisse o estabelecimento de uma conexão sobre múltiplas rotas simultâneamente. Mas antes de se entrar numa explicação mais detalhada sobre o seu modo de operação, é importante salientar que o MPTCP deve ser compatível com o atual TCP para aumentar as suas hipóteses de ser implementado e que este também deve assumir que atualmente a maioria das máquinas possui múltiplos endereços, sendo este um sinal da existência de múltiplas rotas.

Como se pode ver na seguinte Figura 2.1, as modificações introduzidas na pilha protocolar TCP/IP pelo grupo de trabalho no IETF [6], passam pela divisão da camada de transporte em dois subníveis com a adição de uma camada MPTCP e uma camada "Subflow". Apartir deste ponto, o modo de operação do MPTCP deve ser transparente para as camadas superiores e, inclusive, para as camadas inferiores.

Com base no estudo conduzido, as extensões MPTCP foram consolidadas com o estabelecimento de quatro áreas que sofreram algumas modificações. Em primeiro lugar, ao nível do protocolo TCP, foram apresentadas modificações através da adição da habilidade de implementar diversas rotas simultâneamente, mais especificamente para sinalização, gestão de configuração, convergência de dados e término das sessões. Em segundo lugar, as alterações na arquitetura necessárias, onde as decisões ao mais alto nível são tomadas e onde é explicada a separação funcional no qual uma implementação MPTCP pode ser desenvolvida. Em terceiro lugar, atenta-se ao controlo de congestão, apresentando um novo e seguro algoritmo para acoplamento do comportamento das diversas rotas de forma a estas não provocarem qualquer tipo de dano para os outros utilizadores da rede. Por fim, algumas observações são ainda feitas ao nível da aplicação, discutindo-se o impacto que o MPTCP irá ter nas aplicações, quais aplicações que necessitarão o uso do MPTCP e como resultado desses fatores, quais são as extensões API que uma implementação MPTCP deve apresentar. Nas próximas secções irá-se abordar as modificações feitas nas quatro áreas com mais detalhe.

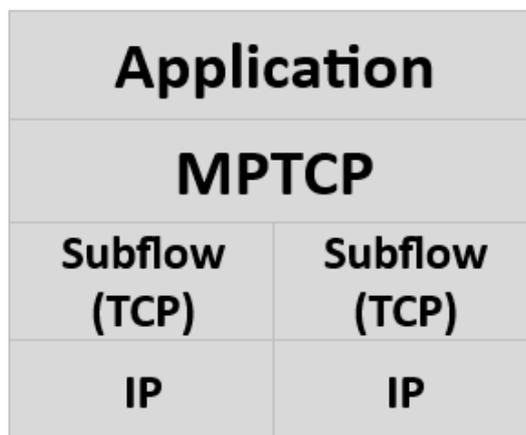


Figura 2.1: Pilha protocolar com a introdução do MPTCP.

2.3 Visão global do modo de operação

Nesta secção encontram-se explicadas as características chave do MPTCP ao mais alto nível, deixando informação mais detalhada para a leitura do artigo [6] do IETF. Como anteriormente mencionado, cada máquina mantém um conjunto de endereços associados com as múltiplas interfaces do terminal. O que o MPTCP faz é dividir os fluxos de dados com origem nas aplicações em segmentos e depois envia-los nas diferentes rotas pré-determinadas. Ao contrário do que se possa imaginar, o MPTCP também utiliza o mecanismo *three-way-handshake* para estabelecer a conexão. Ele apenas adiciona algumas opções às mensagens SYN, SYN/ACK, ACK. Sendo assim, vai-se explicar como a troca de mensagens é feita.

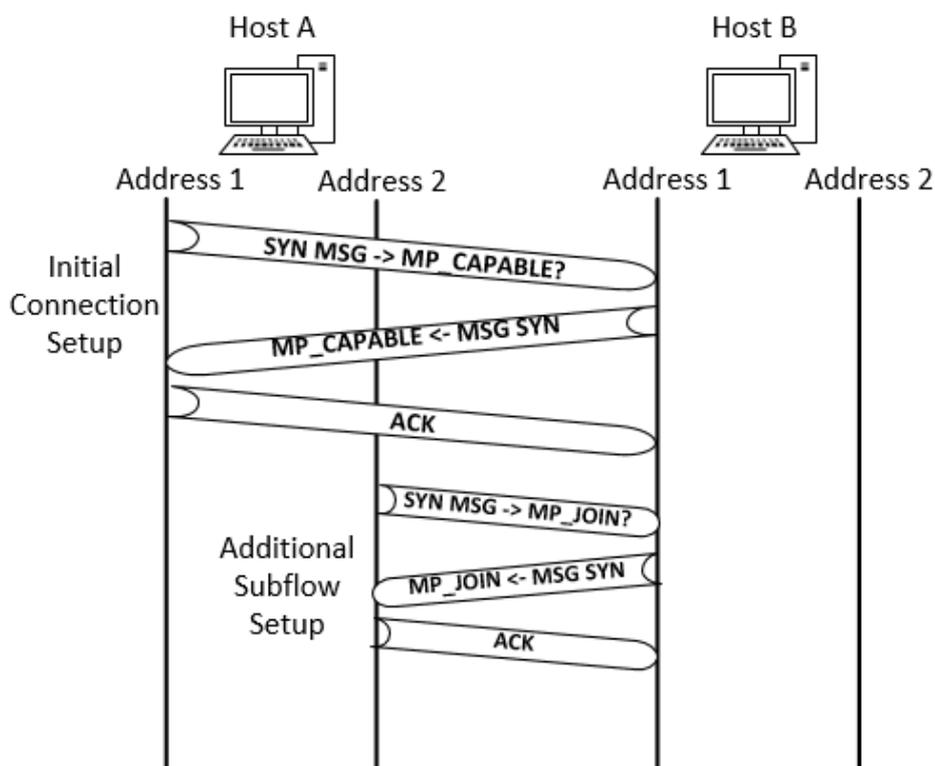


Figura 2.2: Exemplo do modo de operação de um cenário MPTCP.

O estabelecimento de uma conexão MPTCP é realizada através do envio de uma mensagem SYN com a opção MP_CAPABLE entre um dos endereços do cliente e um dos endereços do servidor. Isto demonstra que o cliente suporta uma comunicação MPTCP e, se o servidor também o suportar, vai enviar uma nova mensagem SYN com a opção MP_CAPABLE. A partir deste momento, é possível estabelecer subfluxos adicionais na conexão com o envio da opção MP_JOIN no pacote SYN. A Figura 2.2 exemplifica o estabelecimento de uma conexão MPTCP. Se por alguma razão, o servidor não suportar uma comunicação MPTCP, é assegurada a compatibilidade com o TCP para regressar a uma comunicação TCP regular. Além destas duas opções, existem outras para assegurar a adição e a remoção em caso do endereço alterar-se. É durante o estabelecimento do primeiro subfluxo que se estabelece o primeiro mecanismo de segurança numa comunicação MPTCP através da troca de

chaves encriptadas entre o cliente e o servidor. Trata-se de um algoritmo *hash-based* que o MPTCP utiliza e que mais tarde irá ser abordado com mais detalhe.

No que diz respeito aos dados transmitidos, para assegurar que estes são entregues ordenadamente à camada aplicacional e para assegurar que as retransmissões acontecem individualmente em cada subfluxo TCP, o MPTCP utiliza dois conjuntos de números de sequência. O primeiro ao nível da conexão (isto é, ao nível do MPTCP) é transmitido juntamente com as próprias opções do TCP. O segundo conjunto é utilizado juntamente com os subfluxos (isto é, no nível do TCP) e é transportado no campo estabelecido dos segmentos TCP.

2.4 Arquitetura

Antes de se abordar a arquitetura adotada pelo MPTCP [7] é importante mencionar quais os principais objetivos que o protocolo deve respeitar de forma a obter um bom desempenho nos mais variadíssimos aspetos.

Em primeiro lugar, ao nível operacional, uma conexão MPTCP sobre diversas rotas nunca deve alcançar pior débito que uma única conexão TCP na melhor das rotas e que o protocolo não deve ser menos resiliente a falhas que uma única rota TCP.

Em segundo lugar, ao nível da compatibilidade no que diz respeito às aplicações, o MPTCP deve sempre utilizar o mesmo modelo de serviço que o TCP, isto é, deve garantir uma entrega ordenada, fiável e orientada ao *byte*. Sendo assim, as aplicações devem garantir níveis de débito e resiliência nunca piores que uma conexão TCP sobre qualquer uma das rotas disponíveis e especialmente ser capaz de coexistir com as existentes APIs do TCP. No que diz respeito à compatibilidade na rede, o protocolo deve ser compatível com a atual Internet e ser capaz de ultrapassar os problemas apresentados pelas *middleboxes* tais como *firewalls*, *NATS* e *proxies* que tentam otimizar o desempenho das rotas. Ainda mais, o MPTCP deve trabalhar com trocas entre endereços IPv4 e endereços IPv6, isto é, a conexão pode operar em ambas as redes. A Figura 2.3 demonstra a visão atual da Internet.

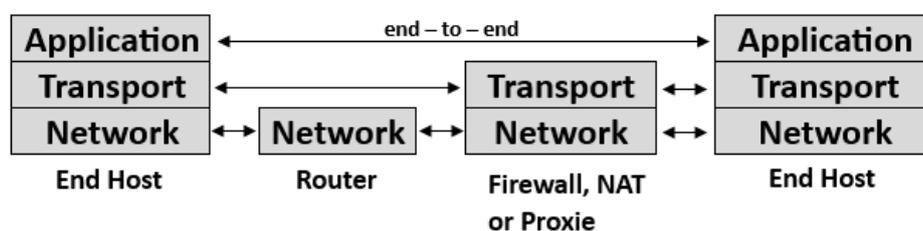


Figura 2.3: Visão atual da Internet.

Em terceiro lugar, o MPTCP deve ser compatível com os restantes utilizadores da rede, isto é, não deve afetar os utilizadores que utilizam uma comunicação TCP regular em meios partilhados, além do impacto que pode ocorrer devido a outro fluxo TCP. Outro importante aspeto é o facto de todos os fluxos MPTCP num meio partilhado terem de dividir a largura de banda equitativamente entre todos como acontece nos meios partilhados que utilizam o TCP.

Finalmente, em termos de segurança, um dos objetivos do MPTCP é nunca garantir um serviço menos seguro que o TCP, apesar das novas ameaças que este traz. A segurança deve ser assegurada com a combinação dos múltiplos mecanismos de segurança existentes no TCP e deve proteger contra as novas ameaças identificadas [8].

Depois de todos estes objetivos serem alcançados, pode-se definir uma nova arquitetura base. O planeamento da arquitetura MPTCP é baseado num outro protocolo, Tng (next-generation transport) [9]. A introdução do MPTCP trouxe uma nova pilha protocolar que pode ser vista na Figura 2.1, sendo, no entanto, necessário a implementação das seguintes funções. O *path manager* faz a descoberta das múltiplas rotas possíveis entre os dois dispositivos, para que no lado do emissor o *packet management* receba os fluxos de dados das aplicações pré-designadas a operar na rede e desempenhe as operações necessárias antes de transmiti-los a um dos subfluxos. De seguida, o subfluxo escolhido adiciona os números de sequência, faz o ACK e passa os dados para a rede. No lado do recetor, os dados recebidos são reordenados e transmitidos para o *packet management* que por sua vez faz a reordenação ao nível da conexão e envia-os para a respetiva aplicação. Por último, o componente responsável pelo *congestion management* juntamente com o *packet management*, agendam quais e com que frequência os segmentos estão prontos a ser enviados para o respetivo subfluxo.

A necessidade de se cumprir/respeitar os objetivos salientados nesta secção, fez com que determinadas soluções para certos tipos de problemas ao mais alto nível fossem reduzidas. Desta forma, é importante assegurar-se uma entrega ordenada dos segmentos e, portanto, é necessário saber-se qual o tamanho do *buffer* porque é este componente que no lado do recetor armazena os segmentos ordenados para posteriormente serem lidos pelas aplicações. Em contraste com o que acontece no tamanho do *buffer* de receção do TCP, este deve ser:

$$\text{Buffersize} = 2 \sum (BW_i) RTT_MAX \quad (2.1)$$

onde o RTT_max é o maior RTT de todos os subfluxos. Sendo assim, é prudente utilizar-se apenas as rotas mais rápidas para a conexão MPTCP, deixando as mais lentas para a *backup*. No que diz respeito ao *buffer* de emissão, a formula de cálculo é a mesma que se utilizou para o cálculo do *buffer* de receção.

2.5 Controlo de congestão

No MPTCP é proposto um novo algoritmo denominado por LIA (*Linked Increases Algorithm*) que visa corrigir os erros que os algoritmos atualmente em uso no TCP trazem num contexto *multipath*. Um dos problemas é a partilha equitativa dos recursos disponíveis nas rotas de um meio partilhado que se encontram cobertas por mais de um subfluxo. O ideal seria a obtenção de um estado denominado por "*resource pooling*", onde o transporte do tráfego fosse feito nas rotas menos conges-

tionadas e, com isto, assemelhando-se a uma única ligação com uma elevada capacidade, o que permitiria uma rede mais eficiente contra falhas.

Em primeiro lugar, o novo algoritmo para o controlo de congestão deve cumprir três objetivos para uma distribuição equitativa da carga em todas as rotas. O primeiro objetivo é melhorar o débito e isto consegue-se, se um fluxo *multipath* obtiver um desempenho pelo menos tão bom quanto o desempenho obtido na melhor rota disponível se somente utilizar-se um único fluxo para transmitir os dados. O segundo objetivo é não utilizar-se em demasia a capacidade dos recursos partilhados pelas diferentes rotas à semelhança do que acontece caso existisse um único fluxo nesse meio. Por fim, o terceiro objetivo é garantir um balanceamento de carga através da deslocação do máximo tráfego possível das rotas mais congestionadas para as rotas menos congestionadas.

O algoritmo lida com a função que inclui o aumento do número de subfluxos e utiliza um comportamento não modificado do TCP em caso de perda. Além disso, o algoritmo depende dos mecanismos tradicionais do TCP para detetar perdas, retransmitir os dados, etc. Um dos problemas mais importante é a quantidade de largura de banda que um utilizador *multipath* deve utilizar no total. Para estimar esta quantidade, o fluxo *multipath* deve saber as taxas de perdas e o *Round-Trip Time* (RTT) em cada fluxo TCP e depois seleccionar a taxa ideal. Resumidamente, o que o algoritmo faz é expandir o atual algoritmo para o controlo de congestão do TCP para funcionar com diversas rotas.

Muito sucintamente, o que o algoritmo faz é aumentar a fase do *congestion avoidance state* através da especificação de como o tamanho da janela aumenta depois de receber um ACK [10]. As propriedades *slow start*, *fast retransmission*, *fast recovery algorithms* e também o decréscimo multiplicativo do *congestion avoidance state* mantêm-se igual ao utilizado no TCP. Em suma, o algoritmo para implementações baseadas em pacotes é calculado utilizando-se a seguinte formula 2.2 para cada ACK recebido nos subfluxos i , aumentando $cwnd_i$ por:

$$\min\left(\frac{\alpha * bytes_acked * MSS_i}{cwnd_total}, \frac{bytes_acked * MSS_i}{cwnd_i}\right) \quad (2.2)$$

O $cwnd_i$ é o tamanho da janela de congestão no subfluxo i , o $cwnd_total$ é o sumatório do tamanho da janela de congestão de todos os subfluxos da conexão, MSS_i é o tamanho máximo do segmento de todos os subfluxos i e o parâmetro α é a agressividade do fluxo *multipath*. Este último parâmetro é selecionado de forma a que o débito agregado de um fluxo *multipath* seja igual ao débito que um fluxo TCP pode atingir na melhor rota.

2.5.1 Algoritmo OLIA

O *Opportunistic Linked Increases Algorithm* (OLIA) [11] é um novo algoritmo para o controlo de congestão que visa corrigir segundo os autores, dois problemas apresentados pelo o algoritmo utilizado no MPTCP e também pela excessiva quantidade de tráfego transmitido nas rotas congestionadas. Eles afirmam que a transição de alguns utilizadores TCP para o MPTCP faz com que o

débito dos restantes utilizadores diminua sem qualquer benefício para os novos utilizadores MPTCP (*non-Pareto optimality*) e que os utilizadores podem ser excessivamente agressivos para os utilizadores TCP. Graças a estes dois problemas, os três princípios que o modo de operação do MPTCP deve respeitar [10] não são cumpridos como seriam de esperar.

O OLIA é um mecanismo para o controlo de congestão baseado em janela similar ao LIA que aco- pla os crescentes aumentos e utiliza o comportamento não modificado do TCP em caso de perdas. Similarmente ao LIA, este algoritmo apenas se aplica à parte do aumento do *congestion avoidance state*. As restantes características do TCP tais como o *slow start*, *fast retransmission*, *fast recovery algorithms*, bem como o decréscimo multiplicativo do *congestion avoidance state* mantêm-se igual.

Com base nos resultados obtidos do modelo de testes por eles desenvolvido, os autores afirmam que este algoritmo proporciona uma alocação *Pareto-optimal* que satisfaz os três princípios do modo de operação do MPTCP e no caso de um determinado utilizador possuir o mesmo RTT em todas as rotas disponíveis, o OLIA maximiza o critério de equitatividade como o comum TCP faz.

A partir do momento em que o objetivo desta tese de mestrado é demonstrar o bom desempenho do MPTCP face ao TCP nos cenários elaborados em aspetos como o débito e balanceamento de carga e visto que este algoritmo introduz melhorias significativas ao funcionamento do MPTCP em comparação com o algoritmo LIA, o seu uso na fase de testes é justificado. Entretanto, aconselha-se a leitura mais detalhada do artigo [11], visto que a explicação ao promenor do funcionamento deste novo algoritmo não faz parte dos objetivos propostos na tese de mestrado e também por se tratar de um algoritmo matematicamente complexo.

2.6 Considerações sobre a interface aplicacional

O MPTCP foi projetado para ser compatível com as aplicações, mas o transporte dos dados difere comparativamente com o TCP [12], o que traz uma certa liberdade para as aplicações explorar. Contudo, o MPTCP introduz um impacto positivo e negativo nas aplicações em diferentes áreas comparado com o uso regular do TCP.

Em primeiro lugar, causa um impacto no desempenho que merece ser mencionado porque a carga é distribuída por todos os subfluxos e, deste modo, a conexão não deve ter pior desempenho comparativamente com o TCP na melhor rota. Mas, no ponto de vista das aplicações [13], o impacto no desempenho é diferente em muitos aspetos. Aspetos como um melhor débito, o que faz com que a largura de banda disponibilizada às aplicações seja maior, podendo até exceder a capacidade máxima de uma das interfaces de rede. A remoção e a adição dos subfluxos provoca uma grande variação na largura de banda da conexão e, portanto, as aplicações, como por exemplo, o FTP que depende da largura de banda disponível no momento para transmitir determinados dados num curto espaço de tempo pode sofrer atrasos na entrega caso a largura de banda diminua drasticamente. Outra característica é o atraso porque os valores do atraso variam consoante as rotas da mesma conexão, por exemplo, o *jitter* para uma aplicação pode parecer alto conforme os dados são enviados nos subfluxos. A utilização de diversos subfluxos de dados traz também problemas às aplicações

mais dependentes de valores de RTT mais estáveis e, por isso, este tipo de aplicações têm de ter em conta os diferentes valores de RTT que cada subfluxo têm no momento. No que diz respeito à melhoria introduzida na resiliência, se por algum motivo um dos subfluxos falhar, todo o tráfego vai ser distribuído ao longo dos restantes subfluxos e, por isso, qualquer pacote perdido pode ser retransmitido nesses subfluxos.

Em segundo lugar, um potencial problema é o impacto causado nas *middleboxes*, apesar do MPTCP ter sido desenvolvido para ultrapassar a maioria das *middleboxes*. No entanto, algumas das *middleboxes* podem não permitir a passagem das mensagens MPTCP devido à presença das opções TCP. Se isto acontecer, o modo de operação do MPTCP deve retroceder para o TCP, adicionando com isso um grande atraso à medida que o primeiro *handshake* falhe. Existem outras *middleboxes* que de uma maneira transparente podem alterar o tamanho do conteúdo de tal forma que o MPTCP só tem de detetar a sua presença com antecedência para depois regressar ao modo de operação TCP. Outro potencial problema é que as aplicações implicitamente assumem um mapeamento um-para-um na *socket* da interface e, portanto, a utilização de vários subfluxos fazem com que esta assunção não seja a mais correta neste tipo de abordagem. Por fim, como se abordou no decorrer desta dissertação, têm-se de ter em atenção com o impacto que isto traz à segurança, pois o MPTCP traz novas vulnerabilidades às quais os agressores tentam se aproveitar para adulterar as conexões.

Em relação ao modo de operação do MPTCP perante as aplicações, este pode afetar todas as interfaces que implicitamente assumem o estabelecimento de uma comunicação TCP entre um único par de endereços e portas TCP, numa *socket* terminal, na interface de rede, ou para uma determinada rota através da rede. Assim, existem duas classes de aplicações, as aplicações *legacy* e as aplicações *MPTCP-aware*. As aplicações *legacy* são aquelas que não estão cientes da API do MPTCP e utilizam a existente API do TCP. Por outro lado, as aplicações *MPTCP-aware* são as que suportam uma interface MPTCP. Não faz parte desta dissertação, explicar todos os efeitos que o MPTCP traz a estas duas classes de aplicações e, por isso, aconselha-se a leitura do artigo [13] para uma melhor compreensão do tópico aqui abordado.

2.7 Considerações sobre a segurança

Novas classes de ameaças [8] vêm com a adição do suporte *multipath* ao protocolo de transporte TCP. Ameaças como os ataques *flooding*, ataques *hijacking* e ataques do tipo *Denial of Services* (DoS) são um dos exemplos entre muitos outros. Portanto é importante abordar-se sobre os três requerimentos chave [6] para o suporte de uma segurança pelo menos igual à qual atualmente o TCP assegura. Em primeiro lugar, é necessário um mecanismo para confirmar que ambos os dispositivos num *handshake* de um subfluxo são os mesmos da configuração original da conexão. Em segundo lugar, é necessário verificar se os novos endereços das máquinas podem receber tráfego antes de serem utilizados como parte da conexão. Por último, é fundamental ter proteção contra ataques do tipo *replay*, isto é, assegurar que um determinado pedido para adicionar ou remover um subfluxo é recente.

Para assegurar que estes requerimentos são cumpridos, o MPTCP utiliza um algoritmo denominado por *hash-based handshake* que, muito sucintamente, funciona com base na troca de chaves a serem enviadas única e exclusivamente no primeiro subfluxo. O estabelecimento de futuros subfluxos é feito através do envio prévio de chaves *hash* e ao qual a conexão identifica-as como "*tokens*". É graças a estas chaves que se podem criar *Hash-based Message Authentication Codes* (HMAC) porque é este mecanismo que simultaneamente verifica tanto a integridade como a autenticidade de uma mensagem. Para proteger contra ataques *replay*, os *handshakes* utilizam números aleatórios de um único uso (*nonces*) permitindo que o HMAC seja o mesmo em dois *handshakes*.

Para terminar, no que diz respeito à segurança, deve ser referenciado que em futuras versões do MPTCP, devem ser estabelecidos outros mecanismos de segurança de maneira a torna-lo mais forte e seguro. Atualmente o melhor a fazer-se é ganhar experiência com os existentes mecanismos para verificar que as ameaças identificadas não afetam o funcionamento do MPTCP.

2.8 Preocupações com as *Middleboxes*

No decorrer da dissertação, falou-se que o funcionamento das *middleboxes* em relação ao MPTCP traz um grande número de problemas. Nesta secção irá-se discutir como o MPTCP lida com estes problemas [6] e, como previamente estabelecido, se ocorrer alguma falha no MPTCP, este deve retornar para uma comunicação TCP. A principal preocupação é como algumas *middleboxes* não deixam passar pacotes contendo novas opções TCP. A solução é introduzida no MPTCP pela utilização da opção "*kind*" onde todos os tipos de opções vão ser definidos como valores "*subtype*", aumentando assim as hipóteses para todos os tipos de opções passarem pelas *middleboxes* sem quaisquer tipo de problemas.

Com base nisto, analisa-se de uma forma breve o comportamento de algumas *middleboxes* perante o modo de operação do MPTCP. Os *Network Address (and Port) Translators* (NATs) modificam os endereços de destino de um pacote, o que significa que uma máquina não vai saber qual o endereço que deve sinalizar ao MPTCP. Outra *middlebox* são os *Performance Enhancing Proxies* (PEPs) que pró-ativamente fazem o ACK aos dados para aumentar o desempenho e graças a isso não vai permitir ao MPTCP trocar os sinais de controlo de congestão entre as máquinas. As *Firewalls* são outras das *middleboxes* que aleatoriamente designam um número inicial de sequência às conexões TCP, o que irá provocar conflitos com os números de sequência que o MPTCP introduz em cada um dos subfluxos. Como no caso dos NATs, as *firewalls* não permitem o estabelecimento de muitas conexões. Existem muitos outros tipos de *middleboxes* tais como *Traffic normalizers*, os *Intrusion Detection Systems* (IDS), etc, onde é importante analisar o seu modo de operação perante o MPTCP, mas os quais não serão abordados no âmbito deste trabalho.

Para completar a análise às preocupações referidas no primeiro parágrafo, ainda existe o problema da segmentação/fusão que pode copiar as opções entre os pacotes e também o problema do tamanho da janela de receção que pode aumentar devido a algumas *middleboxes* ao nível dos subfluxos.

3

Descrição dos Ambientes Simulados

Conteúdo

3.1 Cenários de teste elaborados	16
3.2 Instalação das máquinas	17
3.3 Configuração do endereçamento e rotas	19
3.4 Metodologia	20
3.5 Wireshark	21

Em conformidade com os objetivos estabelecidos para este trabalho, em primeiro lugar é necessário explicar de que forma se irá realizar todo o procedimento de testes ao MPTCP, passando pela demonstração da arquitetura dos cenários de teste elaborados até ao *Hardware* e *Software* que cada uma das máquinas em teste possui. Sendo assim, nas seguintes secções encontram-se explicados todos os detalhes técnicos de maneira a assegurar que os dados recolhidos possam demonstrar a fiabilidade e a eficiência do MPTCP no que diz respeito à obtenção de um bom débito e balanceamento de carga nos cenários de teste propostos.

3.1 Cenários de teste elaborados

As duas arquiteturas de rede que serviram de base à realização dos testes de forma a obter-se os resultados desejados e em linha com os objetivos propostos nesta tese de mestrado são os seguintes.

3.1.1 Cenário Ethernet - Ethernet

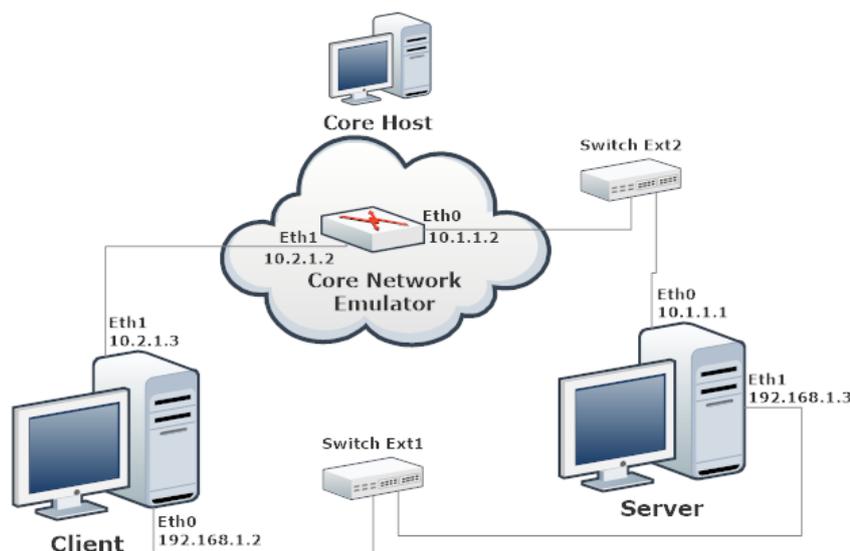


Figura 3.1: Cenário Ethernet - Ethernet.

O primeiro cenário diz respeito a um cenário *Ethernet - Ethernet* entre duas máquinas, a máquina cliente e a máquina servidor. Sendo que ambas possuem duas interfaces de rede, eth0 e eth1, onde se estabelecem duas redes de gamas diferentes.

A primeira, entre a interface eth0 da máquina cliente e a interface eth1 da máquina servidor, ou seja a rede 192.168.1.0/24. A segunda, entre a interface eth1 da máquina cliente e a interface eth0 da máquina servidor. No entanto entre esta segunda rede, existe uma terceira máquina a correr o *software* de emulação CORE [14] de forma a garantir uma maior diversidade de caminhos possíveis entre as máquinas cliente e servidor. Com isto, a segunda rede estabelecida entre a máquina cliente e a máquina servidor não utiliza somente uma gama de endereços, mas sim as gamas 10.1.1.0/24 e

10.2.1.0/24.

É de referir que também foi necessário a introdução de dois *switchs* físicos entre ambas as máquinas porque os Sistemas Operativos Linux instalados em cada uma não permitem a interligação via cabos RJ45 de duas ou mais redes entre duas ou mais máquinas.

3.1.2 Cenário WiFi - Ethernet

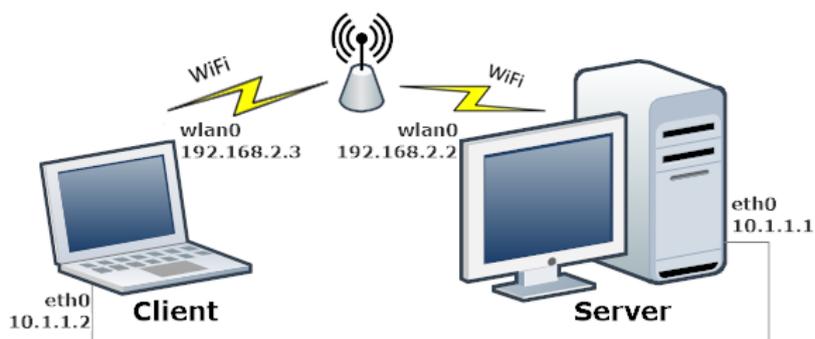


Figura 3.2: Cenário WiFi - Ethernet.

A segunda arquitetura representa um cenário *WiFi - Ethernet* entre novamente a máquina servidor e uma nova máquina cliente, um portátil. A escolha desta nova máquina cliente deve-se à falta de uma interface de rede sem fios na anterior máquina e por isso recorreu-se ao uso do portátil pessoal. Neste cenário, ambas as máquinas encontram-se interligadas fisicamente por um cabo RJ45 e ligadas numa comunicação sem fios através do estabelecimento de uma rede ad-hoc criada exclusivamente na máquina servidor. Como se trata de duas tecnologias de acesso à rede que utilizam meios físicos distintos para difundir os dados, os Sistemas Operativos Linux em cada uma das máquinas não criam qualquer tipo de limitação e, sendo assim, não existe a necessidade de instalar um *router/switch* entre as duas máquinas.

3.2 Instalação das máquinas

Com ambos os cenários definidos, passou-se à montagem e configuração de cada uma das máquinas em cima referidas, tanto ao nível do *hardware* como ao nível do *software*.

3.2.1 Hardware

No que diz respeito às máquinas disponíveis, estas são constituídas por três *desktops* com as seguintes especificações técnicas:

- Máquina servidor - utiliza um processador Intel Pentium Dual Core com uma velocidade de relógio de 2.93 GHz, 2 Gb de memória RAM DDR2, possui uma interface de rede Fast Ethernet

eth1 (capacidade até 100 Mbits/s) e uma interface de rede Gigabit Ethernet eth0 (capacidade até 1000 Mbits/s), esta máquina também possui uma interface de rede sem fios wlan0 que utiliza a norma 802.11g.

- Máquina cliente - utiliza um processador Intel Pentium4 com uma velocidade de relógio de 2.00 GHz, 512 Mb de memória RAM DDR2, possui duas interfaces de rede Fast Ethernet eth0 e eth1 (capacidade até 100 Mbits/s).
- Máquina CORE - utiliza um processador AMD Sempron 3000+ com uma velocidade de relógio de 1.86 GHz, 512 Mb de memória RAM DDR, possui duas interfaces de rede Fast Ethernet eth0 e eth1 (capacidade até 100 Mbits/s).

Em relação à máquina portátil, como se trata de uma máquina mais recente e por isso com *hardware* mais recente, ela possui um processador Intel Core i7 com uma velocidade de relógio de 1.60 GHz, uma memória RAM de 4 Gb DDR3, uma interface de rede Gigabit Ethernet eth0 (capacidade até 1000 Mbits/s) e uma interface de rede sem fios wlan0.

Por fim, os *switchs* externos são dispositivos que permitem às ligações atingirem no máximo débitos de 100 Mbits/s.

3.2.2 Software

Relativamente ao *software*, mais especificamente aos sistemas operativos instalados em cada uma das máquinas, eles foram selecionados com base em limitações técnicas impostas pelos investigadores da Catholique University de Louvain na Bélgica [15] e também por limitações do *hardware* utilizado. Com isto, o sistema operativo instalado em ambos *desktops* foi o Debian Wheezy com uma arquitetura de 32-bit e com um ambiente gráfico LXDE devido ao processamento gráfico de ambas as placas gráficas serem limitados. Já no portátil, instalou-se o sistema operativo Ubuntu 13.04 com uma arquitetura de 64-bit por se tratar de uma máquina com melhores recursos de hardware.

Com os sistemas operativos corretamente instalados e configurados em cada uma das máquinas, passou-se ao *download* e instalação dos pacotes necessários que permitissem o uso do protocolo MPTCP [15] em cada um dos seus Kernels através do uso dos pacotes disponíveis nos repositórios específicos de cada distribuição utilizada. No entanto, para encontrar os pacotes nos repositórios foi necessário o *download* e a instalação de uma chave gpg-atp em cada uma das máquinas para que posteriormente fosse possível fazer o *download* dos pacotes MPTCP.

Em segundo lugar, foi necessário adicionar um novo ficheiro com o nome mptcp.list no diretório /apt/sources.list.d/ de cada máquina e dependendo do sistema operativo de cada máquina respetiva, adicionou-se a seguinte linha no ficheiro criado:

- Ubuntu Raring (13.04)
`deb http://multipath-tcp.org/repos/apt/debian raring main`
- Debian Wheezy (7.0)
`deb http://multipath-tcp.org/repos/apt/debian wheezy main`

De seguida, executa-se dois comandos na *Shell* do Linux de forma a atualizar-se os repositórios de *software* do Linux e instalar-se os pacotes base do MPTCP em cada um dos sistemas.

sudo apt-get update

sudo apt-get install linux-mptcp

Depois da instalação estar concluída, as máquinas são reiniciadas e no respetivo GRUB seleciona-se a nova distribuição Kernel.

3.3 Configuração do endereçamento e rotas

O endereçamento das interfaces de rede e conseqüentemente o estabelecimento das rotas nas tabelas de encaminhamento é um dos passos mais importantes para o correto funcionamento de todo o cenário de teste. As seguintes tabelas representam os endereços escolhidos para cada uma das interfaces de rede das máquinas utilizadas. De salientar que o endereçamento presente nas seguintes tabelas dizem respeito às Figuras 3.1 e 3.2, sendo que a configuração de cada uma das interfaces foram feitas com base no cenário de testes em uso no momento.

Tabela 3.1: Tabela de endereçamento da máquina servidor

Interface	Address	Network Mask	Network	Gateway
Eth0	10.1.1.1	/24	10.1.1.0	10.1.1.2
Eth1	192.168.1.3	/24	192.168.1.0	192.168.1.2
Wlan0	192.168.2.2	/24	192.168.2.0	192.168.2.3

Tabela 3.2: Tabela de endereçamento do desktop cliente

Interface	Address	Network Mask	Network	Gateway
Eth0	192.168.1.2	/24	192.168.1.0	192.168.1.3
Eth1	10.2.1.3	/24	10.2.1.0	10.2.1.2

Tabela 3.3: Tabela de endereçamento da portátil cliente

Interface	Address	Network Mask	Network	Gateway
Eth0	10.1.1.2	/24	10.1.1.0	10.1.1.1
Wlan0	192.168.2.3	/24	192.168.2.0	192.168.2.2

No que diz respeito à máquina onde se instalou o emulador CORE, as configurações necessárias para o correto funcionamento de toda a arquitetura passaram sobretudo pela configuração das rotas estáticas no router emulado de forma a que todo o tráfego com origem na máquina cliente (10.2.1.3) fosse encaminhando para a máquina servidor (10.1.1.1) e vice-versa. Para isto acontecer, em primeiro lugar, atribui-se os endereços IP e as respetivas máscaras às duas interfaces do *router*, ou seja, a interface eth0 fica com o endereço/máscara 10.1.1.2/24 e a interface eth1 fica com o endereço/máscara 10.2.1.2/24. Em segundo lugar, no *router* emulado introduz-se as duas seguintes rotas estáticas:

- ip route add 10.1.1.2/24 eth0 10.1.1.1
- ip route add 10.2.1.2/24 eth1 10.2.1.3

Por fim, na shell do *router* emulado ativa-se o encaminhamento entre as interfaces de rede através da execução do comando `echo1 > \proc\sys\net\ipv4\ip_forwarding`.

Tabela 3.4: Tabela de endereçamento da máquina CORE

Interface	Address	Network Mask	Network	Gateway
Eth0	10.1.1.2	/24	10.1.1.0	10.1.1.1
Eth1	10.2.1.2	/24	10.2.1.0	10.2.1.3

3.4 Metodologia

Com os cenários de teste estabelecidos e com a instalação/configuração das máquinas terminada, era tempo de definir qual a metodologia de testes a seguir de forma a recolher dados que permitisse demonstrar as potencialidades do MPTCP. Sendo assim, o primeiro passo seguido foi definir qual o protocolo aplicacional que gerasse uma maior quantidade de tráfego TCP entre a(s) máquina(s) cliente(s) e a máquina servidor de forma a congestionar uma das ligações e com isso analisar o comportamento do MPTCP, uma vez que este utiliza os múltiplos caminhos disponíveis para transmitir os dados. Com isto, o protocolo ao qual o seu uso mais se justifica é o FTP (*File Transfer Protocol*) por se tratar de um protocolo com o qual é possível transmitir ficheiros de tamanhos consideráveis. O segundo passo foi instalar e configurar o servidor VsFTPD na máquina servidor e nas máquinas cliente instalar e configurar o cliente FTP, Filezilla.

O tamanho do ficheiro a transmitir em ambos os cenários foi de 794 Mb porque só com ficheiros de tamanhos consideráveis é que era possível calcular o débito entregue à aplicação em cada um dos subfluxos estabelecidos e o respetivo débito agregado. Contudo, sabe-se que o débito teórico é calculado pela divisão de um ficheiro com elevado tamanho pelo tempo que este demora a ser transmitido em cada uma das ligações.

Com base nos dois cenários de teste elaborados e de forma a não ocorrer um "estrangulamento" da rede, criou-se única e exclusivamente dois subfluxos de dados entre ambas as máquinas. Foi necessário adotar-se este tipo de abordagem inicial para que a máquina servidor não trabalhasse ao mesmo tempo como um *router*, pois ela própria é a máquina destino dos pacotes enviados pela máquina cliente, evitando a perda imensurável de pacotes e conseqüente degradação do desempenho do MPTCP. E como foi explicado no terceiro capítulo e inclusive exemplificado através da Figura 2.2, se cada uma das duas interfaces de rede de uma respetiva máquina permitir o estabelecimento de uma comunicação MPTCP, isto significa que serão pelo menos criados mais três subfluxos, além do subfluxo da comunicação principal. Então para isto não ocorrer, por exemplo, ativou-se esta funcionalidade em uma das interfaces de ambas as máquinas através do seguinte comando:

- Na máquina cliente:

```
sudo ip link set dev eth0 multipath on
```

- Na máquina servidor:

```
sudo ip link set dev eth1 multipath on
```

No que diz respeito ao procedimento de como os testes vão ser realizados, achou-se fundamental, em primeiro lugar, realizar-se nestes mesmos cenários a transmissão do ficheiro com o FTP utilizando como protocolo de transporte o TCP. Desta forma, consegue-se obter uma base de resultados com os quais posteriormente se pode comparar com os resultados alcançados via MPTCP, e com isto demonstrar ou não as potencialidades do MPTCP perante o TCP nas mesmas condições. Sendo assim, em cada um dos cenários de teste, o primeiro teste realizado foi o da transmissão via TCP do ficheiro selecionado. E para ativar novamente o uso do protocolo TCP em cada um dos sistemas é necessário executar o seguinte comando:

```
sudo sysctl -w net.mptcp.mptcp_enabled=0
```

De seguida, os restantes testes a realizar-se passam sobretudo pelo envio do mesmo ficheiro, só que com o uso do MPTCP e do respetivo algoritmo para o controlo de congestão, neste caso os algoritmos COUPLED e OLIA. O modo de operação de ambos já foi descrito no capítulo 2 e por isso, só é necessário salientar que o algoritmo COUPLED utilizado pelo MPTCP não é o mesmo algoritmo COUPLED utilizado pelo TCP porque o verdadeiro nome do algoritmo utilizado pelo MPTCP é LIA, mas como se trata do *coupling* do LIA, o nome optado foi COUPLED.

Por último, é importante salientar que apesar dos testes serem exatamente os mesmos em ambos os cenários, isto não significa que os resultados sejam exatamente os mesmos porque tratam-se de dois cenários onde uma das tecnologias de acesso à rede é sobre um meio físico sem fios. E como se sabe, a tecnologia WiFi em comparação com a tecnologia Ethernet possui diferentes características ao nível do débito máximo atingível, bem como os atrasos fim-a-fim serem claramente superiores.

3.5 Wireshark

Com todas as configurações realizadas, era tempo de instalar uma ferramenta que permitisse armazenar e analisar de uma forma precisa os resultados obtidos com a simulação nos cenários de teste propostos. A escolha do Wireshark em detrimento de muitas outras ferramentas deveu-se à sua versatilidade e bons conhecimentos do seu funcionamento adquiridos ao longo dos anos, mas sobretudo por tratar-se de uma ferramenta que permite analisar ao detalhe os fluxos de dados recolhidos, bem como as peculiaridades introduzidas pelo MPTCP nos pacotes TCP. Destacam-se as características introduzidas no processo *three-way-handshake*, quais os tipos de dados transportados no campo *Kind* e o tipo de mensagens trocadas entre ambas as máquinas. Além disto, esta ferramenta permite analisar graficamente e estatisticamente a largura de banda, o número de pacotes transmitidos e perdidos, e o RTT (*Round Trip Time*) entre o estabelecimento inicial da conexão TCP e os respetivos subfluxos. Resumidamente, trata-se de uma ferramenta bastante completa e por isso o seu uso é justificado.

4

Discussão dos Resultados

Conteúdo

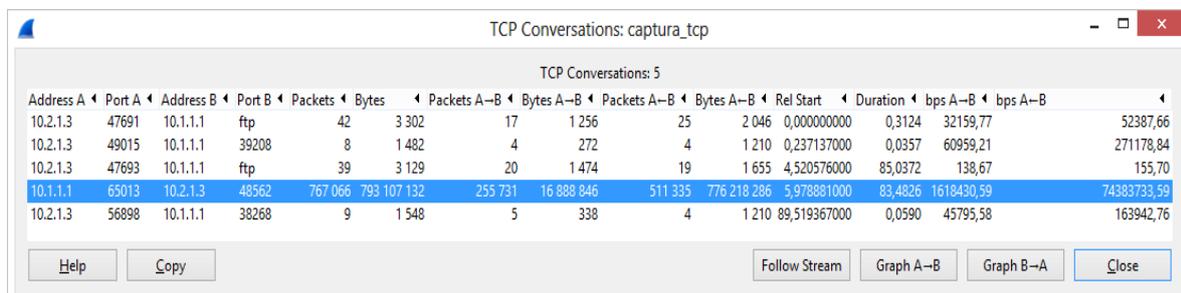
4.1 Cenário Ethernet - Ethernet	24
4.2 Cenário WiFi - Ethernet	35

4.1 Cenário Ethernet - Ethernet

Para analisar o modo de operação do protocolo MPTCP de maneira a testar se este conduz a um bom débito e bom balanceamento de carga em comparação com o TCP, foi necessário recolher dados em ambos os casos para mais tarde poder-se comparar os resultados alcançados. Sendo assim, como se referiu anteriormente, a ferramenta utilizada para recolher os dados foi o Wireshark. Com esta ferramenta, analisou-se subfluxo por subfluxo o tempo que demorou a transmitir o ficheiro selecionado, a percentagem de carga e a largura de banda alcançada em cada um dos subfluxos e na sua totalidade, o número total de pacotes transmitidos e retransmitidos (número de pacotes perdidos), e, por fim, o RTT que demorou para estabelecer cada subfluxo.

4.1.1 TCP

Utilizando-se o TCP significa que será aberta uma única sessão TCP/IP e que apenas existirá um único fluxo de dados entre a máquina cliente e a máquina servidor, particularmente na interface de rede onde a comunicação FTP através do Filezilla se estabeleceu. Neste caso, a comunicação FTP foi estabelecida da máquina cliente para a máquina servidor através da conexão à interface de rede eth0 (10.1.1.1). Através da seguinte figura, consegue-se verificar o tempo que demorou a transmitir o ficheiro, o número de pacotes/bytes transmitidos e a largura de banda alcançada nesta ligação TCP específica.



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B
10.2.1.3	47691	10.1.1.1	ftp	42	3 302	17	1 256	25	2 046	0,000000000	0,3124	32159,77	52387,66
10.2.1.3	49015	10.1.1.1	39208	8	1 482	4	272	4	1 210	0,237137000	0,0357	60959,21	271178,84
10.2.1.3	47693	10.1.1.1	ftp	39	3 129	20	1 474	19	1 655	4,520576000	85,0372	138,67	155,70
10.1.1.1	65013	10.2.1.3	48562	767 066	793 107 132	255 731	16 888 846	511 335	776 218 286	5,978881000	83,4826	1618430,59	74383733,59
10.2.1.3	56898	10.1.1.1	38268	9	1 548	5	338	4	1 210	89,519367000	0,0590	45795,58	163942,76

Figura 4.1: Tabela de Conversações TCP do cenário Ethernet - Ethernet.

Apartir da análise a esta figura, observa-se que o tempo que demorou para transmitir todos os 511 335 pacotes foi de aproximadamente 84 segundos e com isto, a largura de banda média atingida foi de aproximadamente 74 Mbits/s. Graficamente, o débito total alcançado e o número de pacotes transmitidos em toda a conexão podem ser vistos nas seguintes figuras.

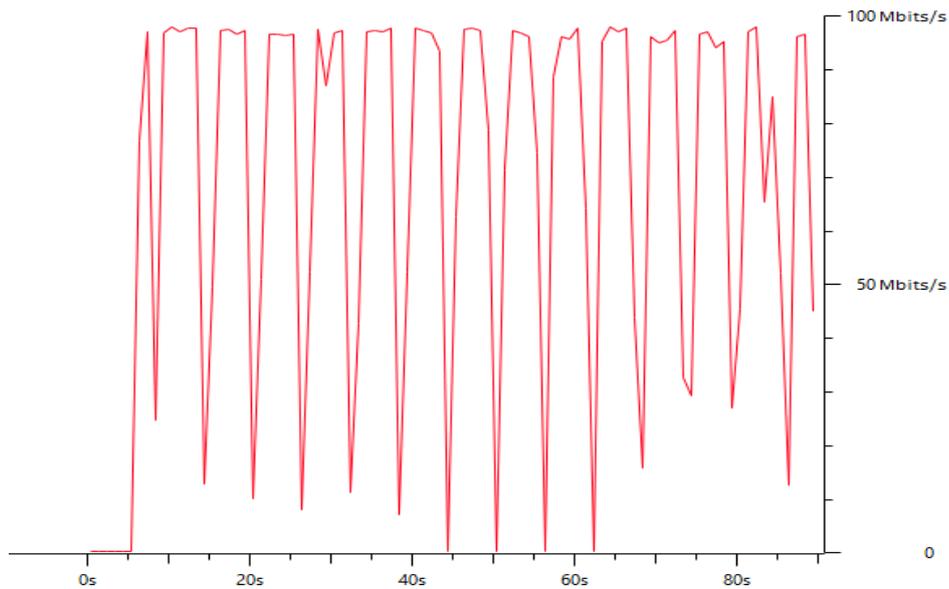


Figura 4.2: Débito total da conexão TCP do cenário Ethernet - Ethernet.

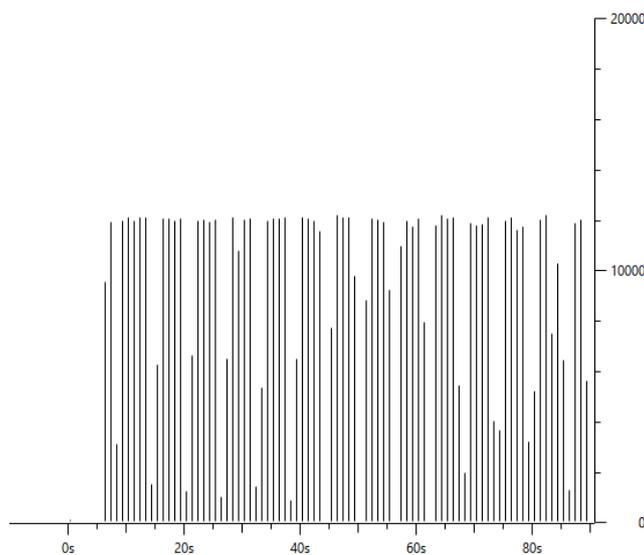


Figura 4.3: Número de pacotes transmitidos no cenário Ethernet - Ethernet.

Outra funcionalidade do Wireshark é permitir a introdução de filtros e com isso poder-se determinar o número de pacotes perdidos. O filtro *tcp.analysis.lost_segment* do tipo *"previous segment not captured"* é uma mensagem que o Wireshark emite quando descobre uma falha nos números de sequência de uma determinada conversação. Isto normalmente indica a ocorrência de uma perda de pacote (isto se o pacote for posteriormente retransmitido) ou a chegada de um pacote fora de ordem (não é considerado como perda) ou a perda de um pacote ACK relacionado com o segmento perdido que previamente não tinha sido capturado pelo recetor (também não é considerado como perda), isto significa que nem sempre se trata de um problema crítico.

Pela aplicação deste filtro, obteve-se 785 pacotes e através do seguinte filtro *tcp.analysis.retransmission* obteve-se 8 pacotes retransmitidos, isto significa que todos os 785 pacotes capturados não correspondem a pacotes perdidos. Sendo assim, também aplicou-se o seguinte filtro *tcp.analysis.out_of_order*

e o resultado conseguido foi de 188 pacotes, o que significa que os restantes 584 pacotes são resultado de perdas relacionadas com as mensagens ACK. No entanto, para poder-se confirmar este último número, aplicou-se o seguinte filtro `tcp.analysis.ack_lost_segment` e o resultado alcançado demonstra que de facto este corresponde ao número de ACKs perdidos.

No Anexo A, pode-se visualizar todos os gráficos resultantes da aplicação destes filtros, excepto o gráfico correspondente ao número de pacotes perdidos porque este encontra-se apresentado através da seguinte figura.

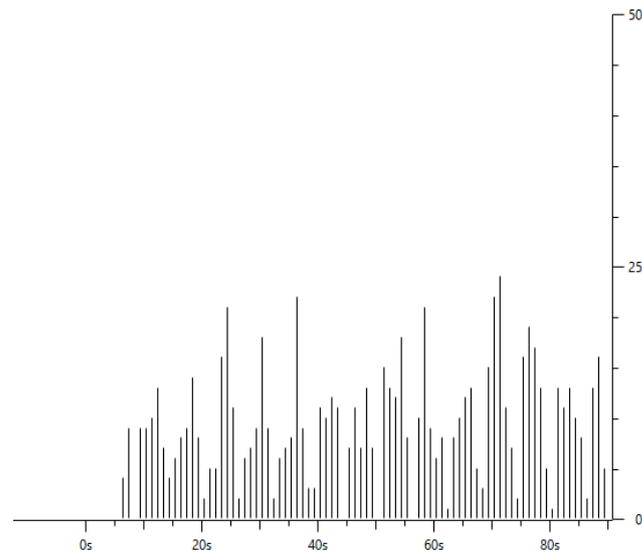


Figura 4.4: Número total de pacotes perdidos no cenário Ethernet - Ethernet.

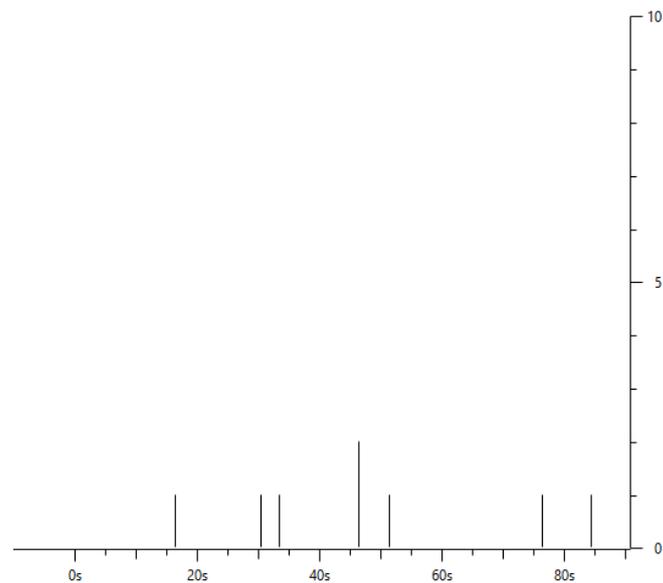


Figura 4.5: Número de pacotes retransmitidos no cenário Ethernet - Ethernet.

Por fim, mas não menos importante, o último parâmetro analisado foi o RTT estabelecido durante o primeiro *three-way-handshake*, por exemplo, entre o envio por parte do cliente do primeiro pacote SYN para o servidor e consequentemente recepção do pacote ACK. Como foi mencionado previa-

mente neste documento, existe uma discrepância entre os tempos obtidos pelo TCP e os tempos obtidos pelo MPTCP, por isso é importante analisar este parâmetro. Sendo assim, o RTT atingido no estabelecimento desta sessão TCP foi de 0,24 milissegundos.

4.1.2 MPTCP com COUPLED

Nesta secção encontram-se representados os resultados alcançados com o uso do protocolo MPTCP, juntamente com o algoritmo para o controlo de congestão COUPLED. Com base no cenário de teste elaborado e de forma a obter-se resultados demonstrativos do potencial do MPTCP, optou-se pelo estabelecimento de dois subfluxos de dados entre a máquina cliente e a máquina servidor e, por isso, apenas se ativou a funcionalidade *multipath* em uma única interface de rede de cada máquina. O porquê do cumprimento desta condição encontra-se explicada em detalhe no Capítulo 3 desta dissertação.

Após o termino da comunicação, ao analisar-se a tabela de conversações verificou-se que, de facto, apenas se estabeleceu o subfluxo de dados da comunicação inicial entre a interface eth1 (10.2.1.3) da máquina cliente e a interface eth0 (10.1.1.1) da máquina servidor e o subfluxo de dados entre a interface eth0 (192.168.1.2) da máquina cliente e a interface eth1 (192.168.1.3) da máquina servidor.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B
10.2.1.3	47696	10.1.1.1	ftp	9	828	5	434	4	394	0,000000000	0,0750	46308,77	42040,68
192.168.1.2	60786	192.168.1.3	ftp	14	1 252	7	614	7	638	0,076729000	0,1478	33230,05	34528,95
10.2.1.3	47697	10.1.1.1	ftp	17	1 601	9	730	8	871	2,856881000	45,9268	127,16	151,72
192.168.1.2	40276	192.168.1.3	ftp	27	2 524	14	1 289	13	1 235	2,943179000	45,8272	225,02	215,59
10.2.1.3	48428	10.1.1.1	10178	344 653	359 075 014	232 136	350 748 688	112 517	8 326 326	4,024875000	44,2362	63431938,38	1505793,70
192.168.1.2	58229	192.168.1.3	10178	345 869	360 274 506	233 000	351 922 124	112 869	8 352 382	4,228914000	44,0320	63939288,57	1517510,06
10.1.1.1	63015	10.2.1.3	42421	11	1 848	6	1 426	5	422	48,781233000	0,0024	4815533,98	1425073,87

Figura 4.6: Tabela de Conversações TCP com o COUPLED no cenário Ethernet - Ethernet.

Da anterior figura, conclui-se que o tempo que demorou para transmitir todos os 465 136 pacotes foi de aproximadamente 44 segundos e com isto, a largura de banda média agregada atingida foi de aproximadamente 127 Mbits/s. Graficamente, o débito total alcançado juntamente com o débito individual de cada fluxo e o número de pacotes transmitidos em toda a conexão podem ser vistos nas seguintes figuras.

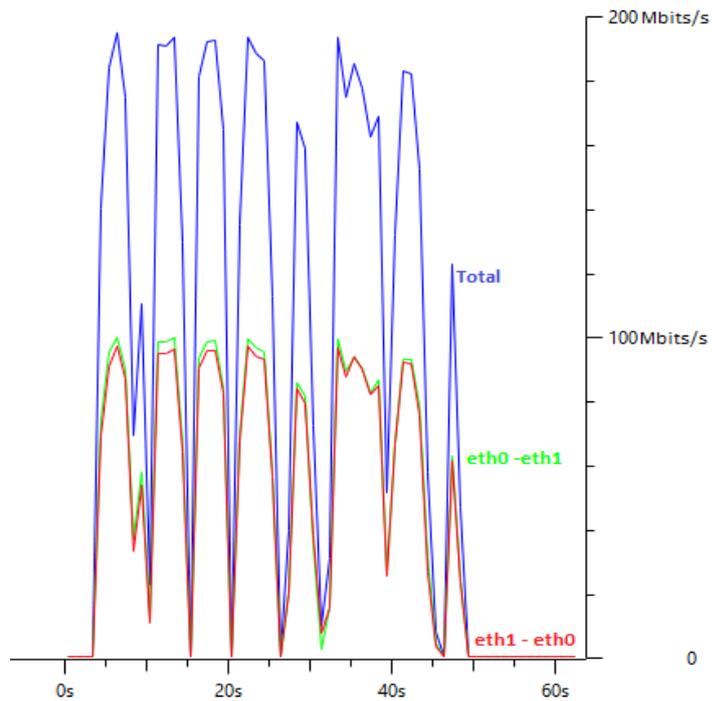


Figura 4.7: Débito total e individual da conexão com o COUPLED no cenário Ethernet - Ethernet.

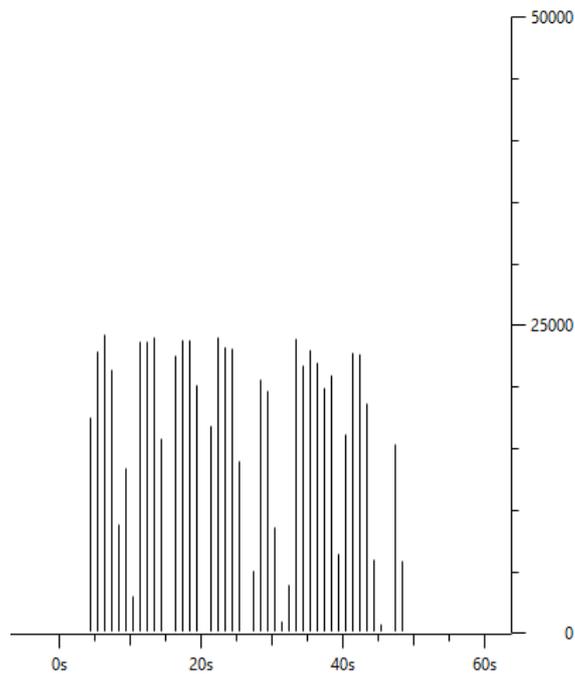


Figura 4.8: Número de pacotes transmitidos com o COUPLED no cenário Ethernet - Ethernet.

Dos 465 136 pacotes transmitidos no decorrer da comunicação, 232 136 pacotes foram enviados através do subfluxo (eth1 - eth0) e 233 300 pacotes foram enviados através do subfluxo (eth0 - eth1), ou seja, sensivelmente em cada um dos subfluxos passou 50% da totalidade da carga. Estes resultados demonstram que de facto o MPTCP faz a distribuição da totalidade da carga pelos subfluxos disponíveis de uma forma equitativa, respeitando assim uma das características que o distingue pe-

rante o TCP.

Para determinar-se o número de pacotes perdidos, desordenados, retransmitidos e o número de ACKs perdidos, recorreu-se ao mesmo procedimento utilizado para o caso do TCP, ou seja, através da aplicação dos vários filtros. Portanto, os resultados alcançados foram de 868 pacotes perdidos na totalidade, sendo que dos quais 7 correspondem a retransmissões e 1 corresponde ao único pacote desordenado. Relativamente ao número de ACKs perdidos, este foi de 928 pacotes. Mais uma vez a maioria destes resultados encontram-se apresentados no Anexo A, à exceção do número de pacotes perdidos na totalidade que está representado na seguinte figura.

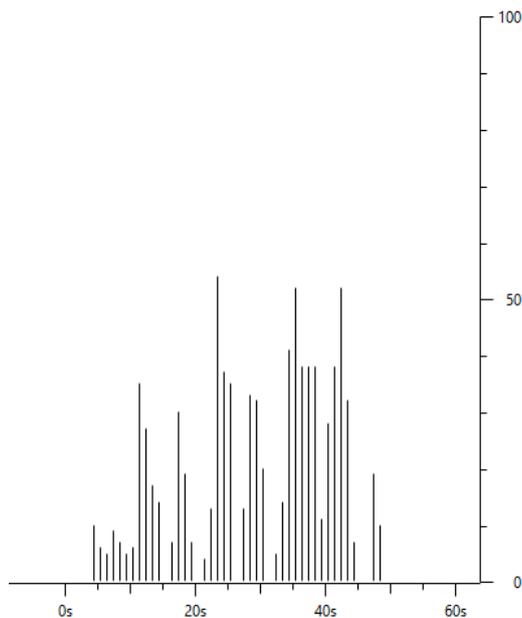


Figura 4.9: Número total de pacotes perdidos com o COUPLED no cenário Ethernet - Ethernet.

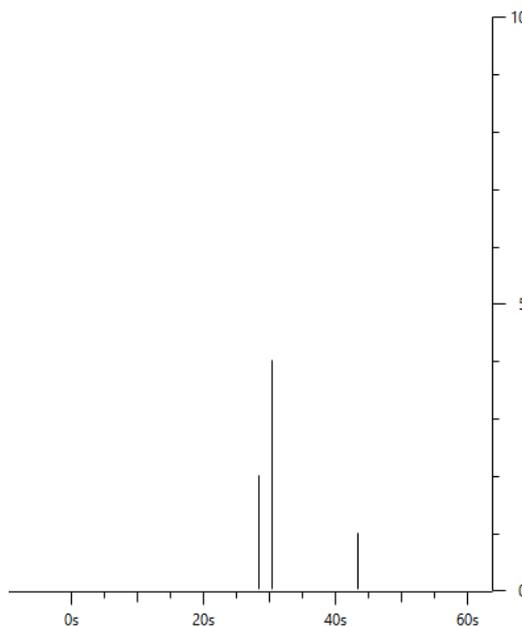


Figura 4.10: Número de pacotes retransmitidos com o COUPLED no cenário Ethernet - Ethernet.

Já em relação ao RTT que demorou para estabelecer cada subfluxo de dados, este foi analisado à semelhança do que aconteceu no caso do TCP. Sendo assim, os resultados obtidos foram de 0,32 milisegundos no subfluxo eth1 - eth0 e de 0,14 milisegundos no subfluxo eth0 - eth1. Estes resultados são normais pelo simples facto de que o caminho percorrido pelos pacotes no primeiro subfluxo possui mais um salto que o caminho percorrido pelo segundo subfluxo, pois os dados ainda tem que passar pela máquina CORE antes de chegar à máquina destino, o que introduz um atraso significativo ao valor final.

4.1.3 MPTCP com OLIA

À semelhança do ocorrido com o uso do algoritmo COUPLED, neste caso, o número de subfluxos criados é o mesmo e também possuem a mesma fonte e destinatário. O que irá distinguir entre o uso do algoritmo OLIA e o uso do algoritmo COUPLED serão os resultados alcançados em cada um dos parâmetros a analisar.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets B→A	Bytes B→A	Rel Start	Duration	bps A→B	bps B→A
10.2.1.3	47704	10.1.1.1	ftp	9	828	5	434	4	394	0,000000000	0,0740	46923,36	42598,62
192.168.1.2	43954	192.168.1.3	ftp	14	1252	7	614	7	638	0,074069000	0,1591	30867,07	32073,60
10.2.1.3	47705	10.1.1.1	ftp	17	1602	9	730	8	872	3,409538000	51,3182	113,80	135,94
192.168.1.2	35398	192.168.1.3	ftp	27	2525	14	1289	13	1236	3,492585000	51,1858	201,46	193,18
10.2.1.3	45310	10.1.1.1	37517	356243	368696034	238257	359963154	117986	8732880	5,071029000	49,5546	58111725,54	1409818,53
192.168.1.2	38853	192.168.1.3	37517	392699	408247150	263920	398717488	128779	9529662	5,283137000	49,3424	64644992,21	1545066,23
10.2.1.3	51364	10.1.1.1	35198	20	2574	10	852	10	1722	54,687702000	13,2273	515,30	1041,48

Figura 4.11: Tabela de Conversações TCP com o OLIA no cenário Ethernet - Ethernet.

Ao analisar-se a anterior tabela, verifica-se que o tempo que demorou para transmitir todos os 502 177 pacotes foi de aproximadamente 50 segundos e com isto, a largura de banda média atingida foi de aproximadamente 123 Mbits/s. Graficamente, o débito total alcançado juntamente com o débito individual de cada fluxo e o número de pacotes transmitidos em toda a conexão podem ser vistos nas seguintes figuras.

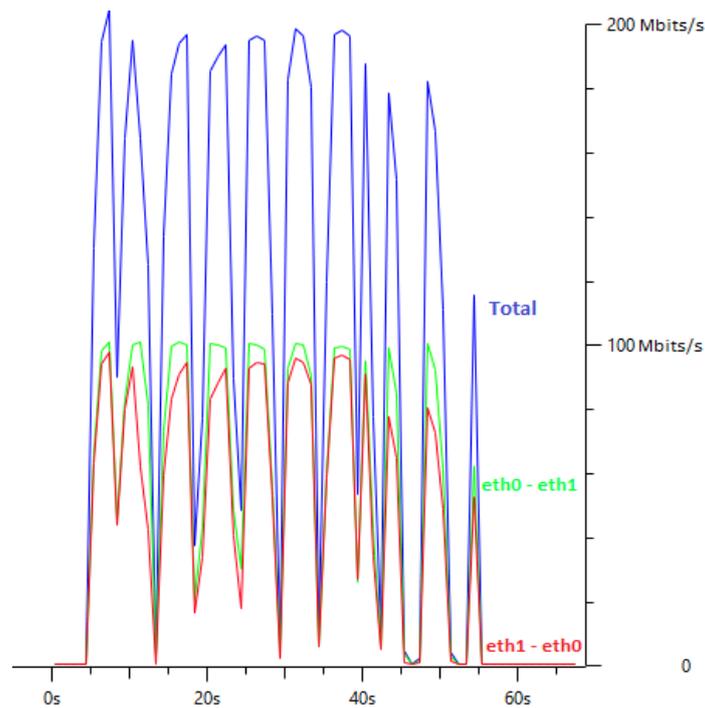


Figura 4.12: Débito total e individual da conexão com OLIA no cenário Ethernet - Ethernet.

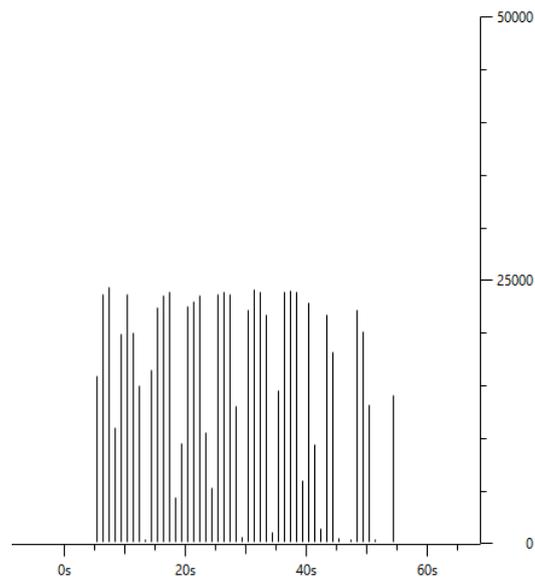


Figura 4.13: Número de pacotes transmitidos com OLIA no cenário Ethernet - Ethernet.

A ocorrência do balanceamento de carga é analisada através da quantidade de pacotes que foram transmitidos em cada um dos fluxos e, portanto, das duas anteriores tabelas retira-se que dos 502 177 pacotes transmitidos na totalidade, 238 257 pacotes foram enviados através do fluxo inicial e 263 920 pacotes foram enviados no subfluxo. Em termos de percentagem, no fluxo inicial passou sensivelmente 47% da carga total e no subfluxo os restantes 53%. Com estes resultados, pode-se afirmar que de facto existiu o balanceamento da carga, mas que em comparação com os resultados conseguidos com o algoritmo COUPLED, a carga não foi balanceada tão equitativamente.

Novamente utilizou-se os filtros para determinar o número de pacotes perdidos, desordenados,

retransmitidos e o número de ACKs perdidos. E, neste caso, o número de pacotes perdidos na totalidade foram de 604 pacotes, sendo que dos quais 2 correspondem a retransmissões e 2 correspondem a pacotes desordenados. Já o número de ACKs perdidos, este foi de 632 pacotes. A maioria destes resultados podem ser verificados no Anexo A, à exceção do número de pacotes perdidos e retransmitidos que se encontra representado na seguinte figura.

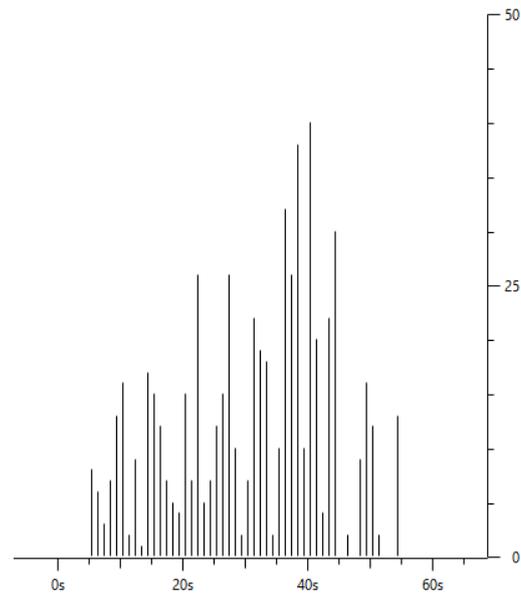


Figura 4.14: Número total de pacotes perdidos com OLIA no cenário Ethernet - Ethernet.

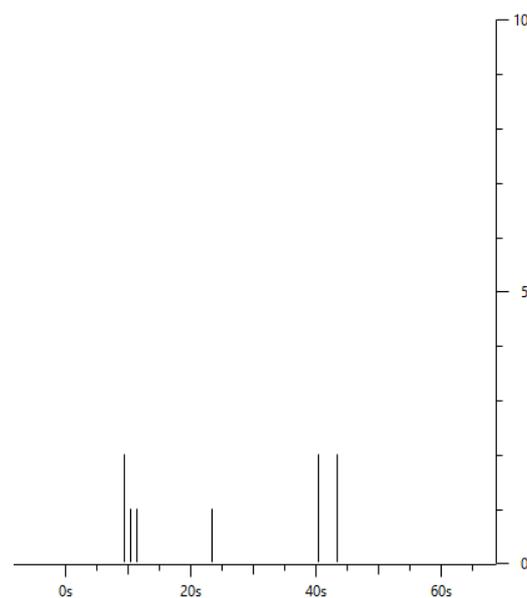


Figura 4.15: Número de pacotes retransmitidos com OLIA no cenário Ethernet - Ethernet.

Por último, o RTT do fluxo de dados foi de 0,3 milissegundos e o RTT do subfluxo foi de 0,2 milissegundos.

4.1.4 Comparação dos resultados

Nesta secção, comparam-se os resultados previamente obtidos de forma a verificar se de facto o MPTCP trouxe melhorias em aspetos como o balanceamento de carga, o débito total, o número de pacotes transmitidos, perdidos, etc, e o RTT alcançado no estabelecimento de cada subfluxo. Todos estes resultados são relacionados com a transferência de um ficheiro de tamanho único durante a comunicação FTP. Para uma melhor compreensão dos resultados conseguidos, as seguintes três tabelas mostram todos os valores recolhidos nos três casos apresentados.

Tabela 4.1: Tabela de Comparação do Débito no cenário Ethernet - Ethernet

Protocolo	Tempo de Transferência (s)	Débito (Mbps)		
		Total	MP 1	MP 2
TCP	84	≈ 74	N/A	N/A
MPTCP c/ COUPLED	44	≈ 127	≈ 63	≈ 64
MPTCP c/ OLIA	50	≈ 123	≈ 65	≈ 58

Através da análise à anterior tabela, observa-se que o débito alcançado com o MPTCP, independentemente do algoritmo para o controlo de congestão em uso, é de facto superior ao débito alcançado com o TCP. Inclusive, pode-se afirmar que o débito com o uso do MPTCP neste cenário quase que chega a ser o dobro do débito alcançado utilizando única e exclusivamente um caminho para transmitir os dados. Isto simplesmente demonstra que o MPTCP é uma solução eficiente para agregar-se a largura de banda disponível dos múltiplos caminhos. Como principal consequência do aumento do débito disponível, o tempo que demorou para transferir o mesmo ficheiro via FTP foi drasticamente reduzido, chegando quase a ser 50% inferior no caso do MPTCP com o algoritmo COUPLED.

Na seguinte figura encontra-se ilustrado um gráfico onde se compara os débitos conseguidos em cada um dos três casos analisados através da aplicação de um *smooth filter* nos gráficos dos débitos anteriormente apresentados.

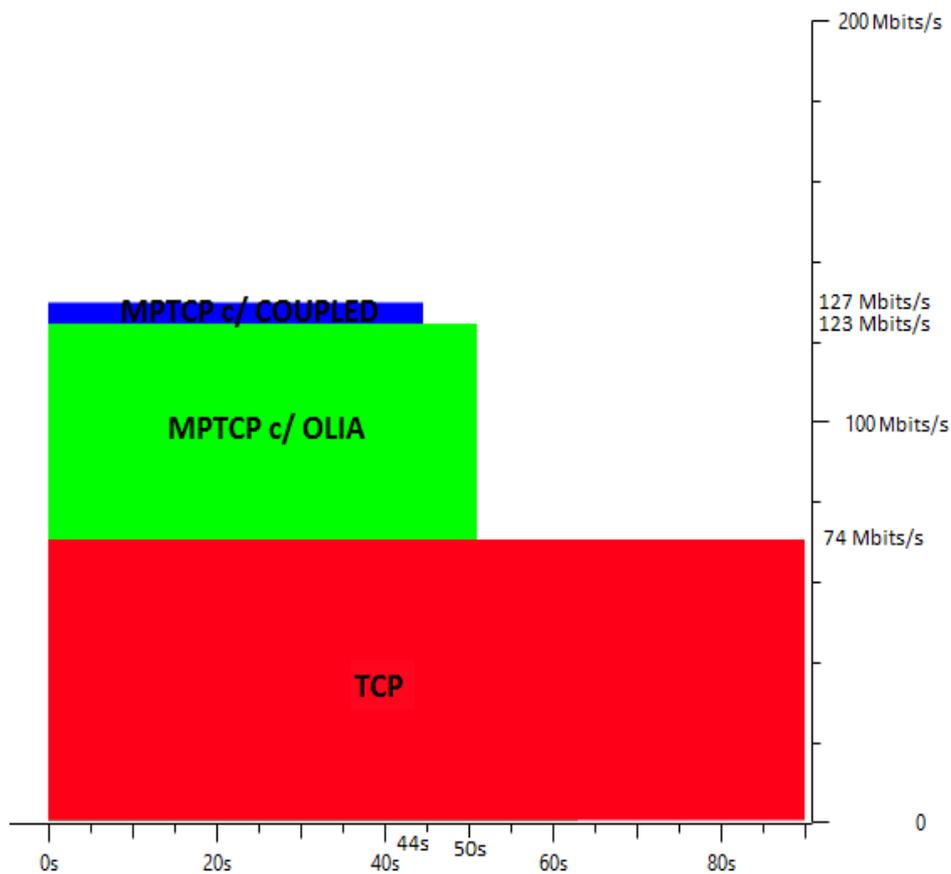


Figura 4.16: Comparação do débito no cenário Ethernet - Ethernet.

Tabela 4.2: Tabela de Comparação do número de Pacotes no cenário Ethernet - Ethernet

Protocolo	Pacotes					Balanceamento	
	Enviados	Perdidos	Retransmitidos	Desordenados	ACK's	Fluxo 1	Fluxo 2
TCP	511 335	785	8	188	584	N/A	N/A
MPTCP c/ COUPLED	465 136	868	7	1	928	232 136	233 300
MPTCP c/ OLIA	502 177	604	2	2	632	238 257	263 920

Analisando a tabela, verifica-se que o balanceamento de carga ocorre com o MPTCP, independentemente do algoritmo para o controlo de congestão em uso. Relativamente a um dos objetivos traçados para esta tese de mestrado, o facto do número de retransmissões ser um parâmetro que diferencia ou não o MPTCP do TCP. Analisando os resultados conseguidos em ambos os casos do MPTCP, o número de pacotes retransmitidos foi ligeiramente inferior apesar de este ser um número bastante reduzido nos três casos. Isto acontece pelo simples motivo de se tratar de cenários ideais, ou seja, cenários onde não existe qualquer tipo de congestão. No entanto, pode-se afirmar que de facto o número de retransmissões é um parâmetro no qual o MPTCP trouxe algumas melhorias.

Tabela 4.3: Tabela de Comparação dos RTT no cenário Ethernet - Ethernet

Protocolo	RTT (ms)	
	MP 1	MP 2
TCP	0,24	N/A
MPTCP c/ COUPLED	0,32	0,14
MPTCP c/ OLIA	0,3	0,2

Por último, o parâmetro RTT que já se esperava ser teoricamente mais alto com o MPTCP porque as mensagens trocadas no *three-way-handshake* são mensagens com um tamanho superior, pois estas levam mais informação em comparação com as mensagens trocadas no TCP. O mesmo já não acontece durante o estabelecimento do segundo subfluxo porque a informação trocada entre as duas máquinas diz respeito às *hashs* das chaves previamente trocadas durante o estabelecimento do primeiro subfluxo. Portanto, o RTT que demorou a estabelecer o segundo subfluxo é consideravelmente inferior ao RTT que demorou a estabelecer o primeiro subfluxo.

4.2 Cenário WiFi - Ethernet

O que distingue este cenário de teste do anterior é o facto de que os dados transmitidos via FTP irão ser enviados através de duas tecnologias de acesso à rede, com fios e sem fios. Utilizou-se novamente a ferramenta Wireshark para adquirir e analisar certos parâmetros tais como, a largura de banda utilizada por cada caminho e respetiva agregação, o número de pacotes transmitidos, retransmitidos, etc, e, por fim, o RTT de cada subfluxo.

4.2.1 TCP

De forma, a não existir uma grande discrepância entre os resultados aqui conseguidos e os resultados atingidos com o MPTCP, independentemente do algoritmo para o controlo de congestão em uso, optou-se por realizar a comunicação FTP entre as interfaces sem fios de ambas as máquinas. Isto porque cada uma das tecnologias de acesso à rede possui diferentes características, quer ao nível dos atrasos, quer ao nível do débito máximo que cada um consegue alcançar. Através da seguinte figura, consegue-se verificar o tempo que demorou a transmitir o ficheiro, o número de pacotes/bytes transmitidos e a largura de banda alcançada nesta ligação TCP específica.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B
192.168.2.3	49970	192.168.2.2	ftp	30	2 365	13	942	17	1 423	0,000000000	300,6688	25,06	37,86
192.168.2.3	60900	192.168.2.2	28965	8	1 482	4	272	4	1 210	0,132027000	0,0163	133562,48	594156,64
10.2.1.1	43362	10.1.1.1	ftp	6	628	4	385	2	243	2,531561000	0,0808	38137,69	24071,32
192.168.2.3	59413	192.168.2.2	50731	822 198	847 923 888	548 602	829 866 532	273 596	18 057 356	2,558236000	1011,0241	6566542,16	142883,69
192.168.2.3	49965	192.168.2.2	ftp	9	886	5	406	4	480	1013,815096000	0,2584	12569,76	14860,80
192.168.2.2	60198	192.168.2.3	36984	8	1 482	4	1 210	4	272	1014,039921000	0,0309	313522,27	70477,73

Figura 4.17: Tabela de Conversações TCP no cenário WiFi - Ethernet.

Analisando a anterior figura, observa-se que o tempo que demorou para transmitir todos os 548 602 pacotes foi de aproximadamente 1011 segundos (16 minutos e 51 segundos) e com isto, a largura de banda média atingida foi de aproximadamente 6.6 Mbits/s. Graficamente, o débito total alcançado e o número de pacotes transmitidos em toda a conexão estão representados nas seguintes figuras.

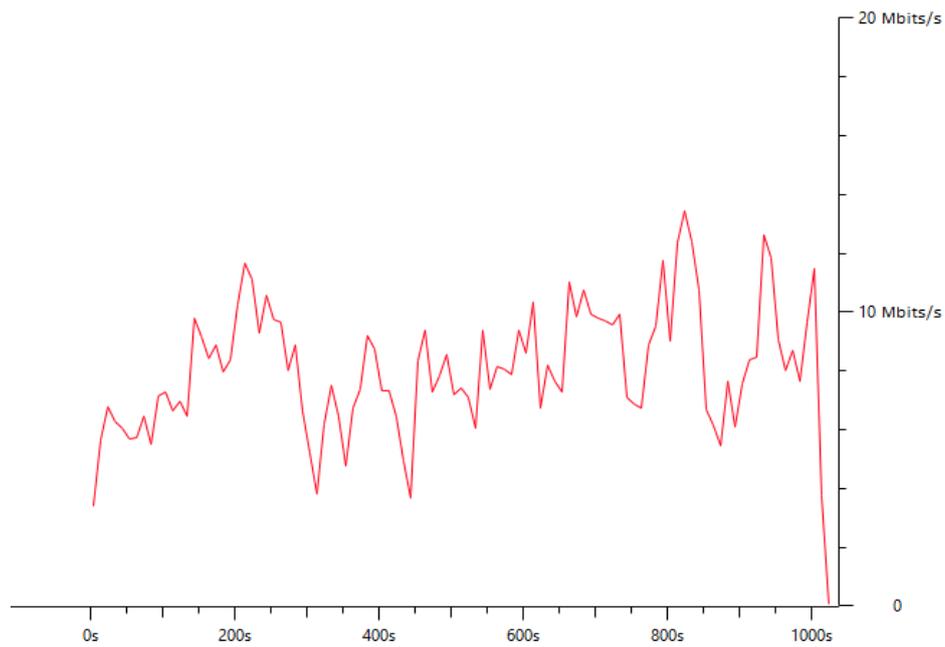


Figura 4.18: Débito total da conexão no cenário WiFi - Ethernet.

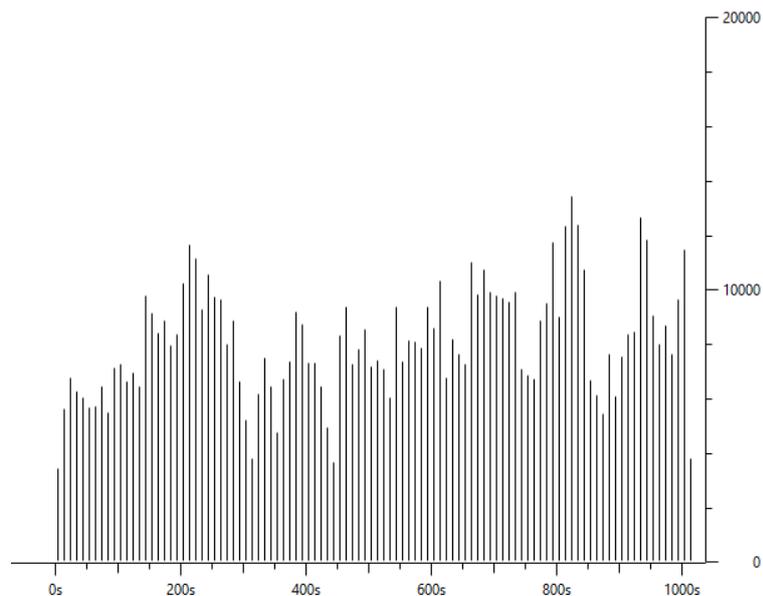


Figura 4.19: Número de pacotes transmitidos no cenário WiFi - Ethernet.

Neste cenário, não existiram quaisquer tipos de perdas, portanto, não existe qualquer tipo de gráfico a apresentar. Analisando estes resultados, pode-se afirmar que de certa forma não foram surpreendentes, apesar da comunicação realizar-se num meio sem fios, mas como as máquinas encontram-se praticamente paralelas uma a outra, não existe quaisquer tipos de obstáculos entre elas.

Como consequência da comunicação ser feita num canal sem fios e, portanto, num canal com menor débito e maiores atrasos, o RTT obtido da análise ao primeiro e único subfluxo estabelecido foi de 1,9 milissegundos. Mais uma vez, este resultado só demonstra que as tecnologias de acesso à

rede utilizadas são completamente heterogêneas.

4.2.2 MPTCP com COUPLED

Após o término da comunicação, ao analisar-se a tabela de conversações verificou-se que apenas se estabeleceu o fluxo de dados da comunicação inicial entre a interface wlan0 (192.168.2.3) da máquina cliente e a interface wlan0 (192.168.2.2) da máquina servidor e o subfluxo de dados entre a interface eth1 (10.2.1.1) da máquina cliente e a interface eth1 (10.1.1.1) da máquina servidor.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B									
192.168.2.3	49957	192.168.2.2	ftp	9	828	5	434	4	394	0,000000000	0,0518	66988,23	60814,20									
10.2.1.1	47860	10.1.1.1	ftp	14	1252	7	614	7	638	0,014193000	0,2134	23019,10	23918,87									
10.2.1.1	49361	192.168.2.2	ftp	19	1682	7	602	12	1080	0,016091000	63,1515	76,26	136,81									
10.2.1.1	56929	10.1.1.1	ftp	5	495	3	311	2	184	3,691498000	215,3273	11,55	6,84									
192.168.2.3	49952	192.168.2.2	ftp	4	392	2	148	2	244	3,691666000	0,2187	5412,72	8923,67									
192.168.2.2	65276	192.168.2.3	47450	120	293	125	523	354	39	114	2	2894	480	81	179	122	628	874	3,706831000	215,2869	107558,05	4556854,07
10.1.1.1	65276	10.2.1.1	41183	689	413	717	437	762	225	562	16	693	104	463	851	700	744	638	3,733741000	215,2567	620397,89	26043119,62
192.168.2.2	65276	10.2.1.1	52761	19	1682	12	1080	7	602	3,745628000	63,0991	136,93	76,32									
10.1.1.1	49132	10.2.1.1	46207	3	234	2	148	1	86	14,313154000	20,2018	58,61	N/A									
192.168.2.2	49132	192.168.2.3	38182	1	74	1	74	0	0	34,514961000	0,0000	N/A	N/A									

Figura 4.20: Tabela de Conversações TCP com o COUPLED no cenário WiFi - Ethernet.

Da anterior figura, conclui-se que o tempo que demorou para transmitir todos os 545 030 pacotes foi de aproximadamente 215 segundos e com isto, a largura de banda média atingida foi de aproximadamente 31 Mbits/s. Graficamente, o débito total alcançado juntamente com o débito individual de cada fluxo e o número de pacotes transmitidos em toda a conexão podem ser analisados nas seguintes figuras.

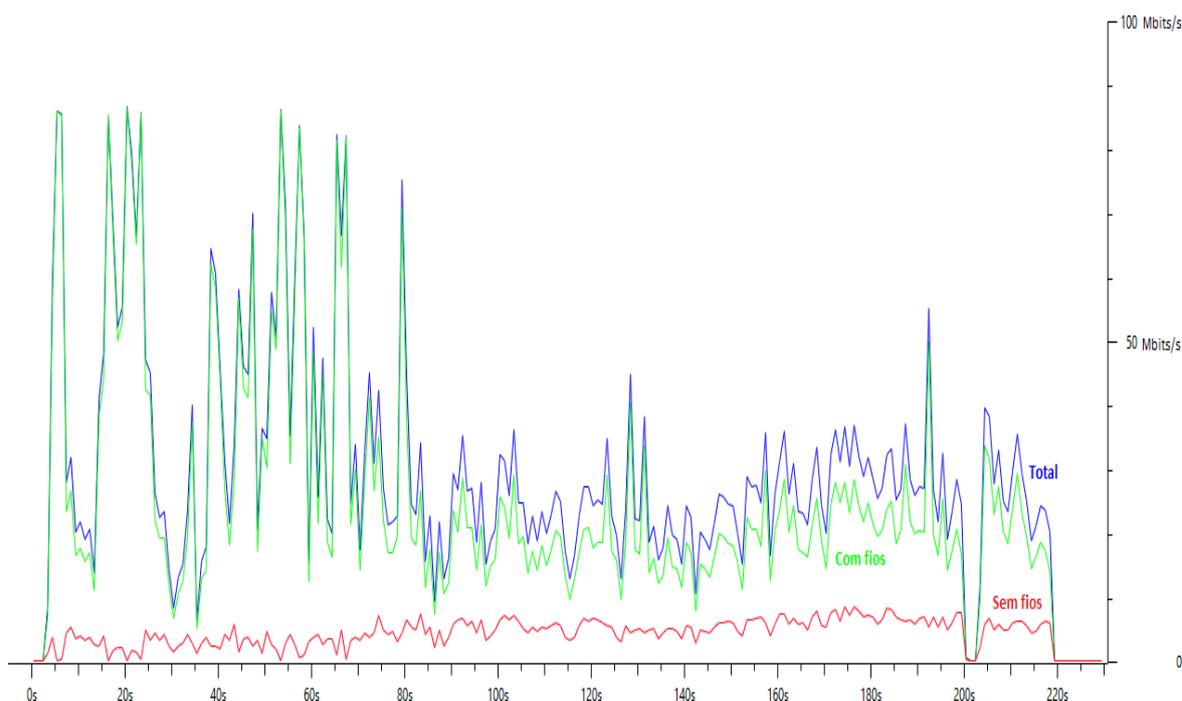


Figura 4.21: Débito total e individual da conexão com o COUPLED no cenário WiFi - Ethernet.

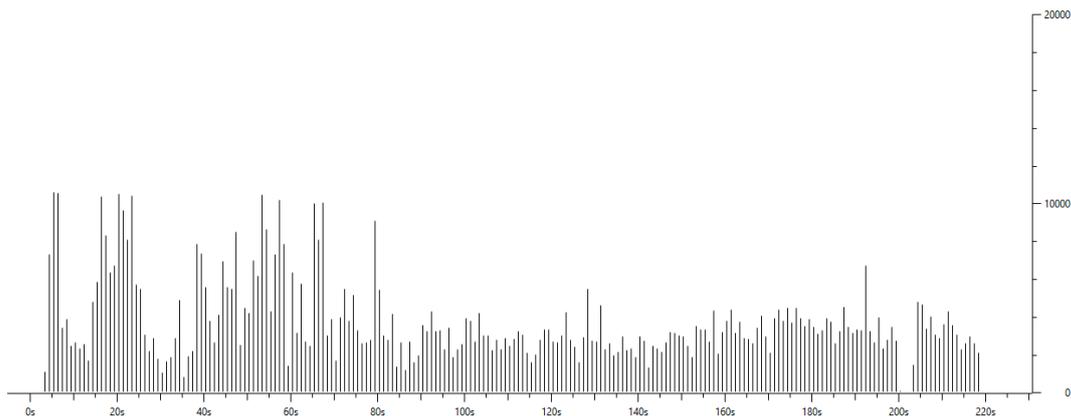


Figura 4.22: Número de pacotes transmitidos com o COUPLED no cenário WiFi - Ethernet.

Dos 545 030 pacotes transmitidos no decorrer da comunicação, 81 179 pacotes foram enviados através do subfluxo de dados sem fios e 463 851 pacotes foram enviados através do subfluxo de dados com fios, ou seja, 15% da carga passou pelo caminho sem fios e os restantes 85% passaram no caminho com fios. Apesar da maioria dos dados terem sido transmitidos no caminho com maior débito e menor número de perdas, o MPTCP faz um balanceamento de carga equitativo entre todos os caminhos disponíveis de tal forma que o débito alcançado no caminho Ethernet é inferior ao débito alcançado com o TCP no primeiro cenário. Isto só demonstra que a heterogeneidade das tecnologias de acesso à rede é um factor que de certa forma limita o desempenho do MPTCP neste tipo de cenários.

No que diz respeito, ao número de pacotes perdidos, desordenados, retransmitidos e o número de ACKs perdidos, recorreu-se ao mesmo procedimento utilizado no anterior cenário, ou seja, através da aplicação de vários filtros. E com esta aplicação, os resultados alcançados foram de 375 pacotes perdidos, sendo que dos quais 38 correspondem a retransmissões e 3 correspondem a pacotes desordenados. De salientar que dos 38 pacotes retransmitidos, só 2 pacotes é que são resultantes de perdas nos subfluxos estabelecidos porque os restantes 36 dizem respeito às tentativas de estabelecimento de comunicação entre os diferentes tipos de tecnologias, ou seja, entre as interfaces wlan0 e as interfaces eth0 e eth1. Sobre o número de ACKs perdidos, este foi de 388 pacotes. À exceção do número de pacotes perdidos e ao número de pacotes retransmitidos que se encontram ilustrados nas seguintes figuras, todos os restantes encontram-se no Anexo A deste documento.

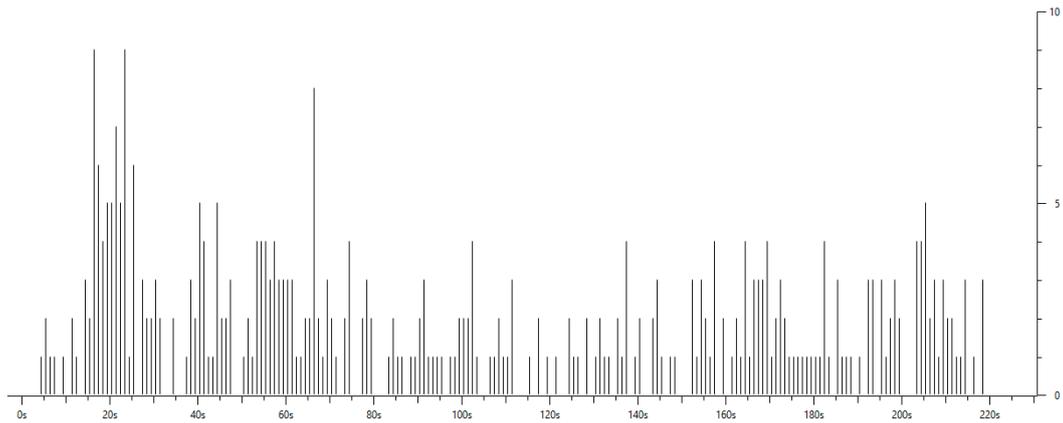


Figura 4.23: Número total de pacotes perdidos com o COUPLED no cenário WiFi - Ethernet.

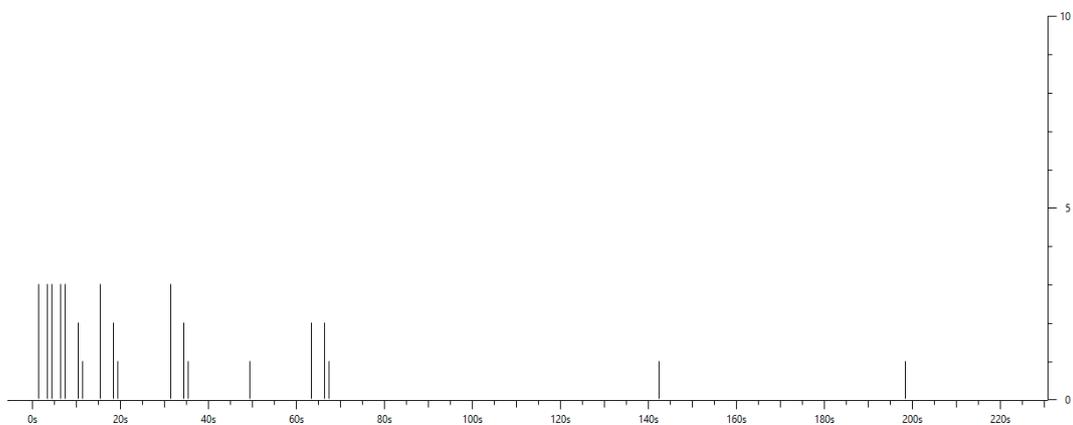


Figura 4.24: Número de pacotes retransmitidos com o COUPLED no cenário WiFi - Ethernet.

Por fim, o RTT que demorou para estabelecer o subfluxo de dados sem fios foi de 1 milissegundo, enquanto que o RTT para estabelecer o subfluxo de dados com fios foi de 0,22 milissegundos.

4.2.3 MPTCP com OLIA

Através da seguinte figura, verificam-se quais foram os subfluxos de dados estabelecidos com sucesso e aqueles que não passaram de simples tentativas. Em tudo semelhante ao COUPLED, a maioria do tráfego voltou a passar entre as interfaces wlan0 de cada máquina e entre a interface eth0 da máquina cliente e a interface eth1 da máquina servidor.

TCP Conversations: captura_OLIA_wlan0

TCP Conversations: 10													
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B
192.168.2.3	49959	192.168.2.2	ftp	9	828	5	434	4	394	0,000000000	0,0550	63074,52	57261,20
10.2.1.1	33896	10.1.1.1	ftp	21	2 015	10	892	11	1 123	0,016006000	0,1645	43367,55	54598,38
10.2.1.1	57715	192.168.2.2	ftp	19	1 682	7	602	12	1 080	0,016477000	63,2011	76,20	136,71
192.168.2.3	37119	192.168.2.2	8961	11	1 848	5	422	6	1 426	0,152342000	0,0292	115660,01	390832,16
192.168.2.3	49961	192.168.2.2	ftp	9	828	5	434	4	394	1,912466000	0,0617	56235,83	51052,80
10.2.1.1	49386	10.1.1.1	ftp	23	2 195	12	1 105	11	1 090	1,936816000	1,3258	6667,74	6577,23
10.2.1.1	32799	192.168.2.2	ftp	17	1 502	7	602	10	900	1,937093000	63,1745	76,23	113,97
192.168.2.2	64608	192.168.2.3	49475	69 090	71 443 884	22 899	1 694 618	46 191	69 749 266	3,151443000	103,1001	131493,02	5412158,63
10.1.1.1	64608	10.2.1.1	44223	343 081	353 908 150	113 053	8 367 246	230 028	347 540 904	3,185234000	103,0664	649464,44	26976075,31
192.168.2.2	64608	10.2.1.1	58448	17	1 506	11	990	6	516	3,195957000	63,1118	125,49	65,41

Help Copy Follow Stream Graph A-B Graph B-A Close

Figura 4.25: Tabela de Conversações TCP com o OLIA no cenário WiFi - Ethernet.

Da anterior figura, conclui-se que o tempo que demorou para transmitir todos os 276 219 pacotes foi de aproximadamente 103 segundos e com isto, a largura de banda média atingida foi de aproximadamente 32 Mbits/s. Graficamente, o débito total alcançado juntamente com o débito individual de cada subfluxo e o número de pacotes transmitidos em toda a conexão podem ser analisados nas seguintes figuras.

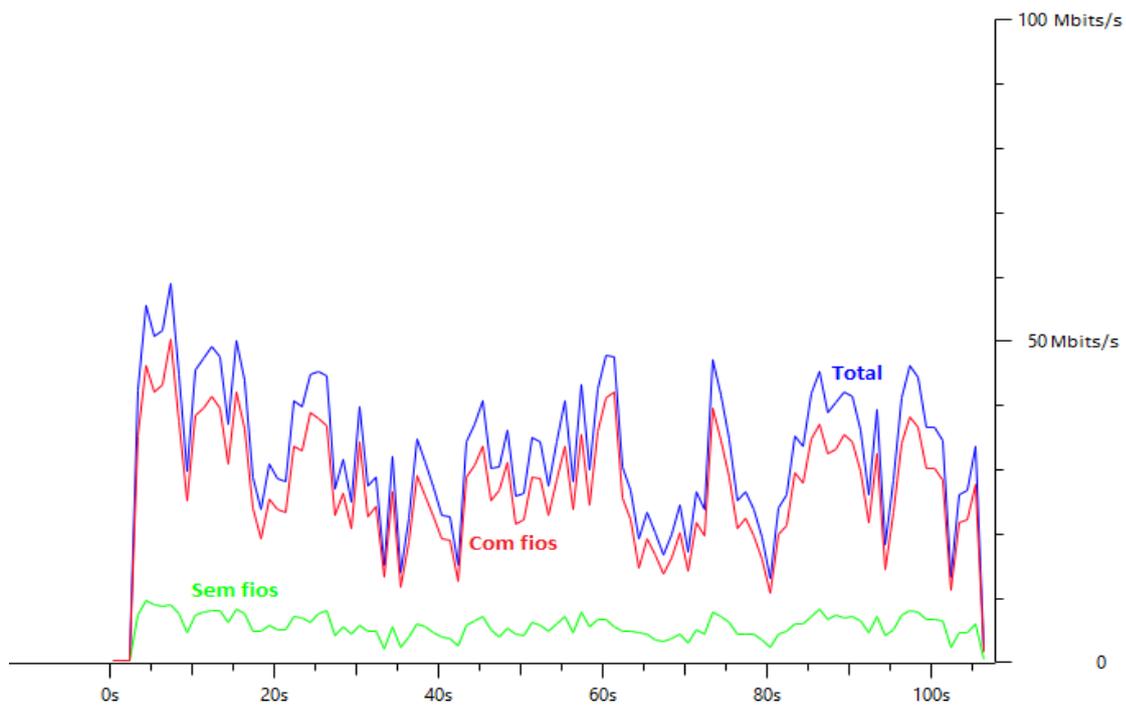


Figura 4.26: Débito total e individual da conexão com OLIA no cenário WiFi - Ethernet.

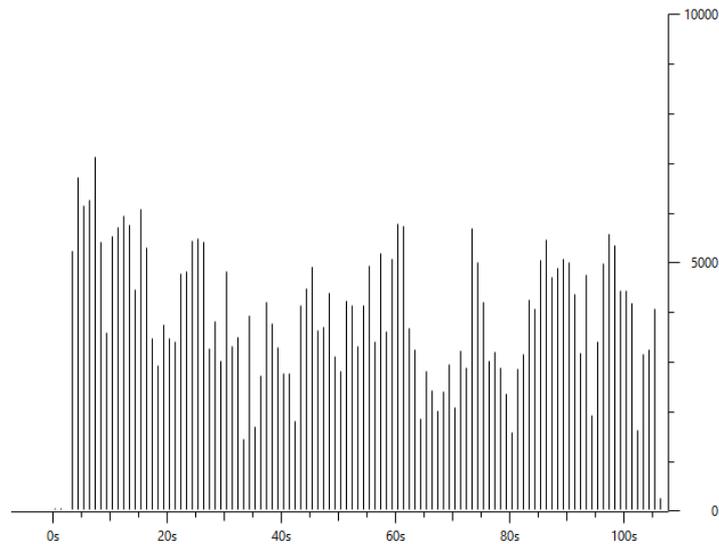
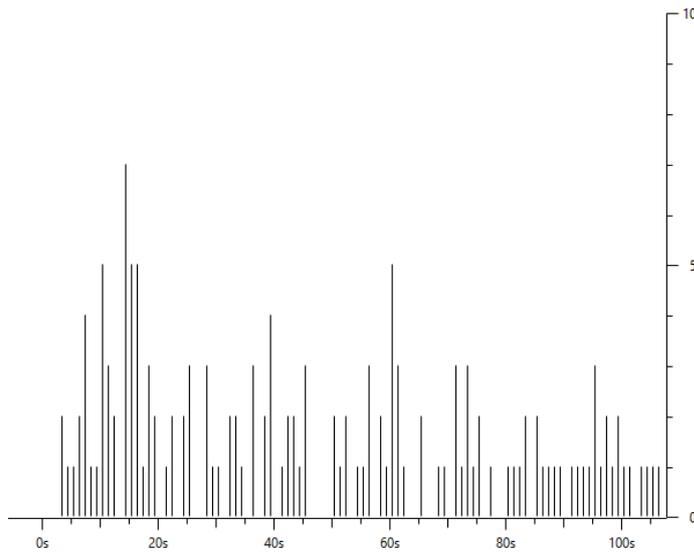


Figura 4.27: Número de pacotes transmitidos com OLIA no cenário WiFi - Ethernet.

Dos 276 219 pacotes transmitidos no decorrer da comunicação, 46 191 pacotes foram enviados através do subfluxo de dados sem fios e 230 028 pacotes foram enviados através do subfluxo de dados com fios, ou seja, 17% da carga passou pelo caminho sem fios e os restantes 83% passaram no caminho com fios. Os restantes 2,2% dizem respeito aos pacotes das tentativas no estabelecimento dos subfluxos de dados entre as interfaces sem fios com as interfaces com fios, mais os pacotes perdidos, retransmitidos, etc.

Relativamente a estes últimos pacotes, com a aplicação dos filtros obteve-se 148 pacotes perdidos, dos quais 54 correspondem a retransmissões e 3 correspondem a pacotes desordenados. Mais uma vez, o real número de pacotes retransmitidos são 6 pacotes porque os restantes 48 pacotes dizem respeito às tentativas do estabelecimento de conexão entre as interfaces de diferentes características. O número de ACKs perdidos no decorrer da comunicação foi de 152 pacotes. No anexo A encontram-se as figuras representativas destes resultados à exceção do número de pacotes perdidos e retransmitidos que se encontram ilustrados nas seguintes figuras.



]Número total de pacotes perdidos com OLIA no cenário WiFi - Ethernet.

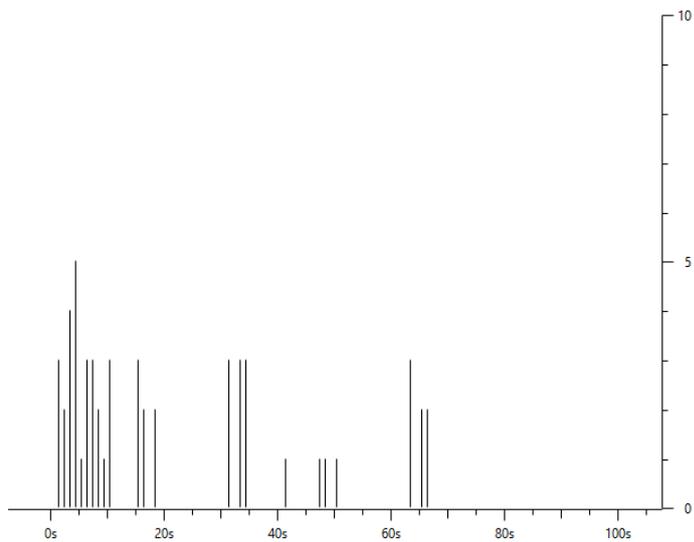


Figura 4.29: Número de pacotes retransmitidos com OLIA no cenário WiFi - Ethernet.

Por fim, o RTT que demorou para estabelecer o subfluxo de dados WiFi foi de 0,89 milisegundos, enquanto que o RTT para estabelecer o subfluxo de dados Ethernet foi de 0,33 milisegundos.

4.2.4 Comparação dos três casos

Nesta secção, vai-se comparar os resultados previamente obtidos de forma a verificar se de facto o MPTCP trouxe melhorias em aspetos como o balanceamento de carga, o débito total, o número de pacotes transmitidos, perdidos, etc, e o RTT alcançado no estabelecimento de cada subfluxo. Todos estes resultados são relacionados com a transferência de um ficheiro de tamanho único durante a comunicação FTP. Para uma melhor compreensão dos resultados conseguidos, as seguintes três tabelas mostram todos os valores recolhidos nos três casos de teste apresentados.

Tabela 4.4: Tabela de Comparação do Débito no cenário WiFi - Ethernet

Protocolo	Tempo de Transferência (s)	Débito (Mbps)		
		Total	MP 1	MP 2
TCP	1011	≈ 7	N/A	N/A
MPTCP c/ COUPLED	215	≈ 31	≈ 5	≈ 26
MPTCP c/ OLIA	103	≈ 32	≈ 5	≈ 27

Através da análise à anterior tabela, observa-se que o débito alcançado com o MPTCP, independentemente do algoritmo para o controlo de congestão em uso, é de facto superior ao débito alcançado com o TCP. Inclusive, pode-se afirmar que o débito com o uso do MPTCP neste cenário chega a ser quatro vezes superior ao débito alcançado utilizando única e exclusivamente um caminho para transmitir os dados. Isto simplesmente demonstra que de facto o MPTCP é uma solução eficiente para agregar-se a largura de banda disponível dos múltiplos caminhos. Como principal consequência do aumento do débito disponível, o tempo que demorou para transferir o mesmo ficheiro via FTP foi drasticamente reduzido, chegando a ser 90% inferior no caso do MPTCP com o algoritmo OLIA.

Na seguinte figura encontra-se ilustrado um gráfico onde se compara os débitos conseguidos em cada um dos três casos analisados através da aplicação de um *smooth filter* nos gráficos dos débitos anteriormente apresentados.

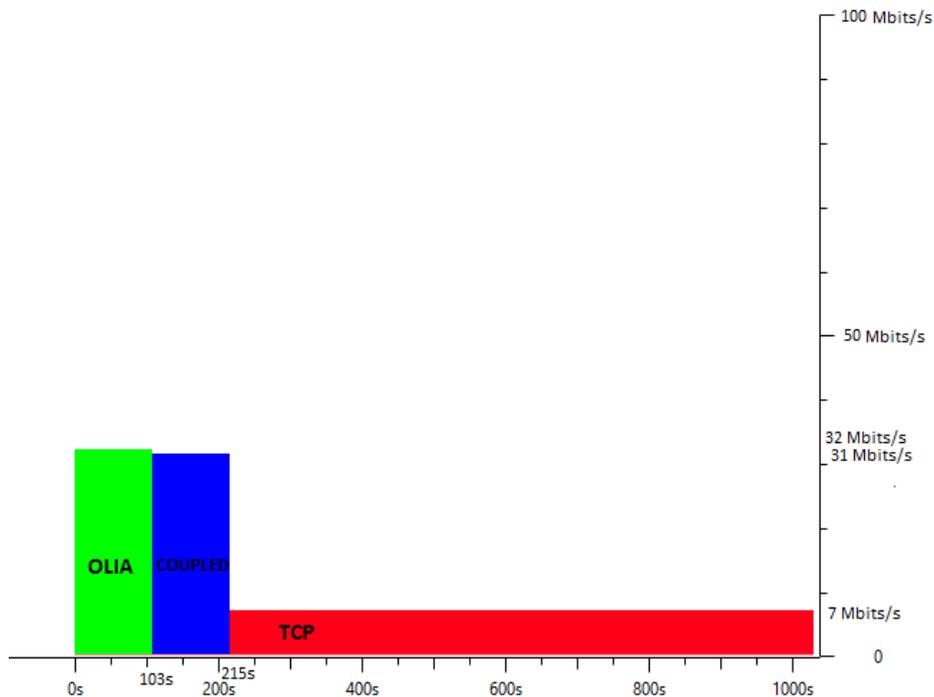


Figura 4.30: Comparação do débito no cenário WiFi - Ethernet.

Tabela 4.5: Tabela de Comparação do número de Pacotes no cenário WiFi - Ethernet

Protocolo	Pacotes					Balanceamento	
	Enviados	Perdidos	Retransmitidos	Desordenados	ACK's	Fluxo 1	Fluxo 2
TCP	548 602	0	0	0	0	N/A	N/A
MPTCP c/ COUPLED	545 030	375	2	3	388	81 179	463 851
MPTCP c/ OLIA	276 219	148	6	3	152	46 191	230 028

Como já era previsto, o balanceamento de carga entre duas tecnologias de acesso à rede distintas, onde cada uma possui valores ao nível do débito máximo atingível, atrasos fim-a-fim, etc, bastante diferentes. Com isto, conclui-se que apesar das melhorias introduzidas pelo MPTCP nesta área, mesmo assim, a heterogeneidade das tecnologias de acesso à rede afeta o desempenho do MPTCP porque este tem de esperar pela recepção dos pacotes transmitidos via WiFi, o que faz com o débito atingido na ligação Ethernet seja menor do que se esperava. Em relação ao parâmetro do número de pacotes retransmitidos, neste cenário o TCP obteve resultados excepcionais ao não perder qualquer tipo de pacote. Portanto, neste cenário o número de retransmissões não é um parâmetro que distinga o MPTCP do TCP.

Tabela 4.6: Tabela de Comparação dos RTT no cenário WiFi - Ethernet

Protocolo	RTT (ms)	
	MP 1	MP 2
TCP	1,9	N/A
MPTCP c/ COUPLED	1	0,22
MPTCP c/ OLIA	0,89	0,33

Por último, a análise ao parâmetro RTT mostra que o tempo de estabelecer um *three-way-handshake* num fluxo de dados TCP numa rede sem fios é claramente superior ao tempo demorado no estabelecimento do fluxo de dados sobre a rede Ethernet. No entanto, o MPTCP trouxe uma redução neste valor quer com o algoritmo COUPLED, quer com o algoritmo OLIA.

5

Conclusões e Trabalho Futuro

Com base nos objetivos inicialmente propostos, foram planejados e desenvolvidos dois cenários que permitissem adquirir resultados acerca do comportamento do MPTCP nas duas tecnologias de acesso, Ethernet e WiFi. O estudo visou duas das mais importantes características na entrega de um serviço com qualidade, elas são: a otimização do débito e o balanceamento de carga. Estas características fazem com que a distribuição da carga por todas as rotas disponíveis contribua para a existência de redes menos congestionadas, o que leva a um melhor aproveitamento dos recursos da rede. Como principal consequência da utilização deste tipo de protocolos em detrimento do TCP, faz com que a otimização das infraestruturas seja menos dispensiosa porque não exige a adição de equipamentos extra para obter resultados semelhantes aos alcançados. Com base nisto, optou-se por uma aplicação onde o seu desempenho fosse afetado pelo nível de QoS na ligação, portanto, o protocolo aplicacional escolhido foi o FTP. Sendo assim, nos próximos parágrafos conclui-se sobre as melhorias introduzidas pelo MPTCP no funcionamento do FTP ao nível da otimização do débito e do balanceamento de carga.

No que diz respeito ao balanceamento de carga, os resultados alcançados no cenário Ethernet - Ethernet mostram que o MPTCP faz uma divisão equitativa da carga por todas as rotas disponíveis, independentemente do algoritmo para o controlo de congestão em uso, mas, quando as tecnologias de acesso à rede são diferentes, isto não acontece da mesma forma. As razões para tal devem-se, sobretudo, às limitações impostas pelo *hardware* utilizado, sendo que o MPTCP não consegue alterar os valores das características de cada uma das interfaces. No entanto, como os resultados demonstram, o MPTCP adapta-se de modo a que a carga seja dividida pelos dois caminhos, ao delimitar o débito atingido na interface de rede com fios. Conclui-se assim que o MPTCP traz benefícios ao nível do balanceamento de carga quando todas as interfaces de rede tiverem características semelhantes em termos de débito e de atraso.

Em relação à otimização do débito, com o MPTCP o débito agregado no cenário Ethernet - Ethernet foi quase o dobro do alcançado com o TCP, o que fez com que o tempo de transferência fosse sensivelmente metade do conseguido com o TCP. O mesmo se sucedeu no cenário WiFi - Ethernet, onde o débito agregado foi quatro vezes superior ao alcançado com o TCP, sendo que com o TCP a transferência via FTP fosse feita única e exclusivamente através de uma conexão sem fios. Mais uma vez, o tempo de transferência foi claramente inferior ao obtido com o TCP. De salientar que estes resultados encorajadores foram conseguidos com a utilização de ambos algoritmos para o controlo de congestão, COUPLED e OLIA. Em jeito de conclusão, o MPTCP faz uso de todas as rotas possíveis, quer com fios e quer sem fios, permitindo que o débito agregado consiga ultrapassar o débito máximo conseguido com a utilização de uma única rota.

O número de retransmissões foi outro dos parâmetros analisados e o que os resultados mostraram foi um decréscimo deste valor no cenário Ethernet - Ethernet e um acréscimo no cenário WiFi - Ethernet. Apesar destes resultados terem sido alcançados, eles são de menor relevância visto que ambos os cenários eram cenários ideais, onde o único tráfego que percorria ambas as rotas era o tráfego FTP da comunicação estabelecida. Portanto, este parâmetro só é mais relevante quando se aplicar o MPTCP em redes de maior dimensão.

Resumindo, no global o MPTCP traz mais benefícios do que perdas na entrega de um melhor QoS para a aplicação FTP mesmo quando as tecnologias de acesso à rede têm características tão distintas uma de outra. Por isso, se tivesse que escolher um protocolo/extensão para transportar os dados pelas diversas rotas disponíveis, esse seria o MPTCP por todas as vantagens que os resultados aqui conseguidos o demonstram.

Como trabalho futuro, um dos objetivos é implementar o MPTCP nos dispositivos móveis, especificamente em *tablets* e *smartphones*, para combinar o uso das redes celulares com o uso das redes sem fios locais. De facto, na minha opinião o MPTCP é mais indicado para esta situação porque o uso de múltiplas rotas para convergir a informação permite a continuação da comunicação em caso de falha de uma delas. No futuro, outro objetivo é aplicar estes dois cenários elaborados em redes de maior dimensão, onde existe uma maior quantidade de dados a circular de forma a obter resultados mais abrangentes do real desempenho do MPTCP.

Bibliografia

- [1] H.-Y. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts," *MobiCom'02*, pages 83-94, New York, NY, USA, July 2002.
- [2] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," *USENIX '04*, 2004.
- [3] K. Rojvibooncha and H. Aida, "An evaluation of multi-path transmission control protocol (M/TCP) with robust acknowledgement schemes," *IEICE Trans. Communications*, 2004.
- [4] J. R. Iyengar, K. C. Shah, and P. D. Amer, "Concurrent multipath transfer using SCTP multihoming," *NETWORKING'08 Proceedings of the 7th international IFIP-TC6 networking conference on AdHoc and sensor networks, wireless networks, next generation internet*, pages 727-734, May 2008.
- [5] M. Becke, J. Iyengar, P. Natarajan, and M. Tuexen, "Load sharing for the stream control transmission protocol (SCTP)," July 2010.
- [6] C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," *IETF, RFC 6824*, October 2012.
- [7] C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural Guidelines for Multipath TCP Development," *IETF, RFC 6182*, January 2011.
- [8] M. Bagnulo, "Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses," *IETF, RFC 6181*, January 2011.
- [9] B. Ford and J. Iyengar, "Breaking Up the Transport Logjam," *ACM HotNets*, October 2008.
- [10] C. Raiciu, M. Handley, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," *IETF, RFC 6356*, July 2011.
- [11] R. Khalilii, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. L. Boudec, "MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution," *CoNEXT'12*, December 2012.
- [12] J. Postel, "Transmission Control Protocol, STD 7," *IETF, RFC 793*, September 1981.
- [13] M. Scharf and A. Ford, "MPTCP Application Interface Considerations," *IETF, RFC 6897*, January 2013.

[14] <http://www.nrl.navy.mil/itd/ncs/products/core>.

[15] <http://mptcp.info.ucl.ac.be/>.



Wireshark

A.1 Cenário Ethernet - Ethernet

A.1.1 TCP

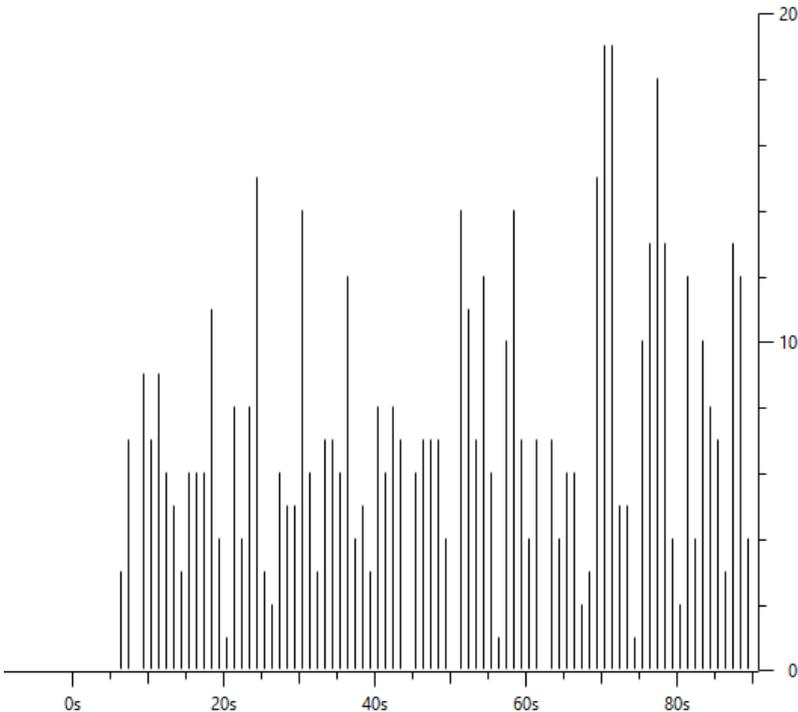


Figura A.1: Número de ACKs perdidos com o TCP no cenário Ethernet - Ethernet.

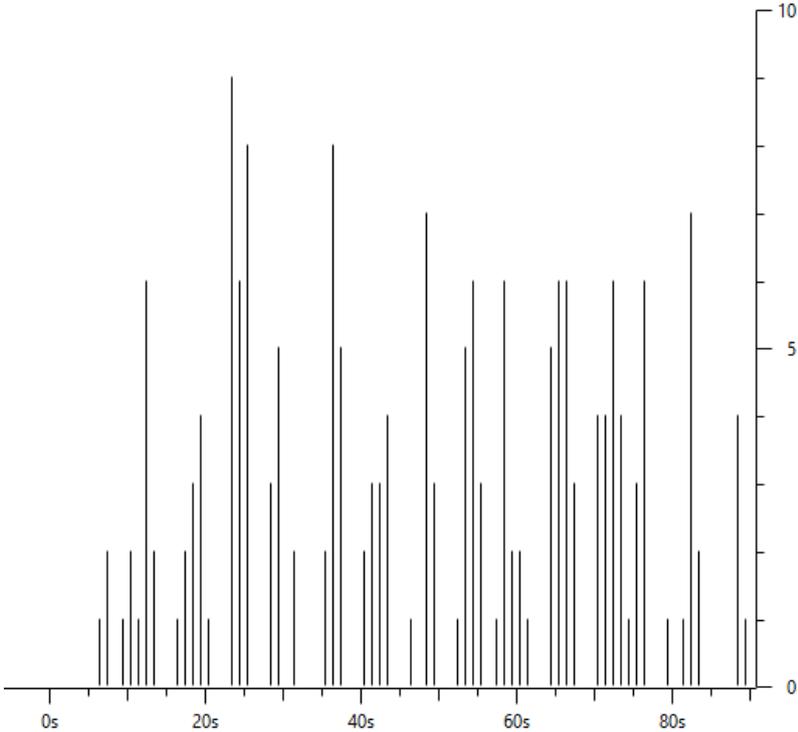


Figura A.2: Número de pacotes desordenados com o TCP no cenário Ethernet - Ethernet.

A.1.2 MPTCP com COUPLED

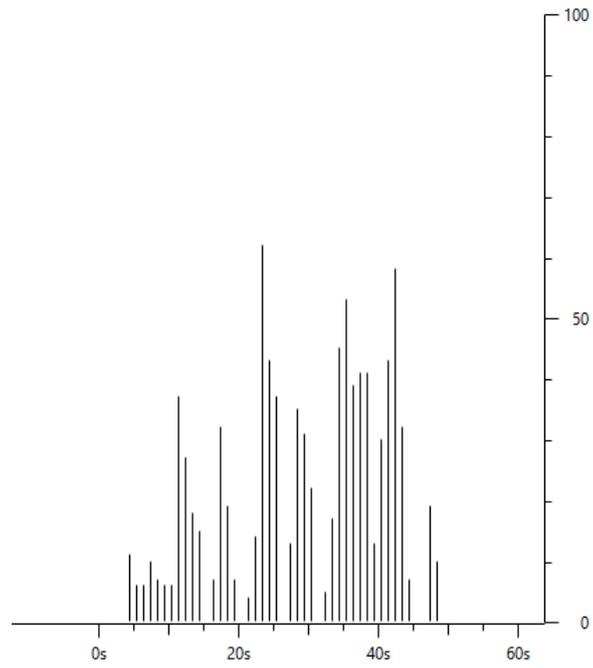


Figura A.3: Número de ACKs perdidos com o COUPLED no cenário Ethernet - Ethernet.

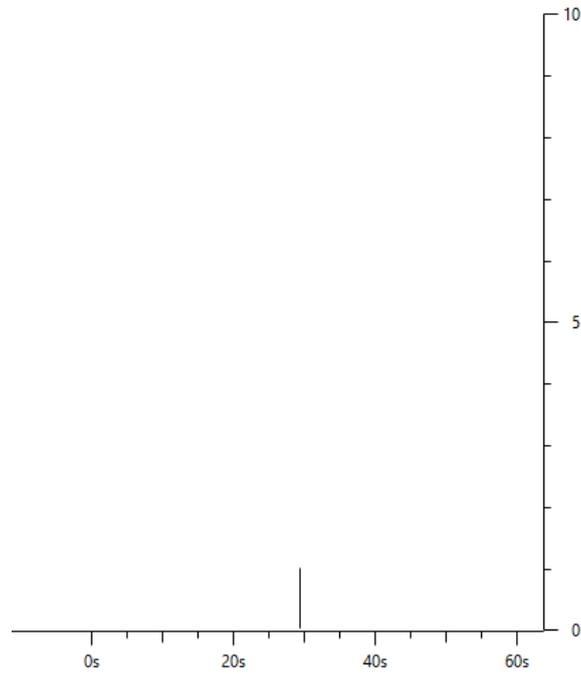


Figura A.4: Número de pacotes desordenados com o COUPLED no cenário Ethernet - Ethernet.

A.1.3 MPTCP com OLIA

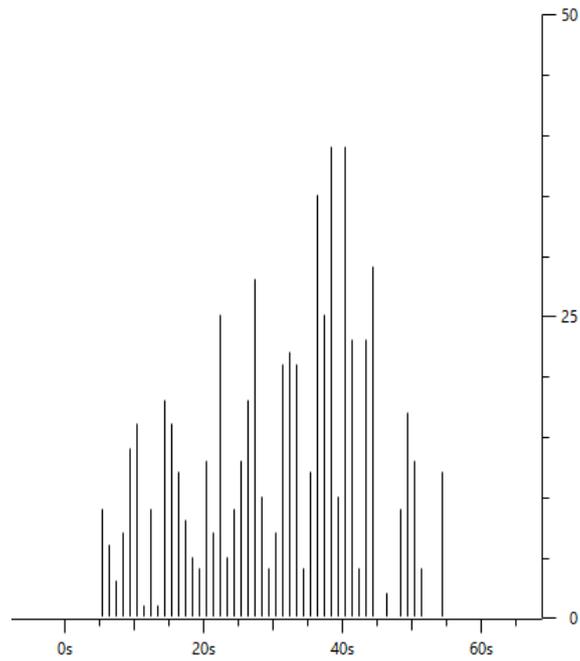


Figura A.5: Número de ACKs perdidos com o OLIA no cenário Ethernet - Ethernet.

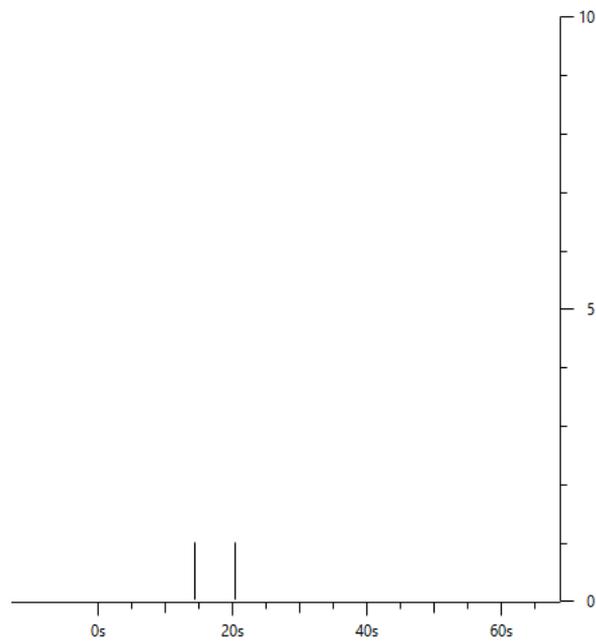


Figura A.6: Número de pacotes desordenados com o OLIA no cenário Ethernet - Ethernet.

A.2 Cenário WiFi - Ethernet

A.2.1 MPTCP com COUPLED

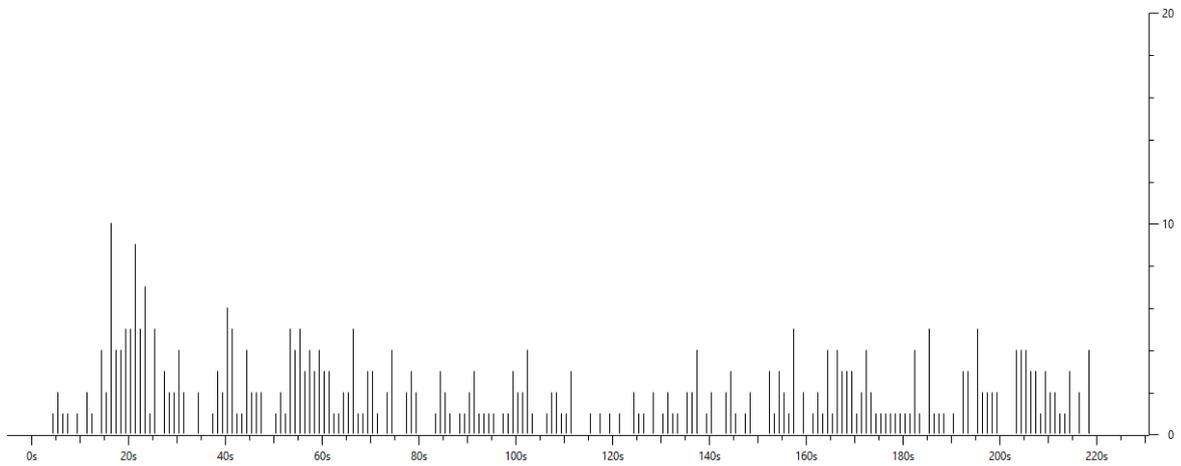


Figura A.7: Número de ACKs perdidos com o COUPLED no cenário WiFi - Ethernet.

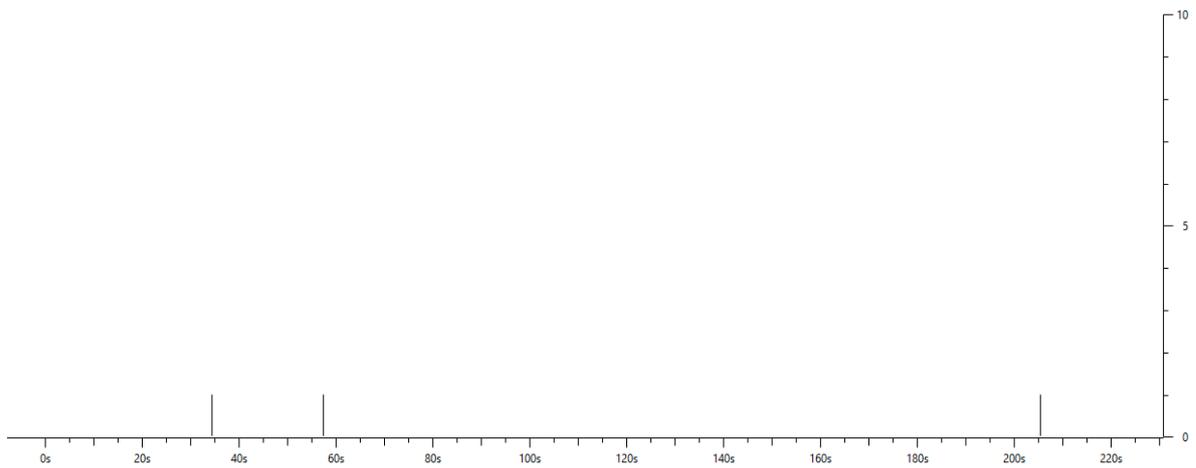


Figura A.8: Número de pacotes desordenados com o COUPLED no cenário WiFi - Ethernet.

A.2.2 MPTCP com OLIA

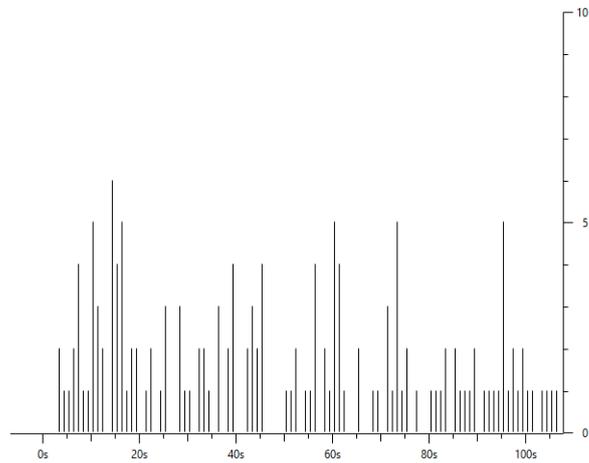


Figura A.9: Número de ACKs perdidos com o OLIA no cenário WiFi - Ethernet.

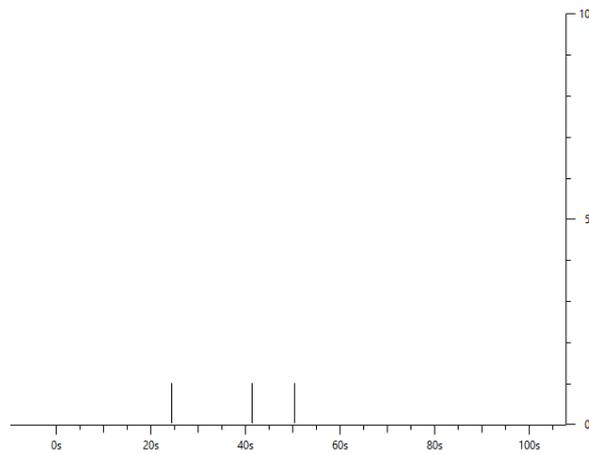


Figura A.10: Número de pacotes desordenados com o OLIA no cenário WiFi - Ethernet.