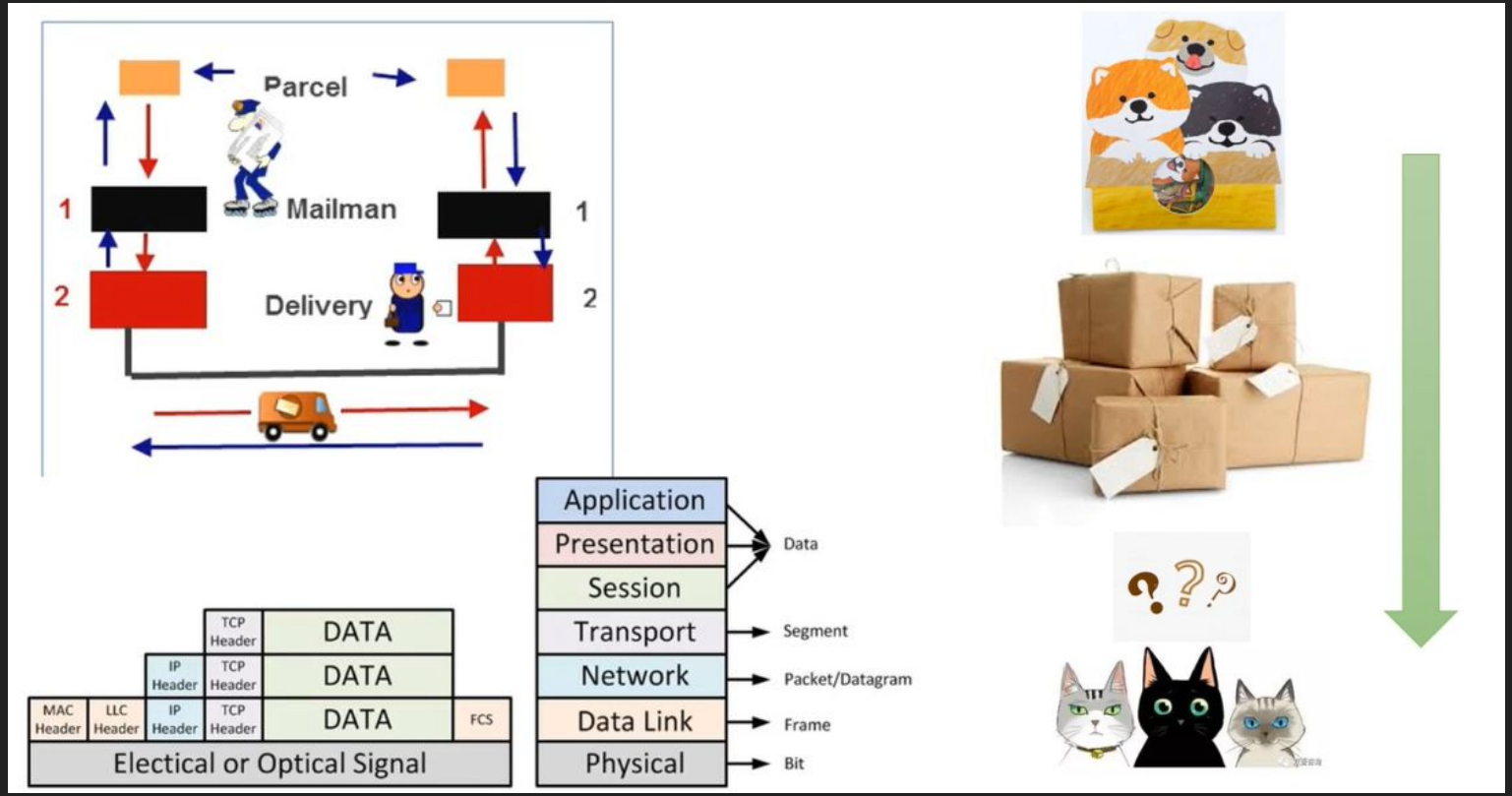
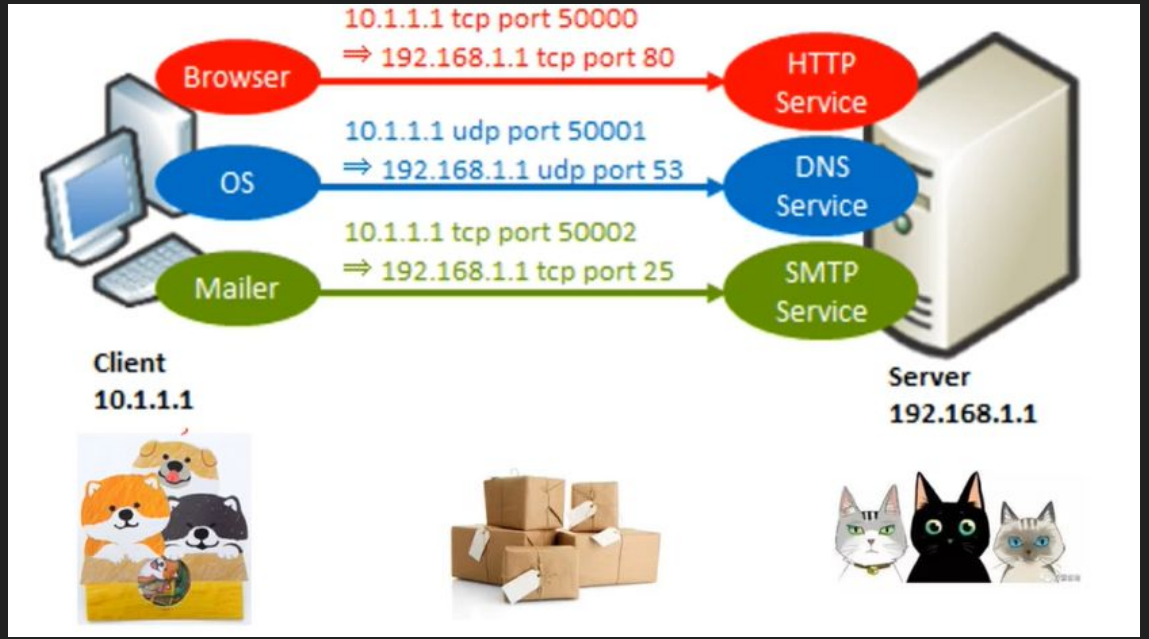
Chapter 2 - TCP/IP Protocol Suite 3

Layered System Review



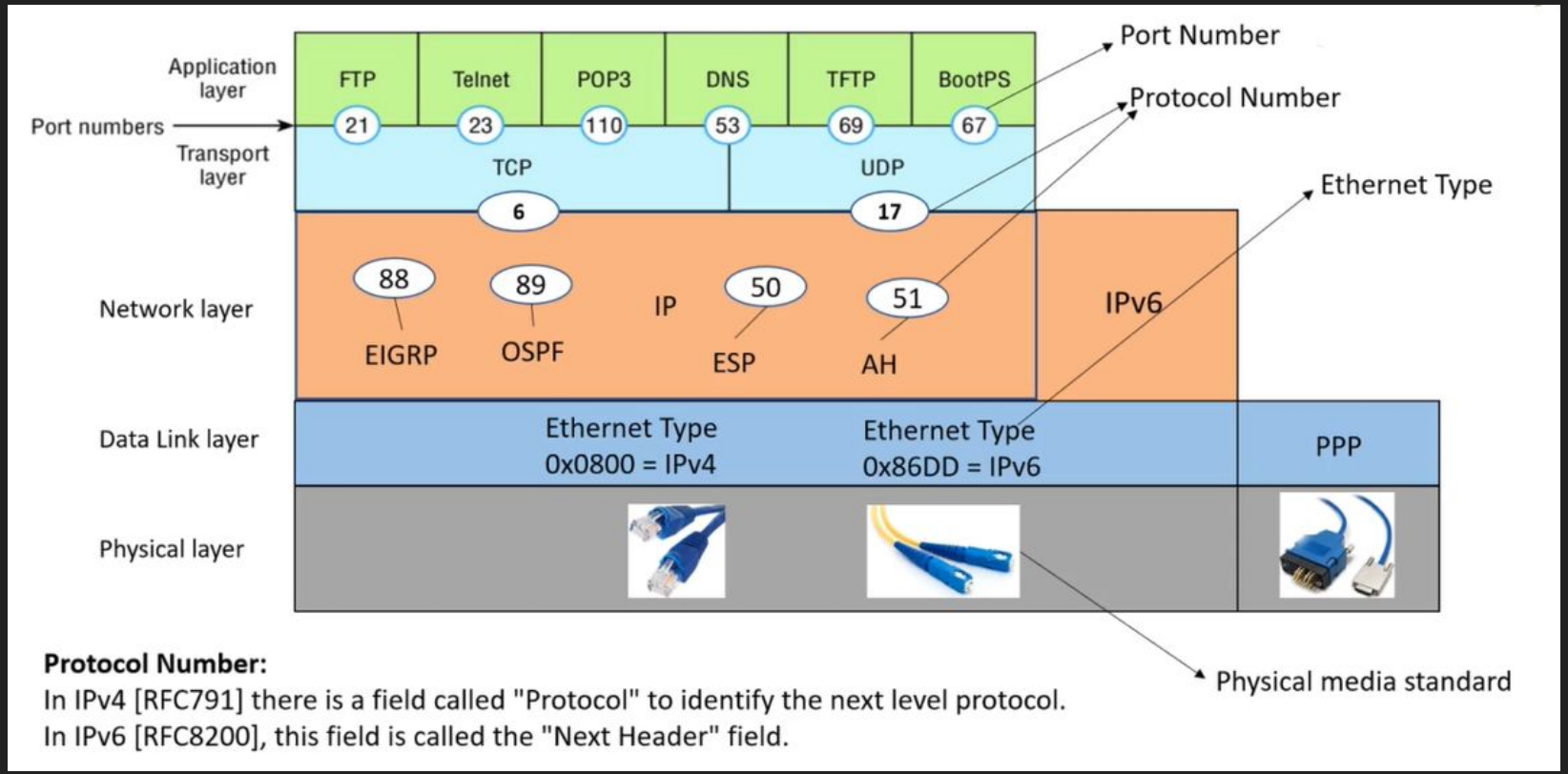
TCP/UDP Port Number



Internet Assigned Numbers Authority

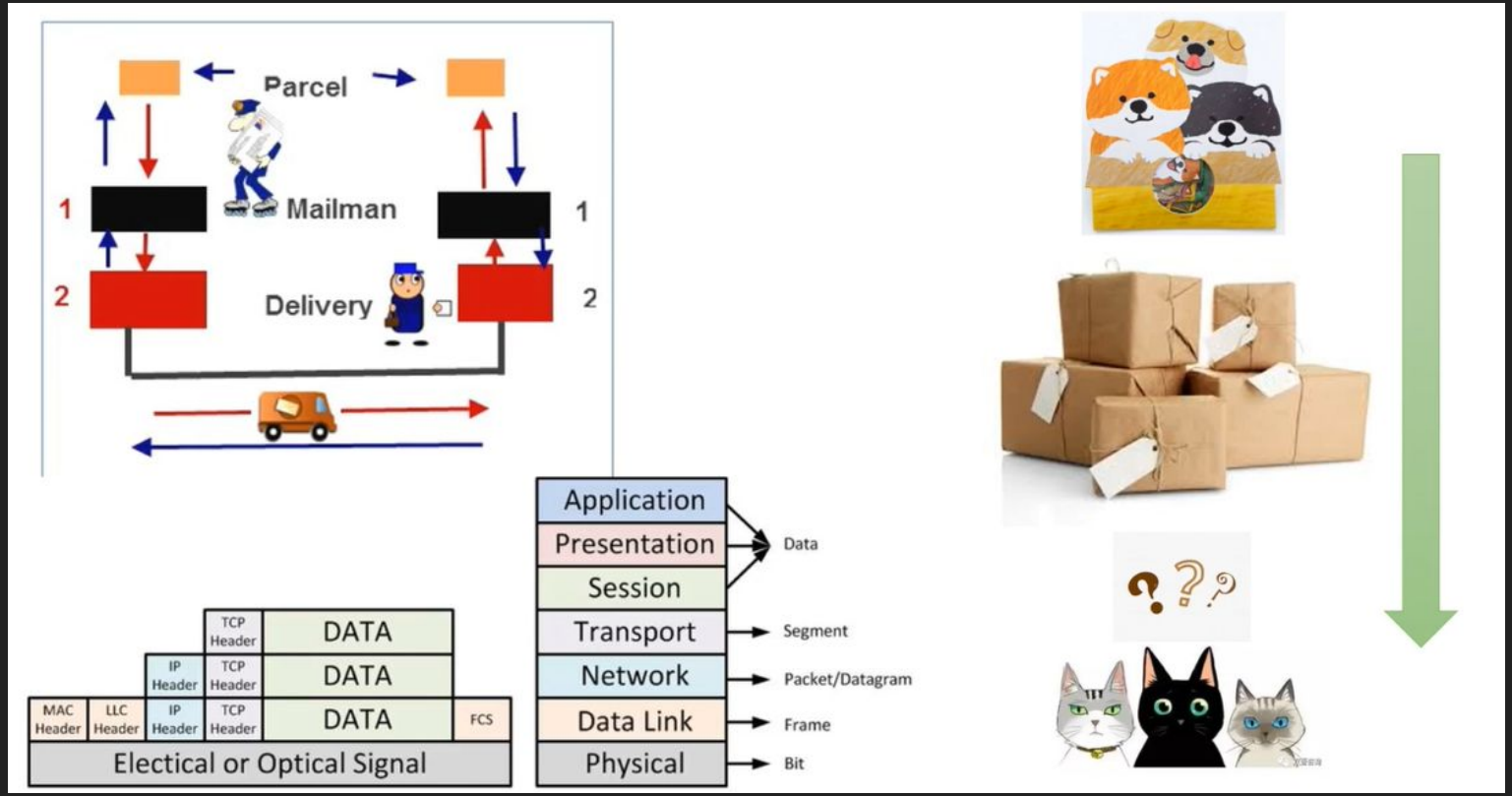
- 0 to 65535 (TCP & UDP)
- 0 to 1024 well-known 众所周知
- 1024 to 49151 registered ports 已注册端口号
- 49152 to 65535 dynamic or private 动态端口号或私有端口号

NIC, Ethernet Type, Protocol Number

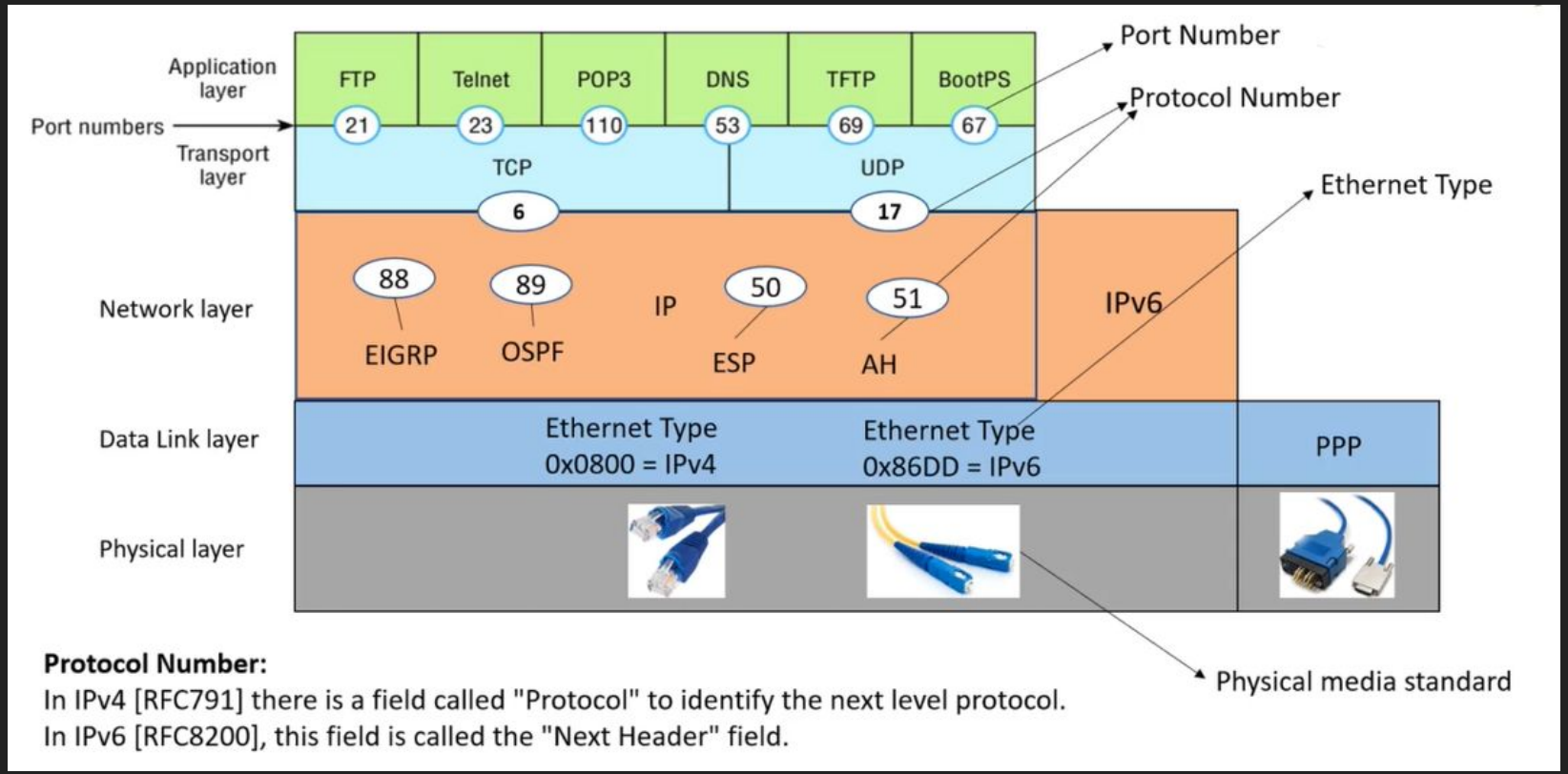


Protocol Number:
In IPv4 [RFC791] there is a field called "Protocol" to identify the next level protocol.
In IPv6 [RFC8200], this field is called the "Next Header" field.

Layered System Review

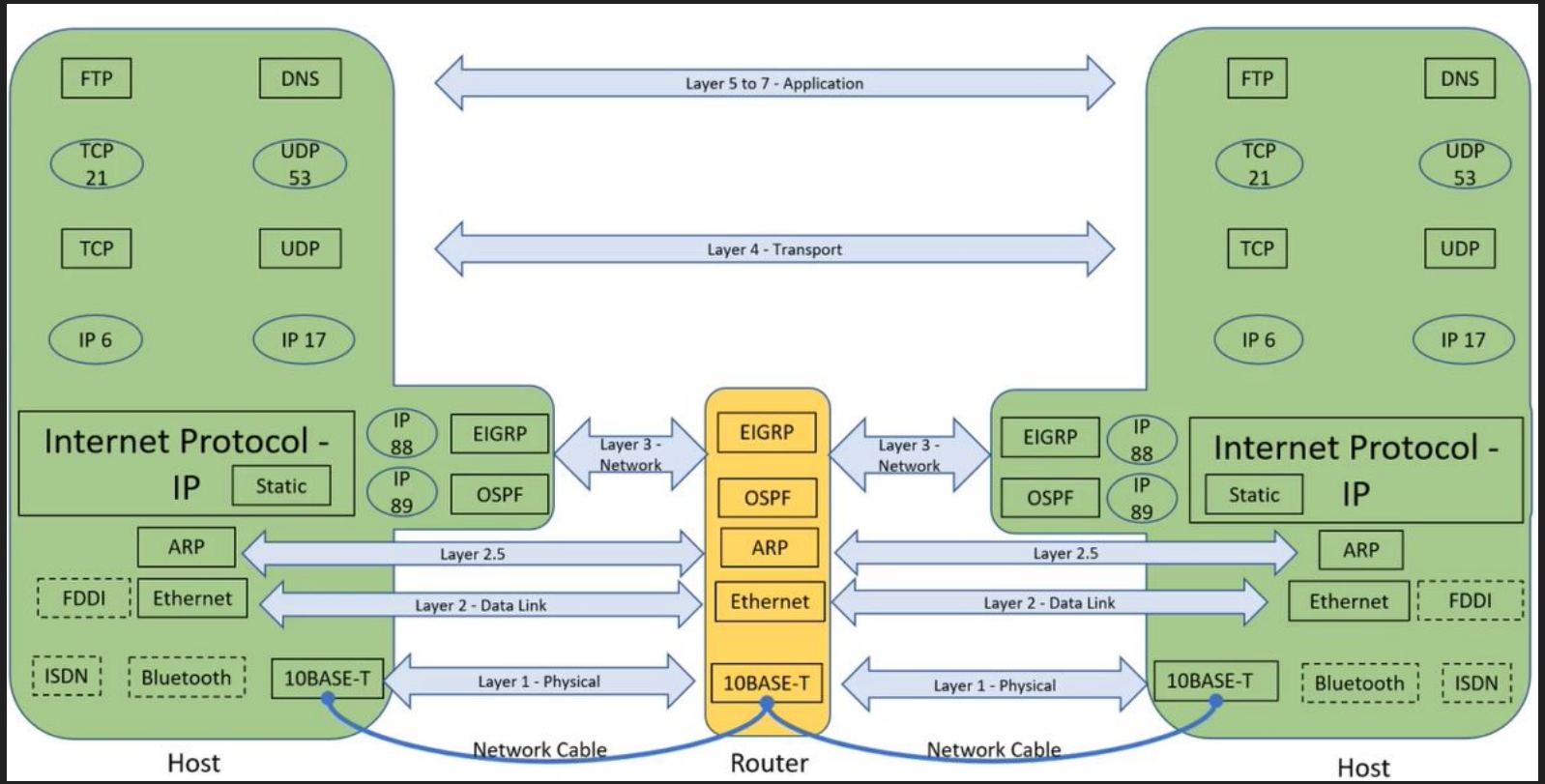


NIC, Ethernet Type, Protocol Number



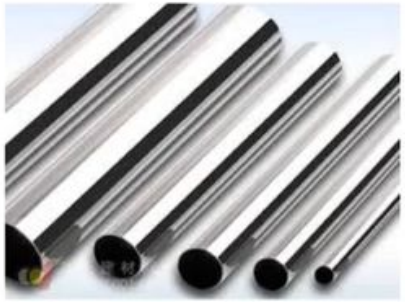
Protocol Number:
In IPv4 [RFC791] there is a field called "Protocol" to identify the next level protocol.
In IPv6 [RFC8200], this field is called the "Next Header" field.

Host to Host Communication



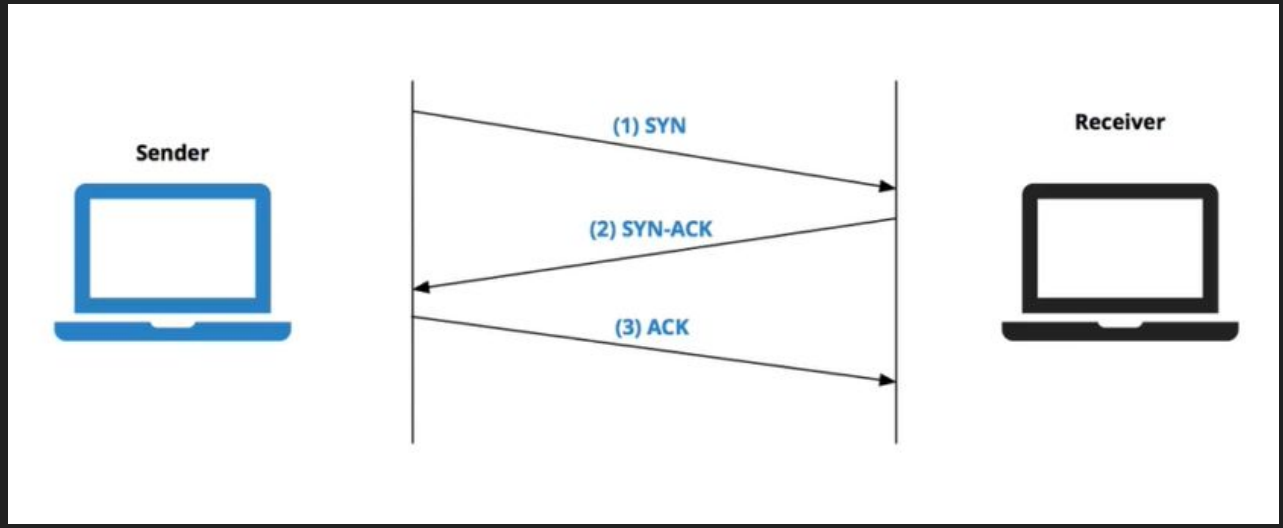
Chapter 2 - TCP/IP Protocol Suite 4

TCP vs UDP



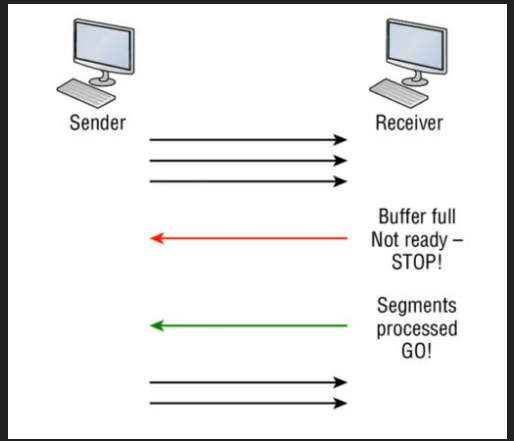
TCP 3-Way Handshake

- Sender's TCP stack contacts the destination's TCP stack to establish a connection. This creates a **virtual circuit**, and this type of communication is known as **connection-oriented**. 发件人的TCP堆栈与目标的TCP堆栈联系以建立连接。这将创建一个**虚拟电路**，这种通信方式称为**面向连接的**。
- When a machine receives a flood of datagrams too quickly for it to process it stores them in memory section called a **buffer**. 当机器接收到过多的数据报以至于无法对其进行处理时，它将它们存储在称为**缓冲区**的内存部分中。



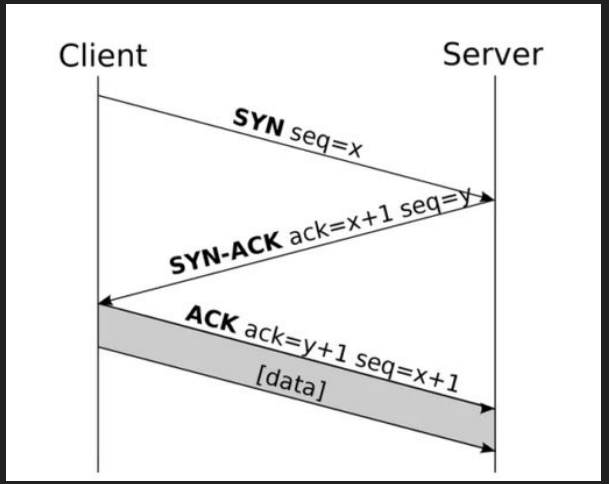
TCP Flow Control

- Buffer will be overflowed if too many concurrent connections. 如果并发连接过多，缓冲区将溢出。
- Flow control prevents a sending host on one side of the connection from overflowing the buffers in the receiving host. 流控制可防止连接一侧的发送主机使接收主机中的缓冲区溢出的情况的出现。
- The purpose of flow control is to provide a way for the receiving device to control the amount of data sent by the sender. 流量控制的目的是为接收设备提供一种控制发送方发送的数据量的方式。



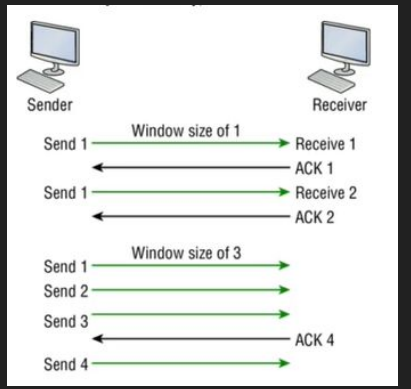
TCP Sequencing & Acknowledgement

- TCP takes large blocks of information from an application and breaks them into segments. TCP从应用程序那获取一整块信息，并将其分为多个部分。
- It numbers and sequences each segment so that the destination's TCP stack can put the segments back into the order the application intended. 它对每个段进行编号和排序，以便目标的TCP堆栈可以将这些段放回应用程序预期的顺序。
- After these segments are sent on the transmitting host, TCP waits for an acknowledgement of the receiving end's TCP virtual circuit session, retransmitting any segments that aren't acknowledged. 这些段在发送主机上发送后，TCP等待接收端的TCP虚拟电路会话的确认，然后重新传输所有未确认的段。



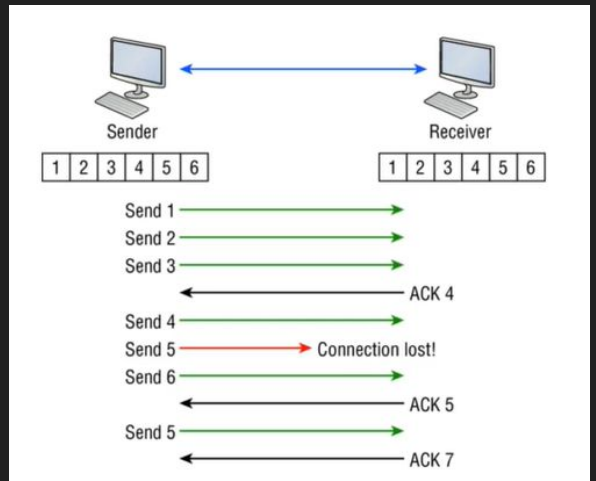
TCP Window Control

- It would be painfully slow if the transmitting machine had to actually wait for an acknowledgement after sending each and every segment 如果发送机在发送完每个段之后必须实际等待确认，那将会很痛苦的缓慢
- The transmitting machine is allowed to send without receiving an acknowledgement is called a **window** 允许发送机发送而不会收到确认，这被称为**窗口**
- Windows are used to control the amount of outstanding, unacknowledged data segments 窗口用于控制未处理的未确认数据段的数量
- TCP/IP measures it by counting the number of bytes TCP / IP通过计数字节数来进行测量
- If you've configured a window size of 1, the sending machine will wait for an acknowledgement for each data segment it transmits before transmitting another one but will allow three to be transmitted before receiving an acknowledgement if the window size is set to 3 (in reality, the transmission isn't based on simple numbers but in the amount of bytes that can be sent!) 如果您将窗口大小配置为1，则发送机将在发送另一个数据段之前等待发送的每个数据段的确认，但如果窗口大小设置为3，则发送机将允许发送三个数据再接收确认。（实际上，传输不是基于简单的数字，而是基于可以发送的字节数！）

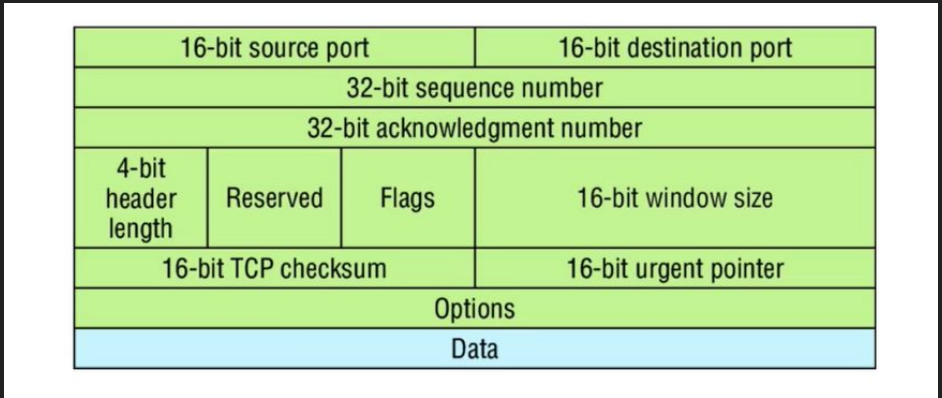


TCP Retransmission

- TCP guarantees that the data won't be duplicated or lost. TCP保证数据不会重复或丢失。
- The sender documents each segment measured in bytes, then sends and waits for this acknowledgement before sending the next segment. 发送方记录以字节为单位的每个段，然后发送并等待此确认，然后再发送下一个段。
- When it sends a segment, the transmitting machine starts a timer and will retransmit if it expires before it gets an acknowledgement back from the receiving end. 发送段时，发送机启动计时器，如果在到达接收端确认之前到期，将重新发送。



TCP Segment Format



- The TCP header is 20 bytes long, or up to 24 bytes with options. TCP标头的长度为20个字节，带选项最多为24个字节。
- TCP creates a lot of overhead. TCP会产生大量开销。
- The cyclic redundancy check (CRC), used because TCP doesn't trust the lower layers and checks everything. The CRC checks the header and data fields. 使用循环冗余校验（CRC），因为TCP不信任较低的层并检查所有内容。CRC检查标头和数据字段。

UDP

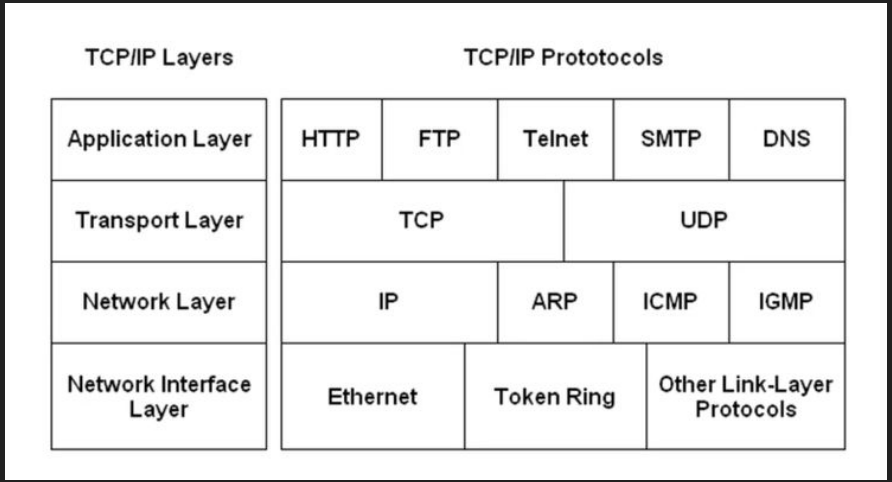
- UDP does not sequence the segments and does not care about the order in which the segments arrive at the destination. UDP不会对段进行排序，也不关心段到达目标的顺序。
- UDP doesn’t create a virtual circuit, not does it contact the destination before delivering information to it. It’s also considered a connectionless protocol. UDP不会创建虚拟电路，在将信息传递给目的地之前，它不会与目的地联系。它也被认为是无连接协议。
- UDP assumes that the application will use its own reliability method. UDP假定应用程序将使用其自己的可靠性方法。
- TCP for reliability or UDP for faster transfers. TCP表示可靠性，而UDP表示传输速度更快。

UDP Segment Format

- UDP, like TCP, doesn't trust the lower layers and runs its own CRC. UDP和TCP一样，也不信任较低的层，而是运行自己的CRC。
- Low overhead. 低开销。

Network Layer Protocols

- Routing and providing a single network interface to the upper layers. 路由并提供到上层的单个网络接口。
- Internet Protocol (IP) 互联网协议（IP）
- Internet Control Message Protocol (ICMP) 互联网控制消息协议（ICMP）
- Address Resolution Protocol (ARP) 地址解析协议（ARP）



TCP vs UDP Summary

TCP	UDP
Sequenced	Unsequenced
Reliable	Unreliable
Connection-oriented	Connectionless
Virtual circuit	Low overhead
Acknowledgments	No acknowledgment
Windowing flow control	No windowing or flow control of any type