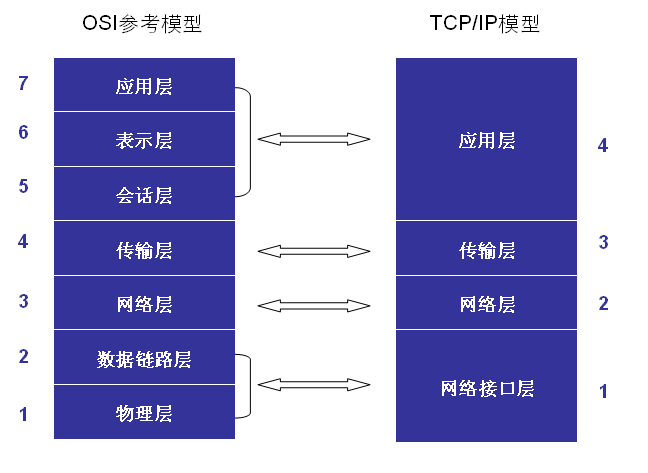
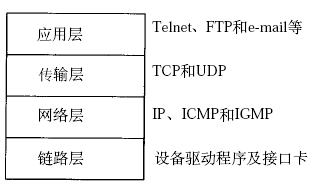
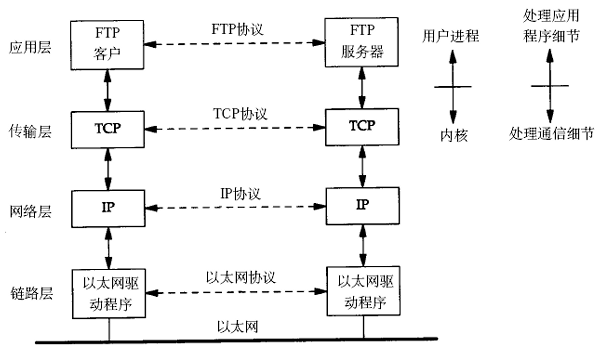
1. **OSI七层模型和TCP四层模型**





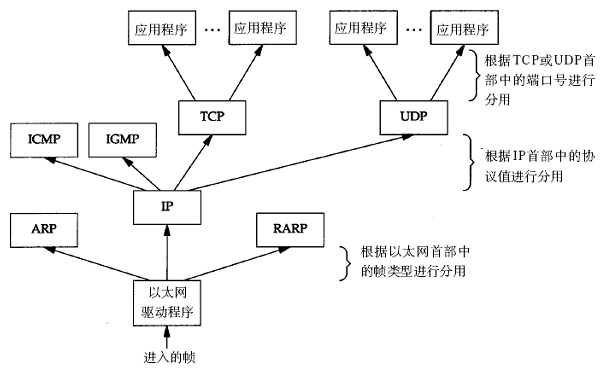
1. **TCP/IP通信过程**



1. **multiplexing过程**

虽然IP、ARP和RARP数据报都需要以太网驱动程序来封装成帧，但是从功能上划分，ARP和RARP属于链路层，IP属于网络层。

虽然ICMP、IGMP、TCP、UDP的数据都需要IP协议来封装成数据报，但是从功能上划分，ICMP、IGMP与IP同属于网络层，TCP和UDP属于传输层。



1. **MAC地址**

MAC地址也叫物理地址、硬件地址，由网络设备制造商生产时烧录在网卡，是全球唯一的。IP地址是32位，MAC地址是48位（如：40-8D-5C-9F-B2-BA）。

网络中数据帧从一个路由节点到下一跳路由节点，是依靠MAC地址而非IP地址进行传输的，因此数据传输时需要知道传输路径上的节点的MAC地址，通过ARP协议获取某IP主机的MAC地址。

1. **ARP协议**

地址解析协议，可以实现**通过IP地址获得对应主机的物理地址（MAC地址）**。

ARP协议要求通信的主机双方必须在**同一个物理网段**。

ARP协议具体通信过程：

①每台主机都会在自己的ARP高速缓冲区建立一个ARP列表，用于存储IP地址与MAC地址的对应关系。当源主机需要发数据包到目标主机时，先检查自己的ARP列表中是否存在该IP地址对应的MAC地址，如果存在则直接将数据包发送到该MAC地址。

②如果源主机没有目标主机的MAC地址时就需要向本地网段发一个ARP请求（ARPrequest）的广播包，用于查询目标主机对应的MAC地址。此ARP请求数据包里包括源主机的IP地址、NAC地址及目标地址的IP地址。

③网段中所有主机收到这个ARP请求后，看数据包中包含的IP地址是否与自己的IP地址一致，如果不一样就忽略；如果一致就把发送端的MAC地址和IP地址添加到自己的ARP列表中，如果该对应关系已经存在，则将其覆盖掉，然后给源主机发送一个ARP响应包(ARPreply)（单播），响应包包括自己的IP地址和MAC地址。

④源主机收到目标主机的响应后，将目标主机的IP地址的和MAC地址添加到自己的ARP列表中，并用此消息开始数据的传输。

⑤如果源主机一直没有收到ARP响应包，则提示ARP查询失败。

1. **RARP协议**

RARP协议**通过MAC地址获取IP地址**。

当设置一台新机器时，其RARP客户端需要向路由器上的RARP服务器请求相应的IP地址。假设在路由表中已经设置了一个记录，RARP服务器将返回IP地址给机器，此机器就会存储起来以便日后使用。

1. **ARP攻击**

ARP协议是建立在信任局域网内所有节点的基础上的，效率很高但是不安全。ARP协议是无状态的协议，它不会检查自己是否发过请求包，也不知道自己是否发过请求包，它也不管是否合法的应答，都会接受并缓存。

假设局域网中有A、B、C三台主机，主机C进行ARP欺骗。

**假冒ARPreply包**：A询问B的MAC地址，此时C冒充B对A进行非法回复，而且是大量的，所以A就会误信C是B，这样主机C就劫持了主机A发送给主机B的信息。

**假冒ARPrequest包**：C直接冒充网关，此时主机C就会不停的发送ARP欺骗广播，大声说我的IP是192.168.0.1，我的硬件地址是mac-c，此时局域网内的所有主机都被欺骗，更改自己的缓存表，此时C就会监听整个局域网发送给互联网的数据。

1. **ICMP协议**

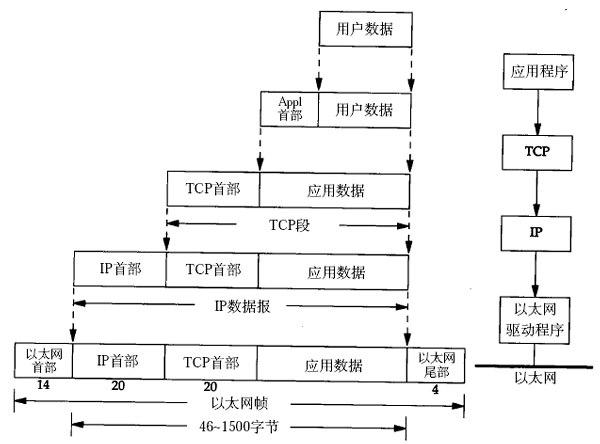
ICMP（Internet Control Message Protocol）（Internet控制报文协议）是一个网络层协议，一个新搭建好的网络往往需要先进行一个简单的测试，来验证网络是否畅通。但是IP协议并不提供可靠的传输，如果包丢了，IP协议并不能通知传输层是否丢包以及丢包原因。所以就需要ICMP来完成这样的功能。

ICMP协议功能：确认IP包是否成功到达目标地址；通知在发送过程中IP包被丢弃的原因。

ICMP是基于IP协议工作的，但是它并不是传输层的功能，仍然把它归结为网络层协议。

ping命令和traceroute命令都是基于ICMP协议的。

1. **数据包封装**

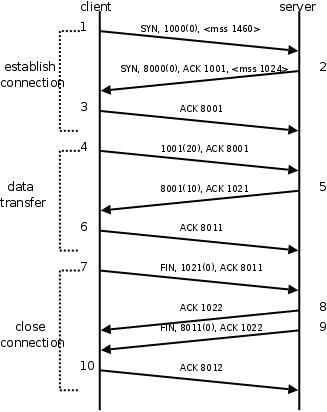


不同的协议对数据包有不同的称谓，在传输层叫做段（segment），在网络层叫数据报（datagram），在链路层叫做帧（frame）

以太网帧中的数据长度规定最小46字节，最大1500字节，ARP和RARP数据包的长度不够46字节，要在后面补填充位。最大值1500称为以太网的最大传输单元（MTU），不同的网络类型有不同的MTU，如果数据包长度大于1500则需要对数据包进行分片。

1500字节指的是有效载荷大小，不包括以太网首部和以太网尾部。

1. **TCP时序图**



**三次握手过程**：

1. 客户端发出段1，**SYN位标识连接请求**，**发送序号**是1000，这个序号在网络通信中用作临时的地址，每发一个数据字节，这个序号就要加1，这样在接收端就可以根据序号排出数据包的正确顺序，也可以发现丢包的情况，另外规定**SYN和FIN位也要占一个字节**，这次虽然没发数据，但是由于发了SYN位，因此下次再发送应该用序号1001。**mss表示最大段尺寸**，如果一个段太大，封装成帧后超过了链路层的最大帧长度，就必须在IP层分片，为了避免这种情况，客户端声明自己的最大段尺寸，建议服务器发来的段不要超过这个长度。
2. 服务器发出段2，也带有SYN位，同时置ACK位表示确认，确认序号是1001，表示“我接收到序号1000及其以前所有的段，请你下次发送序号为1001的段”，也就是应答了客户端的连接请求，同时也给客户端发出一个连接请求，同时声明最大段尺寸mms为1024。
3. 客户端发出段3，对服务器的连接请求进行应答，确认序号是8001。

在这个过程中，客户端和服务器分别给对方发了连接请求，也应答了对方的连接请求，

其中服务器的请求和应答在一个段中发出，因此一共有三个段用于建立连接，称为’‘’三

方握手（three-way-handshake）”’。在建立连接的同时，双方协商了一些信息，例如双

方发送序号的初始值、最大段尺寸等。

**数据传输的过程：**

1.客户端发出段4，包含从序号1001开始的20个字节数据。

2.服务器发出段5，确认序号为1021，对序号为1001-1020的数据表示确认收到，同时请

求发送序号1021开始的数据，服务器在应答的同时也向客户端发送从序号8001开始的10个字

节数据，这称为piggyback。

3.客户端发出段6，对服务器发来的序号为8001-8010的数据表示确认收到，请求发送序

号8011开始的数据。

**四次挥手过程：**

1.客户端发出段7，FIN位表示关闭连接的请求。

2.服务器发出段8，应答客户端的关闭连接请求。

3.服务器发出段9，其中也包含FIN位，向客户端发送关闭连接请求。

4.客户端发出段10，应答服务器的关闭连接请求。

1. **TCP状态转换图**

