U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Faculdade de Engenharia da Universidade do Porto

Redes de Computadores

3º ANO - **MIEIC**

Rede de Computadores

Trabalho Laboratorial 2

Estudantes & Autores

Diogo Silva, up201706892

up201706892@fe.up.pt

João Henrique Luz, up201703782

up201703782@fe.up.pt

21 de **dezembro** de **2019**

Sumário

Este trabalho, desenvolvido no âmbito da unidade curricular de Redes de Computadores, tem como objetivo a configuração de uma rede de computadores e respetivo estudo, tendo sido usado, para tal, um *router* e um *switch* para interligar três computadores (Tuxes). Posteriormente foi desenvolvida uma aplicação para o *download* de ficheiros de um servidor através do protocolo **FTP**.

Posto isto, este relatório tem por objetivo esclarecer as diferentes etapas procedimentais deste projeto. É ainda de salientar que todos os requisitos foram cumpridos, tendo resultado numa rede corretamente configurada e numa aplicação viável para o uso pretendido.

Índice

Tabela de Figuras

A. Introdução

Este trabalho está dividido em duas partes: aplicação de *download* e configuração da rede de computadores.

Na primeira, o objetivo foi conseguido através da implementação do **File Transfer Protocol** (FTP) e do uso do **Transmission Control Protocol** (TCP) a partir de *sockets*. Assim, foi possível garantir a integridade dos dados transferidos e prevenir eventuais falhas de ligação e corrupção de dados.

A segunda fase consistiu na realização de seis experiências com vista a uma melhor compreensão da forma como diferentes computadores e dispositivos interagem entre si dentro duma rede.

O presente relatório procura examinar as diferentes componentes do projeto e a forma como se complementam para o bom funcionamento do mesmo. Assim, este tem a seguinte estrutura:

- **Aplicação de Download**

  Arquitetura e resultados.

- **Configuração da Rede**

  Análise da arquitetura da rede e conclusões para cada experiência.

- **Conclusões**

  Síntese da informação apresentada nas secções anteriores e reflexão sobre os objetivos de aprendizagem alcançados.

B. Aplicação de Download

Para a elaboração da primeira fase do projeto, foi desenvolvida uma aplicação que, uma vez fornecido um URL com o formato: **ftp://<user>:<password>@<host>/<url-path>**, é capaz de fazer o *download* do ficheiro especificado por **url-path** de um servidor FTP. Assim sendo, foi tido em conta o procedimento descrito em **RFC959** para o desenvolvimento desta aplicação.

B1. Arquitetura

Para obter os resultados pretendidos foi concebida uma aplicação na qual são desencadeadas as seguintes fases:

B1.1 Processamento dos argumentos

Recorrendo à função `parse_arguments` é feita a extração das informações necessárias para a conexão ao servidor, autenticação e *download* do ficheiro especificado. Os dados obtidos nesta função são armazenados na `struct url_arg` do tipo `url_syntax`.

B1.2 Início de conexão ao servidor

Uma vez processados os argumentos, é aberto um primeiro *socket* de controlo através do qual serão enviados os comandos do cliente e recebidas as respetivas respostas do servidor. Para tal foi necessária a função `get_IP`, que dado nome do *host* devolve o seu IP público, recorrendo ao DNS. É também de realçar a função `get_reply` responsável por obter os códigos de retorno vindos das respostas do servidor.

B1.3 Autenticação

Para que possa ser possível o envio para o servidor de comandos por parte do cliente é necessária a autenticação do mesmo. Se nos argumentos fornecidos forem especificados um utilizador e uma palavra-passe, a autenticação é feita recorrendo a essas informações. Caso contrário, a operação é feita definindo o utilizador como `anonymous` e a palavra-passe como `pass`, embora o valor desta última seja possa ser qualquer outro.

B1.4 Habilitação do modo passivo

É com o envio do comando PASV que é feita a habilitação do modo passivo, sendo a resposta a este comando por parte do servidor o que permite o cálculo da porta necessária para abrir um

segundo *socket* através do qual será transferido o ficheiro pretendido.

B1.5 *Download* do ficheiro pretendido

Nesta fase do processo, é aberto um novo ficheiro com o nome pedido. De seguida, é enviado o comando RETR para dar início à transferência de dados através do segundo *socket* criado. A leitura e consequente escrita dos dados lidos para o novo ficheiro constituem a operação de *download* do ficheiro.

B1.6 Fim de conexão ao servidor

Após concluída a transferência, são fechados os *sockets* utilizados na mesma, assim como o novo ficheiro obtido.

B2. Resultados

A aplicação pode ser usada em modo anónimo e modo autenticado para download de ficheiros em qualquer diretoria dentro do servidor FTP e de diversos tamanhos. Ao longo do processo são indicadas as etapas realizadas e caso este não seja bem sucedido, é indicado o motivo pelo qual falhou, sendo a interface (num exemplo que demonstra o comportamento esperado) a seguinte:

```
$ ./application ftp://anonymous:pass@ftp.up.pt/parrot/last-update.txt

  [Information parsed]
    - Server: ftp://
    - User: anonymous
    - Password: pass
    - Host: ftp.up.pt
    - URL: parrot/last-update.txt
    - Filename: last-update.txt
  [Connection established]
  [User authenticated]
  [Passive mode enabled]
  [Sending RETR command]
  [Download completed]
  [Sockets disconnected]

$
```

C. Configuração da Rede

C1. Experiência 1: Configuração do IP de uma Rede

Nesta experiência são atribuídos IPs ao Tux1 e Tux4 de forma a ligá-los através *switch*.

1. O que são pacotes ARP e para que são usados?

Address Resolution Protocol (ARP) é um protocolo de comunicação usado para descobrir o endereço da camada de ligação, neste caso o MAC address, associado a um dado endereço da camada da *Internet*, que no nosso caso será o endereço IPv4.

2. Quais são os endereços MAC e IP dos pacotes ARP e porquê?

Nos pacotes ARP estão registados os endereços MAC e IP do computador de origem, e se já estiverem registados na tabela ARP, o endereço MAC e IP de do computador de destino. Caso contrário, o endereço IP é o de destino mas o endereço MAC terá o valor 00:00:00:00:00:00.

3. Quais os pacotes gerados pelo comando *ping*?

Os pacotes gerados pelo comando *ping* são ICMP (Internet Control Message Protocol). Echo Request messages. No entanto, caso ainda não esteja registada na tabela ARP uma correspondência entre todos os endereços MAC e IP envolvidos, são também gerados pacotes ARP.

4. Quais são os endereços MAC e IP dos pacotes *ping*?

No pacote *ping* enviado pelo Tux1 os endereços MAC e IP de origem são os do mesmo, neste caso 172.16.20.1 e 00:21:5a:61:2f:9b, e os de destino são os do Tux4, 172.16.20.254 e 00:22:64:a7:26:a2, respectivamente.

No caso do pacote *ping* recebido pelo Tux1, os endereços MAC e IP são os opostos aos do pacote enviado, ou seja os de origem são os do Tux4 e os de destinos são os do Tux1, referidos anteriormente.

5. Como determinar se a trama recetora Ethernet é ARP, IP ou ICMP?

Essa informação está presente no *Ethernet Header* da própria trama. Nesse cabeçalho existe o campo *EtherType* que determina o protocolo utilizado, entre os quais:

- `0x0806`: Address Resolution Protocol (ARP);
- `0x0800`: Internet Protocol version 4 (IPv4);
- `0x86DD`: Internet Protocol version 6 (IPv6).

Por sua vez, consegue-se determinar se se trata de ICMP pela análise do IPv4 *header*, sendo que para tal o *Protocol Number* terá de corresponder ao `0x01`. Na **Figura 1** pode ser consultado um exemplo.

6. Como determinar o comprimento de uma trama recetora?

O comprimento duma trama recetora pode ser determinado através da análise dos registos feitos pelo Wireshark, inspecionando o campo *Frame Length*, tal como indicado na **Figura 2**.

7. O que é a interface *loopback* e porque é importante?

A interface *loopback* é uma interface de rede virtual usada pelo computador para comunicar com si próprio. É importante para *diagnostics* e *troubleshooting* de modo a verificar se a rede está bem configurada.

C2. Experiência 2: Implementação de duas LANs Virtuais num *Switch*

Procedeu-se à criação de duas LANs virtuais (VLANs) às quais foram associadas Tuxes:

- `VLANy0`: Tux1 e Tux4;
- `VLANy1`: Tux2.

Antes de serem feitos os comandos alusivos à configuração das VLANs no *switch*, foram feitas a seguintes ligações:

- Porta de série do Tux1 ligada à porta *switch console*;
- Porta `eth0` do Tux1 ligada à porta 1 do *switch;*
- Porta `eth0` do Tux4 ligada à porta 4 do *switch*.

1. Como configurar a `VLANy0`?

Estando o Tux1 ligado à consola do *switch* pela porta de série, foi utilizado neste computador o software GTKTerm para inserir os seguintes comandos para criar uma VLAN:

```
$ configure terminal
$ vlan y0
$ end
```

São adicionadas portas, neste caso a porta 1, a uma VLAN utilizando os seguindes comandos:

```
$ configure terminal
$ interface fastethernet 0/1[1]
$ switchport mode access
$ switchport access vlan y0
$ end
```

[1] ou `0/x` para uma porta x

2. Quantos domínios de transmissão existem? Como se pode concluir a partir dos registos?

Existem dois domínios de transmissão: um que engloba o Tux1 e o Tux4 e outro onde está apenas o Tux2. Tal pode ser verificado porque quando o Tux1 efetua um *ping broadcast* recebe a resposta do Tux4, mas quando o mesmo é feito pelo Tux2, nenhuma resposta é obtida, tal como no exemplo da **Figura 3**.

C3. Experiência 3: Configuração de um *Router* em Linux

Nesta experiência procedeu-se à configuração do Tux4 para que funcionasse como um *router*, de forma a estabelecer uma ligação entre as duas VLANs existentes e os computadores presentes nas mesmas.

1. Que rotas há nos Tuxes? Qual o seu significado?

As rotas definidas para o Tux1 foram as seguintes:

- Rota para a rede 172.16.y0.0 (`VLANy0`) através da *gateway* 172.16.y0.1
- Rota para a rede 172.16.y1.0 (`VLANy1`) através da *gateway* 172.16.y0.254

No Tux4 as rotas registadas foram as seguintes:

- Rota para a rede 172.16.y0.0 (`VLANy0`) através da *gateway* 172.16.y0.254
- Rota para a rede 172.16.y0.1 (`VLANy1`) através da *gateway* 172.16.y1.253

Por fim, no Tux2 ficaram definidas as rotas:

- Para a rede 172.16.y1.0 através da *gateway* 172.16.y1.1
- Para a rede 172.16.y0.0 através da *gateway* 172.16.y1.253

2. Que informação é que uma entrada da tabela de *forwarding* contém?

Nesta tabela estão presentes rotas para determinados destinos dentro da rede, tendo os seguintes campos:

- **Network Destination**: destino da rota.
- **Netmask**: juntamente com o campo anterior permite determinar o ID da rede.
- **Gateway**: endereço do próximo ponto para o qual o pacote terá de ser mandado no caminho até ao seu destino final.
- **Interface**: indica qual a interface disponível localmente é responsável por atingir o *gateway*, por exemplo, `eth0` ou `eth1`.
- **Metric**: indica o custo da rota.

3. Que mensagens ARP e endereços MAC associados são observados e porquê?

Aquando da execução do comando *ping* no Tux1 com destino ao Tux4, por exemplo, o Tux de destino responde com uma mensagem ARP na qual está registado o seu endereço MAC e que procura saber qual o endereço MAC correspondente ao IP de origem do *ping*. Nesta mensagem o endereço MAC de origem é o do Tux4 e o valor do endereço MAC de destino é 00:00:00:00:00 dado que não existe correspondência entre este e o IP do Tux que fez *ping* (neste caso, o Tux1).

Por fim, o Tux que fez *ping*, responde com uma mensagem ARP na qual está registado o seu endereço MAC como origem da mensagem, e o endereço MAC do segundo Tux enquanto destino. Deverá ser consultada a **Figura 4**.

Que pacotes ICMP são observados e porquê?

São observados vários pacotes do tipo ICMP de *request* e *reply* já que todos os Tuxes se conseguem "ver" uns aos outros dentro da rede após estarem configuradas as rotas. Caso tal não acontecesse, os pacotes ICMP seriam de *Host Unreachable*.

4. Quais são os endereços IP e MAC associados a pacotes ICMP e porquê?

Os pacotes ICMP contêm associados os endereços IP e MAC dos Tuxes de origem e destino, podendo ser comprovado pela **Figura 5**.

C4. Experiência 4: Configuração de um *Router* Comercial e Implementação da NAT

De forma a completar esta experiência, efetuou-se a configuração do *router* para que tivesse acesso à rede do laboratório. De seguida, passamos à implementação da NAT para que os computadores na rede tivessem acesso à *Internet*.

1. Como se configura uma rota estática num *router* comercial?

Estando agora o Tux1 ligado à consola do router pela porta de série, foi utilizado neste computador o software GTKTerm para inserir os seguintes comandos para configurar:

```
  $ configure terminal
  $ ip route prefix mask {ip-address| interface-typeinterface-
number[ip-address]}
  $ end
```

2. Quais são os caminhos seguidos pelos pacotes nas experiências e porquê?

Se for possível utilizar uma rota existente, os pacotes utilizam a mesma. Se tal não for possível, estes são redirecionados para a rota *default*, que neste caso será a do *router*. A partir desse momento, passa a ser possível utilizar o Tux4 e este é utilizado para que os pacotes

cheguem ao destino.

3. Como se configura a NAT num *router* comercial?

À semelhança da configuração da rota estática, a configuração da NAT foi feita através do através do Tux1, no qual foram introduzidos os comandos presentes na **Figura 6**.

4. O que faz a NAT?

Network Address Translation (NAT) é um método de remapeamento de um endereço IP. Esta técnica tornou-se extremamente importante na conservação do espaço de endereçamento do IPv4 devido à sua atual exaustão. A NAT permite que seja feita a tradução de endereços privados em endereços publicos na transmissão de pacotes entre redes privadas e publicas. Desta forma, permite que dispositivos dentro de uma rede privada tenham acesso à *Internet* através de um único IP público.

C5. Experiência 5: DNS

Nesta experiência procedeu-se à configuração do serviço DNS (Domain Name System) nos Tuxes da nossa rede. Para isso utilizou-se o servidor netlab.fe.up.pt que contém os endereços IP públicos associados a um *host name*, possibilitando assim a sua conversão.

1. Como configurar o serviço DNS num *host*?

Para configurar o serviço DNS num dado *host* temos de garantir que o ficheiro `resolv.conf`, localizado em `/etc/` contém a seguinte configuração:

```
search netlab.fe.up.pt
nameserver 172.16.2.1[2]
```

Indicamos assim que queremos utilizar o servidor DNS netlab.fe.up.pt com o respetivo endereço IP.

2. Que pacotes são trocados pelo DNS e que informações são transportadas?

O *host* envia o *server* o *host name* cujo IP deseja conhecer. De seguida, o *server* envia um pacote com o IP que corresponde ao *host name* desejado, como constatado na **Figura 7**.

C6. Experiência 6: Conexões TCP

Nesta última experiência foi utilizado o protocolo TCP implementado na aplicação feita na primeira parte do trabalho para observar e atestar o correto comportamento da rede.

[2] Endereço na sala I320.

1. Quantas conexões TCP foram abertas pela aplicação FTP?

Foram abertas duas conexões TCP: uma para enviar comandos ao servidor e receber as respetivas respostas e outra para receber os dados do servidor.

2. Em que conexão é transportado o controlo de informação do FTP?

O controlo de informação é transportado na primeira conexão TCP, a responsável pela troca de comandos.

3. Quais as fases duma conexão TCP?

Existem três fases: o estabelecimento da conexão, a fase de troca de dados e a terminação da conexão.

4. Como é que o mecanismo ARQ TCP funciona? Quais os campos TCP relevantes? Que a informação relevante é observada nos registos?

O protocolo TCP utiliza o Automatic Repeat Request (ARQ) em conjunto com o Sliding Window Protocol, um método de controlo de erros na transmissão de dados que usa *acknowledgements* (mensagens enviadas pelo recetor a indicar que recebeu o pacote corretamente). Assim, a cada *byte* enviado é atribuído um *sequence number*, permitindo que o recetor os ordene, descarte duplicados e identifique os que faltam. Existindo uma *window size*, permite que o número de sequência tenha um teto máximo fixo e evita a congestão da rede.

5. Como é que o mecanismo de controlo de congestão TCP funciona? Quais os campos relevantes? Como é que o fluxo de dados da conexão evoluiu ao longo do tempo? Está de acordo com o mecanismo de controlo de congestão TCP?

Para cada conexão, o TCP mantém uma *congestion window*, limitando o número total de bytes *unacknowledged* que podem estar em trânsito (análogo à *sliding window*).

No exemplo da **Figura 8** vemos que a *bitrate* da transferência aumenta exponencionalmente no início até estabilizar. Essa estabilidade mantém-se até existir uma segunda conexão. A partir desse momento verificamos uma queda abrupta da taxa de transferência. Até acabar o *download*, continua-se a verificar alguma instabilidade. Assim, vemos que está de acordo com o mecanismo de controlo de congestão TCP.

6. De que forma é afetada a conexão de dados TCP pelo aparecimento de uma segunda conexão TCP? Como?

Como a *bitrate* da transferência é dividida equitativamente entre as conexões TCP existentes, ao passarem a existir duas transferências em simultâneo, a taxa irá ser reduzida para cerca de metade.

Conclusões

Em suma, reiteramos que cumprimos os objetivos propostos para este projeto. Tendo sido a configuração da rede de computadores efetuada experimentalmente e passo a passo, facilitou a compreensão dos seus conceitos. Para além disto, a aplicação desenvolvida é capaz de realizar o *download* de ficheiros de um servidor recorrendo ao protocolo FTP, estando apta a reagir de forma adequada em várias situações.

Na nossa opinião, este projeto revelou-se muito útil, facilitando, através de uma abordagem prática, a consolidação de conceitos teóricos.

Anexo I

Será de salientar que embora a bancada utilizada seja a 2 da sala I320, por motivos técnicos alguns dos *logs* foram feitos na bancada 1 da mesma sala, podendo haver assim discrepância nos endereços IP e MAC apresentados.

```
Ethernet II, Src: Kye_06:69:3e (00:c0:df:06:69:3e), Dst: HewlettP_5a:7b:3f (00:21:5a:5a:7b:3f)
  > Destination: HewlettP_5a:7b:3f (00:21:5a:5a:7b:3f)
  > Source: Kye_06:69:3e (00:c0:df:06:69:3e)
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.16.10.1, Dst: 172.16.10.254
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x75fd (30205)
  > Flags: 0x4000, Don't fragment
    ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
```

Figura **1**: *Ethernet Header*

```
Encapsulation type: Ethernet (1)
Arrival Time: Nov 20, 2019 10:30:51.350205451 Hora padrão de GMT
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1574245851.350205451 seconds
[Time delta from previous captured frame: 0.541787262 seconds]
[Time delta from previous displayed frame: 0.541787262 seconds]
[Time since reference or first frame: 9.148813337 seconds]
Frame Number: 12
Frame Length: 98 bytes (784 bits)
Capture Length: 98 bytes (784 bits)
```

Figura **2**: *Frame Length*

```
28 26.740437454  172.16.10.1     172.16.10.254    ICMP     98 Echo (ping) request  id=0x41c1, seq=5/1280, ttl=64 (reply in 29)
29 26.740595715  172.16.10.254   172.16.10.1      ICMP     98 Echo (ping) reply    id=0x41c1, seq=5/1280, ttl=64 (request in 28)

59 44.972430504  172.16.10.1     172.16.11.0      ICMP     98 Echo (ping) request  id=0x41cb, seq=8/2048, ttl=64 (no response found!)
60 45.972435183  172.16.10.1     172.16.11.0      ICMP     98 Echo (ping) request  id=0x41cb, seq=9/2304, ttl=64 (no response found!)
```

Figura **3**: Demonstração dos domínios de transmissão

```
    36 33.959318414  HewlettP_5a:7b:3f    Kye_06:69:3e         ARP          60 Who has 172.16.10.1? Tell 172.16.10.254
    37 33.959328681  Kye_06:69:3e         HewlettP_5a:7b:3f    ARP          42 172.16.10.1 is at 00:c0:df:06:69:3e

> Source: HewlettP_5a:7b:3f (00:21:5a:5a:7b:3f)
  Type: ARP (0x0806)
  Padding: 000000000000000000000000000000000000
Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: HewlettP_5a:7b:3f (00:21:5a:5a:7b:3f)
  Sender IP address: 172.16.10.254
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 172.16.10.1
```

Figura 4:Endereços IP e MAC numa mensagem ARP

```
Ethernet II, Src: Kye_06:69:3e (00:c0:df:06:69:3e), Dst: HewlettP_5a:7b:3f (00:21:5a:5a:7b:3f)
> Destination: HewlettP_5a:7b:3f (00:21:5a:5a:7b:3f)
> Source: Kye_06:69:3e (00:c0:df:06:69:3e)
  Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 172.16.10.1, Dst: 172.16.10.254
```

Figura 5:Endereços IP e MAC associados a um pacote ICMP

- Cisco NAT
  http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094e77.shtml

```
conf t
interface gigabitethernet 0/0 *
ip address 172.16.y1.254 255.255.255.0
no shutdown
ip nat inside
exit

interface gigabitethernet 0/1*
ip address 172.16.1.y9 255.255.255.0
no shutdown
ip nat outside
exit

ip nat pool ovrld 172.16.1.y9 172.16.1.y9 prefix 24
ip nat inside source list 1 pool ovrld overload

access-list 1 permit 172.16.y0.0 0.0.0.7
access-list 1 permit 172.16.y1.0 0.0.0.7

ip route 0.0.0.0 0.0.0.0 172.16.1.254
ip route 172.16.y0.0 255.255.255.0 172.16.y1.253
end
```

* In room I320 use interface fastethernet

Figura 6: Configurações NAT

```
4452 12.502539   172.16.20.1   172.16.2.1    DNS   70 Standard query 0xcd09 A google.com
4497 12.585504   172.16.2.1    172.16.20.1   DNS   86 Standard query response 0xcd09 A google.com A 172.217.16.238
```

Figura 7: DNS

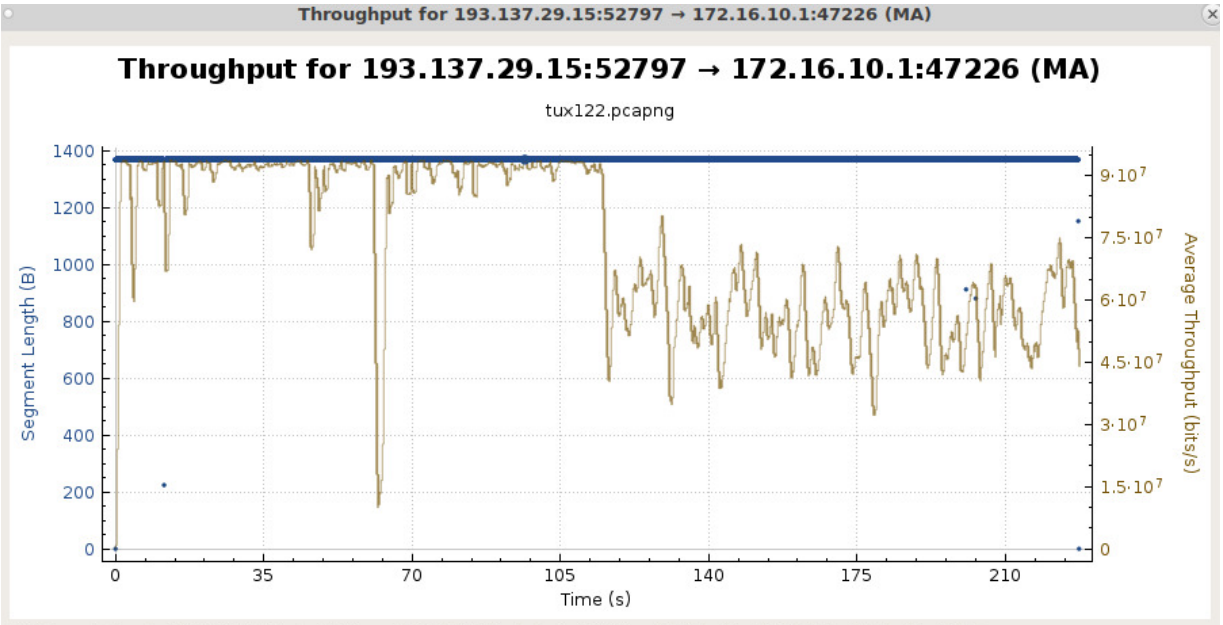

Figura **8**: Bitrate ao longo do transferência simultânea nos dois computadores