

计算机网络课程设计指导书

《计算机网络》课程组 编著

上海海洋大学海洋智能信息实验教学示范中心

目录

实验一	:初识 WIRESHARK3
实验二	: 802.3 协议分析和以太网7
实验三	: PING 命令初探11
实验四	: IP 层协议分析20
实验五	: TCP 协议分析25
实验六	: HTTP 和 DNS 分析 31
实验七	: 利用 Ethereal 分析 ARP 协议35
实验八	:利用 Ethereal 分析 HTTP 协议40

实验 一 初识 WIRESHARK

一、实验目的

通过实验,了解实验环境 WIRESHARK 的基本使用,观察 PING 命令时抓包情况,观察登陆某一网站时抓包情况。

二、实验环境

WIRESHARK

三、实验理论

分组是目前分组交换网络中传输的格式化数据块,是基本的信息传输单位。它与数据报意义近似,但有微小的区别。分组是一个泛指词,而数据报往往用于不可靠服务场合。分组由控制信息和用户数据(payload)构成。

Wireshark(前称 Ethereal)是一个网络封包分析软件。网络封包分析软件的功能 是撷取网络封包,并尽可能显示出最为详细的网络封包资料。Wireshark 使用 WinPCAP 作为接口,直接与网卡进行数据报文交换。在 GNUGPL 通用许可证的保 障范围底下,使用者可以以免费的代价取得软件与其源代码,并拥有针对其源代 码修改及客制化的权利。Ethereal 是目前全世界最广泛的网络封包分析软件之 一。

四、实验内容

1. 使用 WireShark 软件过滤, 抓取封包

四、实验步骤

1. 使用 WireShark 软件过滤, 抓取封包

1.1 教师演示 WireShark 抓包过程及过滤器使用方法(略)。

1.2 在 Windows 的命令提示符界面中输入命令: ipconfig /all, 会显示本机的 网络信息。

1.3 观察运行结果,获得本机的以太网地址, IP 地址。如图 1 所示

	Connection-specific	D	4S	Su	ıf	Fi	ĸ	:	
	Description	•						:	Intel(R) PRO/1000 PL Network Connect
ion									
	Physical Address							:	00-15-58-2F-7E-7E
	Dhcp Enabled							:	No
	IP Address							:	172.16.1.98
	Subnet Mask							:	255.255.255.0
	Default Gateway							:	172.16.1.1
	DNS Services							:	172 16 2 25

4 从 http://openmaniak.com/cn/wireshark_use.php#menus, 学习 WIRESHAR
 K 软件使用方法,主要是过滤条件的设置。

1.5 从 http://product.pconline.com.cn/itbk/wlbg/network/1305/3305248.html#ad=7094, 学习 PING 命令的用法,主要是该命令各个参数的设置。

1.6 PING 旁边同学的 IP, 使用软件过滤, 抓取封包。

五、实验报告要求:

实验报告参考模板如下:



实验报告

题目:

- 学 院: 信息学院
- 专业:
- 班级:
- 学号:
- 姓名:

年月日

5 / 43

一、实验目的

通过实验,了解实验环境 WIRESHARK 的基本使用,观察 PING 命令时抓包情况,观察登陆某一网站时抓包情况。

二、实验环境

WIRESHARK

三、实验内容

四、实验步骤(图文方式叙述)

五、实验结果及分析(遇到的问题与解决)

六、实验体会

(可以从下面两个方面作答: 1、MAC 地址和 IP 地址的区别是什么? 2、给你的某一同 学发送封包,使用软件的过滤器,观察该 IP 的封包,说明你发送包的大小及发送 IP。)

实验 二 802.3 协议分析和以太网

一、实验目的

分析 802.3 协议, 熟悉以太网帧的格式。

二、实验环境

WIRESHARK

三、实验内容

1. 俘获并分析以太网帧

四、实验步骤

1. 俘获并分析以太网帧

1.1 在 IE 窗口中,选择"工具/Internet 选项/删除文件"命令,清空浏览器缓存。如图1和图2所示

nterr	tet 选项	Ő.					?
常规	安全	隐私	内容	连接	程序	高级	
	「 可以 出 地址	更改主页 :(&): h 使用当前	。 ttp://ww ī页 (C)	w. baidu. 使用默i	com/ 人页 @)	使用空白	页 (B)
Int	ernet 临 所查 提高 删除	时文件 看的 Int 以后浏览 : Cookies	ernet 页 的速度。 ;(I)	存储在特册除文件	定的文件 * (2)	夹中,这样 	¥可以 9
历史	2记录 "Hi 速访 列页保存i	story" 何最近查 在历史记:	2件夹中在 看过的页 录中的天	见含有已ù 数 (<u>K</u>): [:	5问页的制	蒈接,可使 清除历史i	用户快
آش (色(Q)	_ [字	体(N)	_ [语]	Ē (L)	辅助功	能(2))

图 1



图 2

1.2 启动 WIRESHARK, 开始分组俘获。如图 3 所示

文件 (2) 編編 (2) 観田 (2) 完位 (2) 新田 (2) 102.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 新田 (2) 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129		Capt	uri	ng	from	VIIva	ire .	Acc	eler	ated	ATD	PCNe	t Ad	lapte	er ((licr	osof	't' s	Pac	ket	Sch	edul	L		
Image: Source Destination Protocol Image: Source Protocol Image: Source <t< td=""><td>文作</td><td>‡(E)</td><td>编</td><td>辑(E)</td><td>视</td><td>图 (V)</td><td>定位</td><td>(G)</td><td>抓包</td><td>(C)</td><td>分析(A</td><td>) 统i</td><td>+ (S)</td><td>电信</td><td></td><td>工具</td><td>(T)</td><td>帮助</td><td>(H)</td><td></td><td></td><td></td><td></td><td></td><td></td></t<>	文作	‡(E)	编	辑(E)	视	图 (V)	定位	(G)	抓包	(C)	分析(A) 统i	+ (S)	电信		工具	(T)	帮助	(H)						
対流: ・ 表达式 清除 应用 No. Time Source Destination Protocol Info 2 1.499945 192,168,74,129 192,168,74,2 NENS Refresh NB <01><02>MSBRC 3 3,000282 192,168,74,129 192,168,74,2 NENS Refresh NB <01><02>MSBRC 4 4,496649 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 5 5,989980 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 6 7,495026 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 7 8,995408 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 8 10,495347 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 9 11,995320 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 10 13,495644 192,168,74,129 192,168,74,2 NENS Refresh NB PBW-9CB3FB9E976 11 14,990983 192,168,74,129 192,168,74,2 NBNS Refresh NB MSHOME<1d> 12 16,491069 192,168,74,129 192,168,74,2 NBNS Refresh NB MSHOME<1d>				0				×	2	<u>-</u>	0, 4	• •		7	₽			•	Q		•••		1	1 🖪	»
No. Time Source Destination Protocol Info 2 1.499945 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>NBRC 3 3.000282 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>NBRC 4 4.496649 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 5 5.989980 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 6 7.495026 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 7 8.995408 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 8 10.495347 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.3995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS	Ì	t滤:														• ₹	赵式			余	应	用			
1 0.0000000 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>MSBRC 2 1.499945 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>MSBRC 3 3.000282 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>MSBRC 4 4.496649 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 5 5.989980 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 6 7.495026 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 7 8.995408 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 8 10.495347 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.1	No.		Ti	ne		Sour	ce				Dest	inatio	n			Prot	ocol	Info							~
2 1.499945 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>MSBRC 3 3.000282 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>MSBRC 4 4.496649 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 5 5.989980 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 6 7.495026 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 7 8.995408 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 8 10.495347 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> # Erthernet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) # Internet Protocol, Src: 192.168.74.129 (192.168.74.129), Dst: 192.168.74.2 (192.168.74.	I		1 U.		JUU	192	.168	./4	.129		192	.168.	74.2			NBI	NS.	Ref	resh	NB	<01.	><0	2>_	_IMSB	RO
3 3.000282 192.168.74.129 192.168.74.2 NBNS Refresh NB <01><02>MSBRC 4 4.496649 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976<			21.	4999	945	192	. 168	.74	.129		192	. 168.	/4.2	-		NBI	NS	Ref	resh	NB	<01	><0	2>_		RC
4 4.496649 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976<			33.	0002	282	192	. 168	./4	.129		192	.168.	/4.2			NBI	NS:	Ref	resh	NB	<01	><0	2>_	_MSB	RC
5 5,989980 192,168.74.129 192,168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 6 7,495026 192,168.74.129 192,168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 7 8,995408 192,168.74.129 192,168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 8 10.495347 192,168.74.129 192,168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.995320 192,168.74.129 192,168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192,168.74.129 192,168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 11 14.990983 192,168.74.129 192,168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192,168.74.129 192,168.74.2 NBNS Refresh NB MSHOME<1d> IP Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) IBNS Refresh NB MSHOME<1d> IP Lehemet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) Imternet Protocol, Src: 192,168.74.129 (192,168.74.129), Dst: 192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2 (192,168.74.2		4	44.	4966	549	192	. 168	.74	.129		192	. 168.	74.2	2		NBI	NS	Ref	resh	NB	PBW	-9CE	33FE	39E97	6<
6 7.495026 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 7 8.995408 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 8 10.495347 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> I2 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> I2 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> II 11 bytes on wire (880 bits), 110 bytes captured (880 bits) III III IIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII			55.	9899	980	192	. 168	.74	.129		192	. 168.	74.2	!		NBI	NS	Ref	resh	NB	PBW	-9CE	33FE	39E97	6<
7 8.995408 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 8 10.495347 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> IP Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) IP Ethernet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) IP Ethernet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) IP User Datagram Protocol, Src Port: netbios-ns (137) 0000 00 50 56 e1 2e 9e 00 0c 29 75 e5 5b 08 00 45 00 .PV, Ju.[E. 0000 00 00 00 00 00 01 20 41 42 41 43 46 50 46 50 45 A BACFPFPE 0000 00 00 00 00 00 01 20 41 42 41 43 46 50 46 50 45 A BACFPFPE IP ETM: In feat14	_	6	57.	4950	026	192	. 168	.74	.129		192	.168.	74.2	2		NBI	NS	Ref	resh	NB	PBW	-9CE	33FE	39E97	6<
8 10.495347 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 9 11.995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> I2 11 11.0 bytes on wire (880 bits), 110 bytes captured (880 bits) III IIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII			78.	9954	108	192	. 168	.74	.129		192	. 168.	74.2	2		NB	NS	Ref	resh	NB	PBW	-9CE	33FE	39E97	6<
9 11.995320 192.168.74.129 192.168.74.2 NBNS Refresh NB PBW-9CB3FB9E976 10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> II 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> II 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> II 14.991069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> II 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> II Internet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) II Internet Protocol, Src Port: netbios-ns (137) III II User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII		8	3 10),495	5347	192	.168	.74	.129		192	.168.	74.2	2		NBI	NS	Ref	resh	NB	PBW	-9CE	33FE	39E97	6<
10 13.495644 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 11 10 bytes on wire (880 bits), 110 bytes captured (880 bits) Image: Comparison of the time of tif time of tif time of time of time of time of time of		9	9 1:	1.995	5320	192	.168	.74	.129		192	.168.	74.2	2		NBI	NS	Ref	resh	NB	PBW	-9CE	33FE	39E97	6<
11 14.990983 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> 12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> Image: State of the state of th		10	0 13	3.495	5644	192	. 168	.74	.129		192	.168.	74.2	2		NBI	NS	Ref	resh	NB	MSH	OME	E<10	d>	
12 16.491069 192.168.74.129 192.168.74.2 NBNS Refresh NB MSHOME<1d> Image: State of the state of t		1	1 14	1.990	0983	192	. 168	.74	.129		192	.168.	74.2			NBI	NS	Ref	resh	NB	MSH	OME	E<10	d>	
Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) Ethernet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) Internet Protocol, Src: 192.168.74.129 (192.168.74.129), Dst: 192.168.74.2 (192.168.74.2) User Datagram Protocol, Src Port: netbios-ns (137) O000 00 50 56 e1 2e 9e 00 0c 29 75 e5 5b 08 00 45 00 .PV,)u.[E. 0010 00 60 02 bf 00 00 80 11 21 fa c0 a8 4a 81 c0 a8 .`1J 0020 4a 02 00 89 00 4c 40 61 80 5b 40 00 00 11 JL @a.[@ 0030 00 00 00 00 01 20 41 42 41 43 46 50 46 50 45 BACFPFE Immediate Annotation Annotation Annotation Annotation Annotation Annotation Annotation Annotation		1.	2 16	5.49:	1069	192	.168	.74	.129		192	.168.	74.2	2		NBI	NS	Ref	resh	NB	MSH	OME	E<10	d>	V
 ■ Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) ■ Ethernet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) ■ Internet Protocol, Src: 192.168.74.129 (192.168.74.129), Dst: 192.168.74.2 (192.168.74.2) ■ User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) ■ User Datagram Protocol, Src 29:75 e5 5b 08 00 45 00 PV)u.[E. 0000 00 50 56 e1 2e 9e 00 0c 29:75 e5 5b 08 00 45 00 PV)u.[E. 0010 00 60 02 bf 00 00 80 11 21 fa c0 a8 4a 81 c0 a8 !J 0020 4a 02 00 89 00 9c 4d 61 80 5b 40 00 00 01 JL @a.[@ 0030 00 00 00 00 01 20 41 42 41 43 4550 46 50 45 A BACFPFPE 	< [>
Ethernet II, Src: Vmware_75:e5:5b (00:0c:29:75:e5:5b), Dst: Vmware_e1:2e:9e (00:50:56:e1:2e:9e) Internet Protocol, Src: 192.168.74.129 (192.168.74.129), Dst: 192.168.74.2 (192.168.74.2) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137), Dst Port: netbios-ns (1	⊞F	ram	ne 1	: 11) byt	es on	wire	(88	30 bit	s), 1	10 by	tes ca	aptur	red (880	bits)									^
Internet Protocol, Src: 192.168.74.129 (192.168.74.129), Dst: 192.168.74.2 (192.168.74.2) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: n	ΞE	the	met	Ш,	Src: ۱	/mwa	are_7	'5:e	5:5b	(00:	0c:29:	75:e5	:5b)	, Dst	t: Vr	mware	e_e1	:2e:9	e (C	0:5	0:56:	e1:2	e:96	∋)	
User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)	∃ ⊞	nter	net	Prot	ocol,	Src:	192.	168	.74.1	.29 (192.16	8.74	129), Ds	t: 1	92.16	8.74	1.2 (1	192.1	168.	74.2)				
0000 00 50 55 55 08 00 00)u.[E. 0010 00 60 02 bf 00 00 121 fa c0 a8 48 100 a8	Ξl	Jser	Da	tagra	am Pr	rotoco	ol, Sr	c Po	ort: n	etbio	s-ns (137),	Dst	Port	: ne	tbios-	ns (137)							
0000 00 50 55 55 50 00 60 0.5 0.5 6 12					-	- 2																			
0010 00 60 02 bf 00 00 80 11 21 fa c0 a8 4a 81 c0 a8 !J 0020 4a 02 00 89 00 8c 40 61 80 5b 40 00 00 01 JL @a.[@ 0030 00 00 00 00 01 20 41 42 41 43 46 50 46 50 45 A BACFPFPE	000	0 0	0 50	0 56	e1 2	e 9e	00 00	c 2	9 75	e5 5	5b 08 0	0 45	00	.PV)u.[Ε.								~
UU2U 4a U2 UU 89 UU 89 UU 4c 40 61 80 5b 40 00 00 01 J A BACFPFPE	001	0 0	0 60	0 02	bf 00	00 8	30 11	1 2	1 fa	c0 a8	3 4a 8:	1 c0 a	8	· _ · · · ·	u li	J	_								
UU3U UU UU UU UU UU UI ZU 41 42 41 43 40 5U 40 5U 45 A DAUFYFYE MRTH: 12 見ご始長、12 見ご始長、12 見ご始長、0 原常立体・D. C. 14	002	04	a 0:	2.00	89.0	0.891	00.4	c 4	0.61	80.5	6 40 C	10 00	01	J	L.	@a.[(0								-
「 Y NEW ALE ACCELED ACED AND ICATEL AUADLED COMPANIES IN AN IN	003	U U Vilwa	U Ul re Å	o UU ccele	rated	AMD P	ZU 4 CNet	1 4 Adat	241 ter (434	to SU 4)数量	tO DU 12 显え	45 【数量	: 12	. A 标记	DALF 数量:	PEPE 0	- 8	2署文	伴D	efault				×

图 3

1.3 在浏览器的地址栏中输入: http://shou.edu.cn。如图4所示



图 4

1.4 停止分组俘获。首先,找到你的主机向服务器 shou.edu.cn 发送的 HTTP GET 报文的分组序号,以及服务器发送到你主机上的 HTTP 响应报文的序号。如图 5 所示

🛛 VIIvare Accelerated AID PCNet Adapter (Iicrosoft's Packet Scheduler) - Vireshark	
文件(正)编辑(正)视图(Y)定位(G)抓包(C)分析(A)统计(S)电信(Y)工具(T)帮助(H)	
过滤:	
No. Time Source Destination Protocol Info	^
13 13:5234/1 128:119:245.12 192:168:74.129 TCP [TCP segment of a reassembled PDU]	
14 13:523564 192:168:74.129 128:119:245:12 1CP ansort-Im-1 > http://ACK.j.Seq=304.Ack=34/3 Win=642	.41
15 13.535446 128.119.245.12 192.168.74.129 TCP [TCP segment of a reassembled PDU]	
16 13:535501 128:119:245:12 192:168:/4.129 TCP [TCP segment of a reassembled PDU]	
17 13.535523 128.119.245.12 192.168.74.129 HTTP HTTP/1.1 200 OK (text/html)	
18 13.535591 192.168.74.129 128.119.245.12 TCP ansoft-Im-1 > http [ACK] Seq=304 Ack=7457 Win=642	24(
19_13.545303192.168.74.129128.119.245.12TCPansoft-Im-2 > http [SYN] Seq=0 Win=64240 Len=0 MS	SS
20 13.579086 192.168.74.129 128.119.245.12 HTTP GET /ethereal-labs/cover_small.jpg HTTP/1.1	
21 13.579370 128.119.245.12 192.168.74.129 TCP http > ansoft-Im-1 [ACK] Seq=7457 Ack=588 Win=642	24(
22 14.013774 128.119.245.12 192.168.74.129 TCP [TCP segment of a reassembled PDU]	
23 14.013829 128.119.245.12 192.168.74.129 TCP [TCP segment of a reassembled PDU]	
24 14.013851 128.119.245.12 192.168.74.129 TCP [TCP segment of a reassembled PDU]	×
	2
	^
₪ [SEQ/ACK analysis]	_
Hypertext Transfer Protocol	
GET /ethereal-labs/cover_small.jpg HTTP/1.1\r\n	
Reauest Method: GET	~
0000 00 50 56 e1 2e 9e 00 0c 29 75 e5 5b 08 00 45 00PV)u.[E.	~
0010 01 44 02 7a 40 00 80 06 76 8c c0 a8 4a 81 80 77 .D.z@vJw	-
0020 f5 0c 04 3b 00 50 cc bc 4e f9 07 01 6e e2 50 18;P., N.,n,P.	
UUSU TAITUISS 9/ UU UU 4/ 45 54 2U 2T 65 /4 68 65 /2SGE /6Ther 〇文体: "C.\TopTumy"(__\TopTumy"(_\)) [の数量: 108 見示数量: 108 長行数量: 0 手车数量: 0 同常文体: T.S)	×

1.5 选择 "Analyze->Enabled Protocols",取消对 IP 复选框的选择,单击 OK。如图 6 所示



图 6

五、实验报告要求:

按实验报告模板撰写实验报告

实验三: PING 命令初探

一、实验目的

掌握各种主要命令的作用;掌握各种网络命令的主要测试方法;理解各种网络命 令主要参数的含义。

二、实验环境

WIRESHARK

三、实验理论

在网络调试的过程中,常常要检测服务器和客户机之间是否连接成功、希望检查本地计算机和某个远程计算机之间的路径、检查 TCP/IP 的统计情况以及系统使用 DHCP 分配 IP 地址时掌握当前所有的 TCP/IP 网络配置情况,以便及时了解整个网络的运行情况,以确保网络的连通性,保证整个网络的正常运行。在 Windows 中提供了以下命令行程序:

(1) ping: 用于测试计算机之间的连接,这也是网络配置中最常用的命令;

(2) ipconfig: 用于查看当前计算机的 TCP/IP 配置;

(3) netstat:显示连接统计;

(4) tracert: 进行源主机与目的主机之间的路由连接分析;

(5) arp: 实现 IP 地址到物理地址的单向映射。

四、实验内容

1. 掌握各种主要命令的作用

2. 掌握各种网络命令的主要测试方法

五、实验步骤

1. 掌握各种主要命令的作用

1.1 Ping 命令

一般情况下,用户可以通过使用一系列 Ping 命令来查找问题出在什么地方,或 检验网络运行的情况时。典型的检测次序及对应的可能故障如下:

(1) ping 127.0.0.1:如果测试成功,表明网卡、TCP/IP 协议的安装、IP 地址、 子网掩码的设置正常。如果测试不成功,就表示 TCP/IP 的安装或运行存在某些 最基本的问题。

(2) ping 本机 IP: 如果测试不成功,则表示本地配置或安装存在问题,应当 对网络设备和通讯介质进行测试、检查并排除。

(3) ping 局域网内其它 IP: 如果测试成功,表明本地网络中的网卡和载体运行正确。但如果收到 0 个回送应答,那么表示子网掩码不正确或网卡配置错误或电缆系统有问题。

(4) ping 网关 IP: 这个命令如果应答正确,表示局域网中的网关或路由器正 在运行并能够做出应答。

(5) ping 远程 IP: 如果收到正确应答,表示成功的使用了缺省网关。对于拨号上网用户则表示能够成功的访问 Internet。

(6) ping localhost: localhost 是系统的网络保留名,它是127.0.0.1 的别 名,每台计算机都应该能够将该名字转换成该地址。如果没有做到这点,则表示 主机文件(/Windows/host)存在问题。

(7) Ping www.163.com (一个著名网站域名): 对此域名执行 Ping 命令, 计算机必须先将域名转换成 IP 地址,通常是通过 DNS 服务器。如果这里出现故障,则表示本机 DNS 服务器的 IP 地址配置不正确,或 DNS 服务器有故障。

如果上面所列出的所有 Ping 命令都能正常运行,那么计算机进行本地和远程通信基本上就没有问题了。但是,这些命令的成功并不表示你所有的网络配置都没

有问题,例如,某些子网掩码错误就可能无法用这些方法检测到。Ping命令的常用参数选项如下:

ping IP-t: 连续对 IP 地址执行 Ping 命令, 直到被用户以 Ctrl+ C 中断。

ping IP -1 2000: 指定 Ping 命令中的数据长度为 2000 字节,而不是缺省的 32 字节。

ping IP -n: 执行特定次数的 Ping 命令。

ping IP -f: 强行不让数据包分片。

ping IP -a: 将 IP 地址解析为主机名。

1.2 IP 配置程序 Ipconfig

发现和解决 TCP/IP 网络问题时,先检查出现问题的计算机上的 TCP/IP 配置。可以使用 ipconfig 命令获得主机 TCP/IP 配置信息,包括 IP 地址、子网掩码和默认网关。命令格式为 ipconfig /options,其中 options 选项如下:

/?: 显示帮助信息。

/all:显示全部配置信息。

/release:释放指定网络适配器的 IP 地址。

/renew: 刷新指定网络适配器的 IP 地址。

/flushdns: 清除 DNS 解析缓存。

/registerdns: 刷新所有 DHCP 租用和重新注册 DNS 名称。

/displaydns:显示DNS 解析缓存内容。

使用带/all 选项的 ipconfig 命令时,将给出所有接口的详细配置报告,包括 任何已配置的串行端口。使用 ipconfig /all 可以将命令输出重定向到某个文

 $13\ /\ 43$

件,并将输出粘贴到其他文档中,也可以用该输出确认网络上每台计算机的 TCP/IP 配置,或者进一步调查 TCP/IP 网络问题。如图1所示。

C: \>									
C:\>ipco	onfig /all								
Windows	IP Configuration								
	Host Name				4			:	ađ
	Primary Dns Suffix							=	
	Node Type							=	Unknown
	IP Routing Enabled.				-			=	Yes
	WINS Proxy Enabled.				-			1	No
Ethernet	t adapter 本地连接: Connection-specific	DN	s s	su:	ff :	i×		-	IC Dive ID100 10/100 Eact Ethowert O
danten	Description		•		-			- 7	TO FILS IFIDD ID/100 FAST ECHEFNEL H
uapcer.	Physical Address							=	00-50-8D-74-3C-EB
	Dhcp Enabled							•	No
	IP Address				-			Ξ	10.112.12.66
	Subnet Mask				-			=	255.255.255.0
	Default Gateway							=	10.112.12.254
	DNS Servers		-					=	202.100.192.68
									210.37.95.1

图 1

1.2.1 另一方法:点击右下角的电脑图像,如图2所示

印 无线网络连接 2 状态	? 🔀
常规 支持	
连接状态	
一 地址类型:	手动配置
ビ IP 地址:	202, 193, 93, 103
子网掩码:	255. 255. 255. 0
默认网关:	202. 193. 93. 254
详细信息 (D)	
Windows 没有检测到此连接的 您无法连接,请单击"修复"	的问题。如果 修复 (2)

图 2

1.2.2 点击详细信息,如图3所示

属性	数值
实际地址 P 地址 子网掩码 ty认网关 DNS 服务器 (INS 服务器	00-25-56-6B-9F-90 202. 193. 93. 103 255. 255. 255. 0 202. 193. 93. 254 202. 193. 80. 72 202. 193. 80. 33

图 3

1.2.3 记下地址信息: IP 地址、子网掩码、默认网关、MAC 地址等信息。

2. 掌握各种网络命令的主要测试方法

2.1 显示网络连接程序 netstat

netstat 命令的功能是显示网络连接、路由表和网络接口信息,可以让用户得知目前都有哪些网络连接正在运作,其命令格式为:

netstat [-a] [-e] [-n] [-s] [-p protocol] [-r] [interval]

参数说明如下:

(1) Netstat -s: -s 选项能够按照各个协议分别显示其统计数据。这样就可以 看到当前计算机在网络上存在哪些连接,以及数据包发送和接收的详细情况等等。 如果应用程序(如 Web 浏览器)运行速度比较慢,或者不能显示 Web 页之类的 数据,那么可以用本选项来查看一下所显示的信息。仔细查看统计数据的各行, 找到出错的关键字,进而确定问题所在。

(2) Netstat -e: -e 选项用于显示关于以太网的统计数据。它列出的项目包括 传送的数据报的总字节数、错误数、删除数、数据报的数量和广播的数量。这些 统计数据既有发送的数据报数量,也有接收的数据报数量。使用这个选项可以统 计一些基本的网络流量。

(3)Netstat -r: -r 选项可以显示关于路由表的信息,类似后面所讲使用 route print 命令时看到的信息。除了显示有效路由外,还显示当前有效的连接。

(4) Netstat -a: -a 选项显示一个所有的有效连接信息列表,包括已建立的连接(ESTABLISHED),也包括监听连接请求(LISTENING)的那些连接。

(5) Netstat -n: 显示所有已建立的有效连接,以数字格式显示地址和端口号。

(6) Netstat -p protocol: 显示由 protocol 指定的协议的连接。protocol 可以是 TCP

或 UDP。如果与-s 选项并用显示每个协议的统计, protocol 可是 TCP、UDP、ICMP 或 IP。

(7) Netstat interval: 重新显示所选的统计,在每次显示之间暂停 interval 秒。按 Ctr1+B 键停止,重新显示统计。如果省略该参数,netstat 将打印一次 当前的配置信息。我们可以利用 netstat 命令查看本机开放端口的方法来检查 自己是否被种了木马或其他黑客程序。进入到命令行下,使用 netstat 命令的 a 和 n 两个参数的组合,如图 4 所示。

C · > >nata	tat _an		
G. Vnets	itat -all		
Active C	Connections		
Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1110	0.0.0.0:0	LISTENING
TCP	0.0.0.0:9800	0.0.0.0:0	LISTENING
TCP	10.112.12.66:139	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1025	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1103	127.0.0.1:1110	CLOSE_WAIT
TCP	127.0.0.1:2210	127.0.0.1:9800	ESTABLISHED
TCP	127.0.0.1:9800	127.0.0.1:2210	ESTABLISHED
UDP	0.0.0.0:500	*:*	
UDP	0.0.0.0:1032	*:*	
UDP	0.0.0.0:1033	*:*	
UDP	0.0.0.0:1748	*:*	
HDP	0-0-0-0:2193	*:*	

图 4

16 / **4**3

其中, "Active Connections"是指当前本机的活动连接; "Proto"是指连接 使用的协议名称; "Local Address"是本地计算机的 IP 地址和连接正在使用 的端口号; "Foreign Address"是连接该端口的远程计算机的 IP 地址和端口 号; "State"则是表明 TCP 连接的状态,可以看到后面几行的监听端口是 UDP 协议的,所以没有 State 表示的状态。

2.2 路由分析诊断程序 tracert

tracert 命令可以用来跟踪一个报文从一台计算机到另一台计算机所走的路径。 入图 5 所示。

命令格式如下:

tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name 参数说明如下:

-d: 不进行主机名称的解析。

-h maximum_hops:最大的到达目标的跃点数。

-j host-list: 根据主机列表释放源路由。

-w timeout:设置每次回复所等待的毫秒数。

比如用户在上网时,想知道从自己的计算机如何走到网易主页,可在MS-DOS 方 式下输入命令 tracert www.163.com。 🖾 命令提示符

```
C:\>tracert www.163.com
Tracing route to www.163.com [202.108.36.172]
over a maximum of 30 hops:
                10 ms
      <10 ms
                         <10 ms
                                  210.41.232.65
  1
                <10 ms
                         <10 ms
      <10 ms
                                  210.41.232.97
  2
                <10 ms
                         <10 ms
  3
      <10 ms
                                  172.16.16.1
                <10 ms
                         <10 ms
  4
      <10 ms
                                  202.112.14.13
  5
                <10 ms
      <10 ms
                         <10 ms
                                  cd0.cernet.net [202.112.53.73]
  6
       20 ms
                20 ms
                          20 ms
                                  202.112.46.181
  7
                                  bjwh4.cernet.net [202.112.46.65]
       40 ms
                 40 ms
                          30 ms
  8
       30 ms
                 40 ms
                          40 ms
                                  202.112.61.162
  9
                  ×
                           *
                                  Request timed out.
 10
       40 ms
                 40 ms
                          30 ms
                                  219.158.11.113
                 40 ms
                          40 ms
                                  202.96.12.30
 11
       40 ms
 12
       30 ms
                 40 ms
                          40 ms
                                  RTR-BTO-A-F9-1-0.bta.net.cn [202.106.192.225]
13
                 40 ms
                          40 ms
                                  RTR-AHL-A-S2-0.bta.net.cn [202.106.192.170]
       30 ms
 14
       30 ms
                 41 ms
                          40 ms
                                  210.74.176.150
 15
       40 ms
                 30 ms
                          40 ms
                                  202.108.36.172
Trace complete.
C: \>
```

- 🗆 ×

图 5

最左边的数字称为"hops",是该路由经过的计算机数目和顺序。"10 ms"是 向经过的第一个计算机发送报文的往返时间,单位为ms。在时间信息之后,是 计算机的名称信息。tracert 最多会显示 30 段"hops",上面会同时指出每次 停留的响应时间,以及网站名称和沿路停留的 IP 地址。

2.3 ARP 地址解析协议

ARP 是 TCP/IP 协议族中的一个重要协议,用于把 IP 地址映射成对应网卡的物理地址。如图 6 所示

常用命令选项:

(1) arp -a: 用于查看高速缓存中的所有项目。

(2) arp -a IP: 如果有多个网卡,那么使用 arp -a 加上接口的 IP 地址,就可以只显示与

该接口相关的 ARP 缓存项目。

(3) arp -s IP 物理地址:向 ARP 高速缓存中人工输入一个静态项目。该项在 计算机引导

过程中将保持有效状态,或者在出现错误时,人工配置的物理地址将自动更新该项目。

(4) arp -d IP: 使用本命令能够人工删除一个静态项目。

<c>版权所有 1985-200</c>	1 Microsoft Corp.	
C:\Documents and Sett	ings\gengqiang646>arp -	-a
Interface: 10.112.12. Internet Address 10.112.12.254	66 0x10003 Physical Address 00-e0-fc-1c-00-fc	Туре dynamic
C:\Documents and Sett	ings\gengqiang646>arp -	-s 10.112.12.33 11-22-33-ee-ff-aa
G: Wocuments and Sett	ings/gengqiang646/arp -	-a
Interface: 10.112.12.	66 0×10003	
Internet Address	Physical Address	Туре
10.112.12.33	11-22-33-ee-ff-aa	static
10.112.12.254	00-e0-fc-1c-00-fc	dynamic
C:\Documents and Sett	ings\gengqiang646>_	
1		`

图6

五、实验报告要求:

按实验报告模板撰写实验报告

实验 四 IP 层协议分析

一、实验目的

理解 IP 层的作用以及 IP 地址的分类方法;理解子网的划分和子网掩码的作用; 掌握 IP 数据包的组成和网络层的基本功能。

二、实验环境

WIRESHARK

三、实验内容

1. 查看本机 IP 地址及子网掩码

2. 利用网络协议分析软件捕获并分析 IP 数据包

3. 利用 PING 命令发送 IP 数据包

4. ping 本地主机,分析 IP 协议

四、实验步骤

1. 查看本机 IP 地址及子网掩码

1.1 在运行中输入 cmd, 出现界面后输入 ipconfig /all。如图 1 所示

G:\Documents and Settings\Administrator>ipconfig /all Windows IP Configuration Host Name . . Primary Dns Suffix : Node Type Unknown IP Routing Enabled. No WINS Proxy Enabled. No Ethernet adapter 本地连接 5: Connection-specific DNS Suffix . : Description Attansic L2 Fast Ethernet 10/100 Base-T A dapter Physical Address. : 00-1B-FC-A6-AE-E2 DHCP Enabled. No Subnet Mask 255.255.255.0 Default Gateway : 172.16.1.1 DNS Servers : 172.16.1.248 202.96.64.68

图 1

1.2 观察运行结果,获得本机的 IP 地址及子网掩码,从上图中的显示结果中可 以看到, ipconfig /all 命令输出包括主机名称,节点类型等,以及网络接口上 的相关配置。从上图中可以看到网络接口配置信息如图 2 所示:

- MAC 地址: 00-1B-FC-A6-AE-E2
- DHCP: 为未启用
- IP地址: 172.16.1.239
- 子网掩码: 255.255.255.0
- 默认网关: 172.16.1.1
- DNS 服务器: 172.16.1.248
- 202. 96. 64. 68

图 2

思考1:分析本主机属于哪一类IP地址,网络号、子网号和主机号分别是什么?

2. 利用网络协议分析软件捕获并分析 IP 数据包

2.1 在本地主机中打开网络协议分析软件,在工具栏中点击"开始"

2.2 在捕获到数据包中,选择 IP 数据包(协议类型为 TCP\IPV4\ICMP)进行分析,如图 3 所示。

$21\ /\ 43$

# Frame 7339: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0	
Ethernet II, Src: 50:78:82:91:11:70 (50:78:82:91:11:70), Dst: Fujianst_15:47:39 (00:1a:a9:15:47:39)	
Destination: FujianSt_15:47:39 (00:1a:a9:15:47:39)	
Address: Fuijan5t 15:47:39 (00:1a:a9:15:47:39)	
0 = IG bit: Individual address (unicast)	
□ Source: 50:78:82:91:11:70 (50:78:82:91:11:70)	
Address: 50:78:82:91:11:70 (50:78:82:91:11:70)	
0	
Type: IP (0x0800)	
□ Internet Protocol version 4, Src: 10.200.70.197 (10.200.70.197), Dst: 202.121.66.191 (202.121.66.191)	
Version: 4	
Header Length: 20 bytes	
□ Differentiated Services Field: 0x00 (DSCP 0x00: Default: ECN: 0x00: Not-ECT (Not ECN-Capable Transport))	
0000 00 = Differentiated Services Codepoint: Default (0x00)	
Total Length: 1500	
Identification: 0x54b4 (21684)	
# Flags: 0x01 (More Fragments)	
Fragment offset: 0	
Time to live: 64	
Protocol: ICMP (1)	
Header checksum: Oxala7 [validation disabled]	
Source: 10.200.70.197 (10.200.70.197)	
Destination: 202.121.66.191 (202.121.66.191)	
[Source GeoTP: Unknown]	
[Source GeoIP: Unknown]	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] Reassemblad Tby4 in frame: 7379	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u> © Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] Reassembled ITv4 in frame: 7379 B Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPV4 in frame: 7379</u> ⊕ Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u> ⊕ Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u> ⊕ Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u> ⊞ Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u> ⊕ Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u> ⊕ Data (1480 bytes)	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] Reassembled IPv4 in frame: 7379	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] Reassembled IPv4 in frame: 7379	
[Source GeoIP: Unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	
[Source GeoIP: unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	* * *
[Source GeoIP: unknown] [Destination GeoIP: Unknown] <u>Reassembled IPv4 in frame: 7379</u>	() () 17.49 2018/11/12

图 3

2.3 分析捕获到的 IP 数据包,在本实验中,只分析数据的的 IP 包头部分。如图 4 所示:

- **总长度**: 60,表示总长度为 60 字节。
- 标识: 0X1F06,此数据包没有进行分片。
- **标志:** 2, 二进制为 010, 表示此数据包不可分片。
- 分段偏移量: 0X0000,此数据包没有进行分片。
- 生存时间: 127,每经过一个路由器,生存时间减1,当生存时间减小为0时, 数据包被丢弃而不被转发。
- 源 IP 地址: 此字段显示了数据包的源地址。
- 目的 IP 地址: 此字段显示了数据包的目的地址。
- 其他:此包头中,没有选项字段,没有填充字段。IP 报头之后的部分为 IP 包中的数据部分。

图 4

思考 2. 所捕获的 IP 数据报,可以发现头部开始并不是版本号,那么头部开始时 什么,实际上我们捕获的是什么?

3. 利用 PING 命令发送 IP 数据包

- 3.1 Ping www.baidu.com -1 2000 (过大的好像会封掉)
- 3.2 分析捕获到的数据包的 IPv4 报头部分,如图 5 所示:
 - 版本信息: IPv4 报文的版本信息为 4。
 - 头部长度: IPv4 报头不含选项和填充字段长度为 20 字节, 是 32 比特的 5 倍。
 - 总长度: 总长度包含 IP 报头长度和 IP 包中的数据长度,协议分析软件将 IP 发送时 自动将上层协议选择为 TCP,因此数据内容为 TCP 报头共 20 字节以及 20 字节数据, 因此接收到的 IP 报文长度为 60 字节。
 - 标识、标识、分段偏移量: 均与分片有关。
 - 生存时间:由于数据包从源端到目的端没有经过任何路由器的转发,TTL 值不变为 128。
 - 校验和:由于发包过程中,标识位和分段偏移修改,因此校验和也和发送的数据稍 有不同。
 - 源目标地址:源目标地址在 IP 包的发送过程中不做修改。

图 5

用同样的方法,在主机 A 中编辑 IP 包,将目的 MAC 地址和目的 IP 地址修改为另 外一组主机的地址如 pc3,发送数据包。注意,封装 IP 包发往不同网段的目的 主机时,目标 MAC 地址选择网关的 MAC 地址,地址本中找不到不同网段的 IP 地 址时,手工输入目的 IP 地址。

思考 3. 所捕获到的报文,其最大长度为多少,原因是什么?

4. ping 本地主机,分析 IP 协议

4.1 在地址本中选择与本主机在同一子网中另一主机的 IP 地址(假设为:

172.16.1.251)。在本机命令行界面下运行: ping 172.16.1.251。

4.2 在 ping 的目的地址的主机上用协议分析器一端捕获数据,记录源、目的物理地址及源、目的 IP 地址。

4.3 按照地址本中的记录,分析捕获数据的 MAC 地址与 IP 地址的对应关系。

23 / 43

4.4 在 ping 目的主机上通过协议分析器, 查看"交互序列图", 了解 PING 程序的会话过程, 如图 6 所示。



图 6

4.5 选择与本主机属于不同子网另一主机的 IP 地址(假设为: 172.16.2.1),在 命令行方式下运行: ping 172.16.2.1。

4.6 协议分析器端捕获数据,记录源、目的物理地址和源、目的 IP 地址。

思考 4. 分析捕获数据的 MAC 地址与 IP 地址是否具有对应关系,比较上面两个实验的结果,分析二者有何不同?

五、实验报告要求

按实验报告模板撰写实验报告

实验 五 TCP 协议分析

一、实验目的

预习 TCP 报文段首部及端口知识;理解 TCP 协议三次握手,四次握手的含义。

二、实验环境

WIRESHARK

三、实验内容

1. 捕获一个从你电脑到远程服务器的 TCP 数据

2. 掌握 TCP 报文段首部的格式、长度以及各字段的功能

3. 学会通过分析 TCP 报文段首部,理解 TCP 握手机制

四、实验步骤

1. 捕获一个从你电脑到远程服务器的 TCP 数据

1.1 选择一个网站上网冲浪,用"TCP"为过滤条件,捕获建立连接和断开连接的数据。如图1所示:





25 / 43

1.2.1 TCP 连接通过称为三次握手的三条报文来建立的。观察以上数据,其中分 组 10 到 12 显示的就是三次握手。第一条报文没有数据的 TCP 报文段(分组 10), 并将首部 SYN 位设置为 1。因此,第一条报文常被称为 SYN 分组。这个报文段里 的序号可以设置成任何值,表示后续报文设定的起始编号。连接不能自动从 1 开始计数,选择一个随机数开始计数可避免将以前连接的分组错误地解释为当前 连接的分组。观察分组 10,Wireshark 显示的序号是 0。选择分组首部的序号字 段,原始框中显示 "9b 8e d1 f5"。Wireshark 显示的是逻辑序号,真正的初 始序号不是 0。如图 2 所示:

🚄 *本地连接 8 [Wireshark	1.10.2 (SVN Rev 51934 from /trunk-1	10)]	And and a starting of	REEL Mount had	- 0 - X -
<u>File Edit View Go Cap</u>	pture <u>Analyze Statistics</u> Telephony	Tools Internals Help			
• • <u> </u>	3 🛅 🗶 😂 🔍 🗢 🌳 🤪 🗿		. 🗹 👹 🖻	1 😼 🎉 🛄	
Filter: tcp		 Expression Clear 	Apply Save		
No. Time	Source	Destination	Protocol I	ength Info	
1 0.000000000	192.168.1.100	220.181.111.64	TCP	54 62943 > http [RST, ACK] Seq=1 Ack=1 Win=0 Len=0	
9 11 2637/1000	192 168 1 100	220 181 111 64	TCP	54.62947 > bttp [ACK] Seg=1 Ack=1 Win=16481 Len=0	
10 11, 589186000	192.168.1.100	10.10.39.250	TCP	66 62953 > ftp [SYN] Seg=0 Win=8192 Len=0 MSS=1460 WS=4	SACK PERM=
11 11.590148000	10.10.39.250	192.168.1.100	TCP	66 ftp > 62953 [SYN, ACK] Seg=0 Ack=1 Win=5840 Len=0 MSS	=1460 SACK
12 11.590192000	192.168.1.100	10.10.39.250	TCP	54 62953 > ftp [ACK] Seq=1 Ack=1 Win=65700 Len=0	
13 11.592272000	10.10.39.250	192.168.1.100	FTP	91 Response: 220 Welcome to Usst Soft FTP Server	
14 11.593683000	192.168.1.100	10.10.39.250	FTP	69 Request: USER download	
15 11.594474000	10.10.39.250	192.168.1.100	TCP	60 ftp > 62953 [ACK] Seq=38 Ack=16 Win=5840 Len=0	
16 11.594606000	10.10.39.250	192.168.1.100	FTP	88 Response: 331 Please specify the password.	
17 11.595930000	192.168.1.100	10.10.39.250	FTP	65 Request: PASS usst	
18 11.602858000	10.10.39.250	192.168.1.100	FIP	// Response: 230 Login successful.	- · ·
 Frags: 0x02 (C) Fragment offse Time to live: Protocol: TCP Header checksu Source: 192.16 Destination: 1 [Source Geol?] [Destination construction of Source opent: 6 Destination index: Sequence numbe Header length: Flags: 0x002 	Non C Fragment) 128 (6) 128 8.1.100 (192.168.1.100) 8.1.100 (192.168.1.100) 10.10.39.250 (10.10.39.25) Unknown] acofP: Unknown] ntrol Protocol, Src Port: 2953 (62953) ort: ftp (21) 2] r: 0 (relative sequen: 32 bytes (SWN)	ould be OxccfO (may t D) 62953 (62953), Dst I ce number)	De caused Port: ftp	by "IP checksum offload"?)] (21), Seq: 0, Len: 0	
000	= Reserved: Not set = Nonce: Not set = Congestion Window R = ECN-Echo: Not set = Pursh: Not set = Push: Not set 1. = Syn: Set .0 = Fin: Not set 100: 8120 100: 8120 40 00 80 06 00 00 co 88 00 00 00 00 00 00 00 00 00 00 00 00 00	educed (CWR): Not set	c		
0040 04 02			Defects		1 •

图 2

1.2.2 SYN 分组通常是从客户端发送到服务器。这个报文段请求建立连接。一旦 成功建立了连接,服务器进程必须已经在监听 SYN 分组所指示的 IP 地址和端口 号。如果没有建立连接,SYN 分组将不会应答。如果第一个分组丢失,客户端通 常会发送若干 SYN 分组,否则客户端将会停止并报告一个错误给应用程序。如果 服务器进程正在监听并接收到来的连接请求,它将以一个报文段进行相应,这个 报文段的 SYN 位和 ACK 位都置为 1。通常称这个报文段为 SYNACK 分组。SYNACK 分组在确认收到 SYN 分组的同时发出一个初始的数据流序号给客户端。如图 3 所示

🚄 *本地连接 8 [Wireshar	k 1.10.2 (SVN Rev 519)	34 from /trunk-1.10)]	-	STREET House Bad		
<u>Elle Edit View Go C</u>	apture <u>A</u> nalyze <u>S</u> tatis	stics Telephony Tools Internals Help				
0 🖲 🛋 🔳 🙇 [🖹 🚨 🗶 😂 I 🔍	🍬 🏟 🤿 📅 👱 🗐 📑 🛛 🗨 Q G	. 🖭 🍯	🛚 🗹 🍕 % 🙀		
Filter: tcp		Expression Clear	Apply	Save		
No. Time	Source	Destination	Protoc	ol Length Info		
1 0.000000000	192.168.1.100	220.181.111.64	TCP	54 62943 > http [RST, ACK]] Seq=1 Ack=1 Win=0 Len=0	
8 11.263670000	220.181.111.6	192.168.1.100	TCP	60 http > 62947 [FIN, ACK]] Seq=1 Ack=1 Win=60 Len=0	
9 11.263741000	192.168.1.100	220.181.111.64	TCP	54 62947 > http [ACK] Seq	=1 Ack=2 Win=16481 Len=0	
10 11.589186000	192.168.1.100	10.10.39.250	TCP	66 62953 > ftp [SYN] Seq=	0 Win=8192 Len=0 MSS=1460	WS=4 SACK_PERM=
11 11.590148000	10.10.39.250	192.168.1.100	TCP	66 ftp > 62953 [SYN, ACK]	Seq=0 Ack=1 Win=5840 Len=	0 MSS=1460 SACK
12 11.590192000	192.168.1.100	10.10.39.250	TCP	54 62953 > ftp [ACK] Seq=	1 Ack=1 Win=65/00 Len=0	
13 11.5922/2000	10.10.39.250	192.168.1.100	FTP	91 Response: 220 Welcome	to Usst Soft FIP Server	
15 11 593083000	10 10 20 250	102 168 1 100	TCD	60 ftm + 62052 [ACK] Fam	29 Ack 16 Win 5940 Lan 0	
15 11. 5944/4000	10.10.39.230	192.108.1.100	ICP CTD	00 TLD > 02955 [ACK] Seq=.	so ACK=10 WIN=3640 Len=0	
17 11 505020000	102 169 1 100	192.108.1.100	FTP	65 Request: BASS west	pectry the password.	
18 11 602858000	10 10 39 250	192 168 1 100	ETP	77 Response: 230 Login su	cessful	
{	10.10.35.250	152.100.1.100		Thesponse. 250 Login Su	cccssrur.	
Fragment offs Time to live: Protocol: TCP Header checks Source: 10.10 Destination: [Source GeoIP [Destination Bransmission CC Source onum Pistream index Source onum Acknowledgmen Header length 00000000000	et: 0 62 (6) un: 0x48b4 [cor .39.250 (10.10. 192.168.1.100 / 192.168.1.100 / 192.168.1.100 / 662PF: Unknown] ort: 62953 (622: .2] 192.0 (relation for the second ftp (21) ort: 62953 (622: .2] 192.0 (relation for the second (SYN, ACK) = Reserved: = Urgent: N = Acknowle = Push: Not 0 = Reset: No. .1. = Syn: Set	rrect] 39.250) (192.168.1.100) , Src Port: ftp (21), Dst Port: 253) (relative ack number) (relative ack number) Not set it set Not set Not set Not set Set set is set it set is set it set	62953 t	(62953), Seq: 0, Ack: 1, Len: t	0	
Window size v	0 = Fin: Not alue: 5840	set				
[Calculated w	rindow size: 584	407				
0010 00 34 00 00 0020 01 64 00 15 0030 16 d0 64 44 0040 03 00 00 00	0 40 00 3e 06 5 f5 e9 <mark>f6 74</mark> 4 00 00 02 04	48 b4 0a 0a 27 fa c0 a8 .4@ a5 ea 9b 8e d1 f6 80 12 .d 05 b4 01 01 04 02 01 03 dD.	.>. H. t			-
🔴 💅 Sequence number I	(tcp.seq), 4 bytes	Packets: 360 · Displayed: 185 (51.4 Profile	Default			

图 3

1.2.3 分组 11 的确认号字段在 Wireshark 的协议框中显示 1,并且在原始框中 的值是"9b 8e d1 f6"(比"9b 8e d1 f5"多1)。这解释了 TCP 的确认模式。 TCP 接收端确认第 X 个字节已经收到,并通过设置确认号为 X+1 来表明期望收到 下一个字节号。分组 11 的序号字段在 Wireshark 的协议显示为 0,但在原始框 中的实际值却是"f6 74 a5 ea"。这表明 TCP 连接的双方会选择数据流中字节 的起始编号。所有初始序号逻辑上都视同为序号 0。

1.2.4 最后,客户端发送带有标志 ACK 的 TCP 报文段,而不是带 SYN 的报文段来 完成三次握手的过程。这个报文段将确认服务器发送的 SYNACK 分组,并检查 TC P 连接的两端是否正确打开合运行。

1.3 关闭连接

当两端交换带有 FIN 标志的 TCP 报文段并且每一端都确认另一端发送的 FIN 包时, TCP 连接将会关闭。FIN 位字面上的意思是连接一方再也没有更多新的数据发送。 然而,那些重传的数据会被传送,直到接收端确认所有的信息。通过分组 43,44 和 54,55 我们可以看到 TCP 连接被关闭。如图 4 所示。



图 4

2. 掌握 TCP 报文段首部的格式、长度以及各字段的功能

2.1 当一个 TCP 发送端传输一个报文段的同时也设置了一个重传计时器。当确认 到达时,这个计时器就自动取消。如果在数据的确认信息到达之前这个计时器超 时,那么数据就会重传。

重传计时器能够自动灵活设置。最初 TCP 是基于初始的 SYN 和 SYN ACK 之间的时间来设置重传计时器的。它基于这个值多次设置重传计时器来避免不必要的重传。 在整个 TCP 连接中,TCP 都会注意每个报文段的发送和接到相应的确认所经历的时间。TCP 在重传数据之前不会总是等待一个重传计算器超时。TCP 也会把一系列重复确认的分组当作是数据丢失的征兆。

 $28\ /\ 43$

在上面的每次跟踪中,我们能观察建立连接的三次握手。在 SYN 分组中,发送端在 TCP 的首部选项中通过包括 SACK permitted 选项来希望使用 TCP SACK。在 S YN ACK 包中接收端表示愿意使用 SACK。这样双方都同意接收选择性确认信息。S ACK 选项如图 5 所示。

🚄 tcp.pcapng [Wireshark	1.10.2 (SVN Rev 51934 from /trunk-1	10)]	Section 2	Married Transfer		
<u>File Edit View Go Ca</u>	pture <u>A</u> nalyze <u>S</u> tatistics Telephony	<u>T</u> ools Internals <u>H</u> elp				
• • 🖌 🔳 🔬 E	B 🖹 🗶 🤁 I 🔍 🔶 🌳 🖓 🎙		. 🖻 🏭 🖻	🕵 🖗 😫		
Filter: tcp		Expression Clear	Apply Save			
No. Time	Source	Destination	Protocol Le	ength Info		
85 42.116370000	192.168.1.100	220.181.111.64	тср	54 62947 > http [RST	F, ACK] Seq=1 Ack=2 Win=0 Len=0	
86 42.191246000	192.168.1.100	220.181.112.122	HTTP	950 GET /i/msg/listen	1?user=57fe6c616e6d5f636e310a0b&c=6_13829	483968
87 42.191332000	192.168.1.100	220.181.111.64	TCP	66 62955 > http [SYN	<pre>N] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SAC</pre>	K_PERM
88 42.191333000	220 181 111 64	102 168 1 100	TCP	66 http > 62955 [SVA	N Sed=0 Win=8192 Len=0 MSS=1460 WS=4 SAC ACK1 Sed=0 Ack=1 Win=8192 Len=0 MSS=14	OR WE
90 42 227 238000	192 168 1 100	220 181 111 64	TCP	54.62955 > http [ACK	(] Seg=1 Ack=1 Win=66176 Len=0	00 13-
91 42.227531000	220.181.111.64	192.168.1.100	TCP	66 http > 62956 [SYN	, ACK] Seg=0 Ack=1 Win=8192 Len=0 MSS=14	08 WS=
92 42.227581000	192.168.1.100	220.181.111.64	ТСР	54 62956 > http [ACK	<pre>Seq=1 Ack=1 Win=66176 Len=0</pre>	
93 42.228080000	220.181.112.122	192.168.1.100	TCP	60 http > 62933 [ACK	<pre>[] Seq=158 Ack=897 Win=129 Len=0</pre>	-
94 42.237791000	192.168.1.100	220.181.111.64	HTTP	955 GET /tb/img/pv.gi	if?fr=tb0_chat&st_mod=lcf&st_type=listenB	ack&st
95 42.2/4593000	220.181.111.64	192.168.1.100	ICP	60 nttp > 62956 [ACK	(j Seq=1 ACK=902 Win=/680 Len=0	
96 42.274594000	192 168 1 100	192.108.1.100	HITP	54 TCB Retransmissi	(GIF89a)	win-16
98 42, 415603000	192.168.1.100	220.181.111.64	TCP	54 [TCP Retransmissi	ion] 62946 > http [FIN, ACK] Seg=1 ACK=1	Win=16
99 42.474610000	192.168.1.100	220.181.111.64	TCP	54 62956 > http [ACK	(] Seg=902 Ack=253 Win=65924 Len=0	
100 43.010637000	192.168.1.100	115,239,210,151	TCP	54 [TCP Retransmissi	ion] 62948 > http [FIN, ACK] Seq=1 Ack=1	Win=16
101 43.015619000	192.168.1.100	220.181.111.64	TCP	54 [TCP Retransmissi	ion] 62946 > http [FIN, ACK] Seq=1 Ack=1	Win=16
107 44.210698000	192.168.1.100	115.239.210.151	тср	54 [TCP Retransmissi	on] 62948 > http [FIN, ACK] Seq=1 Ack=1	Win=16
108 44.215674000	192.168.1.100	220.181.111.64	TCP	54 [TCP Retransmissi	ion] 62946 > http [FIN, ACK] Seq=1 Ack=1	Win=16
109 46.610/8/000	192.168.1.100	115.239.210.151	TCP	54 LICP Retransmissi	ion] 62948 > http [FIN, ACK] Seq=1 ACK=1	Win=16
112 48 624509000	192.168.1.100	180 149 122 184	TCP	54 62020 Shttp FETA	ION] 02940 > NLLP [FIN, ACK] Seq=1 ACK=1	win=10
114 48 624644000	192.168.1.100	180 149 132 184	ТСР	54 62938 > http [FIN	ACK] Seg=1 Ack=1 Win=16504 Len=0	
115 48.624963000	192.168.1.100	180.149.132.184	TCP	66 62957 > http [SYN	Seg=0 Win=8192 Len=0 MSS=1460 WS=4 SAC	K_PERM
116 48.624969000	192.168.1.100	180.149.132.184	TCP	66 62958 > http [SYN	Sea=0 Win=8192 Len=0 MSS=1460 WS=4 SAC	K PERM
<			III			•
0	= Acknowledgment: Not	set				
0.	= Push: Not set					
	1 - Syn: Set					
	0 = Ein: Not set					_
Window size va	alue: 8192					
[Calculated wi	indow size: 8192]					
Checksum: 0x0e	e29 [validation disabled]					
🛾 🗎 Options: (12 l	bytes), Maximum segment s	ize, No-Operation (N	OP), Window	scale, No-Operation	(NOP), No-Operation (NOP), SACK permitte	d
Maximum segn	nent size: 1460 bytes					
Window scale	e: 2 (multiply by 4)					
No-Operation	(NOP)					
No-Operation	NOP)					
■ TCP SACK Per	rmitted Option: True					
0010 00 34 39 e7	40 00 80 06 00 00 c0 a8	01 64 dc b5 .49.0	d			
0020 6f 40 f5 eb	00 50 al 51 ad 1a 00 00	00 00 80 02 0@	P.Q			
0030 20 00 0e 29	00 00 <mark>02 04 05 b4 01 03</mark>	03 02 01 01).				
0070 07702						
🔵 💅 TCP Options (tcp.op	otions), 12 bytes Packets: 360 ·	Displayed: 18 Profile: Defaul	t			

图 5

3. 学会通过分析 TCP 报文段首部,理解 TCP 握手机制

3.1 用显示过滤器 tcp. analysis. retransmission 搜索重传。如图 6 所示。



图 6

3.2 通过观察分组的序号、确认号的变化,研究重传行为。

五、实验报告要求

按实验报告模板撰写实验报告

实验六: HTTP 和 DNS 分析

一、实验目的

分析 HTTP 协议,分析 DNS 协议。

二、实验环境

WIRESHARK

三、实验内容

1.HTTP GET/response 交互

2. HTTP 条件 GET/response 交互

3. 获取长文件

4. 嵌有对象的 HTML 文档

5.HTTP 认证

6. 跟踪 DNS

四、实验步骤

1.HTTP GET/response 交互

1.1 启动 Web browser。

1.2 启动 Wireshark 分组嗅探器。在窗口的显示过滤说明处输入"http",分组 列表子窗口中将只显示所俘获到的 HTTP 报文。

1.3 一分钟以后,开始Wireshark分组俘获。

31 / 43

1.4 在打开的 Web browser 窗口中输入一下地址(浏览器中将显示一个只有一行 文字的非常简单的 HTML 文件):

http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file1.html

1.5 停止分组俘获,如图1所示。



图 1

2. HTTP 条件 GET/response 交互

2.1 启动浏览器,清空浏览器的缓存(在浏览器中,选择"工具"菜单中的 "Internet 选项"命令,在出现的对话框中,选择"删除文件")。

2.2 启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。

2.3 在浏览器的地址栏中输入以下 URL:

http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file2.html

你的浏览器中将显示一个具有五行的非常简单的 HTML 文件。

2.4 在你的浏览器中重新输入相同的 URL 或单击浏览器中的"刷新"按钮。

2.5 停止 Wireshark 分组俘获,在显示过滤筛选说明处输入"http",分组列表 子窗口中将只显示所俘获到的 HTTP 报文。

3. 获取长文件

3.1 启动浏览器,将浏览器的缓存清空。

3.2 启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。

3.3 在浏览器的地址栏中输入以下 URL:

http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file3.html

3.4 停止 Wireshark 分组俘获,在显示过滤筛选说明处输入"http",分组列表 子窗口中将只显示所俘获到的 HTTP 报文。

4. 嵌有对象的 HTML 文档

4.1 启动浏览器,将浏览器的缓存清空。

4.2 启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。

4.3 在浏览器的地址栏中输入以下 URL:

http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file4.html

4.4 停止 Wireshark 分组俘获,在显示过滤筛选说明处输入"http",分组列表 子窗口中将只显示所俘获到的 HTTP 报文。

5. HTTP 认证

5.1 启动浏览器,将浏览器的缓存清空。

5.2 启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。

5.3 在浏览器的地址栏中输入以下 URL:

33 / 43

http://gaia.cs.umass.edu/ethereal-labs/protected_pages/HTTP-etherealfile5.html

浏览器将显示一个 HTTP 文件,输入所需要的用户名和密码(用户名: eth-students,密码:networks)。

5.4 停止 Wireshark 分组俘获,在显示过滤筛选说明处输入"http",分组列表 子窗口中将只显示所俘获到的 HTTP 报文。

6. 跟踪 DNS

6.1 利用 ipconfig 命令清空你的主机上的 DNS 缓存。

6.2 启动浏览器,将浏览器的缓存清空。

6.3 启动 Wireshark 分组俘获器,在显示过滤筛选说明处输入"ip.addr==your _IP_address"(如: ip.addr==10.17.7.23)。

6.4 开始 Wireshark 分组俘获。

6.5 在浏览器的地址栏中输入: http://www.ietf.org

6.6 停止分组俘获。

6.7 开始 Wireshark 分组俘获。

6.8 在 www.mit.edu 上进行 nslookup (即执行命令: nslookup www.mit.edu)。

6.9 停止分组俘获。

6.10 重复上面的实验,只是将命令替换为: nslookup www.aiit.or.kr bitsy. mit.edu

五、实验报告要求

按实验报告模板撰写实验报告

实验 七 利用 Ethereal 分析 ARP 协议

一、实验目的

利用 Ethereal 捕获发生在 ping 过程中的 ARP 报文,加强对 ARP 协议的理解,掌握 ARP 报文格式,掌握 ARP 请求报文和应答报文的区别。

二、实验环境

Ethereal

三、实验内容

1. 利用 Ethereal 捕获分组

2. 在捕获分组中分析 ARP 协议

四、实验步骤

1. 利用 Ethereal 捕获分组

1.1 桌面双击 Ethereal, 启动 Ethereal, 如图 1 所示:

eliziase arabizia	沪 新建文件夹							
I Redæm	77 100 Ma							
75 15								
	All (G). txt							
	19 (4). Get							
578472_S	助建 文本文 档 txt							
jzjwl_do	e.							
Tracest	Ethereal							
课件								
桌面下载								
🤧 A-Ze	7 🕑 🥴	🐼 🤲 🦢 G:\Documents and	🎒 ethereal的使用简	(4) 演示文稿)	A 20 - 20 . pp t	😮 (Untitled) = 8th	😻 计触机科学与工程	

1.2 对 Capture Options 各个选项进行设置,如图2所示:

🙆 Ethereal: Captur	e Options					
Capture		N				
Interface: Broadcom	440x 10/100 Integrated Controller	: \Device\NPF_{6FE4BC25-9AFA-4				
IP address: 192.168.1.1	89					
Link-layer header type:	Ethernet 🗸 Buffer size: 1	🗘 megabyte(s)				
🔽 Capture packets in pr	romiscuous mode					
🔲 Limit each packet to	68 🗍 🗘 bytes					
Capture Filter: not top	port 3389					
Capture File(s)		Display Options				
File:	Browse	Update list of packets in real time				
🔲 Use <u>m</u> ultiple files						
Next file every 1	🗘 megabyte(s) 👻	Automatic scrolling in live capture				
Next file every 1	🗘 minute(s) 🗸	🔲 Hide capture info dialog				
Ring buffer with 2	🗘 files					
🔲 Stop capture after 1	🗘 file(s)	Name Resolution				
Stop Capture		Enable MAC name resolution				
🔲 after 🛛 1	🗘 packet(s)	Enable network name resolution				
🔲 after 🛛 1	🗘 megabyte(s) 🖂					
🔲 after 🛛 1	🗘 minute(s) 🗸	Enable transport name resolution				
Help	点击Start按钮开始捕获	Start Qancel				

图 2

1.3 点击 Start 按钮开始捕获分组,出现 Capture from… 对话框,如图 3 所示

Total	260	% of total	
SCTP	0		0.0%
тср	260		100.0%
UDP	0		0.0%
ICMP	0		0.0%
ARP	0		0.0%
OSPF	0		0.0%
GRE	0		0.0%
NetBIOS	0		0.0%
IPX	0		0.0%
VINES	0		0.0%
Other	0		0.0%
unning	00:00:15		



1.4 点击 Capture from ··· 对话框中 Stop 按钮结束捕获,如图 4 所示。

🚱 Ethereal	Capture	from Br	
Captured Pack	ets		
Total	260	% of total	
SCTP	0		0.0%
TCP	260		100.0%
UDP	0		0.0%
ICMP	0		0.0%
ARP	0		0.0%
OSPF	0		0.0%
GRE	0		0.0%
NetBIOS	0		0.0%
IPX	0		0.0%
VINES	0		0.0%
Other	0		0.0%
Running	00:00:15	op	

图 4

1.5 得到捕获记录,如图5所示。

C cs3	ihttp	- B	ther	eal																														
Eile	Edit	⊻iew	Ωo	⊊apt	ure	Analy	ze S	tatisti	ics	Help																								
		91			6) 🖪	1 >	c (÷)	8	9	4		ŝ	₮	2	: [[¥	Ð	Θ	11	ŦŦ] 6		V	5	Ж	(Ø				
Eilter:												•	E×p	ressio	on Ç	lear	Apply	(
No	Tin	ne		So	urce				De	stinati	on		5	rotoco	il Info	6																		
	1 0.	0000	00	De	11_1	7:34	:e2		Br	oadca	ast	- 2	A	RP	who 100	ha	s 19	2.16	8.1	.1?	Tell	192	168.	1.18	9									
	3 0.	0004	93 04	19	2.16	8.1.	189	D	dn	s.sjt	tu.ed	ez u.cn	D	NS NS	Sta	nda	ind q	uery	A	www.c	5.51	tu.e	du. ci	J N										
	4 0.	0046	19	dn	5.Sj	tu.e	du.c	n	19	2.168	3.1.1	89	D	NS	Sta	nda	ind q	uery	re	spons	e A	202.1	120.1	1.155										
	5 0.	0121	39	19.	2.16	8.1.	189	u cr	ww 10	W.CS.	.sjtu	.eau.	cn I	CP	L16	1 >	116	p [5	YNJ	Seq=	U Le	n=U r	955=_ ~k_1	460 win-	WS=0	1 15	000-0	I SER	=0	60 1	ws=0	T51/-0	TSED	-0
	7 0.	0150	31	19	2.16	8.1.	189	u.cr	WW	W. CS.	situ	.edu.	cn t	CP	116	1 >	htt		CK1	Sed=	1 AC	k=1 \	vin=8	3192	Len	=0 T	SV=9	94225	TS	ER=	0	150-0	IDER	.=0
	8 0.	0150	84	193	2.16	8.1.	189		WW	w.cs.	.sjtu	.edu.	cn H	TTP	GET	1	HTTP,	/1.1																
	9 0.	1281	26	WW	N.CS	.sjt	u.ed	u.cr	n 19	2.168	3.1.1	89	Т	CP	LTC	P S	egme	nt c	fa	reas	semb	led (PDU]	2										
	11 0	1299	18	10	N.CS	.sjt 8 1	u.ea 189	u.cr	1 19. Mont	2.168	5.1.1 situ	adu adu	cn T	CP	116	P S	egme	nt c n [A	CK1	reas	semb 407	ied i Ack-1	1671	win-	81.01	210	n=0	TSV-	94.2	26 -	TSEP-	-34845	61	
	LZ O.	1340	69	WW	N.CS	.sit	u.ed	u.cr	1 19	2.168	3.1.1	89	Т	CP	[TC	PS	eqme	nt c	f a	reas	semb	led i	PDU]		01.01			1.50-	546	~~	- Diarte	-94049		
1	L3 O.	1386	40	WW	N.CS	.sjt	u.ed	u.cr	n 19	2.168	3.1.1	89	т	CP	[TC	P s	egmei	nt_c	f a	reas	semb	led I	PDU]											
	L4 O.	1386	74	19	2.16	8.1.	189		WW 10	W. CS.	.sjtu	.edu.	cn T	CP	116	1 >	- http	p LA	CK1	Seq=	407	ACK=4	4567	w1n=	8192	2 Le	n=0	TSV=	94 Z	26 '	TSER:	=34845	61	
	6 0.	1497	25 49	19	N. CS	8.1.	u.eu 189	u.cr	1 19.	2.100 W.CS.	situ	.edu.	cn T	CP	116	2 3	htt	n [s	YNI	Sed=	OLP	n=0 I	-D0] 455=1	460	WS=0) TS	V=0	TSER	=0					
	L7 O.	1615	47	WW	N. CS	.sjt	u.ed	u.cr	1 19	2.168	3.1.1	89	Т	CP	[TC	PS	eqme	nt c	fa	reas	semb	led I	PDU]											
1	L8 0.	1615	89	193	2.16	8.1.	189		WW	w.cs.	sjtu	.edu.	сп т	CP	116	1 >	- http	p [A	CK]	Seq=	407	Ack=	7463	win=	8192	2 Le	n=0	TSV=	942	26 -	TSER-	34845	61	
	19 0.	1616	08 46	1.01	N. CS	.sjt	u.ed	u.cr	1 19.	2.168	3.1.1	89 odu	CD T	CP	116	P S	egme	nt c	T a	reas	semb	led I	PDU]	win-	674		-0	TEV-	047	76 -	TEED-	24945	61	
-	21 0.	1658	17	WWA	N. CS	.sit	u.ed	u. cr	1 1 9	2.168	3.1.1	.euu. 89	T	CP	LTC LTC	PS	eame	nt c	if a	reas	semb	led	PDII	with	0744	+ LC		150-	942	20	I SER-	-34645	OT	
	22 0	1658	54	54.545	u re	sit	n ad	u cr	1 10	7 168	211	80	т	0	htt	n	. กำล	2 Es	VN.	ACK]	San	-0 0.	-1-1	win-	1753	20.1	on-() MSS	-14	60 1	ws=0	TSV-0	TSED	-0
🔳 Fra	ume 9	(28	8 by	tes (on w	ire,	288	byt	tes	capti	ured)																							
⊞ Eth	erne	t II	, sr	C: D	-Lin	k_2d	:00:	cb ((00:	0f:30	d:2d:	00:cb), [ost:	Del1_	17:	:34:e	2 (00:1	5:05	:17:3	4:e2	2											
	erne	t Pr	otoc n Co	01, 1	brc:	www	.cs.	sjtι σπα	J. ed	u.cn	(202	.120.	1.1: Dct	Dont	DST :	194	2.168 (1161	·	189	(192.	. 168.	1.18	9)											
± 11 c	unsin i:	5510		nu o	1 191	oruc	υι,	SIC	PUI	c. m	cup (80),	DSC	PULC	· · · · · · · · ·	JT (TTOT	., .	sey.	1, /	ACK.	407,	Len	. 22.	-									
0000	00 1	L5 C5	5 17	34 e	2 00	0 OF	3d	2d	00 0	b 08	3 00 4	15 00		4	=		E.		-	-	-					_		T	-	-				
0010	01 1	12 90 ad 00) 0d	40 0	10 71	06 C	e0	51	ca i	/8 01	96	20 a8		@	· {													T						
0030	42 0	da 6	fc	00 0		i 01	08	0a	00	35 2E	91 0		B				+																	
0040	70 1	1 48	3 54	54 5	0 21	F 31	2e	31	20 3	32 30	30 3	20 4f	p	. HTT	P/1 .	1 2	00 0																	
Frame	(frame)), 288	bytes			Caller Cra.76							10		And the second second second				F	261 0	D: 261	M: 0												

图 5

2. 在捕获分组中分析 ARP 协议

2.1 利用 Summary 观察跟踪记录的统计概要,包括通信的总字节数,通信的频率, 分组的大小等统计数据。如图 6 所示。

C Ethereal: Su	mary							
File								
Name:	DUTEMPL		200					
Length:	194123 h	ntes						
Format:	libpcap (to	pdump, Ether	real. etc.)					
Packet size limit:	65535 byt	65535 bytes						
Time								
First packet:	2006-05-2	21 16:00:52						
Last packet:	2006-05-2	21 16:00:57						
Elapsed:	00:00:04							
Capture								
Interface:	Interface: Broadcom 440x 10/100 Integrated Controll							
Dropped packets:	0							
Capture filter:	not top po	rt 3389						
Display								
Display filter:	none							
Marked packets:	0							
Traffic		Captured	Display	red				
Between first and la	astpacket	4.670 sec						
Packets		260						
Avg. packets/sec		55.679						
Avg. packet size		730.000 byte	s					
Bytes		189939						
Avg. bytes/sec		40675.701						
Avg. MBit/sec		0.325						
		lose						

图 6

38 / 43

2.2 利用 Statistics 菜单分析记录,如图 7 所示。

% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
100.00%	260	189939	0.325	0	0	0.00
100.00%	260	189939	0.325	0	0	0.00
100.00%	260	189939	0.325	0	0	0.00
100.00%	260	189939	0.325	186	144541	0.24
28.46%	74	45398	0.078	37	19021	0.00
3.46%	9	7241	0.012	2	2380	0.01
9.62%	25	15954	0.027	20	11897	0.0
0.77%	2	2381	0.004	0	0	0.0
0.38%	1	801	0.001	0	0	0.0
	% Packets 100.00% 100.00% 100.00% 100.00% 28.66% 3.46% 9.62% 0.77% 0.38%	% Packets Packets 100.00% 260 100.00% 260 100.00% 260 100.00% 260 28.46% 74 3.46% 9 9.62% 25 0.77% 2 0.38% 1	% Packets Packets Bytes 100.00% 260 189339 100.00% 260 189339 100.00% 260 189339 100.00% 260 189339 100.00% 260 189339 28.46% 74 45398 3.46% 9 7241 9.62% 25 15954 0.77% 2 2381 0.38% 1 801	Packets Packets Byte Mbity 100.00% 260 189339 0.325 100.00% 260 189339 0.325 100.00% 260 189339 0.325 100.00% 260 189339 0.325 28.46% 744 45398 0.078 3.46% 9 7241 0.012 9.62% 225 15954 0.027 0.77% 2 2381 0.001 0.38% 1 801 0.011	% Packets Packets Bytes Mbit/s End Packets 100.00% 260 189939 0.325 0 100.00% 260 189939 0.325 0 100.00% 260 189939 0.325 0 100.00% 260 189939 0.325 0 100.00% 260 189939 0.325 186 28.46% 74 45388 0.078 377 3.46% 9 7241 0.012 2 9.62% 25 15954 0.027 200 0.738% 1 801 0.001 0	% Packets Packets Bytes Mbits End Packets End Bytes 100.00% 260 18939 0.325 0 0 100.00% 260 18939 0.325 0 0 100.00% 260 18939 0.325 0 0 100.00% 260 18939 0.325 0 0 100.00% 260 18939 0.325 168 144541 28.46% 74 45398 0.078 37 19021 3.46% 9 7241 0.012 2 2380 9.62% 25 15954 0.027 20 11897 0.738% 1 801 0.001 0 0 0.38% 1 801 0.001 0 0

图 7

五、实验报告要求:

按实验报告模板撰写实验报告

实验 八 利用 Ethereal 分析 HTTP 协议

一、实验目的

利用 Ethereal 捕获一次网页打开的过程,通过观察整个网页获得全过程,加强 对 HTTP 协议的理解,通过观察捕获分组分析和理解 HTTP 协议细节和格式。

二、实验环境

Ethereal

三、实验内容

1. 利用 Ethereal 捕获分组

2. 分析捕获分组中 HTTP 协议细节

四、实验步骤

- 1. 利用 Ethereal 捕获分组
- 1.1 桌面双击 Ethereal, 启动 Ethereal, 如图 1 所示:



图 1

40 / 43

1.2 对 Capture Options 各个选项进行设置,如图2所示:

CEthereal: Capt	ure Options										
Capture											
Interface: Broadco	m 440x 10/100 Integrated Controlle	r: \Device\NPF_{6FE4BC25-9AFA-4									
IP address: 192.168.	1.189										
Link-layer header type	e: Ethernet 🗸 Buffer size: 1	🗢 megabyte(s)									
Capture packets in promiscuous mode											
🔲 Limit each packet f	to 68 😂 bytes										
Capture Filter:	tcp port 3389	•									
Capture File(s)		Display Options									
File:	Browse	Update list of packets in real time									
Use <u>m</u> ultiple files											
📃 Next file every	1 🗘 megabyte(s) 👻	Automatic scrolling in live capture									
📃 Next file every	1 minute(s)	🔲 Hide capture info dialog									
Ring buffer with	2 🗘 files	News Deschution									
🔲 Stop capture after	1 🗯 file(s)	Name Resolution									
Stop Capture		Enable MAC name resolution									
🔲 after 🛛 1	🗘 packet(s)	Enable network name resolution									
🔲 after 🛛 1	🗘 megabyte(s) 🗸										
🗌 after 🛛 1	🗘 minute(s) 👻	Enable transport name resolution									
Help	点击Start按钮开始捕获	Start Cancel									

图 2

1.3 点击 Start 按钮开始捕获分组,出现 Capture from… 对话框,如图 3 所示

lotal	260	% of total	
SCTP	0		0.0%
TCP	260		100.0%
UDP	0		0.0%
ICMP	0		0.0%
ARP	0		0.0%
OSPF	0		0.0%
GRE	0		0.0%
NetBIOS	0		0.0%
IPX	0		0.0%
VINES	0		0.0%
Other	0		0.0%
Running	00:00:15		

图 3 41 / 43

1.4 点击 Capture from ··· 对话框中 Stop 按钮结束捕获,如图 4 所示。

G Ethereal:	Capture	from Br	
Captured Pack	ets		
Total	260	% of total	
SCTP	0		0.0%
TCP	260		100.0%
UDP	0		0.0%
ICMP	0		0.0%
ARP	0		0.0%
OSPF	0		0.0%
GRE	0		0.0%
NetBIOS	0		0.0%
IPX	0		0.0%
VINES	0		0.0%
Other	0		0.0%
Running	00:00:15	top	

图 4

1.5 得到捕获记录,如图5所示。

😋 cs3http - Ethereal	
Elle Edit View Go Capture Analyze Statistics Help	
$\blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \models \blacksquare \times \Leftrightarrow \blacksquare \models \blacksquare \Rightarrow =$	> 🗢 🛧 🖢 🗐 📑 🔍 Q. Q. 🕅 🖼 🖾 🔛 💥 🔯
Eilter:	Expression Qlear Apply
No Time Source Destination	Protocol Info
1 0.000000 Dell_17:34:e2 Broadcast	ARP who has 192.168.1.1? Tell 192.168.1.189
2 0.000493 D-L1Rk_20:00:CD Dell_17:34:62	ARP 192.108.1.1 is at upon situ adu co
4 0 004619 dos situ edu co 192168 1 189	DNS Standard query response 4 202 120 1 155
5 0.012139 192.168.1.189 www.cs.situ.edu.cr	TCP 1161 > http [SYN] Seg=0 ep=0 MSS=1460 WS=0 TSV=0 TSER=0
6 0.015000 www.cs.sjtu.edu.cn 192.168.1.189	TCP http > 1161 [SYN, ACK] Seg=0 Ack=1 win=17520 Len=0 MSS=1460 WS=0 TSV=0 TSER=0
7 0.015031 192.168.1.189 www.cs.sjtu.edu.cr	1 TCP 1161 > http [ACK] seq=1 Ack=1 win=8192 Len=0 TSV=94225 TSER=0
8 0.015084 192.168.1.189 www.cs.sjtu.edu.cr	HTTP GET / HTTP/1.1
9 0.128126 www.cs.sjtu.edu.cn 192.168.1.189	TCP [TCP segment of a reassembled PDU]
10 0.129918 www.cs.sjtu.edu.cn 192.168.1.189	TCP [TCP segment of a reassembled PDU]
11 0.129946 192.168.1.189 www.cs.sjtu.edu.cr	1 TCP 1161 > http [ACK] Seq=407 Ack=1671 Win=8192 Len=0 TSV=94226 TSER=3484561
12 0.134069 WWW.CS.Sjtu.edu.ch 192.168.1.189	TCP [TCP segment of a reassembled PD0]
14 0 129674 102 169 1 190 www.cs.sjtu.euu.cn 192.100.1.109	TCP [TCP segment of a reassembled point of a reassembled point of the start of the start of a reassembled received and the start of the
15 0 145552 www.cc.citu.edu.cn 102 168 1 189	TCP ITCP segment of a reasonabled PDUI
16 0 149749 192 168 1 189 www.cs.situ.edu.cr	TCP [162 Segment of a reassembled roo]
17.0.161547 www.cs.situ.edu.cn.192.168.1.189	TCP [TCP segment of a reassembled PDU]
18 0.161589 192.168.1.189 www.cs.situ.edu.cr	TCP 1161 > http [ACK] Seg=407 Ack=7463 win=8192 Len=0 TSV=94226 TSER=3484561
19 0.161608 www.cs.situ.edu.cn 192.168.1.189	TCP [TCP segment of a reassembled PDU]
20 0.161646 192.168.1.189 www.cs.sjtu.edu.cr	1 TCP 1161 > http [ACK] seg=407 Ack=8911 win=6744 Len=0 TSV=94226 TSER=3484561
21 0.165817 www.cs.sjtu.edu.cn 192.168.1.189	TCP [TCP segment of a reassembled PDU]
22 0 165854 www.cs situ adu co 102 168 1 180	TCD http://iko//sev.acv1/sep.0/acv1/wip_17530/lep_0/wss_1460/wss0/tsv20/tsep_0
I Frame 9 (288 bytes on wire, 288 bytes captured)	
Ethernet II, Src: D-Link_2d:00:cb (00:0f:3d:2d:00:cb),	. Dst: Dell_17:34:e2 (00:15:c5:17:34:e2)
Internet Protocol, Src: www.cs.sjtu.edu.cn (202.120.1.	.155), Dst: 192.168.1.189 (192.168.1.189)
■ Transmission Control Protocol, Src Port: http (80), Ds	st Port: 1161 (1161), Seq: 1, Ack: 407, Len: 222
0000 00 15 c5 17 24 a2 00 0f 2d 2d 00 ch 08 00 45 00	
0010 01 17 90 0d 40 00 7h 06 e0 5f ce 78 01 9h c0 e8	
0020 01 bd 00 50 04 89 e4 23 04 eb 3b 7c c9 43 80 18	P. # :1.C.
0030 42 da 63 fc 00 00 01 01 08 0a 00 35 2b 91 00 01	B.cŚ+
0040 70 11 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f	p.HTTP/1 .1 200 O
0050 Mb 0d 01 52 65 72 76 65 72 25 20 4d 60 62 72 6E	K Serve P MS-200
rame (rame), zee bytes	[P: 261 D: 261 M: 0

图 5

- 2. 分析捕获分组中 HTTP 协议细节
- 2.1 利用 Analyze 菜单 Summary 分析记录。如图 6 所示。

🕲 Ethereal: Su	mary						
File							
Name:	D:\TEMP\ether>000KZIG29S						
Length:	194123 bytes						
Format:	libpcap (tcpdump, Ethereal, etc.)						
Packet size limit:	65535 bytes						
Time							
First packet:	2006-05-21 16:00:52						
Last packet:	2006-05-21 16:00:57						
Elapsed:	00:00:04						
Capture							
Interface:	Broadcom 440x 10/100 Integrated Controller						
Capture filter:	U not ton nort 2290						
Display	nor top po						
Display Display filter	none						
Marked packets:	0						
Traffic	1	Captured	Displayed				
Between first and last packet		4.670 sec					
Packets		260					
Avg. packets/sec		55.679					
Avg. packet size		730.000 bytes	5				
Bytes		189939					
Avg. bytes/sec		40675.701					
Avg. MBit/sec		0.325					
Close							

图 6

2.2 利用 Protocol Hierarchy Statistics(协议层次统计)观察各个协议的统计。如图7所示。

Ethereal: Protocol Hierarchy St	atistics	÷.					
Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
l Frame	100.00%	260	189939	0.325	0	0	0.00
Ethernet	100.00%	260	189939	0.325	0	0	0.00
Internet Protocol	100.00%	260	189939	0.325	0	0	0.0
Transmission Control Protocol	100.00%	260	189939	0.325	186	144541	0.2
🖃 Hypertext Transfer Protocol	28.46%	74	45398	0.078	37	19021	0.0
Line-based text data	3.46%	9	7241	0.012	2	2380	0.0
Compuserve GIF	9.62%	25	15954	0.027	20	11897	0.0
JPEG File Interchange Format	0.77%	2	2381	0.004	0	0	0.0
		Ōĸ	ן				

图 7

五、实验报告要求:

按实验报告模板撰写实验报告