第 3 章

“包”罗万象——三层协议分析

在第 2 章我们了解了网络层次结构中的第二层协议：VLAN 和 ARP 的原理和分析方法，本章顺序向上介绍第三层网络层协议。

3.1 车水马龙的数据包世界——IP 是如何工作的

如果想深入了解网络流量是如何在不同的网络中传输的，就一定要了解网络搬运工“IPv4”(后文简称 IP)，还有当前正在稳健发展的 IPv6。

要了解 IP，首先需从 IP 地址和 IP 报头(Packet Header)这两块内容着手。

3.1.1 网络层地址

网络上的通信会使用逻辑地址(即网络层地址、IP 地址)和物理地址(数据链路层地址、MAC 地址)。IP 地址允许不同广播域之间的设备进行相互通信，MAC 地址则用于同一广播域中直接使用交换机互联的设备之间的通信，通常情况下，正常通信需要这两种地址协同工作。关于这两种地址协同工作的 ARP，已经在上一章有所介绍，本节主要介绍 IP 地址。

IP 地址是一个 32 位(4 字节)的地址，用来标识一台连接网络的设备，这台设备的 IP 地址在网络内是唯一的。IP 地址是一串长度为 32 位的二进制字符，如 11000000101010000000000100000001。不难发现，IP 地址表示起来极其不方便。因此，IP 地址有一种压缩方法，即点分十进制表示法，该方法将 32 位的地址分成 4 个 8 位的段，每段之间用“.”分隔，再将每一段的数字从二进制转换为十进制。由此，上文中提及的 32 位字符串可表示为 192.168.1.1。

IP 地址分为 5 类：A 类地址、B 类地址、C 类地址、D 类地址(用于组播)、E 类地址(用于科研)。不同类别的地址具有不同长度的网络位和主机位，具备相同网络位的地址之间属于同网段 IP。为了标识 IP 地址中哪些位是网络位，哪些位是主机位，需引入子网掩码的概念。

子网掩码与 IP 地址等长，都是 32 位，由连续的 1 和 0 组成，1 对应的位为网络位，0 对应的位为主机位。例如上文中所叙述的 192.168.1.1 地址，若前 24 位为网络位，后 8 位为主机位，则对应掩码前 24 位为 1，后 8 位为 0，可以记为 11111111111111111111111100000000、255.255.255.0 或/24，计算机通过把 IP 地址和子网掩码转换成二进制后，将二进制的 IP 地址和子网掩码进行“逻辑与”运算，得出 IP 地址所属的网段号。

每当计算机和其他 IP 进行通信时,都需要将对方的 IP 地址和本地网段进行比较,若发现目的 IP 地址和本机的 IP 地址属于同一网段,则在本广播域进行数据包转发,直接将对方 IP 对应的 MAC 地址填写在二层目的 MAC 字段,这个过程需要 ARP 配合直接请求目标 IP 地址对应的 MAC 地址;如果目的 IP 地址和本机的 IP 地址不在同一网段,则把数据包发送到本机的网关地址(网关一般是一台具有路由功能的网络设备),由网关将数据转发到其他广播域(网关通过查询路由下一跳转发到其他广播域)。

在 IP 地址中,有些特殊地址被某些协议保留,永久专用;有些特殊地址只能在私网中使用,不能用于公网;还有些特殊地址有其他的作用,这些特殊地址不能用于普通组网。特殊地址列表如表 3.1 所示。

表 3.1 特殊 IP 地址列表

地 址	特 殊 用 途
0.0.0.0	本地网络中的主机,仅作为源 IP 地址使用
127.0.0.0/8	主机回送地址,通常只用 127.0.0.1
169.254.0.0/16	IP 链路本地地址,只用于一条链路,通常自动分配
192.0.2.0/24	用于 TEST-NET-1 地址,不会出现在公共互联网中
192.88.99.0/24	用于 6to4 中继(任播地址)
224.0.0.X/24	IANA 保留组播地址,仅作为目的 IP 地址使用
255.255.255.255/32	本地网络(受限的)广播地址
10.0.0.0/8	专用网络(内网)地址,不会出现在公共互联网中
172.16.0.0/12	专用网络(内网)地址,不会出现在公共互联网中
192.168.0.0/16	专用网络(内网)地址,不会出现在公共互联网中

3.1.2 IP 数据包格式

IP 数据包格式如图 3.1 所示。数据包中的大多数字段对网络层转发和网络流量分析都有很重要的作用。

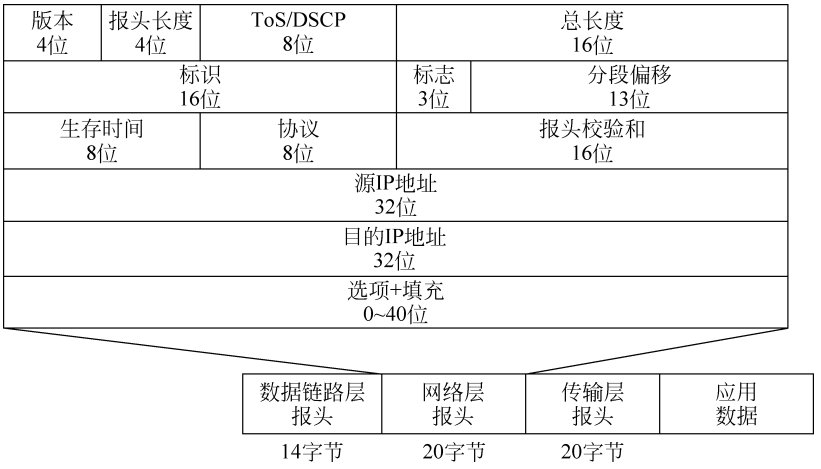


图 3.1 IP 数据包格式

下面详细介绍IP数据包的组成。

(1) 版本(Version)：该字段长度为4位，定义了IP的版本，目前常见的版本是IPv4。这个字段向接收方运行的IP协议栈指出该IP数据包使用的版本和格式，当读取的报头信息为0100时，代表是IPv4数据包，接收方将按照图3.1中的IPv4报头格式对数据流进行解码，当读取的报头信息为0110时，代表是IPv6数据包，接收方将按照IPv6报头格式对数据流进行解码(IPv6报头格式将在3.6节介绍)。

(2) 报头长度(Header Length)：该字段长度为4位，定义了IP报头的长度，以4字节为单位计算，IP报头的长度是可变的(在20～60字节)，将该字段中的值乘以4得出当前数据包的报头大小，例如报头长度值为0101，将该值转换成十进制得到结果5，再乘以4后得出当前报头长度为20字节。

(3) ToS/DSCP(Type of Service/Differentiated Services Code Point，服务类型/差分服务代码点)：该字段长度为8位，用于声明数据包在网络中的转发优先级，用以支持网络服务质量(Quality of Service，QoS)。最早由RFC791定义，前3位为优先级，接着4位分别为relay、throughout、reliability、cost，最后一位保留，这些位的使用方式在RFC1349中有详细介绍，现已废弃不用。改用RFC2474中的DSCP差分服务代码点，前6位为DSCP优先级，后2位保留，其中定义了CS、AF、EF、BE等概念。

(4) 总长度(Total Length)：该字段长度为16位，以字节为单位定义了数据包的总长度(报头长度＋数据长度＝总长度；反之，数据长度＝总长度－报头长度)。

(5) 标识(Identification，ID)、标志(Flag)、分段偏移(Fragment Offset)：这三个字段长度分别为16位、3位、13位，三个字段共同实现了IP数据包的分片功能，有关分片功能的细节，将在下一节进行详细介绍。

(6) 生存时间(Time To Live)：该字段长度为8位，为最大路由器跳数，每处理经过一个路由器，值递减1，如果减1后这个字段值变为0，路由器就丢弃这个数据包。该字段用于消除网络三层环路对设备性能带来的影响，是一个网络发生三层环路后的补救措施。

(7) 协议(Protocol)：该字段长度为8位，定义了使用此IP包后面携带的高层协议，若该字段值为1，则表示高层协议为ICMP；常见的值还包括2为IGMP、6为TCP、17为UDP、89为OSPF等。

(8) 报头检验和(Header Checksum)：该字段长度为16位，IP分组的检验和仅覆盖报头，不管数据部分，其对IP报头进行校验，若出现错误，则丢弃数据包。

(9) 源IP地址：该字段长度为32位，发送数据包的主机IP地址。

(10) 目的IP地址：该字段长度为32位，发送数据包的目的IP地址。

(11) 选项＋填充：长度为0～40位不等，IP选项包含一些IP数据包中不具备的功能字段，当需要使用这些功能时，这些功能附带在IP报头的后面来实现，如让数据包在转发时记录时间戳，或让数据包在转发时记录转发的路由器，或遵循一些特定的路径进行转发等，这些记录的信息填充在选项字段，附加在IP包后面，选项的信息类别如图3.2所示。

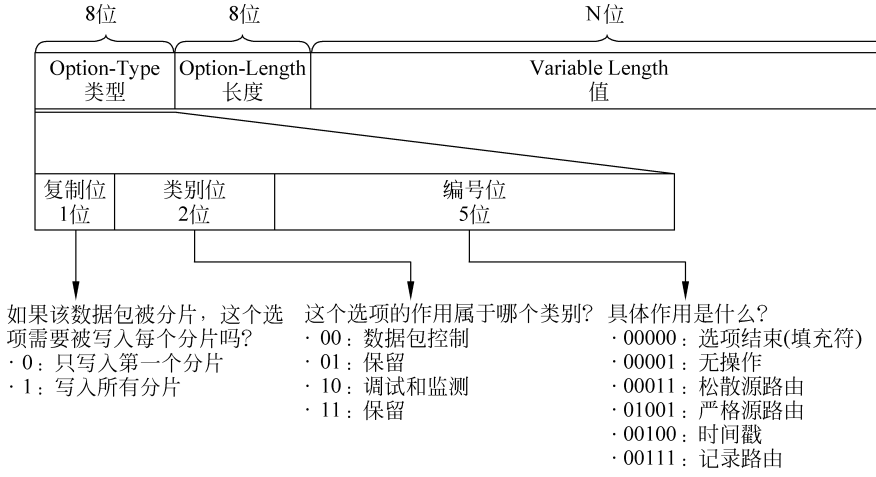


图 3.2 IP 选项

3.2 一车拉不下——如何重组被分片的数据包

当 IP 数据包长度超过了网络的 MTU 限制(假设有 5000 字节,正常 MTU 大小为 1500 字节),一车拉不走的时候,需要对数据包进行分片发送(分几车拉走)。

IP 数据包设定的总长度是 1500 字节,算上 IP 报头本身的 20 字节,也就是一次携带最多 1480 字节的有效载荷,按照这个标准,发送 5000 字节的数据实际上要分成 4 个分片进行发送。进一步考虑如下问题。

(1) 问题一：分片发到了对端,对端如何知道该将哪几个分片进行重组?

因为网络中每一秒都有很多的数据包,假设 1s 内发送了 5 个 5000 字节的数据包,则将会产生 20(5×4)个分片到达对端,对端应该如何把这 20 个分片正确重组为 5 个数据包呢? 此时需要使用到标识字段(下文简称 ID)去辨别。网络中的每一个 IP 数据包都具有一个唯一的 ID 号,比如第一次发的包 ID 为 12345,第二个包就应该是 12346,然后是 12347……当数据包被分片之后,被分出的那几个片应该继承数据包原有的 ID 号,换句话说,几个分片应具有相同的 ID 号。接收方可以将具有相同 ID 号的分片进行重组,以确保重组是正确的。

(2) 问题二：对于发送方来说,发送方会根据自身的 MTU 对数据包进行分片切割,那么对于接收方来说,如何判断一共有多少个分片,以及后面到底还有没有其他分片?

标志字段就是为了解决这样的问题而设计的。该字段一共有三位,三位分别是保留位、DF(Don't Fragment,禁止分片)位和 MF(More Fragment,更多分片)位。当 DF 设置为 1 的时候,表示这个数据包禁止被分片。也就是说,货物必须被一车拉走,如果一车拉不走的话,干脆就别拉了。这就是 DF 的作用。

在 Windows 主机中,使用 ping 命令后面加上-f 参数就可以给发送的 ping 包设置 DF 启用。后面的 MF 的意思是“还有更多”,前文提到数据包被分成 4 片之后,4 片具有相同

的ID号。到底是被分了几片？哪一片是最后一片？这时候就通过MF来表示。MF＝1表示“后面还有其他分片”，MF＝0表示“这是最后一个分片”。假如数据包被分了4片，它的前3片都应该是把MF设置为1的。这样对端收到了前3片之后，就能知道后面还有几片还没有收到，应该暂时不进行重组操作。当收到最后MF为0的分片的时候，就可以开始重组了。

(3) 问题三：接收到数据的顺序可能是混乱的，怎么控制？

网络中的数据包可能存在负载的路径，即去往同一个点可能有多条不同的路。这样的话当数据包走不同的路到达目的地时，可能由于网络拥塞的原因出现“先出发的反倒来晚了”，和生活中的情况很类似。这样一来对于接收方来说，接收到数据的顺序可能是混乱的。因此，如果草率按照接收到数据的顺序对分片包进行重组的话，可能会导致重组后的数据包内容的顺序不正确。

实际上，问题三是最好解决的，给每一个分片包分配一个编号，让接收方按照编号顺序进行重组就可以了。但IP对这件事情看待得更加严谨，并不为包进行编号，而是为每字节进行编号。下面的例子对编号进行了解释。

假设网络MTU为1500，发送的IP数据包大小为5000字节：

第一个分片发送的数据为第0～1479字节(计算机是从0开始计数的)。

第二个分片发送的数据为第1480～2959字节。

第三个分片发送的数据为第2960～4439字节。

第四个分片发送的数据为第4440～4999字节(因为是从0开始计数的，所以最后1字节是第4999字节)。

了解上述原理后，就明白“分段偏移”的作用了。分片包通过将自己所承载的首字节编号填写到“分段偏移”字段中，即把前文中叙述的0、1480、2960、4440填充到“分段偏移”字段，接收方通过这些数值即可对分片包按照正确的顺序进行重组，这些数值称为“偏移量”。

但由于IP报头的设置，段偏移字段的长度只有13位，能够表示的最大偏移量为8192(2^{13})，而IP包总长度字段的长度有16位，能够表示的最大长度为65536(2^{16})，当一组分片数据包的总长度超过8192时，由于偏移量数字限制的原因，无法再为8192以上的数字表示段偏移。这就好像银行卡存款余额设置了最大13位，因此存款上限是9999999999999元；如果希望存更多钱，则由于存满了，银行系统不支持了。当然，存款的例子是十进制的，而计算机网络数据包内容是二进制的。

为了解决这个问题，RFC791文档规定：段偏移字段中写入的偏移量大小是真实数据的大小除以8，即可解决上述问题。因为IP报头格式中的总长度、段偏移字段位数正好差了3位，因此是8(2^3)。正是由于这里的8倍差概念，RFC规定分片包除了MF＝0的分片外，所有MF＝1的分片的数据量大小必须为8的倍数。

学习到这里，再回过头看标识、标志、分段偏移这三个字段。

(1) 标识：该字段长度为16位，每一个数据包都具有一个唯一的标识，多用于分片。所有的分片都具有相同的标识，具有相同标识号的分片组装成同一个数据包，使得数据包分片和重组程序通过该字段完成，以确保不同数据包的片段不混在一起。

(2) 标志：该字段长度为 3 位，用于分片。第一位保留；第二位称为“禁止分片”位，若值为 1，则不能分片，若无法转发出去则丢弃，并返回 ICMP 差错数据包；第三位是“更多分片位”，为 1 表示后面还有分片，为 0 表示为最后一片。

(3) 分段偏移：该字段长度为 13 位，用于分片。13 位字段表示分片在整个数据包中的相对位置，是数据在原始包中的偏移量，以 8 字节为度量单位。由于 13 位二进制数字最大能表示的十进制数为 8191，因此一个 IP 分片所能携带的最大数据量为 8191 乘以 8，即 65528 字节。

分片情况如图 3.3 所示，这个实例是基于 MTU 为 1500 的情况，分片 3000 字节的数据，左侧为分片前，右侧为分片后。

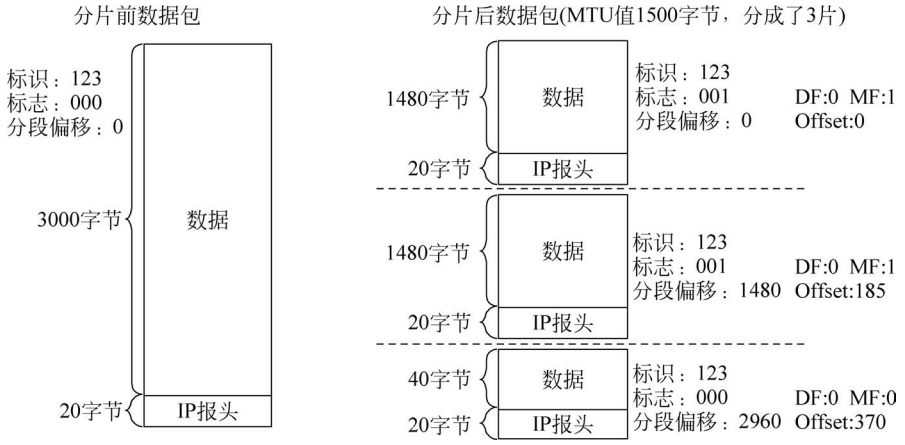


图 3.3 IP 包分片示例

图 3.3 展示了经过 IP 承载时的分片情况，左侧是分片之前，需要注意图中的“分段偏移”是手工计算出来的偏移量，而 Offset 则是真正在传输时使用的偏移量(数值除以 8)。

实际上，标识、标志、分段偏移三个字段不仅能够用于数据包分片，在流量分析领域，这些字段也可以起到很重要的作用。

例如，在工作中需要对疑难故障进行排查，怀疑故障来自防火墙系统错误时(注意是系统错误而不是策略配置错误，防火墙策略配置错误比较容易凭借经验发现，而系统错误是网络设备研发人员的代码问题导致数据转发出错，这类故障可能导致对应该修改后再转发的数据包直接进行了透传处理，或对应该透传的数据包修改后进行了转发等)，要通过流量分析发现此类故障，可以在防火墙的流量出入端口分别去捕获同一条会话，观察这条会话数据包在墙前和墙后的区别。但某些防火墙启用了网络地址转换功能，数据经过防火墙之后的 IP，甚至是端口都已经被改变了。经过网络地址转换和未经网络地址转换的数据包的地址和端口都不一样，因此很难去分辨墙前和墙后的同一组流量。此时，可以通过 IP 数据包的 ID 字段去判断，因为网络地址转换只改变 IP 地址和端口，不会改变数据包 ID，因此通过寻找具有相同 ID 的包来判断这是否是网络地址转换前后的同一组流量是 ID 字段在流量分析领域的巧妙用法。

3.3 IP丢包了怎么办

由于IP数据包的IP处于第三层，IP数据包的转发属于尽力而为，因此IP对于丢包是没有通知机制的，但是可以用ICMP的通告机制来弥补IP不具备的通知功能。从这一节中可以看到，原来大家认为ICMP的ping工具只能测试网络联通性，实际上ICMP不仅能够测试网络联通性，还能用来传输报告差错的包和网络控制包。

3.3.1 ICMP概述

IP是一种不可靠的协议，无法进行差错控制，因此ICMP(Internet Control Message Protocol，互联网控制报文协议)被设计出来用于弥补IP层的缺陷。ICMP是IP的伴侣，它们的关系如图3.4所示。

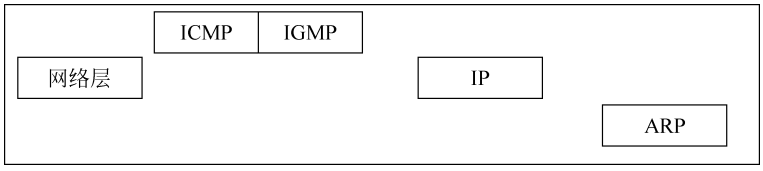


图3.4 ICMP与IP的关系

ICMP允许主机或路由器报告差错情况和提供有关异常情况的报告。一般来说，ICMP数据包提供针对网络层的错误诊断、拥塞控制、路径控制和查询服务4项大的功能。

ICMP可用于ping、traceroute、路由表动态更新、路径MTU发现、提供发现问题的线索、服务拒绝(UDP)、作为攻击手段、识别目标操作系统、识别目标上开启的服务类型、非授权地修改路由表等。

3.3.2 ICMP数据包的格式

ICMP本身是一个网络层协议。ICMP数据包首先要封装成IP数据包，才会被转交到数据链路层、物理层处理后进行发送。

在一个IP数据包中，如果协议字段值是1，就表示IP上层数据是ICMP数据包，ICMP数据包所在的位置如图3.5所示。

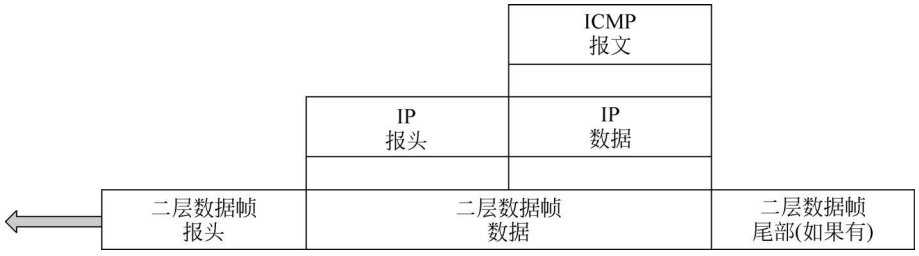


图3.5 ICMP数据包所在的位置

ICMP 数据包由一个 8 字节的报头和不固定长度的数据部分组成。每一种 ICMP 数据包类型的报头的格式都不同,但是所有 ICMP 数据包的前三个字段是一致的,都是“类型”“代码”“校验和”。ICMPv4 数据包的格式如图 3.6 所示。

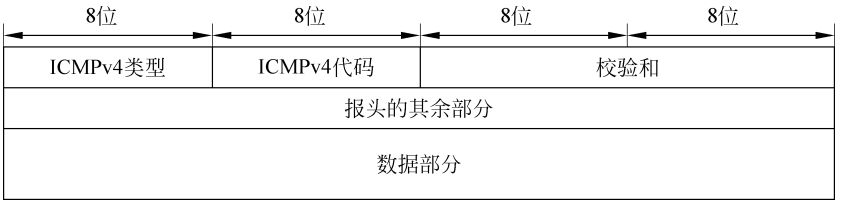


图 3.6 ICMPv4 数据包的格式

ICMP 数据包中各字段的意义如下。

(1) 类型:声明这个 ICMP 数据包的类型,例如回显请求包、差错控制包等。常见的 ICMP 数据包类型值为 8 表示回显请求(ping 请求),类型值为 0 表示回显应答(ping 应答)。ICMP 数据包的其他类型如表 3.2 所示。

表 3.2 ICMP 数据包类型列表

类 型	ICMP 数据包类型的描述
0	回显应答(ping 应答,与类型 8 的 ping 请求一起使用)
3	目的不可达
4	源点抑制
5	重定向
8	回显请求(ping 请求,与类型 0 的 ping 应答一起使用)
9	路由器公告(与类型 10 一起使用)
10	路由器请求(与类型 9 一起使用)
11	超时(0 和 1)
12	参数问题
13	时标请求(与类型 14 一起使用,用于发送方计算往返时间)
14	时标应答(与类型 13 一起使用,13、14 已作废)
15	信息请求(与类型 16 一起使用,已作废)
16	信息应答(与类型 15 一起使用,已作废)
17	地址掩码请求(与类型 18 一起使用)
18	地址掩码应答(与类型 17 一起使用)

(2) 代码:声明这个类型中的详细操作,例如当类型为 3 的 ICMP 不可达包代码字段声明了造成不可达的具体原因,后文将对各种 ICMP 不可达类型进行介绍。

(3) 校验和(第 3、4 字节):2 字节的校验和字段用于检测 ICMP 数据包在传输过程中是否发送错误,校验和的计算覆盖了整个包(报头和数据)

(4) 报头的其余部分(第 5~8 字节):因为每一种类型的 ICMP 数据包具备不同的功能,因此对于不同类型的 ICMP 数据包,这个字段的作用也不同。

(5) 数据部分(后续字节):对于 ICMP 来说,ICMP 头全长 8 字节,之后的内容为 ICMP 数据。ICMP 在设计时实现了许多功能,例如测试连通性、报告错误消息、对数据

包进行重定向等，这些不同的功能一般来说应该使用不同的数据包实现，而ICMP却要使用同一种数据包实现上述功能，因此对于不同类型的数据包，灵活的“报头的其余部分”是对ICMP实现不同功能的最大限度支持，每一种ICMP数据包的“报头的其余部分”均有不同功能。后文将为读者展示不同ICMP数据包类型的“报头的其余部分”的区别。

3.3.3 ping程序原理

ping这个名字源于声呐定位操作，是潜水艇专业人员的专用术语，表示回应的声呐脉冲，而在网络中可以利用ping程序来探测某台主机是否在线以及本机与被探测主机之间的网络时延，一般来说，如果不能ping通某台主机，那么就表示不能连接上那台主机，但是随着网络安全意识的增强，为了在内网中隐蔽主机，主机或者防火墙会限制ping包的转发与响应，这时可能出现ping程序显示某台主机不可达的情况，但同时却可以通过Telnet或SSH程序远程登录该主机。这个时候就不能单纯通过ping程序返回的结果来判断主机是否在线，但是如果能够ping通某台主机，就表明该主机一定在线。目前绝大多数操作系统都自带了ping程序，主要的功能是用来检测网络的连通情况和分析网络速度。

ping程序的工作原理是：向网络上的另一台主机发送ICMP回显请求包，如果目标系统收到ICMP回显请求包，它将返回ICMP回显应答包，并且其返回的ICMP数据包数据字段内容与请求包数据字段内容一致。ping程序可以利用回显请求包和应答包的时间差计算两台主机的往返时间。

在Windows中，默认发送4个ping包，每秒一个，然后输出一些统计信息之后退出。而在Linux中，ping程序默认会不停发送ping包，直到用户按Ctrl+C组合键中断为止。

使用ping程序测试网络连通性的步骤如下：

(1) 使用ipconfig(Linux下使用ifconfig)1观察本地网络设置是否正确。

(2) ping 127.0.0.1，ping回送地址，检查本地的TCP/IP有没有设置好。

(3) ping本机IP地址，检查本机的IP地址是否设置有误。

(4) ping本地网关地址，检查硬件设备是否有问题，也可以检查本机与本地网络连接是否正常(在非局域网中这一步可以忽略)。

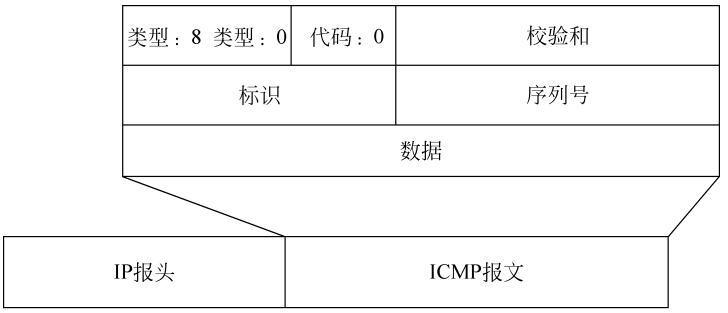
(5) ping公网IP地址，检查本机与外部网络的连接是否正常。

3.3.4 ping包——ICMP回显请求、回显应答数据包

ping程序所使用的ICMP数据包有ICMP回显请求数据包(用于本端ping请求)和ICMP回显应答数据包(用于对端ping应答)两类。这两类数据包的类型分别为8(ping请求)和0(ping应答)，对于类型为8或0的ICMP数据包，代码字段无实际意义，永远填充为0。

为了让ICMP数据包能够满足ping程序的使用要求，“报头的其余部分”被设计成为ICMP标识、ICMP序列号这两个字段，ICMP标识字段的作用是区分不同的进程发起的ping请求，例如在Linux系统中开启两个窗口，分别运行ping，那么每个窗口将使用一个

固定的 ping 标识,这可以区分不同的窗口发起的 ping 请求。而在一个窗口内,可以每秒发起一个 ping 请求,ICMP 序列号字段就是用于区分本窗口内发起的不同 ping 请求,每个请求使用一个序列号。两个字段的作用如图 3.7 所示。



类型8、代码0表示回显请求,类型0、代码0表示回显应答。

图 3.7 ICMP-ping 数据包格式

ICMP 回显请求/应答包的字段所表示的功能说明如下。

(1) 标识:2 字节的标识字段,不同的操作系统会设置为不同的值,比如 UNIX 系统在实现 ping 程序时,把 ICMP 数据包中的标识字段设置为发送进程的 ID 号,这样即使在同一台主机上同时运行多个 ping 程序,也可以识别出返回的信息。而 Windows 系统中 ping 程序将其标识字段始终置为 1。

(2) 序列号:2 字节的序列号,用于识别每一对回显请求与应答,每发送一个新的回显请求该数字就加 1。

(3) 数据:是回显的信息部分,不同操作系统所携带的值不相同,通常可以通过不同操作系统中 ping 程序的默认回显数据来判断发送 ping 请求数据包的操作系统的类型。

在 Windows 中,ICMP 数据包的回显数据长度默认为 32 字节,其内容为英文小写字母循环(abcdefg…w),截图如图 3.8 所示。

在 Linux 中,ICMP 包的回显数据长度默认为 56 字节,其内容为时间戳+回显数据(0x10…0x37),在 Cisco 路由器、交换机设备中,ICMP 包的默认内容模式是 0xabcd,截图如图 3.9 所示。

3.3.5 数据包超时该如何通告——ICMP 超时包

网络中传输的 IP 数据包可能由于超时而失效,此时确认数据包发生超时的中间设备将会对数据包的始发点发送 ICMP 报文进行通知。目前数据包的超时原因有以下两种:

(1) 收到的数据包 TTL 为 1,不能再继续进行转发,此时触发 TTL 超时。

(2) 收到的 IP 数据包分片不全,缺失中间某个分片,导致分片数据包重组无法进行,此时触发重组超时。

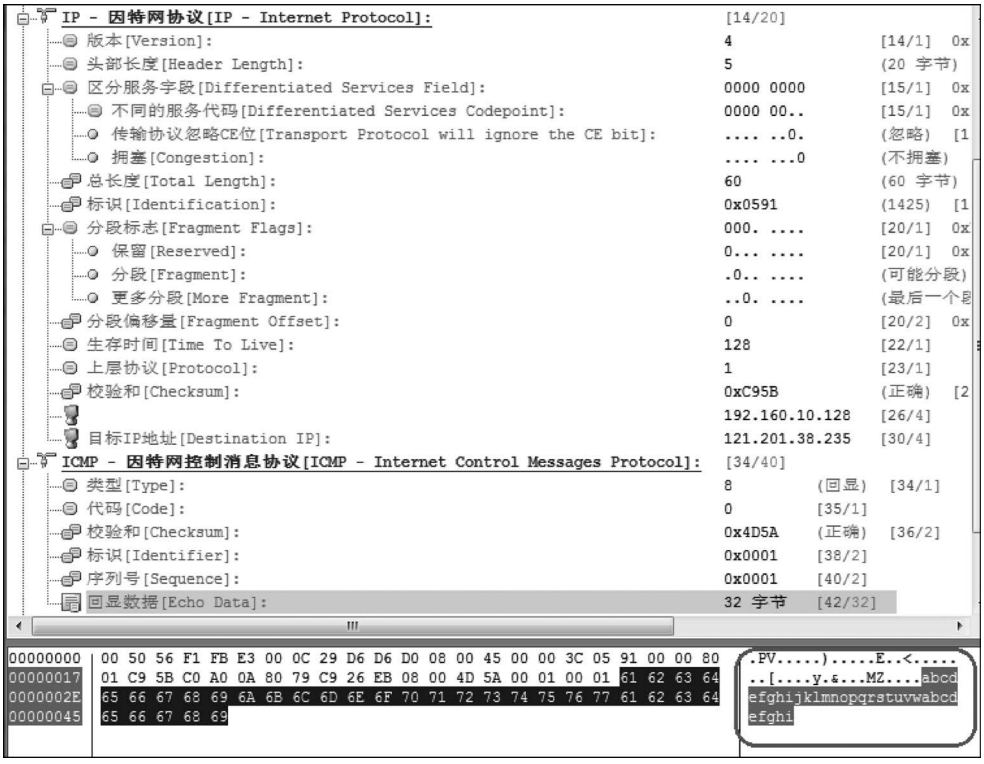


图 3.8 Windows 系统发送的 ping 数据包解码

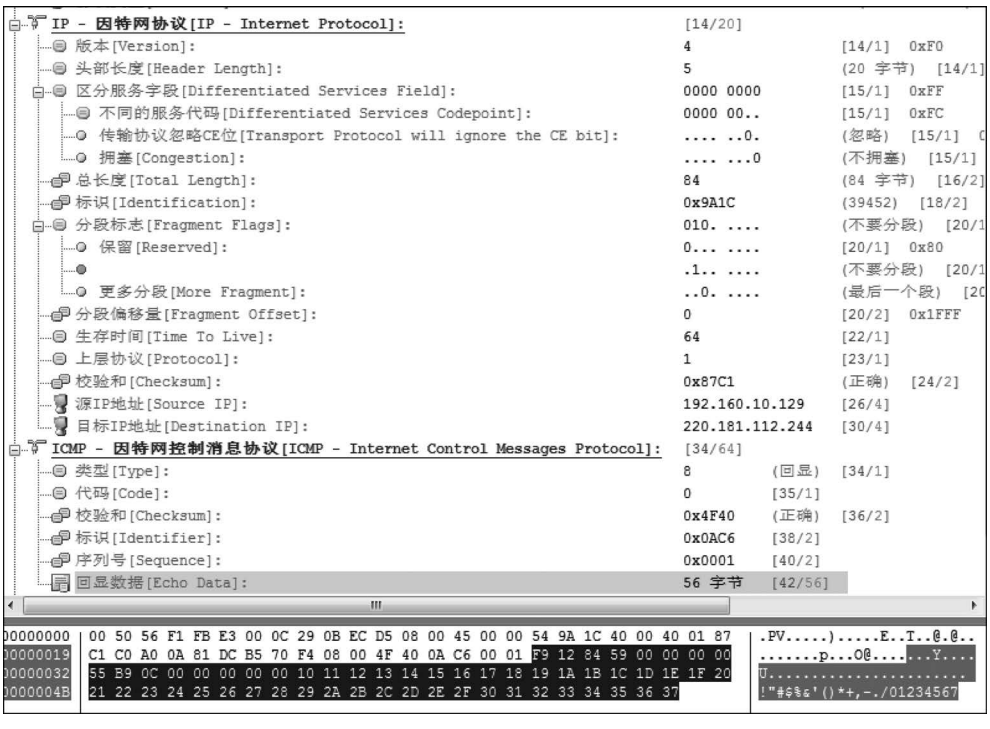


图 3.9 Linux 系统发送的 ping 数据包解码

ICMP 超时包所使用的类型、代码和意义如表 3.3 所示。

表 3.3 ICMP 超时包的类型、代码和意义

类 型	代 码	代 码 意 义
11	0	传输期间 TTL 为 0
11	1	分片重组超时

告知始发者 TTL 超时或重组超时的工作由类型为 11、代码为 0 的 ICMP 数据包承担,一旦路由器将数据包的 TTL 递减为 0,就丢弃这个数据包,并向原始数据包的始发地址发送 ICMP 超时包进行通知,如图 3.10 所示。

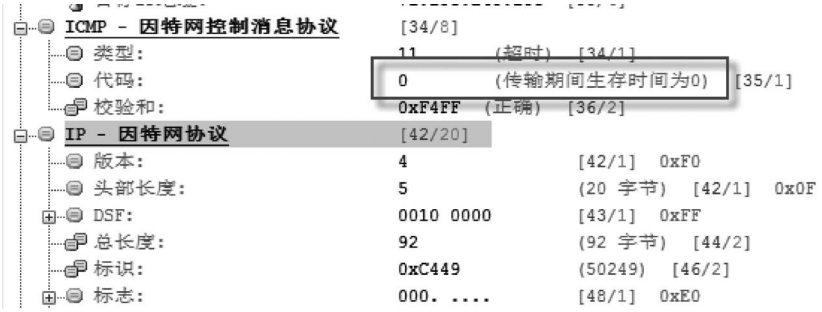


图 3.10 ICMP TTL 超时包解码

类型 11、代码 1 表示数据包目的地址设备,在规定的时间内没有收到所有的分片,此时它将丢弃已收到的相关分片,并向源地址发送超时包,如图 3.11 所示。

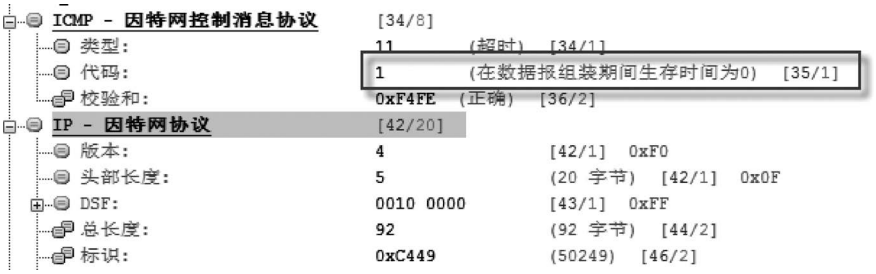


图 3.11 ICMP 重组超时包解码

3.3.6 数据包无法到达目的地该如何通告——ICMP 不可达包

当 IP 数据包由于各种原因,网络设备无法将数据包正确转发到对端,或数据包已经转发到对端主机,但无法到达对端的对应协议、对应端口时,需要对原始数据包的源点进行通告,此时使用 ICMP 目的不可达包来进行通告。

数据包不能正确到达的原因有很多,如协议未开启、端口未开启、网络不可达等。为了让 ICMP 能够在通告时把数据包不可达的具体原因返给发起端,类型 3 的 ICMP 数据包使用不同的代码来表示不同的 ICMP 不可达的几种常见情况,如表 3.4 所示。

表 3.4 ICMP 不可达包的类型、代码和意义

类 型	代 码	代码意义
3	0	目的网络不可达(路由器找不到目标网络)
3	1	目的主机不可达(到达网络却找不到目标主机)
3	2	目的协议不可达(IP 上层协议未运行)
3	3	目的端口不可达(相关服务端口未开放)
3	4	需要分片才能通过,但设置了不分片 DF 位
3	5	基于给出路径的 ping 失败
3	6	目的网络未知
3	7	目的主机未知
3	8	源被隔离
3	9	与目的网络之间的通信被禁止
3	10	与目的主机之间的通信被禁止
3	11	目的网络拒绝请求的 QoS 级别
3	12	目的主机拒绝请求的 QoS 级别
3	13	由于过滤,通信被强制禁止
3	14	主机越权
3	15	优先权中止生效

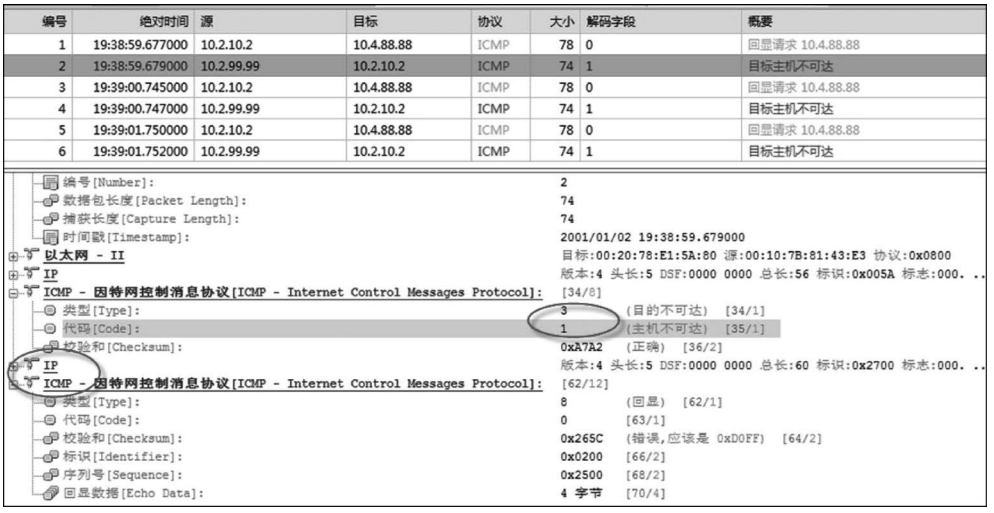
在表 3.3 中,错误代码主要分为两大类:其中一类是代码 2 或 3 的终点不可达包,这类不可达包只能由终点主机创建,这是因为数据包先传递到终点主机,然后由目的主机再处理其协议和端口;另一类除了代码 2 和 3 之外的其余代码表示的错误只可能出现在中间路由器上。

由于 IP 制定得较早,大部分类型 3 的不可达包已经很难在现有网络中看到了,因此本节只介绍一种典型的终点产生的 ICMP 不可达包和一个典型的中间路由器产生的 ICMP 不可达包。

代码 1:主机不可达,为中间路由器产生的 ICMP 不可达包,IP 数据包将经过多个路由器的转发,当数据包被转发到最后一跳路由器时(该路由器应是终点主机的网关),如果终点主机未开机、网络中断,则最后一跳路由器无法正常将数据包转发至终点主机,于是返回类型为 3、代码为 2 的 ICMP 主机不可达数据包,如图 3.12 所示。序号为 1 的数据包是 10.2.10.2 给 10.4.88.88 发送的 ping 请求数据包,但该包未能到达终点主机 10.4.88.88,则最后一跳路由器 10.2.99.99 给始发主机 10.2.10.2 返回 ICMP 主机不可达数据包。

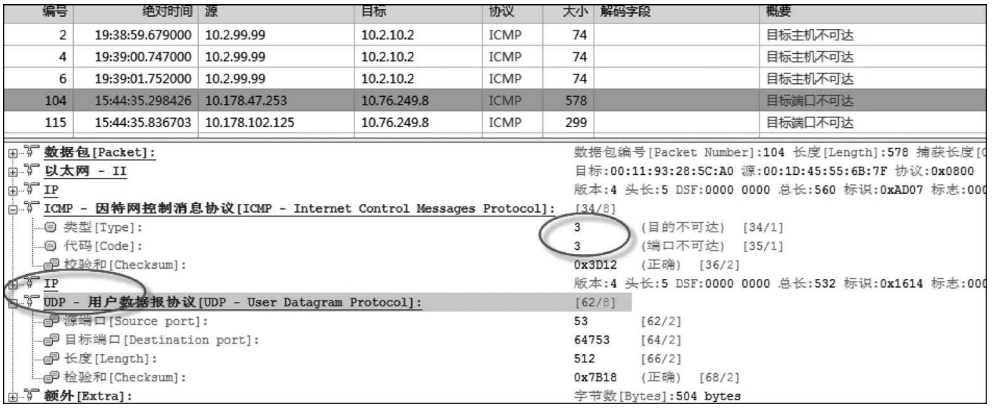
【注意】 最后一跳路由器可能有多个接口配置了不同 IP 地址,这里返回数据包使用的是离 10.2.10.2 最近的 10.2.99.99 这个 IP 地址,作为 ICMP 不可达包的源 IP 地址。

代码 3:端口不可达,为终点主机产生的 ICMP 不可达包,数据包要交付的目标应用程序此时没有运行。该包常与 UDP 包成对出现,这是由于主机访问了对方不存在的 UDP 端口,为了通知这次不存在的端口没有访问成功。此时对方主机将返回“ICMP 端口不可达”包。如图 3.13 所示,编号为 104 的数据包是 10.178.47.253 给 10.76.249.8 发送的 ICMP 端口不可达包,说明之前从 10.76.249.8 访问 10.178.47.253 的某个包没有成功,具体是哪个包访问失败了,可以通过 ICMP 不可达包的数据部分观察解码发现。



编号	绝对时间	源	目标	协议	大小	解码字段	概要
1	19:38:59.677000	10.2.10.2	10.4.88.88	ICMP	78	0	回显请求 10.4.88.88
2	19:38:59.679000	10.2.99.99	10.2.10.2	ICMP	74	1	目标主机不可达
3	19:39:00.745000	10.2.10.2	10.4.88.88	ICMP	78	0	回显请求 10.4.88.88
4	19:39:00.747000	10.2.99.99	10.2.10.2	ICMP	74	1	目标主机不可达
5	19:39:01.750000	10.2.10.2	10.4.88.88	ICMP	78	0	回显请求 10.4.88.88
6	19:39:01.752000	10.2.99.99	10.2.10.2	ICMP	74	1	目标主机不可达

图 3.12 ICMP 主机不可达数据包解码



编号	绝对时间	源	目标	协议	大小	解码字段	概要
2	19:38:59.679000	10.2.99.99	10.2.10.2	ICMP	74		目标主机不可达
4	19:39:00.747000	10.2.99.99	10.2.10.2	ICMP	74		目标主机不可达
6	19:39:01.752000	10.2.99.99	10.2.10.2	ICMP	74		目标主机不可达
104	15:44:35.298426	10.178.47.253	10.76.249.8	ICMP	578		目标端口不可达
115	15:44:35.836703	10.178.102.125	10.76.249.8	ICMP	299		目标端口不可达

图 3.13 ICMP 端口不可达数据包解码

3.4 案例 3-1:ping 大包丢包的原因分析

这是一个早年的案例,但在现在来看仍不过时。当时在一个偏远的矿区,从矿区访问集团网络有丢包的情况,且小包不丢,只丢大包,说是网络性能问题,接下来一起来看这个问题是如何通过流量分析技术发现、证明、解决的。

故障环境说明如下:

(1) 办公机器都属于 10.12.128.0/24 网段。

(2) 办公机器通过一个二层的接入交换机、光电转换器接入集团核心交换机。具体网络拓扑图如图 3.14 所示。

故障现象为从测试机 ping 服务器,大包丢包严重,但小包正常,且之前没有出现过类似丢包现象,网络环境、设备也未发生过变更。因此,发生故障的原因可能并不是某些设

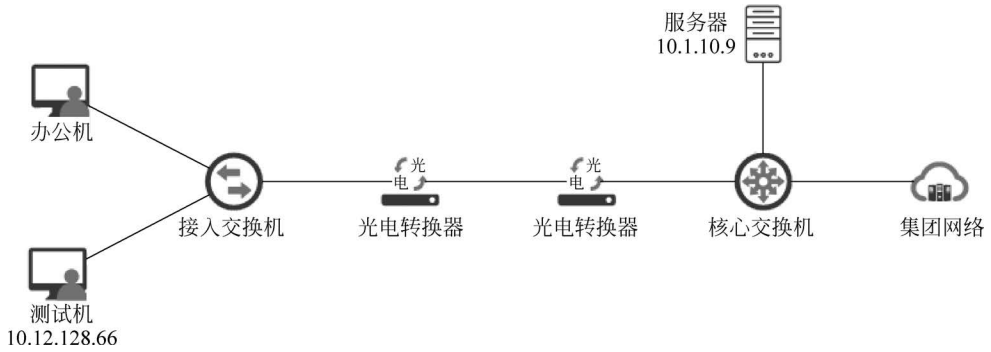


图 3.14 故障环境拓扑图

备的操作或变更而引发的，需要结合实际情况，根据了解掌握的情况来讨论针对本次故障的分析思路和方法。

首先可以判断流量经过的路径，从而判断可能存在的丢包点；然后通过前后几个采集点同时抓取流量的方法，对抓取到的流量进行比对，从而确定产生丢包的点。

对于本次故障来说，可能存在的故障点如图 3.15 所示。

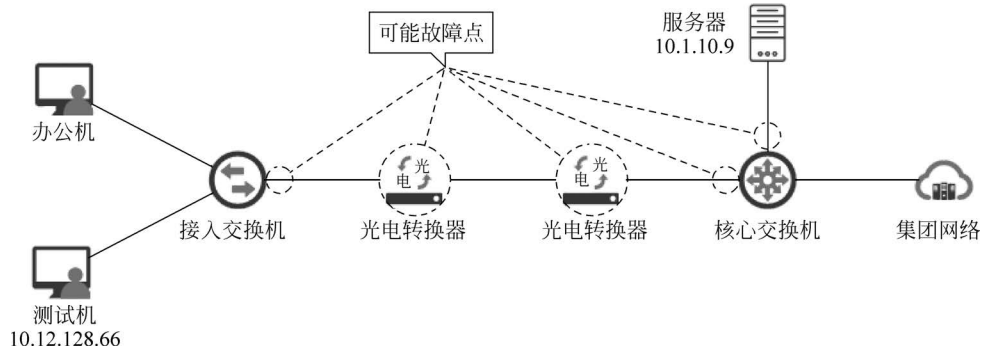


图 3.15 可能存在的故障点

在实际的分析过程中，需要考虑到抓包的方便性和相应中间设备的功能特性选取数据包捕获点。在这个故障环境下，主要选择在接入交换机与核心交换机上抓取数据包，网络信息流量抓取点如图 3.16 所示。

在对两个选定的抓包点部署好抓包后，开始重现故障，在测试机器 10.12.128.66 上使用如下命令测试网络的大包传输情况：ping 10.1.10.9 -l 10000 -t。

当输入上述 ping 命令后，将产生 10 000 字节的 ping 数据和 8 字节的 ICMP 报头，总计 10 008 字节，这个字节数量很显然超过了默认的以太网 MTU 1500 字节，因此这个数据包必然会被分片，结合前文的分片知识点不难计算，如果以 1500 字节的 MTU 发送 10 008 字节数据，每个分片都需要携带一个 20 字节的 IP 报头，则每次发送的真实数据量是 1500－20＝1480，以 1480 每个包的量发送数据，10 008 除以 1480 的结果为 6 余 1128，字节将会被分成 7 个分片进行发送，其中 6 个 1500 字节的分片包，1 个 1148 字节的分片包。通过上述测试命令重现了故障现象：大文件传输丢包情况较为严重。

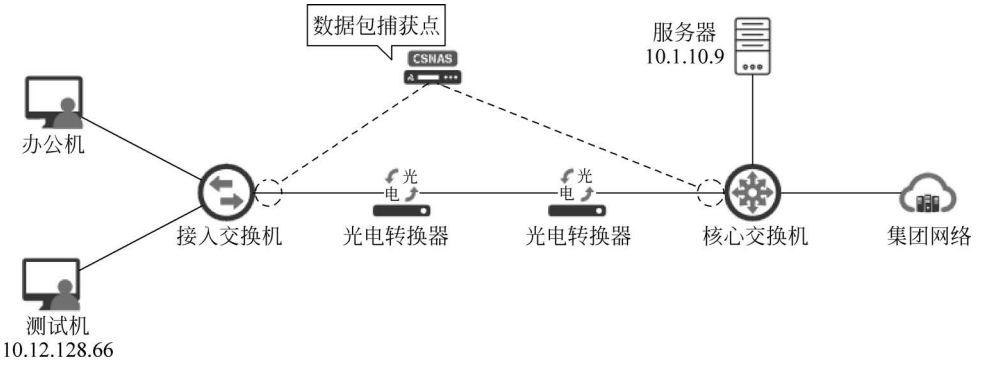


图 3.16　流量捕获点设置

图 3.17 展示了在接入交换机上抓取到的数据包,通过分析观察可以看到软件中显示的大小是 1518 字节和 1166 字节,这是二层数据帧的总长度。若要以二层长度计算得出三层包长,则需要去掉二层帧头、帧尾的 18 字节。计算后,结果为 6 个 1500 字节的包和 1 个 1148 字节的包,将预先计算得到的结果与在接入交换机上捕获到的数据包进行比对,结果一致。因此可以说明:数据从测试机器始发,传输到接入交换机时,没有发生丢包。

10.12.128.66	10.1.10.9	ICMP	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,166

图 3.17　在接入交换机上捕获到的数据包

由于接入交换机的抓包点设置在了去往核心交换的位置,因此这个抓包点实际上是交换机发出的数据包,可以证明,接入交换机发出的分片包没有任何问题。但在发送了 7 个分片到对端后,收到的 ICMP 重组超时包信息捕获如图 3.18 所示。

图 3.18　在接入交换机上捕获到的 ICMP 重组超时包

类型为 11、代码为 1 的 ICMP 数据包意味着数据包的某个分片丢失,导致整个数据包不能进行重组,从而报错。因此,可以在这里得到接入交换机分析的小结:中间某个大包在传输的过程中被丢弃了,导致接收端在重组阶段超时,而接入交换机出口抓到了所有的分片包,即丢失的某个分片包不是在接入交换机上丢弃的。

结合前文所述的对比分析法，继续分析核心交换机6509上抓取的数据包，在核心交换机上抓取到的数据包如图3.19所示。

源	目标	协议	大小
10.12.128.66	10.1.10.9	ICMP	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,518
10.12.128.66	10.1.10.9	IP Fragment	1,166
10.1.10.9	10.12.12...	ICMP	74

图3.19 在核心交换机上抓取到的数据包

可以看到这里少了一个1518字节的包，说明7个分片里丢了一个，由于这个抓包点是进入核心交换机的点，因此可以得到结论：丢弃的某个分片在到达核心交换机6509前就被丢弃。

结合拓扑结构进行对比分析，发现某个分片包是在接入交换机转发之后、核心交换机6509接收之前被丢弃的，那么可能被丢弃的位置只剩下光电转换器了。

使用替换法将接入交换机端的光电转换器更换为一个全新的光电转换器，测试一切正常，故障解决。

当时怀疑是光电转换器的问题，但是没有证据，只能看到现象：ping小包不掉包，光电转换器厂家也解释不了为什么。直到通过网络流量分析技术对故障的现象、数据包表现进行了判断，结合实际情况将产生错误的“证据”摆在面前，才令光电转换器厂家在事实面前无从辩解。

3.5 案例3-2：如何发现大型网络中的环路问题

当环路发生时，会出现网络及应用访问缓慢、网络丢包，甚至无法正常提供服务的问题。通常大型的网络中定位和发现网络环路是比较困难的，本案例将介绍如何通过网络分析技术发现网络环路。

1. 问题描述

某公司网络全部为内部网络，不与互联网连接，出口防火墙连接集团内网，下连核心交换机，核心交换机下连下属单位防火墙，如图3.20所示。

前一段时间上午8～10点网络及应用访问缓慢，内网用户ping DMZ区服务器时会产生大量丢包，甚至无法正常提供服务，而且会不定时地出现网络访问慢的问题，严重地影响了正常的工作。经过一段时间的排查，并没有发现网络及应用产生故障的原因。

这时通过网络中部署的科来网络回溯分析系统对之前发生的问题进行了长时间的回溯分析，定位故障发生的时段来重现故障当时的情景，以便帮助运维人员找到产生问题的根本原因，从而解决问题。故障时段的流量趋势图如图3.21所示，该时段的流量统计信息如图3.22所示。

图3.21和图3.22为发生异常的3h的流量趋势与概要视图，对网络总流量及进出流量做出统计，峰值达到了682.35Mbps，带宽利用率为70%左右，瞬时的利用率甚至更高。

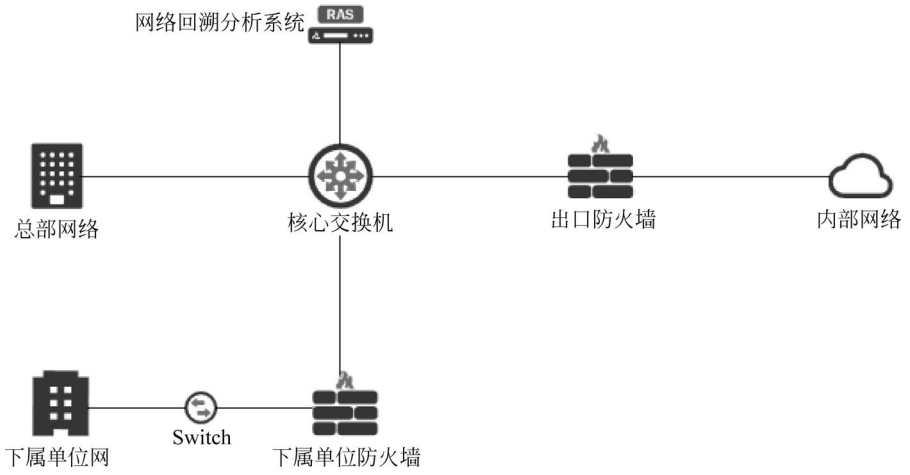


图 3.20 故障网络拓扑

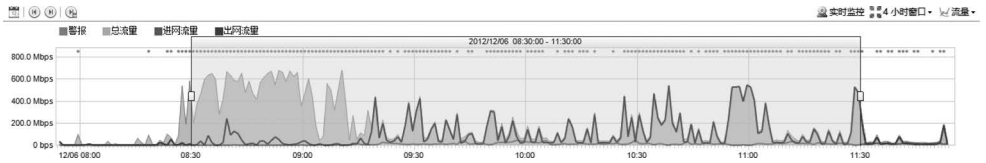


图 3.21 故障时段的网络流量趋势

总流量	峰值	谷值	bps	合计字节	峰值	谷值	bps	合计字节
总流量	682.35 Mbps	3.68 Mbps	193.52 Mbps	324.41 GB	682.35 Mbps	21.08 Mbps	239.62 Mbps	312.99 GB
进出网流量	峰值	谷值	bps	合计字节	峰值	谷值	bps	合计字节
进网流量	48.93 Mbps	10.76 kbps	10.25 Mbps	17.18 GB	48.93 Mbps	10.76 kbps	11.42 Mbps	14.92 GB
出网流量	551.02 Mbps	173.82 kbps	86.33 Mbps	144.73 GB	551.02 Mbps	481.66 kbps	107.11 Mbps	139.90 GB
网内流量	3.85 Mbps	1.24 kbps	1.14 Mbps	1.91 GB	3.85 Mbps	1.24 kbps	1.16 Mbps	1.51 GB
网外流量	680.70 Mbps	89.37 kbps	95.77 Mbps	160.55 GB	680.70 Mbps	89.37 kbps	119.91 Mbps	156.62 GB

图 3.22 故障时段的网络流量统计信息

这就可能会造成大量的数据包丢失。

2. 分析过程

经过对网络应用的分析,发现这 3h 的数据中,未知的 UDP 应用流量占用了总流量的 99%以上,详细信息如图 3.23 所示。

通过对未知 UDP 应用的深入挖掘分析,可以发现大量 UDP 2425 端口的单方向通信,详细信息如图 3.24 所示。

所以基本可以确定网络中产生大数据量传输导致网络慢的原因就是内网中这些使用 UDP 2425 端口进行通信的数据占用了网络的大量带宽,导致网络中产生了很多丢包,造成访问应用系统慢。

经过查阅资料和 UDP 会话分析发现,飞秋(FeiQ)软件使用的是 UDP 2425 端口,飞秋是一款局域网聊天传送文件的绿色软件,它参考了飞鸽传书(IPMSG)和 QQ,完全兼容飞鸽传书协议。

再查找占用带宽较大的 IP 地址,发现基本所有大流量传输的 IP 地址均为“该公司下

概要统计 | 网络应用 | IP地址 | 网段统计 | 物理地址 | IP会话 | 物理会话 | TCP会话 | UDP会话 | 警报

网络应用[68/68]

名称	字节数	数据包	每秒字节	每秒数据包
未知UDP应用	265.52 GB	407,543,172	26.40 Mbps	37,735
未知TCP应用	18.41 GB	25,475,882	1.83 Mbps	2,358
HTTP	7.58 GB	10,449,142	753.99 kbps	967
POP3	2.13 GB	2,632,913	212.23 kbps	243
SSH	2.03 GB	2,609,994	201.81 kbps	241
RSH	1.42 GB	3,742,467	141.26 kbps	346
MSRDP	1.20 GB	2,187,579	119.19 kbps	202
SMTP	979.06 MB	1,134,744	95.06 kbps	105
HTTP Proxy	774.65 MB	789,444	75.21 kbps	73
DNS	482.57 MB	6,093,299	46.85 kbps	564
BitTorrent	168.87 MB	662,497	16.40 kbps	61
Oracle	83.22 MB	314,794	8.08 kbps	29
SNMP	79.40 MB	570,976	7.71 kbps	52
Kerberos	76.57 MB	497,998	7.43 kbps	46

图 3.23　故障时段的网络应用流量统计

概要统计 | 网络应用 | IP地址 | 网段统计 | 物理地址 | IP会话 | 物理会话 | TCP会话 | UDP会话 | 警报

网络应用[68/68] ▸ UDP会话[13972/13972]

网段统计 内网IP 外网IP IP会话 TCP会话 UDP会话　搜索UD

名称	地址1 ->	端口1 ->	<- 地址2	<- 端口2	字节...	数据包 ->	<- 数据包
未知UDP应用	[redacted].82	2425	[redacted].134	2425	1.64 MB	2,245	0
未知TCP应用	[redacted].20	2425	[redacted].156	2425	1.63 MB	2,225	0
HTTP	[redacted].20	2425	[redacted].35	2425	1.55 MB	2,111	0
POP3	[redacted].82	2425	[redacted].114	2425	1.54 MB	2,115	0
SSH	[redacted].20	2425	[redacted].231	2425	1.53 MB	2,095	0
RSH	[redacted].20	2425	[redacted].84	2425	1.49 MB	2,035	0
MSRDP	[redacted].82	2425	[redacted].192	2425	1.48 MB	2,037	0
SMTP	[redacted].20	2425	[redacted].43	2425	1.48 MB	2,025	0

图 3.24　故障时段的 UDP 会话统计信息

属单位”网段的 IP 地址。

3. 网络环路分析

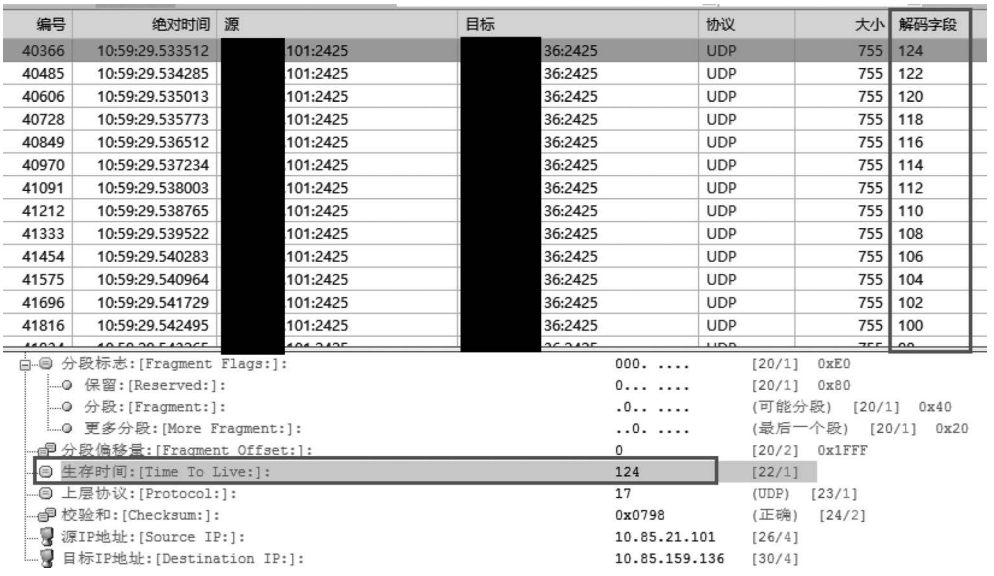
通过下载数据包进行精细分析，可以对其中的两台主机传输的数据包进行解码分析，发现数据中存在大量 IP 端口相同且具有相同的 IP 标识的数据包，这就证明了两台主机之间传输的数据包为同一个数据包，详细信息如图 3.25 所示。

编号	绝对时间	源	目标	协议	大小	解码字段
40366	10:59:29.533512	[redacted]101:2425	[redacted]136:2425	UDP	755	0x6ADD
40485	10:59:29.534285	[redacted]101:2425	[redacted]136:2425	UDP	755	0x6ADD
40606	10:59:29.535013	[redacted]101:2425	[redacted]136:2425	UDP	755	0x6ADD
41575	10:59:29.540964	[redacted]101:2425	[redacted]136:2425	UDP	755	0x6ADD
41696	10:59:29.541729	[redacted]101:2425	[redacted]136:2425	UDP	755	0x6ADD
41816	10:59:29.542495	[redacted]101:2425	[redacted]136:2425	UDP	755	0x6ADD

区分服务字段:[Differentiated Services Field:]:	0000 0000	[15/1]	0xFF
不同的服务代码:[Differentiated Services Codepoint:]:	0000 00..	[15/1]	0xFC
传输协议忽略CE位:[Transport Protocol will ignore the CE bit:]:	.... ..0.	(忽略) [15/1]	0x02
拥塞:[Congestion:]:	0	(不拥塞) [15/1]	0x01
总长度:[Total Length:]:	737	(737 字节) [16/2]	
标识:[Identification:]:	0x6ADD	(27357) [18/2]	
分段标志:[Fragment Flags:]:	000.	[20/1]	0xE0
保留:[Reserved:]:	0...	[20/1]	0x80
分段:[Fragment:]:	.0..	(可能分段) [20/1]	0x40
更多分段:[More Fragment:]:	..0.	(最后一个段) [20/1]	0x20
分段偏移量:[Fragment Offset:]:	0	[20/2]	0x1FFF

图 3.25　大量具有相同 IP 标识的重复数据包

再来定位数据包中的 TTL 字段,发现数据包的 TTL 值呈现逐步递减的趋势,每个数据包 TTL 值减 2,详细信息如图 3.26 所示。这就说明了这个数据包在传输的过程中经过了两个三层设备的处理后又回到了核心交换机与防火墙上连接的接口,被再次捕获。



编号	绝对时间	源	目标	协议	大小	解码字段
40366	10:59:29.533512	[redacted]101:2425	[redacted]36:2425	UDP	755	124
40485	10:59:29.534285	[redacted]101:2425	[redacted]36:2425	UDP	755	122
40606	10:59:29.535013	[redacted]101:2425	[redacted]36:2425	UDP	755	120
40728	10:59:29.535773	[redacted]101:2425	[redacted]36:2425	UDP	755	118
40849	10:59:29.536512	[redacted]101:2425	[redacted]36:2425	UDP	755	116
40970	10:59:29.537234	[redacted]101:2425	[redacted]36:2425	UDP	755	114
41091	10:59:29.538003	[redacted]101:2425	[redacted]36:2425	UDP	755	112
41212	10:59:29.538765	[redacted]101:2425	[redacted]36:2425	UDP	755	110
41333	10:59:29.539522	[redacted]101:2425	[redacted]36:2425	UDP	755	108
41454	10:59:29.540283	[redacted]101:2425	[redacted]36:2425	UDP	755	106
41575	10:59:29.540964	[redacted]101:2425	[redacted]36:2425	UDP	755	104
41696	10:59:29.541729	[redacted]101:2425	[redacted]36:2425	UDP	755	102
41816	10:59:29.542495	[redacted]101:2425	[redacted]36:2425	UDP	755	100

分段标志:[Fragment Flags:]: 000. .... [20/1] 0xE0
保留:[Reserved:]: 0... .... [20/1] 0x80
分段:[Fragment:]: .0.. .... (可能分段) [20/1] 0x40
更多分段:[More Fragment:]: ..0. .... (最后一个段) [20/1] 0x20
分段偏移量:[Fragment Offset:]: 0 [20/2] 0x1FFF
生存时间:[Time To Live:]: 124 [22/1]
上层协议:[Protocol:]: 17 (UDP) [23/1]
校验和:[Checksum:]: 0x0798 (正确) [24/2]
源IP地址:[Source IP:]: 10.85.21.101 [26/4]
目标IP地址:[Destination IP:]: 10.85.159.136 [30/4]

图 3.26 重复数据包的 TTL 呈递减重复出现

经过确认,在防火墙上发现一条为 192.168.0.0/16 指向核心交换机的路由。这就造成了“下属公司”网段中发往 192.168.0.0/16 网段的数据包,由于在核心交换机没有精确匹配的路由,因此通过核心交换机的默认路由指向防火墙,而经过防火墙后被防火墙的 192.168.0.0/16 路由指回核心交换机,这样就形成了路由环路。

4. 分析结论

通过对内网的整体流量分析,发现大量未知 UDP 2425 的流量,占用总带宽的 99%,导致其他网络访问缓慢。经过“下载分析”发现,这是由于路由环路导致的。

其中“下属公司”的网段到总部的一些网段之间的路由配置存在问题,产生路由环路,造成核心交换机和防火墙之间传输了大量数据,阻塞链路带宽,进而造成网络传输效率降低,产生网络问题。

5. 紧急处理办法及优化建议

通过联系“下属公司”的网络管理员,禁止了“下属公司”的防火墙到核心交换机的 UDP 2425 的流量,之后网络流量恢复正常,故障现象基本消失,网络恢复正常。

针对本次流量异常情况,建议修改防火墙上的路由配置,精细路由条目,进行整理规划,或禁止 UDP 2425 的流量。

类似的路由环路可以通过“黑洞路由”的方式避免,在上级路由器使用汇总路由,而下级路由器配置默认路由,同时汇总的网段中有部分子网未使用的情况下,最好在下级设备中额外配置一条静态路由,将汇总的大网段指向 null 0 接口。例如,上级设备(防火墙)配置 192.168.0.0/16 指向下级核心交换机,下级核心交换机则配置 192.168.0.0/16 指向

null 0 接口(针对 Cisco 路由器)。由于路由转发遵循精确匹配原则,这样不会影响下级路由器已配置的子网访问,只是将目标地址为未配置的子网主机的数据包丢弃,避免环路发生。

6. 价值

通过网络分析技术能够通过 IP TTL 及 IP ID 的变化快速发现并确定网络环路的大小,帮助用户精细配置路由条目,避免不必要的流量占用大量带宽。

3.6 下一代网络世界 IPv6 到底难在哪里

IPv6(Internet Protocol Version 6,互联网协议第 6 版)是互联网工程任务组(The Internet Engineering Task Force,IETF)设计的用于替代 IPv4 的下一代 IP。

3.6.1 IPv6 的发展历程

在过去几十年中,基于 IP 的互联网发展非常迅速,事实上 IP 也是一个非常成功的协议,它经受住了全球上亿台计算机互联的考验。但是其对网络规模的预估受限于当年的设计格局,随着网络的急速发展,尤其是云计算、云存储、虚拟化、SDN 的不断推出,IP 越来越显示出它的局限性。

IP 地址空间的紧缺直接限制了 IP 技术应用的进一步发展,到 1996 年已将 80%的 A 类地址、50%的 B 类地址、10%的 C 类地址全部分配出去了,当时有人预测,到 2010 年 IP 地址将全部用完。

但是到目前为止,绝大部分互联网企业还在使用 IP 地址,这得益于两大技术:可变长子网掩码(Variable Length Subnet Mask,VLSM)技术和地址转换(Network Address Translation,NAT)技术的使用,新技术的使用极大地节省了 IP 地址的使用,延缓了资源被耗尽的时间。

但是新技术只能延缓资源的使用,不能从根本上解决网络对地址资源的需求,尤其是地址转换技术有其自身的固有缺点,只是延长 IP 使用寿命的权宜之计。

同时,新技术的发展,如 5G、穿戴产品、智能家电等都需要一个全球单播地址,还有 IPv4 对新的安全性、服务质量等需求的支持也具有一些局限性。所以迫切需要一种新的技术彻底解决目前 IP 面临的问题。

为了解决互联网发展过程中遇到的问题,早在 20 世纪 90 年代初,互联网工程任务组就开始着手下一代互联网协议 IPng(IP-the next generation)的制定工作。IETF 在 RFC1550 文档中公布了新协议需要实现的主要目标:

(1) 支持无限大的地址空间。

(2) 减小路由表的大小,使路由器能更快地处理数据包。

(3) 提供更好的安全性,实现 IPv4 的安全。

(4) 支持多种服务类型,并支持组播。

(5) 支持自动地址配置,允许主机不更改地址实现异地漫游。

(6) 允许新旧协议共存一段时间。

(7) 协议必须支持可移动主机和网络。

1994 年 7 月,IETF 决定以 SIPP(Simple IP Plus,由 RFC1710 描述)作为 IPng 的基础,同时把地址位数由 64 位增加到 128 位。新的 IP 称为 IPv6,最终技术细节体现在 IETF 的 RFC1752 文档中。

3.6.2 IPv6 的新特点

相对于 IPv4,IPv6 具有以下新特点:

(1) 全新的数据包格式。对比 IPv4,IPv6 报头字段更少,更精简。

(2) 巨大的地址空间。IPv6 地址的位数达到 128 位,相比 IPv4 增长了 4 倍。在 IPv4 中,32 位地址理论上可编址的节点数是 2^{32},也就是 4294967296 个地址。而 IPv6 拥有 2^{128} 个地址。

(3) 全新的地址配置方式。为了简化主机地址配置,IPv6 除了支持手工地址配置和有状态自动地址配置(利用 DHCP 服务器动态分配地址)外,还支持一种无状态地址配置技术。在无状态地址配置中,网络上的主机能自动给自己配置 IPv6 地址。

(4) 更好的服务质量支持。IPv6 在报头中新定义了一个叫作流标签的特殊字段。IPv6 的流标签字段使得网络中的路由器可以对属于一个流的数据包进行识别,并提供特殊处理。利用这个标签,路由器无须打开传送的内层数据包就可以识别流,这样即使数据包有效载荷已经进行了加密,仍然可以实现对服务质量的支持。

(5) 内置的安全性。IPv6 本身就支持 IPSec,包括 AH 和 ESP 等扩展报头,这就为网络安全提供了一种基于标准的解决方案,提高了不同 IPv6 实现方案之间的互操作性。

(6) 全新的邻居发现协议。IPv6 中的邻居发现协议(Neighbor Discovery Protocol,NDP)是一系列机制,用来管理相邻节点的交互。该协议用更多有效的单播和组播包取代了 IP 中的 ARP、ICMP 路由器发现和 ICMP 路由器重定向,并在无状态地址自动配置中起到了不可或缺的作用。该协议是 IPv6 的一个关键协议,也是 IPv6 和 IPv4 的一个显著区别。

(7) 良好的扩展性。因为 IPv6 在标准报头的后面添加了扩展报头,所以 IPv6 可以很方便地实现功能的扩展。IP 报头中的选项长度最大为 40 字节,而 IPv6 扩展报头的长度相比 IP 几乎不受限,只受到 IPv6 数据包的长度限制。

(8) 内置的移动性。由于采用了 Routing Header 和 Destination Option Header 等扩展报头,使得 IPv6 提供了内置的移动性。

3.6.3 IPv6 地址表示方法

IP 地址利用“点分十进制”来表示,例如 172.17.11.11。IPv6 的地址有 128 位字长,因此利用“冒号分十六进制”的方式来表示,但经过十六进制显示的 IPv6 地址仍显冗长,因此可以进一步进行地址压缩。根据在 RFC2373(IPv6 Addressing Architecture)中的定义,IPv6 地址有 3 种表示方法,即首选表示法、压缩表示法和内嵌 IPv4 地址的 IPv6 地址,这里详细讨论前两种表示方法。

1. 首选表示法

IPv6的128位地址是每16位划分为一段，每段被转换为十六进制数，并用冒号隔开。这种表示方法叫冒号十六进制表示法。转换后的IPv6地址如图3.27所示。

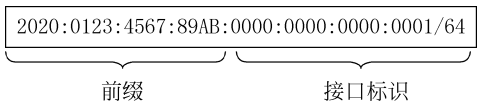


图3.27 IPv6地址首选表示法

在图3.27中，可以看出IPv6地址分为前缀部分和接口标识部分，前缀相当于IP地址中的“网络位”，接口标识相当于IP地址中的“主机位”。区分网络位和主机位所使用的子网掩码在IPv6中称为“前缀长度”，默认标识子网的IPv6地址前缀长度为64。

2. 压缩表示法

当IPv6地址中有很多0，甚至一段地址均以0作为填充时，书写和对比都比较麻烦，所以可以使用压缩表示法把连续出现的0进行压缩表示，也可以把每段中出现的第一个0删除。

RFC2373中规定，当地址中存在一个或多个连续的16位为0的字符时，为了缩短地址长度，用::(两个冒号)表示，但一个IPv6地址中只允许有一个::出现。

所以图3.28的IPv6地址使用压缩表示法表示，可以缩写为如图3.27所示的地址。

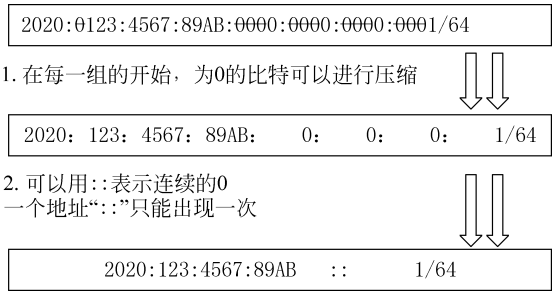


图3.28 IPv6地址压缩表示法

3.6.4 EUI-64算法

对于IP路由器而言，配置一个接口地址的动作为：配置一个地址并指定一个掩码。IPv6路由器地址的配置方法基本类似：配置一个IPv6地址并指定一个前缀长度。这里需要特别注意的是，IPv6不再有掩码的概念。

相对于主机用途的多样性，主机地址希望能够实现自动配置，目前有两种自动配置技术：有状态自动配置和无状态自动配置。这里的无状态自动配置协议是相对有状态自动配置协议DHCP的，本节不再赘述DHCP的配置方法，需要了解相关知识的读者可自行查阅相关文档和书籍。

无状态地址自动配置技术基于对主机使用的IPv6地址的如下结构性假设：一个主机的IPv6地址由64位前缀和64位接口ID组成。要想实现整个IPv6地址的动态配置，

实际上就是分别实现这两部分的动态配置的过程。

一般来讲,主机需要的前缀地址可能是路由器接口的前缀,为了自动获得这个前缀,在路由器和主机之间运行一个无须配置主机的协议即可,参见 3.6.2 节 NDP 的介绍,在此不做过多介绍。

64 位的接口 ID 自动生成则用到了经典的 EUI-64 算法。EUI-64 算法由 1998 年的 RFC2464 定义,是一种基于 48 位的 MAC 地址生成 64 位的接口 ID 的算法,工作原理如下:

(1) 在 IEEE 分配的 ID 和厂商编制的 ID 之间插入 16 位二进制字符 1111111111111110,即十六进制的 FFFE,使得 48 位的 MAC+16 位填充正好能够得到 64 位的接口 ID。

(2) 再将 64 位中的第 7 位反转,形成 IPv6 地址的接口 ID,加上 IPv6 前缀形成完整的 IPv6 地址。

例如,假设 MAC 地址为 00-50-56-C0-00-08,则插入 FFFE 后的结果为 0050:56FF:FEC0:0008,将第 7 位二进制数进行反转后的结果为 0250:56FF:FEC0:0008。如果此时能够有一个 IPv6 地址前缀,则可以和 EUI-64 共同组合成为一个完整的 IPv6 无状态自动生成地址。

但随着时间的推移,人们发现 EUI-64 并不安全,因为该算法可以通过 IPv6 无状态地址进行逆运算,推出主机的 MAC 地址。所以可以使用基于设备随机生成的后 64 位来形成无状态自动地址配置,在 Windows 10 操作系统中,观察自己的 IPv6 无状态自动生成地址,会发现这些地址的生成并不使用 EUI-64 算法,而是采用了随机数值。

3.6.5 IPv6 地址类型

IP 地址分为单播、组播和广播地址,而 IPv6 中分为单播、组播、任播地址,注意 IPv6 地址中没有广播,而是增加了任播的概念。IPv6 地址的类型如图 3.29 所示。下面对 IPv6 的地址类型进行详细介绍。

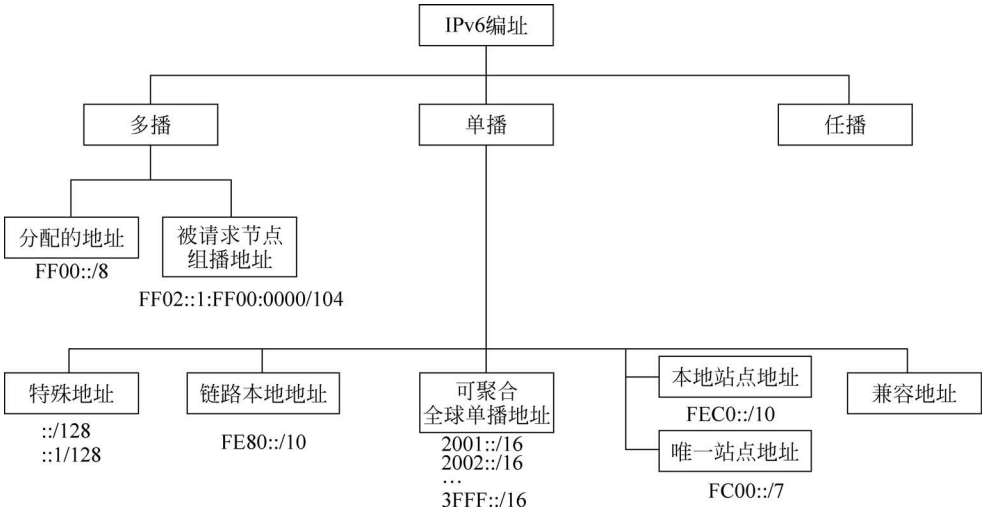


图 3.29 IPv6 地址的分类

1. 特殊地址

环回地址的格式为0:0:0:0:0:0:0:1或::1。全0表示为::/128,仅用于接口没有分配地址时作为源地址。在重复地址检测中使用,含有未指定地址的包不会被转发。环回地址表示为::1/128,等同于IP地址的127.0.0.1。

2. 链路本地地址

前缀为FE80::/10,包括从FE80::/16E到FEBF::/16的所有地址。同时,扩展EUI-64格式或随机数值的接口标识符作为IPv6地址中的后64位。顾名思义,此类地址用于同一链路上的节点间的通信,不能在站点内的子网间路由。

3. 全球单播地址

前缀为2000::/3,包括从2000::/16到3FFF::/16的所有地址。全球单播地址相当于IP的公网地址(IPv6的诞生根本上就是为了解决IP公网地址耗尽的问题)。这种地址在全球的路由器间可以路由。

4. 本地站点地址

前缀为FEC0::/10,包括从FEC0::/16到FEFF::/16的所有地址,以前是用来部署私网的,但RFC3879中已经不建议使用这类地址,建议使用唯一本地地址。

5. 唯一站点地址

前缀为FC00::/7,包括从FC00::/16到FDFF::/16的所有地址。相当于IP的私网地址(10.0.0.0、172.16.0.0、192.168.0.0),在RFC4193中新定义的一种解决私网需求的单播地址类型,对应NAT,用于保护内网隐私,同时用来代替废弃使用的站点本地地址。

6. 兼容地址/过渡地址

为使现有网络能从IPv4平滑过渡到IPv6,需要用到一些IPv6转换机制。通过在IPv6的某些十六进制段内嵌这IPv4的地址,例如IPv6地址中的64:ff9b::10.10.10.10,此IPv6地址最后4字节内嵌一个IPv4地址,这类地址主要用于IPv4/IPv6的过渡技术中。

7. 组播地址

前缀为FF00::/8,包括从FF00::/16到FFFF::/16的所有地址。所谓组播,是指一个源节点发送的单个数据包能被特定的多个目的节点接收到。在IP网络中,组播地址的最高位被设为1110,即从11100000到11101111开头的地址都是组播地址(十进制224～239)。在IPv6网络中,组播地址也有特定的前缀标识,其最高位前8位为1,即以FF开头。图3.30显示了组播地址的结构。

8位	4位	4位	112位
1111 1111	Flags	Scop	group ID

图3.30 IPv6组播地址的结构

标志(Flags)字段有4位,目前只使用了最后1位(前三位必须为0)。当该值为0时,表示当前的组播地址是由IANA所分配的一个永久组播地址(通常给各种协议的组播通信使用);当该值为1时,表示当前的组播地址是一个临时组播地址(非永久分配地址)。

范围(Scop)用来限制组播数据流在网络中发送的范围,该字段占 4 位。其取值说明如下:

- 0:预留。
- 1:节点本地范围。
- 2:链路本地范围。
- 5:站点本地范围。
- 8:组织本地范围。
- E:全球范围。
- F:预留。

其他取值没有定义。

8. 任播地址

任播地址是 IPv6 特有的地址类型,它用一个地址来标识一组网络设备。路由器会将目标地址是任播地址的数据包发送给距离本路由器最近的一个网络接口。接收方只需是一组设备中的某一个即可。与生活中从某购物网站购物的体验一样,对于一件畅销的商品而言,会有来自全国各地的订单,为了加快客户收到货物的速度,如果是北京周边城市收货的订单,则从北京仓发货,如果是上海周边城市收货的订单,则从上海仓就近发货。

IPv6 地址和 IPv4 地址的等效项对比如表 3.5 所示。

表 3.5 IPv6 地址和 IPv4 地址的等效项对比

IPv4 地址	IPv6 地址
互联网地址类别	IPv6 中无此概念
组播地址(224.0.0.0/4)	IPv6 组播地址(FF00::/8)
广播地址	IPv6 中无广播的概念
未指定地址(0.0.0.0)	IPv6 未指定地址(::)
本地回环地址(127.0.0.1)	IPv6 本地回环地址(::1)
公有地址	IPv6 全球单播地址
私有地址(10.0.0.0/8,172.16.0.0/12,192.168.0.0/16)	IPv6 唯一本地地址(FD00::/8)
	站点本地地址(FEC0::/10)
自动专用 IP 地址(169.254.0.0/16)	IPv6 链路本地地址(FE80::/64)
表示方式:点分十进制	表示方式:冒号分十六进制(经过压缩的)
前缀表示方式:点分十进制形式或前缀长度形式表示的子网掩码	前缀表示方式:仅支持前缀长度形式表示

3.6.6 IPv6 数据包格式

接下来介绍 IPv6 数据包格式,具体字段信息如图 3.31 所示。

字段内容如下:

(1) 版本(Version):长 4 位,IP 报头为 0100,IPv6 报头为 0110。

(2) 流量类别(Traffic Class):长 8 位,与 IP 中的 DSCP 字段功能相同,由 RFC2474 与 RFC3168 定义,作用等同于 IP 中的 ToS/DSCP 字段。

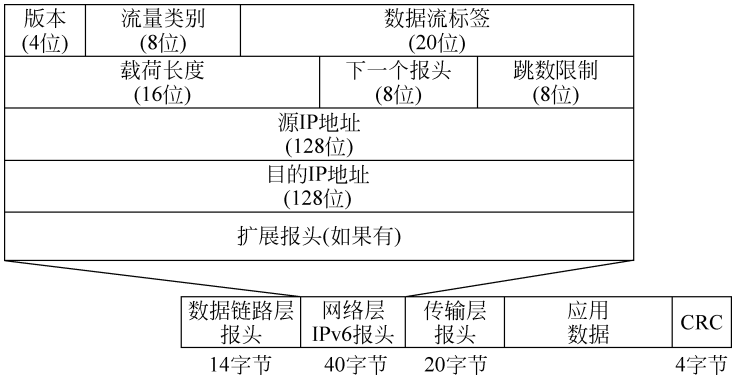


图 3.31 IPv6 数据包格式

(3) 数据流标签(Flow Label)：长 20 位，从网络层将三元组或五元组相同的一组数据标记为同一个数据流，由 RFC6437 定义。

(4) 载荷长度(Payload Length)：标记了 IPv6 报头携带数据的长度。

(5) 下一个报头(Next Header)：当 IPv6 报头不携带扩展报头时，该字段与 IP 中的 Protocol 字段作用相同，当包中携带扩展报头时，该字段用于标明扩展报头的类别。

(6) 跳数限制(Hop Limit)：与 IP 中的 TTL 字段作用相同。

(7) 源 IP 地址(Source Address)：表示发送方的地址，长度为 128 位。

(8) 目的 IP 地址(Destination Address)：表示接收方的地址，长度为 128 位。

IPv6 和 IPv4 对比，有以下区别：

(1) IPv6 报头是定长的(固定为 40 字节)，IPv4 报头是变长的。

(2) IPv6 中 Hop Limit 字段的含义类似于 IPv4 的 TTL。

(3) IPv6 中的 Traffic Class 字段的含义类似于 IPv4 中的 ToS/DSCP。

(4) IPv6 的报头取消了校验和字段。

(5) IPv6 报头相比 IPv4 去掉了如下部分：

① 报头长度：IPv6 报头为固定长度的 40 字节。

② 标识、标志和偏移字段：由于不是每一个数据包都需要分片，因此分片字段移动到了扩展选项中。

③ 选项和填充：选项由扩展报头处理，填充字段也去掉。

3.6.7 IPv6 扩展报头

IPv6 报头中的 Next Header 字段和 IPv4 中的 Protocol 字段作用类似，表示“承载上一层的协议类型”；同时，如果 IPv6 数据包携带了扩展选项，则这个字段表示“扩展选项类型”。

IPv6 扩展报头是跟在基本 IPv6 报头后面的可选报头。因为 IP 报头中包含所有的选项，所以每个中间路由器都必须检查这些选项是否存在，如果存在可选项，就必须处理它们，这种设计方法降低了路由器转发 IP 数据包的效率。为了解决这个问题，在 IPv6 中，相关选项被移到了扩展报头中，IPv6 扩展报头的字段信息如图 3.32 所示。

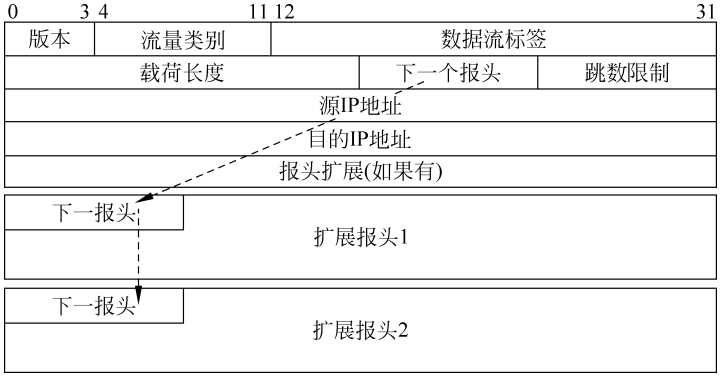


图 3.32 IPv6 扩展报头示意图

IPv6 扩展报头包括以下类型。

(1) 逐跳选项报头：该扩展报头被每一跳处理，可包含多种选项，如路由器告警选项。

(2) 目的选项报头：目的地处理，可包含多种选项，如 Mobile IPv6 的家乡地址选项。

(3) 路由报头：指定源路由，类似于 IP 源路由选项，IPv6 源节点用来指定信息报到达目的地的路径上必须经过的中间节点。IPv6 基本报头的目的地址不是分组的最终目的地址，而是路由扩展报头中所列的第一个地址。

(4) 分段报头：IP 包分片信息，只由目的地处理。

(5) 认证报头：IPSec 用扩展报头，只由目的地处理。

(6) 封装安全净载报头：IPSec 用扩展报头，只由目的地处理。

3.7 适应下一代网络的 ICMP 与 ARP

在 IPv4 中，ICMP 向源地址报告关于向目的地传输 IP 数据包的错误和信息。它为诊断、控制和管理目的定义了一些消息，如目的不可达、数据包超长、超时、回送请求和回送应答等。在 IPv6 中，ICMPv6(Internet Control Message Protocol version 6，互联网控制包协议版本 6)除了提供 ICMPv4 常用的功能外，还定义了其他机制所需的 ICMPv6 消息，例如邻居节点发现、无状态地址配置(包括重复地址检测)、路径 MTU 发现等。

3.7.1 ICMPv6 概述

与 ICMP 一样，ICMPv6 的作用是弥补 IPv6 的缺陷，是 IPv6 的伴侣。与 IPv4 不同的是，ICMPv6 还负责 ARP、IGMP 的功能，ICMPv6 的协议号为 58，由 RFC4443 定义。ICMPv6 与 IPv6 的关系如图 3.33 所示。

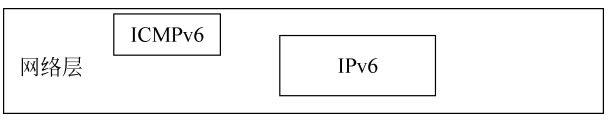


图 3.33 ICMPv6 与 IPv6 的关系

3.7.2 ICMPv6 数据包格式

ICMPv6 数据包分为两类：差错包和信息包。差错包用于报告在转发 IPv6 数据包的过程中出现的错误。信息包主要包括回送请求包(Echo Request)和回送应答包(Echo Reply)。ICMPv6 数据包的消息类型如表 3.6 所示。

表 3.6 ICMPv6 数据包的消息类型

消息类型	类型编号	ICMPv6 报文类型的描述
错误性消息	1	目的不可达
	2	数据包太大
	3	超时
	4	参数问题
	100	私人实验
	101	私人实验
	127	保留，用于扩展 ICMPv6 错误消息
信息性消息	128	回显请求
	129	回显应答
	135	邻居请求报文 NS
	136	通报报文 NA
	200	私人实验
	201	私人实验
	255	保留，用于扩展 ICMPv6 参考消息

ICMPv6 差错包的 8 位类型字段中的最高位都是 0，ICMPv6 信息包的 8 位类型字段中的最高位都是 1。因此，对于 ICMPv6 差错包的类型字段，其有效值范围为 0～127，而信息包的类型字段有效范围为 128～255。ICMPv6 数据包格式如图 3.34 所示。

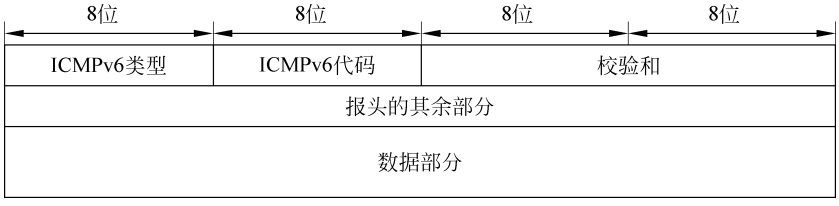


图 3.34 ICMPv6 数据包格式

1. ICMPv6 目的不可达

当数据包无法被转发到目标节点或上层协议时，路由器或目标节点发送 ICMPv6 目的不可达(Destination Unreachable)差错包，ICMPv6 目的不可达数据包格式如图 3.35 所示。

在目的不可达包中，类型(Type)字段值为 1，每一个代码值都定义了具体的含义，这些内容和 ICMPv4 有很多相似之处，例如这里的类型 1 和 IP 中的类型 3 相同，代码 4 和 IP 中的代码 3 相同，代码 0 和 IP 中的代码 0 相同。这里列出常见的代码列表，若想详细了解这些代码的含义，可查看 RFC2463 原始文档。

类型：1	代码：0～6	校验和
未使用		
填充内容：被报错的数据包 (为了满足最小MTU)		

图 3.35　ICMPv6 目的不可达数据包格式

- 代码 0：没有到达目的地的路由。
- 代码 1：禁止与目的地进行通信。
- 代码 2：超出源地址范围。
- 代码 3：地址无法访问。
- 代码 4：端口不可达。
- 代码 5：源地址入口/出口策略失败。
- 代码 6：拒绝到达目的地的路线。

如果是由于拥塞丢包，则不生成 ICMPv6 目的不可达数据包，ICMPv6 并未使用“报头的其余部分”字段，该字段被填充为 0。

2. ICMPv6 数据包超长包

这个类型的 ICMPv6 包是一个新增的类型，用于实现 IPv6 的“路径 MTU 发现”功能，如果由于出口链路的 MTU 小于 IPv6 数据包的长度而导致数据包无法转发，则路由器会发送 ICMP 数据包超长包进行错误通告，同时告知本机接收的最小 MTU 值。该包被用于 IPv6 的路径 MTU 发现处理。ICMPv6 数据包超长包的格式如图 3.36 所示。

类型：2	代码：0	校验和
MTU		
填充内容：被报错的数据包 (为了满足最小MTU)		

图 3.36　ICMPv6 数据包超长包格式

- 类型：2。
- 代码：发送方填充为 0，接收方忽略。
- MTU：链路的最大传输单元。

该数据包是 IPv6 PMTU 的路径 MTU 测试功能中使用的数据包。

3. ICMPv6 超时

ICMPv6 中的超时包和 ICMPv4 中的超时包含义完全相同，仅是把 IP 中的类型 11 变更为类型 3，代码 0 和代码 1 与 IP 中的定义没有区别。当路由器收到一个 IPv6 报头中的跳数限制(Hop Limit)字段值为 1 的数据包时，会丢弃该数据包并向源发送 ICMPv6 超时包。超时包的格式如图 3.37 所示。

- 类型：3。
- 代码：为 0 表示传输期间 Hop Limit 为 0。一旦路由器将数据包的生存时间的字段值递减为 0，就丢弃这个数据包，并向源地址发送 ICMP 超时包。为 1 表示在数

类型：3	代码：0～1	校验和
未使用		
填充内容：被报错的数据包 (为了满足最小MTU)		

图 3.37 ICMPv6 数据包超时包格式

据包分片重组时间超时。当最后的终点在规定的时间内没有收到所有的分片时，它就丢弃已收到的分片，并向源地址发送超时包。

4. ICMPv6 参数问题

如果收到填充错误的 IPv6 数据包，或当 IPv6 报头或者扩展报头出现错误，导致数据包不能进一步处理，则 IPv6 节点会丢弃该数据包并向源发送此包，指明问题的位置和类型。参数问题包的格式如图 3.38 所示。

类型：4	代码：0～2	校验和
指针		
填充内容：被报错的数据包 (为了满足最小MTU)		

图 3.38 ICMPv6 参数问题包格式

- 类型：4。
- 代码：为 0 表示遇到错误的 IPv6 报头字段，为 1 表示遇到无法识别的 Next Header 字段，为 2 表示遇到无法识别的 IPv6 Option。代码 1 和代码 2 是代码 0 的子集，如果处理数据包的 IPv6 节点发现以下字段中的问题：IPv6 报头或扩展报头，使其无法处理数据包，它必须丢弃数据包，并且应该向数据包的源发出 ICMPv6 参数问题消息，指示问题的类型和位置。
- 指针：指明了数据包在什么位置出现了错误。

5. ICMPv6 回显请求

ICMPv6 回显请求是 ICMPv6 中的 ping 请求包，结构如图 3.39 所示。

类型：128	代码：0	校验和
标识		序列号
数据		

图 3.39 ICMPv6 ping 请求包格式

- 类型：128。
- 代码：0。
- 标识/序列号：与 ICMPv4 中相同。

6. ICMPv6 回显应答

ICMPv6 回显应答是 ICMPv6 中的 ping 应答包，如图 3.40 所示。

类型：129	代码：0	校验和
标识		序列号
数据		

图 3.40　ICMPv6 ping 应答包格式

- 类型：129。
- 代码：0。
- 标识/序列号：与 ICMPv4 中相同。

7. ICMPv6 邻居请求

类型为 135，ICMPv6 邻居请求(Neighbor Solicitation，NS)包实现了 IPv6 中的 ARP 请求包功能，并且这是 NDP 数据包的一种，NDP 使用 ICMPv6 包进行承载，实现地址解析、跟踪邻居状态、重复地址检测、路由器发现以及重定向等功能，如图 3.41 所示。

类型：135	代码：0	校验和
保留字(只用于不可达检测报文)		
目的地址		
选项(长度不定，可以是发送方的链路层地址)		

图 3.41　ICMPv6 邻居请求包格式

- 类型：135。
- 代码：必须置 0。
- 目的地址：即要解析的 IPv6 地址。
- 选项：发送此消息主机的链路层地址。

8. ICMPv6 邻居通告

ICMPv6 邻居通告(Neighbor Advertisement，NA)包为 ICMPv6 中的 ARP 应答数据包，类型为 136，ICMPv6 邻居通告包的格式如图 3.42 所示。

类型：136			代码：0	校验和
R	S	O	保留字(发送者初始化为0)	
目的地址				
选项(长度不定，可以是发送方的目的链路层地址)				

图 3.42　ICMPv6 邻居通告包格式

- 类型：136。
- 代码：必须置 0。
- R：路由器(Router)标志。当 R 置 1 时，表示发送者是一个路由器。
- S：被请求(Solicited)标志。当 S 置 1 时，表示发送这个邻居通告是用于响应一个邻居请求的。S 位在邻居不可达检测机制中用于可达性的确认。
- O：重载(Override)标志。

3.8 实验：一次 IPv6 网络环境的 Tracert 流量分析

小张老师偶然发现家中的宽带网络已经升级支持 IPv6，不禁感慨科技的进步对老百姓生活品质的提高，好奇的小张老师使用家里的宽带测试了一下到达 IPv6 站点的连通性，并使用科来 CSNAS 对这个过程进行了数据包的抓取和观察。数据包分析如图 3.43 所示。

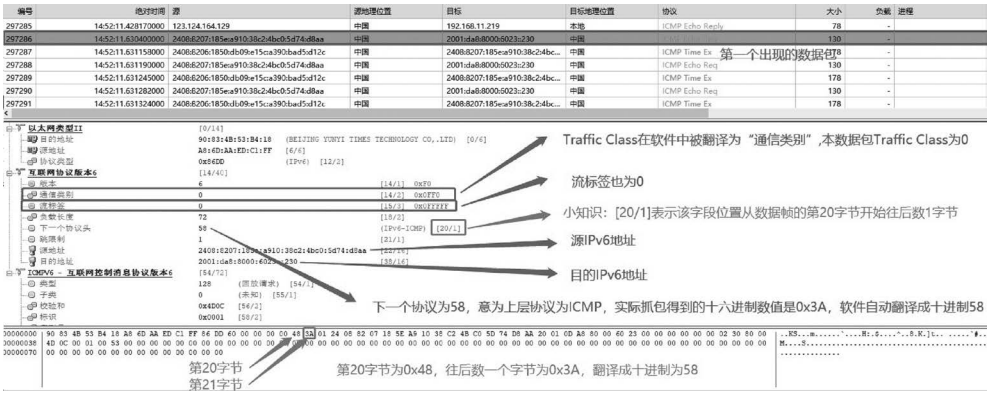


图 3.43 对 Tracert 流量进行分析

第一个出现的 IPv6 数据包的源 IPv6 地址是 2048:8207:185e:a910:38c2:4bc0:5d64:d8aa，这便是小张老师家中主机的 IPv6 公网地址。

该数据包的目的 IPv6 地址是 2001:da8:8000:6023:230，这是测试的目的地址。

该数据包的 Traffic Class 和 Flow Label 都是 0，第一个 0 表示这个数据包在网络中没有优先级，第二个 0 表示数据流编号，不过不知道为什么还没有使用起来。

Next Header 字段的值为 0x3A 或 58（一个是十六进制，一个是十进制，意义一样），这个编号表示上层是 ICMPv6。

通过观察前后数据包的 Hop Limit 不难发现，这实际上是一次 Tracert 的过程，在 IPv6 中的 Tracert 过程和 IP 中并无原理上的区别，唯一的区别在于三层协议一个用的是 IPv4，一个用的是 IPv6。

观察这些数据包的 Hop Limit 字段，结合 Tracert 的 TTL 递增原理（在 IPv6 中，Hop Limit 起到 TTL 的作用，通过观察数据包可以看到发送的 ping 请求 TTL 从 1 开始递增），不难发现 Tracert 行为是由网络中源地址为 2408:8207:185e:a910:38c2:4bc0:5d74:d8aa 的主机发起的，测试的目标地址为 2001:da8:8000:6023::230。

该 Tracert 行为发送的 ICMPv6 数据包类型为 128，代码为 0。这是 ICMPv6 ping Request 数据包。中间节点返回的数据包类型为 3，代码为 0。这是 ICMPv6 中的 TTL 超时数据包。

同时，测试方对每个测试节点进行了 3 次测试，这也与 IPv4 中完全一致。测试到达第 17 个 TTL 时，能够正常收到类型为 129、代码为 0 的 ping Reply 标志，表示测试结束。

到达目的地的路径如图 3.44 所示。

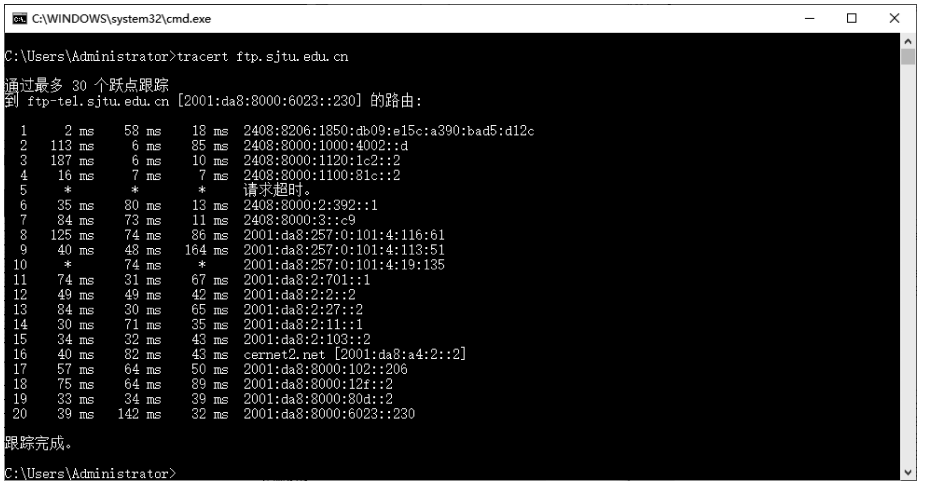


图 3.44 实际的 Tracert 结果

3.9 习 题

1. 在 IP 分片中,IP 标识、标志、分段偏移的作用是什么?

2. 在进行网络流量分析时,IP 标识还有什么其他作用?

3. 在 IP 数据包报头,TTL 字段的作用是什么?

4. 在进行网络流量分析时,TTL 还有什么其他作用?

5. 若 IP MTU 为 1500,在执行 ping 命令时,以不触发 IP 分片为前提条件,最大可指定 ping 包的长度为多少?

6. 当同时启动多个 ping 程序去 ping 同一个地址时,目标返回 ping 包,操作系统如何识别返回的 ping 包对应哪个程序?

7. 如何通过流量分析判断网络环路的存在?

8. 本机数据包分析法与对比分析法有什么区别?

9. IPv6 中的 ARP 如何实现?