

姓名：蔡与望

学号：2020010801024

1. 点到点

什么是点到点？

发送端将数据传给与它直接相连的设备，相邻设备传输数据，最终将数据传到接收端。**管道化信道。**

为什么要进行点到点传输？

在数据链路层，点到点可以理解为一个节点的数据链路层接受 IP 层数据封装后，将数据帧从链路上发送到与其相连的下个节点。它的优点是不参与整个传输，发送端设备资源利用利用率高。当接收端出现问题，点到点传输可以利用存储转发技术进行缓冲。

怎么样理解其与端到端相比的差距？

无法确定发送端是否接收或何时到数据，不能保证数据传输的可靠性。

2. 端到端

什么是端到端？

数据从发送端到接收端要经过各种各样的交换设备；建立连接-发送数据-确认收到，这个过程称为端到端传输。**存储式信道。**

为什么要进行端到端传输？

端到端关注发送端到接收端的整体过程。是建立在点到点通信基础之上，它在运输层面上说明了整体进程的数据传输。优点是链路建立后，发送端能确定接收端收到数据，数据在经历中间交换设备不需要存储转发，传输延迟小。

怎么样理解其与点到点相比的差距？

发送端参与整个数据传输过程，会导致对发送端设备的浪费；接收端如果出现问题，整个端到端运输不能实现。

3. 帧同步（成帧）

什么是帧同步？

在发送端提供每帧的起始标记，在接收端检测并获取这一标志。

确定一个有效数据块在比特流中的起始与终止位置。

为什么要进行帧同步？

在数字时分多路通信系统中，使接收端和发送端的帧结构完全一致，从而能够正确分离每一路的信号，正确交换数字信息。

- 从比特流中取出有效数据
- 将 PDU 解封为 Header + SDU。
- 提取控制信息，完成控制。

怎么样实现帧同步？

起止式同步法，插入特殊同步码组法。假同步问题：信息码中也可能会出现与帧同步码组一样的码元组合，从而导致帧结构不能被正确识别出来。

- 字节计数法：用长度定界。
 - 面向位的首尾定界符法：用一串特定数字定界。
 - 面向字符的首尾定界符法：用某个特定字节结合转义符定界。
 - 物理层违例编码：用非正常码型定界。
 - 校验和法：CRC 校验。
-

4. ARQ

什么是 ARQ?

自动重传请求 (Automatic Repeat-reQuest) 是数据链路层的错误纠正协议之一, 接收方请求发送方重传出错的数据报文来恢复出错的报文。

为什么要引入 ARQ?

接收方通过 ARQ 请求发送方重传出错的信息 (后向纠错), 保障数据传输过程中的可靠性和准确性。

怎么样分类 ARQ?

停止等待, 回退 n 帧, 选择性重传, 混合 ARQ。

5. 流量控制

什么是流量控制?

是利用软件或硬件, 来实现对网络数据流量进行控制的一种措施。控制发送方发送数据的速度不能太快。

为什么要进行流量控制?

随着网络技术的快速发展, 基于网络的应用越来越多、越来越复杂。种类繁多的应用正在吞噬着越来越多的网络资源。网络作为一种新的传媒载体, 也正在遭受媒体的冲击。尤其是网络视频、个人媒体、传统电视等媒体向互联网的渗入是的网络中的流量急剧上升, 这使得运营商的运营和管理成本大幅度增长。运营商可以应用限流的方法控制网络流量。

怎么样实现流量控制？

现阶段互联网上的流量主要由 P2P 和 HTTP 产生，据估计这两种流量已经占到全部流量的 70% 以上，并且仍呈上升趋势。因此流量控制的重点是 P2P 和 HTTP，降低这两种协议产生的流量将有效降低网络整体流量。通过对多种网络流量控制系统的比较，然后采用一种最优的系统。将系统部署在网络出口来缓存 P2P 和 HTTP 流量，对同一种资源的后续请求将有缓存来响应，从而降低网络流量、节省带宽并提高用户体验。

让发送速度低于双方接口速度、网络与接收方处理速度。

6. 三次握手

什么是三次握手？

TCP 建立连接时，双方需要对会话的相关信息进行一次确认，形象形容为三次握手。

为什么要先进行三次握手？

通过三次握手，通信双方可以从对方那里了解关于这次会话的相关信息：

1. 对方报文发送的开始序号；
2. 对方发送数据的缓冲区大小。
3. 能被接收的最大报文段长度。
4. 被支持的 TCP 选项。

怎么进行三次握手？

第一次握手，客户端向服务器发送一个 SYN(seq=x)，进入 SYN_SENT 状态；

第二次握手，服务器向客户端发送 SYN(seq=y)，ACK=x+1，进入 SYN_RECV 状态；

第三次握手，客户端向服务器发送 $ACK=y+1$ ，双方进入 ESTABLISHED 状态，连接成功。

16. 套接字

什么是套接字？

是对网络中，不同主机上的应用进程之间，进行双向通信的端点的抽象。

为什么要引入套接字概念？

套接字上联应用进程，下联网络协议栈，是应用程序通过网络协议进行通信的接口。通过套接字，程序员可以很方便地访问 TCP/IP，从而开发各类网络程序。

怎么创建一个套接字？

确定其 IP 地址类型 (IPv4, IPv6)、数据传输方式 (面向连接、无连接)、传输协议 (TCP、UDP)，然后调用 `socket()` 函数，绑定对应地址。

17. TCP RTT

什么是 RTT？

RTT (Round Trip Time)，一次连接的往返时间，即数据从发送到接收再到确认的时间差。

为什么要设置 RTT?

对于不同的网络拥塞状况，RTT 的算法能够计算一个对应的、合适的超时时间，从而实现高效的重传。**是流量控制中的重要环节。**

怎么样得到一个合理的 RTT?

首先在发送端发包时记下初始时刻 t_0 ，收到 ACK 时再记录时刻 t_1 ，于是 $RTT = t_1 - t_0$ 。多次重复，记录多个 RTT。

然后平滑计算 $SRTT = \alpha SRTT + (1 - \alpha)R, \alpha = 0.875$ 。

18. TCP RTO

什么是 RTO?

RTO (Retransmission Time Out)，重传超时时间，超过这个时间没回复就重传。

为什么要计算 RTO?

在数据传输的过程中，经常会有数据包丢失的情况出现，此时就需要一个合理的、不长不短的时间，超过这个时间，发送方就进行重传。

怎么样计算 RTO?

$$EstimatedRTT = 0.875 EstimatedRTT + 0.125 SampleRTT$$

$$RTO = EstimatedRTT + 4 DevRTT$$

理论上 $RTO = 2RTT$ ，但 RTT 太敏感，不准确；

$$RTTVar = \beta RTTVar + (1 - \beta) |SRTT - RTT|$$

$$RTO = SRTT + 4RTTVar$$