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

DESVENDANDO TCP

Arthur Cadore Matuella Barcella

TAREFA:

Desvendando o TCP - Número de Sequência, Controle de Erros, Transmissão *Full-Duplex*

Objetivos

- Verificar alguns mecanismos do protocolo TCP:
  - Controle de Erros: Significado de Número de Sequência, ACK;
  - Controle de Fluxo: Significado do campo Windows Size; Funcionamento do controle de fluxo;
  - Transmissão Full-Duplex.

Configuração do Laboratório

1. O roteiro será executado sobre máquinas virtuais, através do uso do Imunes.
2. Abra um terminal e baixe o aquivo de configuração da rede a ser utilizada e um arquivo auxiliar dos experimentos:

```
cd ~
wget -4 http://docente.ifsc.edu.br/odilson/RED29004/TCP_Num_Seq_Erro.imn
wget -4 http://docente.ifsc.edu.br/odilson/RED29004/arq30Bytes.txt
```

- **1 PARTE:**

No.	Time	Source	Destination	Protocol	Length	Info
	0.000000	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:01
	12.288002	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:00
	13.779656	42:00:aa:…	Broadcast	ARP	42	Who has 10.0.0.21? Tell 10.0.0.20
	13.779684	42:00:aa:…	42:00:aa…	ARP	42	10.0.0.21 is at 42:00:aa:00:00:01
	13.779691	10.0.0.20	10.0.0.21	TCP	74	53968 → 5555 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3989390915 TSecr=0 WS=1024
	13.779724	10.0.0.21	10.0.0.20	TCP	74	5555 → 53968 [SYN, ACK] Seq=0 Ack=1 Win=10 Len=0 MSS=1460 SACK_PERM=1 TSval=1653822008 TSecr=3989390915
	13.779753	10.0.0.20	10.0.0.21	TCP	66	53968 → 5555 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=3989390915 TSecr=1653822008
	13.984000	10.0.0.20	10.0.0.21	TCP	76	53968 → 5555 [PSH, ACK] Seq=1 Ack=1 Win=64512 Len=10 TSval=3989391119 TSecr=1653822008
	13.984048	10.0.0.21	10.0.0.20	TCP	66	[TCP ZeroWindow] 5555 → 53968 [ACK] Seq=1 Ack=11 Win=0 Len=0 TSval=1653822212 TSecr=3989391119
	13.984125	10.0.0.21	10.0.0.20	TCP	66	[TCP Window Update] 5555 → 53968 [ACK] Seq=1 Ack=11 Win=1024 Len=0 TSval=1653822212 TSecr=3989391119
	13.984146	10.0.0.20	10.0.0.21	TCP	86	53968 → 5555 [PSH, ACK] Seq=11 Ack=1 Win=64512 Len=20 TSval=3989391119 TSecr=1653822212
	13.984250	10.0.0.21	10.0.0.20	TCP	66	5555 → 53968 [ACK] Seq=1 Ack=31 Win=1024 Len=0 TSval=1653822213 TSecr=3989391119
	36.416270	10.0.0.20	10.0.0.21	TCP	66	53968 → 5555 [FIN, ACK] Seq=31 Ack=1 Win=64512 Len=0 TSval=3989413551 TSecr=1653822213
	36.416356	10.0.0.21	10.0.0.20	TCP	66	5555 → 53968 [FIN, ACK] Seq=1 Ack=32 Win=1024 Len=0 TSval=1653844645 TSecr=3989413551
	36.416395	10.0.0.20	10.0.0.21	TCP	66	53968 → 5555 [ACK] Seq=32 Ack=2 Win=64512 Len=0 TSval=3989413552 TSecr=1653844645
	120.831989	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:01

Após desabilitar a função "relative sequence number":

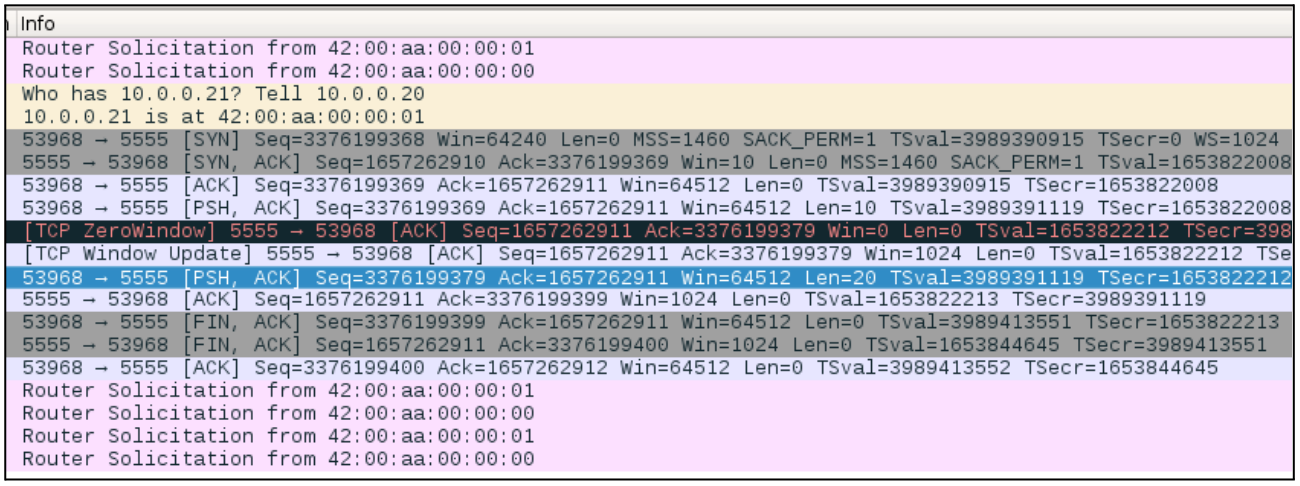
No.	Time	Source	Destination	Protocol	Length	Info
	0.000000	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:01
	12.288002	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:00
	13.779656	42:00:aa:…	Broadcast	ARP	42	Who has 10.0.0.21? Tell 10.0.0.20
	13.779684	42:00:aa:…	42:00:aa…	ARP	42	10.0.0.21 is at 42:00:aa:00:00:01
	13.779691	10.0.0.20	10.0.0.21	TCP	74	53968 → 5555 [SYN] Seq=3376199368 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3989390915 TSecr=0 WS=1024
	13.779724	10.0.0.21	10.0.0.20	TCP	74	5555 → 53968 [SYN, ACK] Seq=1657262910 Ack=3376199369 Win=10 Len=0 MSS=1460 SACK_PERM=1 TSval=1653822008
	13.779753	10.0.0.20	10.0.0.21	TCP	66	53968 → 5555 [ACK] Seq=3376199369 Ack=1657262911 Win=64512 Len=0 TSval=3989390915 TSecr=1653822008
	13.984000	10.0.0.20	10.0.0.21	TCP	76	53968 → 5555 [PSH, ACK] Seq=3376199369 Ack=1657262911 Win=64512 Len=10 TSval=3989391119 TSecr=1653822008
	13.984048	10.0.0.21	10.0.0.20	TCP	66	[TCP ZeroWindow] 5555 → 53968 [ACK] Seq=1657262911 Ack=3376199379 Win=0 Len=0 TSval=1653822212 TSecr=398
	13.984125	10.0.0.21	10.0.0.20	TCP	66	[TCP Window Update] 5555 → 53968 [ACK] Seq=1657262911 Ack=3376199379 Win=1024 Len=0 TSval=1653822212 TSe
	13.984146	10.0.0.20	10.0.0.21	TCP	86	53968 → 5555 [PSH, ACK] Seq=3376199379 Ack=1657262911 Win=64512 Len=20 TSval=3989391119 TSecr=1653822212
	13.984250	10.0.0.21	10.0.0.20	TCP	66	5555 → 53968 [ACK] Seq=1657262911 Ack=3376199399 Win=1024 Len=0 TSval=1653822213 TSecr=3989391119
	36.416270	10.0.0.20	10.0.0.21	TCP	66	53968 → 5555 [FIN, ACK] Seq=3376199399 Ack=1657262911 Win=64512 Len=0 TSval=3989413551 TSecr=1653822213
	36.416356	10.0.0.21	10.0.0.20	TCP	66	5555 → 53968 [FIN, ACK] Seq=1657262911 Ack=3376199400 Win=1024 Len=0 TSval=1653844645 TSecr=3989413551
	36.416395	10.0.0.20	10.0.0.21	TCP	66	53968 → 5555 [ACK] Seq=3376199400 Ack=1657262912 Win=64512 Len=0 TSval=3989413552 TSecr=1653844645
	120.831989	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:01
	151.555968	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:00
	380.928002	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:01
	446.464002	fe80::400…	ff02::2	ICMPv6	70	Router Solicitation from 42:00:aa:00:00:00

1) Qual o número de sequência normalizado pelo Wireshark de cada segmento de dados transmitido (do Transmissor para o Receptor) e qual o significado do número de reconhecimento em cada um deles?

Info
Router Solicitation from 42:00:aa:00:00:01
Router Solicitation from 42:00:aa:00:00:00
Who has 10.0.0.21? Tell 10.0.0.20
10.0.0.21 is at 42:00:aa:00:00:01
53968 → 5555 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3989390915 TSecr=0 WS=1024
5555 → 53968 [SYN, ACK] Seq=0 Ack=1 Win=10 Len=0 MSS=1460 SACK_PERM=1 TSval=1653822008 TSecr=3989390915
53968 → 5555 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=3989390915 TSecr=1653822008
53968 → 5555 [PSH, ACK] Seq=1 Ack=1 Win=64512 Len=10 TSval=3989391119 TSecr=1653822008
[TCP ZeroWindow] 5555 → 53968 [ACK] Seq=1 Ack=11 Win=0 Len=0 TSval=1653822212 TSecr=3989391119
[TCP Window Update] 5555 → 53968 [ACK] Seq=1 Ack=11 Win=1024 Len=0 TSval=1653822212 TSecr=3989391119
53968 → 5555 [PSH, ACK] Seq=11 Ack=1 Win=64512 Len=20 TSval=3989391119 TSecr=1653822212
5555 → 53968 [ACK] Seq=1 Ack=31 Win=1024 Len=0 TSval=1653822213 TSecr=3989391119
53968 → 5555 [FIN, ACK] Seq=31 Ack=1 Win=64512 Len=0 TSval=3989413551 TSecr=1653822213
5555 → 53968 [FIN, ACK] Seq=1 Ack=32 Win=1024 Len=0 TSval=1653844645 TSecr=3989413551
53968 → 5555 [ACK] Seq=32 Ack=2 Win=64512 Len=0 TSval=3989413552 TSecr=1653844645
Router Solicitation from 42:00:aa:00:00:01

O reconhecimento é o primeiro byte enviado no pacote correspondente ao sequence number.

2) Qual o número de sequência real de cada segmento de dados transmitido (do Transmissor para o Receptor) e qual o significado do número de reconhecimento em cada um deles?



3) Como foi reconhecido cada segmento enviado? É igual ao número de sequência ou é um número acima? Justifique.

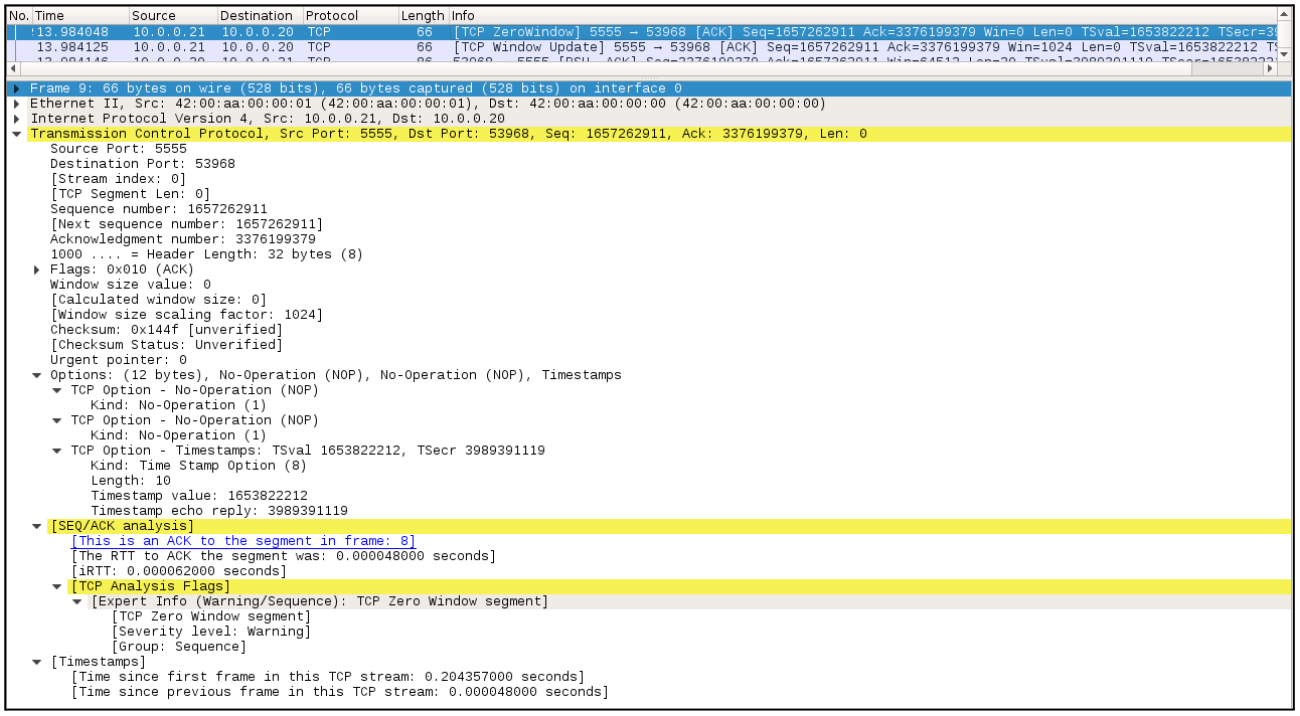
R: A partir do pacote de ACK, é igual ao número do último byte recebido + 1.

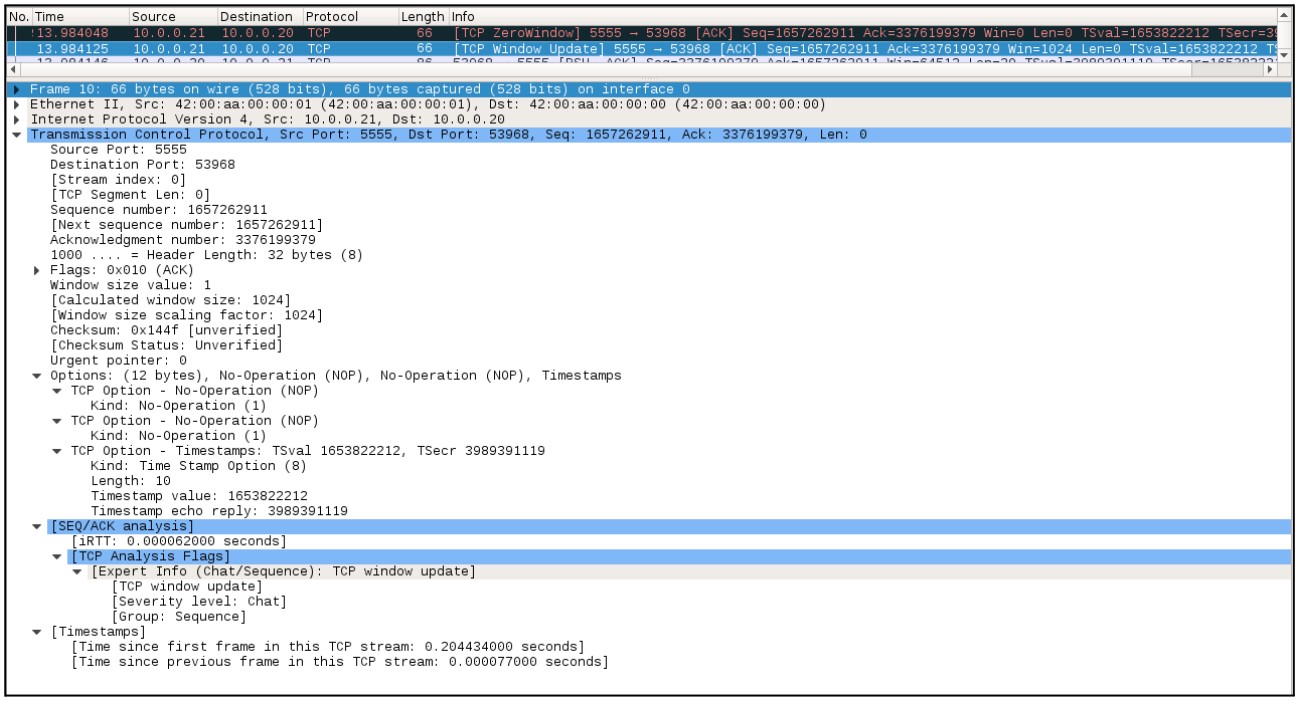
4) Qual o significado das mensagens, inseridas pelo Wireshark, "TCP ZeroWindow" e "TCP Window Update"?

O TCP ZeroWindow é enviado quando o buffer do receptor sobrecarrega, ele então solicita que o transmissor pare de transmitir.

O TCP Window Update é enviado a partir do momento em que o receptor processa os pacotes do buffer (após ter enviado um TCP ZeroWindow).

Abaixo estão as mensagens trocadas nos dois segmentos:

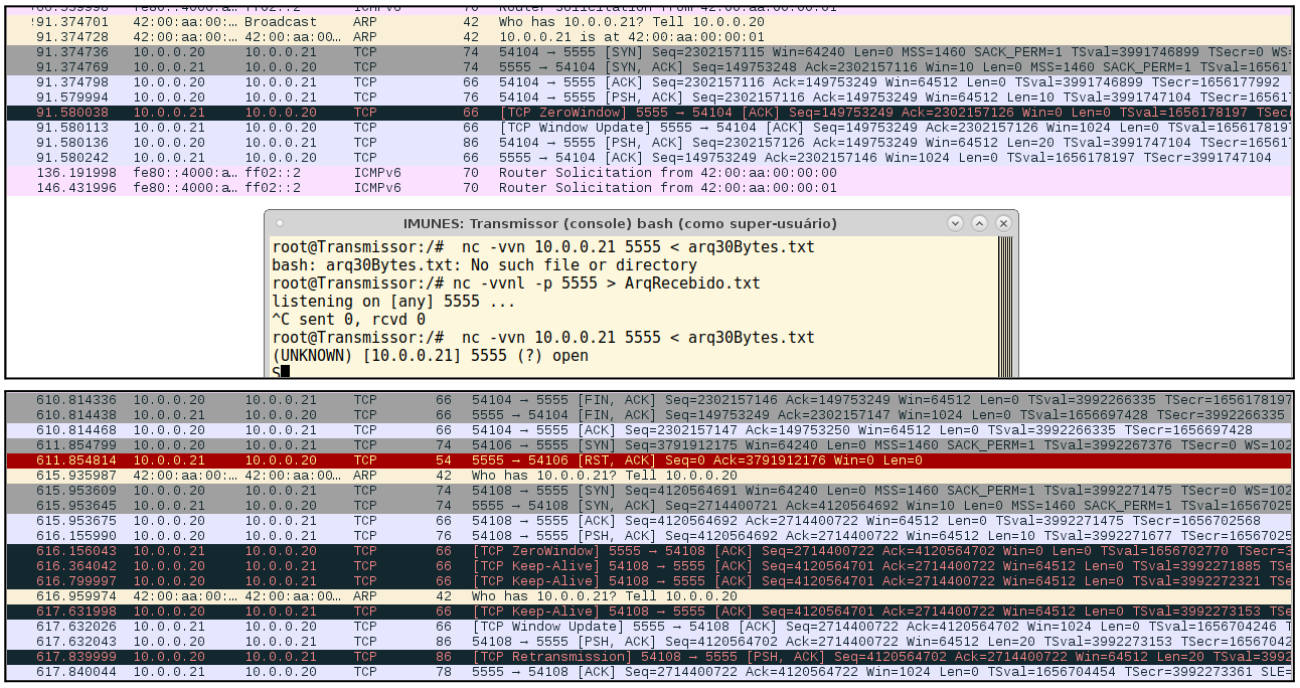

No. Time Source Destination Protocol Length Info
13.984048 10.0.0.21 10.0.0.20 TCP 66 [TCP ZeroWindow] 5555 → 53968 [ACK] Seq=1657262911 Ack=3376199379 Win=0 Len=0 TSval=1653822212 TSecr=3
13.984125 10.0.0.21 10.0.0.20 TCP 66 [TCP Window Update] 5555 → 53968 [ACK] Seq=1657262911 Ack=3376199379 Win=1024 Len=0 TSval=1653822212 T
Frame 9: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: 42:00:aa:00:00:01 (42:00:aa:00:00:01), Dst: 42:00:aa:00:00:00 (42:00:aa:00:00:00)
Internet Protocol Version 4, Src: 10.0.0.21, Dst: 10.0.0.20
Transmission Control Protocol, Src Port: 5555, Dst Port: 53968, Seq: 1657262911, Ack: 3376199379, Len: 0
Source Port: 5555
Destination Port: 53968
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1657262911
[Next sequence number: 1657262911]
Acknowledgment number: 3376199379
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window size value: 0
[Calculated window size: 0]
[Window size scaling factor: 1024]
Checksum: 0x144f [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
TCP Option - No-Operation (NOP)
Kind: No-Operation (1)
TCP Option - No-Operation (NOP)
Kind: No-Operation (1)
TCP Option - Timestamps: TSval 1653822212, TSecr 3989391119
Kind: Time Stamp Option (8)
Length: 10
Timestamp value: 1653822212
Timestamp echo reply: 3989391119
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 8]
[The RTT to ACK the segment was: 0.000048000 seconds]
[iRTT: 0.000062000 seconds]
[TCP Analysis Flags]
[Expert Info (Warning/Sequence): TCP Zero Window segment]
[TCP Zero Window segment]
[Severity level: Warning]
[Group: Sequence]
[Timestamps]
[Time since first frame in this TCP stream: 0.204357000 seconds]
[Time since previous frame in this TCP stream: 0.000048000 seconds]



5) Qual a relação entre os campos "Len=", "Seq=", "Ack=", "Win=" e o tamanho do segmento de dados?

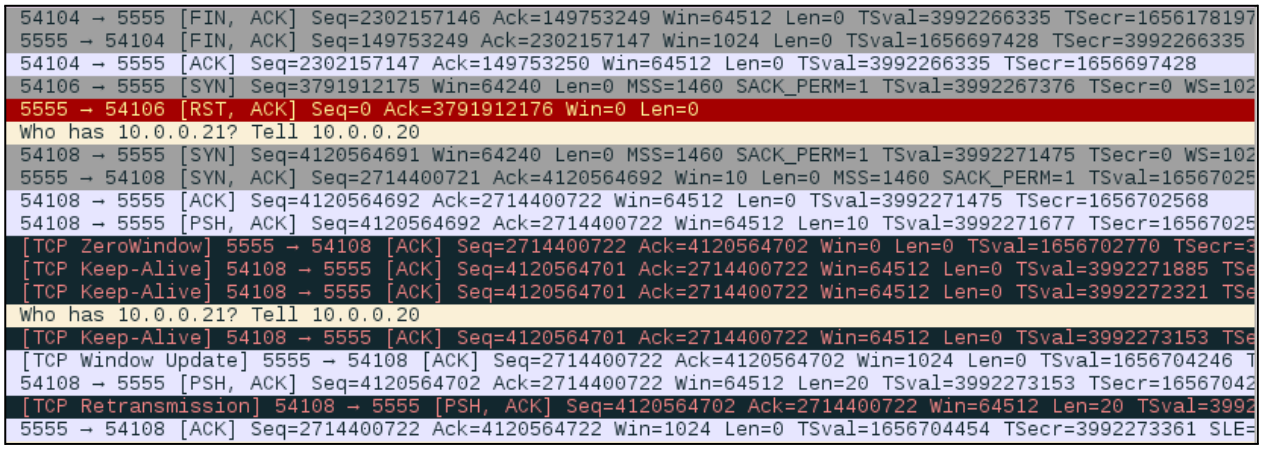
- Length: comprimento do pacote;
- Sequence: sequência do pacote enviado (primeiro byte do pacote enviado);
- Acknowledgement: sequência do próximo byte a ser enviado (próximo byte a ser enviado pelo transmissor).
- Window: é o tamanho da janela de receptor (buffer do receptor);

- **PARTE 2:**



1) Qual o número de sequência (normalizado pelo Wireshark) de cada segmento de dados transmitido (de PC1 para PC2) e qual o significado do número de reconhecimento em cada um deles?

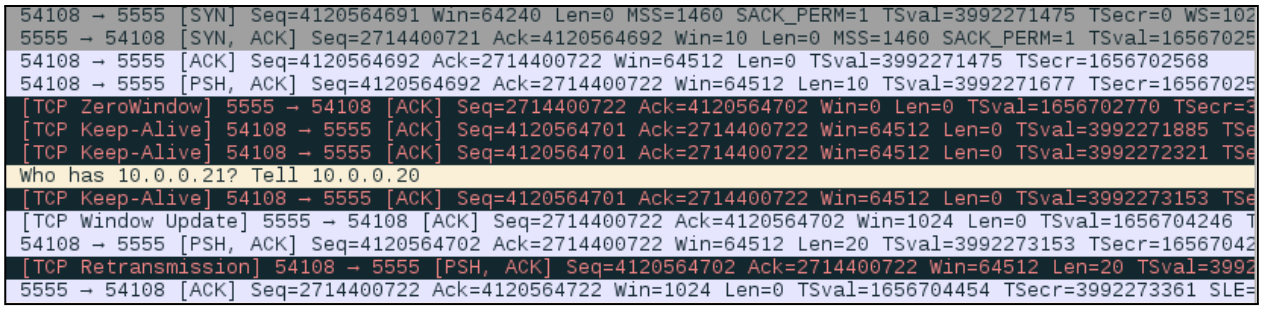
R): Segue na imagem abaixo:



2) Como foi reconhecido cada segmento enviado?

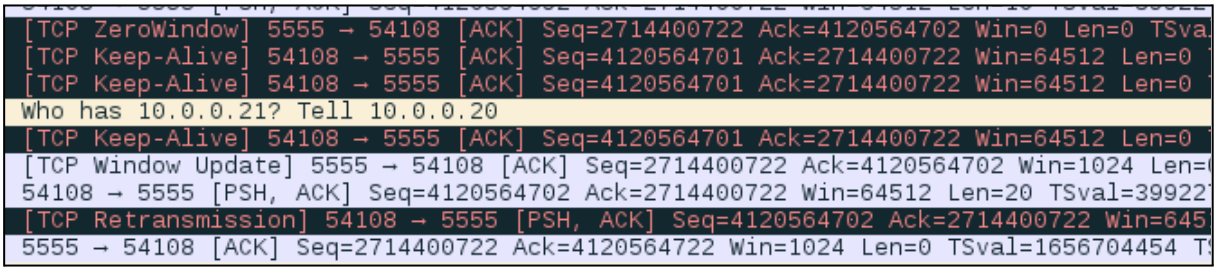
Através do ACK de cada pacote correspondente.

3) Houve perda de pacotes? Como você identificou isso?



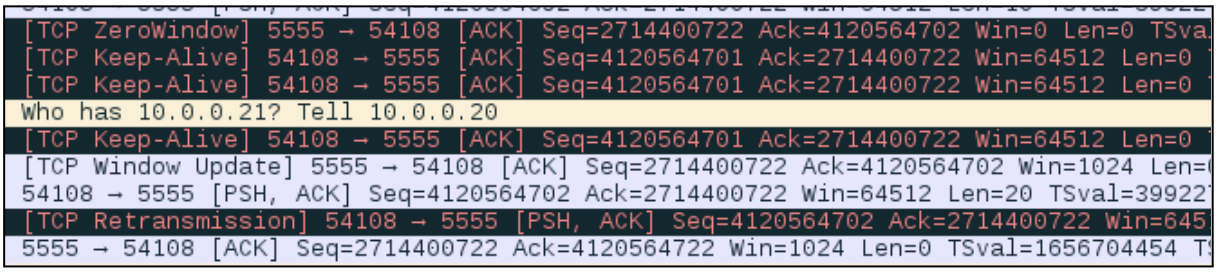
Penúltimo pacote da captura, identifica um TCP transmission por perda de pacote.

4) Os pacotes perdidos foram retransmitidos? Justifique.



R: Sim, acima é possível notar que o pacote foi retransmitido após o envio do TCP Retransmission

5) Qual o significado da mensagem, inserida pelo Wireshark, "TCP Retransmission"? Como você justificaria uma perda de segmento sem acesso a essa informação?



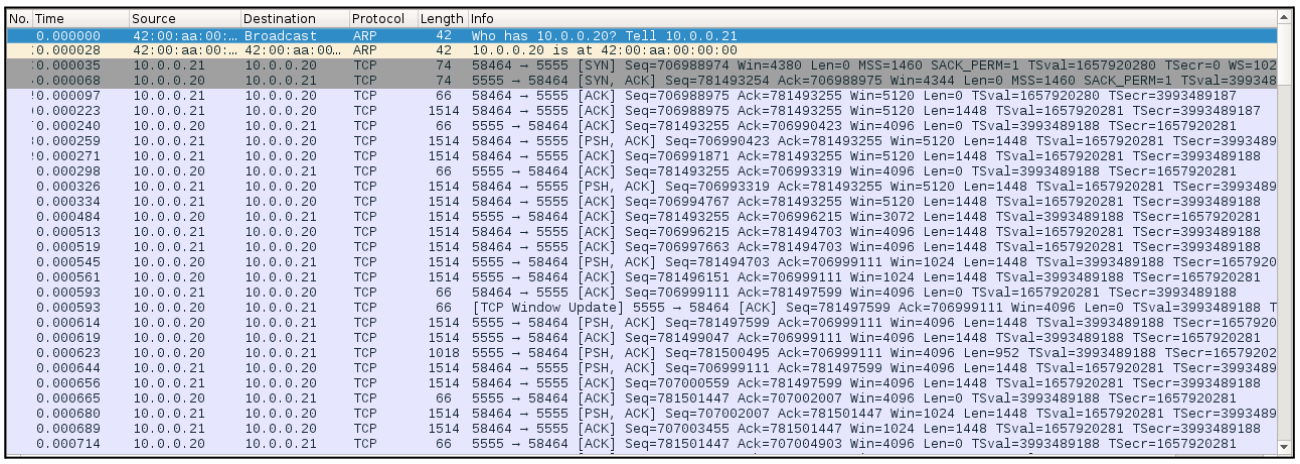
R: Sim, acima é possível notar que o pacote foi retransmitido após o envio do TCP Retransmission

Poderia ser utilizado também os números de ACK transmitidos, caso um deles seja repetido significa que um pacote foi perdido ou chegou fora de ordem.

6) Qual o significado das cores diferenciadas, inseridas pelo Wireshark, nos diversos segmentos apresentados?

R: Pacotes com erro ou atípicos são destacados pelo wireshark.

- **PARTE 3:**



No.	Time	Source	Destination	Protocol	Length	Info
	0.000000	42:00:aa:00:…	Broadcast	ARP	42	Who has 10.0.0.20? Tell 10.0.0.21
	0.000028	42:00:aa:00:…	42:00:aa:00…	ARP	42	10.0.0.20 is at 42:00:aa:00:00:00
	0.000035	10.0.0.21	10.0.0.20	TCP	74	58464 → 5555 [SYN] Seq=706988974 Win=4380 Len=0 MSS=1460 SACK_PERM=1 TSval=1657920280 TSecr=0 WS=102
	0.000068	10.0.0.20	10.0.0.21	TCP	74	5555 → 58464 [SYN, ACK] Seq=781493254 Ack=706988975 Win=4344 Len=0 MSS=1460 SACK_PERM=1 TSval=399348
	0.000097	10.0.0.21	10.0.0.20	TCP	66	58464 → 5555 [ACK] Seq=706988975 Ack=781493255 Win=5120 Len=0 TSval=1657920280 TSecr=3993489187
	0.000223	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=706988975 Ack=781493255 Win=5120 Len=1448 TSval=1657920281 TSecr=3993489187
	0.000240	10.0.0.20	10.0.0.21	TCP	66	5555 → 58464 [ACK] Seq=781493255 Ack=706990423 Win=4096 Len=0 TSval=3993489188 TSecr=1657920281
	0.000259	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [PSH, ACK] Seq=706990423 Ack=781493255 Win=5120 Len=1448 TSval=1657920281 TSecr=3993489
	0.000271	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=706991871 Ack=781493255 Win=5120 Len=1448 TSval=1657920281 TSecr=3993489188
	0.000298	10.0.0.20	10.0.0.21	TCP	66	5555 → 58464 [ACK] Seq=781493255 Ack=706993319 Win=4096 Len=0 TSval=3993489188 TSecr=1657920281
	0.000326	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [PSH, ACK] Seq=706993319 Ack=781493255 Win=5120 Len=1448 TSval=1657920281 TSecr=3993489
	0.000334	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=706994767 Ack=781493255 Win=5120 Len=1448 TSval=1657920281 TSecr=3993489188
	0.000484	10.0.0.20	10.0.0.21	TCP	1514	5555 → 58464 [ACK] Seq=781493255 Ack=706996215 Win=3072 Len=1448 TSval=3993489188 TSecr=1657920281
	0.000513	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=706996215 Ack=781494703 Win=4096 Len=1448 TSval=1657920281 TSecr=3993489188
	0.000519	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=706997663 Ack=781494703 Win=4096 Len=1448 TSval=1657920281 TSecr=3993489188
	0.000545	10.0.0.20	10.0.0.21	TCP	1514	5555 → 58464 [PSH, ACK] Seq=781494703 Ack=706999111 Win=1024 Len=1448 TSval=3993489188 TSecr=1657920
	0.000561	10.0.0.20	10.0.0.21	TCP	1514	5555 → 58464 [ACK] Seq=781496151 Ack=706999111 Win=1024 Len=1448 TSval=3993489188 TSecr=1657920281
	0.000593	10.0.0.21	10.0.0.20	TCP	66	58464 → 5555 [ACK] Seq=706999111 Ack=781497599 Win=4096 Len=0 TSval=1657920281 TSecr=3993489188
	0.000593	10.0.0.20	10.0.0.21	TCP	66	[TCP Window Update] 5555 → 58464 [ACK] Seq=781497599 Ack=706999111 Win=4096 Len=0 TSval=3993489188 T
	0.000614	10.0.0.20	10.0.0.21	TCP	1514	5555 → 58464 [PSH, ACK] Seq=781497599 Ack=706999111 Win=4096 Len=1448 TSval=3993489188 TSecr=1657920
	0.000619	10.0.0.20	10.0.0.21	TCP	1514	5555 → 58464 [ACK] Seq=781499047 Ack=706999111 Win=4096 Len=1448 TSval=3993489188 TSecr=1657920281
	0.000623	10.0.0.20	10.0.0.21	TCP	1018	5555 → 58464 [PSH, ACK] Seq=781500495 Ack=706999111 Win=4096 Len=952 TSval=3993489188 TSecr=16579202
	0.000644	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [PSH, ACK] Seq=706999111 Ack=781497599 Win=4096 Len=1448 TSval=1657920281 TSecr=3993489
	0.000656	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=707000559 Ack=781497599 Win=4096 Len=1448 TSval=1657920281 TSecr=3993489188
	0.000665	10.0.0.20	10.0.0.21	TCP	66	5555 → 58464 [ACK] Seq=781501447 Ack=707002007 Win=4096 Len=0 TSval=3993489188 TSecr=1657920281
	0.000680	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [PSH, ACK] Seq=707002007 Ack=781501447 Win=1024 Len=1448 TSval=1657920281 TSecr=3993489
	0.000689	10.0.0.21	10.0.0.20	TCP	1514	58464 → 5555 [ACK] Seq=707003455 Ack=781501447 Win=1024 Len=1448 TSval=1657920281 TSecr=3993489188
	0.000714	10.0.0.20	10.0.0.21	TCP	66	5555 → 58464 [ACK] Seq=781501447 Ack=707004903 Win=4096 Len=0 TSval=3993489188 TSecr=1657920281

1) Os arquivos foram corretamente trocados entre as duas máquinas? Dica: Responda observando o conteúdo dos arquivos, que são exclusivos e bem criativos :).

```
root@Receptor:/# ls -lh
total 156K
-rw-r--r--   1 root root  42K Oct 24 14:10 Arq_recebido.rx
-rw-r--r--   1 root root  38K Oct 24 14:10 Cliente.tx
drwxr-xr-x   1 root root 4.0K May 15  2019 bin
drwxr-xr-x   2 root root 4.0K Feb  3  2019 boot
-rw-r--r--   1 root root  114 Oct 24 14:09 boot.conf
drwxr-xr-x  15 root root 3.5K Oct 24 14:09 dev
drwxr-xr-x   1 root root 4.0K Oct 24 14:09 etc
drwxr-xr-x   2 root root 4.0K Feb  3  2019 home
drwxr-xr-x   1 root root 4.0K Mar 26  2019 lib
drwxr-xr-x   2 root root 4.0K Mar 26  2019 lib64
```

```
root@Transmissor:/# ls -lh
total 156K
-rw-r--r--   1 root root  38K Oct 24 14:10 Arq_recebido.rx
-rw-r--r--   1 root root  42K Oct 24 14:09 Servidor.tx
drwxr-xr-x   1 root root 4.0K May 15  2019 bin
drwxr-xr-x   2 root root 4.0K Feb  3  2019 boot
-rw-r--r--   1 root root  114 Oct 24 14:09 boot.conf
drwxr-xr-x  15 root root 3.5K Oct 24 14:09 dev
drwxr-xr-x   1 root root 4.0K Oct 24 14:09 etc
drwxr-xr-x   2 root root 4.0K Feb  3  2019 home
drwxr-xr-x   1 root root 4.0K Mar 26  2019 lib
drwxr-xr-x   2 root root 4.0K Mar 26  2019 lib64
drwxr-xr-x   2 root root 4.0K Mar 26  2019 media
```

Sim, os dados passados foram corretamente, podemos notar pelo comprimento dos arquivos.

2) Onde pode ser observado a comunicação full-duplex? Obs.: Foque a análise nos segmentos que contém [PSH, ACK].

Nota-se através dos nos pacotes transmitidos com altos comprimentos no pacote e ACKs entregues pelo receptor

3) Qual é a relação entre os comandos no terminal tanto do cliente como do servidor com a comunicação full-duplex?

No terminal os comandos foram apenas para receber/enviar o arquivo de dados, a comunicação full-duplex é necessária para encaminhar as confirmações de recebimento.

4) Como os ACKs são propagados, em pacotes exclusivos ou de carona (piggyback) com os dados?

São programados através de carona, é possível notar isso através do ACK do pacote enviado (através da soma dos valores de (len + seq)).