

1. Introdução

O objetivo deste trabalho é analisar o tráfego de uma rede, identificando o estabelecimento da conexão, a transferência de dados e a finalização da conexão. Foi usada a aplicação Wireshark que analisa o tráfego de rede.

2. Descrição do Funcionamento

2.1 Estabelecimento de conexão

A verificação do estabelecimento de conexão é feito através de pacotes e flags. O usuário primeiramente envia uma flag **SYN** junto com o pacote TCP com o requerimento de estabelecer uma conexão. Se tudo sair como planejado o servidor responde com um **SYN + ACK**. Caso o cliente receba esse **SYN + ACK** ele envia para o servidor um **ACK** e assim estabelece a conexão.

2.2 Transferência de Dados

TCP/IP é um conjunto de protocolos que segue o modelo OSI, dentre os diversos protocolos contidos nele neste relatório usaremos apenas o HTTP e FTP.

2.3 Finalização da Conexão

Quando um dos agentes tomar a iniciativa de encerrar a conexão, o mesmo enviará um pacote TCP com a flag **FIN**. Se não houver erros, o outro lado deve confirmar o pedido com um **ACK** e em seguida enviará um **FIN** e após o requerente enviar um **ACK** a conexão é finalmente encerrada.

3. Desenvolvimento

Para realizar o trabalho, a URL www.netshoes.com.br foi monitorada, cujo IP é 23.77.40.240.

Foi utilizado o filtro “(tcp or http) and ip.addr ==23.77.40.240” para facilitar a visualização dos pacotes necessários para o trabalho.

3.1 Estabelecimento da Conexão

No.	Time	Source	Destination	Protocol	Length	Info
13	3.036017	192.168.25.14	23.77.40.240	TCP	66	58286 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
16	3.068313	23.77.40.240	192.168.25.14	TCP	66	80 → 58286 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1452 SACK_PERM=1 WS=32
17	3.068374	192.168.25.14	23.77.40.240	TCP	54	58286 → 80 [ACK] Seq=1 Ack=1 Win=66560 Len=0

Figura 1: Pacotes para realizar a conexão

O primeiro passo é um pedido inicial do IP 192.168.25.14 para realizar a conexão, usando a flag SYN com um valor aleatório. O protocolo utilizado é o TCP e a porta utilizada pela Camada de Aplicação é a 58286.

Wireshark · Packet 13 · wireshark_D72A5C7B-DC51-422D-961F-696074E02501_20171026195540_a08728

> Frame 13: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0

> Ethernet II, Src: AsustekC_65:78:97 (88:d7:f6:65:78:97), Dst: Sagemcom_d6:4f:b3 (4c:17:eb:d6:4f:b3)

▼ Internet Protocol Version 4, Src: 192.168.25.14, Dst: 23.77.40.240

- 0100 = Version: 4
- 0101 = Header Length: 20 bytes (5)
- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 52
- Identification: 0x7c38 (31800)
- > Flags: 0x02 (Don't Fragment)
- Fragment offset: 0
- Time to live: 128
- Protocol: TCP (6)
- Header checksum: 0x6498 [validation disabled]
- [Header checksum status: Unverified]
- Source: 192.168.25.14
- Destination: 23.77.40.240
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]

▼ Transmission Control Protocol, Src Port: 58286, Dst Port: 80, Seq: 0, Len: 0

- Source Port: 58286
- Destination Port: 80
- [Stream index: 0]
- [TCP Segment Len: 0]
- Sequence number: 0 (relative sequence number)
- Acknowledgment number: 0
- 1000 = Header Length: 32 bytes (8)
- > Flags: 0x002 (SYN)
- Window size value: 64240
- [Calculated window size: 64240]
- Checksum: 0x8d6d [unverified]
- [Checksum Status: Unverified]
- Urgent pointer: 0
- > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted

Figura 2: Pacote do primeiro passo

O segundo passo seria o pedido de confirmação do servidor, nesse caso 104.88.122.217, enviando SYN+ ACK para o cliente (192.68.25.14), cujo valor de ACK é o valor recebido (que seria 0) acrescentando uma unidade. Valor de ACK = 1. Valor de SYN = 0.

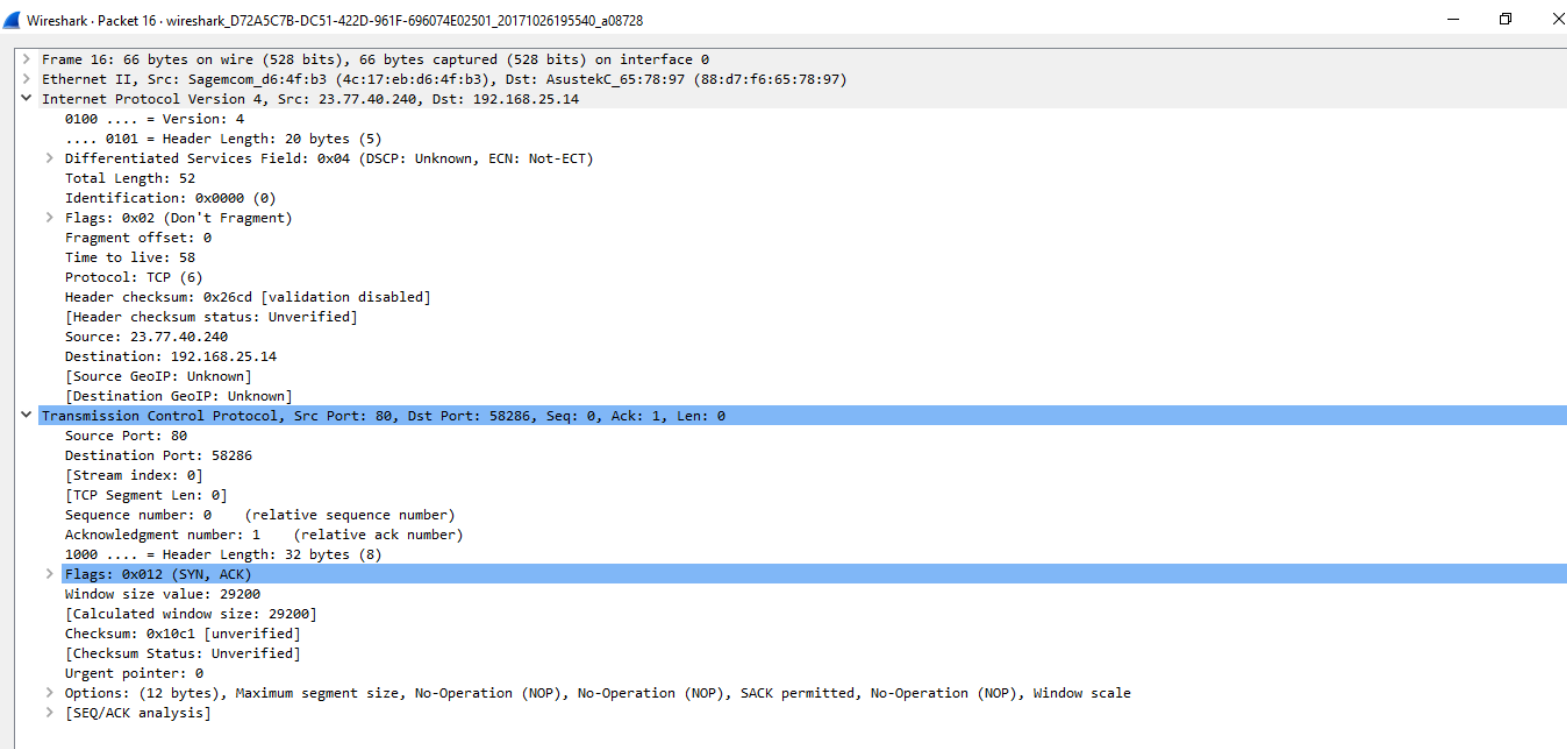


Figura 3: Pacote do Segundo Passo.

O terceiro passo é estabelecer finalmente a conexão entre servidor e cliente. O cliente retorna um ACK, com SYN de mesmo número recebido pelo servidor.

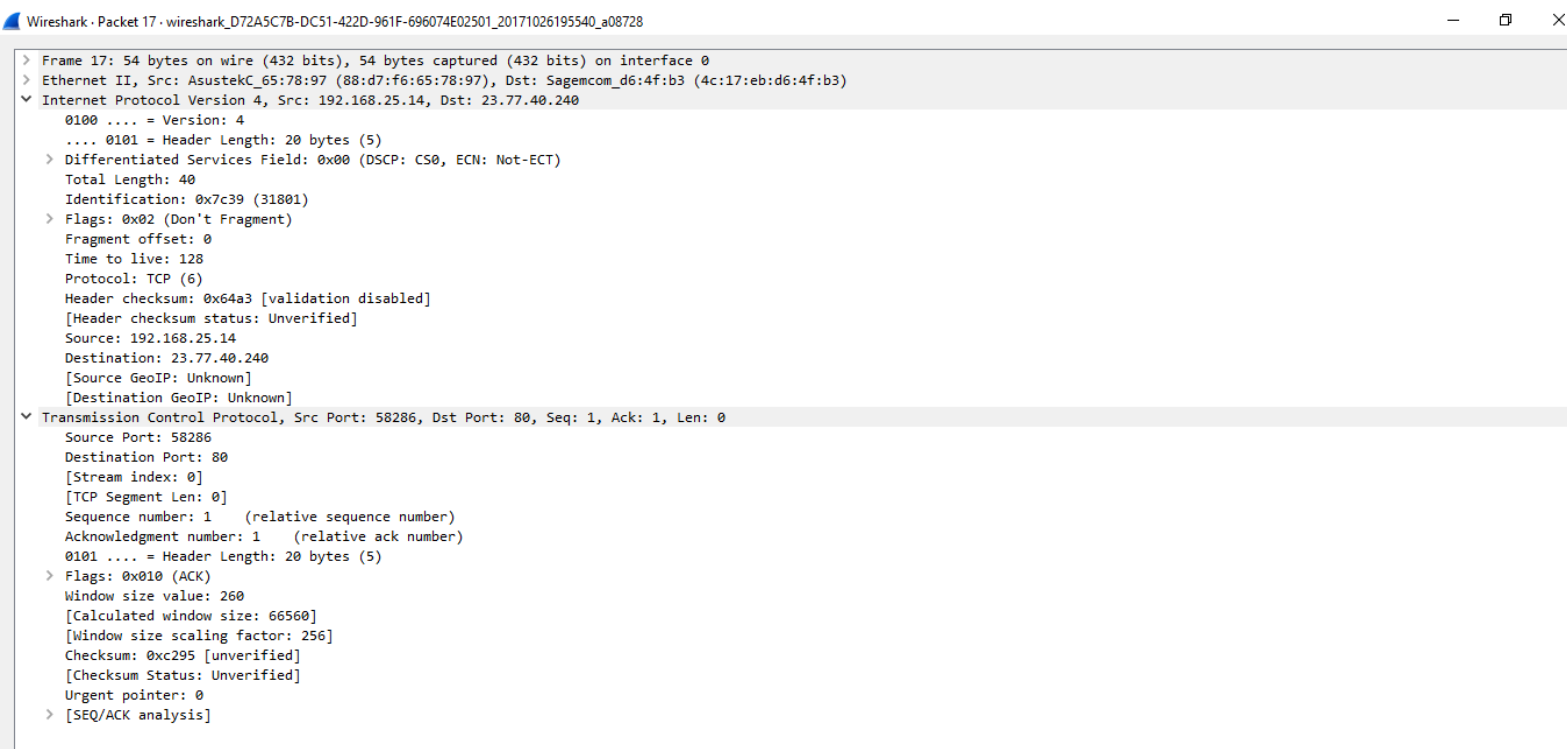


Figura 4: Pacote do Terceiro Passo.

3.2 Transferência de dados

Quando a comunicação se torna estável entre o navegador e o servidor, o primeiro pacote HTTP é enviado e é possível ver a informação enviada.

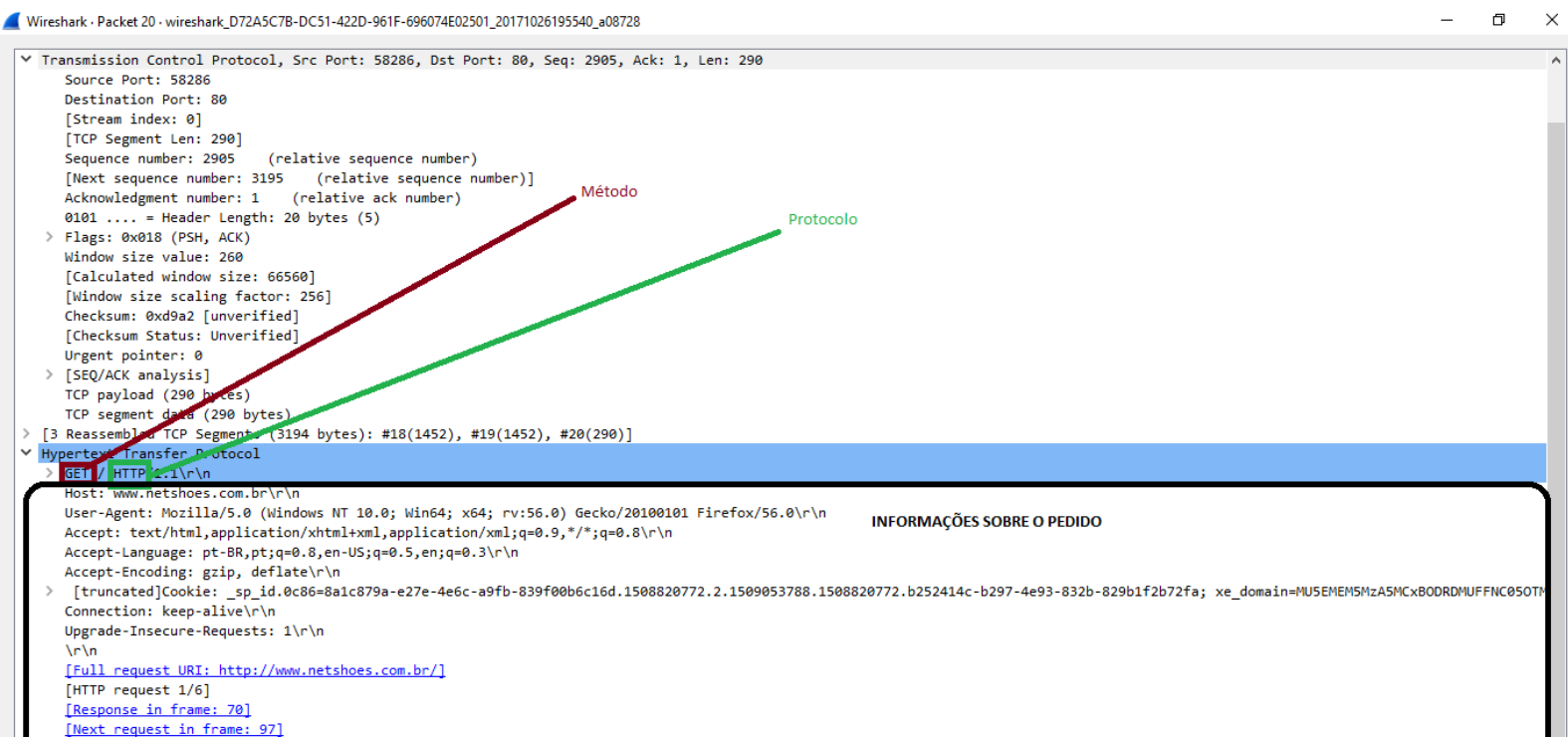


Figura 5: Informações do pacote enviado.

3.3 Finalizando a conexão

Para a conexão ser finalizada são enviados 3 pacotes.

2188	8.578763	192.168.25.14	23.77.40.240	TCP	54	58299 → 80	[FIN, ACK] Seq=1 Ack=1 Win=66560 Len=0
2206	8.609484	23.77.40.240	192.168.25.14	TCP	60	80 → 58296	[FIN, ACK] Seq=1 Ack=2 Win=29216 Len=0
2207	8.609547	192.168.25.14	23.77.40.240	TCP	54	58296 → 80	[ACK] Seq=2 Ack=2 Win=66560 Len=0

Figura 6: Pacotes que finalizam a conexão.

No último pacote com dados, em que o servidor envia, é ativada a flag [FIN].

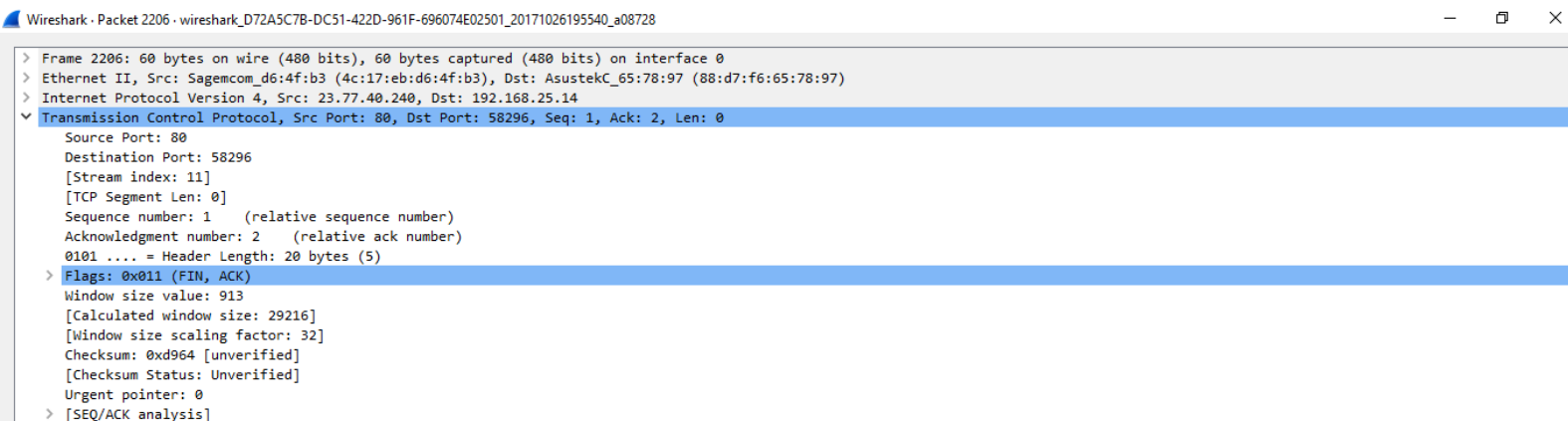


Figura 7: Primeiro passo para a finalização da conexão.

Em seguida, o usuário confirma ao servidor o pedido de término e verifica se o servidor finalizará também.

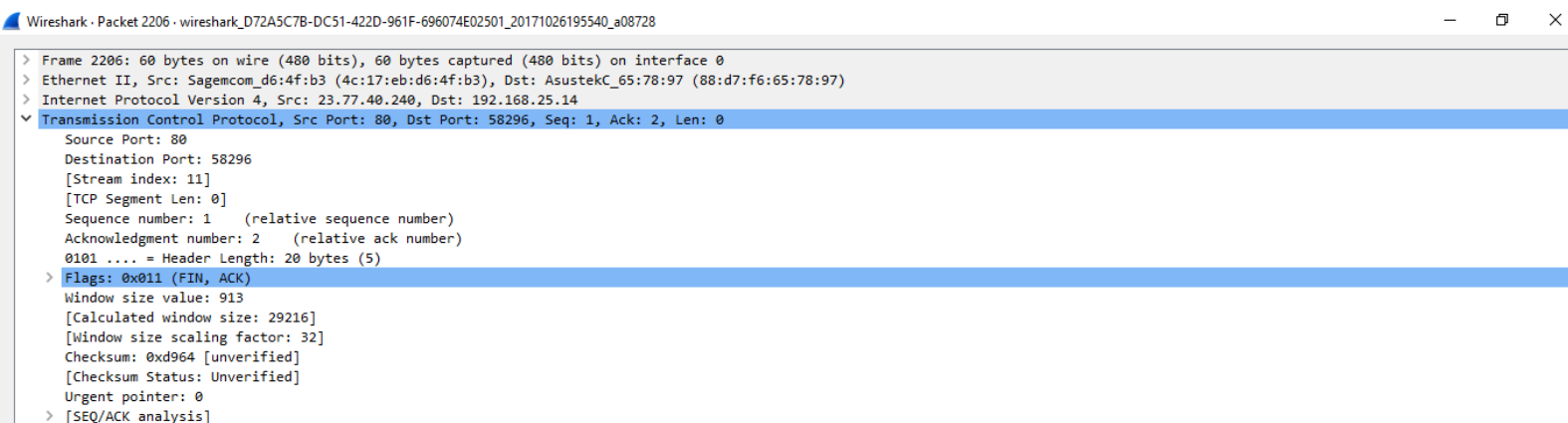


Figura 8: Segundo e terceiro passos para a finalização da conexão.

Por fim, o servidor confirma o pedido e a conexão é desfeita.

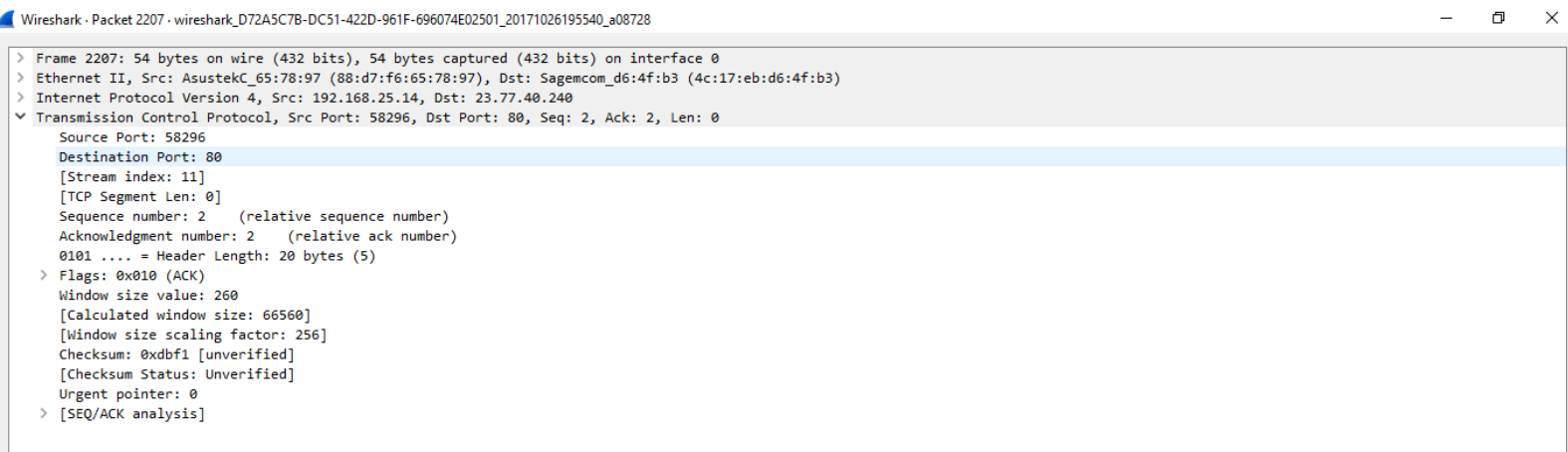


Figura 9: Quarto passo para a finalização da conexão.

4. Referências Bibliográficas

- [1] <https://pt.wikipedia.org/wiki/TCP/IP>
- [2] [https://pt.wikipedia.org/wiki/Hypertext Transfer Protocol](https://pt.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
- [3] <https://pplware.sapo.pt/microsoft/windows/aprenda-a-usar-o-sniffer-wireshark-parte-i/>
- [4] <https://www.wireshark.org/>
- [5] <https://nandovieira.com.br/entendendo-um-pouco-mais-sobre-o-protocolo-http>
- [6] <https://www.citisystems.com.br/protocolo-tcp-ip/>