

传输层

传输层服务:TCP/UDP

传输层协议为运行在不同HOST上的进程提供了一种逻辑通信机制。

工作方式（端系统）：

- 发送方：将应用递交的消息分成一个或多个报文段（segment），并向下传给网络层。
- 接收方：将接收到的报文段组装成消息，向上提交给应用层

网络层与传输层的区别：

- 网络层（IP协议）主机之间的逻辑通信机制
- 传输层：应用进程之间的逻辑通信机制，位于网络层之上，依赖于网络层服务，对网络层服务进行增强。

TCP 协议

可靠、按序的交付服务：数据不丢失，顺序不会乱的服务

- 拥塞控制
- 流量控制
- 连接建立

UDP

不可靠的交付服务

- 基于“尽力而为”的网络层，没有做可靠性的扩展

以上协议不保证延迟方面

多路复用和多路分用

- 多路分用：接收端使用的。传输层根据报文头部信息将收到的报文段交给正确的Socket，即不同的进程。
- 多路复用：发送端使用。从多个Socket接收数据，为每块数据封装上头部信息，生成报文段，交给网络层

分用如何工作：（TCP/UDP都有的功能）

主机接收到IP数据报（datagram）

- 每个数据报携带源IP地址，目的IP地址。
- 每个数据报携带一个传输层的段
- 每个段携带源端口号与目的端口号

主机收到后，传输层协议提取IP地址和端口号，将报文段导向相应的Socket（进程）。

网络层不关注端口号。

UDP

UDP的Socket用二元组标识（目的IP地址，目的端口号）。

主机收到UDP段后，检查UDP段导向绑定在端口号的Socket。

来自不同源IP地址/源端口号的IP数据包被导向同一个Socket。

TCP

由四元组标识==（源IP地址，源端口号，目的IP地址，目的端口号）

UDP

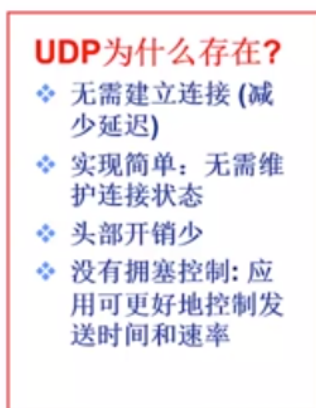
user datagram protocol

基于Internet IP协议

- 复用/分用
- 简单的错误校验（原因：路由器在存储转发的过程中有可能出错，端到端的原则）

由于IP协议是尽力而为，UDP也是尽力而为，所以会丢失。

无连接：UDP发送方和接收方之间不需要握手，每个UDP端的处理独立于其他段



常用于流媒体应用、DNS、SNMP

如何在UDP上实现数据可靠：在应用层增加数据可靠机制



可靠数据传输

什么是可靠：不错、不乱、不丢

- 可靠数据传输协议

该协议基本结构：接口

只考虑单向数据传输，但控制信息双向流动

Rdt1.0：可靠信道上的可靠数据传输

1. 底层信道完全可靠

不会发生错误、不会丢弃分组

2. 发送方和接收方的FMS（状态机）独立

Rdt2.0：产生位错误的信道

底层信道可能翻转分组中的位 (bit) , 不会丢包

+ 利用校验和检验位错误

如何从错误中恢复?

- 确认机制 (ACK) : 接收方显式地告知发送方已正确接收
- NAK: 接收方显式地告知发送方分组有错误
- 发送方收到NAK后, 重传分组

基于这种机制地协议是 ARQ (Automatic Repeat Request) 协议

Rdt2.1 和2.2

2.0的缺陷: ACK和Nak坏掉, 会发送方会重传, 即重复分组。

如何解决重复分组问题: 2.1

1. 发送方给每个分组增加序列号, 序列号只要0/1就够了
2. 接收方丢弃重复分组

Rdt2.1 : 接收方, 应对ACK和NAK破坏

Rdt3.0

假设信道既可能发生错误, 也可能丢失分组。

方法: 发送方等待“合理”的时间

+ 如果没收到ACK, 重传

- 如果分组或ACK只是延迟而不是丢了
 1. 重传会产生重复, 序列号机制能够处理。
 2. 接收方需在ACK中显式告知所确认的分组
- 需要定时器

对于ppt里最后的一个问题, 答案发送pkt0: 无论超时以后重复发送多少次分组, 只要收到一个正确的ack0无干扰回复, 状态机就从右上角的状态转移到右下角的状态, 此时无论再收到多少个ack0或ack1回复都不会再改变此状态, 此状态只由上层命令触发改变

