|  |  |  |  |
| --- | --- | --- | --- |
| **《计算机网络》实验报告** | | | |
| **实验编号** | 实验2 | **实验名称** | Ethernet |
| **姓名** | 李彦浩 | **学号** | 202100300063 |
| **班级** | 工业软件班 | **成绩** | （空着） |
| 1. **实验目的**   探究以太网帧的细节。以太网协议是一种通用的数据链路层协议，现代计算机通过以太网交换机连接到网络。 | | | |
| 1. **实验要求**   wireshark：使用wireshark软件工具用于捕获和检查数据包跟踪。。  ping：使用“ping”来发送和接收消息。ping是一个标准的命令行实用程序，用于检查另一台计算机是否有响应。 | | | |
| 1. **实验内容**   例子：   1. Capture a Trace. 2. Inspect the Trace 3. Ethernet Frame Structure 4. Scope of Ethernet Addresses 5. Broadcast Frames 6. Explore on your own | | | |
| 1. **实验过程**  Step 1:Capture a Trace(链路追踪) 尝试使用标准命令行指令ping进行链路追踪。   1. ping一个远程服务器，[比如www.baidu.com](http://xn--www-eo8er22f.baidu.com)。   1  启动wireshark。通过过滤器“icmp”来捕获以太网帧。确保捕获选项中的”enable MAC name resolution/Resolve MAC address”开启，这可以提供网络供应商的信息。并且确保Link-layer header type选项是Ethernet。  2  3  捕获开启后，在命令行ping一个远程服务器，wireshark将捕获ping数据包。  4 Step 2: Inspect the Trace(追踪分析) 即便我们在802.11接口下（无线网）捕获流量，看到的依旧是Ehternet（有线网）header。因为我们在捕获时的选项是Ehternet模式，因此操作系统将802.11的帧头信息转换成了伪以太网帧头（pseudo-Ethernet header)。所以我们看到的总是(pseudo)Ethernet header。  tips:   1. 有两种以太网。IEEE 802.3和DIX Ethernet(在wireshark中叫做Ethernet II)。wireshark捕获的是后者。 2. 在wireshark中没有展示附加信息位(preamble)。这是物理层机器帮助网络接口卡（NIC,Network Interface Card)识别成帧开头的。除了这个没有任何用。前导码共8个字节，前7个均为10101010，最后一个字节10101011。   5  有destination和source address字段。wireshark通过解码其中的OUI（Organizationally Unique Identifier)部分来获取NIC的供应商，比如Intel。  6  有一个Type字段。对于ping而言，Type应该是IP。意味着以太网负载了IP的包。同时还有一个Length字段。在DIX Ethernet中，这个字段由接受方的硬件决定，从合法成帧的前置附加信息位开始，到校验和结束。并附在数据包中传给高层。  8   1. 没有Data字段。后面紧跟的是IP帧头。 2. 没有pad位。pad位将在以太网帧小于64个字节（以太网帧的最小size)时附加在帧尾。 3. 大部分抓到的包中没有校验和（即便校验和确实存在）。一般来说以太网硬件在接受并检查之后就会把这部分扔掉。Wireshark一般很难见到。 4. 也没有VLAN字段。  Step 3: Ethernet Frame Structure(以太网帧结构) 标识出以太网头部字段的位置和字节大小，以及以太网帧有效负载的范围。在最后加上一个4个字节的16位校验和。（虽然wireshark中显示不出来，但是确实是有的）。  9  如图所示，以太网帧头为14个字节。包括了目标MAC，源MAC和Type字段。  因此以太网帧结构如下：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Destination MAC 6B | Source MAC 6B | Type 2B | Ethernet Payload 46-1500B(图中为60B） | Check Sum 4B | | | | |
| Ethernet Payload字段最少46B是因为已经包含了Pad字段。Pad字段用来填充少于46字节的数据字段（46 = 64（最小长度）-14（Ethernet帧头长度）-Check\_Sum(4B）） Step 4: Scope of Ethernet Addresses(以太网地址范围) 每个以太网帧都有源地址和目标地址。但这并不是本机或者远程主机的的以太地址，因为以太帧只能在局域网中寻址，因此这个地址实际上是路由器或者默认网关的以太地址。  而IP协议块中的IP地址展示了完整的源于目标终端。这分别对应本机于目标主机。  10  根据抓到的包，得到Ethernet地址，IP地址   |  | **源** | **目的** | | --- | --- | --- | | IP地址 | 10.27.243.63 | 39.156.66.10 | | Ethernet地址 | 48:89:e7:82:05:fa | 74:25:8a:e3:6d:a9 |   源IP是不变的，校园网分配的静态IP。但目标IP是会变的，因为我ping的百度，百度是上了CDN和负载均衡的，流量分配到哪个服务器节点是未知的。  而Ethernet地址的目的MAC，是当前子网网关MAC地址。  终端-路由节点示意图如下。（draw.io画的）  11 Step 5: Broadcast Frames(组播帧) 目标地址有三种形式，用目标地址的第一位进行区分：   1. 第一位为0，对应普通地址。   12  第一位为1，对应组地址（组播）。(Multicast traffic)只对组内需求者进行转发。  13  Multicast有一个特殊形式，如果组播预留以太网地址（reserved ethernet address)，那么称为Broadcast。对所有终端进行通信。  14  全为1，对应特殊地址广播。(unicast traffic)  通过过滤器”ether multicast”捕获组播以太网帧。  15  等待30s以捕获背景流量。对于大多数以太网来说，一般有一个稳定的交互背景流量以保持网络状态，所以不需要启动某一个应用程序来捕获流量了。而”ether multicast”将捕获multi/broadcast的以太网帧，但一般不捕获unicast帧。  16 Q1 : 广播以太网址的标准形式是什么？ 只要抓到那一帧就好说了。  17  可以看到，广播的以太网址为ff:ff:ff:ff:ff:ff。 Q2 : 以太网址的哪一个比特用来区分这一帧属于单播还是组播/广播？ 抓一个单播和一个广播的就行。  18  这里可以看到unicast在第一个字节的最后一位为0，而multi/broadcast的第一个字节的最后一位为1。  所以是第一个字节的最后一个比特，也就是说第8个比特。 Explore on your own(IEEE 802.3) 过滤器“llc”（lower case for “LLC”)  我抓不到802.3协议下的以太网帧。只能那实验指导书里的样例了。（据说这样的帧很稀有）  19 Q1 : IEEE 802.3和LLC协议帧头有多长？相比于DIX Ethernet如何？注意到Trailer/Padding字段和校验和应该属于这个部分，但这两个字段可能出现在帧尾。 DST6B，SRC6B，Legth2B，Trailer8B，校验和4B，总计26B。DIX Ethernet已经提过了，14+4=18B。 Q2 : 接受方计算机是怎么知道某一帧是遵循DIX Ethernet还是IEEE 802.3的？提示：使用wireshark看一下Ethernet协议格式描述。 对比一下两种写一下的Ethernet头部。DST和SRC都是6B，但由于数据段长度最多为1500B。所以大于0X05DC的所有值都可以作为Type类型。这样通过比较第13、14字节的数据大小就可以知道是IEEE 802.3还是DIX Ethernet了。比如0x0800→IPv4,0x86DD→IPv6,0x0806→ARP。 Q3 : 若IEEE 802.3没有Type这个字段，更高层是如何判断自己的协议的？利用wireshark来寻找多路分解键。(demultiplexing key) 20  需要看Logical-Link Control块中的DSAP和SSAP字段。LLC子层（逻辑链路控制子层）包括了DSAP（目的服务访问点，Destination Service Access Point)和SSAP（源服务访问点，Source Service Access Point)字段。这两个字段组合在一起形成多路分解键的功能。  例如，二者均取0xff时，该帧为Netware-ETHERNET帧，用来承载NetWare类型的数据。都取0xaa时，变成ETHERNET\_SNAP帧。  本图中都取的0X42，该帧就是用于运行STP（生成树协议，Spanning Tree Protocol）的。  为了证明我的实验心得确实超过了一页A4纸，我就从下一页开始写实验心得了。因此，你可以在下一页看到我的实验心得。  **四．实验心得**  本次实验内容是以太网协议层。这是一个驻留在数据链路层上的协议。之前在理论课的课堂上我们也学习过数据链路层上的一些基本概念与协议。比如成帧，差错、流量控制，停等协议、滑动窗口协议等。但那只限于理论。虽然我们知道了数据链路层在传输和接受数据帧的策略，但不太了解Ethernet协议的格式。比如源MAC地址，目标MAC地址，多路分解键在协议头的位置。包括IEEE 802.3帧与DIX Ethernet帧在协议字段上的区别。  首先实验指导书教给我标准命令行指令ping，用于检查网络中的另一台计算机是否有响应。例如ping www.baidu.com可以检查百度的节点服务器是否有响应。随后尝试在”icmp”过滤器（Internet控制报文协议）中捕获ping的数据包。  随后我们就开始分析捕获到的数据帧。WireShark中展示出来的数据帧和真实的协议帧头并不完全吻合。比如wireshark中是不展示padding字段或preamble字段的。后者是用来补充整个帧以符合最小帧长的。（这个就是课上提到的，为了能够检测冲突，最小帧长l=2t\*最大比特率，其中t为最远两个站之间的传播时延）而后者是前导码，用来成帧/帧同步，即理论课上提到的将每一帧区分开的方式。像DIX Ethernet中的区分方式就是前导码前7B为10101010，最后1B为10101011。但在计算上我们应该把这两帧的长度计入Ethernet协议的帧头中。  除了这些不同之处，实验指导书还为我们介绍了一些概念。比如目标MAC地址与源MAC地址，以及他们的含义。比如wireshark可以解码组织识别码部分来区分不同网络接口卡的供应商。还有IEEE 802.3中采用Length字段来标识数据段长度，并顶替了原有的Type字段所代表的多路复用键。  第三步中要求我们自己理清以太网帧结构。首先实验指导书提醒我们:wireshark中是不展示校验和字段的，但是我们知道帧传输时一定有4B的校验和。因此需要在后面附上这个字段。根据wireshark的抓包结果，我给出了Ethernet帧的头部字段示意图。  第四步要求我们画出主机之间以太逻辑地址与路由器的交互示意图。之所以会有路由器/网关在这个流程中出现，是因为目标MAC地址并非远程主机的地址。因为以太地址只能在局域网中寻址。所以目标MAC地址显示的是路由器或者子网默认网关的以太地址。不过IP地址确实对应了源计算机与目标主机的IP地址。在实验指导书的引导下，我画出了主机与路由器的交互示意图，并且在上面标识了Ethernet和Internet作用的位置，并根据wireshark抓包结果，列出了MAC地址与IP地址表。  第五步提到了一个很重要的概念。叫做组播/广播。一个数据帧并非只能仅仅指定一台目标主机进行传输，而是可以在群组中转发，甚至可以向所有其他主机广播这个数据帧。因此广播也称为组播的一种特殊形式。我自己从网上查阅了一些资料，了解单播、组播、广播的概念，并找了一些示意图插入到自己的报告中，以此形成更具象化的记忆与理解。  第五步的重要问题是：如何区分单播、组播、广播？这一定是通过协议中的某些字段来进行区分的。因此我们要先尝试捕获到单播、组播、广播的数据包。然后调查其中的字段，看看是哪些字段将他们区分开了，并且归纳总结这种区分的规则。我按照实验指导书的提示，直接捕获背景流量，就抓到了UDP协议下的背景流量（为了保持网络状态，计算机之间一直会维护一个基本的流量过程）。由于在逻辑上广播是组播的一种特殊形式，因此我想应该先把单播和组播区分开了，然后再区分一般组播和广播。我先查看了单播的标识字段，wireshark中很清晰的标识，在目标MAC地址中第一个字节的最后一个比特为0的，就是单播；与此同时，目标MAC地址中第一个字节最后一个比特为1的，就是组播。那么一般组播和广播有什么区别呢？我仔细观察了multicast和broadcast的以太地址，发现广播的以太地址一定是ff:ff:ff:ff:ff:ff。也就是全为1。通过这一特殊的地址，便把组播和广播区分开了。  在Explore On Your Own中，实验指导书安排我们去发掘IEEE 802.3工作组制定的协议规范。IEEE 802.3协议和DIX Ethernet协议本质上都是以太网协议，但是其中的细节不同，最大的区别就是，前者在12B的MAC地址后是Length字段，标识了后续数据段长度。但是后者在12B的MAC地址后是Type字段，标识了多路分解键。比如IPv4。那么问题就来了，同样的2B，接收方计算机如何通过协议帧头来区分这是IEEE 802.3还是DIX Ethernet呢？通过对后续数据段的分析，我发现数据段的最大长度为1500字节，也就是说最大编码为0x05DC,那么高于0x05DC的编码就都可以用来作为Type字段的类型了。比如IPv4的Type字段就是0x0800。  但既然Type字段没了，相当于就没有多路分解键了。更高层怎么知道自己的协议是什么呢？这就涉及到IEEE 802.3的顶层协议LLC（Logic Link Control)协议了，这个协议中有两个字段，一个是DSAP（目标服务访问点，Destination Service Access Point)，一个是SSAP（源服务访问点，Source Service Access Point)。这两个字段共同组成了多路分解键。比如实验指导书中提供的截图中DSAP和SSAP字段均为0x42，这就代表高层协议为STP协议（生成树协议,Spanning Tree Protocol)。  通过Lab2，我学习到了以太网协议层的格式与各字段的内涵。还有其不同实现方式的区别（IEEE 802.3与DIX Ethernet）。这将对之后计算机网络的学习起到重要的铺垫作用。 | | | |