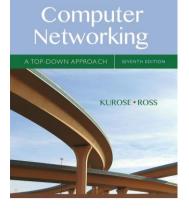
CS 447: Network and Data Communication

Wireshark Lab #03: Ethernet & ARP

© 2005-2017, J.F Kurose and K.W. Ross, All Rights Reserved



<u>Note</u>: Make sure you produce your answers and any packet prints in PDF. Moodle will only accept PDF files.

In this lab, we'll investigate the Ethernet protocol and the ARP protocol. Before beginning this lab, you'll probably want to review sections 6.4.1 (Link-layer addressing and ARP) and 6.4.2 (Ethernet) in the text¹. RFC 826 (<u>ftp://ftp.rfc-editor.org/in-notes/std/std37.txt</u>) contains the gory details of the ARP protocol, which is used by an IP device to determine the IP address of a remote interface whose Ethernet address is known.

Capturing and Analyzing Ethernet Frames

Let's begin by capturing a set of Ethernet frames to study. Do the following²:

- First, make sure your browser's cache is empty.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser <u>http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-lab-file3.html</u>. Your browser should display the rather lengthy US Bill of Rights.
- Stop Wireshark packet capture. First, find the packet numbers (the leftmost column in the upper Wireshark window) of the HTTP GET message that was sent from your computer to gaia.cs.umass.edu, as well as the beginning of the HTTP response message sent to your computer by gaia.cs.umass.edu. You should see a screen that looks something like this (where packet 55 in the screenshot below contains the HTTP GET message).

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

 $^{^{\}rm 2}$ If you are unable to run Wireshark live on a computer, you can download the zip file

<u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u> and extract the file *ethernet--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 ethernet-ethereal-trace-1 trace file. You can then use this trace file to answer the questions below.

NomeNomeNomeNomeNomeNomeNome0.550012.10.10.20.7.7.0.50.50.500 <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>							
4 5 4 500 400 7 12 16 2.0.7 10 10 12 16 2.0.7 10 40 10 5 5 5 5 5 5 5 5 5 5 5 4 2.0 5 2 5 5 1 2.0 5 2.0 5 5 5 5 5 5 5 5 4 2.0 5 2 5 5 1 2.0 5 5 5 5 5 5 5 5 5 4 2.0 5 2 5 5 1 2.0 5 5 5 5 5 5 5 5 5 5 4 2.0 5 2 5 5 1 2.0 5 5 5 5 5 5 5 5 5 5 1 2.0 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5							
9 5 9812344 192 104 30, 75 8, 4, 4 006 97 5 standard query backs 0047 eds 097 9 5 5 9812948 8, 4, 4 192 104 30, 75 0.6 95 5 standard query response backs 000 KKY eds NKY DKKY DKKY DKKY DKKY DKKY DKKY DKKY	43 5.859300057	192.168.243.75	8.8.4.4			lu OPT	
4 5 9,19101 8,14.4 10,14,20.7 95 95 95,191010 10,14,20.7 95 95,191010 10,14,20.7 95 95,191010 10,14,20.7 10,15,20.7 <td< td=""><td>44 5.897832072</td><td>8.8.4.4</td><td>192.168.243.75</td><td>DNS</td><td>899 Standard query response @x6ddb DS um</td><td>ss.edu NSEC3 RRSIG SOA a.edu-servers.net RRSIG NSEC3 RRSIG OPT</td><td></td></td<>	44 5.897832072	8.8.4.4	192.168.243.75	DNS	899 Standard query response @x6ddb DS um	ss.edu NSEC3 RRSIG SOA a.edu-servers.net RRSIG NSEC3 RRSIG OPT	
9 9	45 5.898121948	192.168.243.75	8.8.4.4	DNS	97 Standard query 0x96ca DNSKEY edu OPT		
45 5 175437 8.8.4 19.18 42.0.7. 195 19 5 1000000 45 00 10 10000000000000000000	46 5.914914083	8.8.4.4	192.168.243.75	DNS	995 Standard query response 0x96ca DNSKE	edu DNSKEY DNSKEY RRSIG OPT	
49 5 920516 8.1.4 10.16 10.2017 105 118 5 1000000000000000000000000000000000	47 5.915101136	192.168.243.75	8.8.4.4	DNS	97 Standard query 0x6ce5 DS edu OPT		
95 9.502209 92.168.247.3 124.19.26.72 107 74.495.24.8 [570] Seq4 Mindel Level 455.146 540.C [580] Seq Min	48 5.917562457	8.8.4.4	192.168.243.75	DNS	139 Standard query response 0x20ea SOA u	ass.edu SOA grid-soa.it.umass.edu OPT	
51 5 3222956 (192.105.241.75) 125.119.246.12 170" 74.4552 - 80 [SVII] Seq4 hund-242 (Leen M SS-1446 34CL/PENE 1 Tisal-38935285 Tiscret %5-128 52 5 45443991 129.105.241.75 125.119.246.12 170" 74.4552 - 80 [XVII] Seq4 hund-242 (Leen M SS-1446 34CL/PENE 1 Tisal-38935286 Tiscret %5-128 55 5 4545991 129.105.241.75 128.119.246.12 170" 74.862 - 48152 [SVIII, ACc) Seq4 Acc+ 1419.21964 Leen M SS-1466 34CL/PENE 1 Tisal-38935286 Tiscret 352.2483787 55 5 4545991 129.105.241.75 128.119.246.12 170" 64.41552 - 681 (ACc) Seq4 Acc+ 1419.21964 Leen M SS-1466 34CL/PENE 1 Tisal-38935246 Tiscret 352.2483787 56 6.44866109 122.119.245.12 129.105.241.75 170" 74.88 - 4552 (ACC) Seq4 Ack+ 76 hun-2880 Leen M SS-1466 34CL/PENE 1 Tisal-3952246 Sis-122 56 6.44866109 122.119.245.12 129.105.241.75 170" 64.8554 + 84.1252 (ACC) Seq4 Ack+76 hun-2880 Leen M SS-1466 34CL/PENE 1 Tisal-3952246 Sis-122 56 6.4486671 129.105.241.75 129.105.241.75 129.105.241.75 129.105.241.75 129.105.241.75 170" 64.8554 + 84.124 (ACC) Seq476 Ack+267 hun-2880 Leen M KS-1466 34CL/PENE 1 Tisal-3952246 (Tisc-13852246 HCF) respect of a reassembled F00] 66 6.4458671 129.105.241.75 129.105.241.75 129.105.241.75 129.105.241.75 129.105.241.75 129.105.241.75 129.105.241.	49 5.922656516	8.8.4.4	192.168.243.75	DNS	136 Standard query response 0x24cb SOA c	.umass.edu SOA unix1.cs.umass.edu OPT	
91 91 <t< td=""><td>50 5.930817495</td><td>8.8.4.4</td><td>192.168.243.75</td><td>DNS</td><td>409 Standard query response 0x6ce5 DS ed</td><td>DS RRSIG OPT</td><td></td></t<>	50 5.930817495	8.8.4.4	192.168.243.75	DNS	409 Standard query response 0x6ce5 DS ed	DS RRSIG OPT	
51 5 957777 124.19.24.12 124.18.24.7.5 172 7 4 8 - 4552 150, 464 50.7 164 9 150, 243.7.5 156 9 150, 243.7.5 156 9 150, 243.7.5 156 9 150, 243.7.5 156 9 150, 243.7.5 156 9 150, 243.7.5 157 9 150, 243.1.5 158 4 - 4552 (MC) Seq476 Active149 Nine4252 lateered Toxi=1252244843 Since-3352244431 158 540, 156 540, 15	51 5.932229956	192.168.243.75	128.119.245.12	TCP	74 41552 - 80 [SYN] Seq=0 Win=64240 Len-	0 MSS=1460 SACK_PERM=1 TSval=300582592 TSecr=0 WS=128	
51 5 3684447 192,104,217 102,109,240.7 102,109,240.7 102,109,240.7 102,109,240.7 102,109,240.7 102 104,100,250.7 <	52 5.945483988	192.168.243.75	128.119.245.12	TCP	74 41554 - 80 [SYN] Seq=0 Win=64240 Len-	0 MSS=1460 SACK_PERM=1 TSval=300582606 TSecr=0 WS=128	
55 55 <td< td=""><td>53 5.985787377</td><td>128.119.245.12</td><td>192.168.243.75</td><td>TCP</td><td>74 80 - 41552 [SYN, ACK] Seq=0 Ack=1 Win</td><td>=28960 Len=0 NSS=1460 SACK_PERM=1 TSval=3252843787 TSecr=300582592 WS=128</td><td></td></td<>	53 5.985787377	128.119.245.12	192.168.243.75	TCP	74 80 - 41552 [SYN, ACK] Seq=0 Ack=1 Win	=28960 Len=0 NSS=1460 SACK_PERM=1 TSval=3252843787 TSecr=300582592 WS=128	
56 6 webser/set 122.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.119.245.12 121.	54 5.985848447	192.168.243.75	128.119.245.12	TCP	66 41552 - 80 [ACK] Seq=1 Ack=1 Win=642	6 Len=0 TSval=300582646 TSecr=3252843787	
51 6. W10900 120.140.240.7 121.19.240.1 107 64 41546 + 80 [ACX] Sept A Advi M InveH236 (Leve) TSIN0-1308326404 TSICer:3525244384 55 6. AH14507 122.119.245.1 152.164 240.7 170 64 4152 + 80 [ACX] Sept A Advi M InveH236 (Leve) TSIN0-1308326404 TSICer:35253443 (TSCr:38525464 (TCP segment of a reassenbled POU) 66 6. AH146110 122.119.245.1 120.16 24.01.7 170 64 4152 + 80 [ACX] Sept A Advi M InveH236 (Leve) TSIN0-130832 (TSCr:2325248343 TSCr:3852564 (TCP segment of a reassenbled POU) 66 6. AH1667 (Leve) TSIN0-13083 (Leve)	55 5.986083608	192.168.243.75	128.119.245.12	HTTP	541 GET /wireshark-labs/HTTP-ethereal-la	-file3.html HTTP/1.1	
51 6 4446397 122.119.245.12 192.168.247.5 172 66 84 1552 102.101.20	56 6.000066899	128.119.245.12	192.168.243.75	TCP	74 80 - 41554 [SYN, ACK] Seq=0 Ack=1 Win	=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3252843801 TSecr=300582606 WS=128	
916 6.444108 122.119.245.12 192.168.240.75 TO 154.88 - 4452. (AGI) Seq-174 Acta-06 Win-34480 Len-124254444134 Tacc-1385226446 [TO* sequent of a reassenbled POU] 86 6.44158271 122.119.245.12 122.119.245.12 122.119.245.12 121.119.245.12 TO 151.48 a - 4522. (AGI) Seq-174 Acta-076 Win-34480 Len-1448 TSval-12525444841 Tscc-138552646 [TO* sequent of a reassenbled POU] 86 6.4465941 122.119.245.12 122.119.245.12 170 151.48 a - 4552. (AGI) Seq-174 Acta-076 Win-34808 Len-1448 TSval-12525443431 Tscc-138552646 [TO* sequent of a reassenbled POU] 86 6.4465741 122.119.245.12 170.18.44.75 TO 151.48 a - 4552. (AGI) Seq-1747 Acta-027 Win-64282 Len-1484 TSval-12526443431 Tscc-138552646 [TO* sequent of a reassenbled POU] 86 6.4466774 122.119.245.12 170 64.4552 + 80 (AGI) Seq-1747 Acta-026 Win-34802 Len-1448 TSval-12526443431 Tscc-13855264 [TO* sequent of a reassenbled POU] 85 6.4466774 122.119.245.12 170.18.48 - 4552. (AGI) Seq-1747 Acta-026 Win-34802 Len-1448 TSval-12526443431 Tscc-13852264 [TO* sequent of a reassenbled POU] 85 6.4466774 122.119.245.12 192.168.24.37 Win 14.85 (XGI) Seq-1476 Acta-028 Win 48.45 (Win-34801 Len-148 TSval-125244444 [Win-470] 85 6.1410 121.119.245.12 Win 14.119 Win 14.118 Win 14.118 Win 14.118 85 6.1410 121.119.245.12 W	57 6.000109046	192.168.243.75	128.119.245.12	TCP	66 41554 - 80 [ACK] Seq=1 Ack=1 Win=642	6 Len=0 TSval=300582660 TSecr=3252843801	
46 6.4198217 192.164.343, 75 112.119.245.12 170* 64 4552 + 80 (ACX) Seq-126 Active149 Nine-41282 Line=0 This-1-08552172 Tiscr=232243843 61 6.44655766 123.119.245.12 192.168.243, 75 TCP 1514 80 - 41552 (ACX) Seq-126 Active149 Nine-41282 Line=0 This-1-08552174 Tiscr=232243843 62 6.44665541 123.119.245.12 192.168.243, 75 TCP 1514 80 - 41552 (ACX) Seq-126 Active149 Nine-41282 Line=0 This-1-08552740 Tiscr=232243843 63 6.44665744 123.119.245.12 192.168.243, 75 TCP 1514 80 - 41552 (ACX) Seq-2297 Active176 Nine-30882 Line=1448 Tixval=35522464 (TCP segment of a reassembled PDU) 64 6.467238 123.119.245.12 192.168.243, 75 TCP 1514 80 - 41552 (ACX) Seq-297 Active176 Nine-30882 Line=1448 Tixval=35522464 (TCP segment of a reassembled PDU) 65 6.44665718 123.119.245.12 107 Lines 245, 75 TCP 551 481 Pitty 1.1 248 Dix Sequent d(4228 bits), on interface updat2chall, id 0 75 554 199242 128.119.245.12 107.168.243, 75 NTT 553 HTP/1.1 248 Dix Sequent d(4238 bits) on interface updat2chall, id 0 81 51 50 cc 119.245.12 119.245.12 119.245.12 119.245.12 119.245.12 81 51 50 cc 119.245.12 119.245.12 119.245.12 119.245.12 119.245.12 81 50 Tix obs 242 7	58 6.041458077	128.119.245.12	192.168.243.75	TCP	66 80 - 41552 [ACK] Seq=1 Ack=476 Win=3	080 Len=0 TSval=3252843842 TSecr=300582646	
61 6.4665790 122.119.245.12 192.164.247.5 TP 154.88 - 41552 (AC) Sec_1149 A 4ck-406 Kiin-30803 Lene-1448 Txiun-12522418441 Taccr-108152016 (TP segment of a reassenbled PDU) 63 6.4665744 128.119.245.12 121.119.245.12 TP 64.1552 . # (AC) Sec_126 A 2ck-75 Kiin-30803 Lene-1448 Txiun-12522418441 Taccr-10815207 Tiscr-202228443 64 6.44627288 129.108.431.75 128.119.245.12 TP 64.1552 . # (AC) Sec_126 A 2ck-455 Kiin-30803 Lene-1448 Txiun-1252243434 Tscr-208253646 (TP segment of a reassenbled PDU) 65 6.4466774 129.108.431.75 129.108.431.75 129.108.431.75 HTP 136 M 2ch (Tscr/2017 Viscr/2012 Viscr/2018 Viscr/20	59 6.041461169	128.119.245.12	192.168.243.75	TCP	1514 80 - 41552 [ACK] Seq=1 Ack=476 Win=3	080 Len=1448 TSval=3252843843 TSecr=300582646 [TCP segment of a reassembled PDU]	
42.6.444095511 102.164.341,75 110.246.12 TCP 66.44522 180.42(1) Sep-476.4ki-2307 Tion 120041232 Tion 2252243843 35.6.444657431 123.119.245.12 119.246.12 119.246.12 119.246.122 Long Tion 12004	60 6.041509217	192.168.243.75	128.119.245.12	TCP	66 41552 - 80 [ACK] Seq=476 Ack=1449 Win	=64128 Len=0 T5val=300582702 T5ecr=3252843843	
61 6.46672441 128.119.245.12 119.245.12 112.119.245.12 113.119.24	61 6.046656796	128.119.245.12	192.168.243.75	TCP	1514 80 - 41552 [ACK] Seq=1449 Ack=476 Win	=30088 Len=1448 TSval=3252843843 TSecr=300582646 [TCP segment of a reassembled PDU]	
64 6.40202816 192.163.241.75 128.119.246.12 119.246.12 117 64 15527.47 A 42-4545 kine4338 Lene 3 326 Lene 3	62 6.046695543	192.168.243.75	128.119.245.12	TCP	66 41552 - 80 [ACK] Seq=476 Ack=2897 Win	=64128 Len=0 T5val=300582707 T5ecr=3252843843	
e5 6.44665778 128.19.245.12 192.163.243.75 HTTP 583 HTTP/1.1.289 OK (text/html) e55.541 bytes on wire (4328 bits), 541 bytes captured (4328 bits) on interface wipbi2Pului, id 0 f55.541 bytes on wire (4328 bits), 541 bytes (aptured (4328 bits) on interface wipbi2Pului, id 0 f56.25788 f56.2578 f66.27778 f56.25788 f66.27778 f56.2778 f58.28778 f66.2788 f578.287458 f58.287458 	63 6.046657443	128.119.245.12	192.168.243.75	TCP	1514 80 - 41552 [ACK] Seq=2897 Ack=476 Win	=30088 Len=1448 TSval=3252843843 TSecr=300582646 [TCP segment of a reassembled PDU]	
55: 541 bytes on wire (4128 birs), 541 bytes captured (4128 birs) on interface wiphs/20uld, id 0 0000 c4 e9 86 9f 80 80 80 ef 80 ef 80 80 80 80 80 ef 80 00 42 00 90 80 80 45 00 0000 0000 met 11, src: legr_4/2.6: (0 80 ef 98.42; 56; 50), Dir: 1p-Linfly frib-Ba (c4:ep:84:9f:8b) as) 0000 02 of ef 64 68 40 86 52 ef 72 c1 at 73 40 88 77 00 e .sk.w met 11, src: legr_4/2.6: (0 80 ef 98.42; 50), Dir: 1p-Linfly frib-Ba (c4:ep:84:9f:8b) as) 0000 02 of ef 64 68 40 86 52 ef 72 c1 at 73 40 88 77 00 e .sk.w matistion Control Protocol, Src Part: 4552, Dif Port: 80, Seq: 1, Ack: 1, Len: 475 04 of 45 98 49 77 00 71 65 43 20 478 05 61 62 20 62 61 62 20 66 65 33 26 67 7 0.107 /w Interhalm text Transfer Protocol 05 61 62 20 64 69 48 95 7 0.021 77 60 72 65 60 65 72 0.021 7 /w Interhalm 05 61 62 20 64 65 20 73 20 63 70 01 20 72 /w Interhalm 0000 06 61 62 20 64 69 65 70 0.021 7 /w Interhalm 05 61 62 20 64 66 57 20 40 66 77 0.021 7 /w Interhalm 0000 06 61 7 0 /w Interhalm 06 61 62 70 20 48 95 70 0.02 67 66 70 70 70 70 70 70 70 70 70 70 70 70 70	64 6.046723050	192.168.243.75	128.119.245.12	TCP	66 41552 - 80 [ACK] Seq=476 Ack=4345 Win	=63360 Len=0 T5val=300582707 T5ecr=3252843843	
met II, Src: Legn_22:R:06 (00 ef 804-22.6c:76), Dit: Tp-LinkT_9f:Bb.8a (24-87.48-9f:Bb.8a) 02 ef ef 6d 49 ef 86 6 2e 73 ce as f3 4b 88 77 0 e s	65 6.046657738	128.119.245.12	192.168.243.75	HTTP	583 HTTP/1.1 200 OK (text/html)		
met II, Src: Legn_22:R:06 (00 ef 804-22.6c:76), Dit: Tp-LinkT_9f:Bb.8a (24-87.48-9f:Bb.8a) 02 ef ef 6d 49 ef 86 6 2e 73 ce as f3 4b 88 77 0 e s						2000 rd ag 2d of 2h	
mail Protocol Version 4, Str: 192.116, 243.75, p. ttr. 122.119.246.12 10 for 42 as equivalent to 10 for 42 as equivalent to 10 for 40 as 10 as 71 and 84 as 10 as 71 and 71 as 10 as 1							
atistion Control Protocol, Src Part: 41552, Dst Part: 80, Seq. 1, Ack: 1, Lem: 475 000 47 45 54 50 21 77 60 72 65 72 66 17 26 60 u GET //e instant text Transfer Protocol 000 47 45 54 50 21 77 7 60 72 66 17 26 60 u GET //e instant text Transfer Protocol 000 47 45 54 50 21 77 60 72 66 17 26 60 u GET //e instant 000 47 45 54 50 21 77 60 72 66 17 26 60 u GET //e instant 000 47 45 54 50 21 77 60 72 66 17 26 00 u GET //e instant 000 47 45 54 50 21 77 60 72 66 17 26 00 u GET //e instant 000 47 45 54 50 21 77 60 72 66 17 26 00 u GET //e instant 000 47 45 54 50 21 77 60 72 66 17 26 00 u GET //e instant 000 47 10 10 10 10 10 10 10 10 10 10 10 10 10					0:03 (C4:09:04:91:0D:03)		
text Transfer Protocol 20 6c 01 02 73 27 44 54 54 59 20 65 74 06 65 72 -labs/14T Th-ether text Transfer Protocol 00 6c 01 02 73 27 44 55 4 59 20 65 74 06 65 72 -labs/14T Th-ether 00 6c 01 02 73 27 45 54 59 20 65 74 06 55 72 10 10 110 710 710 710 710 710 710 710 71							
Bit Dis C 20 C			41552, DSt POIt. 80,	seq. I, A	A. 1, Len. 475	0050 2d 6c 61 62 73 2f 48 54 54 50 2d 65 74 68 65 72 -labs/HT TP-ether	
entr 74 3a 28 75 66 66 73 75 21 15 ala, cs.umass entr 28 67 66 76 67 67 76 76 66 76 67 67 76 76 66 76 76 67 67 67 67 67 67 66 76 67 67 67 67 67 66 76 67 67 67 67 66 76 67 67 67 67 67 67 66 76 67	text fighter P	101001					
Book Zer 65 64 75 68 49, 43 67 Fer for 6 for 63 37 46 96 76 66 -exit Connection Mail Ja Ja 20 31 68 49, 453 76 67 76 76 66 76 78 24 69 47 66 71.1 (bg grade-1n New Ja Ja 20 31 68 49, 55 76 67 72 (51 64 65 20 49 66 71.1 (bg grade-1n New Ja 36 73 72 65 23 75 55 23 75 55 73 75 65 73 74 73 Ja Security Ja 36 78 78 78 78 78 78 78 78 78 78 78 78 78							
0000 54 3a 28 31 66 4b 35 57 06 67 72 61 64 65 2d 45 66 T: 1 Up grade-In 0000 73 65 63 75 72 65 73 65 71 75 66 73 74 73 3a secure-8 equests: 0000 80 35 72 65 73 26 57 12 65 67 73 74 3a secure-8 equests: 0000 80 35 72 65 73 26 41 67 65 66 74 3a 28 1 User -Agent: 0000 80 37 76 76 66 73 76 72 65 73 74 76 75 76 76 77 77 3a Secure-8 equests: 0000 80 37 76 76 66 77 72 67 78 76 78 76 77 73 75 76 77 77 73 75 Secure-8 equests: 0000 80 37 76 76 66 77 78 72 86 78 73 76 77 76 77 78 78 78 76 76 70 77 77 78 78 78 78 76 76 70 78 70 78 78 78 75 78 78 78 78 78 78 78 78 78 78 78 78 78							
ence 72 65 63 75 72 65 24 52 65 72 75 65 73 77 47 3 2a 5 secure-k equents: 6002 26 31 60 46 55 73 65 72 24 44 67 55 66 74 3a 24 8 1.00 secure-Agent: 6004 44 67 74 69 66 66 61 27 15 2a 20 26 28 58 31 31 Mozillar 5.0 (011 6001 3b 24 64 66 66 75 73 72 67 28 73 85 35 63 43 42 20 8 1.100x 186,64)							
0000 20 31 00 00 57 56 57 26 57 2 20 41 67 65 56 74 3a 20 1. User -Agent: 0000 46 67 7a 69 65 65 61 27 35 2a 30 2a 28 58 31 31 Mozilla/ 5.0 (X11 0010 3b 20 46 66 67 37 57 2a 78 2a 78 34 29 2a 34 20 2a 34 34 20							
00f0 35 20 4c 69 6e 75 78 20 78 38 36 5f 36 34 29 20 ; Linux x86_64)						00d0 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 1 User -Agent:	

• Since this lab is about Ethernet and ARP, we're not interested in IP or higher-layer protocols. So let's change Wireshark's "listing of captured packets" window so that it shows information only about protocols below IP. To have Wireshark do this, select *Analyze->Enabled Protocols*. Then uncheck the IPv4 box (use the filter at the top) and select OK. You should now see a Wireshark window that looks like the following:

ile Edit View Go C								
1 🔲 🔬 🎯 土	🗋 🕅 🙆 🔍				•			
								C 3 ·
50 5.930817495	Tp-LinkT_9f:8b:0a	Legra_42:0c:9b	0×0800	409 IPv4				
	Legra_42:0c:9b		0x8898					
52 5.945483988	Legra_42:0c:9b		0x8890					
	Tp-LinkT_9f:8b:0a		0x0800					
54 5.985840447	Legra_42:0c:9b		0x0800					
55 5.986083608			0x8800					_
56 6.000066899	Tp-LinkT_9f:8b:0a	Legra_42:0c:9b	0x8800	74 IPv4				
57 6.000109046	Legra_42:0c:9b	Tp-LinkT_9f:8b:0a	0x0800	66 IPv4				
58 6.041458077	Tp-LinkT_9f:8b:0a	Legra_42:0c:9b	0x8800	66 IPv4				
			0x8890					
	Legra_42:0c:9b	Tp-LinkT_9f:8b:0a	0x8898	66 IPv4				
61 6.046656796	Tp-LinkT_9f:8b:0a	Legra_42:0c:9b	0x0800	1514 IPv4				
62 6.046695543	Legra_42:0c:9b	Tp-LinkT_9f:8b:0a	0x8800	66 IPv4				
	Tp-LinkT_9f:8b:0a	Legra_42:0c:9b	0x8890	1514 IPv4				
64 6.046723050	Legra_42:0c:9b	Tp-LinkT_9f:8b:0a	0x8800	66 IPv4				
65 6.046657738		Legra_42:0c:9b	0x0800					
		Tp-LinkT_9f:8b:0a	0x8880	66 IPv4				
67 6.089243946	Legra_42:0c:9b	Tp-LinkT_9f:8b:0a	0x889x9	236 IPv4				
68 6.123157165	Tp-LinkT_9f:8b:0a	Legra_42:0c:9b	0x0800	83 IPv4				
69 6.123159033	Tp-LinkT_9f:8b:0a		0x0800	60 IPv4				
70 6.123372154	Legra_42:0c:9b	Tp-LinkT_9f:8b:0a	0x8800	70 IPv4				
ma 55: 541 hutar a	o wire (4239 bits)	541 bytes captured (4)	970 bitcl		wlodelikulul id 8	0000 c4 e9 84 9f 8b 0a 00 0f 00 42 0c 9b 08 00 45 00		
		0:42:0c:9b), Dst: Tp-I				0010 02 0f e0 fd 40 00 40 06 2e 73 c0 a8 f3 4b 80 77		
a (527 bytes)	10_42.00.30 (00.01.0	0.42.00.50), 0st. 1p-1	.111K1_91.0	J.08 (C4.63.)	54.51.00.001	0020 f5 0c a2 50 00 50 38 60 1e ce 36 d8 9f ad 80 18		
a (527 bytes)						0030 01 f6 e8 fe 00 00 01 01 08 0a 11 ea 86 f6 c1 e2 0040 75 0b 47 45 54 20 2f 77 69 72 65 73 68 61 72 6b		
						0050 2d 6c 61 62 73 2f 48 54 54 50 2d 65 74 68 65 72		
						0060 65 61 6c 2d 6c 61 62 2d 66 69 6c 65 33 2e 68 74 0070 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73		
						0070 6d 6c 20 48 54 54 50 2T 31 2e 31 0d 0a 48 6T 73 0080 74 3a 20 67 61 69 61 2e 63 73 2e 75 6d 61 73 73		
						0090 2e 65 64 75 @d @a 43 6f 6e 6e 65 63 74 69 6f 6e		
						00a0 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 44 4e		
						00b0 54 3a 20 31 0d 0a 55 70 67 72 61 64 65 2d 49 6e 00c0 73 65 63 75 72 65 2d 52 65 71 75 65 73 74 73 3a		
						00d0 20 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20		
						00e0 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31		
						00f0 3b 20 4c 69 6e 75 78 20 78 38 36 5f 36 34 29 20		
						0100 41 70 70 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 0110 33 36 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20		
						0120 47 65 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 37 39		
winschark winterfeite	u1 20200112154530 16vPvk r						71 (100.0%) · Dropped: 0 (0.0%)	Profile: Def

In order to answer the following questions, you'll need to look into the packet details and packet content windows (lower left and right panels on the screenshots. You could have a different arrangement).

Select the Ethernet frame containing the HTTP GET message. (Recall that the HTTP GET message is carried inside of a TCP segment, which is carried inside of an IP datagram, which is carried inside of an Ethernet frame; reread section 1.5.2 in the text if you find this encapsulation a bit confusing). Expand the Ethernet II information in the packet details window. Note that the contents of the Ethernet frame (header as well as payload) are displayed in the packet contents window.

Answer the following questions, based on the contents of the Ethernet frame containing the HTTP GET message. 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* \rightarrow *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 48-bit Ethernet address of your computer?
- 2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? (Hint: the answer is no). What device has this as its Ethernet address? [Note: this is an important question, and one that students sometimes get wrong. Re-read pages 468-469 in the text and make sure you understand the answer here.]
- 3. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?
- 4. How many bytes from the very start of the Ethernet frame does the ASCII "G" in "GET" appear in the Ethernet frame?

Next, answer the following questions, based on the contents of the Ethernet frame containing the first byte of the HTTP response message.

- 5. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu (Hint: the answer is no). What device has this as its Ethernet address?
- 6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?
- 7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?
- 8. How many bytes from the very start of the Ethernet frame does the ASCII "O" in "OK" (i.e., the HTTP response code) appear in the Ethernet frame?

The Address Resolution Protocol

In this section, we'll observe the ARP protocol in action. We strongly recommend that you re-read section 6.4.1 in the text before proceeding.

ARP Caching

Recall that the ARP protocol typically maintains a cache of IP-to-Ethernet address translation pairs on your computer. The arp command (in both MSDOS and Linux/Unix) is used to view and manipulate the contents of this cache. Since the arp command and the ARP protocol have the same name, it's understandably easy to confuse them. But keep in mind that they are different - the arp command is used to view and manipulate the ARP cache contents, while the ARP protocol defines the format and meaning of the messages sent and received, and defines the actions taken on message transmission and receipt.

Let's take a look at the contents of the ARP cache on your computer:

• **MS-DOS**. The arp command is in *c:\windows\system32*, so type either "arp" or "c:\windows\system32\arp" in the MS-DOS command line (without quotation marks).

³ What is meant by "annotate"? Please highlight and annotate where you've found answers and add an explaination of what you've found in what you've highlighted.

• Linux/Unix/MacOS. The executable for the arp command can be in various places. Popular locations are */sbin/arp* (for linux) and */usr/etc/arp* (for some Unix variants).

If on Windows try arp -a. If on Linux try arp -n. This will display the current ARP cache on your computer.

9. Write down the contents of your computer's ARP cache. What is the meaning of each column value?

In order to observe your computer sending and receiving ARP messages, we'll need to clear the ARP cache, since otherwise your computer is likely to find a needed IP-Ethernet address translation pair in its cache and consequently not need to send out an ARP message.

- **MS-DOS**. The MS-DOS "arp -d *" command will clear your ARP cache. The -d flag indicates a deletion operation, and the * is the wildcard that says to delete all table entries.
- Linux/Unix/MacOS. The "ip -s -s neigh flush all" on Linux will clear your ARP cache. You may have to elevate yourself to root to run this command.

Observing ARP in Action

Do the following⁴:

- Clear your ARP cache, as described above.
- Next, make sure your browser's cache is empty.
- Start up the Wireshark packet sniffer
- Enter the following URL into your browser <u>http://gaia.cs.umass.edu/wireshark-labs/HTTP-</u> <u>wireshark-lab-file3.html</u>. Your browser should again display the rather lengthy US Bill of Rights.
- Stop Wireshark packet capture. Again, we're not interested in IP or higher-layer protocols, so change Wireshark's "listing of captured packets" window so that it shows information only about protocols below IP. To have Wireshark do this, select *Analyze->Enabled Protocols*. Then uncheck the IPv4 box and select OK. You should now see a Wireshark window that looks like the following:

⁴ The *ethernet-ethereal-trace-1* trace file in <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u> was created using the steps below (in particular after the ARP cache had been flushed).

a display filter <				engt Info		
103 14.803642177	fe80::7598:6dea:16f5:	. ff02::1:2	DHCPv6	168 Solicit XID: 0x1cff87 CID: 000100012504ad63b8ca3a92ce9d		-
182 21.385700289	Apple_f1:dd:c1	Broadcast	ARP	60 ARP Announcement for 146.163.150.100		
81 21.291511058	Apple_f3:52:a6	Broadcast	ARP	60 ARP Announcement for 146.163.150.106		
80 21.203325255	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.218? Tell 146.163.150.252		
79 21.138671130	Apple_f1:de:0a	Broadcast	ARP	60 ARP Announcement for 146.163.150.101		
78 21.130309352	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.195? Tell 146.163.150.252		
77 21.100545838	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.184? Tell 146.163.150.252		
76 21.068022486	Apple_f1:dd:a3	Broadcast	ARP	60 ARP Announcement for 146.163.150.99		
74 20.507737943	Apple_f0:de:ba	Broadcast	ARP	60 ARP Announcement for 146.163.150.97		
50 19.464948491	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.184? Tell 146.163.150.252		
35 18.131087365	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.117? Tell 146.163.150.252		
34 18.130773869	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.218? Tell 146.163.150.252		
31 18.070151821	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.195? Tell 146.163.150.252		
28 17.508220102	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.195? Tell 146.163.150.252		
05 15.070257049	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.117? Tell 146.163.150.252		
04 15.070012733	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.218? Tell 146.163.150.252		
87 13.728718547	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.218? Tell 146.163.150.252		
84 12.030266793	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.117? Tell 146.163.150.252		
83 11.846923623	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.117? Tell 146.163.150.252		
82 11.470488193	Avaya_a3:05:1b	Broadcast	ARP	60 Who has 146.163.150.27? Tell 146.163.150.253		
69 8.430068955	Avaya_a3:05:1b	Broadcast	ARP	60 Who has 146.163.150.27? Tell 146.163.150.253		
51 5.369820946	Avaya_a3:05:1b	Broadcast	ARP	60 Who has 146.163.150.27? Tell 146.163.150.253		
21 3.073593287	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.131? Tell 146.163.150.252		
17 2.712840122	Avaya_a3:05:1b	Broadcast	ARP	60 Who has 146.163.150.27? Tell 146.163.150.253		
1 0.000000000	Avaya_a3:0d:1b	Broadcast	ARP	60 Who has 146.163.150.131? Tell 146.163.150.252		
182: 60 bytes on v	vire (480 bits), 60 bytes o	aptured (480 bits) o	n interface enp0s31f6	263 ID-4	0000 ###########98 10 e8 f1 dd c1 08 06 00 01	
et II, Src: Apple_f1	:dd:c1 (98:10:e8:f1:dd:c	1), Dst: Broadcast (fl			0010 08 00 06 04 00 01 98 10 e8 f1 dd c1 92 a3 96 64d 0020 00 00 00 00 00 00 92 a3 96 64 00 00 00 00 00d	
ss Resolution Prote	col (ARP Announcemer					
rdware type: Ether	net (1)					
tocol type: IPv4 (0	×0800)					
rdware size: 6						
tocol size: 4						
gratuitous: True]						
announcement: Tr						
	Apple_f1:dd:c1 (98:10:e	8:f1:dd:c1)				
nder IP address: 14	6 163 150 100					

Answer the following questions:

- 10. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP request message?
- 11. Give the hexadecimal value for the two-byte Ethernet Frame type field. What upper layer protocol does this correspond to?
- 12. Download the ARP specification from <u>ftp://ftp.rfc-editor.org/in-notes/std/std37.txt</u>. A readable, detailed discussion of ARP is also at <u>http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html</u>.
 - a. How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?
 - b. What is the value of the opcode field within the ARP-payload part of the Ethernet frame in which an ARP request is made?
 - c. Does the ARP message contain the IP address of the sender?
 - d. Where in the ARP request does the "question" appear the Ethernet address of the machine whose corresponding IP address is being queried?
- 13. Now find the ARP reply that was sent in response to the ARP request.
 - a. How many bytes from the very beginning of the Ethernet frame does the ARP *opcode* field begin?
 - b. What is the value of the *opcode* field within the ARP-payload part of the Ethernet frame in which an ARP response is made?
 - c. Where in the ARP message does the "answer" to the earlier ARP request appear the IP address of the machine having the Ethernet address whose corresponding IP address is being queried?
- 14. What are the hexadecimal values for the source and destination addresses in the Ethernet frame containing the ARP reply message?
- 15. Open the ethernet-ethereal-trace-1 trace file in <u>http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip</u>. The first and second ARP packets in this trace correspond to an ARP request sent by the computer running Wireshark, and the ARP reply sent to the computer running Wireshark by the computer with the ARP-requested Ethernet address. But there is yet another computer on

this network, as indicated by packet 6 - another ARP request. Why is there no ARP reply (sent in response to the ARP request in packet 6) in the packet trace?

Extra Credit

1. The arp command:

arp -s InetAddr EtherAddr

allows you to manually add an entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. What would happen if, when you manually added an entry, you entered the correct IP address, but the wrong Ethernet address for that remote interface?

2. What is the default amount of time that an entry remains in your ARP cache before being removed. You can determine this empirically (by monitoring the cache contents) or by looking this up in your operation system documentation. Indicate how/where you determined this value.