

1.) List up to 10 different protocols that appear in the protocol column in the unfiltered packet-listing window. As I don't have control over the data flowing over your network the time of your lab, I don't know exactly how many and what protocols those will be. I do expect that you have a bunch (if less than 5, please look harder). Just list out those that you see, but don't bother to list more than 10.

1. UDP
2. TCP
3. MDNS
4. IGMPv2
5. ICMPv6
6. HTTP
7. DNS
8. ARP
9. DHCP
10. AJP13

2.) Open your web browser and use the following address [www.tripod.com](http://www.tripod.com). Once the page loads wait 3 seconds and stop the capture. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.) Include a screenshot and describe where you got the data to answer this question.

It took .349 seconds from the initial HTTP GET message to the HTTP OK reply. Since the site was permanently moved though, there was a second GET message sent after the first and before the reply. So the time from that GET message to the OK reply was .123 seconds. I found the data to answer this question within the red box. In the info section i looked for GET and OK, the the corresponding times next to them. Then I subtracted the more recent time value from the older time value.

No.	Time	Source	Destination	Protocol	Length	Info
41	12:28:56.637104	10.0.0.208	209.202.252.55	HTTP	393	GET / HTTP/1.1
45	12:28:56.750785	209.202.252.55	10.0.0.208	HTTP	533	HTTP/1.1 301 Moved Permanently (text/html)
54	12:28:56.863476	10.0.0.208	209.202.252.55	HTTP	399	GET / HTTP/1.1
78	12:28:56.986650	209.202.252.55	10.0.0.208	HTTP	740	HTTP/1.1 200 OK (text/html)
90	12:28:57.135745	10.0.0.208	209.202.254.60	HTTP	368	GET /ly/tpSite/css/tpStyle.css HTTP/1.1
93	12:28:57.139554	10.0.0.208	209.202.254.60	HTTP	366	GET /ly/tpSite/css/fancy.css HTTP/1.1
98	12:28:57.155151	10.0.0.208	209.202.254.60	HTTP	353	GET /ly/js/jquery-1.8.2.min.js HTTP/1.1
99	12:28:57.155325	10.0.0.208	209.202.254.60	HTTP	361	GET /ly/js/jquery.ba-hashchange.min.js HTTP/1.1
102	12:28:57.244844	10.0.0.208	209.202.254.60	HTTP	346	GET /ly/tpSite/js/tp.js HTTP/1.1
105	12:28:57.257097	209.202.254.60	10.0.0.208	HTTP	1436	HTTP/1.1 200 OK (text/css)
107	12:28:57.257512	10.0.0.208	209.202.254.60	HTTP	358	GET /ly/tpSite/js/flexslider-min.js HTTP/1.1
118	12:28:57.259069	209.202.254.60	10.0.0.208	HTTP	1142	HTTP/1.1 200 OK (text/css)
122	12:28:57.260451	10.0.0.208	209.202.254.60	HTTP	349	GET /ly/tpSite/js/fancy.js HTTP/1.1
126	12:28:57.279424	209.202.254.60	10.0.0.208	HTTP	1345	HTTP/1.1 200 OK (application/javascript)
146	12:28:57.292955	2601:f02:9e10...:2607:f8b0:400a:809...		HTTP	442	GET /css?family=Source+Sans+Pro:400,200,300,700 HTTP/1.1
149	12:28:57.360260	209.202.254.60	10.0.0.208	HTTP	518	HTTP/1.1 200 OK (application/javascript)
153	12:28:57.372674	209.202.254.60	10.0.0.208	HTTP	682	HTTP/1.1 200 OK (application/javascript)
168	12:28:57.382154	209.202.254.60	10.0.0.208	HTTP	306	HTTP/1.1 200 OK (application/javascript)
186	12:28:57.392304	209.202.254.60	10.0.0.208	HTTP	261	HTTP/1.1 200 OK (application/javascript)

Frame 54: 399 bytes on wire (3192 bits), 399 bytes captured (3192 bits) on interface 0  
Ethernet II, Src: Apple\_30:62:0e (38:f9:d3:30:62:0e), Dst: ArrisGro\_af:6d:29 (b0:da:f9:af:6d:29)  
Internet Protocol Version 4, Src: 10.0.0.208, Dst: 209.202.252.55  
Transmission Control Protocol, Src Port: 49505, Dst Port: 80, Seq: 1187095179, Ack: 1301988148, Len: 345

**Hypertext Transfer Protocol**

**GET / HTTP/1.1\r\n**

    Host: www.tripod.lycos.com\r\n

    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:74.0) Gecko/20100101 Firefox/74.0\r\n

    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\n

    Accept-Language: en-US,en;q=0.5\r\n

    Accept-Encoding: gzip, deflate\r\n

    Connection: keep-alive\r\n

    Upgrade-Insecure-Requests: 1\r\n

    \r\n

**[Full request URI: http://www.tripod.lycos.com/]**

0000 b0 da f9 af 6d 29 38 f9 d3 30 62 0e 08 00 45 00 ...m)8...0b...E...

0010 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00

3. What is the Internet address (IP address) of [www.tripod.com](http://www.tripod.com)? What is the Internet address of your computer (This might be a private address, if you are behind a NAT device. No worries, we'll learn about that later)? Include a screenshot and describe where you got the data to answer this question.

- The IP address of [www.tripod.com](http://www.tripod.com) is 209.202.252.55 and the IP address of my computer is 10.0.0.208. I got the data to answer this question from the source and destination portion of the packet listing window. My IP is the one sending the GET request and [www.tripod.com](http://www.tripod.com) is the one sending the OK response

The screenshot shows a Wireshark interface with the following details:

- Frame 54:** 399 bytes on wire (3192 bits), 399 bytes captured (3192 bits) on interface 0
- Ethernet II:** Src: Apple\_30:62:0e (38:f9:d3:30:62:0e), Dst: ArrisGro\_af:6d:29 (b0:d:a:f9:af:6d:29)
- Internet Protocol Version 4:** Src: 10.0.0.208, Dst: 209.202.252.55
- Transmission Control Protocol:** Src Port: 49505, Dst Port: 80, Seq: 1187095179, Ack: 1301988148, Len: 345
- Hypertext Transfer Protocol:**
  - GET / HTTP/1.1**
  - Host: www.tripod.lycos.com\r\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:74.0) Gecko/20100101 Firefox/74.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8\r\nAccept-Language: en-US,en;q=0.5\r\nAccept-Encoding: gzip, deflate\r\nConnection: keep-alive\r\nUpgrade-Insecure-Requests: 1\r\n\r\n
  - Full request URL:** http://www.tripod.lycos.com/

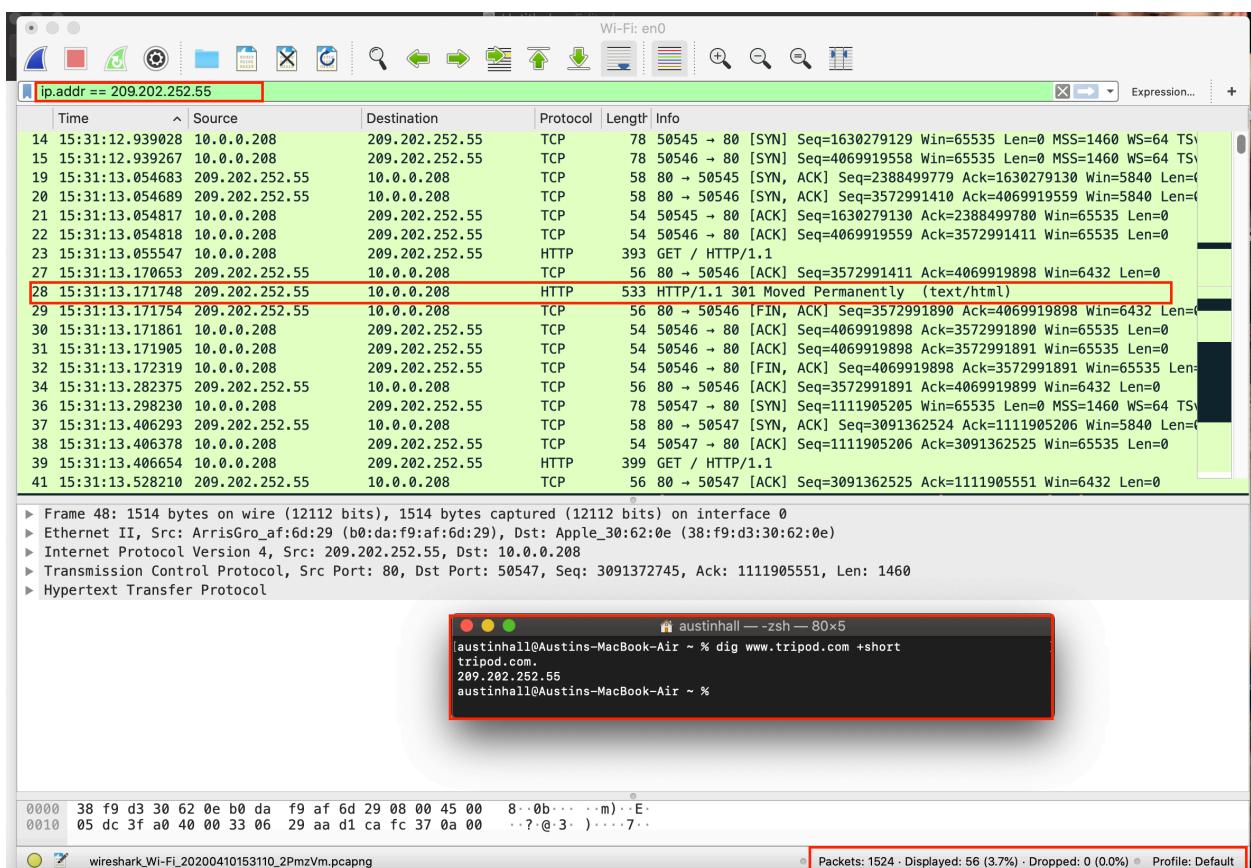
The packet list pane shows the following table:

No.	Time	Source	Destination	Protocol	Length	Info
41	12:28:56.637104	10.0.0.208	209.202.252.55	HTTP	393	GET / HTTP/1.1
45	12:28:56.750785	209.202.252.55	10.0.0.208	HTTP	533	HTTP/1.1 301 Moved Permanently (text/html)
54	12:28:56.863476	10.0.0.208	209.202.252.55	HTTP	399	GET / HTTP/1.1
78	12:28:56.986650	209.202.252.55	10.0.0.208	HTTP	740	HTTP/1.1 200 OK (text/html)
90	12:28:57.135745	10.0.0.208	209.202.254.60	HTTP	368	GET /ly/tpSite/css/tpStyle.css HTTP/1.1
93	12:28:57.139554	10.0.0.208	209.202.254.60	HTTP	366	GET /ly/tpSite/css/fancy.css HTTP/1.1
98	12:28:57.155151	10.0.0.208	209.202.254.60	HTTP	353	GET /ly/js/jquery-1.8.2.min.js HTTP/1.1
99	12:28:57.155325	10.0.0.208	209.202.254.60	HTTP	361	GET /ly/js/jquery.ba-hashchange.min.js HTTP/1.1
102	12:28:57.244844	10.0.0.208	209.202.254.60	HTTP	346	GET /ly/tpSite/js/tp.js HTTP/1.1
105	12:28:57.257097	209.202.254.60	10.0.0.208	HTTP	1436	HTTP/1.1 200 OK (text/css)
107	12:28:57.257512	10.0.0.208	209.202.254.60	HTTP	358	GET /ly/tpSite/js/flexslider-min.js HTTP/1.1
118	12:28:57.259069	209.202.254.60	10.0.0.208	HTTP	1142	HTTP/1.1 200 OK (text/css)
122	12:28:57.260451	10.0.0.208	209.202.254.60	HTTP	349	GET /ly/tpSite/js/fancy.js HTTP/1.1
126	12:28:57.279424	209.202.254.60	10.0.0.208	HTTP	1345	HTTP/1.1 200 OK (application/javascript)
146	12:28:57.292955	2601:602:c980:9e10...	2607:f8b0:400a:809...	HTTP	442	GET /css?family=Source+Sans+Pro:400,200,300,700 HTTP/1.1
149	12:28:57.360260	209.202.254.60	10.0.0.208	HTTP	518	HTTP/1.1 200 OK (application/javascript)
153	12:28:57.372674	209.202.254.60	10.0.0.208	HTTP	682	HTTP/1.1 200 OK (application/javascript)
168	12:28:57.382154	209.202.254.60	10.0.0.208	HTTP	306	HTTP/1.1 200 OK (application/javascript)
186	12:28:57.392304	209.202.254.60	10.0.0.208	HTTP	261	HTTP/1.1 200 OK (application/javascript)

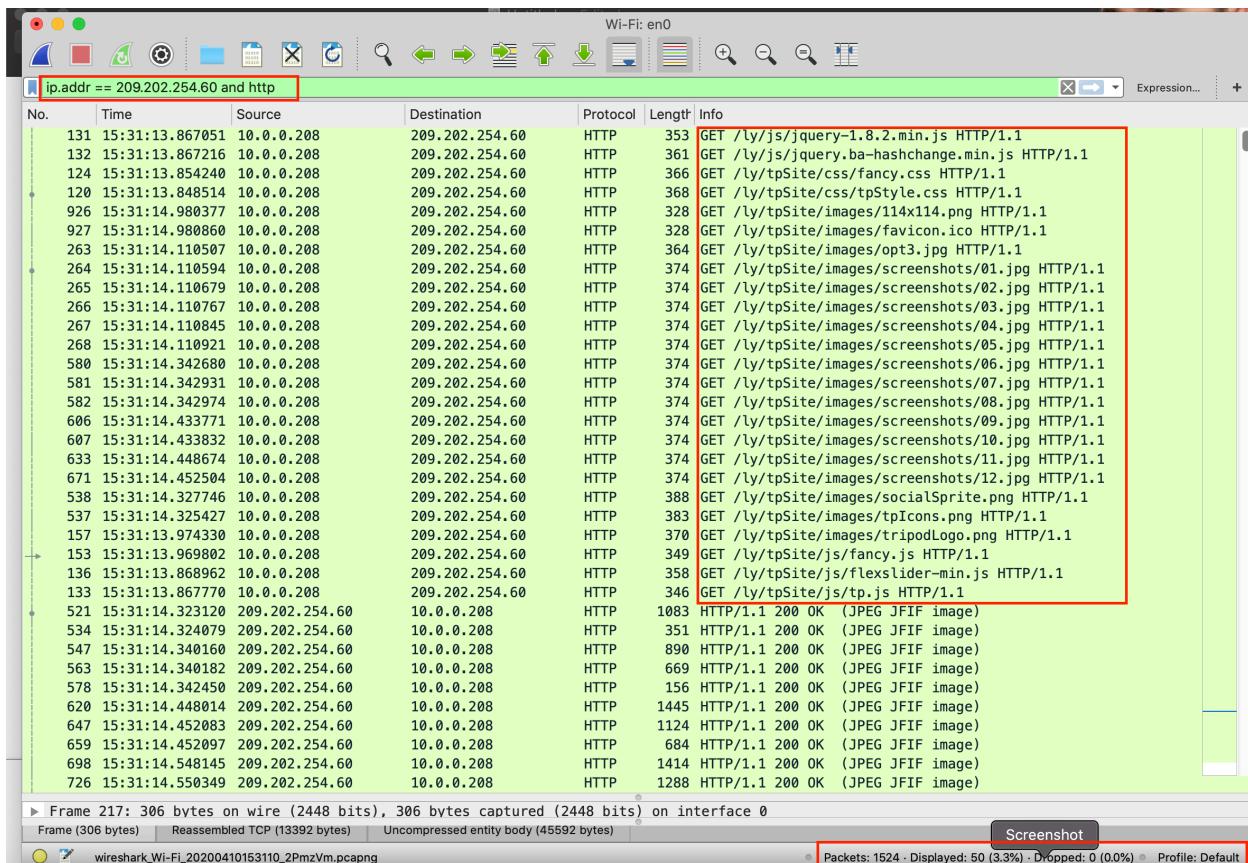
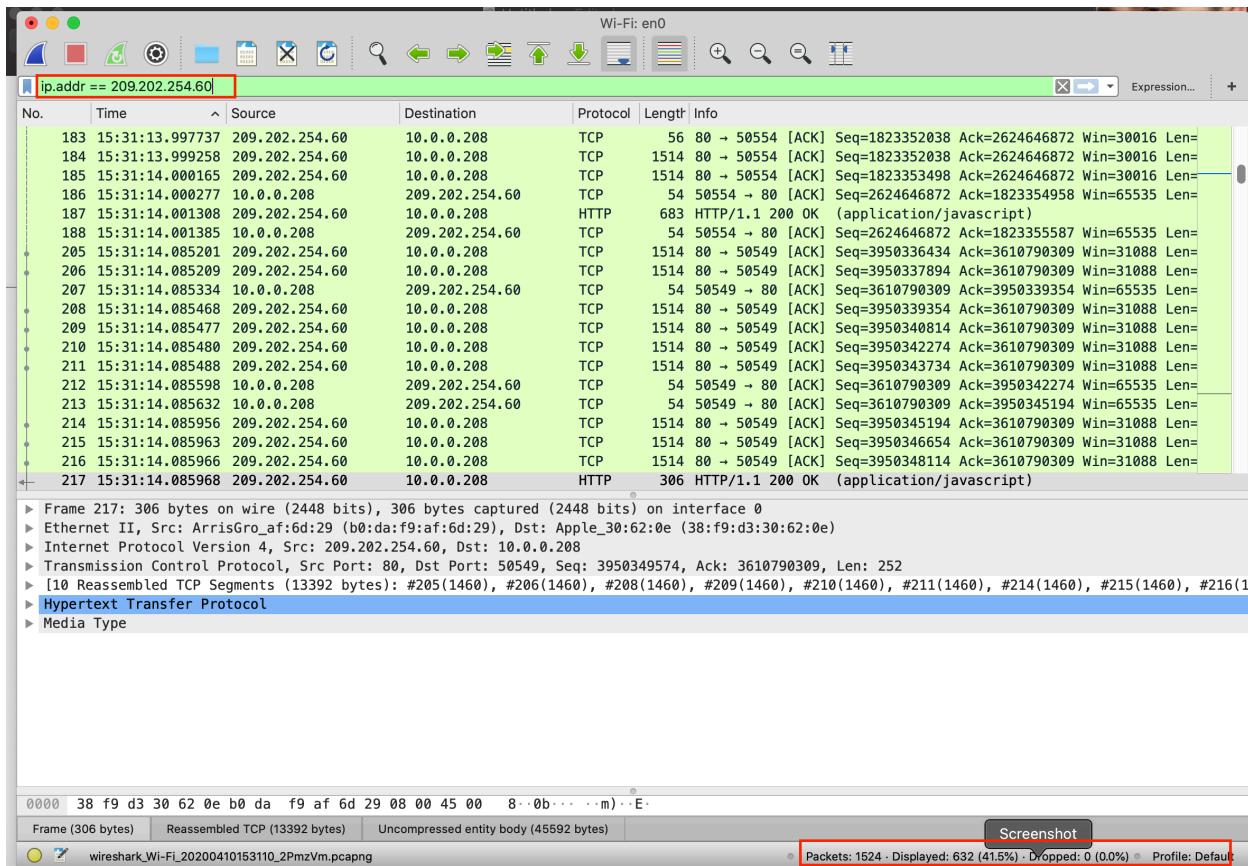
4. How many packets did you capture (total of all protocols, not just HTTP)? Now, use display filters to determine how many packets contain your ip address (hint: Use ip.addr instead of the clumsy ip.src or ip.dst format). What is this filter you used? Now, reverse the filter to determine how many packets don't contain your ip address. See any problems here? If not, you've already figured out the point of this question, so explain how you did so. If so, how can this problem be fixed? What are the appropriate display filters to use? How does Wireshark warn you of such a problem? (This is an important detail to remember about Wireshark. Please ensure you've discussed the problem well enough so that the grader can ensure you explored it thoroughly. If your numbers show you don't have a problem, then figure out how you might reverse the filter in such a way as to cause a problem.)
- I captured a total of 1505 packets. The number of packets that contain my IP address is 1228 packets. This filter that I used is ip.addr == 10.0.0.208. To reverse the filter, I just added a “!” before ip.addr == 10.0.0.208 and didn't see a problem. I think the purpose of this exercise is to show that when using the comparator “!=“ as in “ip.addr != 10.0.0.208”, it will give undesired results. This is because the expression is read as “the packet contains a field named ip.addr with a value different from 10.0.0.208”. Since an IP datagram contains both a source and destination the expression will evaluate to true whenever at least one of the two addresses differs from 10.0.0.208. Wireshark warns of this problem by displaying the warning “!=‘ is deprecated or may have unexpected results.

5. Using the address [www.tripod.com](http://www.tripod.com). Once the page loads wait 3 seconds and stop the capture. Use your newly acquired Wireshark skills to capture the process when your browser loads the front page of tripods website. How many packets did you capture? Were all of them HTTP? How many HTTP requests did you make? Were all the replies "200 OK"? Did you find anything else interesting? Please ensure you have examined this packet capture in detail, using appropriate Wireshark functionality. Write up what you observed(include screen captures to justify your answers).

- All justifications are marked in red in the screenshots.



- I captured 56 packets related to [www.tripod.com](http://www.tripod.com), but the site has been “permanently moved” to [www.tripod.lycos.com](http://www.tripod.lycos.com) (so for the rest of the question I'll answer using that site).



Wi-Fi: en0

ip.addr == 209.202.254.60 and http

No.	Time	Source	Destination	Protocol	Length	Info
606	15:31:14.433771	10.0.0.208	209.202.254.60	HTTP	374	GET /ly/tpSite/images/screenshots/09.jpg HTTP/1.1
607	15:31:14.433832	10.0.0.208	209.202.254.60	HTTP	374	GET /ly/tpSite/images/screenshots/10.jpg HTTP/1.1
633	15:31:14.448674	10.0.0.208	209.202.254.60	HTTP	374	GET /ly/tpSite/images/screenshots/11.jpg HTTP/1.1
671	15:31:14.452504	10.0.0.208	209.202.254.60	HTTP	374	GET /ly/tpSite/images/screenshots/12.jpg HTTP/1.1
538	15:31:14.327746	10.0.0.208	209.202.254.60	HTTP	388	GET /ly/tpSite/images/socialSprite.png HTTP/1.1
537	15:31:14.325427	10.0.0.208	209.202.254.60	HTTP	383	GET /ly/tpSite/images/tpIcons.png HTTP/1.1
157	15:31:13.974330	10.0.0.208	209.202.254.60	HTTP	370	GET /ly/tpSite/images/tripodLogo.png HTTP/1.1
153	15:31:13.969802	10.0.0.208	209.202.254.60	HTTP	349	GET /ly/tpSite/js/fancy.js HTTP/1.1
136	15:31:13.868962	10.0.0.208	209.202.254.60	HTTP	358	GET /ly/tpSite/js/flexslider-min.js HTTP/1.1
133	15:31:13.867770	10.0.0.208	209.202.254.60	HTTP	346	GET /ly/tpSite/js/tp.js HTTP/1.1
521	15:31:14.323120	209.202.254.60	10.0.0.208	HTTP	1083	HTTP/1.1 200 OK (JPEG JFIF image)
534	15:31:14.324079	209.202.254.60	10.0.0.208	HTTP	351	HTTP/1.1 200 OK (JPEG JFIF image)
547	15:31:14.340160	209.202.254.60	10.0.0.208	HTTP	890	HTTP/1.1 200 OK (JPEG JFIF image)
563	15:31:14.340182	209.202.254.60	10.0.0.208	HTTP	669	HTTP/1.1 200 OK (JPEG JFIF image)
578	15:31:14.342450	209.202.254.60	10.0.0.208	HTTP	156	HTTP/1.1 200 OK (JPEG JFIF image)
620	15:31:14.448014	209.202.254.60	10.0.0.208	HTTP	1445	HTTP/1.1 200 OK (JPEG JFIF image)
647	15:31:14.452083	209.202.254.60	10.0.0.208	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)
659	15:31:14.452097	209.202.254.60	10.0.0.208	HTTP	684	HTTP/1.1 200 OK (JPEG JFIF image)
698	15:31:14.548145	209.202.254.60	10.0.0.208	HTTP	1414	HTTP/1.1 200 OK (JPEG JFIF image)
726	15:31:14.550349	209.202.254.60	10.0.0.208	HTTP	1288	HTTP/1.1 200 OK (JPEG JFIF image)
749	15:31:14.560025	209.202.254.60	10.0.0.208	HTTP	795	HTTP/1.1 200 OK (JPEG JFIF image)
775	15:31:14.568161	209.202.254.60	10.0.0.208	HTTP	754	HTTP/1.1 200 OK (JPEG JFIF image)
923	15:31:14.976072	209.202.254.60	10.0.0.208	HTTP	1230	HTTP/1.1 200 OK (JPEG JFIF image)
221	15:31:14.085978	209.202.254.60	10.0.0.208	HTTP	481	HTTP/1.1 200 OK (PNG)
596	15:31:14.433029	209.202.254.60	10.0.0.208	HTTP	333	HTTP/1.1 200 OK (PNG)
598	15:31:14.433031	209.202.254.60	10.0.0.208	HTTP	1426	HTTP/1.1 200 OK (PNG)
1015	15:31:15.090034	209.202.254.60	10.0.0.208	HTTP	1401	HTTP/1.1 200 OK (PNG)
162	15:31:13.987306	209.202.254.60	10.0.0.208	HTTP	1345	HTTP/1.1 200 OK (application/javascript)
166	15:31:13.992946	209.202.254.60	10.0.0.208	HTTP	518	HTTP/1.1 200 OK (application/javascript)
187	15:31:14.001308	209.202.254.60	10.0.0.208	HTTP	683	HTTP/1.1 200 OK (application/javascript)
217	15:31:14.0885968	209.202.254.60	10.0.0.208	HTTP	306	HTTP/1.1 200 OK (application/javascript)
250	15:31:14.107532	209.202.254.60	10.0.0.208	HTTP	261	HTTP/1.1 200 OK (application/javascript)
1012	15:31:15.090030	209.202.254.60	10.0.0.208	HTTP	930	HTTP/1.1 200 OK (image/vnd.microsoft.icon)
149	15:31:13.969341	209.202.254.60	10.0.0.208	HTTP	1142	HTTP/1.1 200 OK (text/css)
155	15:31:13.973566	209.202.254.60	10.0.0.208	HTTP	1436	HTTP/1.1 200 OK (text/css)

Frame 521: 1083 bytes on wire (8664 bits), 1083 bytes captured (8664 bits) on interface 0

Frame (1083 bytes) Reassembled TCP (56509 bytes)

Screenshot

wireshark-Wi-Fi\_20200410153110\_2PmzVm.pcapng

Packets: 1524 · Displayed: 50 (3.3%) · Dropped: 0 (0%) · Profile: Default

Wi-Fi: en0

ip.addr == 209.202.254.60 and lhttp

No.	Time	Source	Destination	Protocol	Length	Info
118	15:31:13.847746	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610789700 Ack=3950323666 Win=65535 Len=
141	15:31:13.967837	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790014 Ack=3950326586 Win=65535 Len=
144	15:31:13.967988	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790014 Ack=3950329506 Win=65535 Len=
150	15:31:13.969446	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790014 Ack=3950332426 Win=65535 Len=
151	15:31:13.969446	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790014 Ack=3950335346 Win=65535 Len=
152	15:31:13.969446	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790014 Ack=3950336434 Win=65535 Len=
207	15:31:14.085334	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790309 Ack=3950339354 Win=65535 Len=
212	15:31:14.085598	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790309 Ack=3950342274 Win=65535 Len=
213	15:31:14.085632	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790309 Ack=3950345194 Win=65535 Len=
222	15:31:14.086201	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790309 Ack=3950348114 Win=65535 Len=
223	15:31:14.086202	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790309 Ack=3950349826 Win=65535 Len=
305	15:31:14.216429	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950352746 Win=65535 Len=
310	15:31:14.216574	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950355666 Win=65535 Len=
311	15:31:14.216574	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950358856 Win=65535 Len=
321	15:31:14.217123	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950361506 Win=65535 Len=
322	15:31:14.217124	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950364426 Win=65535 Len=
324	15:31:14.217159	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950367346 Win=65535 Len=
333	15:31:14.219440	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950370266 Win=65535 Len=
334	15:31:14.219440	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950373186 Win=65535 Len=
335	15:31:14.219440	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950376106 Win=65535 Len=
336	15:31:14.219480	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950379026 Win=65535 Len=
341	15:31:14.220252	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950381946 Win=65535 Len=
342	15:31:14.220295	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950384866 Win=65535 Len=
351	15:31:14.221224	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950387786 Win=65535 Len=
352	15:31:14.221244	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950390706 Win=65535 Len=
510	15:31:14.322230	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950393626 Win=65535 Len=
524	15:31:14.323180	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950396546 Win=65535 Len=
525	15:31:14.323210	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950399466 Win=65535 Len=
526	15:31:14.323211	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950402386 Win=65535 Len=
528	15:31:14.323235	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950405306 Win=65535 Len=
529	15:31:14.323242	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=3610790629 Ack=3950406335 Win=65535 Len=
591	15:31:14.432564	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=361079058 Ack=395040255 Win=65535 Len=
592	15:31:14.432565	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=361079058 Ack=3950412175 Win=65535 Len=
600	15:31:14.433071	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=361079058 Ack=3950415095 Win=65535 Len=
601	15:31:14.433087	10.0.0.208	209.202.254.60	TCP	54	50549 → 80 [ACK] Seq=361079058 Ack=3950418015 Win=65535 Len=

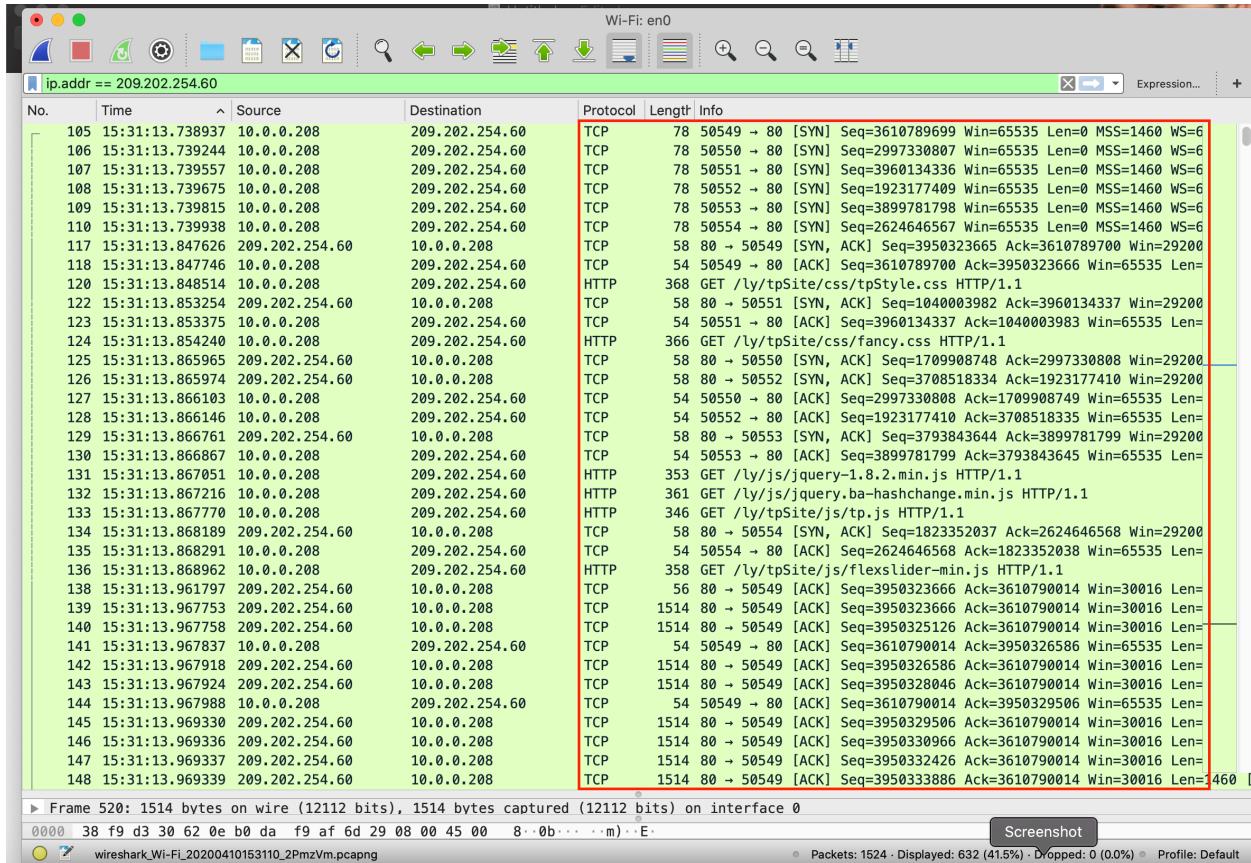
Frame 520: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

0000 38 f9 d3 30 62 0e b0 da f9 0d 29 08 00 45 00 8...b0...m...E·

Screenshot

Hypertext Transfer Protocol: Protocol

Packets: 1524 · Displayed: 582 (38.2%) · Dropped: 0 (0%) · Profile: Default



- I captured 632 packets from www.tripod.lycos.com. Not all of them were HTTP, there were 50 HTTP packets captured and 582 TCP packets captured. I made 25 HTTP GET requests, and all of the replies were 200 OK. One interesting thing I found is that there was a GET request for every media type displayed on the web page. There was also a three way TCP handshake after each GET request and before each reply. One more interesting thing is that there were several ACKs sent in quick succession.