作者：刘雪松
联系方式：18115503264

车载Ethernet介绍

扫一扫二维码，关注我的视频号

虚拟仪器

微信扫描二维码，关注我的公众号

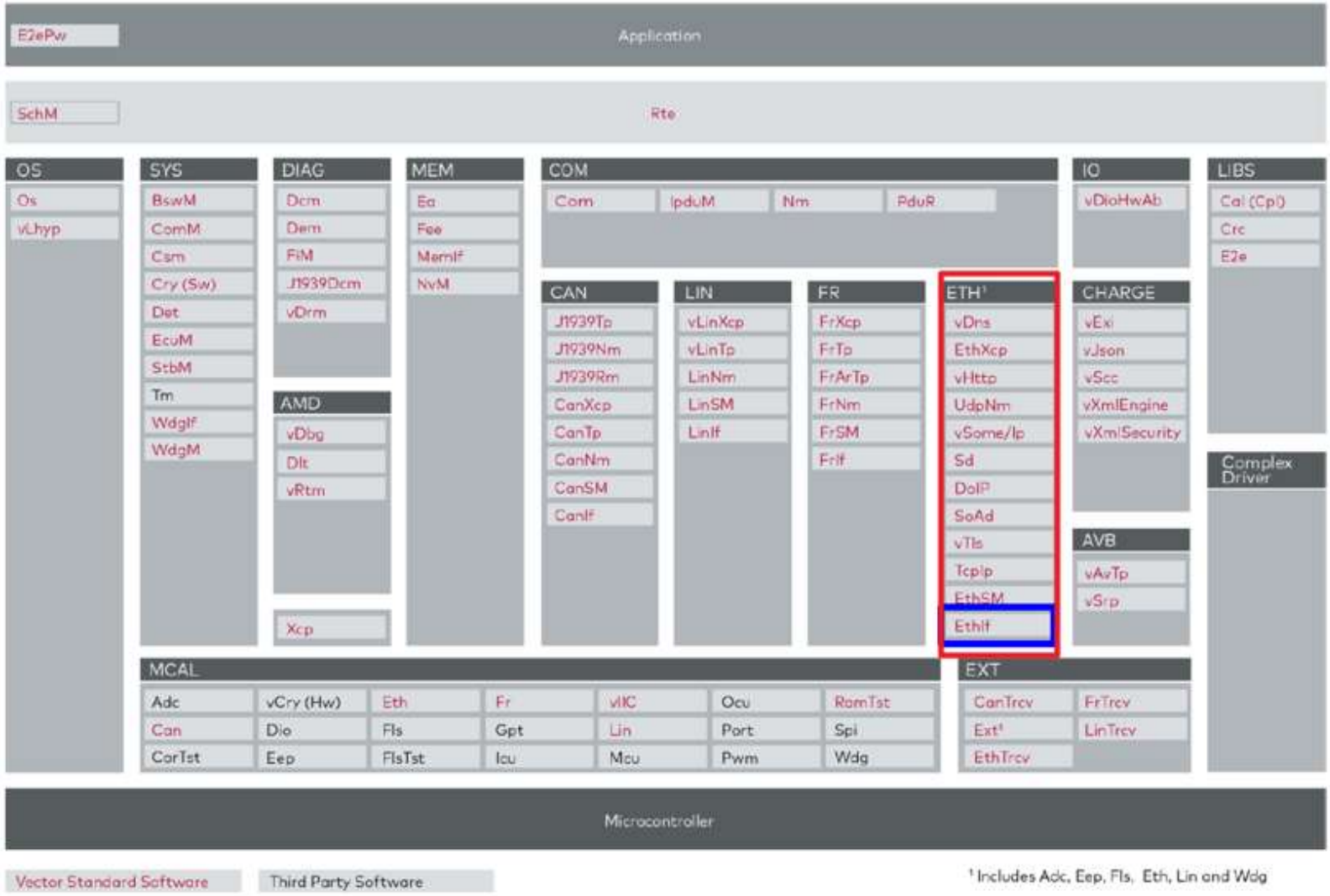
车载ethernet要素

- Ethernet优势
- Autosar架构中Ethernet
- 车载Ethernet中常用术语的关系

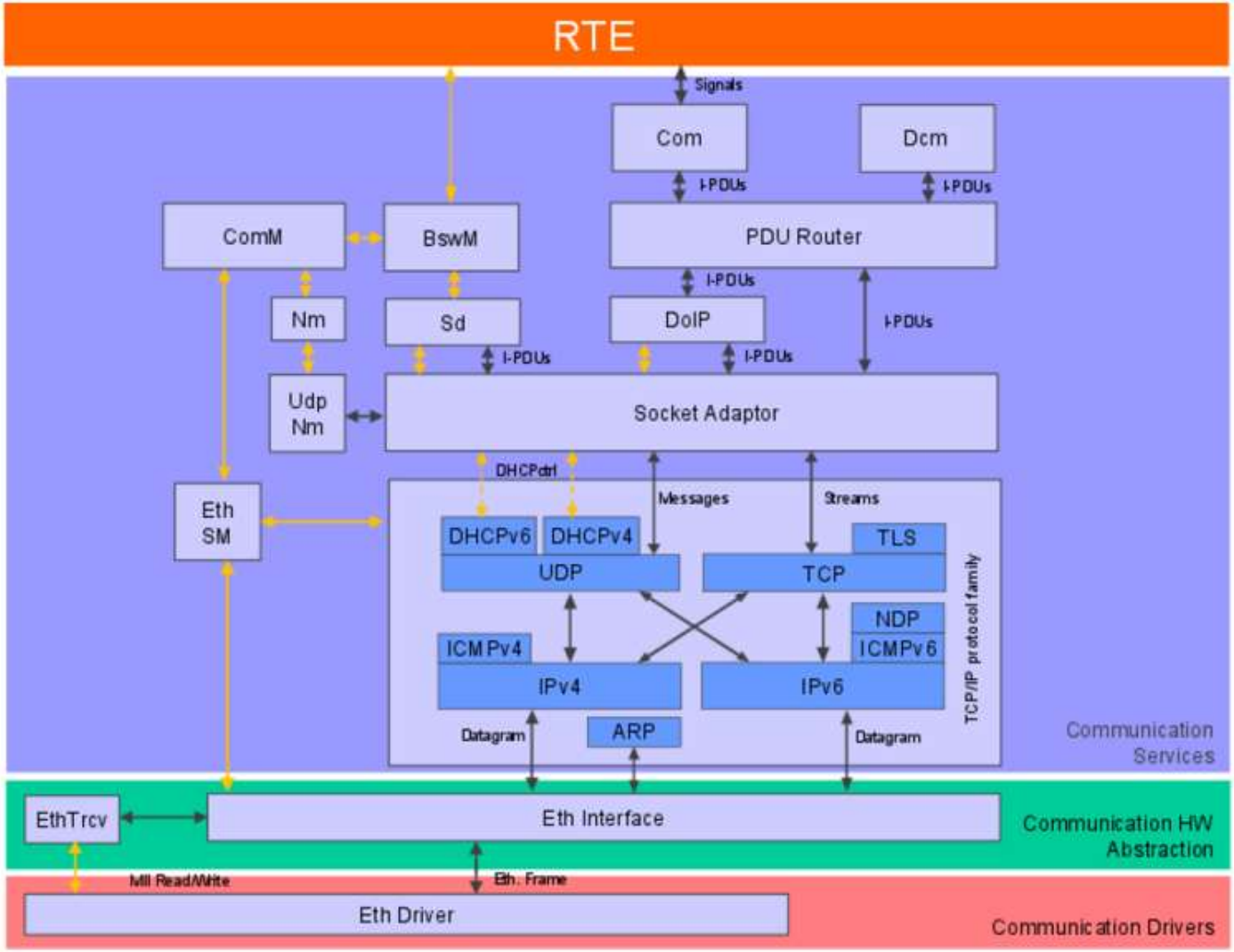
1.Ethernet优势

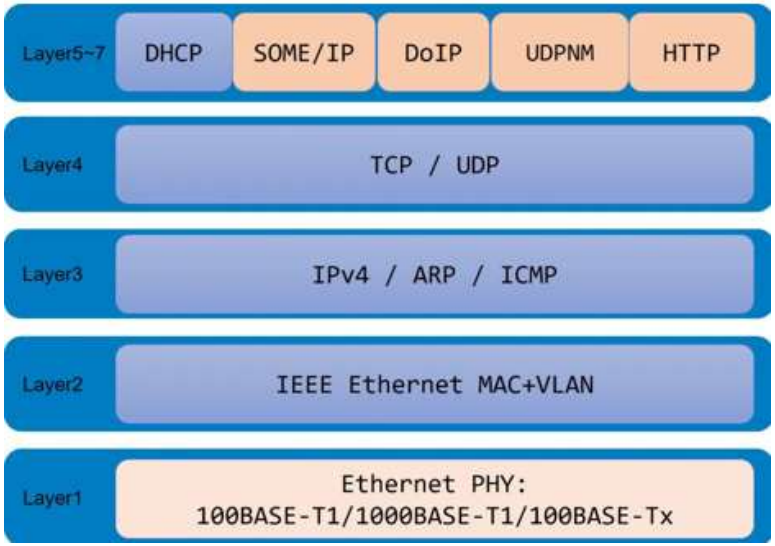
- Ethernet和传统的CAN/LIN/Flexray之类常用的汽车总线相比，提供了更高了带宽，适合汽车智能化发展趋势。
- Ethernet是成熟的，开放的总线模式，已经发展多年各类协议齐全，性价比适合做车载网络。
- Ethernet易拓展，随着汽车技术的不断发展，汽车以太网提供了可扩展的解决方案，可以满足未来的数据需求和新功能。
- 数据优先，以太网可以支持服务质量（QoS）机制，允许关键的安全相关数据优先于不太关键的信息，确保基本功能优先；

2.Autosar架构中Ethernet

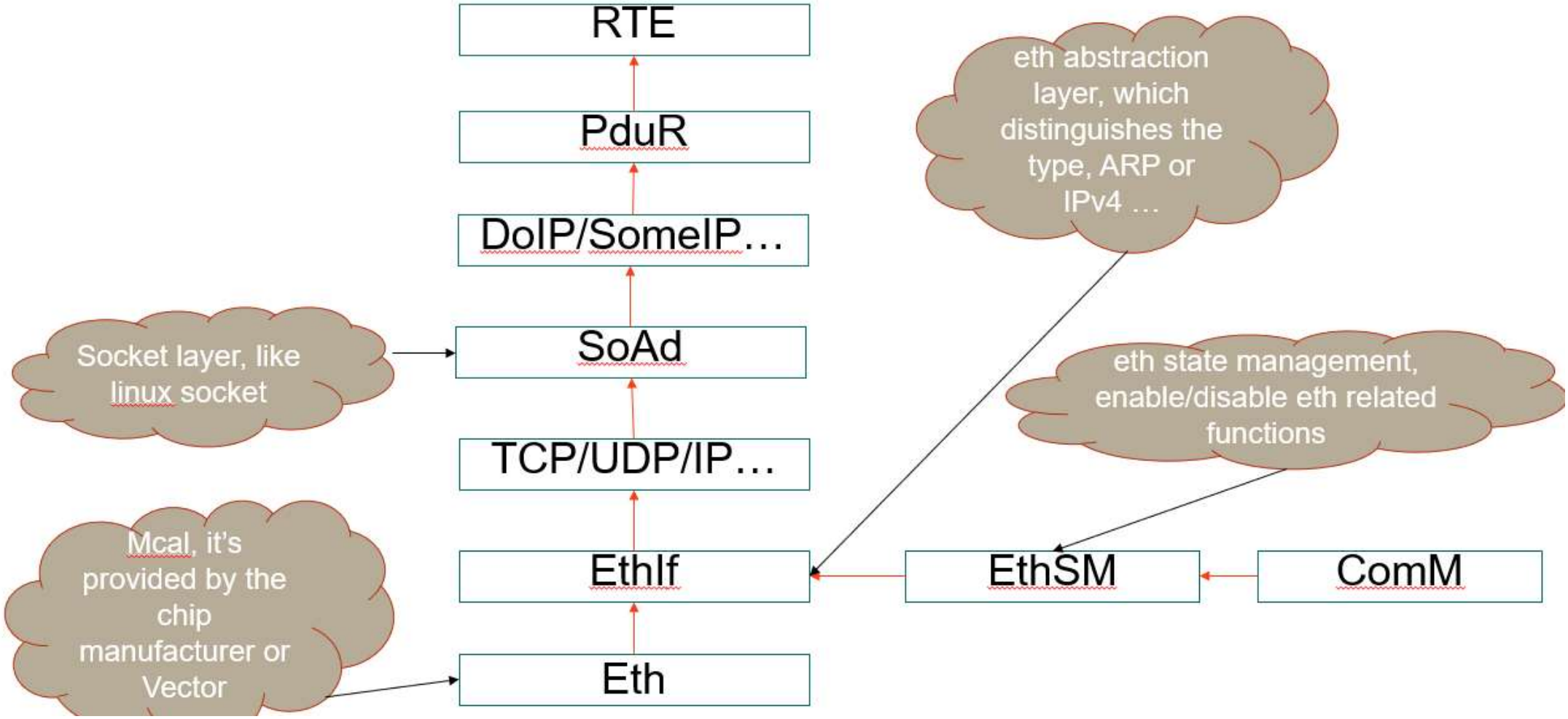


2.Autosar架构中Ethernet

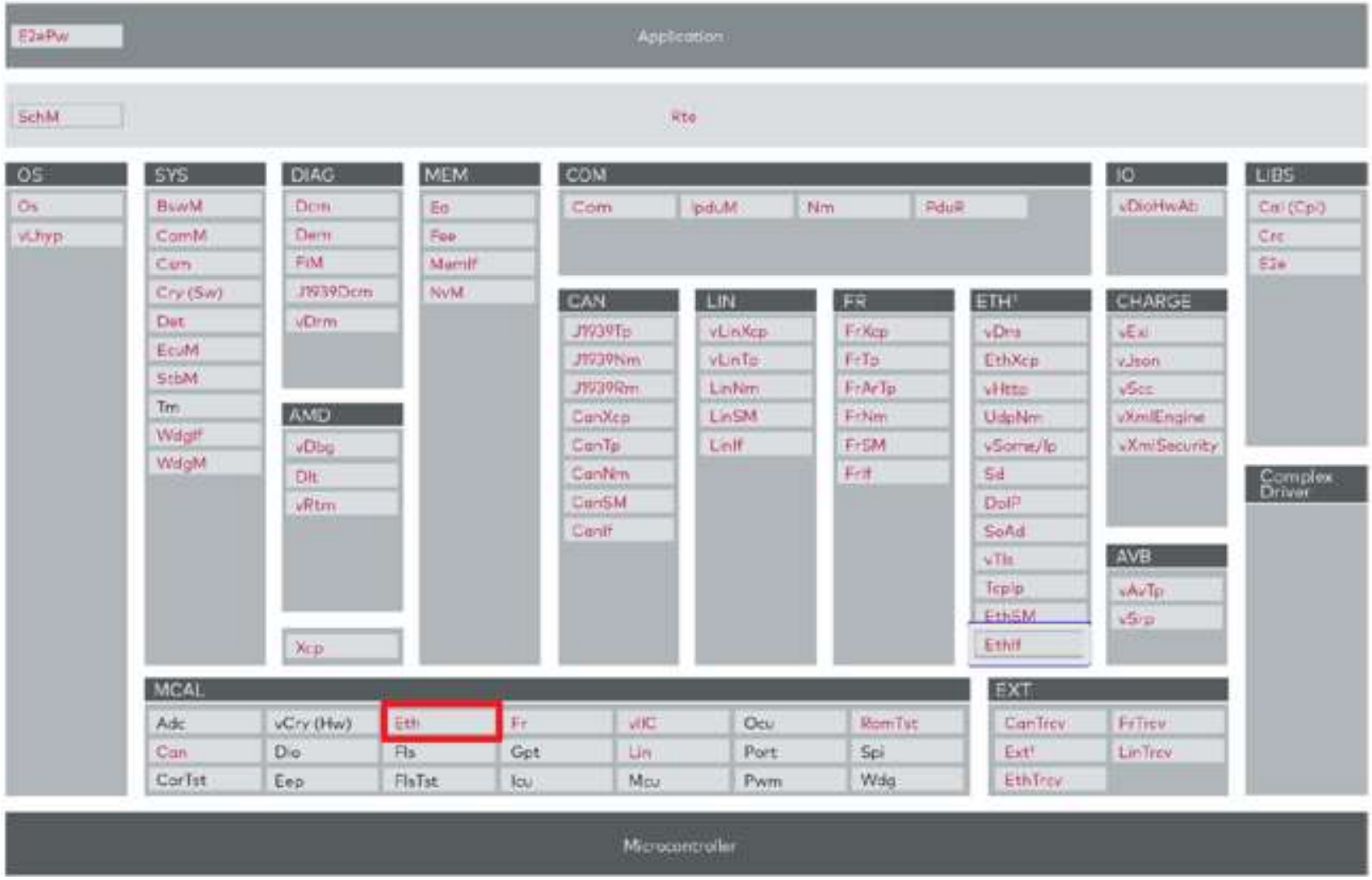




3.车载Ethernet中常用术语的关系

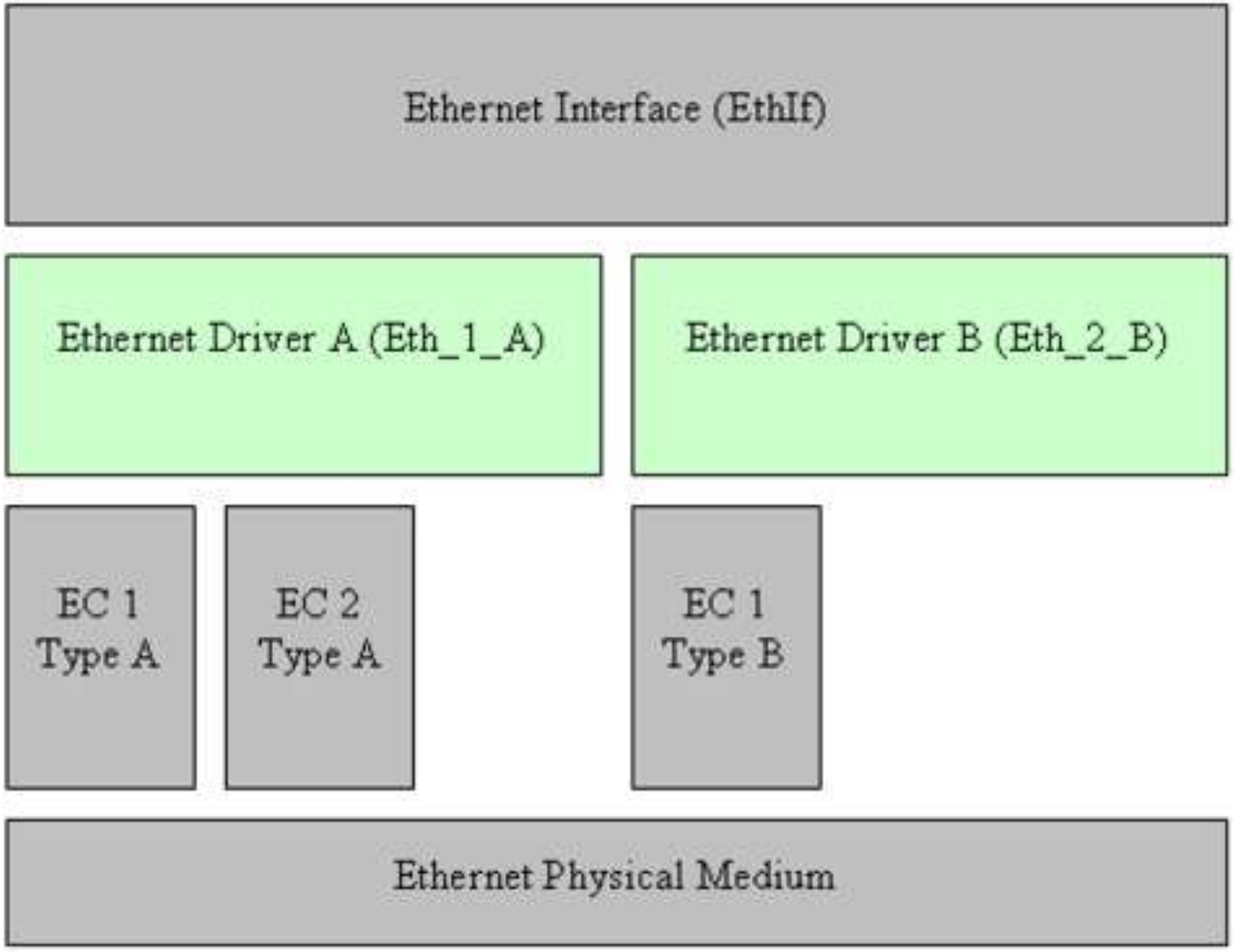


3.1ETH底层驱动



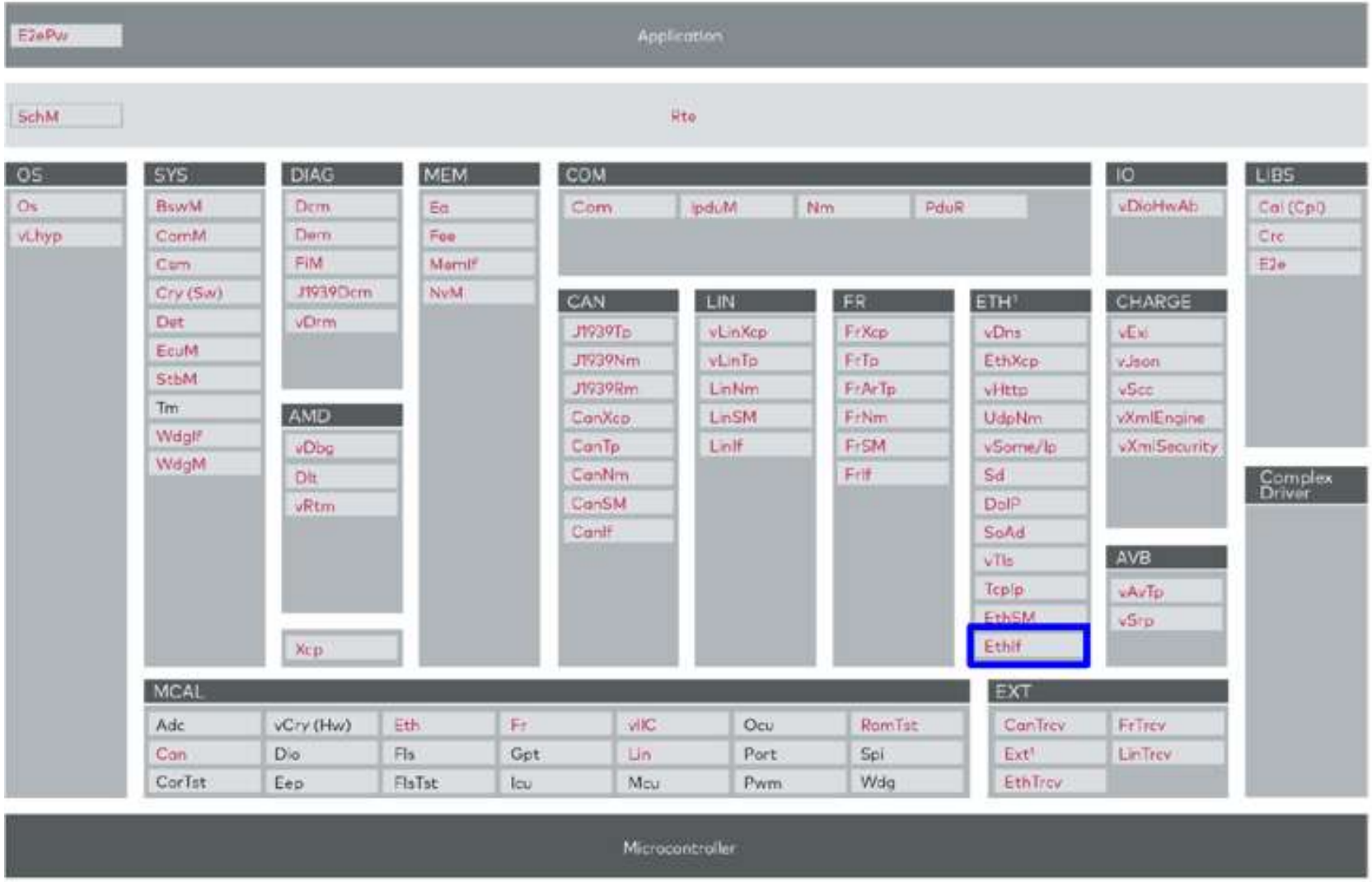
1.车载Autosar中ETH驱动最接近底层，由芯片供应商提供最新的底层驱动。
2.底层只负责解析出数字信号，和适配总线速率（100/1000）

3.2 ETHIF/ETHSM



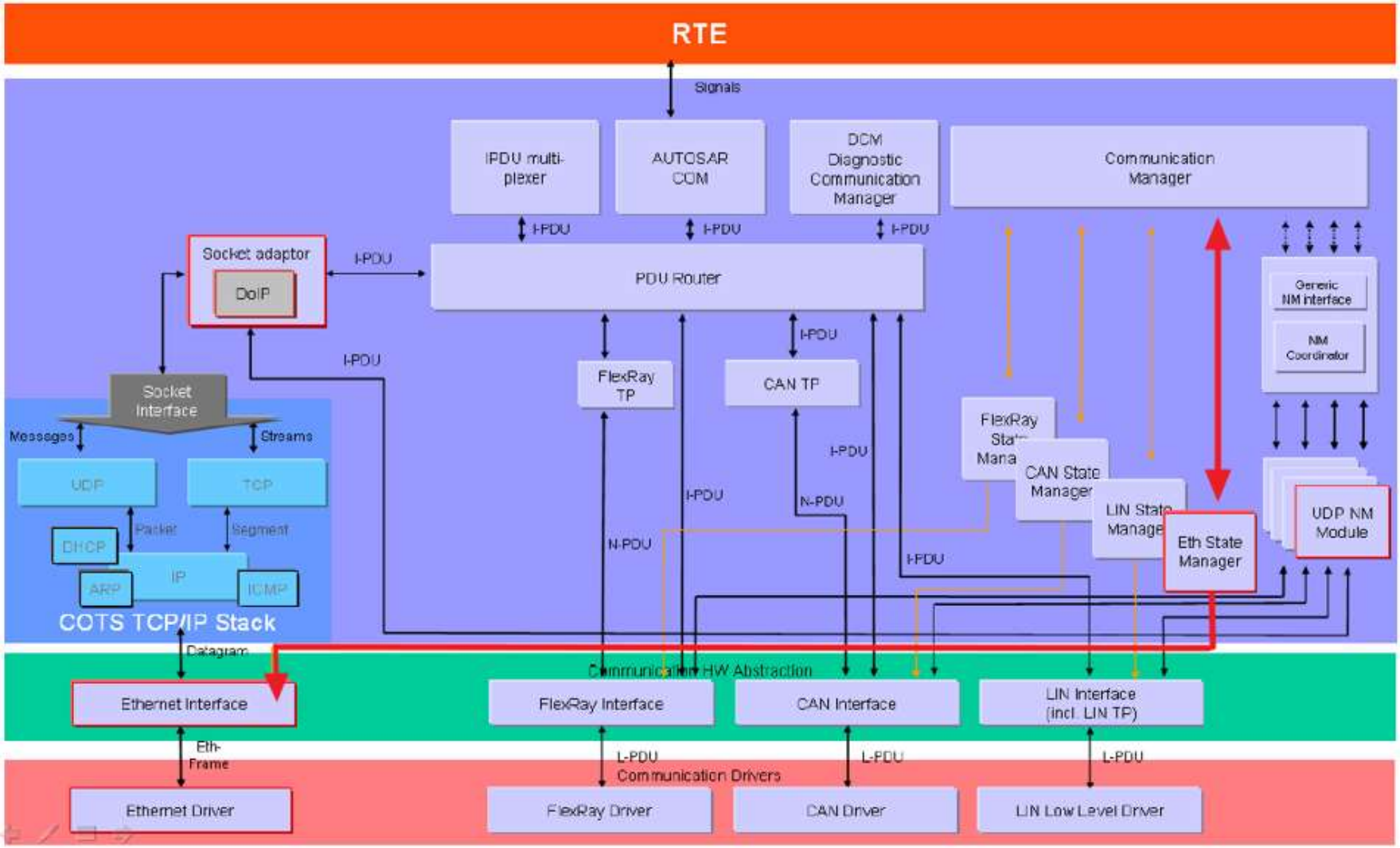
1.单个ETH驱动程序仅支持一个类型的控制器硬件，但是能支持多个同类型控制器。
2. MTU = Destination MAC Address (6 bytes) + Source MAC Address (6 bytes) + EtherType (2 bytes) + VLAN Tag (4 bytes) + IP header (20 bytes) + TCP header (20 bytes) + MSS(1460bytes) + Frame Check Sequence (FCS) or CRC (4 bytes)，每帧数据最大1522BYTE

3.2 ETHIF

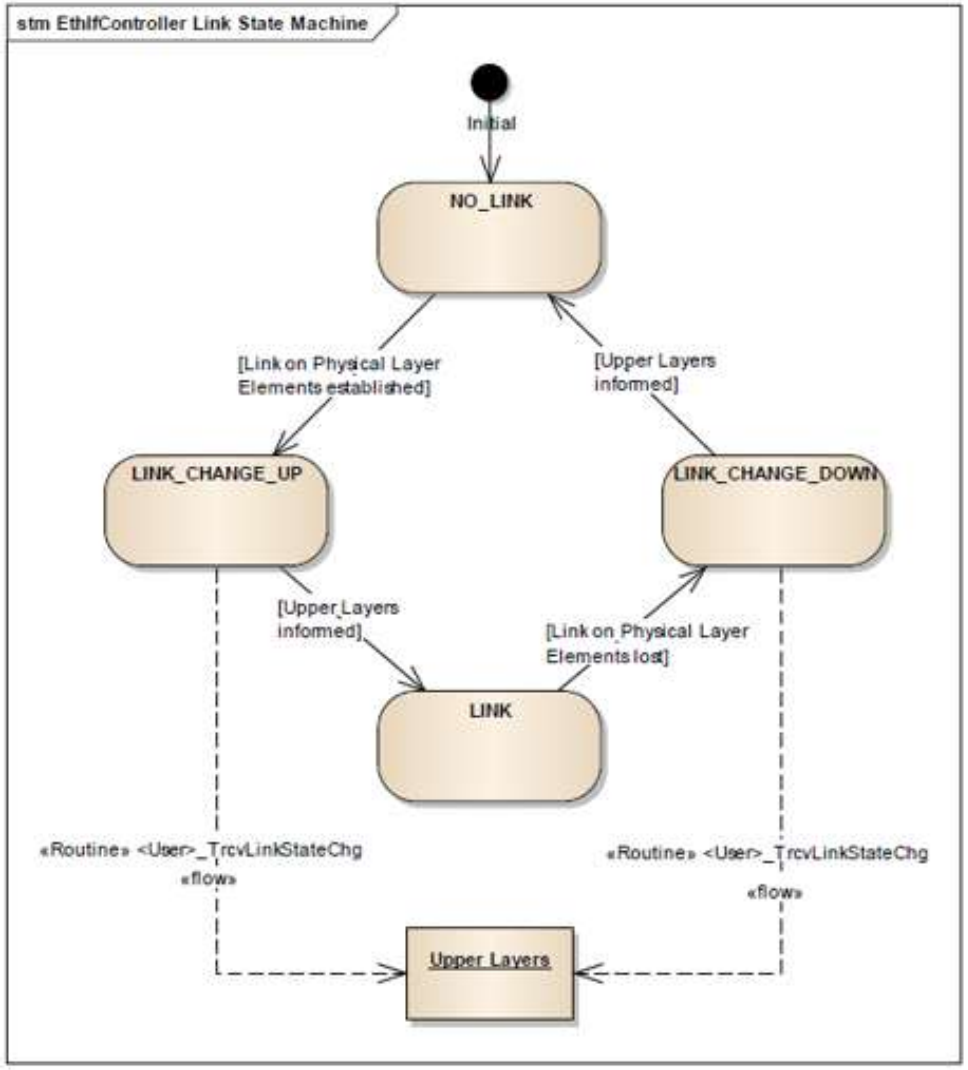


1.Ethif是ETH的抽象层，ETHif不会直接访问硬件避免硬件更换带来的问题。ETHIF只对ETH每一帧进行分类，比如区分ARP帧，IPV4/6帧
2.ETHIF可以实现模块的二层交换机功能

3.2 ETHSM在AUTOSAR位置和作用



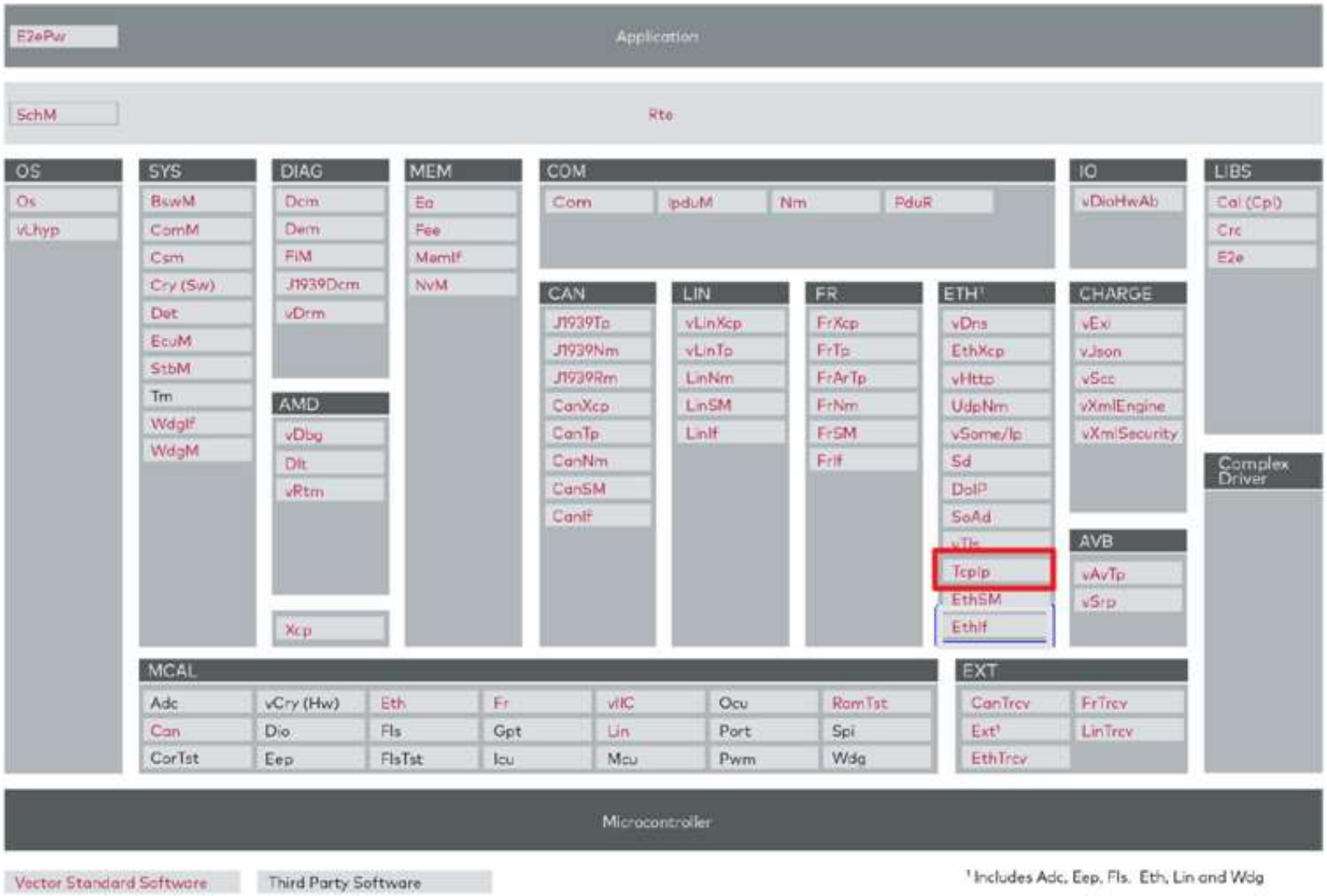
3.2 ETHSM



1.EthSM主要用来管理和监控ETH通信状态

- ETHSM在模块上电时初始化ETH硬件，包括配置和激活接口硬件
- EthSM 模块监控和管理以太网通信的不同状态。这些状态可能包括初始化、通信中、故障等。
- ETHSM可能会监测以太网链路的状态，以确定网络是否处于连接状态。
- ETHSM负责管理以太网接口的配置。这包括分配 IP 地址、子网掩码和其他网络参数。确保正确的配置对于网络通信的成功至关重要。
- EthSM 模块通常包括错误检测和处理机制，以应对网络问题。它可以检测和报告网络故障，然后采取适当的措施，例如重新配置网络接口或通知系统管理员。

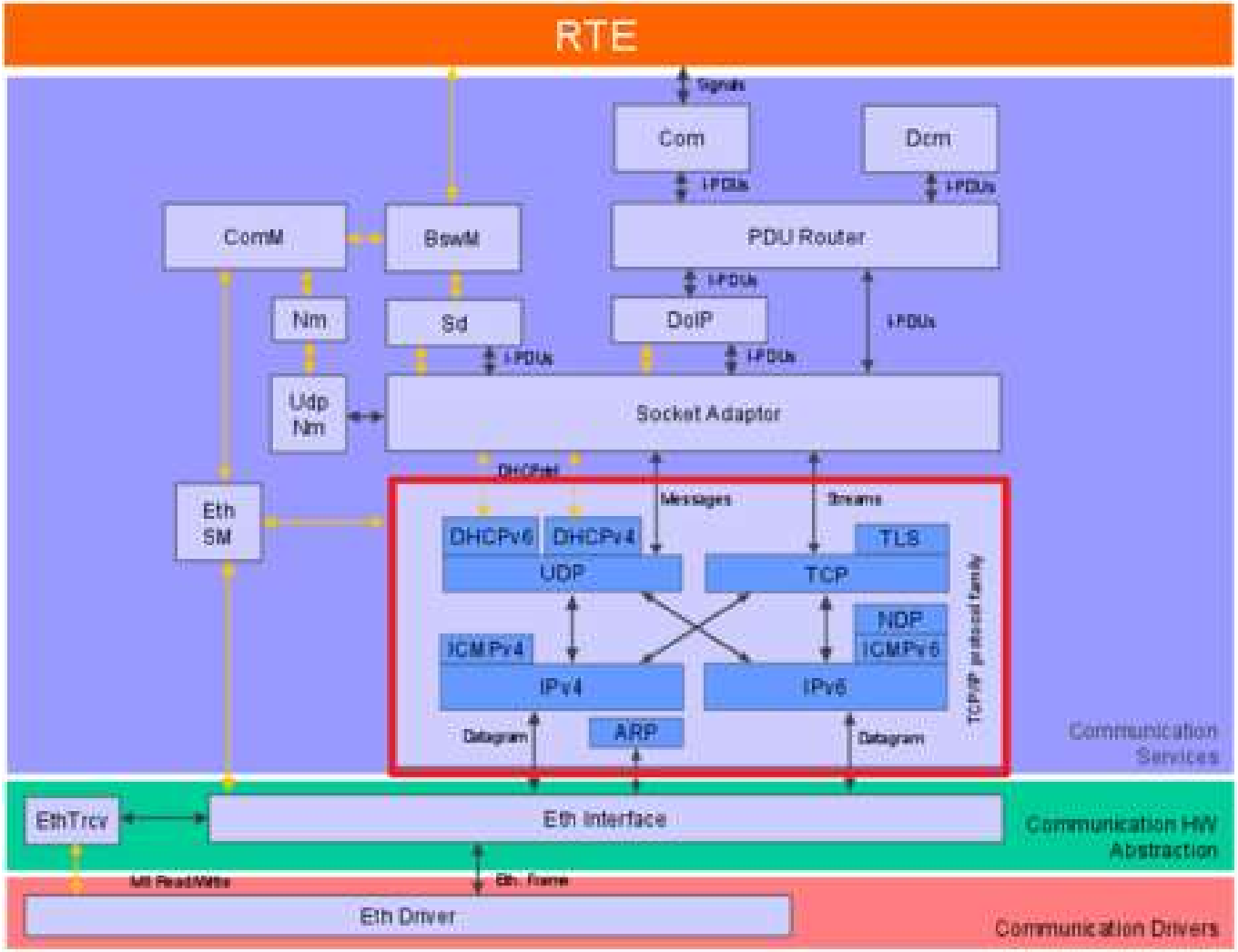
3.4 TCP/IP模块



1.TCP/IP模块在autosar的位置

2.相当于传输层，介于数据链路层和传输层之间

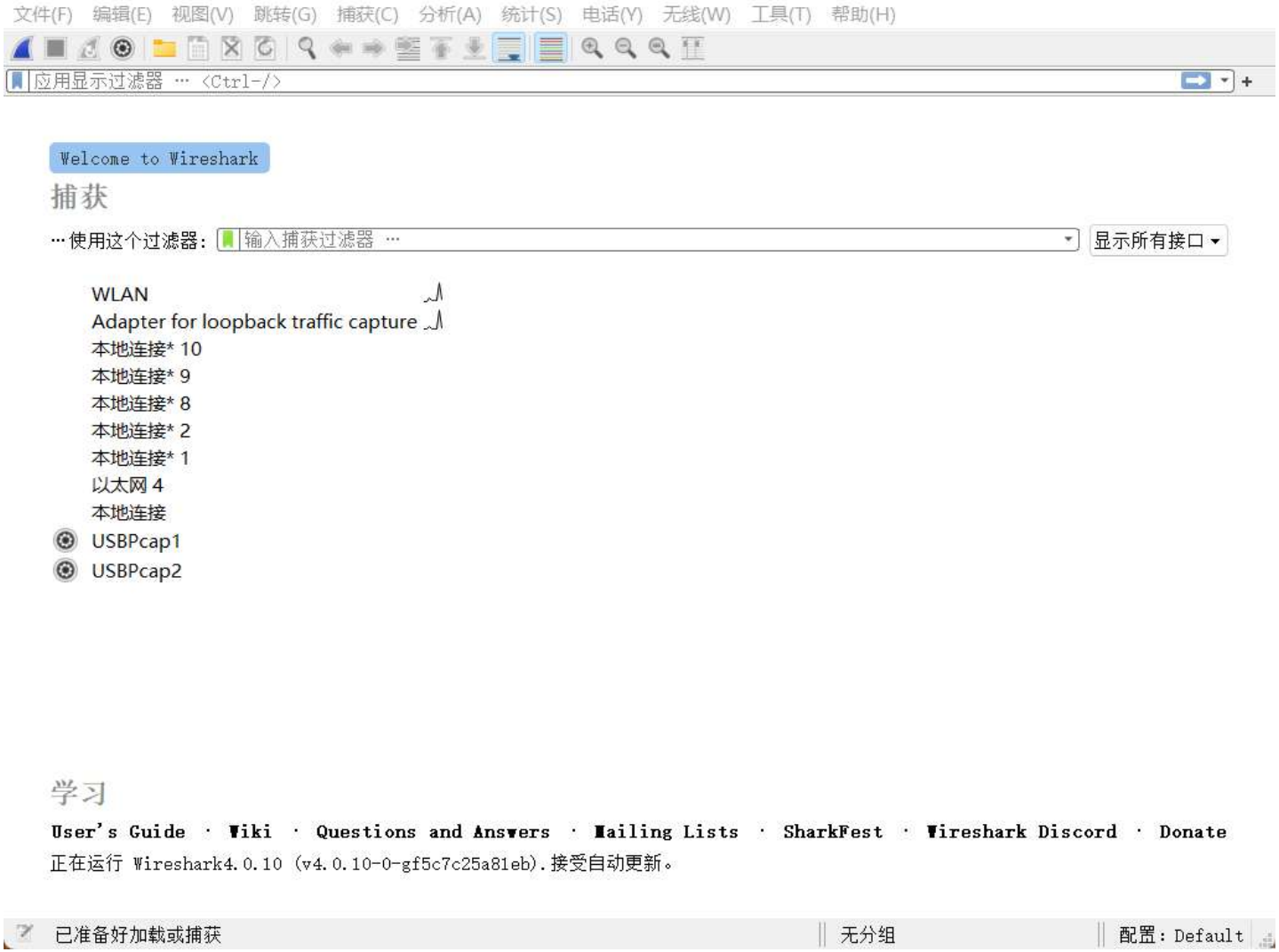
3.4 TCP/IP模块



1.TCP/IP模块在autosar的位置

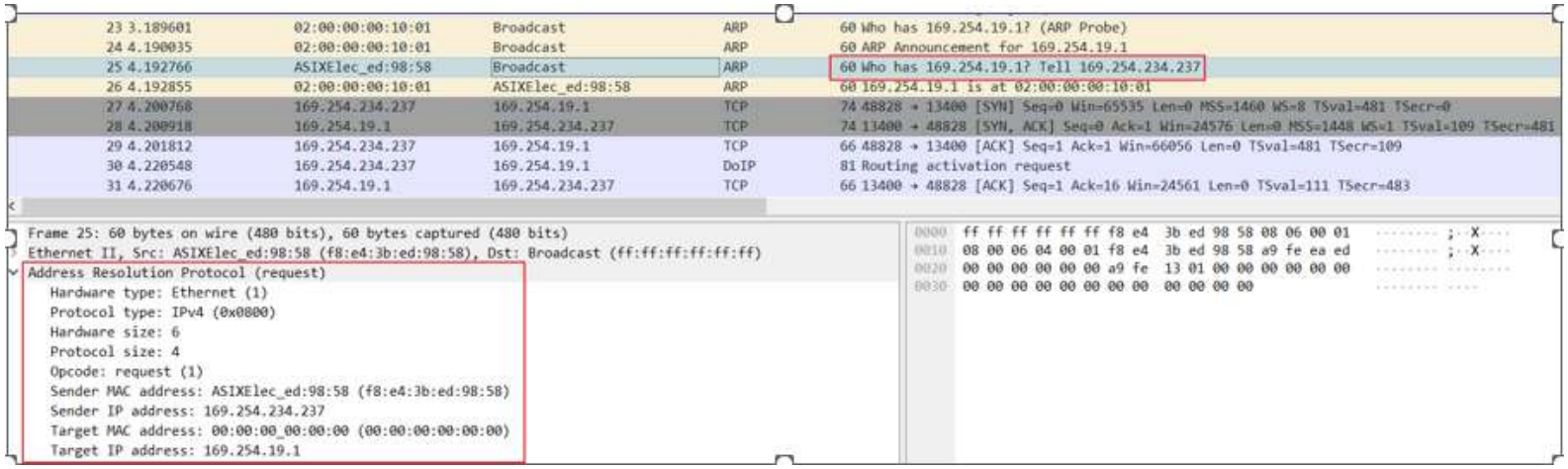
2.相当于传输层，介于数据链路层和传输层之间

NOTE:常用ETH工具---wireshark

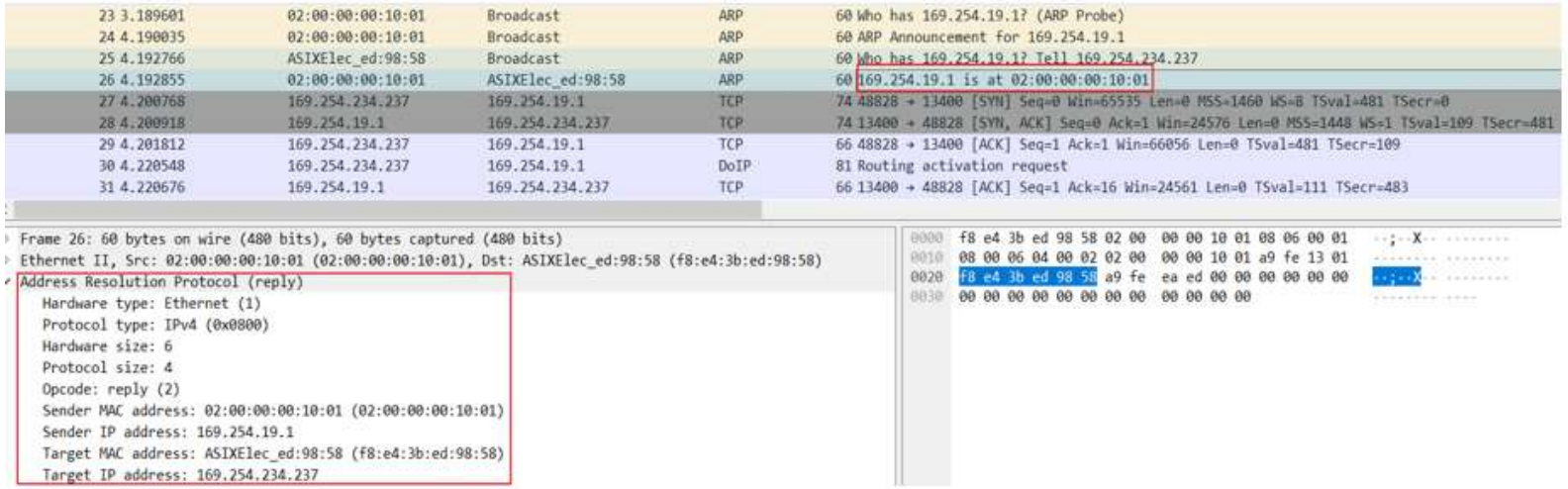


- Wireshark（前称Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是截取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换。
- 后续ETH以wireshark程序为例

3.4.1 ARP报文

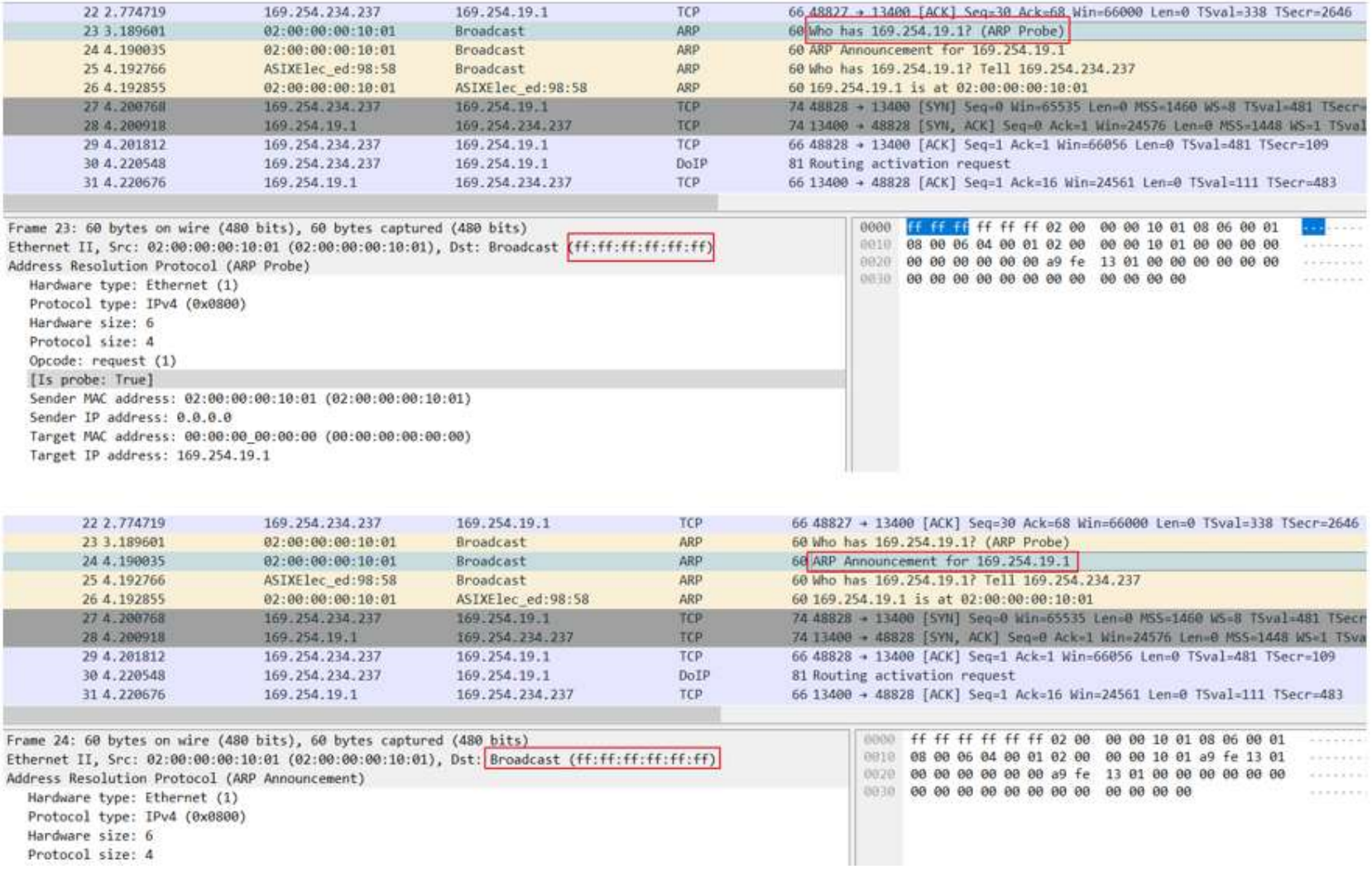


1.ARP探测是网络中的一种探测和验证机制，用于保证IP地址分配的准确性和网络设备之间的通信有效性。在探测期间，设备会广播一个ARP请求，等待其他设备的回复，以确定是否存在IP地址冲突或者验证IP地址的可用性。如果其他设备回复了ARP请求，设备会根据回复结果来采取适当的行动，例如选择一个未被使用的IP地址或解决冲突问题。



2. ARP通告是一种广播消息，会在整个本地网络中传播。因此，网络中的所有设备都会收到这些通告消息。在ARP通告期间，其他设备会更新它们的ARP缓存，以反映设备的新IP地址和对应的MAC地址。这样，网络中的所有设备就可以知道彼此的新地址，确保网络通信的正常运作。

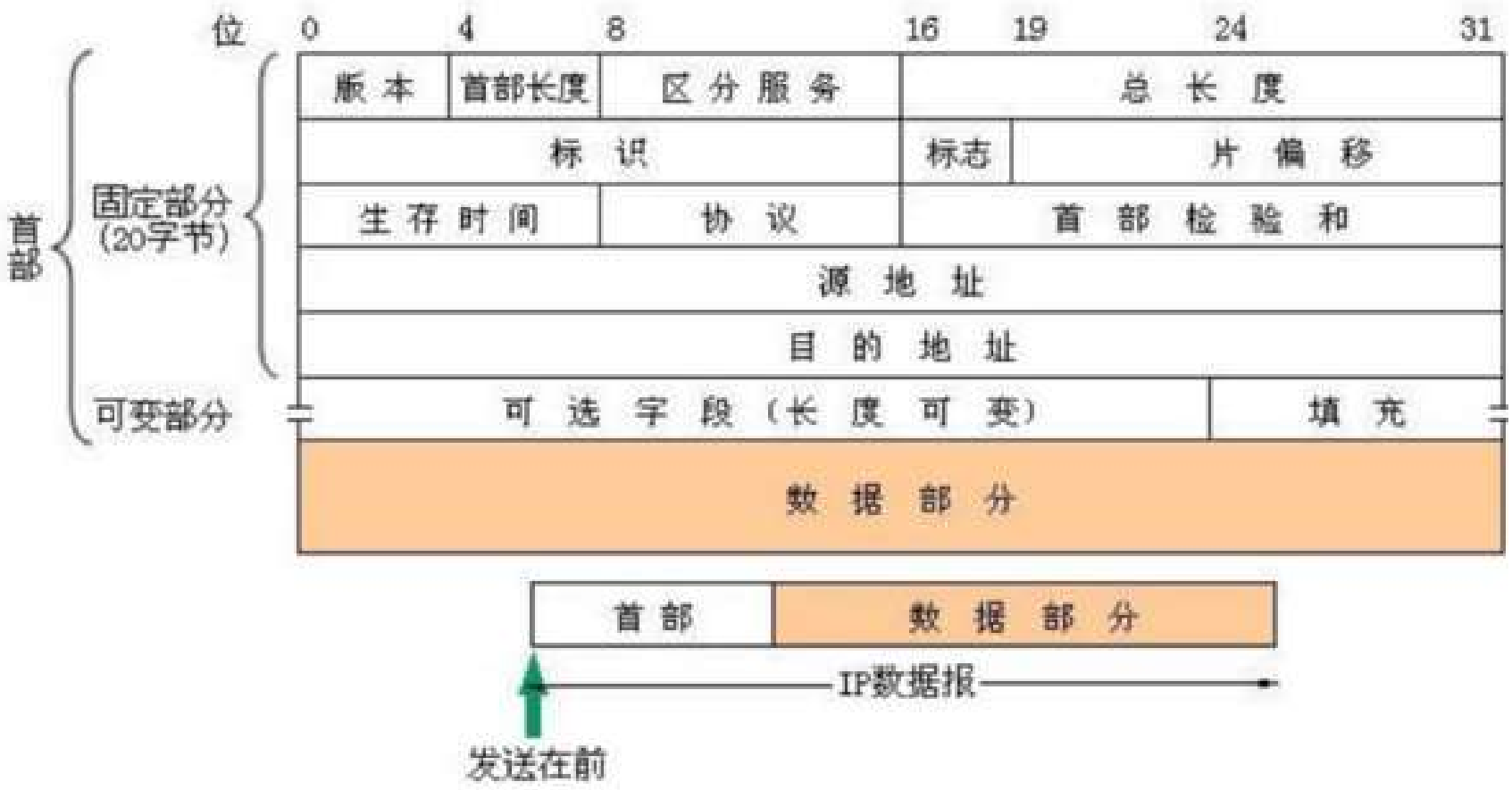
3.4.2 自动分配IP地址(AUTOIP)



3.4.2 自动分配IP地址(AUTOIP)

AutoIP（Automatic Private IP Addressing）是一种用于自动分配IP地址的协议，通常用于局域网（LAN）中，特别是在没有DHCP服务器可用的情况下。
以下是AutoIP的分配过程：
1.设备启动：设备在网络上启动并连接到LAN，但没有可用的DHCP服务器来分配IP地址。
2.IP地址请求：设备检测到没有可用的DHCP服务器后，它会尝试使用AutoIP来分配一个私有的IP地址。设备会广播一个ARP（Address Resolution Protocol）请求，其中包含特殊的IP地址范围，通常为169.254.0.0/16。
3.竞争检查：如果多个设备在同一时刻尝试分配AutoIP地址，它们可能会发生IP地址冲突。为了解决这个问题，设备会随机等待一段时间，然后再次广播ARP请求。如果这个IP地址已经被另一个设备分配，那么它将收到ARP响应，指示该地址已被占用，需要重新选择一个IP地址。
4.IP地址分配：一旦设备找到一个未被使用的AutoIP地址，它将自己分配这个IP地址。这个IP地址通常是169.254.x.x的形式，其中x是一个随机选择的数值。设备还会设置子网掩码为255.255.0.0，这意味着设备可以与同一子网内的其他AutoIP设备通信，但不能跨子网通信。
5.本地通信：一旦设备成功分配了AutoIP地址，它可以与同一子网内的其他设备进行本地通信。这种通信方式通常用于发现和配置网络服务，以便设备能够连接到网络或执行本地任务

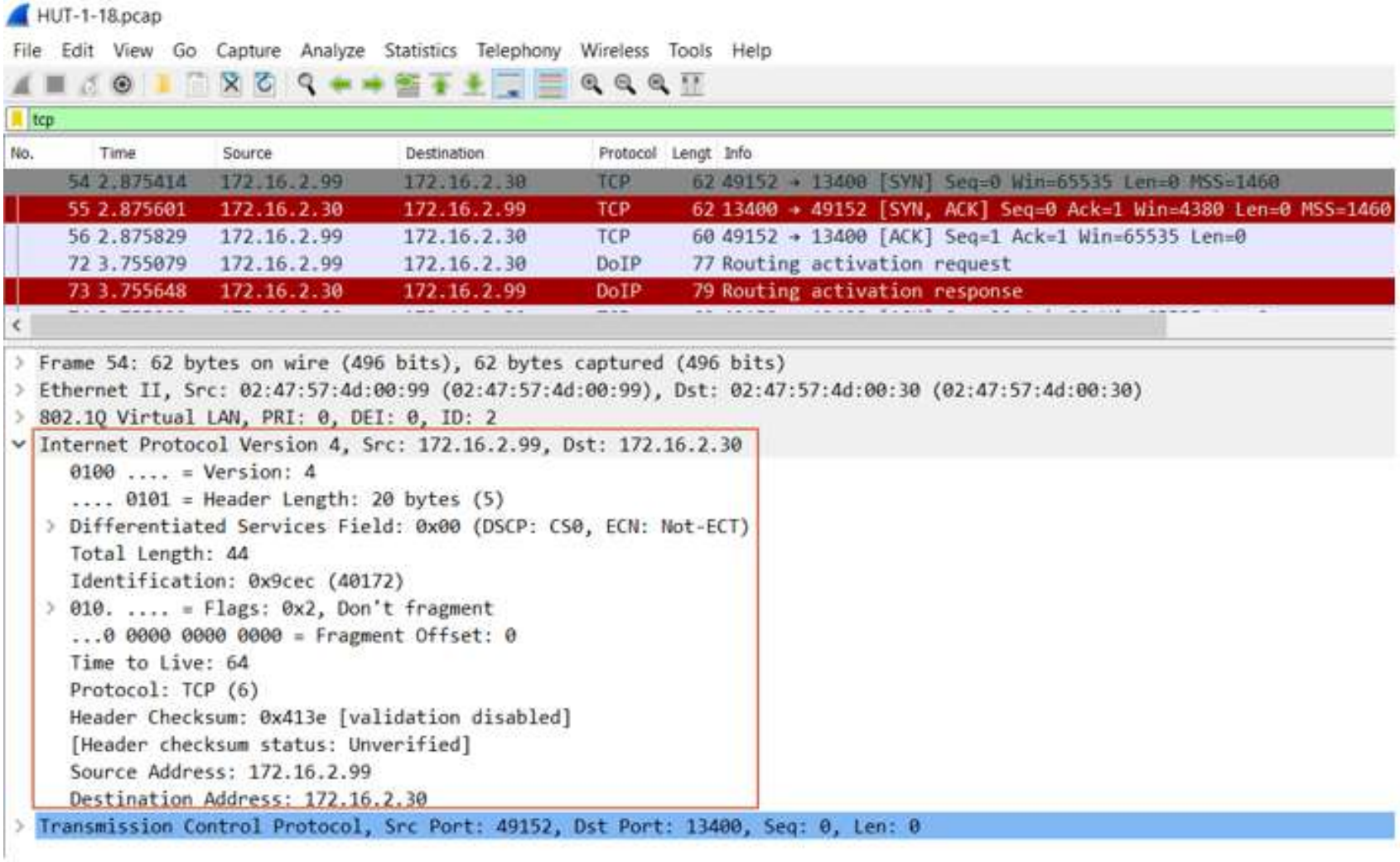
3.4.3 IP报文结构



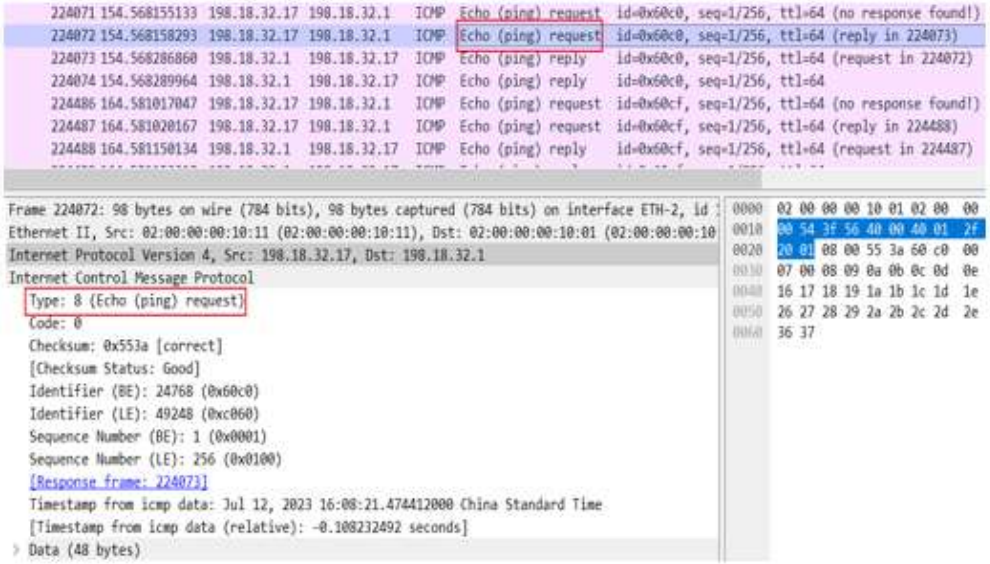
IP为每个连接到网络的设备提供一个唯一的地址，称为IP地址，允许设备之间相互发送和接收数据。

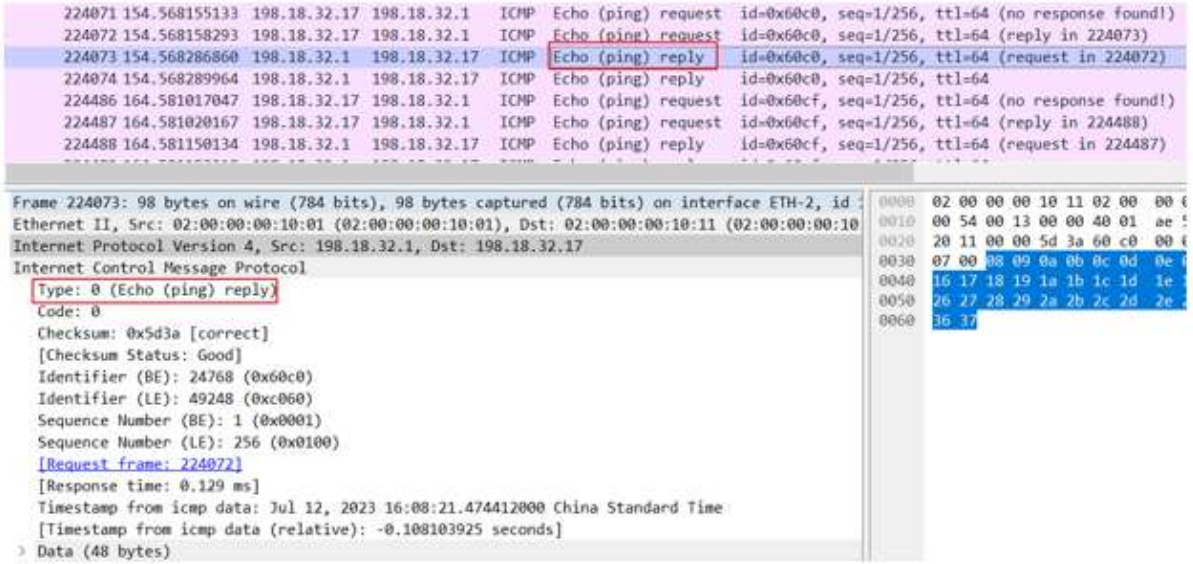
IP协议负责将数据分割成数据包，并为每个数据包添加必要的头部信息，以指定源IP地址和目标IP地址。当一台设备想要与同一网络或另一个网络的设备通信时，它将数据封装成IP数据包并发送到目标IP地址。

3.4.3 IP报文结构


HUT-1-18.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Lengt Info
54 2.875414 172.16.2.99 172.16.2.30 TCP 62 49152 → 13400 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
55 2.875601 172.16.2.30 172.16.2.99 TCP 62 13400 → 49152 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 MSS=1460
56 2.875829 172.16.2.99 172.16.2.30 TCP 60 49152 → 13400 [ACK] Seq=1 Ack=1 Win=65535 Len=0
72 3.755079 172.16.2.99 172.16.2.30 DoIP 77 Routing activation request
73 3.755648 172.16.2.30 172.16.2.99 DoIP 79 Routing activation response
Frame 54: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Ethernet II, Src: 02:47:57:4d:00:99 (02:47:57:4d:00:99), Dst: 02:47:57:4d:00:30 (02:47:57:4d:00:30)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 2
Internet Protocol Version 4, Src: 172.16.2.99, Dst: 172.16.2.30
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 44
Identification: 0x9cec (40172)
010. .... = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x413e [validation disabled]
[Header checksum status: Unverified]
Source Address: 172.16.2.99
Destination Address: 172.16.2.30
Transmission Control Protocol, Src Port: 49152, Dst Port: 13400, Seq: 0, Len: 0

3.4.4 ICMP报文





ICMP主要用于以下几个方面：

1.错误报告：当网络设备（例如路由器或主机）处理IP数据包时遇到问题时，它可能会生成ICMP错误消息，向数据包的发送方报告问题。

2.Ping和Traceroute：ICMP被用于"ping"命令，该命令发送ICMP回显请求消息来检查远程主机是否可达，并测量数据包往返的往返时间（RTT）。

3.Path MTU Discovery：ICMP用于发现两个主机之间的最大传输单元（MTU）。路径MTU发现有助于避免大数据包的分段，提高网络效率。

4.重定向消息：路由器可以使用ICMP重定向消息通知主机有关特定目标通过其他网关有更好的路由可用。

5.超时消息：当数据包的生存时间（TTL）字段为零时，表示数据包在网络中存在的时间太长，路由器可以向源发送方发送ICMP超时消息。

3.4.5 UDP协议

UDP datagram header

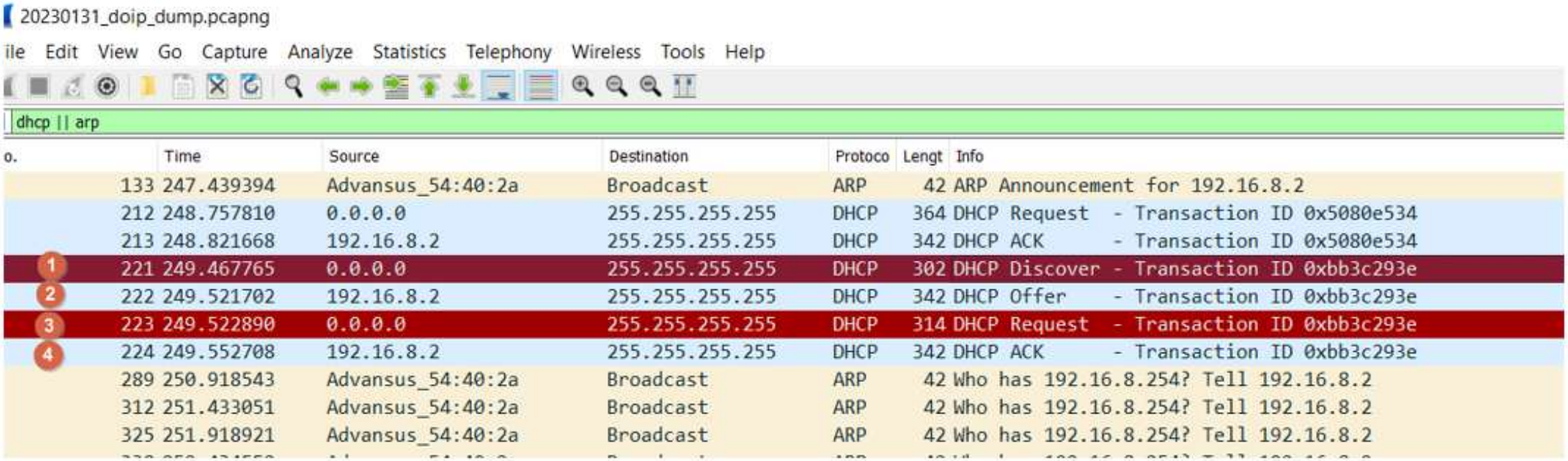
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

- UDP协议的主要特点包括：

1. 无连接：与TCP不同，UDP在发送数据之前不建立连接。每个UDP数据包都是独立的，并包含所有必要的信息，包括源IP地址和目标IP地址以及端口号。
2. 不可靠：UDP不提供错误检查或丢包重传功能。一旦UDP数据包被发送，发送方不等待接收方的确认。如果在传输过程中出现数据包丢失或损坏，就没有机制来恢复它。
3. 低开销：UDP的简单性使得它具有较低的开销，与TCP相比。无需建立和维护连接，并且控制机制较少，这对某些实时应用程序可能有益处。

- UDP通常用于对实时数据传输至关重要且可以容忍某些数据包丢失的应用程序。一些UDP使用示例包括：
- 实时音视频流传输（例如VoIP应用程序，实时视频流传输）。
- 在线游戏，其中低延迟至关重要。
- DHCP（动态主机配置协议），用于动态获取IP地址。

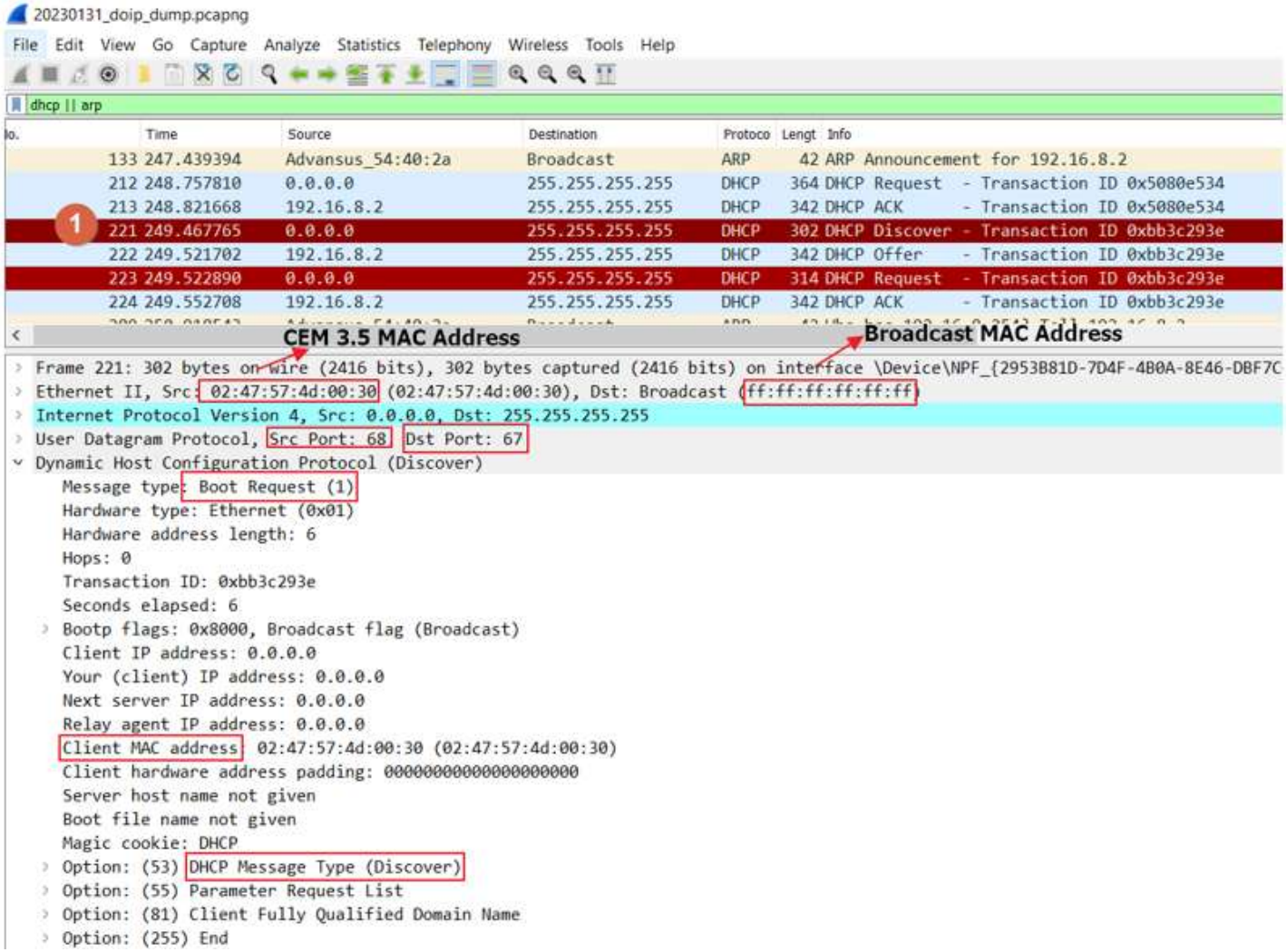
3.4.6 DHCP



- DHCP的主要功能包括：

1. 自动IP地址分配：当设备加入网络时，它可以向DHCP服务器请求一个IP地址。DHCP服务器会从预定义的地址池中动态分配一个可用的IP地址给请求的设备。
2. IP地址租约：由DHCP服务器分配的IP地址并不是永久的，而是租借给设备一段特定的时间（租约时间）。在租约到期之前，设备可以续约租约，如果不再需要该IP地址，它将释放该地址，供其他设备未来使用。
3. 子网掩码和默认网关分配：除了分配IP地址外，DHCP服务器还提供子网掩码和默认网关信息，使设备能够确定其所在网络的边界，并知道如何将数据流量发送到局域网之外的目标。
4. DNS服务器配置：DHCP还可以提供域名系统（DNS）服务器的IP地址，使设备能够将域名转换为IP地址，实现与互联网的通信。
5. 其他网络配置参数：DHCP可以传递其他配置信息，如域名、时间服务器和其他相关设置，以帮助设备完成网络配置。

3.4.6 DHCP工作原理

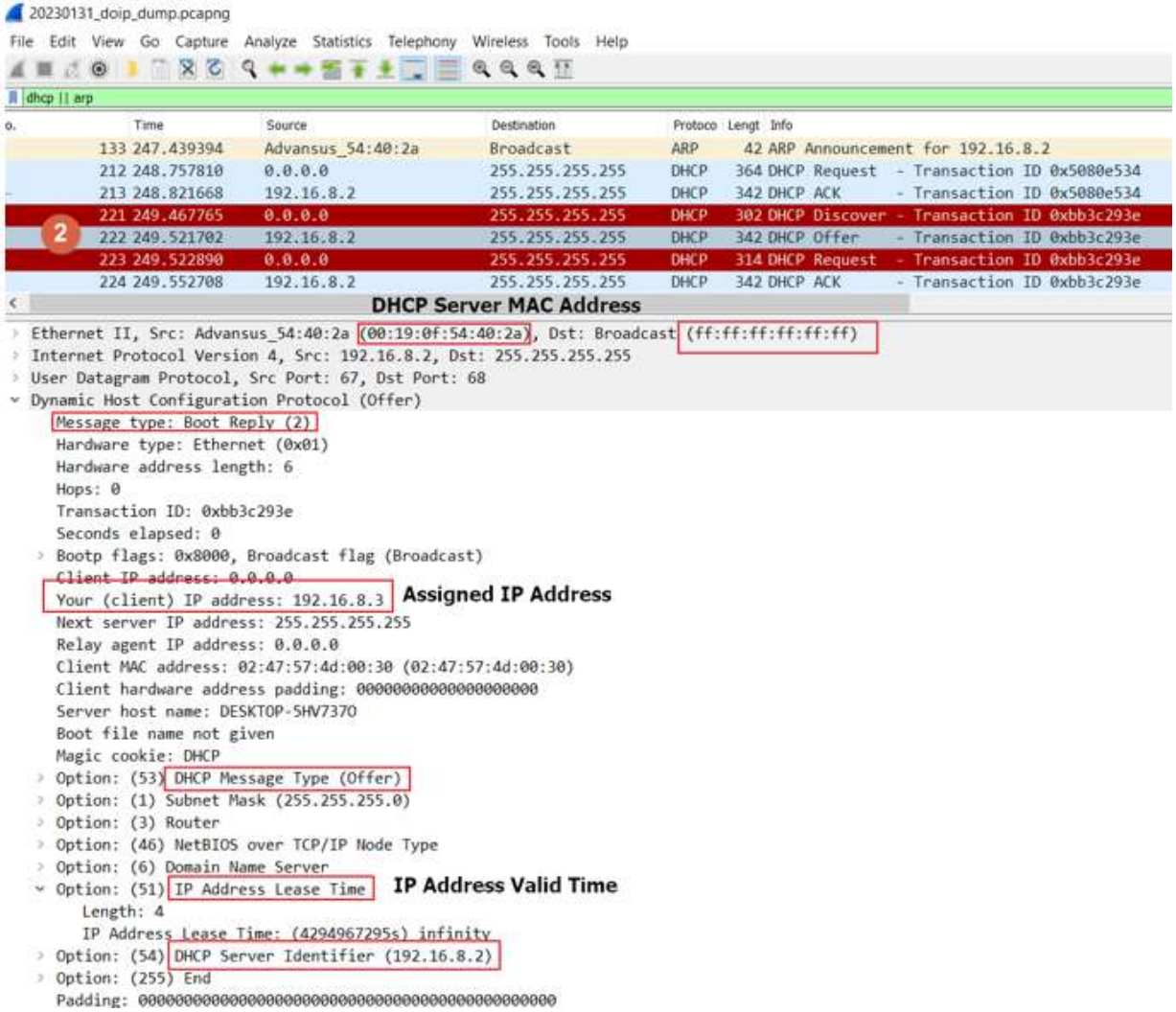


DHCP有四个阶段

1、发现阶段
客户端广播向所有服务器发送DHCP discover报文。（此时客户端DHCP请求使用的端口为UDP 68）

发现阶段的客户端是不知道服务器的IP地址的所以会广播发送DHCP discover报文寻找DHCP服务器，目标地址为255.255.255.255，只要在这个网络中的IP地址的服务器都可以接收到DHCP客户端的IP地址请求。

3.4.6 DHCP工作原理

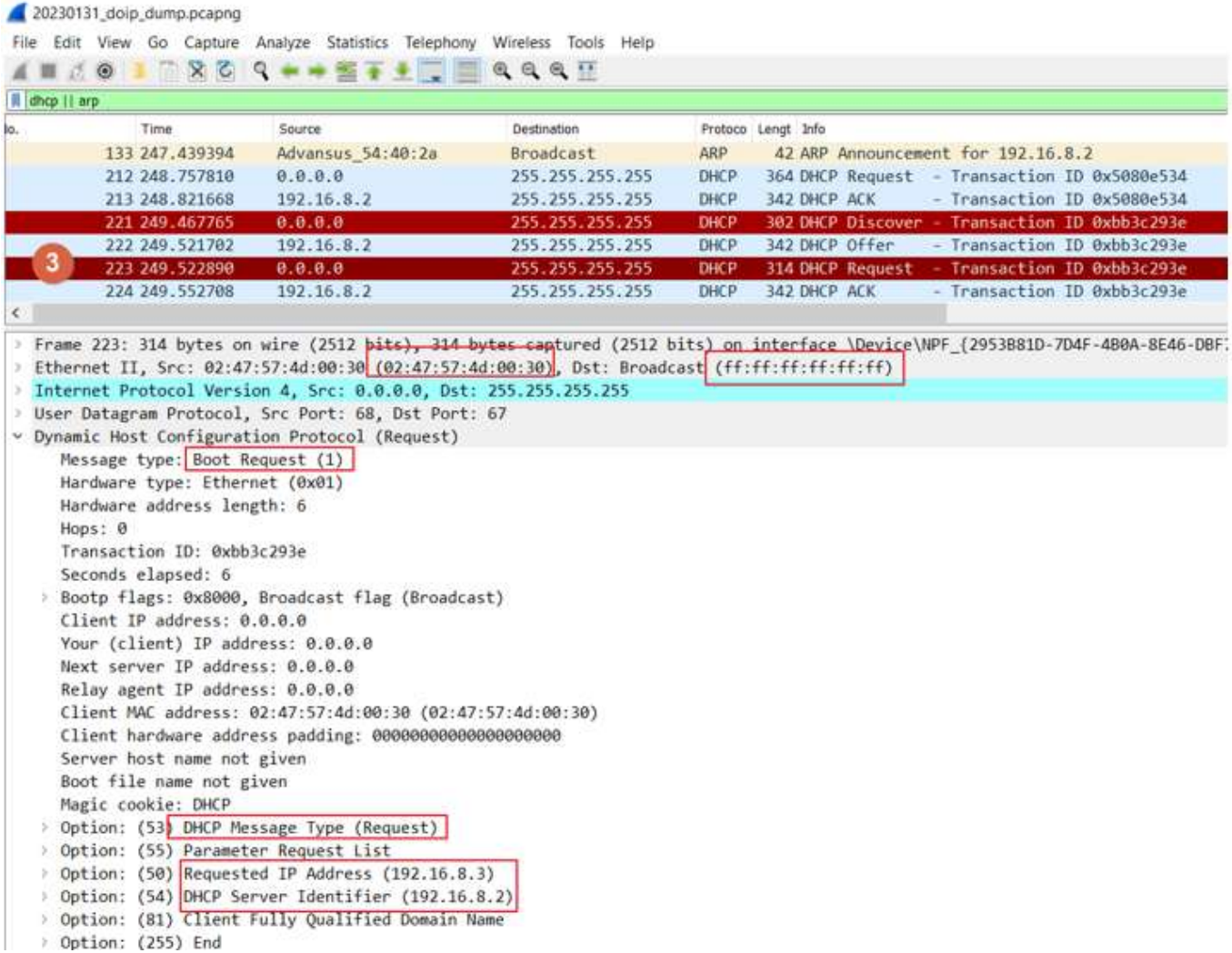


DHCP有四个阶段

2、提供阶段
DHCP服务器使用DHCP offer回应DHCP客户端。（服务器回应客户端时使用的端口为UDP 67，客户端选择最先回复DHCP offer报文作为客户端请求的服务器）

提供阶段：DHCP服务器收到DHCP客户端发送的DHCP discover报文后回应DHCP offer报文，DHCP offer 报文中包含了分配给客户端指定mac地址的IP地址，dns，租期，网关等参数。网络内的所有客户端根据封装在DHCP offer报文中的mac地址选择是否接收此报文。

3.4.6 DHCP工作原理

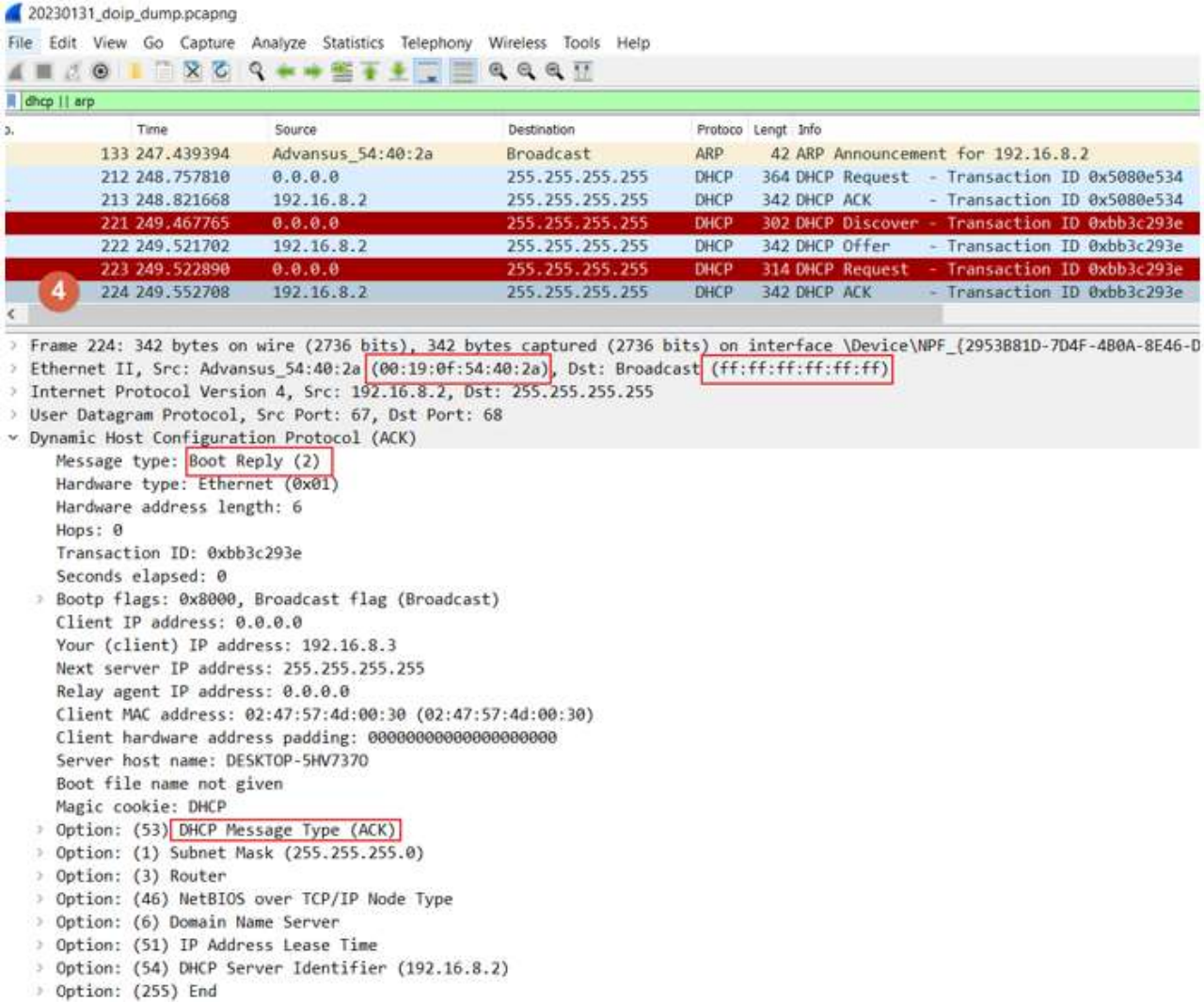


DHCP有四个阶段

3、选择阶段
DHCP客户端广播发送DHCP request选择报文。

DHCP客户端以广播方式发送DHCP discover报文，在同一个网段内有多个服务器回应DHCP offer报文则客户端会选择第一个回应DHCP offer的服务器来进行IP地址选择并广播发送DHCP request报文。以广播方式发送DHCP REQUEST报文，是为了通知所有的DHCP服务器，它将选择某个DHCP服务器提供的IP地址，其他DHCP服务器可以重新将曾经分配给客户端的IP地址分配给其他客户端。

3.4.6 DHCP工作原理



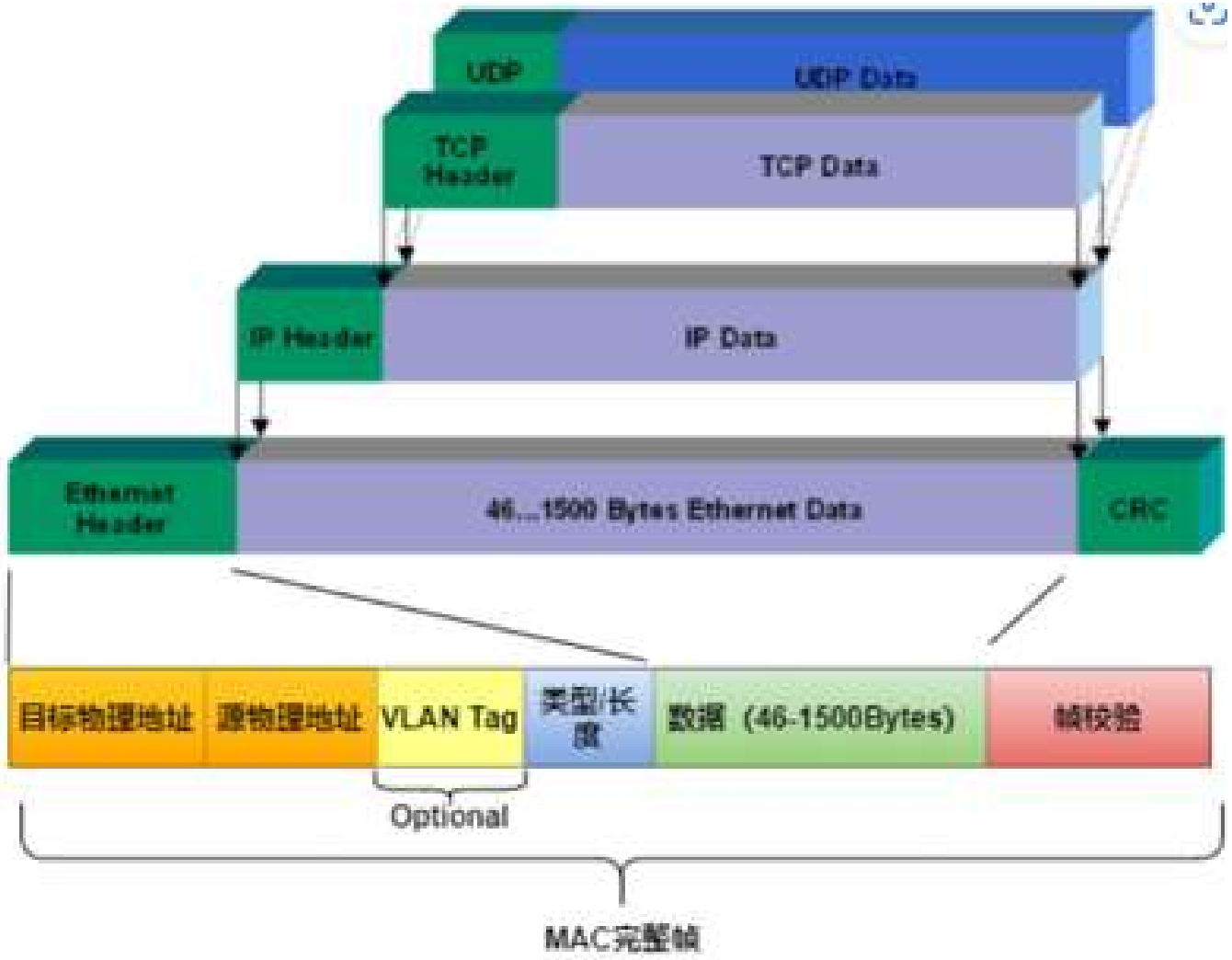
DHCP有四个阶段

4、确认阶段

服务器回应DHCP ack确认报文。

DHCP服务器收到DHCP客户端发送的DHCP request报文之后会回复DHCP ACK来确认分配IP地址。DHCP客户端受到服务器发送过来的DHCP ACK之后会免费发送广播arp报文来探测在这个网段中是否有使用该IP地址的客户端，如果没有则使用该IP地址，如果有客户端会向服务器发送DECLINE报文，并重新向服务器请求IP地址，同时，服务器会将此地址列为冲突地址。当服务器没有空闲地址可分配时，再选择冲突地址进行分配，尽量减少分配出去的地址冲突。如果因为DHCP客户端和DHCP服务器协商和DHCP服务器受到DHCP request报文过慢导致此IP地址已经分配给其他客户端，DHCP服务器会回复DHCP NAK报文给客户端，客户端受到DHCP NAk报文之后会重新发送DHCP discover报文寻找IP地址。

3.4.7 TCP协议



TCP的主要特点包括：

1.连接建立：在数据传输开始之前，TCP通过三次握手过程在发送方和接收方之间建立连接。这个过程确保双方都准备好交换数据，并建立初始的数据序列号。

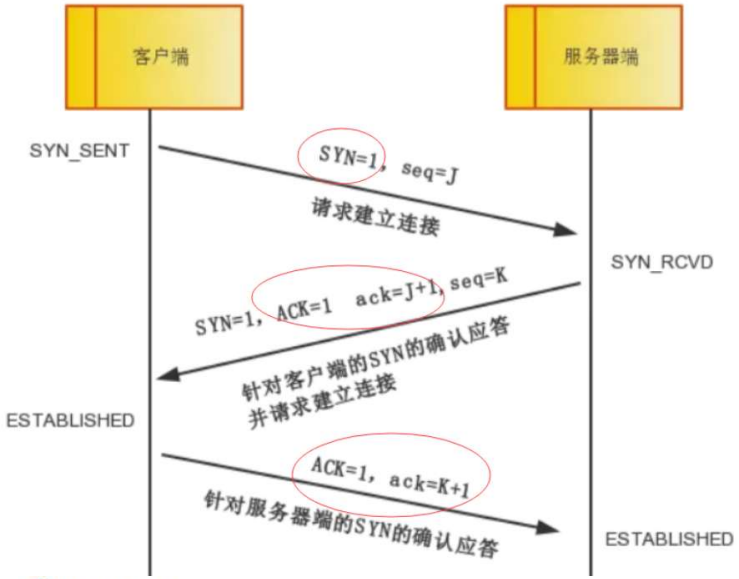
2.可靠性：TCP保证发送方发送的数据会被接收方正确接收，没有丢失或损坏。如果在传输过程中数据段丢失，接收方会向发送方请求重新传输。TCP还确保数据段按正确的顺序接收。

3.流量控制：TCP使用流量控制机制来防止数据过载。接收方向发送方传递缓冲区的可用空间情况，允许发送方调节数据传输速率，避免过载接收方。

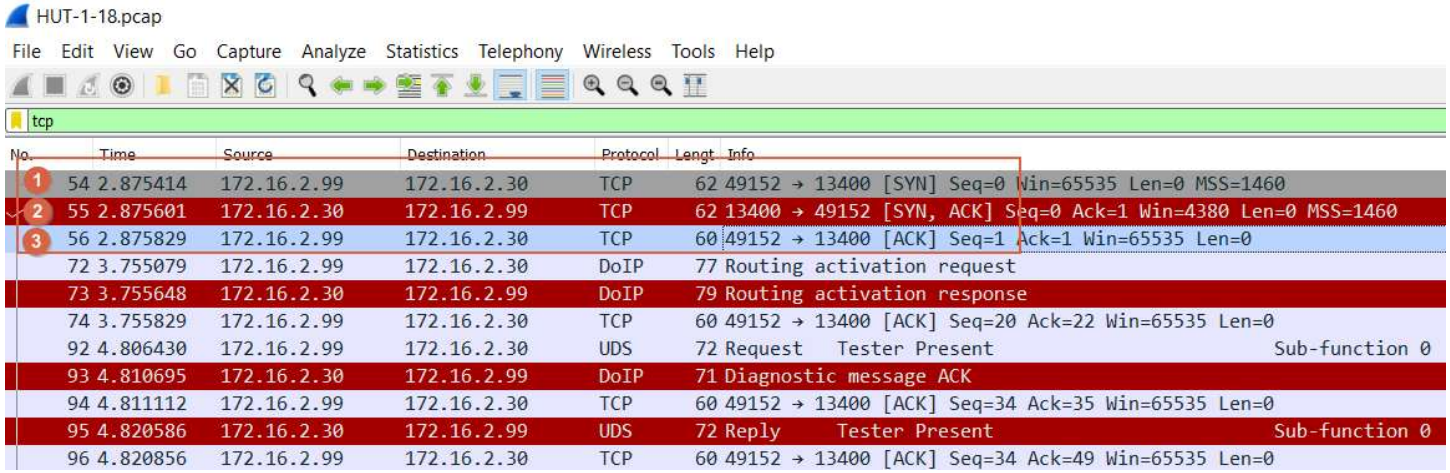
4.拥塞控制：TCP实现拥塞控制来管理网络流量，防止网络拥塞。它根据网络条件调整数据传输速率，避免数据丢失，并保持最佳性能。

5.连接终止：当数据传输完成后，TCP通过四次握手过程终止发送方和接收方之间的连接。

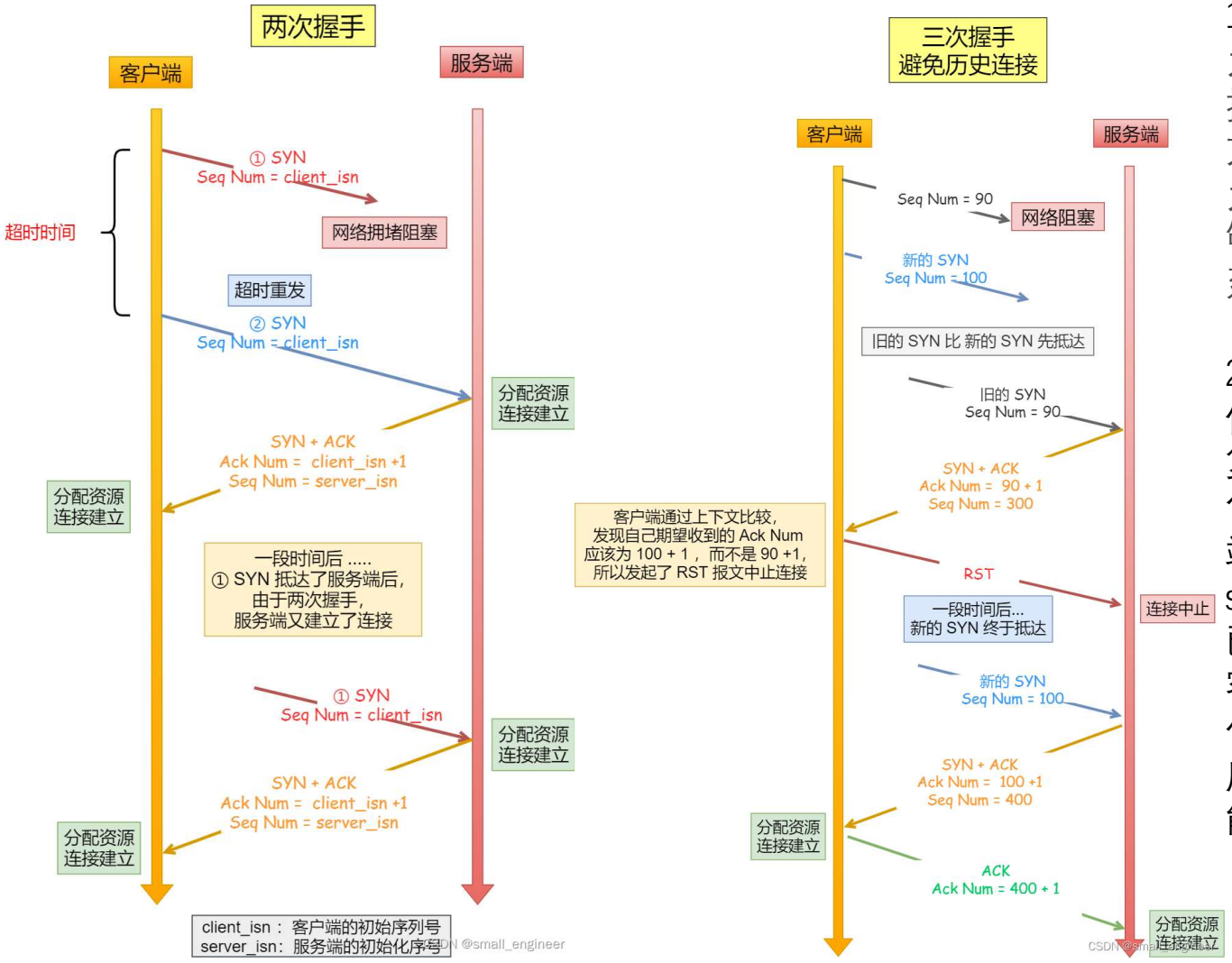
3.4.7 TCP协议—三次握手建立连接



- Init状态，双方都处于close状态，先是接收端监听某个端口，进入listen状态
- step1：发送syn包时，希望向接收端建立连接，并初始化序列号。此时发送端进入syn_sent状态
- step2：接收端接受到了syn包，发送ack+syn包，并初始化序列号，并将发送的序列号+1放入确认应答号中。此时接收端进入syn_rcvd状态
- step3：发送ack包，给确认应答号+1，表示收到了接收端的报文，进入establelisten状态服务端收到了客户端的报文后，也进入了establelisten状态。
- 注意：这里的第三次握手是可以携带数据的，因为发送端已经确认了服务端有接受和发送的能力。而其他两次并没有确认对方的接受能力



3.4.7 TCP为何要三次握手



1.总结：

为什么不使用两次握手：无法防止历史连接的建立，会造成资源浪费，不能同步双方序列号

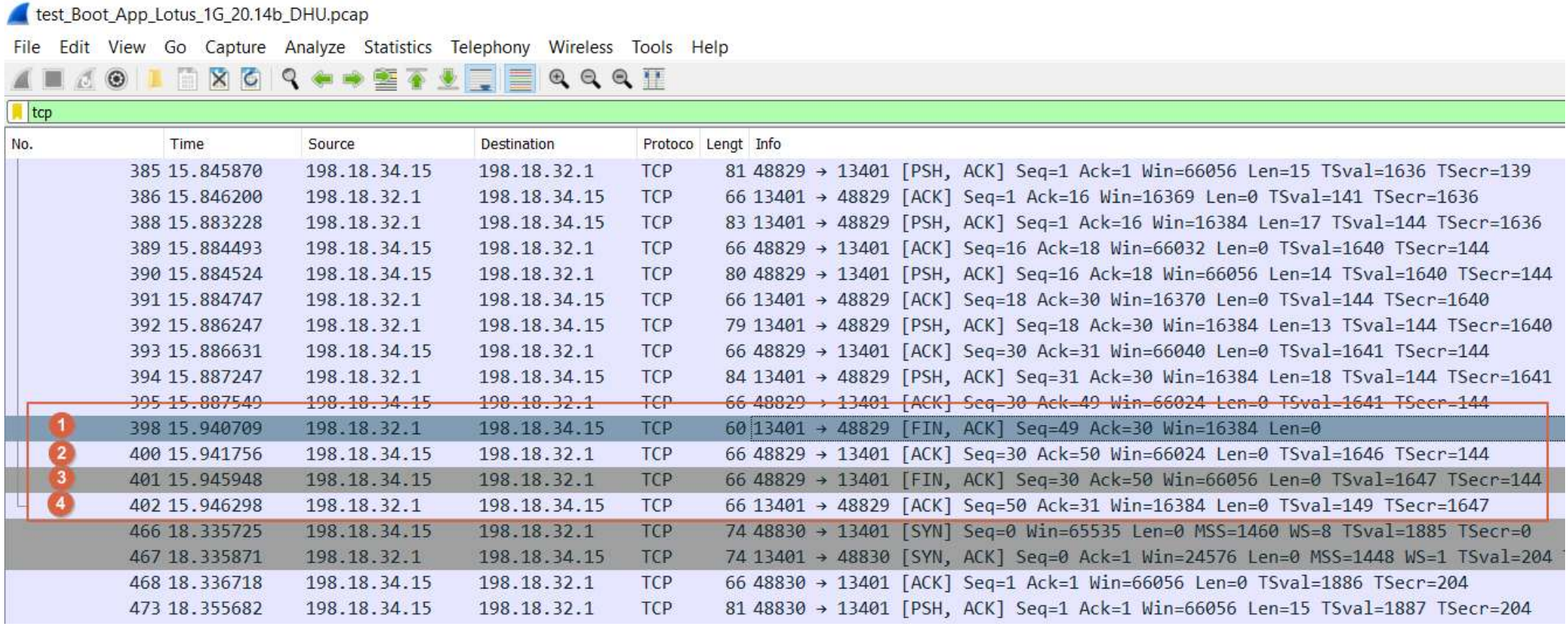
为什么不使用四次握手：三次握手已经足够，再多一次握手只会造成更多的连接时延

2.序列号的作用：

使对方有序的接受数据包避免对方接收重复的数据包可以知道自己发送出去的数据包，哪些是被接收的三次握手的话，客户端发送一个syn报文，服务端接受到后发送syn+ack报文，表示客户端的初始化序列号已经收到。

客户端在回复ack报文，表示服务端的初始化序列号也已经收到。这样才能确保双方序列号都被同步了。如果是两次握手，只能保证一方的序列号被成功接收

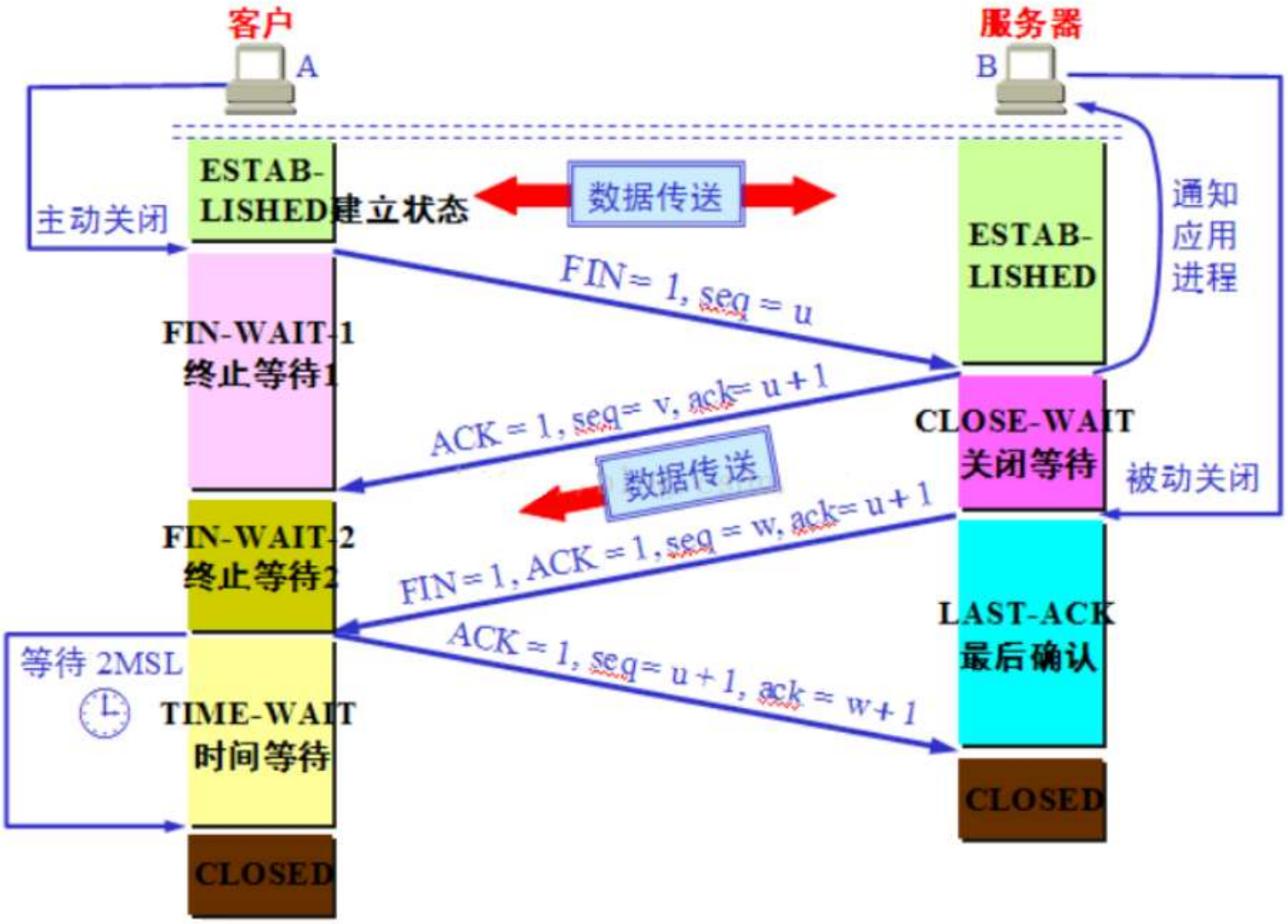
3.4.7 TCP 协议---四次挥手释放连接



3.4.7 TCP 协议---四次挥手释放连接

第一次挥手：Client端发起挥手请求，并且停止发送数据，向Server端发送标志位是FIN报文段，FIN=1，设置序列号seq=u（等于前面已经传送过来的数据的最后一个字节的序号加1），此时，Client端进入FIN_WAIT_1（终止等待1）状态，这表示Client端没有数据要发送给Server端了（TCP规定，FIN报文段即使不携带数据，也要消耗一个序号）；
第二次挥手：Server端收到了Client端发送的FIN报文段，向Client端返回一个标志位是ACK=1的报文段，ack设为seq加1（u+1），并且带上自己的序列号seq=v，服务端就进入了CLOSE-WAIT（关闭等待）状态，TCP服务器通知高层的应用进程，客户端向服务器的方向就释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的数据）；
第三次挥手：服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，**FIN=1，ack=u+1**，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq=w，此时，服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。
第四次挥手：客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是seq=u+1，此时，客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2**MSL（最长报文段寿命）的时间后，当客户端依然没有收到回复，才进入CLOSED状态。服务器只要收到了客户端发出的确认，立即进入CLOSED状态。可以看到，服务器结束TCP连接的时间要比客户端早一些。

3.4.7 TCP 协议---四次挥手释放连接



为什么连接的时候是三次握手，关闭的时候却是四次握手？

因为当Server端收到Client端的SYN连接请求报文后，可以直接发送SYN+ACK报文。其中ACK报文是用来应答的，SYN报文是用来同步的。所以建立连接只需要三次握手。

但是关闭连接时，当Server端收到FIN报文时，很可能并不会立即关闭SOCKET，所以只能先回复一个ACK报文，告诉Client端，“你发的FIN报文我收到了”。只有等到我Server端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送。故需要四步握手。

3.4.7 TCP 协议---四次挥手释放连接

为什么TIME_WAIT状态需要经过2MSL(最大报文段生存时间)才能返回到CLOSE状态?
2MSL，two Maximum Segment Lifetime，即两个最大段生命周期，（简述：防止ack报文丢失，Server再次发送Fin报文，一来一回最长时间就是2MSL）

虽然按道理，四个报文都发送完毕，我们可以直接进入CLOSE状态了，但是我们必须假象网络是不可靠的，有可以最后一个ACK丢失。所以TIME_WAIT状态就是用来重发可能丢失的ACK报文。在Client发送出最后的ACK回复，但该ACK可能丢失。Server如果没有收到ACK，将不断重复发送FIN片段。所以Client不能立即关闭，它必须确认Server接收到了该ACK。Client会在发送出ACK之后进入到TIME_WAIT状态。Client会设置一个计时器，等待2MSL的时间。如果在该时间内再次收到FIN，那么Client会重发ACK并再次等待2MSL。所谓的2MSL是两倍的MSL(Maximum Segment Lifetime)。MSL指一个片段在网络中最大的存活时间，2MSL就是一个发送和一个回复所需的最大时间。如果直到2MSL，Client都没有再次收到FIN，那么Client推断ACK已经被成功接收，则结束TCP连接。

作者：刘雪松
联系方式：18115503264

谢谢观看

扫一扫二维码，关注我的视频号

虚拟仪器

微信扫描二维码，关注我的公众号