

# 1 传输层简介

## 1.1 传输层的位置和功能

传输层位于网络层和应用层之间。如下图。

这里的Transport service interface指的是：operations provided by transport layer to application programs。这些operations具体就是一些原语，例如：

从而，（从使用的原语角度看，）两个进程之间的通信过程可以表示为：

网络层负责的是：提供不同主机之间的逻辑通信（对等层之间的通信好像是“水平”的，实际上要经过下层的协议栈）；应用层上运行着不同的应用。因此，传输层提供的功能是：**为不同主机的应用进程之间提供逻辑通信**。如下图：

## 1.2 传输层寻址与端口

### 1.2.1 端口与端口号

不同的进程怎么使用同一个传输层的协议（复用）与另一个主机的进程进行通信？答案是使用**不同的端口**。端口号是16位的比特串，用来标识正在通信的进程。端口号可以分为2类：

- **服务器端使用的端口号。** (1) 0 ~ 1023：熟知的、已经固定进程的端口号，IANA将其分配给FTP, DNS等TCP/IP最重要的一些应用程序。 (2) 1024 ~ 49151：登记端口号，需要到IANA登记然后即可自由使用。
- **客户端使用的端口号。** 49152 ~ 65535：短暂端口号，仅在客户进程运行时才动态分配并使用。

下图展示了：不同端口号使用同一个IP地址，通过网络层进行通信。

### 1.2.2 套接字

在网络中，可以使用IP地址来标识不同的主机，结合端口号就可以标识不同主机上的不同进程。这就是套接字（socket）：**socket = (IP address, port number)**。

### **1.3 连接建立？**

(PPT种这部分貌似不会考到？而且感觉是网络层的内容)

### **1.4 无连接服务与面向连接服务**

## **2 UDP**

UDP的功能只有：在IP基础上进行复用、分用、差错检测。

### **2.1 UDP数据报格式**

UDP的首部大小为8B (64 bits)，其中：

- UDP长度表示的是首部字段+数据字段的长度，最小为8（仅有首部8B）；
- UDP校验和可以省略。

### **2.2 UDP校验方法**

## **3 TCP**

### **3.1 TCP数据报格式**

TCP的首部大小为 $20 + 4n$  B ( $n \geq 0$  bits)，其中：

- 

注：实际实现中，TCP的数据报作为IP数据报的一部分（数据部分）。

### **3.2 TCP连接管理**

TCP连接采取client/server model，client发出连接请求，server进行回复。

注：TCP连接的对象是：套接字，而非IP地址或者端口号！

### 3.2.1 TCP连接建立

TCP采取“三次握手协议”，如下图：

注意， $ACK = x + 1$ 的数据报是对 $SEQ = x$ 的数据报的回应。下图更详细展示了client和server的状态：

注：如果双方同时发起连接建立的请求（如下图），则应当确保双方之间只建立1个连接。

### 3.2.2 TCP连接释放

TCP采取“四次握手协议”。为什么会相对复杂？因为TCP是一个全双工协议，必须单独拆除每一条信道，两个方向的接收、发送都需要单独关闭。方法如下图：

(1) 首先客户端向服务器发送一段TCP报文表明其想要释放TCP连接。

- 标记位为FIN，表示请求释放连接；
- 随后客户端进入FIN-WAIT-1阶段，即半关闭阶段，并且停止向服务端发送通信数据。

(2) 服务器接收到客户端请求断开连接的FIN报文后，结束ESTABLISHED阶段，进入CLOSE-WAIT阶段并返回一段TCP报文。

- 标记位为ACK，表示接收到客户端释放连接的请求；
- 序号为Seq = v，确认号为Ack = u + 1，表示是在收到客户端报文的基础上，将其序号值加1作为本段报文确认号Ack的值；
- 随后服务器开始准备释放服务器端到客户端方向上的连接；
- 客户端收到服务器发送过来的TCP报文后，确认服务器已经收到了客户端连接释放的请求，随后客户端进入FIN-WAIT-2阶段。

(3) 服务器端在发出ACK确认报文后，服务器端会将遗留的待传数据传送给客户端，待传输完成后即经过CLOSE-WAIT阶段，便做好了释放服务器端到客户端的连接准备，再次向客户端发出一段TCP报文。

- 标记位为FIN和ACK，表示已经准备好释放连接了；

- 序号为Seq = w，确认号Ack = u + 1，表示是在收到客户端报文的基础上，将其序号Seq的值加1作为本段报文确认号Ack的值；
- 随后服务器端结束CLOSE-WAIT阶段，进入LAST-ACK阶段，并且停止向客户端发送数据。

(4) 客户端收到从服务器发来的TCP报文，确认了服务器已经做好释放连接的准备，于是进入TIME-WAIT阶段，并向服务器发送一段报文。

- 标记位为ACK，表示接收到服务器准备好释放连接的信号；
- 序号为Seq= u + 1，表示是在已收到服务器报文的基础上，将其确认号Ack值作为本段序号的值；确认号为Ack= w + 1，表示是在收到了服务器报文的基础上，将其序号Seq的值加1作为本段报文确认号的值.

最后，客户端开始在TIME-WAIT阶段等待2 MSL，随后进入CLOSED阶段，由此完成四次挥手。

### 3.3 TCP可靠传输

### 3.4 TCP流量管理

### 3.5 TCP拥塞控制