MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

RELATÓRIO TÉCNICO

LABORATÓRIO - TCP x UDP

Arthur Cadore Matuella Barcella

TAREFA:

TCP x UDP

Objetivos

- O objetivo desses experimentos é evidenciar as diferenças entre os protocolos TCP e UDP.
- Ambos protocolos de transporte podem ser usados por aplicações que precisem se comunicar. Porém cada um deles têm certas propriedades, então a escolha precisa ser realizada baseada nas necessidade de comunicação a ser feita pela aplicação.

Roteiro

O que aconteceria se um arquivo fosse transferido de um computador a outro com ambos protocolos?

O roteiro será executado sobre máquinas virtuais, através do uso do Imunes. É o primeiro contato, por hora não se preocupe muito com ele, somente siga os passos.

1. Abra um terminal e baixe o aquivo de configuração da rede a ser utilizada e um arquivo auxiliar dos experimentos:

```
wget -4 http://docente.ifsc.edu.br/odilson/RED29004/TCPxUDP.imn
wget -4 http://docente.ifsc.edu.br/odilson/RED29004/original.txt
```

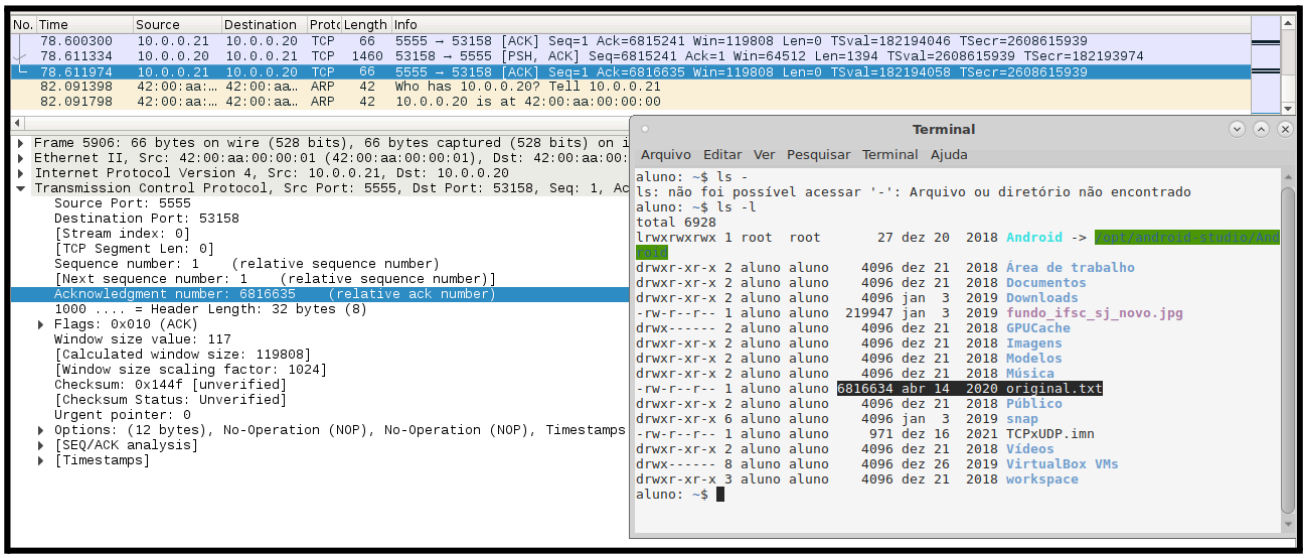
2. Observe o tamanho do arquivo auxiliar transferido, original.txt, ele deve ter exatamente 6816634 bytes (cerca de 6,6 MB). Você pode fazer isso com o comando **ls -l /home/aluno/lab/shared/original.txt**.

Transferência utilizando o protocolo TCP

```
78.236826   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6771801 Win=117760 Len=0 TSval=182193683 TSecr=2608615600
78.248459   10.0.0.20  10.0.0.21  TCP  1514   53158 → 5555 [ACK] Seq=6771801 Ack=1 Win=64512 Len=1448 TSval=2608615600 TSecr=182193635
78.260568   10.0.0.20  10.0.0.21  TCP  1514   53158 → 5555 [ACK] Seq=6773249 Ack=1 Win=64512 Len=1448 TSval=2608615624 TSecr=182193659
78.261193   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6774697 Win=118784 Len=0 TSval=182193707 TSecr=2608615600
78.272671   10.0.0.20  10.0.0.21  TCP  1514   53158 → 5555 [ACK] Seq=6774697 Ack=1 Win=64512 Len=1448 TSval=2608615624 TSecr=182193659
78.296874   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6776145 Ack=1 Win=64512 Len=2896 TSval=2608615649 TSecr=182193683
78.297487   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6779041 Win=119808 Len=0 TSval=182193744 TSecr=2608615624
78.321076   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6779041 Ack=1 Win=64512 Len=2896 TSval=2608615673 TSecr=182193707
78.321712   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6781937 Win=119808 Len=0 TSval=182193768 TSecr=2608615673
78.345303   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6781937 Ack=1 Win=64512 Len=2896 TSval=2608615709 TSecr=182193744
78.345943   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6784833 Win=119808 Len=0 TSval=182193792 TSecr=2608615709
78.369526   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6784833 Ack=1 Win=64512 Len=2896 TSval=2608615733 TSecr=182193768
78.370165   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6787729 Win=119808 Len=0 TSval=182193816 TSecr=2608615733
78.381636   10.0.0.20  10.0.0.21  TCP  1514   53158 → 5555 [ACK] Seq=6787729 Ack=1 Win=64512 Len=1448 TSval=2608615733 TSecr=182193768
78.405862   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6789177 Ack=1 Win=64512 Len=2896 TSval=2608615758 TSecr=182193792
78.406500   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6792073 Win=119808 Len=0 TSval=182193853 TSecr=2608615733
78.430088   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6792073 Ack=1 Win=64512 Len=2896 TSval=2608615782 TSecr=182193816
78.430727   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6794969 Win=119808 Len=0 TSval=182193877 TSecr=2608615782
78.454311   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6794969 Ack=1 Win=64512 Len=2896 TSval=2608615818 TSecr=182193853
78.454954   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6797865 Win=119808 Len=0 TSval=182193901 TSecr=2608615818
78.478533   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6797865 Ack=1 Win=64512 Len=2896 TSval=2608615818 TSecr=182193853
78.502756   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6800761 Ack=1 Win=64512 Len=2896 TSval=2608615842 TSecr=182193877
78.503397   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6803657 Win=119808 Len=0 TSval=182193949 TSecr=2608615842
78.526979   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6803657 Ack=1 Win=64512 Len=2896 TSval=2608615867 TSecr=182193901
78.527617   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6806553 Win=119808 Len=0 TSval=182193974 TSecr=2608615867
78.551207   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6806553 Ack=1 Win=64512 Len=2896 TSval=2608615915 TSecr=182193949
78.551848   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6809449 Win=119808 Len=0 TSval=182193998 TSecr=2608615915
78.575429   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6809449 Ack=1 Win=64512 Len=2896 TSval=2608615915 TSecr=182193949
78.576069   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6812345 Win=119808 Len=0 TSval=182194022 TSecr=2608615915
78.599657   10.0.0.20  10.0.0.21  TCP  2962   53158 → 5555 [ACK] Seq=6812345 Ack=1 Win=64512 Len=2896 TSval=2608615939 TSecr=182193974
78.600300   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6815241 Win=119808 Len=0 TSval=182194046 TSecr=2608615939
78.611334   10.0.0.20  10.0.0.21  TCP  1460   53158 → 5555 [PSH, ACK] Seq=6815241 Ack=1 Win=64512 Len=1394 TSval=2608615939 TSecr=182193974
78.611974   10.0.0.21  10.0.0.20  TCP    66   5555 → 53158 [ACK] Seq=1 Ack=6816635 Win=119808 Len=0 TSval=182194058 TSecr=2608615939
```

Verifique o tamanho do arquivo recebido. Ele é igual ao arquivo original? E quanto tempo levou para transmiti-lo?

Foram necessários 78 segundos para a transmissão. O tamanho do pacote é igual ao original, descrito abaixo:



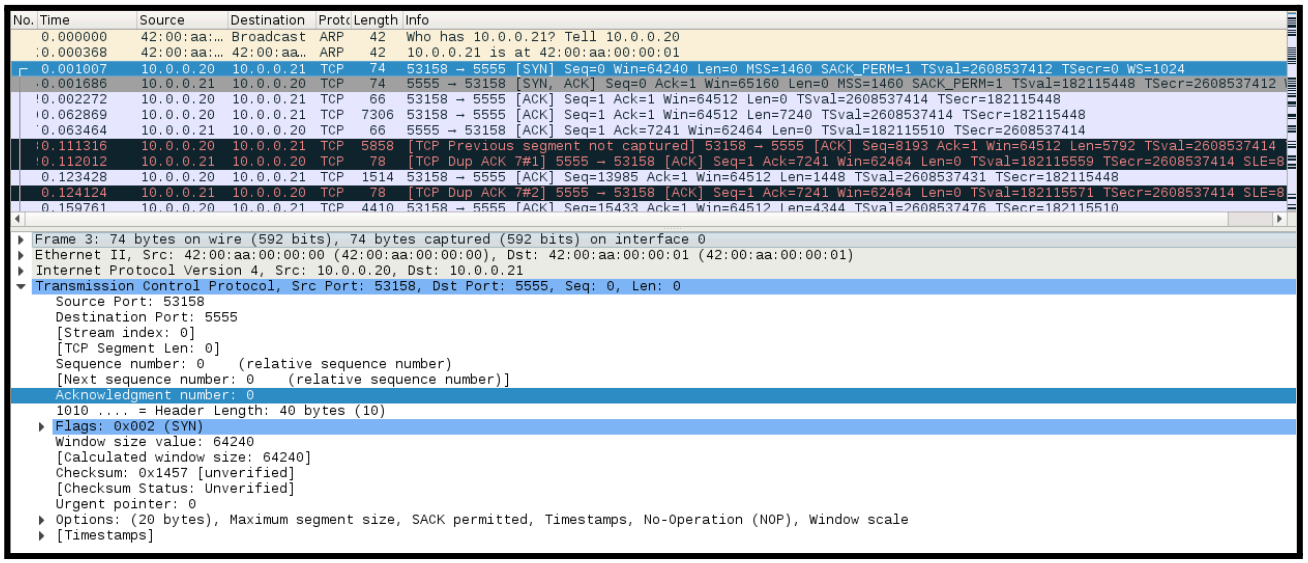
Analisando a captura de pacotes do WireShark responda. Quais as portas origem e destino escolhidas pelo cliente e servidor?

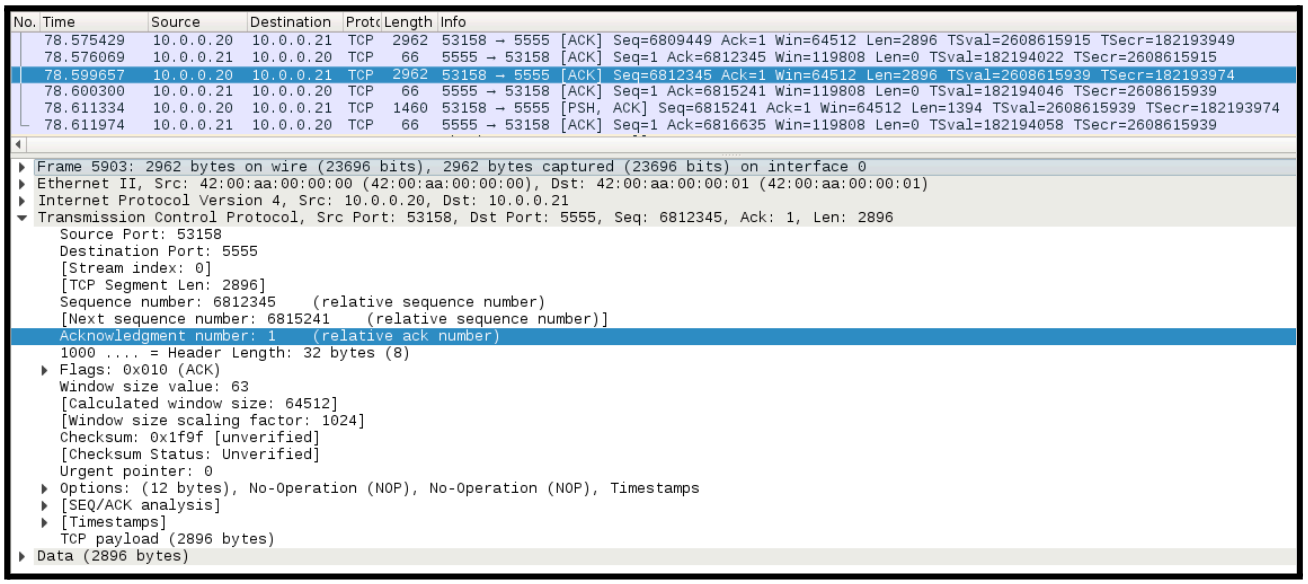
Qual é o número de sequência do primeiro e do último pacote?

Seq: 0 inicial e 6812345 final.

Qual é o número de sequência do primeiro e do último ACK?

ACK: 0 inicial e 1 final;





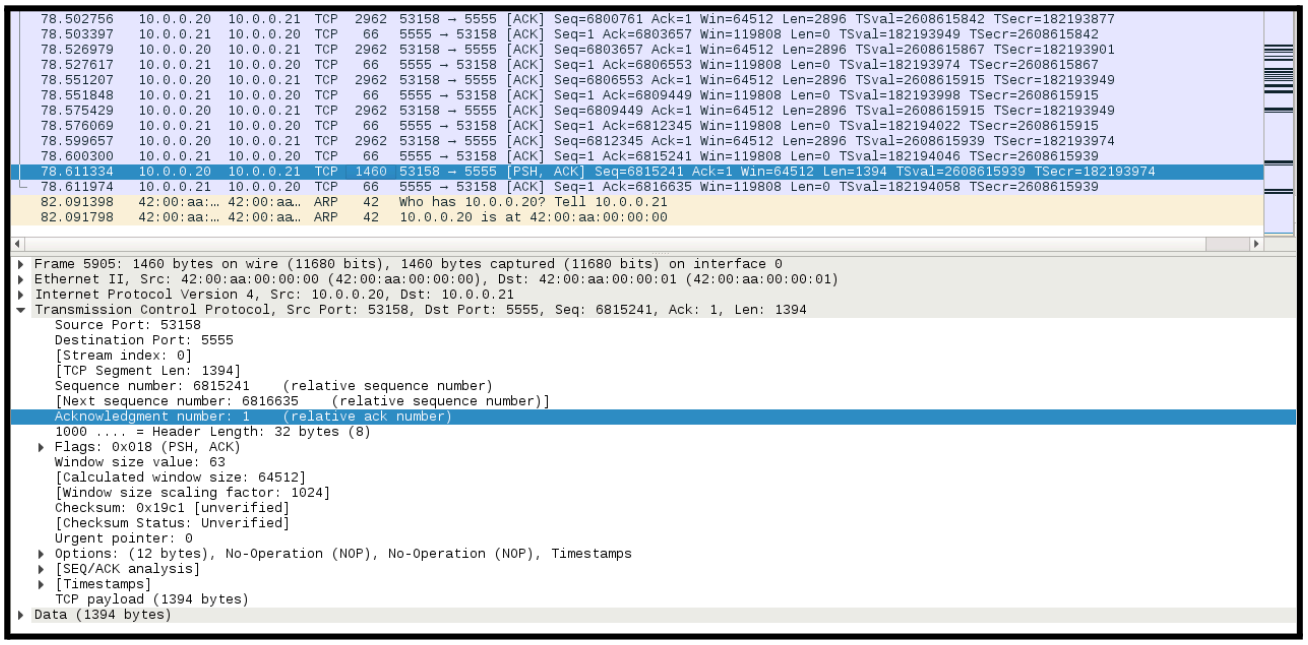
Calcule e mostre o procedimento de cálculo do tamanho do arquivo pela análise dos pacotes? Qual é a maneira mais fácil?

Apresente os cálculos ou descreva a maneira de obtenção do valor. Dica: observe o primeiro e o último número de sequência e faça uma correlação com o tamanho do arquivo.

6816634 bytes / 1514 bytes = 4502,4002642 pacotes;

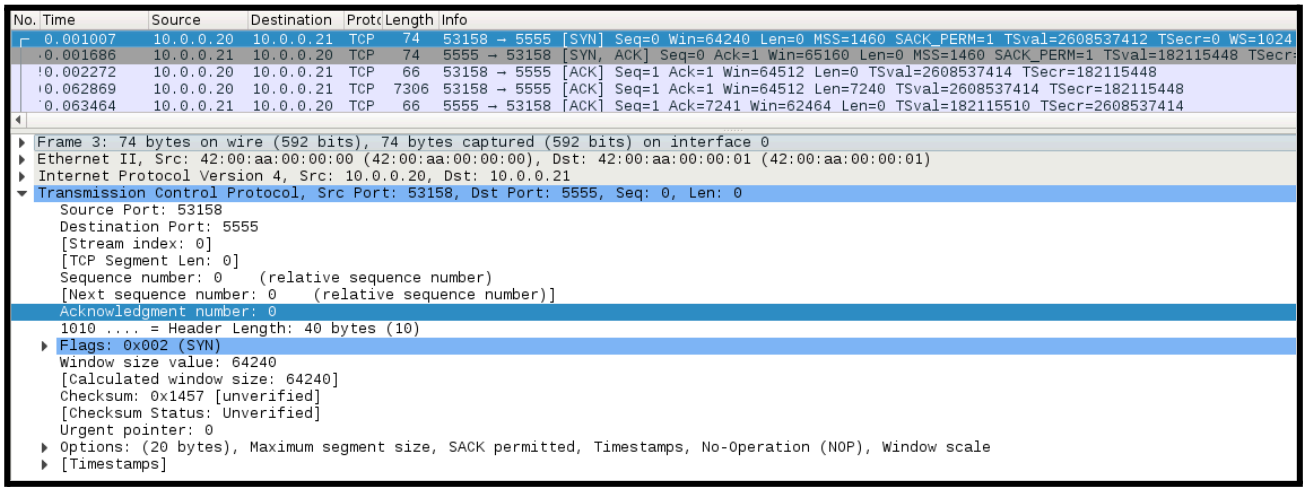
Ou seja, 4502 pacotes e mais 1 pacote com o comprimento de: 1460 bytes.

Qual é o tamanho do último segmento de dados recebido? Perceba que ele é diferente dos demais, que vem "cheios".



Comprimento de 1460 bytes (explicação na questão anterior).

Apresente os segmentos do 3-way handshake e analise os campos do cabeçalho, que os identificam. Estão de acordo com a norma apresentada na literatura (em sala de aula)?



Sim, o cliente estabelece previamente o three-way handshake, e então, inicia a comunicação de dados através do ACK no tempo (0.062889).

Transferência utilizando o protocolo UDP

Caso não tenha fechado o Imunes na Parte 1 vá direto para o Item 6.

1. Execute o Imunes.
2. Carregue o arquivo:

```
File >> Open >> TCPxUDP.imn
```

   - Será apresentada uma simples rede a ser utilizada no experimento, composta de 2 PC com um enlace de 1000 kbps.

3. Inicie a simulação da rede no Imunes:

```
Experiment >> Execute
```

   - Dica: para abrir um terminal de uma das máquinas da rede a ser simulada basta dar um duplo clique sobre a mesma.

4. Vamos simular uma taxa de perdas para tornar o ambiente de simulação um pouco mais desafiador. Para tal vamos utilizar o comando **tc** (*Traffic Control*) executando no terminal das duas máquinas:

```
tc qdisc replace dev eth0 root netem loss 10%
```

Verifique o tamanho do arquivo recebido. Ele é igual ao arquivo original? E quanto tempo levou para transmiti-lo?

```
root@Receptor:/# ls -l
total 7628
-rw-r--r--   1 root root 6816634 Oct 17 12:59 arquivoTCP
-rw-r--r--   1 root root  917504 Oct 17 13:55 arquivoUDP
drwxr-xr-x   1 root root    4096 May 15  2019 bin
drwxr-xr-x   2 root root    4096 Feb  3  2019 boot
-rw-r--r--   1 root root     114 Oct 17 12:52 boot.conf
drwxr-xr-x  15 root root    3580 Oct 17 12:52 dev
drwxr-xr-x   1 root root    4096 Oct 17 12:52 etc
```

Comprimento total: 6816634
Comprimento recebido via UDP: 917504
Comprimento restante: 5899130

Analisando a captura de pacotes do WireShark responda:

Qual é o identificador do primeiro e do último pacote? Existe?

Não existe, pois é UDP.

É possível calcular o tamanho do arquivo pela análise dos pacotes? É mais fácil ou difícil que no caso da transferência via TCP?

Sim, é possível através da soma de todos os lengths de cada pacote recebido.

Compare as transferências feitas com os protocolos TCP e UDP em relação, principalmente, ao tempo gasto para transmitir o arquivo e a integridade de dados. O que eles têm em comum?

São protocolos da camada de transporte;
Os dois protocolos tem porta de origem / destino;
Os dois protocolos possuem checksum;

Que diferenças lhe pareceram mais pronunciadas?

O protocolo TCP é orientado a conexão, então tem confiabilidade na recepção de pacotes. O UDP encaminha os pacotes com mais agilidade.

Como isso deve afetar as aplicações que usam esses protocolos?

As aplicações voltadas para a transmissão de arquivos (visando apenas a camada de transporte), devem encaminha-los via TCP.

Tráfego de voz e vídeo, podem ser encaminhado via UDP, streaming de vídeo da mesma maneira.