Universidade Presbiteriana Mackenzie
Faculdade de Computação e Informática

Redes de computares

Profa. Dra. Cláudia Josimar Abrão de Araújo

Atividade elaborada pelo Prof. Dr. Bruno da Silva Rodrigues

Analisando uma transferência TCP do seu computador para um servidor remoto usando Wireshark

Introdução

Neste laboratório, investigaremos detalhadamente o comportamento do protocolo TCP. Faremos isso analisando um traço dos segmentos TCP enviados e recebidos ao transferir um arquivo de 150KB (contendo o texto de Lewis Carrol's Alice's Adventures in Wonderland) do seu computador para um servidor remoto. Estudaremos o uso da TCP de números de seqüência e reconhecimento para fornecer transferência confiável de dados; veremos o algoritmo de controle de congestionamento da TCP - início lento e evasão de congestionamento - em ação; e analisaremos o mecanismo de controle de fluxo anunciado pelo receptor da TCP. Também consideramos brevemente a configuração da conexão TCP e investigaremos o desempenho (throughput e round trip trip) da conexão TCP entre o seu computador e o servidor.

*Experiência de uso do software Wireshark – esta experiência foi proposta por Kurose**

Procedimento

1. *Abra o navegador e acesse: http://gaia.cs.umass.edu/ethereal-labs/alice.txt*
2. *Salve o texto do livro Alice no país das maravilhas num arquivo Alice.txt (ou copie e cole o texto num bloco de notas). 3. Abra o wireshark e inicie a captura de pacotes*
4. *Acesse o site: http://gaia.cs.umass.edu/ethereal-labs/TCP-etherealfile1.html*

Objetivos da atividade:

- Estudar o protocolo TCP usando wireshark analisando segmentação, *transferência confiável de dados e controle de congestionamento*

Bibliografias

KUROSE, J. F. e ROSS, K. W. Redes de Computadores e a Internet – Uma Nova Abordagem – Pearson

Internet Engineering Task Force. Disponível em: hps://www.ietf.org/rfc/rfc1035.txt

5. *Após acessar o site, selecione o arquivo alice.txt" para carregar o arquivo para o servidor gaia.cs.umass.edu*

6. *Uma vez que o arquivo foi carregado, uma breve mensagem de parabéns será exibida na janela do navegador. Pare a captura de pacotes do Wireshark.*

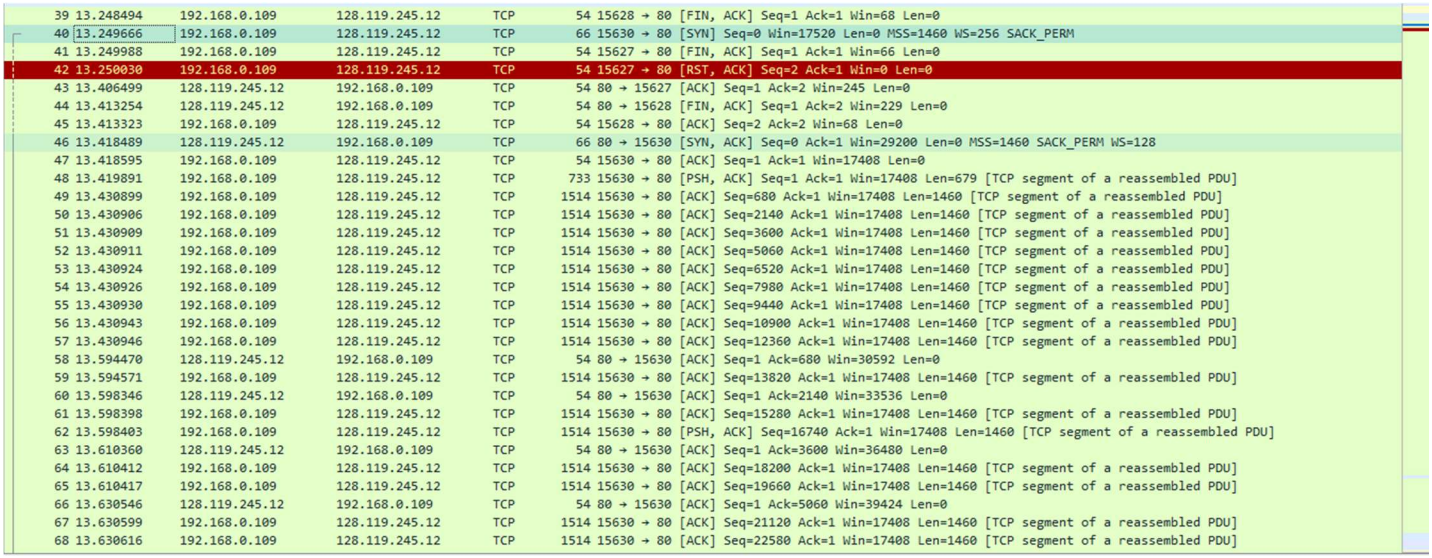
1. Responda as questões em negrito com fonte vermelha.

Sempre que possível, ao responder uma pergunta, você deve entregar uma impressão do (s) pacote (s) dentro do rastreamento que você usou para responder a pergunta

Após abrir o arquivo analise os pacotes e responda:

Questão 1 (1,5). *Localize a sequência de pacotes trocados entre o cliente e o servidor gaia e localize o Three-way handshake.*

a) *Apresente um print da tela onde aparecem os pacotes do Three-way Handshake*



39	13.248494	192.168.0.109	128.119.245.12	TCP	54	15628 → 80 [FIN, ACK] Seq=1 Ack=1 Win=68 Len=0
40	13.249666	192.168.0.109	128.119.245.12	TCP	66	15630 → 80 [SYN] Seq=0 Win=17520 Len=0 MSS=1460 WS=256 SACK_PERM
41	13.249988	192.168.0.109	128.119.245.12	TCP	54	15627 → 80 [FIN, ACK] Seq=1 Ack=1 Win=66 Len=0
42	13.250030	192.168.0.109	128.119.245.12	TCP	54	15627 → 80 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
43	13.406499	128.119.245.12	192.168.0.109	TCP	54	80 → 15627 [ACK] Seq=1 Ack=2 Win=245 Len=0
44	13.413254	128.119.245.12	192.168.0.109	TCP	54	80 → 15628 [FIN, ACK] Seq=1 Ack=2 Win=229 Len=0
45	13.413323	192.168.0.109	128.119.245.12	TCP	54	15628 → 80 [ACK] Seq=2 Ack=2 Win=68 Len=0
46	13.418489	128.119.245.12	192.168.0.109	TCP	66	80 → 15630 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM WS=128
47	13.418595	192.168.0.109	128.119.245.12	TCP	54	15630 → 80 [ACK] Seq=1 Ack=1 Win=17408 Len=0
48	13.419891	192.168.0.109	128.119.245.12	TCP	733	15630 → 80 [PSH, ACK] Seq=1 Ack=1 Win=17408 Len=679 [TCP segment of a reassembled PDU]
49	13.430899	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=680 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
50	13.430906	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=2140 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
51	13.430909	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=3600 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
52	13.430911	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=5060 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
53	13.430924	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=6520 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
54	13.430926	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=7980 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
55	13.430930	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=9440 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
56	13.430943	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=10900 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
57	13.430946	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=12360 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
58	13.594470	128.119.245.12	192.168.0.109	TCP	54	80 → 15630 [ACK] Seq=1 Ack=680 Win=30592 Len=0
59	13.594571	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=13820 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
60	13.598346	128.119.245.12	192.168.0.109	TCP	54	80 → 15630 [ACK] Seq=1 Ack=2140 Win=33536 Len=0
61	13.598398	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=15280 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
62	13.598403	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [PSH, ACK] Seq=16740 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
63	13.610360	128.119.245.12	192.168.0.109	TCP	54	80 → 15630 [ACK] Seq=1 Ack=3600 Win=36480 Len=0
64	13.610412	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=18200 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
65	13.610417	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=19660 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
66	13.630546	128.119.245.12	192.168.0.109	TCP	54	80 → 15630 [ACK] Seq=1 Ack=5060 Win=39424 Len=0
67	13.630599	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=21120 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]
68	13.630616	192.168.0.109	128.119.245.12	TCP	1514	15630 → 80 [ACK] Seq=22580 Ack=1 Win=17408 Len=1460 [TCP segment of a reassembled PDU]

b) *Explique o que acontece em cada uma das etapas do Three-way Handshake.*

SYN (Synchronize):

O cliente envia um pacote TCP com o bit SYN (Synchronize) definido como 1 e escolhe um número de sequência inicial (ISN - Initial Sequence Number).

O número de sequência inicial é usado para sincronizar a comunicação entre o cliente e o servidor, permitindo a identificação única de cada segmento de dados.

Este pacote é enviado ao servidor para iniciar a conexão e solicitar a sincronização.

ACK (Acknowledgment):

O cliente recebe o pacote SYN-ACK do servidor.

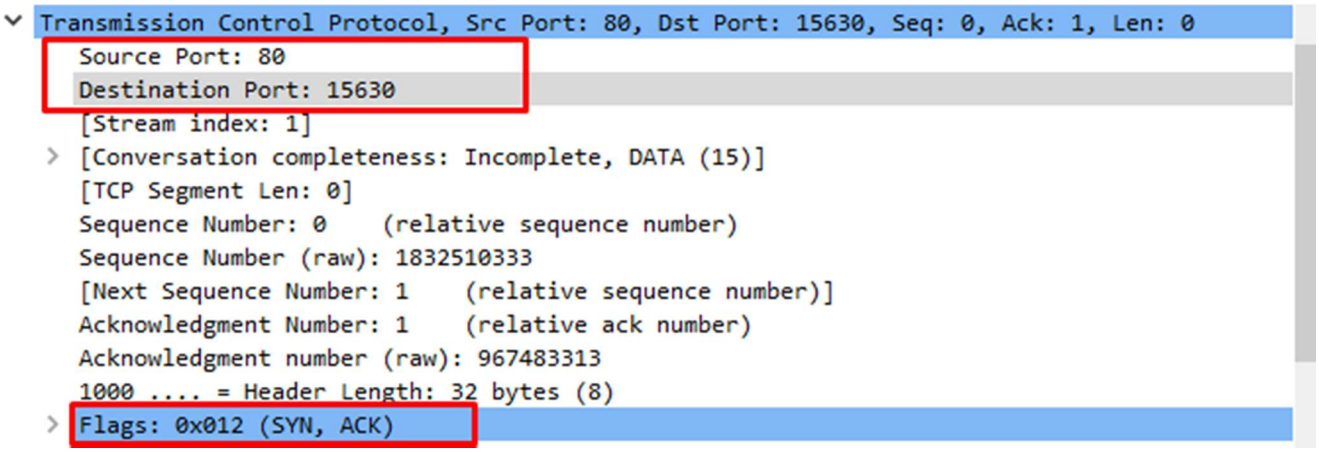
O cliente confirma a solicitação de sincronização do servidor enviando um pacote ACK.

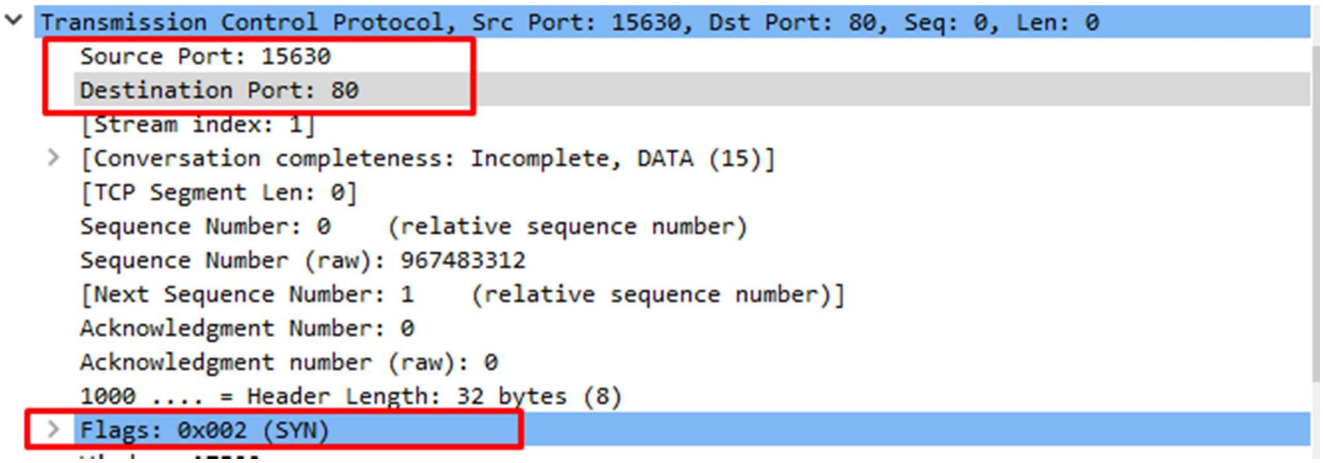
O cliente define o bit ACK como 1 e envia o número de sequência inicial do servidor mais um (o próximo número de sequência esperado pelo servidor).

Este pacote é enviado de volta ao servidor para confirmar a sincronização da conexão.

***Questão 2(1,5)**. Qual informação no cabeçalho do segmento identifica se o segmento é SYN ou SYNACK? Localize a informação dentro do cabeçalho TCP e apresente um print do local que serviu como base para sua resposta.*

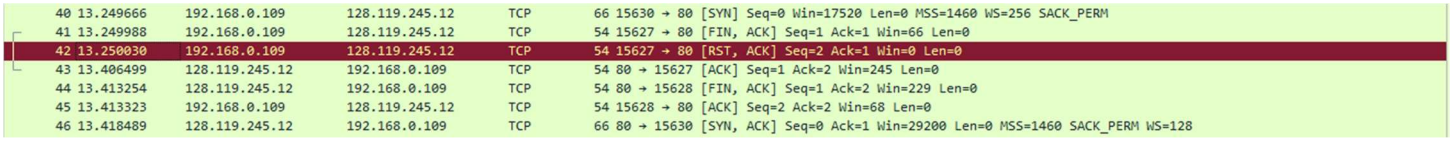
Pode se identificar pelas portas de servidor e cliente. No SYNACK ele vai da porta 80(cliente) para a porta 15630(a porta do servidor) enquanto o SYN ele vai da porta 15630(porta do servidor) para a porta 80(cliente).

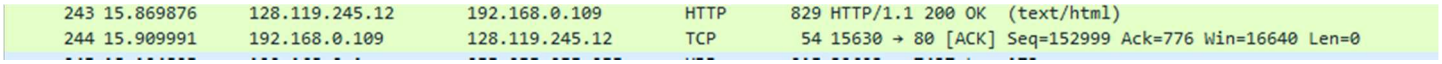




***Questão 3(2,0)**. Essa foi uma transmissão segura? Discorra sobre como você chegou a essa conclusão? Observe que segura não é a mesma coisa que confiável. Apresente a tela e **justifique sua resposta** com base na tela apresentada.*

Sim, foi uma transmissão segura. Pode se dizer que foi segura, pois, as etapas SYN, SYNACK e ACK for ocorrida sem nenhum problema, podendo dizer que a mensagem não foi perdida ou ocorreu nada de errado entre os processos.





***Questão 4(2,0)**. Qual o tamanho da janela de recepção do servidor gaia?*

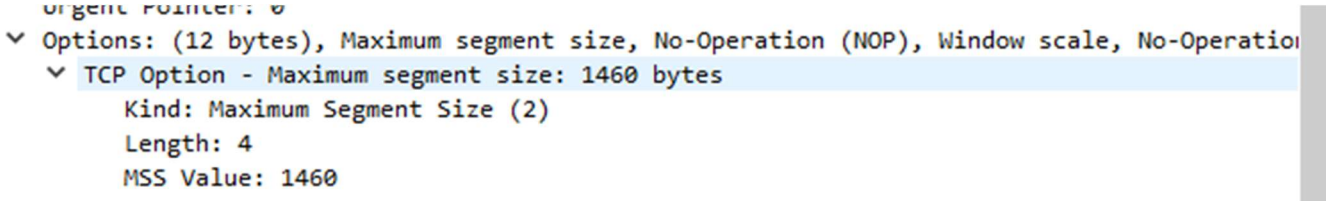
***a) (1,0)** Apresente a tela com um círculo de onde você tirou essa informação.*

***b) (1,0)** Qual a função da janela de recepção?*

Desempenha um papel fundamental no controle de fluxo em uma conexão TCP Sua função é permitir que o remetente (servidor) regule a quantidade de dados que pode enviar ao destinatário (cliente) antes de receber uma confirmação de recebimento

***Questão 5(1,5)**. O que é maximum segment size (MSS)? Qual o MSS nessa comunicação? Apresente a tela de onde você tirou essa informação*

***a) (0,5)** Apresente um print com o valor do MSS.*



***b) (1,0)** Qual a função do MSS?*

O MSS especifica o tamanho máximo de segmento que o receptor aceitará receber, e é um campo opcional que deve ser enviado apenas na transmissão de uma requisição de conversação.

Questão 6(1,5)**. Ao analisar o cabeçalho TCP temos a informação de iRTT (inicial RTT) e RTT, apresente o segmento onde o iRTT foi determinado. Compare com as respostas ACK e verifique se o RTT aumentou ou diminuiu. Apresente os valores e as telas onde as informações foram retiradas. **Justifique o motivo da variação entre os valores de RTT.

***Nota:** O Wireshark possui um recurso agradável que permite traçar o RTT para cada um dos segmentos TCP enviados. Selecione um segmento TCP na janela "listagem de pacotes capturados" que está sendo enviada do cliente para o servidor gaia.cs.umass.edu. Em seguida, selecione: **Estatísticas-> Gráfico de fluxo TCP> Gráfico de tempo de viagem**.*