

计算机网络期末综合实验

林宇浩 21311274

一、应用层

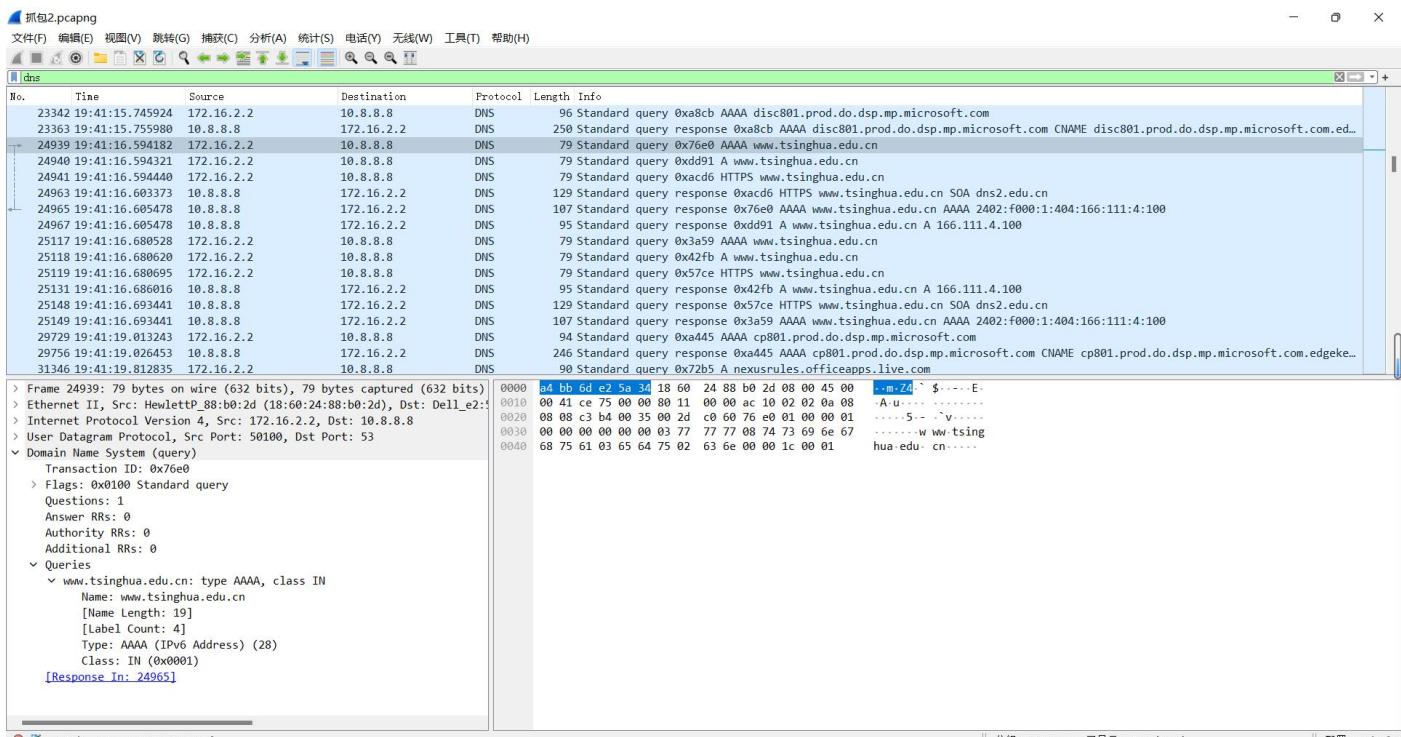
(1) DNS 协议

DNS 协议的基本运行过程为：用户在浏览器中输入域名，操作系统首先查询本地 DNS 缓存，若有则返回 IP 地址。若本地缓存中没有，系统向本地 DNS 服务器发起查询。本地 DNS 服务器有缓存则返回，否则向根域名服务器查询，根服务器指导至顶级域名服务器，逐级查询直至权威域名服务器。最终，权威服务器返回 IP 地址，本地 DNS 服务器缓存并返回给用户设备，实现域名解析。DNS 协议采用递归和迭代查询确保解析到目标 IP 地址。

以下是常用 DNS 协议查询类型：

查询类型	名称代码	含义
1	A	将域名解析到一个指定的 IPV4 的 IP 上。
2	NS	用来指定该域名由哪个 DNS 服务器来进行解析，类似于托管，将子域名交给其他 DNS 服务器解析。
5	CNAME	规范名称，可以将注册的不同域名都转到同一个规范名称上，由这个规范名称统一解析 IP 地址
15	MX	电子邮件交互
16	TXT	文本信息
28	AAAA	将域名解析到一个指定的 IPV6 的 IP 上。
65	HTTPS	非标准 DNS 查询类型，查询域名的 HTTPS 服务端点信息

以下是 DNS 报文的抓包结果：



可以看到第 24939 至 25149 帧的范围是关于清华大学官网 <http://www.tsinghua.edu.cn> 的 DNS 查询，下面对这些报文进行分析。

对于第 24939 帧包含的报文，该报文是一个 DNS 查询报文，源 IP 地址是 172.16.2.2，目的 IP 地址是 10.8.8.8，查询类型是 AAAA，表示查询的目标是 www.tsinghua.edu.cn 的 IPv6 地址。查询报文使用 UDP 协议，目的端口是 53，即 DNS 端口。

我们按照同样的方法分析其他的报文，分析结果见下表：

帧号	行为
24939	查询 www.tsinghua.edu.cn 的 IPv6 地址
24940	查询了域名 www.tsinghua.edu.cn 的 IPv4 地址
24941	查询了域名 www.tsinghua.edu.cn 的 HTTPS 服务端点信息，并希望通过递归查询获取该信息
24963	对之前 DNS 查询报文的响应，其中包含了关于 www.tsinghua.edu.cn 域名的授权服务器信息
24965	对之前 DNS 查询报文的响应，其中包含了 www.tsinghua.edu.cn 域名的 IPv6 地址 2402:f000:1:404:166:111:4:100
24967	对之前 DNS 查询报文的响应，其中包含了 www.tsinghua.edu.cn 域名的 IPv4 地址，为 166.111.4.100
25117	查询了域名 www.tsinghua.edu.cn 的 IPv6 地址，并希望通过递归查询获取该地址
25118	查询了域名 www.tsinghua.edu.cn 的 IPv4 地址，并希望通过递归查询获取该地址
25119	查询了域名 www.tsinghua.edu.cn 的 HTTPS 服务端点，并希望通过递归查询获取该信息
25131	对之前 DNS 查询报文的响应，其中包含了 www.tsinghua.edu.cn 域名的 IPv4 地址，为 166.111.4.100
25148	对之前 DNS 查询报文的响应，表明 www.tsinghua.edu.cn 的 HTTPS 服务端点信息需要从 tsinghua.edu.cn 的权威域名服务器上获取
25149	对之前 DNS 查询报文的响应，表明 www.tsinghua.edu.cn 的 IPv6 地址为 2402:f000:1:404:166:111:4:100

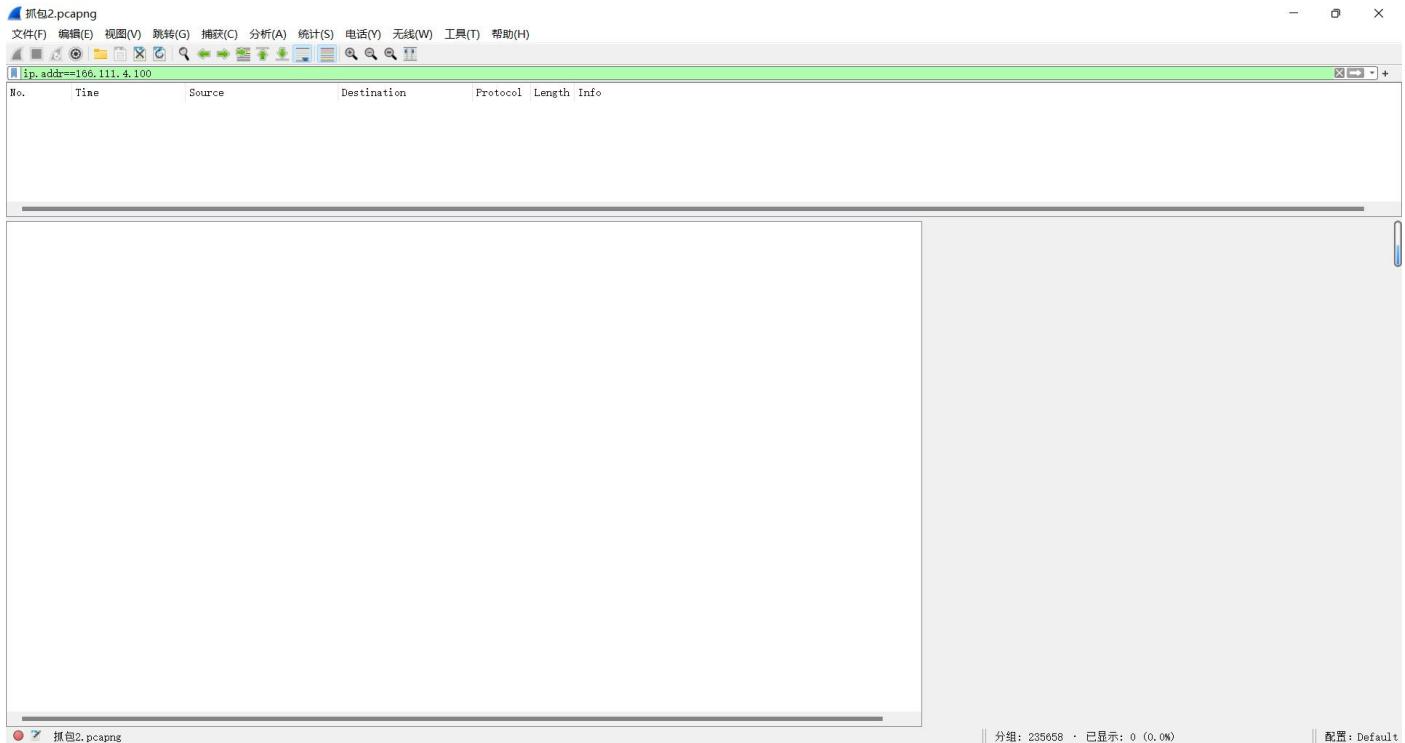
(2) HTTP 协议（应用层）和 TCP 协议（运输层）

HTTP (Hypertext Transfer Protocol, 超文本传输协议) 是一种用于在 Web 浏览器和服务器之间传输数据的协议。其基本运行过程如下：客户端通常是浏览器，其向服务器发送 HTTP 请求，请求中包含要访问的资源的信息，如 URL、请求方法等。服务器接收到请求后，处理并返回一个 HTTP 响应，其中包含了请求的资源或相关信息，以及响应状态码表示请求的处理结果。通信过程中可以包含多次请求和响应。HTTP 协议是无状态的，每个请求之间没有直接关联，为了保持状态，通常使用 Cookie 等机制。这种请求-响应模型支持 Web 中的数据传输和信息交互，使得浏览器能够加载和呈现 Web 页面。

下面我们对 HTTP 协议进行分析，由于 HTTP 协议和 TCP 协议连接紧密，所以我们把运输层的 TCP 协议也在这里一同分析。

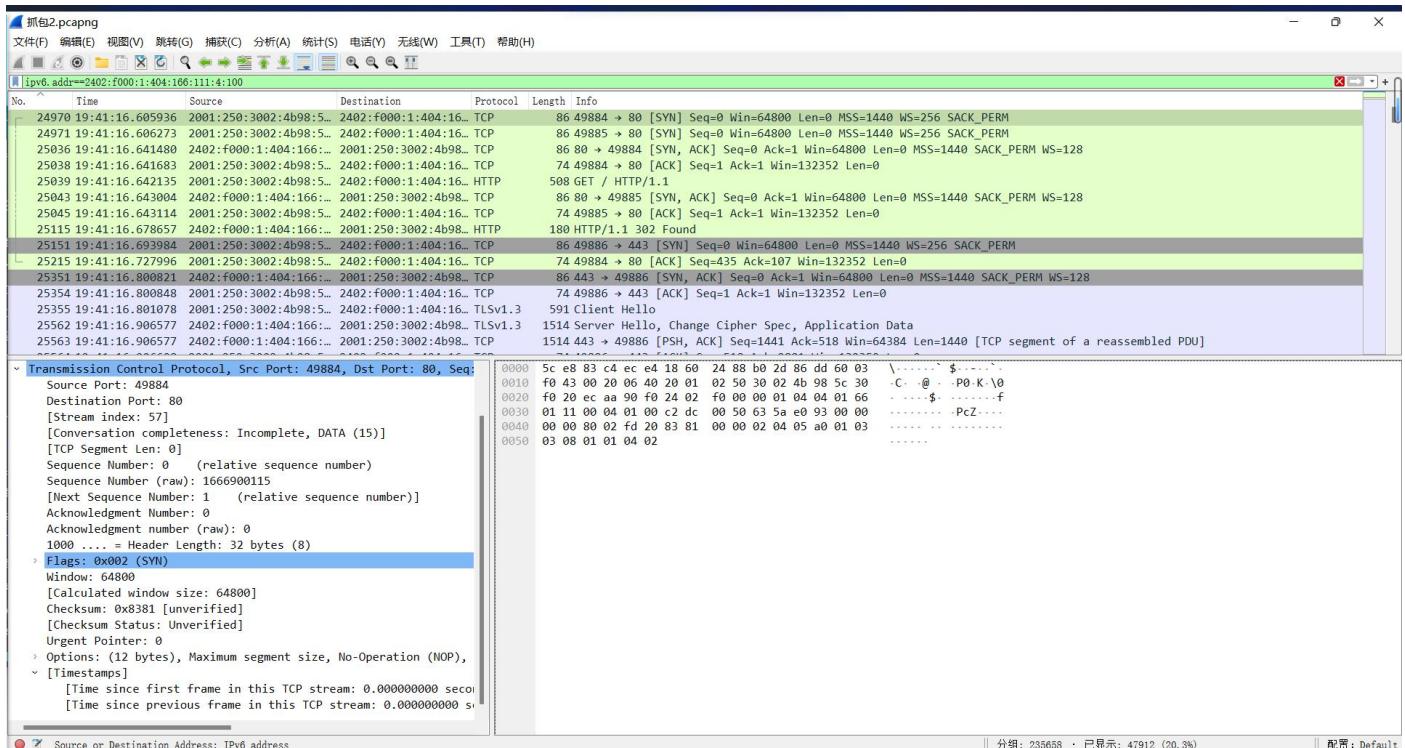
通过刚才对 DNS 报文的分析，我们得到了服务器的 IPv6 地址为 2402:f000:1:404:166:111:4:100 和 IPv4 地址为 166.111.4.100。

搜索对应的 IPv4 地址，我们发现没有捕捉到相关报文，如下所示：



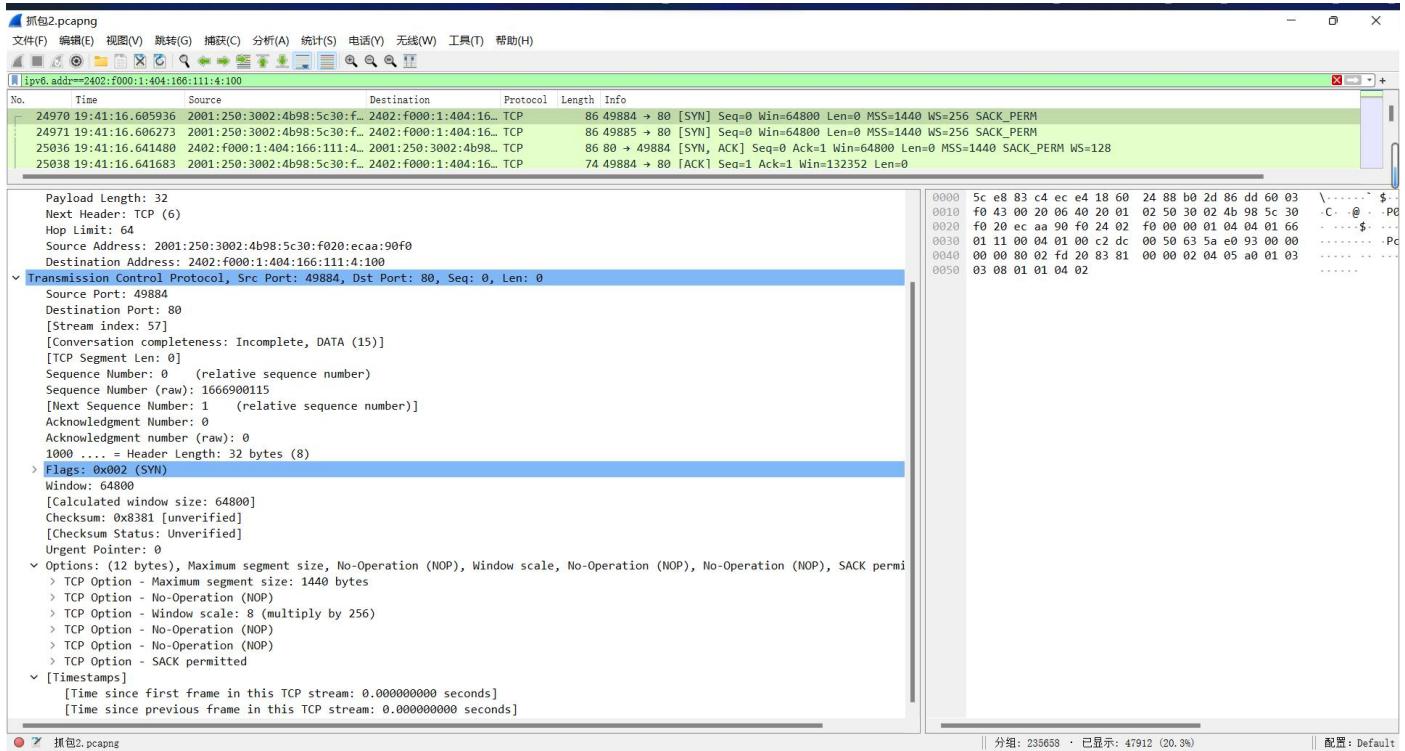
说明客户端和服务器直接是用 IPv6 协议进行通信的，所以我们下面关注对应的 IPv6 地址。

过滤对应的 IPv6 地址后如下所示：

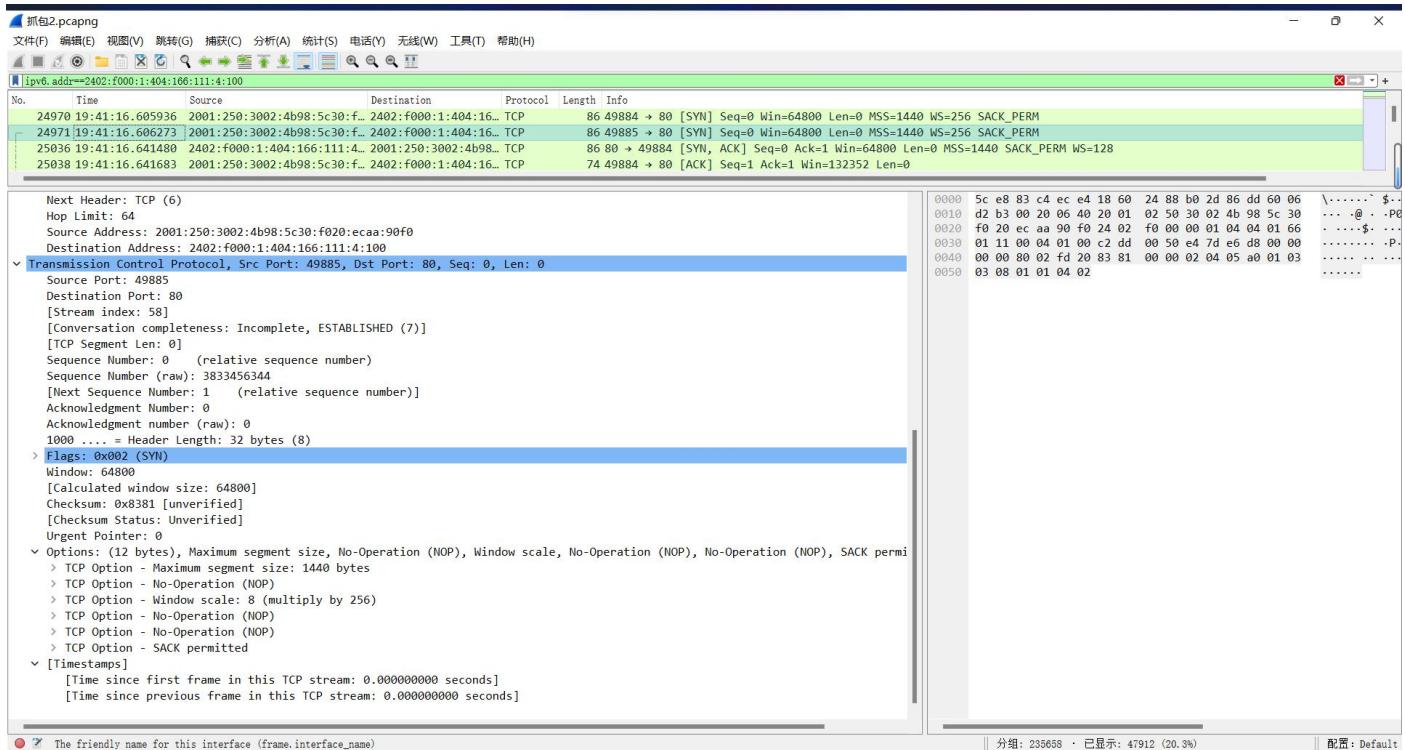


经过过滤和仔细检查全部的 TCP 连接，发现主机总共有 3 个端口对目的服务器发起了 TCP 连接，下面我们对这三个连接逐一分析。

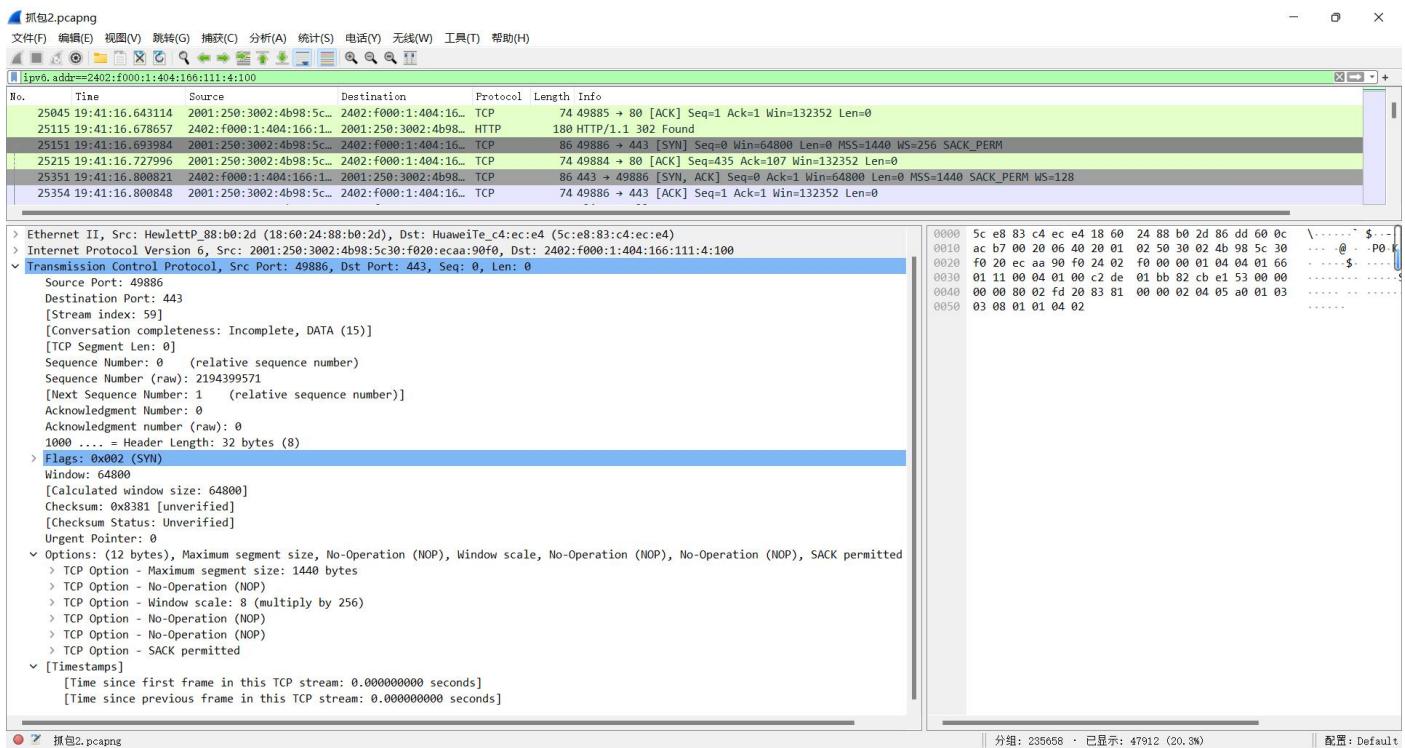
对于下图中第 24970 帧包含的报文，这是一个 TCP 的连接建立请求（SYN）数据包，源地址为 2001:250:3002:4b98:5c30:f020:ecaa:90f0，目标地址为 2402:f000:1:404:166:111:4:100，源端口为 49884，目标端口为 80。这表示设备在尝试与目标地址的端口 80 建立连接。



对于下图中第 24971 帧包含的报文，这个报文是另一个 TCP 连接建立请求，源地址为 2001:250:3002:4b98:5c30:f020:ecaa:90f0，目标地址为 2402:f000:1:404:166:111:4:100，源端口为 49885，目标端口为 80。

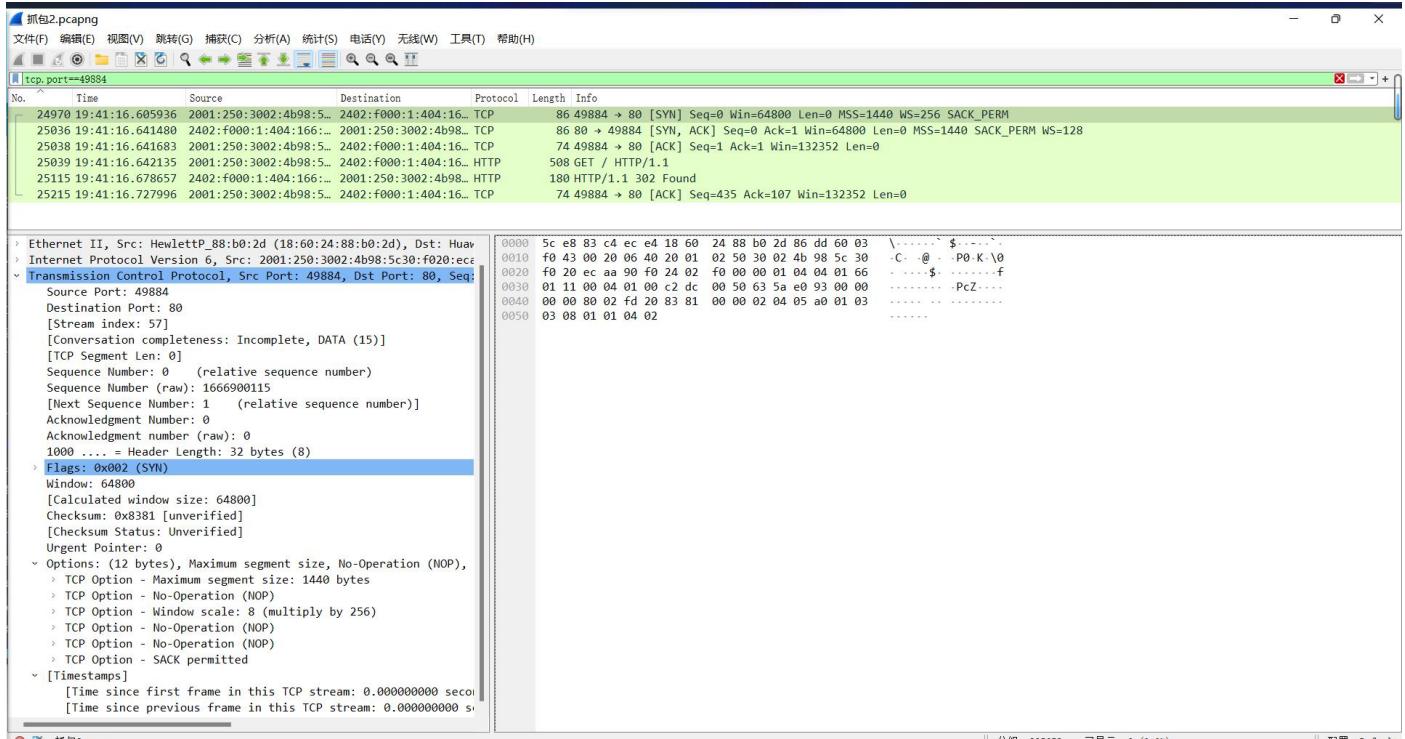


对于下图中第 25151 帧包含的报文，源地址为 2001:250:3002:4b98:5c30:f020:ecaa:90f0，目标地址为 2402:f000:1:404:166:111:4:100，源端口为 49886，目标端口为 443，这是 HTTPS 协议的默认端口。这是一个 IPv6 网络上的 TCP 连接请求，目标是进行 HTTPS 连接。



下面我们通过区分源端口号对这 3 个 TCP 连接进行分析：

首先是源端口 49884 发起的 TCP 连接，如下所示：



我们点开每一个链路帧，逐一对各个帧进行分析，如下表所示：

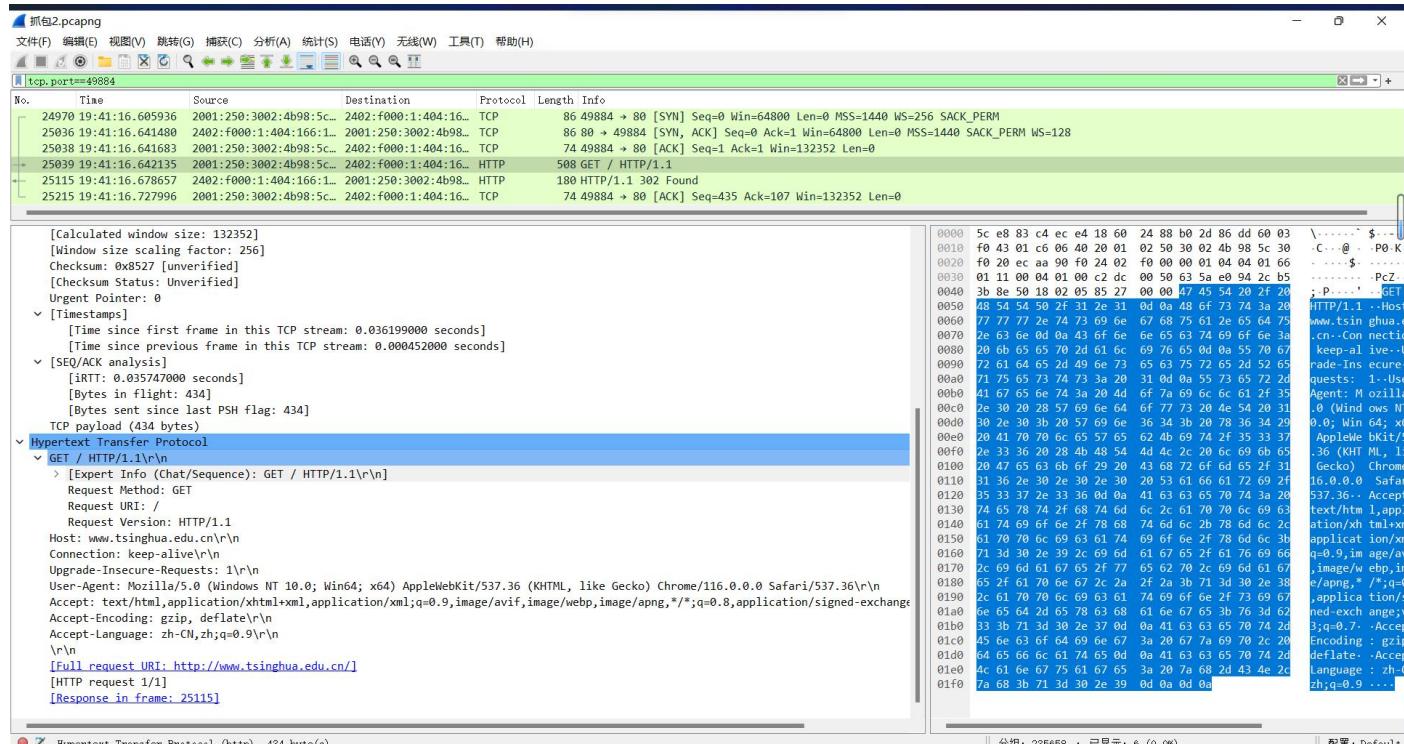
帧号	行为
24970	IPv6 主机通过 TCP 协议尝试与目标主机建立连接，源端口为 49884 的发送方发送一个 SYN 标志的报文，其中包括一些 TCP 选项，如最大段大小和窗口缩放。

25036	该报文表示源端口为 80 的目标主机已经收到并确认了源端口为 49884 的主机发起的连接请求，并同意建立连接。连接的建立过程中，已经完成了三次握手的前两步。
25038	该报文表示连接建立的最后一步，源端口号为 49884 的主机已经确认目标端口号为 80 的主机的握手响应，连接成功建立。
25039	该报文表示一个通过 IPv6 和 TCP 的 GET 请求，源端口号为 49884 的主机请求访问 http://www.tsinghua.edu.cn/ ，并发送了一些 HTTP 头部信息
25115	该报文表示一个 HTTP 302 Found 的重定向响应，指示客户端应该通过 HTTPS 访问 https://www.tsinghua.edu.cn/
25215	该报文表示对之前发送的数据的确认，确认号为 107，说明成功接收到序列号为 107 的数据。我们知道确认号用于确认已经成功接收到的数据，号码表示期望接收的下一个字节的序列号。这里确认号为 107，与上一个 25115 号帧中其序列号为 1，长度为 106 对应，说明接收无误。

综上所述，在这个连接中，我们的主机作为客户端向服务器发起了 TCP 连接，建立连接后客户端通过 HTTP 协议发起了一个 GET 请求，服务器返回了 302 Found 的状态码，要求客户端重定向到另一个基于 HTTPS 协议的 URL，即 <https://www.tsinghua.edu.cn/>。

现在我们具体分析一下该连接中传输的两个 HTTP 报文。

第一个 HTTP 报文如下所示：



在请求行中，“GET”为 HTTP 请求方法，表示获取资源。

“/”表示请求的资源路径，这里表示根目录。

“HTTP/1.1”是 HTTP 协议版本号，表示使用 HTTP 1.1 协议。

在头部字段，“Host: www.tsinghua.edu.cn\r\n”指定被请求资源的主机名。在虚拟主机的环境中，服务器可以根据该字段的值来决定提供哪个主机的网站内容。

“Connection: keep-alive\r\n”表示客户端希望与服务器保持持久连接，以便进行多个请求/响应交换。

“Upgrade-Insecure-Requests: 1\r\n”告知服务器，客户端支持 HTTPS，并希望服务器升级到 HTTPS 连接。

“User-Agent”部分是客户端用户代理标识，描述了客户端的应用程序、操作系统、厂商等信息。这

里是 Chrome 浏览器的标识。

“Accept”部分告诉服务器客户端可接受的媒体类型及其相对优先级，用于协商服务器返回的内容类型。

“Accept-Encoding: gzip, deflate\r\n”指定客户端支持的内容编码方式，这里表示支持 gzip 和 deflate 压缩方式。

“Accept-Language: zh-CN, zh;q=0.9\r\n”表示客户端接受的自然语言，用于协商服务器返回的内容语言。

“\r\n”两个回车换行符表示 HTTP 头部结束，后面可能包含请求体（在这个例子中是空的）。

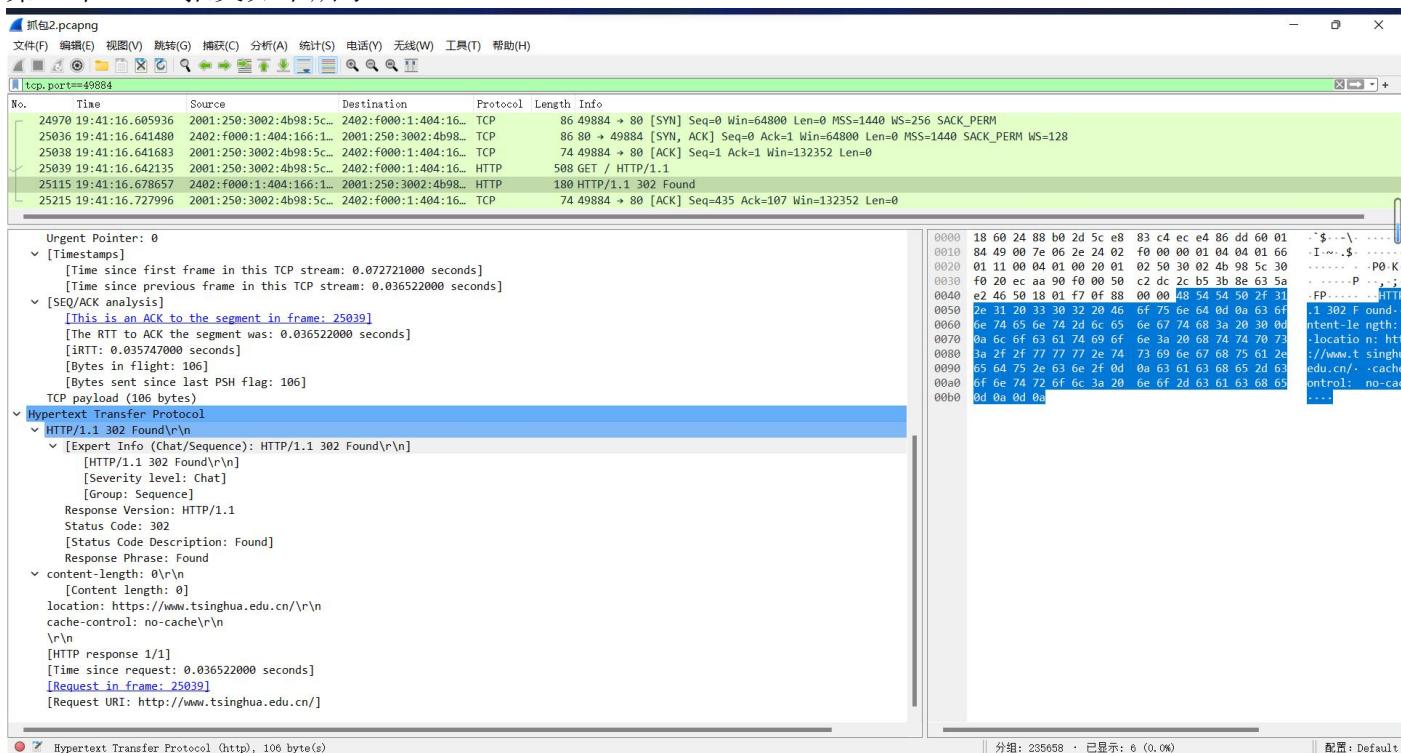
接下来的部分是 wireshark 软件生成的提示，“Full Request URI”表示完整的请求 URI，即 http://www.tsinghua.edu.cn/，完整的请求 URI 会包括协议、主机、端口和路径等信息。

“HTTP request 1/1”表示这是请求中的第一部分，总共有一个请求部分。

“Response in frame: 25115”表示该请求的响应在帧 25115 中。

总结来看，这个 HTTP 请求是一个 GET 请求，请求的资源是根目录/。请求头中包含了一些标准的 HTTP 头部字段，用于描述请求的方法、资源、协议版本、主机等信息，以及一些用于协商内容的字段。

第二个 HTTP 报文如下所示：



在 HTTP 响应行中，“HTTP/1.1 302 Found\r\n”表示 HTTP 协议版本为 1.1。

状态码为 302，响应短语为 Found。这个响应状态码表明请求的资源已被临时重定向到新的位置，由 Location 头部提供。

在响应头部，“content-length: 0\r\n”表示响应内容长度为 0。在这种情况下，因为是 302 Found，通常不包含实体主体内容。

“location: https://www.tsinghua.edu.cn/\r\n”指示客户端应该重定向到的新位置，这里是 https://www.tsinghua.edu.cn/。302 状态码通常用于实现临时性的重定向。

“cache-control: no-cache\r\n”指定缓存控制策略，这里是 no-cache，表示不应缓存此响应。

“\r\n”表示 HTTP 头部的结束。

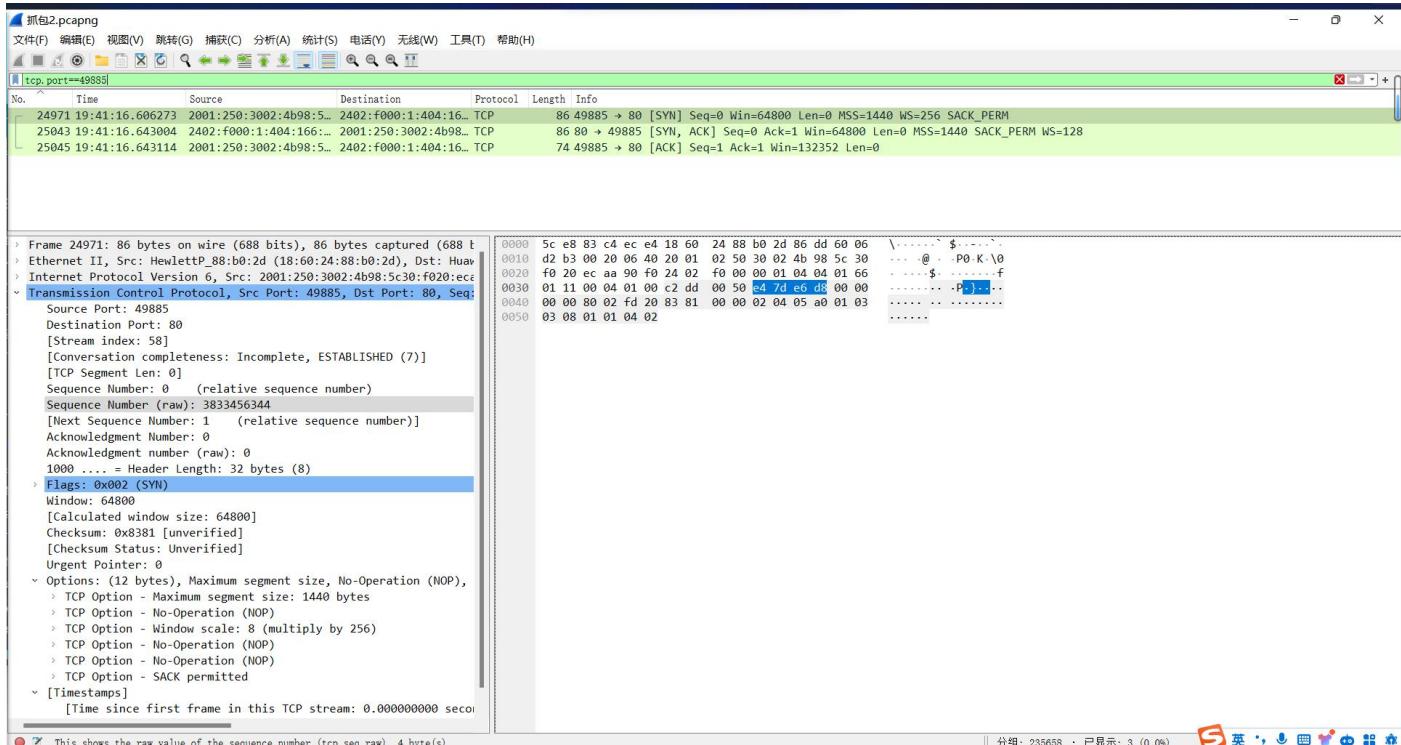
在 HTTP 响应信息部分，这一部分是 wireshark 软件生成的提示，“[HTTP response 1/1]”表示这是 HTTP 响应中的第一部分，说明这个响应是一个完整的响应。

“[Time since request: 0.036522000 seconds]”表示自请求发送后经过的时间，这里是0.036522000秒。

“[Request in frame: 25039]”表示与此响应相关联的请求在帧25039中。

“[Request URI: http://www.tsinghua.edu.cn/]”表示请求的URI是http://www.tsinghua.edu.cn/，与此响应相关联。

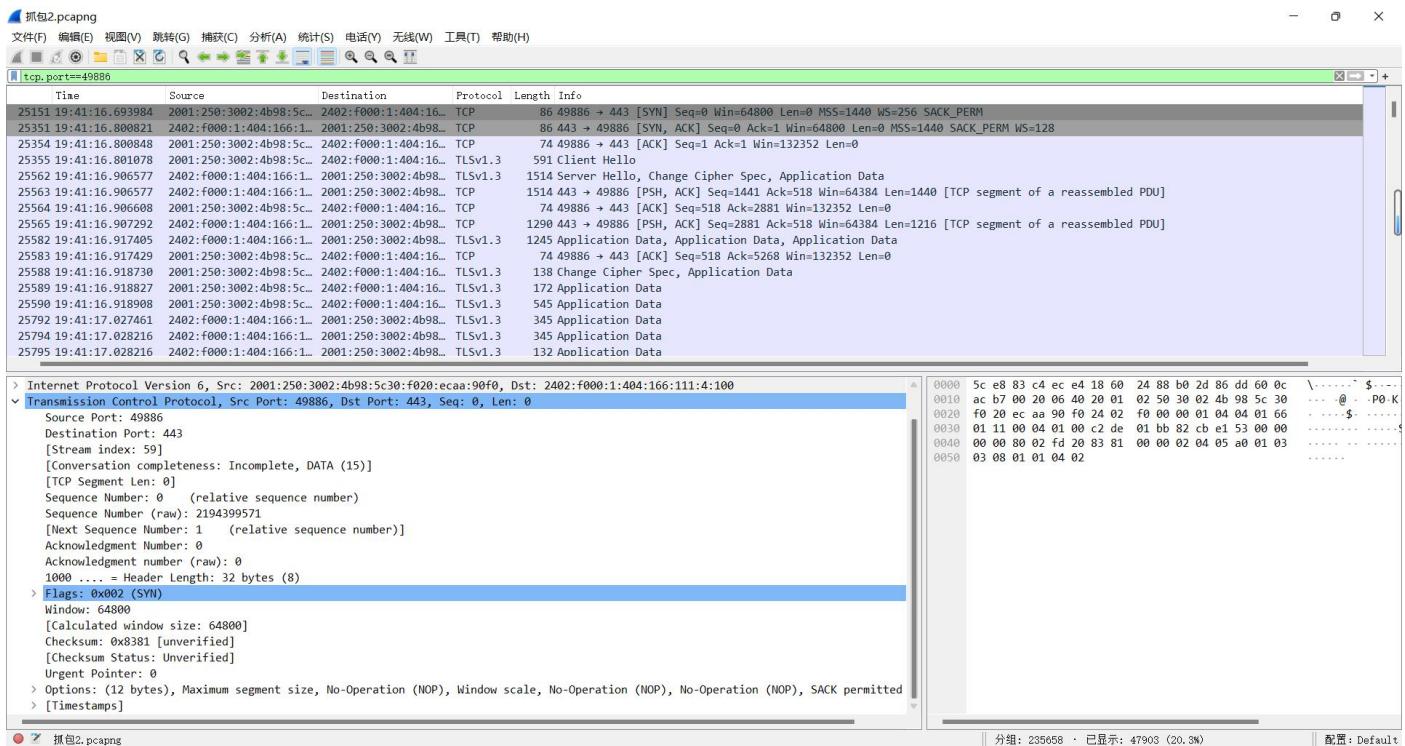
然后我们分析源端口49885发起的TCP连接，如下所示：



帧号	行为
24971	这个报文是TCP连接的初始握手阶段。源主机的IPv6地址为2001:250:3002:4b98:5c30:f020:ecaa:90f0，端口为49885，目的主机IPv6地址为2402:f000:1:404:166:111:4:100，端口为80。客户端向服务器发送了一个SYN（同步）请求，请求建立TCP连接。
25043	这个报文是TCP连接建立的第二步，是对前一个SYN请求的确认，并在同一个报文中发起对方的SYN请求，完成三次握手过程。标志位为SYN, ACK，表示连接正在建立。连接状态为“Incomplete, ESTABLISHED (7)”，表明连接还没有完全建立，但处于建立中。
25045	这个报文是TCP连接建立的第三步，是对前一个SYN, ACK报文的确认，表示连接已经建立。

可以看到这个TCP连接建立之后没有进一步发送数据。

最后我们分析源端口 49886 发起的 TCP 连接，如下所示：



帧号	行为
25151	这个报文尝试在 IPv6 网络中建立到目的地址的 TCP 连接，目的端口是标准的 HTTPS 端口 443。报文的内容表明这是一个初始的连接请求，由于序列号为 0，表示此连接尚未建立。
25351	这个报文是对之前发起的连接请求的响应，表示目标系统已收到连接请求（SYN），并确认建立了连接。报文中的标志位标明这是一个 SYN, ACK 报文，表示同时是连接请求的确认和连接建立成功的通知。
25354	这个报文是连接建立的第三步，表示连接已经建立并确认，是对之前接收到的连接请求的响应。报文的标志位标明这是一个 ACK 报文，表示对方已成功接收到之前发送的连接请求（SYN）并确认建立连接。
25355	这个报文是一个 TLS 客户端发送的 Client Hello 消息。该消息是 TLS 握手协议的一部分，用于启动安全连接的建立。在这个报文中，客户端提供了支持的密码套件、压缩方法等信息，并请求与服务器建立安全连接。报文中的 JA3 和 JA3 Fullstring 是用于识别 TLS 客户端的指纹信息。整个过程表明客户端正在尝试与服务器建立一个安全的 TLS 连接，用于安全的网页浏览。
25562	为 TLS 握手过程中的服务器响应。具体来说，它包含了 Server Hello 消息，其中服务器选择了加密套件和其他 TLS 握手过程所需的信息。服务器发送 Change Cipher Spec 消息，表示从现在开始通信将使用新的加密参数。最后，服务器发送加密的应用层数据，可能是 HTTPS 的一部分。整个过程表明服务器成功地接受了客户端的连接请求，并建立了安全连接。
25563	这是一个包含应用层数据的 TLS 报文。这里数据长度为 1440 字节，表示在 TLS 安全连接上通过 HTTPS 或其他加密应用传输的数据。标志位 PSH 表示推送数据给应用层。这个数据包是之前握手过程的后续步骤，表示服务器已经成功地接受了客户端的请求并建立了安全连接，现在可以进行加密的应用层通信。
25564	这是 TCP 的 ACK 确认数据包，用于确认上一个数据包，即 25563 帧对应数据包的接收。ACK 确认了服务器成功接收了来自客户端的数据，序列号从 1441 至 2880，并且服务器的下一次期望的数据序列号是 518。这个数据包是 TCP 连接维持和流控制的一部分，确保双方的通信同步。在时间戳和 Round-Trip Time 分析中，RTT 为 0.000031000 秒，表示从上一个数据包发送到接收 ACK 的时间。

25565	这是一个包含应用层数据的 TCP 报文，数据长度为 1216 字节，表示在 TLS 安全连接上通过 HTTPS 或其他加密应用传输的数据。标志位 PSH 表示推送数据给应用层。
25582	该报文是 TCP 数据流的一部分，属于应用层的 HTTPS 通信。TLSv1.3 的使用表明双方之间建立了安全连接，确保了数据的加密传输。
.....

综上所述，这些报文创建了一个建立在 TCP 和 TLSv1.3 协议之上的 HTTPS 连接。发送方和服务器之间经过三次握手建立了 TCP 连接，然后通过 TLSv1.3 协议进行安全通信，最终传输了一些加密的 HTTP 数据。

现在我们总结这三个不同源端口发起的 TCP 连接。

首先，主机的 49884 端口和 49885 端口在非常相近时间内，先后向目的服务器发送了 TCP 连接请求，49884 端口发起的 TCP 连接先完成建立，建立连接后客户端通过 HTTP 协议发起了一个 GET 请求，服务器返回 302 Found 状态码，要求客户端重定向到基于 HTTPS 协议的 URL。期间 49885 端口发起的 TCP 连接也完成建立，但是之后没有继续使用。之后，主机的 49886 端口发送了 TCP 连接请求，与服务器建立了 HTTPS 连接，后面客户端和服务器通过 HTTPS 进行加密通信。

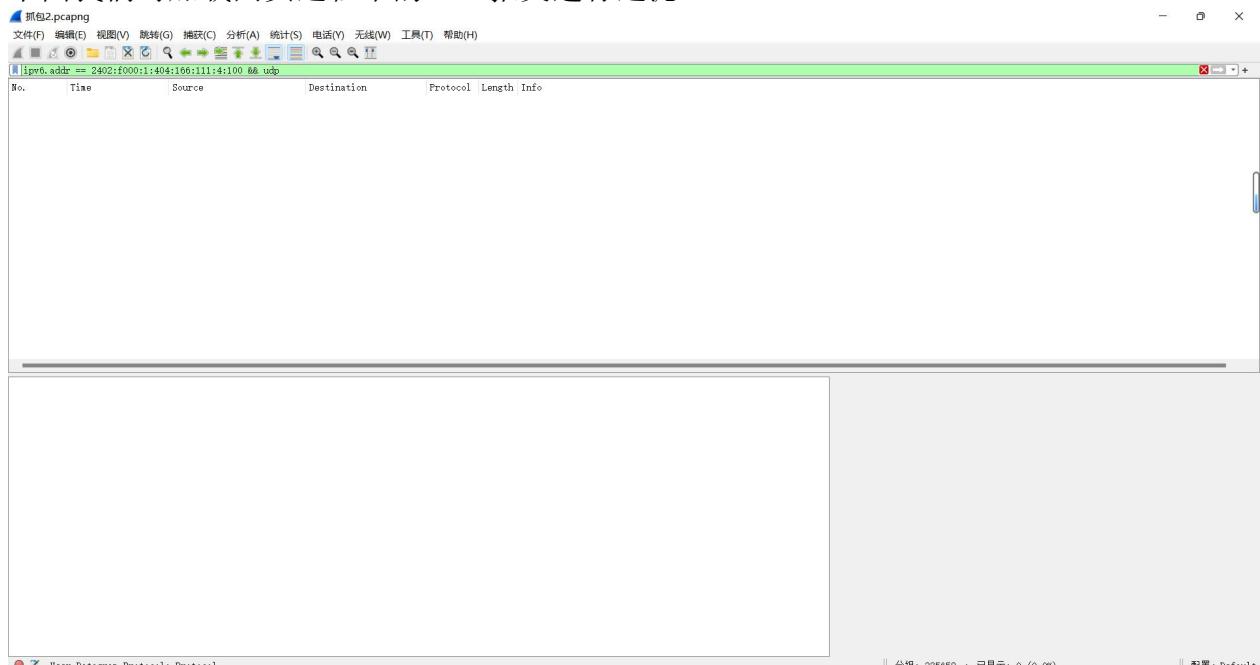
我们看到一开始主机用不同的端口发起了两个 TCP 连接请求，这可能是主机在同时尝试与服务器建立两个独立的 TCP 连接从而实现并行连接，而在连接重定向到 HTTPS 后，通信均通过 HTTPS 连接传输，这两个连接后面都没有被继续使用。

二、传输层

(1) UDP 协议

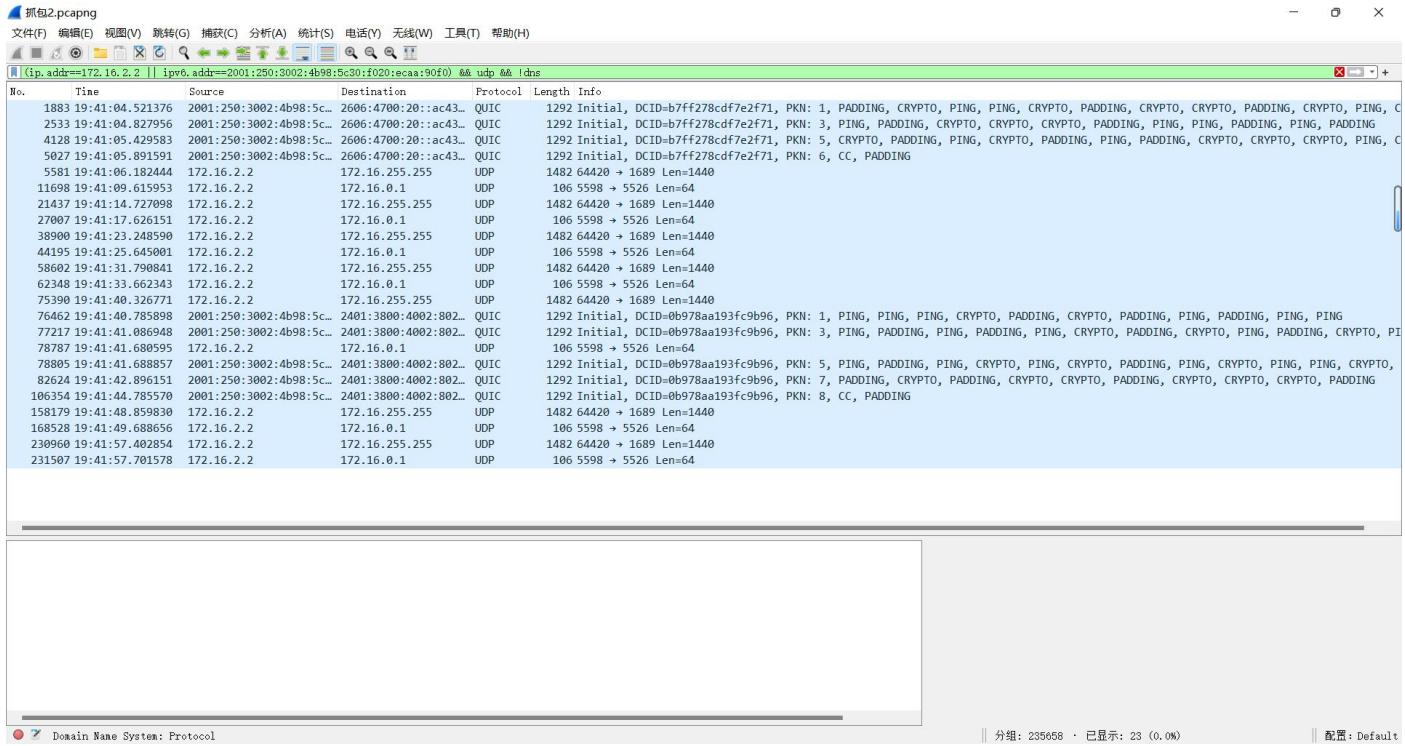
UDP (User Datagram Protocol, 用户数据报协议) 是一种无连接的、不可靠的通信协议。其基本运行过程如下：发送端将要传输的数据封装成 UDP 数据包，包括源端口、目标端口、数据长度等信息。然后，通过网络传输至接收端。UDP 不进行握手和连接建立，因此不存在三次握手的过程。接收端收到 UDP 数据包后，直接提取数据并交给应用程序处理。由于 UDP 不保证可靠性，数据包可能在传输过程中丢失或乱序，且没有重传机制。UDP 适用于对实时性要求较高、可以容忍少量数据丢失的应用场景，如音频和视频传输。UDP 的简单性和低开销使其在特定应用中得到广泛应用。

下面我们将对加载网页过程中的 UDP 报文进行过滤：



可以看到，通过过滤器，在与服务器通信的过程中并没有使用到 udp 数据包，但是也许网页的某些对象通过一个 URL 寻址，在获取这个对象的时候可能使用了 UDP 协议进行通信，但是由于 HTTPS 协议是加密的，我们很难通过报文发现是否有在加载网页的过程中使用到了 UDP 协议。

不过，过滤掉除 DNS 以外的 UDP 报文后，发现 UDP 数据包并不多，如下所示，可以一条条进行分析

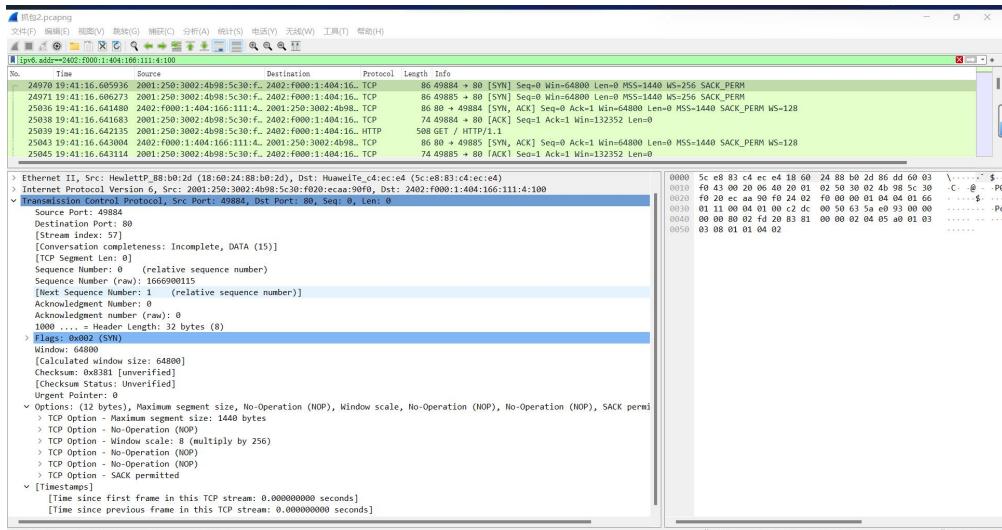


通过逐一检查并与网页加载时间比较，发现这里并没有与网页加载直接相关的 UDP 报文，所以可以推断网页获取 HTML 文件和对象的过程中，并没有使用到 UDP 协议。

(2) TCP 协议

TCP (Transmission Control Protocol, 传输控制协议) 是一种面向连接的、可靠的通信协议。其基本运行过程如下：在进行通信前，客户端和服务器建立连接，通过三次握手 SYN、SYN-ACK、ACK 确立双方的通信信道。一旦连接建立，数据传输阶段开始，数据被分割为小块，即 TCP 段，并按序发送。接收方收到数据后发送确认应答，若发送方未收到确认，会进行重传。连接维持期间，通过流量控制和拥塞控制机制来调整数据的传输速率，确保网络上的可靠、有序传输。通信结束后，双方发起四次挥手来关闭连接，确保数据完整性。TCP 协议提供可靠的、面向连接的通信，适用于许多网络应用。

我们刚才已经在应用层部分将 TCP 协议与 HTTP 协议一同分析了，这里进行一些补充。



我们以前面分析 HTTP 协议时的 TCP 报文段为例子，下面对 TCP 报文段的结构进行分析。

Source Port(源端口)为“49884”是数据包的发送方使用的端口号，用于标识发送方的应用程序或服务。

Destination Port(目标端口)为“80”，是数据包的接收方使用的端口号，通常是 HTTP 服务的默认端口。

Sequence Number(序列号)为“0”，用于标识 TCP 数据流中的每个字节的位置。在这个数据包中，序列号为 0，表示这是连接的初始阶段。

Acknowledgment Number(确认号)为“0”，是期望接收的下一个字节的序列号。在这个初始的 SYN 数据包中，确认号为 0，因为尚未收到任何数据。

Flags(标志)为“0x002 (SYN)”，指示 TCP 数据包的状态。在这里，SYN 标志被设置，表示这是一个连接请求。

Window(窗口大小)为“64800”，指示发送方还能接收多少字节的数据，而不需要等待确认。

在 Options(选项)中包含了一些选项字段，如最大段大小(Maximum segment size)、窗口缩放(Window scale) 和 SACK 允许(Selective Acknowledgment) 等。这些选项提供了有关连接和数据传输的附加信息。

Checksum(校验和)为“0x8381 [unverified]”，用于检测数据包在传输过程中是否发生了错误。

Urgent Pointer(紧急指针)为“0”，紧急指针用于指示紧急数据的结束位置。在这个数据包中，紧急指针为 0，表示没有紧急数据。

Timestamps(时间戳)用于测量报文往返时间等，这有助于调整数据传输的性能。这里包含了时间戳选项，用于记录时间信息。

我们在应用层部分将 TCP 协议与 HTTP 协议进行了一同分析，在运输层这部分进一步分析了 TCP 协议，现在我们分析应用层与传输层协议的消息交换过程：

在开始通信之前，应用层需要通过传输层的协议来建立连接。通常，应用层会使用 TCP 用于可靠的连接的传输层协议。通过 TCP 的三次握手，能够建立起客户端和服务器之间的连接。这个过程包括客户端向服务器发送 SYN(同步) 请求，服务器回应 ACK(确认) 和 SYN，最后客户端再次回应 ACK。

在连接建立之后，应用层开始准备要发送的消息。应用层协议，如 HTTP 或 SMTP 等，其定义了消息的格式和规范。应用层消息被传递到传输层，并封装成传输层协议的数据包，如 TCP 报文段。TCP 会添加头部信息，包括源端口、目标端口、序列号、确认号、标志位等。使用 TCP，传输层会负责确保数据的可靠传输。它通过序列号、确认号、重传机制等手段来保证数据的可靠性。如果数据包在传输过程中丢失或损坏，TCP 会负责重新发送。接收方的传输层接收到数据包后，进行解封装，将数据传递给应用层。应用层根据应用层协议的规定，解析数据包，获取原始的应用层消息。应用层接收到消息后，进行相应的处理。这可能包括解析数据、执行特定的应用逻辑、生成响应等。应用层协议规定了如何处理接收到的消息。在连接建立后，应用层和传输层的消息交互可以是持续的。多个应用层消息可以通过同一连接进行交换，从而减少连接的建立和关闭开销。当应用层完成消息的交互后，可以选择关闭连接。在 TCP 中，通过四次挥手来关闭连接。

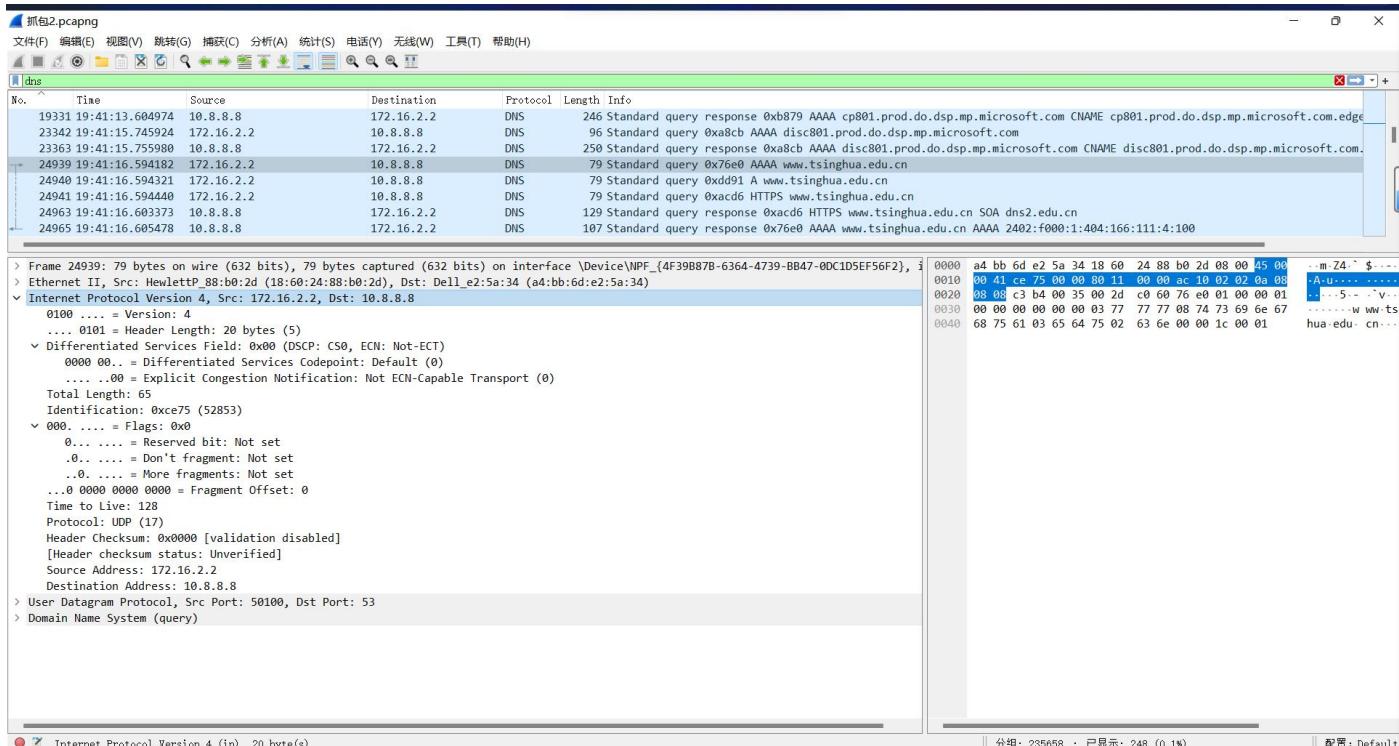
三、网络层

(1) IP 协议

IP (Internet Protocol, 互联网协议) 是网络层协议，用于在互联网上进行数据包的传输。其基本运行过程如下：发送端将数据划分成数据包，并在数据包中添加源和目标 IP 地址等信息。然后，数据包通过网络传输，经过多个路由器和网络节点，每个节点根据目标 IP 地址决定下一跳的路径。数

据包最终到达目标主机，根据目标 IP 地址交给相应的网络层协议处理。IP 协议是无连接、不可靠的，不保证数据包的传输顺序、不提供错误检测和纠正。其主要功能是实现点对点的数据传输，为更高层的协议提供可靠的传输基础。IP 地址的唯一性和全球唯一的路由表是实现互联网规模的关键。

我们使用下面的 DNS 报文分析 IPv4 数据报



“Version: 4” 表示 IPv4 的版本号，这里是 IPv4。

“Header Length: 20 bytes (5)” 表示 IPv4 头部的长度，这里是 20 字节。

“Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)” 表示服务质量的字段，包括 DSCP（区分服务代码点）和 ECN（显式拥塞通知）。

“Total Length: 65” 表示整个 IPv4 数据包的长度，包括头部和数据部分。

“Identification: 0xce75 (52853)” 表示标识字段，用于唯一标识一个数据包。

“Flags: 0x0” 表示标志字段，通常用于控制分片。

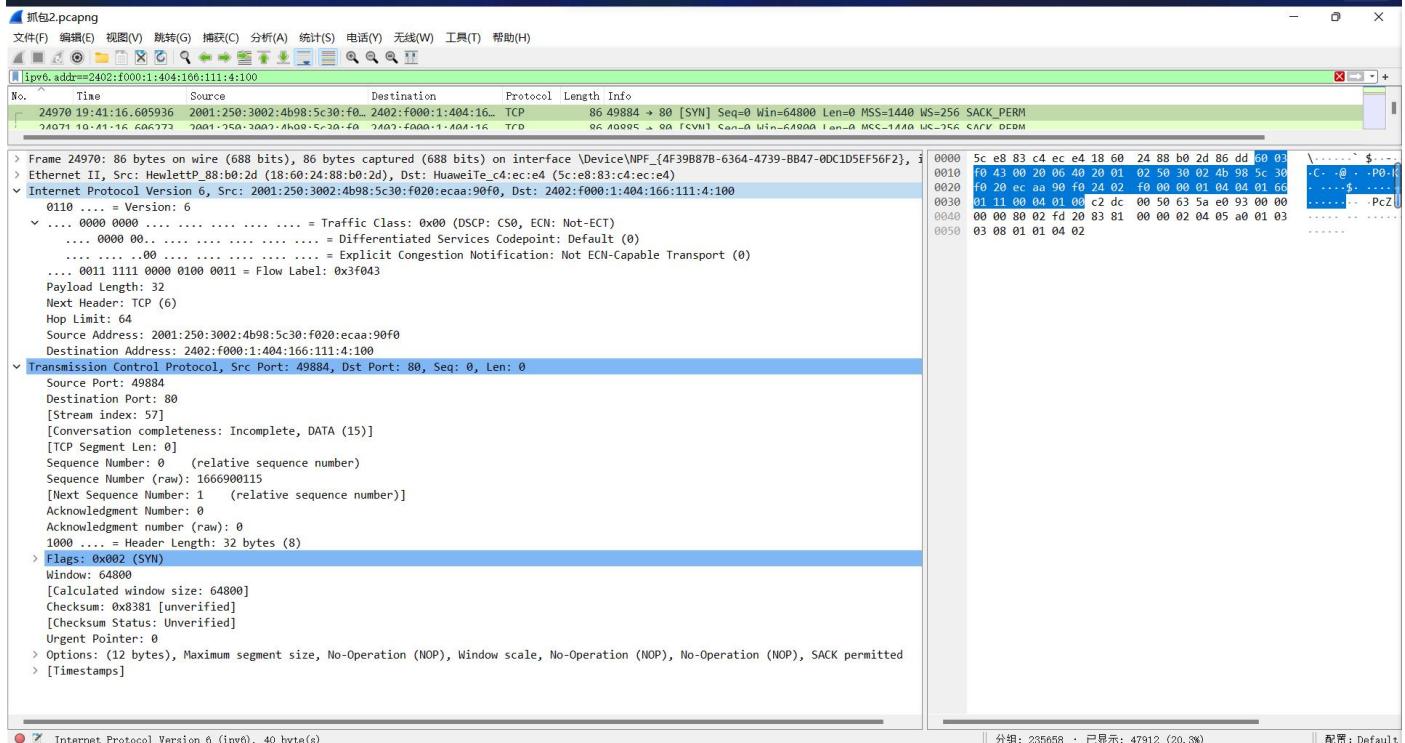
“Time to Live: 128” 表示数据包的生存时间，每经过一个路由器，该值减 1，直到为 0，此时数据包被丢弃。

“Protocol: UDP (17)” 表示传输层协议，这里是 UDP。

“Header Checksum: 0x0000 [validation disabled]” 表示头部的校验和，用于检测头部信息是否正确。

“Source Address: 172.16.2.2, Destination Address: 10.8.8.8” 表示 IPv4 数据包的源 IP 地址和目标 IP 地址。

使用下面的 TCP 报文分析 IPv6 数据报：



“Version: 6” 表示 IPv6 的版本号，这里是 IPv6。

“Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)” 表示流量类别字段，包括 DSCP（区分服务代码点）和 ECN（显式拥塞通知）。

“Flow Label: 0x3f043” 表示流标签，用于标识同一流的数据包。

“Payload Length: 32” 表示 IPv6 数据包有效负载的长度。

“Next Header: TCP (6)” 表示下一层协议是 TCP。

“Hop Limit: 64” 表示数据包的生存跳数，类似 IPv4 的生存时间 (Time to Live)。

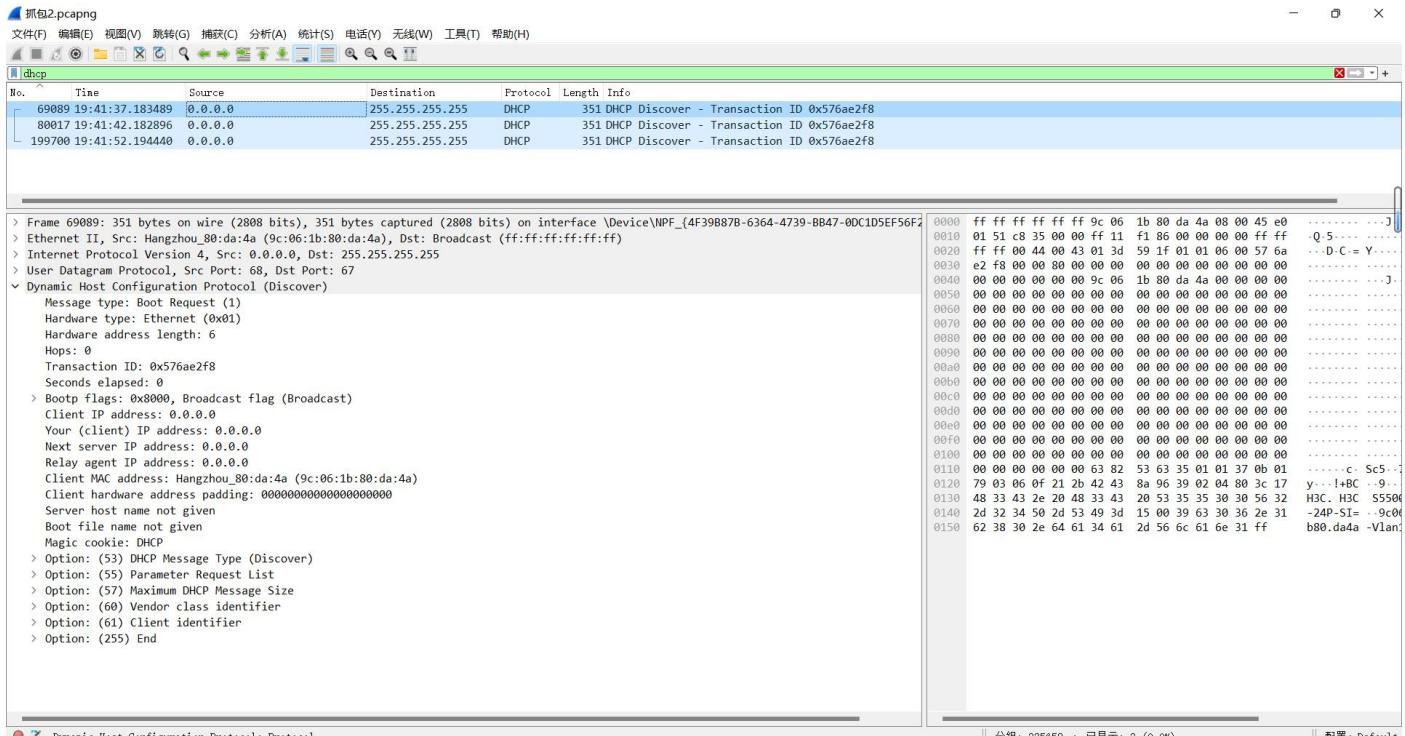
“Source Address: 2001:250:3002:4b98:5c30:f020:ecaa:90f0” 表示 IPv6 数据包的源 IPv6 地址

“Destination Address: 2402:f000:1:404:166:111:4:100” 表示 IPv6 数据包的目标 IPv6 地址

(2) DHCP 协议

DHCP (Dynamic Host Configuration Protocol, 动态主机配置协议) 是用于自动分配网络设备 IP 地址及其他网络配置信息的协议。其基本运行过程如下：当设备连接到网络时，它会发送一个 DHCP 请求广播，请求一个可用的 IP 地址。DHCP 服务器在网络中接收到这个请求后，选取一个未分配的 IP 地址并将其分配给设备，同时提供其他网络配置信息如子网掩码、网关、DNS 服务器等。设备接收到 DHCP 服务器的响应后，配置自身网络参数。租约时间确定了设备可以使用该 IP 地址的时间。当租约过期或设备离开网络时，IP 地址将返回到 DHCP 服务器的可用池中。DHCP 协议的动态分配机制使得网络管理更为灵活和自动化。

下面是过滤出的 DHCP 消息：



消息类型（Message Type）为“Boot Request (1)”，表示这是一个 DHCP 的发现消息。DHCP 协议定义了不同的消息类型，包括 Discover、Offer、Request、Acknowledgment 等。Discover 消息用于客户端在网络上查找可用的 DHCP 服务器。

硬件类型（Hardware Type）为“Ethernet (0x01)”，表示硬件类型为以太网，即客户端的网络适配器是以太网适配器。

事务 ID（Transaction ID）为“0x576ae2f8”，用于唯一标识 DHCP 事务。在 DHCP 的交互中，事务 ID 用于匹配客户端和服务器之间的通信，确保它们彼此之间的消息是关联的。

秒数计时器（Seconds Elapsed）为 0，是客户端已经等待服务器响应的时间，这里为 0 表示刚刚开始等待。

Bootp 标志（Bootp Flags）为“0x8000, Broadcast flag (Broadcast)”表示该消息是广播消息，发送给网络上的所有设备。这是 DHCP Discover 消息的一种特性。

客户端 IP 地址（Client IP address）为“0.0.0.0”，是客户端尚未分配到 IP 地址。

客户 MAC 地址（Client MAC address）为“Hangzhou_80:da:4a (9c:06:1b:80:da:4a)”，是客户端的物理地址，用于唯一标识客户端设备。

下面分析 DHCP 选项：

Option 53 – DHCP 消息类型（Discover）：表示这是一个 DHCP Discover 消息，用于发现可用的 DHCP 服务器。

Option 55 – 参数请求列表：包含客户端请求的特定参数列表，如子网掩码、路由器、DNS 等。

Option 57 – 最大 DHCP 消息大小：1152 字节，表示 DHCP 消息的最大尺寸。

Option 60 – 厂商类别标识符：H3C_H3C_S5500V2-24P-SI，指定了客户端的设备类型。

Option 61 – 客户标识符：9c06.1b80.da4a-Vlan1，提供了客户端的标识信息。

Option 255 – 结束：表示 DHCP 选项的结束。

可以发现，这条 DHCP Discover 消息的目的是启动 DHCP 过程，以获取必要的网络配置信息，为客户端设备分配合适的 IP 地址和其他网络参数，使设备能够顺利地加入网络。

DHCP 的消息交换过程包括四个主要步骤：首先，DHCP 客户端在网络中广播 Discover 消息，以寻找可用的 DHCP 服务器。接着，DHCP 服务器收到 Discover 消息后，向客户端发送 Offer 消息，其中包含

可用的 IP 地址和其他配置信息。客户端选择一个 Offer 并发送 Request 消息，请求分配特定的 IP 地址。最后，DHCP 服务器确认客户端的请求，向其发送 Acknowledge 消息，完成 IP 地址和其他配置信息的分配。

由于 DHCP 的过程通常相对迅速，在局域网中，DHCP 服务器往往能够快速响应客户端的请求，完成 IP 地址的分配。所以我们打开 wireshark 前，本机的 IP 地址已完成分配，我们无法抓到本机自身的 DHCP 消息，我们抓到一般都是同一子网下其他主机的 DHCP 消息。

(3) OSPF 协议

OSPF（Open Shortest Path First，开放最短路径优先）是一种用于路由的链路状态协议，其基本运行过程如下：路由器在 OSPF 域内通过 Hello 消息进行邻居关系的发现，建立相邻关系的路由器之间交换链路状态信息，包括链路的状态和成本。这些信息被用来构建一个拓扑图，表示网络中各个路由器和链路之间的关系。通过计算最短路径树，每个路由器得知到达目标网络的最优路径。OSPF 使用 Dijkstra 算法来计算最短路径。一旦路由表建立，路由器就能根据最短路径选择转发数据包。OSPF 通过周期性的 Hello 消息和事件驱动的链路状态更新来保持网络状态的最新。OSPF 协议的优点包括快速收敛、支持大规模网络和提供负载平衡等特性。

下面是捕捉到的 OSPF 协议的 HELLO 报文：

The screenshot shows a Wireshark capture of OSPF traffic. The packet list pane displays several OSPF Hello Packets (Protocol: OSPF, Length: 78-90 bytes). The details pane shows the structure of an OSPF Hello Packet, including fields like Version: 3, Type: Hello Packet (1), Length: 36, Source OSPF Router: 10.44.185.254, Area ID: 0.0.0.4, and Checksum: 0xa631 [correct]. The bytes pane shows the raw hex and ASCII data of the packet.

可以看到，源地址为 fe80::e4a5:8e4e:1da1:ddd4%9 是本主机发出的 OSPFv3 Hello Packet。OSPF Hello 报文充当邻居发现的手段。路由器通过周期性地发送 Hello 报文，通告自己的存在，以及一些基本的 OSPF 信息。当其他路由器收到这些 Hello 报文时，它们可以识别出邻居，并建立邻居关系。子网中的路由器能够自动学习网络拓扑，动态选择最优路径，并自适应网络拓扑的变化。

可以发现，当前子网中，除了本机外，还存在一个 IP 地址为 172.18.186.126 的设备，其通过发送 OSPFv2 Hello Packet 声明其信息。

```
选择 C:\Windows\system32\cmd.exe
C:\Users\D502>ipconfig
Windows IP 配置

以太网适配器 以太网 3:
    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 WLAN:
    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

以太网适配器 校园网:
    连接特定的 DNS 后缀 . . . . . :
    IPv6 地址 . . . . . : 2001:250:3002:4b98:813c:d7cf:c2cf:d0d1
    临时 IPv6 地址 . . . . . : 2001:250:3002:4b98:5c30:f020:ecaa:90f0
    本地链接 IPv6 地址 . . . . . : fe80::e4a5:8e4e:1da1:ddd4%9
    IPv4 地址 . . . . . : 172.16.2.2
    子网掩码 . . . . . : 255.255.0.0
    默认网关 . . . . . : fe80::5ee8:83ff:fec4:ece4%9
                           172.16.0.1

C:\Users\D502>
```

我们可以看到，OSPF 协议使用的是本地链接 IPv6 地址。我们这里分析一下这三种地址的区别。

IPv6 地址：

IPv6 地址是全球唯一的标识符，用于标识网络中的设备。由 128 位长度的地址组成，通常以 16 进制表示，以冒号分隔。IPv6 地址的前缘部分表示网络，后缘部分表示主机。

临时 IPv6 地址：

临时 IPv6 地址是为了提高隐私和安全性而引入的。它们在临时性需求时动态生成，并在设备重新启动后更改。临时 IPv6 地址通常用于向外发起连接，以减少设备被跟踪的风险。

本地链接 IPv6 地址：

本地链接 IPv6 地址是在设备上的本地网络通信中使用的地址。这些地址通常以 fe80 开头，后跟设备的标识符。本地链接地址只在特定的本地链路上可用，不会被路由到其他网络。

四、链路层

(1) ARP 协议

ARP (Address Resolution Protocol, 地址解析协议) 是用于将 IP 地址映射到物理 MAC 地址的协议。其基本运行过程如下：当设备需要与目标设备通信时，首先检查本地 ARP 缓存表，查看是否已知目标 IP 地址对应的 MAC 地址。如果在缓存中找到，则直接使用该 MAC 地址进行通信。若缓存中没有对应的记录，设备将发起 ARP 请求，广播一个 ARP 请求帧到网络，询问该 IP 地址对应的 MAC 地址。目标设备收到 ARP 请求后，回应一个 ARP 响应，其中包含自己的 MAC 地址。发起请求的设备将这个 MAC 地址存储在本地 ARP 缓存中，并使用它进行后续通信。ARP 协议实现了在网络层和链路层之间的地址映射，确保数据能够正确传递到目标设备。

```
C:\Windows\system32\cmd.exe
172.16.0.1

C:\Users\D502>ipconfig /all

Windows IP 配置

主机名 . . . . . : D52_05
主 DNS 后缀 . . . . . : 
节点类型 . . . . . : 混合
IP 路由已启用 . . . . . : 是
WINS 代理已启用 . . . . . : 否

以太网适配器 以太网 3:

媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . : 
描述 . . . . . : Realtek Common Ethernet Controllers
物理地址 . . . . . : 44-33-4C-0E-CE-82
DHCP 已启用 . . . . . : 是
自动配置已启用 . . . . . : 是

无线局域网适配器 WLAN:

媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . : 
描述 . . . . . : Ralink RT61 Turbo Wireless LAN Card
物理地址 . . . . . : 00-0D-0A-4B-09-EF
DHCP 已启用 . . . . . : 是
自动配置已启用 . . . . . : 是

以太网适配器 校园网:

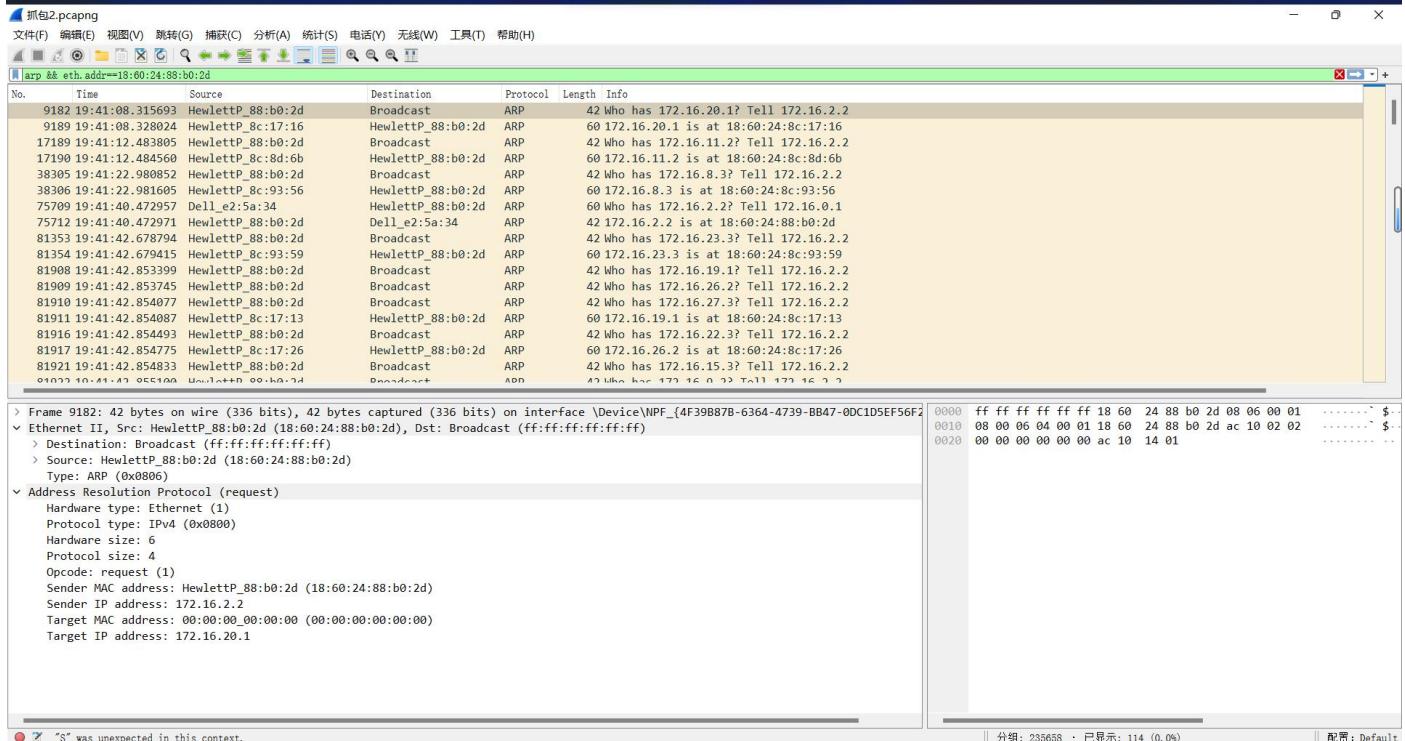
连接特定的 DNS 后缀 . . . . . : 
描述 . . . . . : Realtek PCIe GBE Family Controller #2
物理地址 . . . . . : 18-60-24-88-B0-2D
DHCP 已启用 . . . . . : 否
自动配置已启用 . . . . . : 是
IPv6 地址 . . . . . : 2001:250:3002:4b98:813c:d7cf:c2cf:d0d1(首选)
临时 IPv6 地址 . . . . . : 2001:250:3002:4b98:5c30:f020:ecaa:90f0(首选)
本地链接 IPv6 地址 . . . . . : fe80::e4a5:8e4e:1dal:ddd4%9(首选)
IPv4 地址 . . . . . : 172.16.2.2(首选)
子网掩码 . . . . . : 255.255.0.0
默认网关 . . . . . : fe80::5ee8:83ff:fed4:ec04%9
                      172.16.0.1
DHCPv6 IAID . . . . . : 169369636
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-2C-7F-10-AC-00-88-99-00-12-F3
DNS 服务器 . . . . . : 10.8.8.8
                           10.8.4.4
TCPIP 上的 NetBIOS . . . . . : 已启用

C:\Users\D502>
```

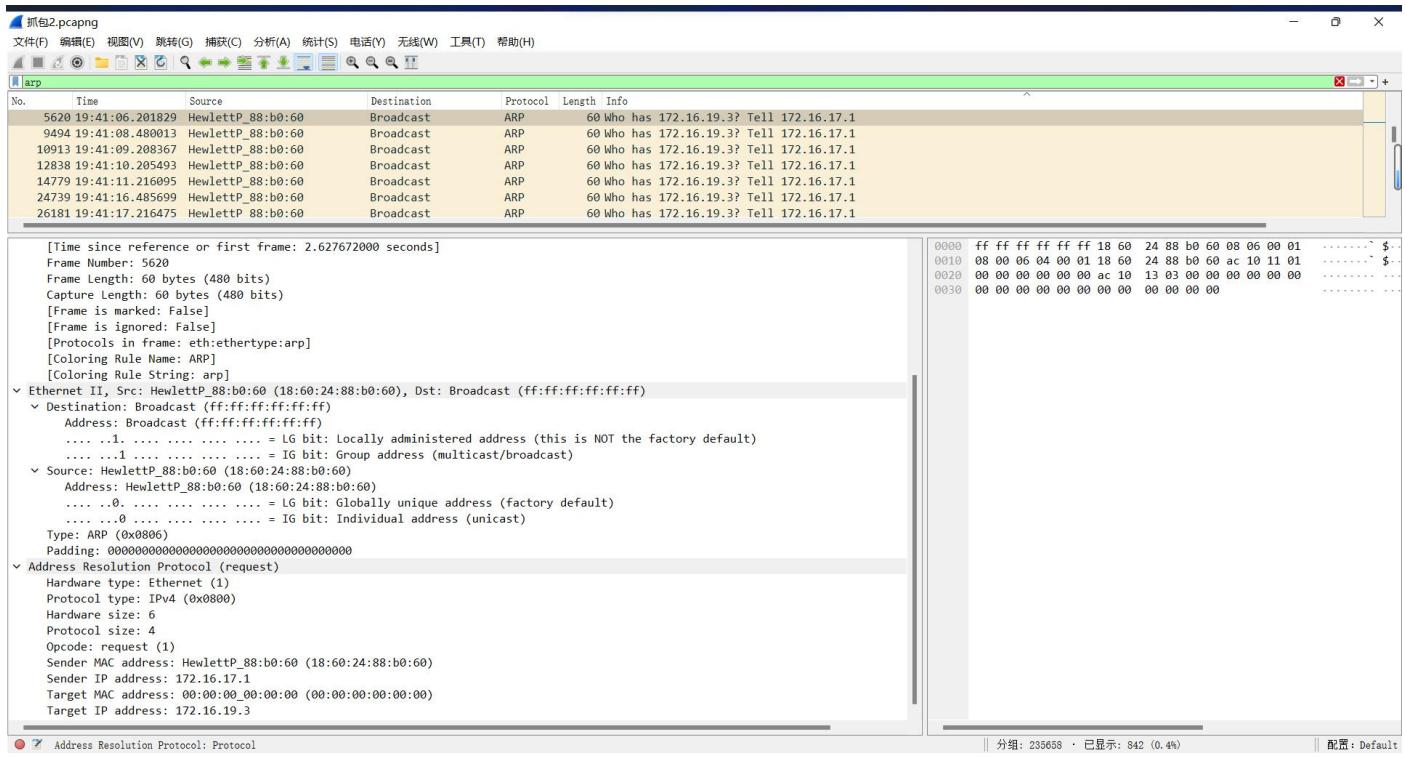
如图所示，本机对应适配器的 MAC 地址为 18:60:24:88:b0:2d



捕获到的 arp 数据包如下所示



现在对 ARP 分组进行分析



对于上图中第 5620 帧包含的 ARP 分组分析如下：

“Ethernet (1)” 表示硬件类型为以太网。

“IPv4 (0x0800)” 表示协议类型为 IPv4。

“Hardware size” 值为 6，是硬件地址长度，表示 MAC 地址的长度为 6 个字节。

“Protocol size” 值为 4，是协议地址长度，表示 IPv4 地址的长度为 4 个字节。

“request (1)” 是操作码，表示这是一个 ARP 请求。

“HewlettP_88:b0:2d (18:60:24:88:b0:2d)” 是发送 ARP 请求的设备的 MAC 地址。

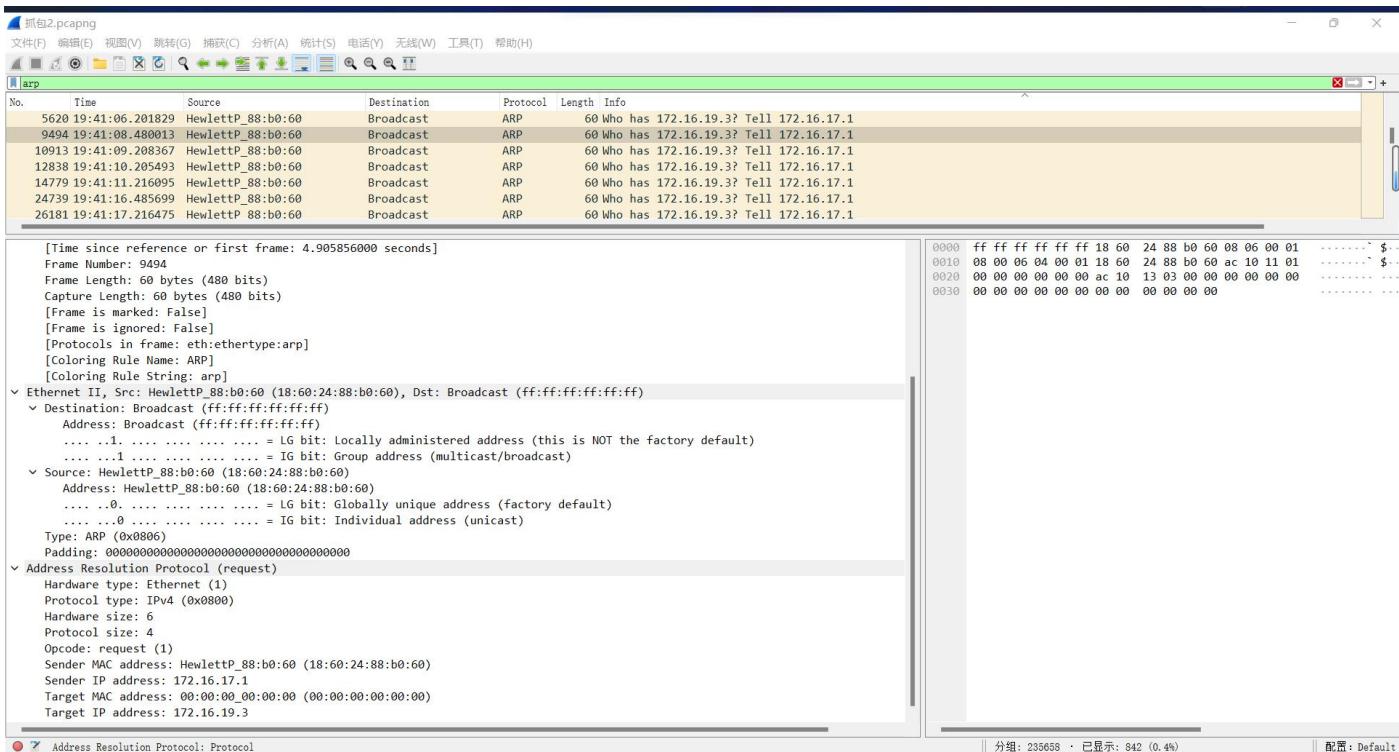
“18:60:24:88:b0:2d” 被显示为 “HewlettP_88:b0:2d” 是因为 Wireshark 通常会尝试根据 MAC 地址来识别设备的制造商。这里 Wireshark 识别这个 MAC 地址的前缀 “18:60:24” 属于惠普 (Hewlett-Packard) 制造商。

“172.16.2.2”是发送ARP请求的设备的IP地址。

“00:00:00:00:00:00”是目标MAC地址，在ARP请求中，这个字段通常被设置为0，因为请求的目的是获取目标设备的MAC地址。

“172.16.20.1”是目标设备的IP地址，对应于发送ARP请求的设备想要解析的IP地址。

所以这个ARP请求的目的是获取IP地址为172.16.20.1的设备的MAC地址，以建立通信。



对于上图中第9494帧包含的ARP分组分析如下：

“Ethernet (1)”表示硬件类型为以太网。

“IPv4 (0x0800)”表示协议类型为IPv4。

“Hardware size”值为6，是硬件地址长度，表示MAC地址的长度为6个字节。

“Protocol size”值为4，是协议地址长度，表示IPv4地址的长度为4个字节。

“reply (2)”是操作码，表示这是一个ARP回复。

“172.16.20.1”是发送ARP回复的设备的IP地址。

“HewlettP_88:b0:2d (18:60:24:88:b0:2d)”是目标MAC地址，对应于ARP请求中的发送方MAC地址。

“172.16.2.2”是目标设备的IP地址，对应于ARP请求中的发送方IP地址。

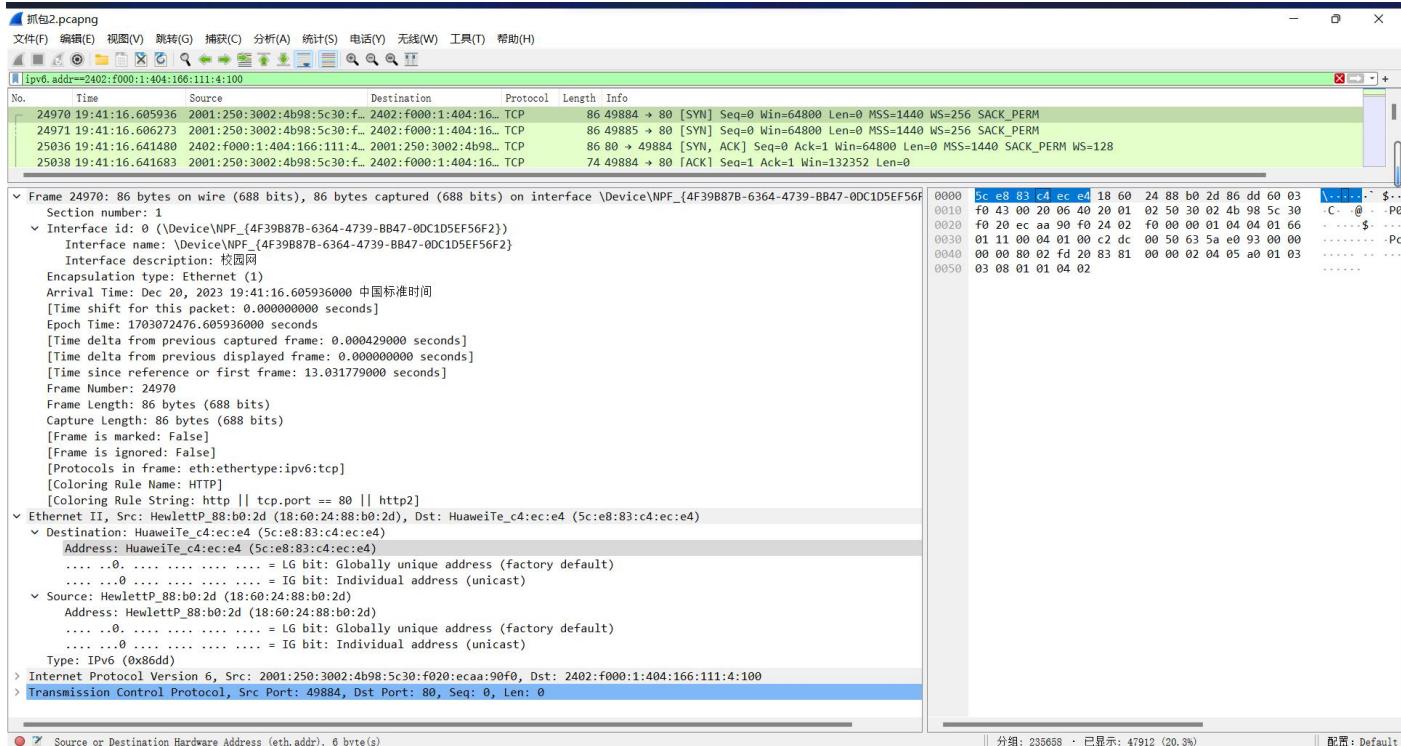
所以这个ARP回复的目的是告知发出ARP请求的设备，IP地址为172.16.20.1的设备的MAC地址是“HewlettP_8c:17:16”，以建立通信。

(2) 以太网

以太网是一种在局域网中广泛应用的协议，其基本运行过程如下：设备在以太网中通过封装数据成帧的方式进行通信。首先，将传输的数据按照以太网帧格式封装，包括目标MAC地址、源MAC地址、数据字段以及帧校验序列等信息。帧被发送到网络上，通过物理层的传输介质（如电缆、光纤等）传输到目标设备。接收设备通过物理层接收到帧后，检查帧的目标MAC地址。如果是本设备的地址，就将帧传递到链路层进行解封装，提取数据。如果不是本设备的地址，设备会忽略帧或进行转发。以太网使用CSMA/CD（Carrier Sense Multiple Access with Collision Detection），载波监听多点接入与

冲突检测) 协议, 确保多个设备可以在同一时间内共享传输介质。如果检测到冲突, 设备会执行退避算法, 随机等待一段时间后重新尝试发送。这种基于帧的通信方式和冲突检测机制使得以太网能够有效地支持局域网中的设备间通信。

下面以第一个 TCP 连接的以太网帧为例进行分析



目标地址 (Destination Address) 为 “HuaweiTe_c4:ec:e4 (5c:e8:83:c4:ec:e4)”, 该字段占用 6 个字节, 指示帧的目标设备的物理地址。

源地址 (Source Address) 为 “HewlettP_88:b0:2d (18:60:24:88:b0:2d)”, 该字段占用 6 个字节, 指示帧的发送设备的物理地址。

类型 (Type) 为 “IPv6 (0x86dd)”, 该字段占用 2 个字节, 标识了封装在以太网帧中的上层协议。在这里, 类型字段表明上层协议是 IPv6。

五、总结

1. 应用层协议的运行过程:

DNS 协议:

用户输入域名 `http://www.tsinghua.edu.cn`。

操作系统查询本地 DNS 缓存, 若无结果, 向本地 DNS 服务器发起查询。

本地 DNS 服务器递归查询, 最终获取到目标域名的 IP 地址。

操作系统将 IP 地址返回给浏览器。

HTTP 协议:

浏览器发起 HTTP 请求, 包含目标 URL 等信息。

服务器接收请求, 处理后返回 HTTP 响应。

浏览器解析响应, 获取 HTML 文件及其他资源。

2. 传输层协议的运行过程:

TCP 协议:

在 HTTP 层, 浏览器使用 TCP 协议与服务器建立连接, 进行可靠的数据传输。

初始阶段通过三次握手建立连接，后续进行数据传输。
数据分割为 TCP 段，按序发送到服务器。
服务器收到 TCP 段后发送确认应答。
连接维持期间，通过流量控制和拥塞控制调整传输速率。

3. HTTPS 协议的运行过程：

TLS/SSL 协议：

在 HTTP 协议的基础上，进行安全套接层的加密通信。
在 TCP 连接建立后，进行 TLS 握手，确保通信的安全性。
数据在传输前进行加密，保护隐私信息。

4. 网络层协议的运行过程：

IP 协议：

将数据包划分为 IP 数据包，添加源和目标 IP 地址。
数据包经过多个路由器和网络节点，每个节点决定下一跳的路径。
最终数据包到达目标主机。

DHCP 协议：

设备连接到网络时，发送 DHCP 请求，获取 IP 地址及其他网络配置信息。
DHCP 服务器分配未用的 IP 地址给设备，设备配置网络参数。

OSPF 协议：

如果有多个路径可选，路由器使用 OSPF 协议计算最短路径。
OSPF 协议确保路由表中包含到目标网络的最优路径。

5. 数据链路层协议的运行过程：

ARP 协议：

设备在链路层通过 ARP 协议将目标 IP 地址映射为物理 MAC 地址。
ARP 请求广播到网络，获取目标设备的 MAC 地址。
目标设备响应 ARP 请求，提供 MAC 地址。

以太网协议：

数据通过以太网帧封装，包含目标 MAC 地址和源 MAC 地址。
帧通过物理层传输介质到达目标设备。
目标设备通过物理层接收帧，检查目标 MAC 地址，进行解封装。
综合上述过程，从用户输入网址到网页显示，各层协议相互配合，确保了数据的可靠传输和网络通信的正常进行。