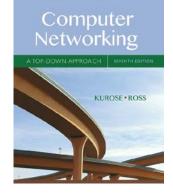
CS 447: Network and Data Communication Wireshark Lab #02: TCP

 $\ensuremath{\mathbb{C}}$ 2005-2016, J.F Kurose and K.W. Ross, All Rights Reserved



In this lab, we'll investigate the behavior of the celebrated TCP protocol in detail. We'll do so by analyzing a trace of the TCP segments sent and received in transferring a 150KB file (containing the text of Lewis Carrol's *Alice's Adventures in Wonderland*) from your computer to a remote server. We'll study TCP's use of sequence and acknowledgement numbers for providing reliable data transfer; we'll see TCP's congestion control algorithm – slow start and congestion avoidance – in action; and we'll look at TCP's receiver-advertised flow control mechanism. We'll also briefly consider TCP connection setup and we'll investigate the performance (throughput and round-trip time) of the TCP connection between your computer and the server.

Before beginning this lab, you'll probably want to review sections 3.5 and 3.7 in the text¹.

◆ Capturing a bulk TCP transfer from your computer to a remote server

Before beginning our exploration of TCP, we'll need to use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server. You'll do so by accessing a Web page that will allow you to enter the name of a file stored on your computer (which contains the ASCII text of *Alice in Wonderland*), and then transfer the file to a Web server using the HTTP POST method (see section 2.2.3 in the text). We're using the POST method rather than the GET method as we'd like to transfer a large amount of data *from* your computer to another computer. Of course, we'll be running Wireshark during this time to obtain the trace of the TCP segments sent and received from your computer.

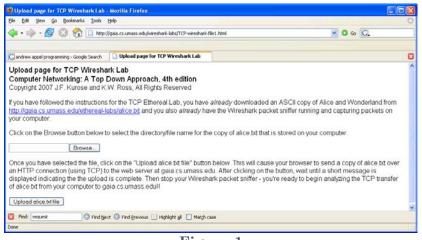


Figure 1

¹ References to figures and sections are for the 7th edition of our text, *Computer Networks, A Top-down Approach, 7th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2016.*

Do the following:

- Start up your web browser. Go the <u>http://gaia.cs.umass.edu/wireshark-labs/alice.txt</u> and retrieve an ASCII copy of *Alice in Wonderland*. Store this file somewhere on your computer.
- Next go to <u>http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html</u>.
- You should see a screen that looks like similar to Figure 01 above.
- Use the *Browse* button in this form to enter the name of the file (full path name) on your computer containing *Alice in Wonderland* (or do so manually). Don't yet press the "*Upload alice.txt file*" button.
- Now start up Wireshark and begin packet capture (*Capture->Start*) and then press *OK* on the Wireshark Packet Capture Options screen (we'll not need to select any options here).
- 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 congratulations message will be displayed in your browser window.
- Stop Wireshark packet capture. Your Wireshark window should look similar to the window shown below.

Ile Edit View Co Capture Analyze Statistics Telephony Tools Internals Help Image: Construct Analyze Statistis Image: Construct An		_																		
Image: Solution of the second seco	000						0.0000000000			Constant States			100 A 100		6)]					
<pre> illter: tcp</pre>	<u>F</u> ile <u>I</u>	<u>E</u> dit <u>V</u> iew	<u>G</u> o <u>C</u> a	pture <u>/</u>	<u>A</u> nalyze	<u>S</u> tatist	ics T	elep	hony	<u> </u>	<u>I</u> nte	rnals	<u>H</u> elp)						
Io. Time Source Destination Protocol Length Info 20 0.306692 192.168.1.102 128.119.245.12 TCP 1514 TCP segment of a reasser 21 0.307571 192.168.1.102 128.119.245.12 TCP 1514 TCP segment of a reasser 22 0.309593 192.168.1.102 128.119.245.12 TCP 1514 TCP segment of a reasser 24 0.356437 128.119.245.12 192.168.1.102 TCP 60 http > health-polling fAC 25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling fAC 26 0.448613 128.119.245.12 192.168.1.102 TCP 60 http > health-polling fAC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling fAC 29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling fAC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 ICP segment of a reasser <td< td=""><td>e i</td><td>u 🖭 🤮</td><td>i 🕍 🗌</td><td> </td><th>X 6</th><td></td><td>O₆</td><td></td><td>🛸 🗳</td><td>• 7</td><td>⊉</td><td></td><td></td><td> ⊕</td><td>Θ</td><td>11</td><td>++</td><td></td><td>\mathbf{V}</td><td>•</td></td<>	e i	u 🖭 🤮	i 🕍 🗌	 	X 6		O ₆		🛸 🗳	• 7	⊉			⊕	Θ	11	++		\mathbf{V}	•
20 0.306692 192.168.1.102 128.119.245.12 TCP 1514 TCP segment of a reassent 21 0.307571 192.168.1.102 128.119.245.12 TCP 1514 TCP segment of a reassent 22 0.308699 192.168.1.102 128.119.245.12 TCP 1514 TCP segment of a reassent 23 0.309553 192.168.1.102 128.119.245.12 TCP 946 TCP segment of a reassent 24 0.356437 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576417 128.119.245.12 192.168.1.102 TCP 1514 [TCP segment of a reassent 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassent 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassent 32 0.578329 192.168.1.102	Filter:	tcp						•	Expre	ssion	. Cl	ear	Apply							
21 0.307571 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 22 0.306699 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 23 0.30553 192.168.1.102 128.119.245.12 TCP 946 [TCP segment of a reasser 24 0.356437 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576417 128.119.245.12 192.168.1.102 TCP 1514 [TCP segment of a reasser 31 0.57667 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578399 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578399 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser <t< td=""><td>No.</td><td>Time</td><td></td><td>Source</td><th></th><td></td><td>10</td><td>esti</td><td>nation</td><td></td><td></td><td>F</td><td>rotoc</td><td>o I Le</td><td>ngth</td><td>Info</td><td></td><td></td><td></td><td></td></t<>	No.	Time		Source			10	esti	nation			F	rotoc	o I Le	ngth	Info				
22 0.308699 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 23 0.309553 192.168.1.102 128.119.245.12 TCP 946 [TCP segment of a reasser 24 0.356437 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 26 0.448613 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 60 http > health-polling [AC 32 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 52 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 53 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 54 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 54 0.577329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 55 0.577329 192.168.1.102 (192.168.1.102), Dst: 118.19.245.12 (102.99.19.19.245.12) 56 0 000 00:05 0 Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) 57 0 Version: 4 58 0 Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) 59 0 Tdentification: 0x1e2f (7727) 59 Flags: 0x02 (Don't Fragment) 50 Fragment of fset: 0 50 Time to live: 128 50 00 00 62 5d af 73 00 20 e0 8a 70 1a 08 00 45 00%.rspE. 50 00 00 62 5d af 73 00 20 e0 8a 70 1a 08 00 45 00%.rspE. 50 00 00 62 5d af 73 00 20 e0 8a 70 1a 08 00 45 00%.rspE. 50 00 00 6 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00%.rspE. 50 00 00 00 6 95 5a c0 a8 01 66 80 77 .		20 0.3066	92	192.168.	1.102		1	28.1	19.245.	12		Т	CP		1514	[TCP	segm	ent of	a rea	asser
23 0.309553 192.168.1.102 128.119.245.12 TCP 946 [TCP segment of a reasser 24 0.356437 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 26 0.448613 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576471 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 31 0.577355 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes ▷ Differentiated Services Field: 0X00 (DSCP 0X00: Default; ECN: 0X00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 200 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00		21 0.3075	71 .	192.168,	1.102		1	28.13	19.245.	12		Т	CP		1514	[TCP	segm	ent of	a rea	asser
24 0.356437 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 26 0.448613 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassent 31 0.577285 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassent 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassent 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassent Therme 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 200 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00		22 0.3086	99 :	192.168.	1.102		1	28.13	19.245.	12		Т	CP		1514	[TCP	segm	ent of	a rea	asser
25 0.400164 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 26 0.448613 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen		23 0.3095	53	192.168,	1.102		1	28.13	19.245.	12		Т	CP		946	[TCP	segm	ent of	a rea	assem
26 0.448613 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 31 0.577355 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 192.168.1.102 Dst LinksysG dataf:73 (00:06:25:dataf:73) 110.1102 </td <td></td> <td>24 0.3564</td> <td>37 .</td> <td>128.119.</td> <th>245.12</th> <td></td> <td>1</td> <td>92.10</td> <td>58.1.10</td> <td>2</td> <td></td> <td>Т</td> <td>CP</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>		24 0.3564	37 .	128.119.	245.12		1	92.10	58.1.10	2		Т	CP							
27 0.500029 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset 31 0.577835 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2 (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment of fae: 0 Time to live: 128 P		25 0.4001	64 .	128.119.	245.12		1	92.10	58.1.10	2		Т	CP							
28 0.545052 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [Ac 29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [Ac 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 31 0.577385 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser TCP 1514 [TCP segment of a reasser 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser TCP		26 0.4486	13 .	128.119.	245.12		1	92.10	58.1.10	2		Т	CP							
29 0.576417 128.119.245.12 192.168.1.102 TCP 60 http > health-polling [AC 30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.576329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser 32 0.576329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) TCP 1514 [TCP segment of a reasser Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) TCP 1514 [TCP segment of a reasser Internet Protocol Version 4, Src: 192.168.1.102 192.168.1.102, Dst: Linksys6_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) Fragent of fas: 0 Time to live: 128 Protocol: TCP (6) Total Total 020 e0 8a 70 1a 08 00 45 00 000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00 100 05 dc 1e 2f 400		27 0.5000	29 :	128.119.	245.12		1	92.10	58.1.10	2		Т	CP							
30 0.576671 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset 31 0.577385 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasset Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Ethernet II, Src: Actionte 88:70:1a (00:20:e0:88:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes > Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) > P Elags: 0x02 (Don't Fragment) Fragment of fiset: 0 Time to live: 128 Protocol: TCP (6)		28 0.5450	52 .	128.119.	245.12		1	92.10	58.1.10	2										
31 0.577385 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen] 32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reassen] Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Ethernet II, Src: Actionte 8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 2000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00 %.rspE. 100 05 dc le 2f 40 00 80 06 ef 5a c0 a8 01 66 80 77 /eZf.w																				
32 0.578329 192.168.1.102 128.119.245.12 TCP 1514 [TCP segment of a reasser] Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Image: Comparison of the																				
Frame 31: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) Ethernet II, Src: Actionte 8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes > Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) > Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 2000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00																				
Ethernet II, Src: Actionte_Ba:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) D00 00 625 da af 73 00 20 e0 8a 70 1a 08 00 45 00		32 0.5783	29 .	192.168.	1.102		1	28.13	19.245.	12		Т	CP		1514	[TCP	segm	ent of	a rea	asser
Ethernet II, Src: Actionte_Ba:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73) Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) D00 00 625 da af 73 00 20 e0 8a 70 1a 08 00 45 00)+
<pre>Internet Protocol Version 4, Src: 192.168.1.102 (192.168.1.102), Dst: 128.119.245.12 (128.119.245.12) Version: 4 Header length: 20 bytes ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00%.rspE. 100 05 dc 1e 2f 40 00 80 06 9f 5a c0 a8 01 66 80 77, /e Zf.w</pre>																				
Version: 4 Header length: 20 bytes ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1ezf (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00																				
Header length: 20 bytes ▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) ▶ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00			col Versi	ion 4, S	Src: 192	.168.1.1	102 (1	92.10	58.1.10	2), Ds	t: 12	8.119	.245.	12 (12	28.119	9.245	.12)			
<pre>▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 1500 Identification: 0x1e2f (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 000 00 625 da af 73 00 20 e0 8a 70 1a 08 00 45 00%.rspE. 100 05 dc 1e 2f 40 00 80 06 9f 5a c0 a8 01 66 80 77, /e Zf.w</pre>																				
Total Length: 1500 Identification: 0x1e2f (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 200 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00																				
Identification: 0x1e2f (7727) ▷ Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 200 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00				ces Fie	ld: 0x00) (DSCP (0x00:	Defa	ult; EC	N: 0x0	0: No	t - ECT	「(Not	ECN-	Capab	le Tr	anspo	rt))		
<pre>b Flags: 0x02 (Don't Fragment) Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 000 00 62 5 da af 73 00 20 e0 8a 70 1a 08 00 45 00%.spE. 100 05 dc le 2f 40 00 80 06 9f 5a c0 a8 01 66 80 77,@Zf.w</pre>																				
Fragment offset: 0 Time to live: 128 Protocol: TCP (6) 000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00%.spE. 100 05 dc 1e 2f 40 00 80 06 9f 5a c0 a8 01 66 80 77/@Zf.w																				
Time to live: 128 Protocol: TCP (6) 000 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00				ragment)															
Protocol: TCP (6) 2000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00																				
000 00 06 25 da af 73 00 20 e0 8a 70 1a 08 00 45 00%.spE. 010 05 dc le 2f 40 00 80 06 9f 5a c0 a8 01 66 80 77/@ Zf.w																				
010 05 dc le 2f 40 00 80 06 9f 5a c0 a8 01 66 80 77/@ Zf.w	Pr	rotocol: TO	P (6)																	
									%s.	p.	E.									
120 t5 0c 04 99 00 50 0d d6 4a dd 24 a2 74 1a 50 10 P 1 4 t P																				1
)30 44 70 91 a4 00 00 20 74 6f 20 68 65 72 20 67 72 Dp t o her gr)40 65 61 74 20 64 65 6c 69 67 68 74 20 69 74 20 66 eat deli ght it f																				
File: "/Users/kurose/Umass/ #Packets: 213 Displayed: 202 Marked: 0 Load time: 0:00.009 #Profile: Default																				

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author's computers². You may well find it valuable to download this trace even if you've captured your own trace and use it, as well as your own trace, when you explore the questions below.

² Download the zip file <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u> and extract the file tcp-ethereal-trace-1. The traces in this zip file were collected by Wireshark running on one of the author's computers, while performing the steps indicated in the Wireshark lab. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the *File* pull down menu, choosing *Open*, and then selecting the tcp-ethereal-trace-1 trace file.

\clubsuit A first look at the captured trace

Before analyzing the behavior of the TCP connection in detail, let's take a high level view of the trace.

• First, filter the packets displayed in the Wireshark window by entering "tcp" (lowercase, no quotes, and don't forget to press return after entering!) into the display filter specification window towards the top of the Wireshark window.

What you should see is series of TCP and HTTP messages between your computer and gaia.cs.umass.edu. You should see the initial three-way handshake containing a SYN message. You should see an HTTP POST message. Depending on the version of Wireshark you are using, you might see a series of "HTTP Continuation" messages being sent from your computer to gaia.cs.umass.edu. Recall from our discussion in the earlier HTTP Wireshark lab, that is no such thing as an HTTP Continuation message – this is Wireshark's way of indicating that there are multiple TCP segments being used to carry a single HTTP message. In more recent versions of Wireshark, you'll see "[TCP segment of a reassembled PDU]" in the Info column of the Wireshark display to indicate that this TCP segment contained data that belonged to an upper layer protocol message (in our case here, HTTP). You should also see TCP ACK segments being returned from gaia.cs.umass.edu to your computer.

Answer the following questions, by opening the Wireshark captured packet file *tcp-ethereal-trace-*1 in <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u> (that is download the trace and open that trace in Wireshark; see footnote 2). Whenever possible, when answering a question you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout³ to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

- 1. What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it's probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the "details of the selected packet header window" (refer to Figure 2 in the "Getting Started with Wireshark" Lab if you're uncertain about the Wireshark windows.
- 2. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

If you have been able to create your own trace, answer the following question:

3. What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

Since this lab is about TCP rather than HTTP, let's change Wireshark's "listing of captured packets" window so that it shows information about the TCP segments containing the HTTP messages, rather than about the HTTP messages. To have Wireshark do this, select *Analyze*-

³ What do we mean by "annotate"? If you hand in a paper copy, please highlight where in the printout you've found the answer and add some text (preferably with a colored pen) noting what you found in what you 've highlight. If you hand in an electronic copy, it would be great if you could also highlight and annotate.

>Enabled Protocols. Then uncheck the HTTP box and select *OK.* You should now see a Wireshark window that looks like:

⊖ ⊖ (File			race-1 [Wireshark 1.6.7 (SVN Rev		nk-1.6)]	
<u>r</u> ne	<u>Ean view Go</u>		tistics Telephony <u>T</u> ools <u>I</u> r			
		🖹 🖹 🗶 🕄 🖉	🖥 🔍 🔶 🏟 🌳 🔁 💈		€ € ₪ 🖭 🕅	•
Filter	tcp		▼ Expression	Clear Apply		
No.	Time	Source	Destination	Protoco	l Length Info	
	1 0.000000	192.168.1.102	128.119.245.12	TCP	62 health-polling > htt	p [SY
	2 0.023172	128.119.245.12	192.168.1.102	TCP	62 http > health-pollin	g [SY
	3 0.023265	192.168.1.102	128.119.245.12	TCP	54 health-polling > htt	
	4 0.026477	192.168.1.102	128.119.245.12	TCP	619 health-polling > htt	
	5 0.041737	192.168.1.102	128.119.245.12	TCP	1514 health-polling > htt	
	6 0.053937	128.119.245.12	192.168.1.102	TCP	60 http > health-pollin	
	7 0.054026	192.168.1.102	128.119.245.12	TCP	1514 health-polling > htt	
	8 0.054690	192.168.1.102	128.119.245.12	TCP	1514 health-polling > htt	
	9 0.077294	128.119.245.12	192.168.1.102	TCP	60 http > health-pollin	
	10 0.077405	192.168.1.102	128.119.245.12	TCP	1514 health-polling > htt	
	11 0.078157	192.168.1.102	128.119.245.12	TCP	1514 health-polling > htt	
	12 0.124085	128.119.245.12	192.168.1.102	TCP	60 http > health-pollin	
	13 0.124185	192.168.1.102	128.119.245.12	TCP	1201 health-polling > htt	p [PS
)+
🕨 Fran	ne 1: 62 bytes or	wire (496 bits), 62 b	ytes captured (496 bits)			
			:e0:8a:70:1a), Dst: Linksys(6_da:af:73 (00:	:06:25:da:af:73)	ĺ
▽ D		sysG_da:af:73 (00:06:29				L L
		sG_da:af:73 (00:06:25:				
			t: Individual address (unica			
			t: Globally unique address (factory defaul	t)	
⊽ S		Ba:70:1a (00:20:e0:8a:				
		te_8a:70:1a (00:20:e0:				
			t: Individual address (unica			
	0	= LG bi	t: Globally unique address (factory defaul	t)	
		3 00 20 e0 8a 70 la 0		Ε.		
		08006 a518 c0 a8 0				
		0 0d d6 01 f4 00 00 0 0 02 04 05 b4 01 01 0		р.		
0030	40 00 16 69 00 0	0 0 2 0 4 0 5 0 4 0 1 0 1 0	4 02 @			

This is what we're looking for - a series of TCP segments sent between your computer and gaia.cs.umass.edu. We will use the packet trace that you have captured (and/or the packet trace *tcp-ethereal-trace-1* in <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u>; see earlier footnote) to study TCP behavior in the rest of this lab.

File: "/Users/kurose/Umass/.... Packets: 213 Displayed: 202 Marked: 0 Load time: 0:00.011 Profile: Default

✤ TCP Basics

Answer the following questions for the TCP segments:

- 4. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?
- 5. What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did gaia.cs.umass.edu determine that value? What is it in the segment that identifies the segment as a SYNACK segment?
- 6. What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll 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.
- 7. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see Section 3.5.3, page 242

in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 242 for all subsequent segments.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the "listing of captured packets" window that is being sent from the client to the gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph->Round Trip Time Graph*.

- 8. What is the length of each of the first six TCP segments?⁴
- 9. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?
- 10. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?
- 11. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 250 in the text).
- 12. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

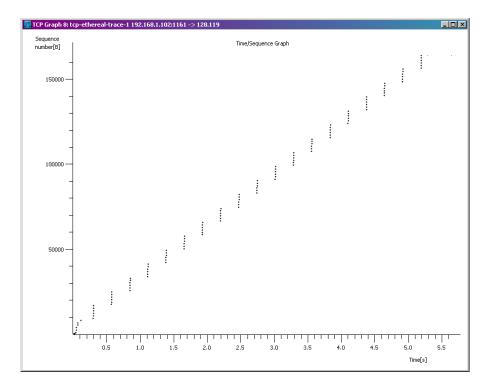
✤ TCP congestion control in action

Let's now examine the amount of data sent per unit time from the client to the server. Rather than (tediously!) calculating this from the raw data in the Wireshark window, we'll use one of Wireshark's TCP graphing utilities - *Time-Sequence-Graph(Stevens)* - to plot out data.

• Select a TCP segment in the Wireshark's "listing of captured-packets" window. Then select the menu : *Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens)*. You should see a plot that looks similar to the following plot, which was created from the captured packets in the packet trace *tcp-ethereal-trace-1* in <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u> (see earlier footnote):

Here, each dot represents a TCP segment sent, plotting the sequence number of the segment versus the time at which it was sent. Note that a set of dots stacked above each other represents a series of packets that were sent back-to-back by the sender.

⁴ The TCP segments in the tcp-ethereal-trace-1 trace file are all less that 1460 bytes. This is because the computer on which the trace was gathered has an Ethernet card that limits the length of the maximum IP packet to 1500 bytes (40 bytes of TCP/IP header data and 1460 bytes of TCP payload). This 1500 byte value is the standard maximum length allowed by Ethernet. If your trace indicates a TCP length greater than 1500 bytes, and your computer is using an Ethernet connection, then Wireshark is reporting the wrong TCP segment length; it will likely also show only one large TCP segment rather than multiple smaller segments. Your computer is indeed probably sending multiple smaller segments, as indicated by the ACKs it receives. This inconsistency in reported segment lengths is due to the interaction between the Ethernet driver and the Wireshark software. We recommend that if you have this inconsistency, that you perform this lab using the provided trace file.



Answer the following questions for the TCP segments the packet trace *tcp-ethereal-trace-1* in <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u>

- 13. Use the *Time-Sequence-Graph(Stevens*) plotting tool to view the sequence number versus time plot of segments being sent from the client to the gaia.cs.umass.edu server. Can you identify where TCP's slowstart phase begins and ends, and where congestion avoidance takes over? Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.
- 14. Answer each of two questions above for the trace that you have gathered when you transferred a file from your computer to gaia.cs.umass.edu