

Lab 8 (Week 9)

TCP Connection Analysis

CAN201

Dr. Gordon Boateng and Dr. Fei Cheng

Outline

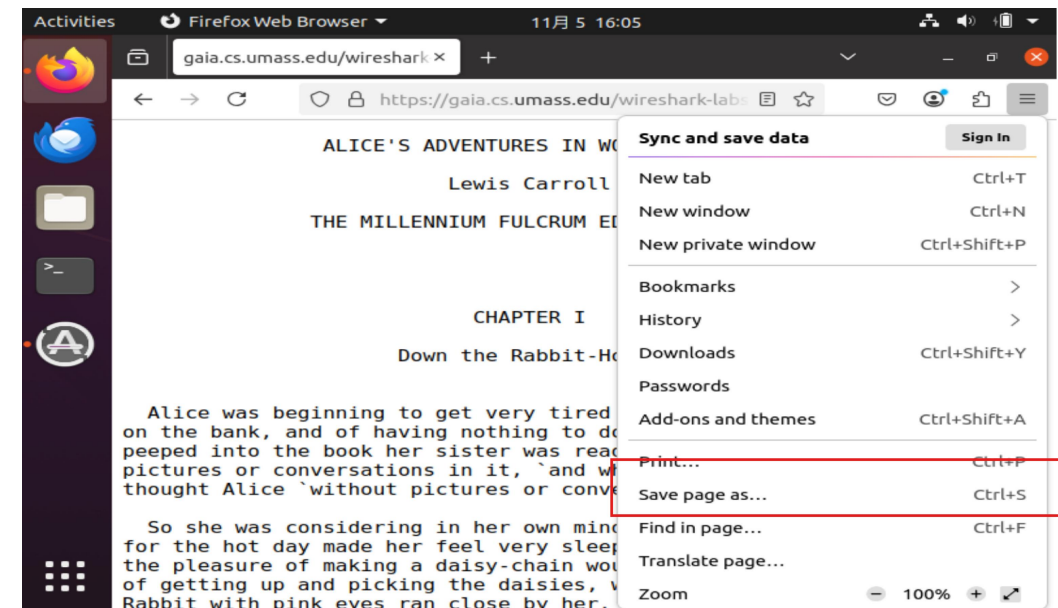
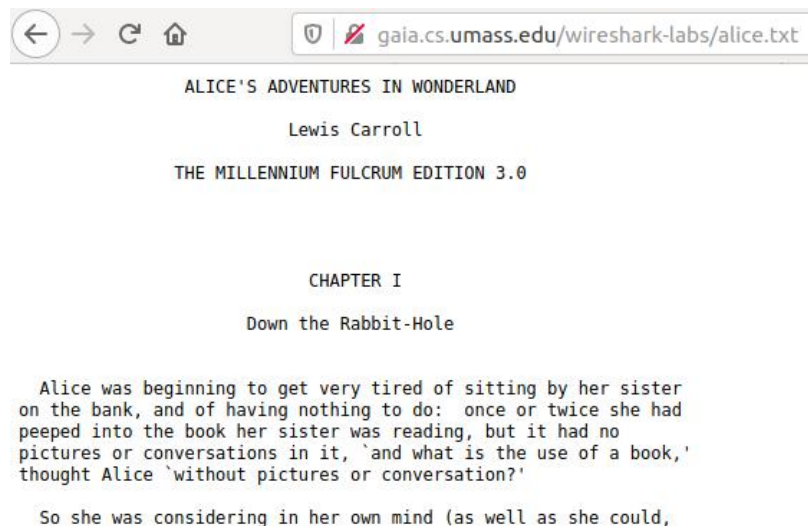
- TCP connection analysis
- Lab Exercise

TCP connection analysis

1. Analyze a trace of the TCP segments sent and received in transferring a 150KB file from your computer to a remote server.
2. We will need to use **Wireshark** to obtain a packet trace of the TCP transfer of a file from our computer to a remote server.
3. We will do so by accessing a Web page that will allow us to enter the name of a file stored on our computer, and then transfer the file to a Web server using the **HTTP POST** method while running Wireshark during this time to obtain the trace of TCP segments.

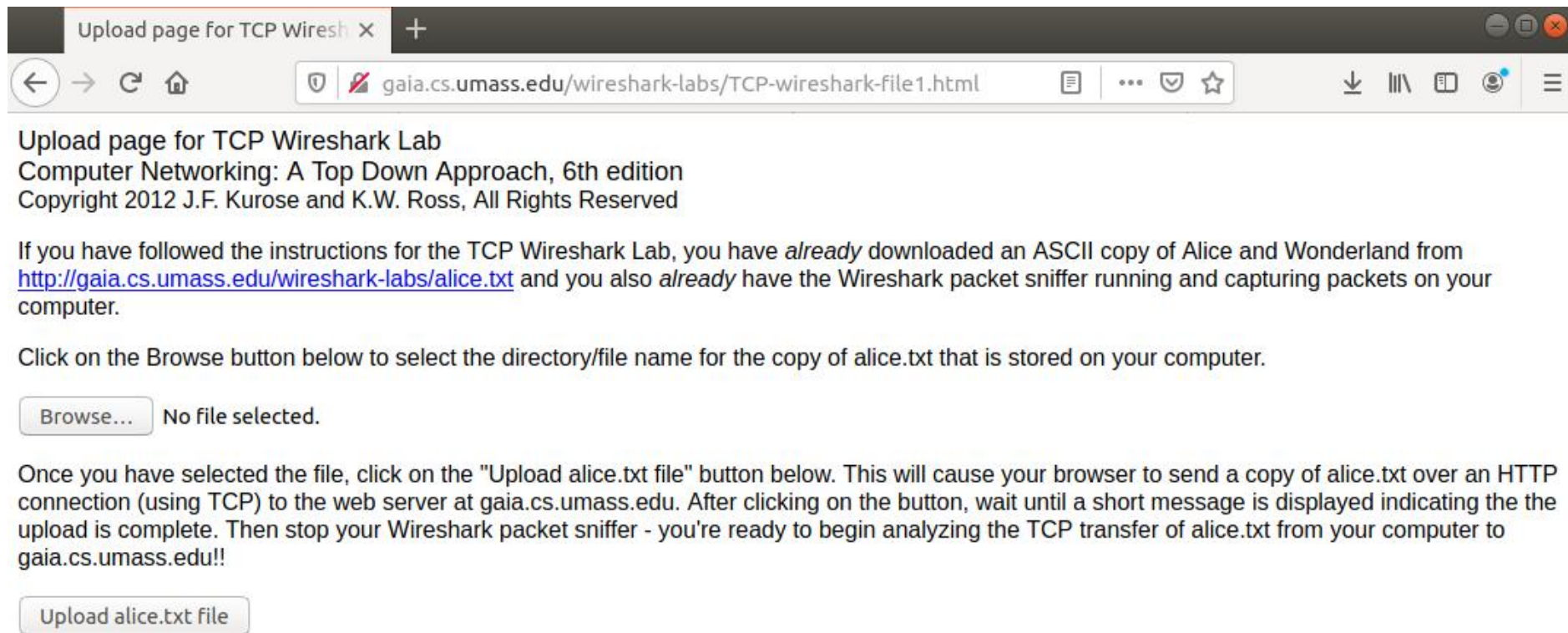
Test run for the TCP connection analysis

1. Start up your web browser. Go to <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and retrieve an ASCII copy of *Alice in Wonderland*. Store this file somewhere on your computer.
 - Note: Just save the page as alice.txt on your local computer (Ubuntu OS)



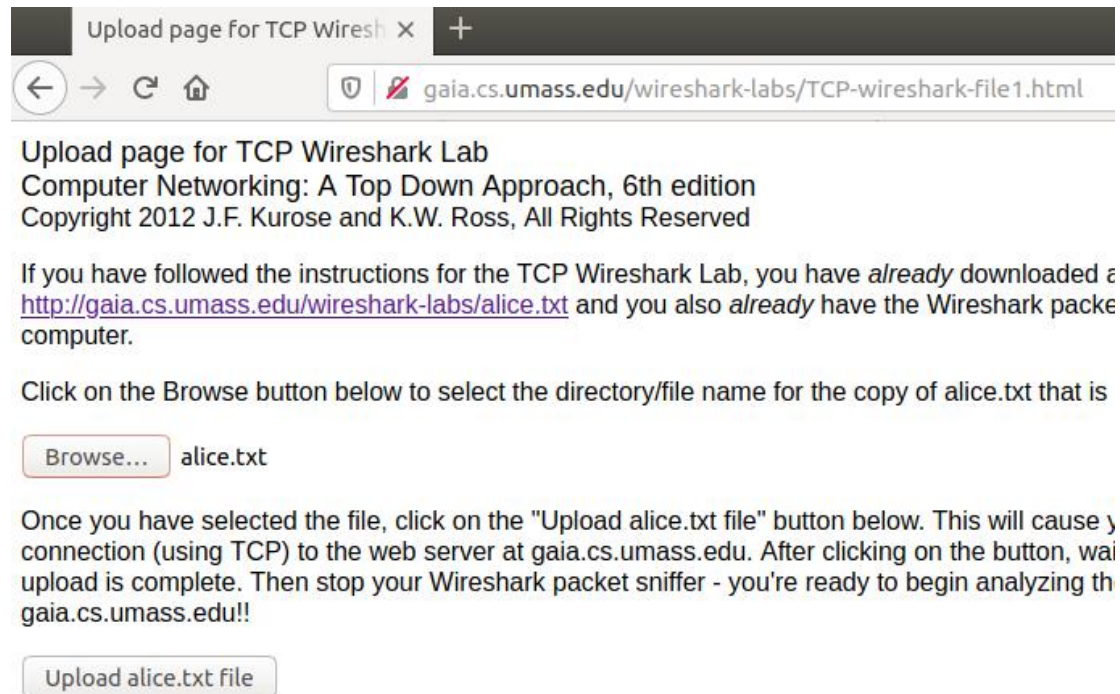
Test run for the TCP connection analysis

2. Next, go to <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>. You should see a screen like this (below)



Test run for the TCP connection analysis

3. Use the *Browse* button in this form to enter the name of the file (alice.txt) on your computer.
 - Note: Do not press the “*Upload alice.txt file*” button yet.



The screenshot shows a web browser window with the address bar displaying 'gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html'. The page content includes the title 'Upload page for TCP Wireshark Lab', the book title 'Computer Networking: A Top Down Approach, 6th edition', and the copyright notice 'Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved'. Below this, there is a paragraph of instructions: 'If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded a file named <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet capture file on your computer.' This is followed by the instruction 'Click on the Browse button below to select the directory/file name for the copy of alice.txt that is'. Below this text is a form with a 'Browse...' button and the text 'alice.txt'. At the bottom of the form is an 'Upload file' button.

Upload page for TCP Wireshark Lab
Computer Networking: A Top Down Approach, 6th edition
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded a file named <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet capture file on your computer.

Click on the Browse button below to select the directory/file name for the copy of alice.txt that is

Browse... alice.txt

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your computer to establish a TCP connection to the web server at gaia.cs.umass.edu. After clicking on the button, wait until the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the traffic on the connection to gaia.cs.umass.edu!!

Upload file

Test run for the TCP connection analysis

4. Now start up Wireshark and begin packet capture (*Capture->Start*) and then press *ENTER* on the *Wireshark* screen (we do not need to select any options here).
5. Returning to your browser, press the “*Upload alice.txt file*” button to upload the file to the `gaia.cs.umass.edu` server. Once the file has been uploaded, a short congratulatory message will be displayed in your browser window.
6. Stop Wireshark packet capture. Your Wireshark window should look similar to the window shown below.

Test run for the TCP connection analysis

No.	Time	Source	Destination	Protocol	Length	Info
208	18.497182758	10.0.2.9	128.119.245.12	TCP	74	59298 → 80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460
209	18.557937230	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0
210	18.557979294	10.0.2.9	128.119.245.12	TCP	54	59296 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
211	18.558499238	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=1 Ack=1 Win=42340 Len=2920
212	18.558527709	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=2921 Ack=1 Win=42340 Len=2920
213	18.558537730	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=5841 Ack=1 Win=42340 Len=2920
214	18.558678362	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=8761 Ack=1 Win=42340 Len=2920
215	18.558695855	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=11681 Ack=1 Win=42340 Len=2920
216	18.559006789	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=2921 Win=32768 Len=0
217	18.559022071	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=14601 Ack=1 Win=42340 Len=2920
218	18.559038039	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=17521 Ack=1 Win=42340 Len=2920
219	18.559133314	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=5841 Win=32768 Len=0
220	18.559141794	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=20441 Ack=1 Win=42340 Len=2920
221	18.559156489	10.0.2.9	128.119.245.12	TCP	2974	59296 → 80 [ACK] Seq=23361 Ack=1 Win=42340 Len=2920
222	18.559276628	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=8761 Win=32768 Len=0
223	18.559287559	10.0.2.9	128.119.245.12	TCP	1514	59296 → 80 [PSH, ACK] Seq=26281 Ack=1 Win=42340

```

▶ Frame 208: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_7e:1f:c1 (08:00:27:7e:1f:c1), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▶ Internet Protocol Version 4, Src: 10.0.2.9, Dst: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 59298, Dst Port: 80, Seq: 0, Len: 0

```

```
0000 52 54 00 12 35 00 08 00 27 7e 1f c1 08 00 45 00 RT<5... ' ~... E
0010 00 3c 99 81 40 00 40 06 1f ae 0a 00 02 09 80 77 < @ @ ... w
0020 f5 0c e7 a2 00 50 53 27 48 bf 00 00 00 00 a0 02 ... PS' H ...
0030 a5 64 81 bb 00 00 02 04 05 b4 04 02 08 0a e4 ac ... d ...
0040 1f 50 00 00 00 00 01 03 03 09 ... P ...
```


Test run for the TCP connection analysis

- To find the POST command, we will need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

No.	Time	Source	Destination	Protocol	Length	Info
333	19.610069807	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=135781 Win=32768 Len=
334	19.610091974	10.0.2.9	128.119.245.12	HTTP	2634	POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1
335	19.610316564	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=138701 Win=32768 Len=
336	19.610648849	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=141621 Win=32768 Len=
337	19.610802645	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=144541 Win=32768 Len=
338	19.611114812	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=147461 Win=32768 Len=
339	19.611237465	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=150381 Win=32768 Len=
340	19.611552327	128.119.245.12	10.0.2.9	TCP	60	80 → 59296 [ACK] Seq=1 Ack=152961 Win=32768 Len=
341	19.963978552	128.119.245.12	10.0.2.9	HTTP	831	HTTP/1.1 200 OK (text/html)
342	19.963998212	10.0.2.9	128.119.245.12	TCP	54	59296 → 80 [ACK] Seq=152961 Ack=778 Win=41958 Len=

▶ Frame 334: 2634 bytes on wire (21072 bits), 2634 bytes captured (21072 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_7e:1f:c1 (08:00:27:7e:1f:c1), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
 ▶ Internet Protocol Version 4, Src: 10.0.2.9, Dst: 128.119.245.12
 ▶ Transmission Control Protocol, Src Port: 59296, Dst Port: 80, Seq: 150381, Ack: 1, Len: 2580
 ▶ [72 Reassembled TCP Segments (152960 bytes): #211(2920), #212(2920), #213(2920), #214(2920), #215(2920), #217(2920), #218(2920), #219(2920), #220(2920), #221(2920), #222(2920), #223(2920), #224(2920), #225(2920), #226(2920), #227(2920), #228(2920), #229(2920), #230(2920), #231(2920), #232(2920), #233(2920), #234(2920), #235(2920), #236(2920), #237(2920), #238(2920), #239(2920), #240(2920), #241(2920), #242(2920), #243(2920), #244(2920), #245(2920), #246(2920), #247(2920), #248(2920), #249(2920), #250(2920), #251(2920), #252(2920), #253(2920), #254(2920), #255(2920), #256(2920), #257(2920), #258(2920), #259(2920), #260(2920), #261(2920), #262(2920), #263(2920), #264(2920), #265(2920), #266(2920), #267(2920), #268(2920), #269(2920), #270(2920), #271(2920), #272(2920), #273(2920), #274(2920), #275(2920), #276(2920), #277(2920), #278(2920), #279(2920), #280(2920), #281(2920), #282(2920), #283(2920), #284(2920), #285(2920), #286(2920), #287(2920), #288(2920), #289(2920), #290(2920), #291(2920), #292(2920), #293(2920), #294(2920), #295(2920), #296(2920), #297(2920), #298(2920), #299(2920), #300(2920), #301(2920), #302(2920), #303(2920), #304(2920), #305(2920), #306(2920), #307(2920), #308(2920), #309(2920), #310(2920), #311(2920), #312(2920), #313(2920), #314(2920), #315(2920), #316(2920), #317(2920), #318(2920), #319(2920), #320(2920), #321(2920), #322(2920), #323(2920), #324(2920), #325(2920), #326(2920), #327(2920), #328(2920), #329(2920), #330(2920), #331(2920), #332(2920), #333(2920), #334(2920), #335(2920), #336(2920), #337(2920), #338(2920), #339(2920), #340(2920), #341(2920), #342(2920), #343(2920), #344(2920), #345(2920), #346(2920), #347(2920), #348(2920), #349(2920), #350(2920), #351(2920), #352(2920), #353(2920), #354(2920), #355(2920), #356(2920), #357(2920), #358(2920), #359(2920), #360(2920), #361(2920), #362(2920), #363(2920), #364(2920), #365(2920), #366(2920), #367(2920), #368(2920), #369(2920), #370(2920), #371(2920), #372(2920), #373(2920), #374(2920), #375(2920), #376(2920), #377(2920), #378(2920), #379(2920), #380(2920), #381(2920), #382(2920), #383(2920), #384(2920), #385(2920), #386(2920), #387(2920), #388(2920), #389(2920), #390(2920), #391(2920), #392(2920), #393(2920), #394(2920), #395(2920), #396(2920), #397(2920), #398(2920), #399(2920), #400(2920), #401(2920), #402(2920), #403(2920), #404(2920), #405(2920), #406(2920), #407(2920), #408(2920), #409(2920), #410(2920), #411(2920), #412(2920), #413(2920), #414(2920), #415(2920), #416(2920), #417(2920), #418(2920), #419(2920), #420(2920), #421(2920), #422(2920), #423(2920), #424(2920), #425(2920), #426(2920), #427(2920), #428(2920), #429(2920), #430(2920), #431(2920), #432(2920), #433(2920), #434(2920), #435(2920), #436(2920), #437(2920), #438(2920), #439(2920), #440(2920), #441(2920), #442(2920), #443(2920), #444(2920), #445(2920), #446(2920), #447(2920), #448(2920), #449(2920), #450(2920), #451(2920), #452(2920), #453(2920), #454(2920), #455(2920), #456(2920), #457(2920), #458(2920), #459(2920), #460(2920), #461(2920), #462(2920), #463(2920), #464(2920), #465(2920), #466(2920), #467(2920), #468(2920), #469(2920), #470(2920), #471(2920), #4

Demo

Demo video link below:

<https://box.xjtlu.edu.cn/f/9e299a5a7c1641538530/>

Lab Exercise

1) Practice using Wireshark to capture TCP segments during a file upload

Instructions:

1. Start Wireshark on the working network interface.
2. Visit <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and save the file locally as alice.txt
3. Open <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>
4. Begin Wireshark capture, then upload alice.txt via the web form (HTTP POST).
5. Stop capture after the upload completes.

Deliverable:

Take a screenshot showing the sequence of TCP packets exchanged during upload, including the POST command segment. Find two HTTP requests and analyze their handshakes (SYN, ACK, RST). For each of them, write down the source address and destination address information at the packet details pane.

Lab Exercise

2) Explore flow control and retransmission

Instructions:

1. Apply the filter `tcp.analysis.flags` in Wireshark.
2. Identify if any segments were retransmitted.

Questions:

- Were any retransmissions detected? If yes, what could be the cause?
- Find the window size from the packet details pane. What does this indicate about the congestion or flow-control mechanism?