# 第五章 运输层

#一、运输层协议概述

1. 进程之间的通信
   运输层向高层用户屏蔽了下面网络核心的细节(如网络拓扑，路由选择协议等)，使应用进程看见的好像是两个运输层实体之间有一条端到端的逻辑通信信道
2. 运输层的两个主要协议
   用户数据报协议UDP：
   - 传输数据之前**不需要先建立连接**
   - 收到UDP报后**不需要给出任何确认**
   - 不提供可靠交付，但是这是最有效的工作方式
     传输控制协议TCP
   - **提供可靠的，面向连接的运输方式**
   - 不提供广播或者多播服务
   - **开销较多**
3. 运输层端口
   复用：应用进程都可以通过运输层再传送到IP层
   分用：运输层从IP层收到发送给应用进程的数据后，必须**分别交付给指明的各应用进程**
   为了指明各个应用进程，在运输层使用协议端口号，把端口作为通信的抽象终点，这种端口是软件端口。
   两个计算机进行通信的时候**不仅要知道对方的IP地址，而且还需要知道对方的端口号**

---

#二、用户数据报UDP

1. UDP概述
   UDP的主要特点：
   - 无连接，发送数据之前不需要建立连接
   - 使用尽最大努力交付，不保证可靠交付
   - 面向报文，UDP**一次传送和交付一个完整的报文**
   - 没有拥塞控制，网络出现拥塞不会使源主机的发送速率降低，很适合多媒体通信的要求
   - 支持一对一，一对多，多对一，多对多等交互通信
   - 首部开销小，只有八个字节
     应用程序必须选择合适大小的报文：
   - 如果报文太长，IP层在传送时可能会进行分片，降低IP层的效率
   - 如果报文太短，会使IP数据报的首部相对长度过大，降低IP层的效率
     UDP的复用和分用：

- 复用：将UDP数据报组装成不同的IP数据报，发送到互联网
- 分用：根据UDP数据报首部的目的端口号，将数据报分别传送到相应的端口，以便应用进程到端口读取数据
  接收方根据UDP端口号将报文通过端口交给相应的进程，如果发现目的端口号不正确，就丢弃该报文，并且由ICMP发送端口不可达给发送方

2. UDP的首部格式
   - 源端口：源端口号，需要回信的时候使用，不需要可全用零
   - 目的端口：目的端口号，终点交付报文时使用
   - 长度：UDP用户数据报的长度，其最小值是8
   - 校验和：检查UDP用户数据报在传输过程中是否有差错，如果有就丢掉
     **UDP校验和是将首部和数据部分一起进行检验**

---

#三、传输控制协议TCP概述

1. TCP最主要的特点
   - TCP是**面向连接**的运输层协议
   - 每一条TCP连接都只能有两个端点，TCP连接只能是**点对点的**
   - TCP提供**可靠交付**的服务
   - TCP提供**全双工通信**
   - 面向字节流：TCP中的流指的是流入或者流出进程的字节序列，虽然应用程序和TCP的交互是一次一个数据块，但是TCP把应用进程交下来的数据看成一连串**无结构的字节流**
2. TCP连接
   TCP连接的端点是套接字 socket:
   **套接字** socket=(IP地址：端口号)
   每一条TCP连接都唯一地被通信的**两个端点(套接字)所确定：**
   TCP连接 : :={socket1，socket2}={(IP1:port1),(IP2,port2)}

---

#四、可靠传输的工作原理

1. 停止等待协议
   停止等待
   发送方每发送完一个分组就停止发送，**等待对方确认**(ACK)，在收到之后再发送下一个分组
   全双工通信的双方既是发送方也是接收方

   超时重传
   **出现差错例如接收方收到分组的时候检查出了差错，直接丢弃该分组**，或者该分组在传输过程中就丢失了，这样就无法收到确认

- 发送方为每一个已经发送的分组设置一个超时计时器
- 只要在超时计时器到期之前接收到了相应的确认，就撤销该计时器，继续发送下一个分组
- 若在规定时间内没有收到确认，那就认为分组错误或者丢失，重发该分组

出现差错
**确认丢失**：若接收方接受到了对应的分组，但是回复的确认丢失了，发送方在超时计时器到期后会重传该分组，假定接受方又收到了该分组，就会采取两个行动：丢弃该分组和向发送方发送确认
**确认迟到**：接收方发送的确认迟到了，导致发送方又重传了分组，接收方会收到重复的分组，丢弃并且重传确认，**发送方接收到重复的确认会丢弃**

信道利用率

**信道利用率** $$U = \frac{T_D}{T_D + \mathrm{RTT} + T_A} \qquad \textbf{(5-3)}$$

停止等待协议的要点：
- 停止等待：发送方每次只发送一个分组，收到确认之后再发送下一个
- 暂存：发送完一个分组之后，发送方必须暂存已经发送分组的副本，以备重发
- 编号：**对每一个分组和确认都进行编号**
- 超时重传：设置一个超时计时器，如果超时后仍未收到确认，发送方自动重传分组
- 超时计时器的设置时长应该比分组传输的平均往返时间长点
- 简单但是信道利用率过低

2. 连续ARQ协议
发送窗口：发送方维持的一个发送窗口，**位于发送窗口的分组可以被连续发送出去**
发送窗口滑动：**发送方每次收到一次确认，就把发送窗口向前滑动一个分组的位置**
累积确认：**接收方对按序到达的最后一个分组发送确认**，表示该分组之前的所有分组均正确收到
连续ARQ采用的是回退N：**在发送分组出现错误的时候需要退回重传已经发送过的N个分组**

---

#五、TCP报文段的首部格式
TCP报文段首部的前20个字节是固定的，因此TCP首部最小长度为20字节

- 源端口和目的端口，各占两个字节，端口是运输层和应用层的服务接口，运输层的复用和分用功能通过端口实现
- 序号，TCP连接的数据流中每一个字节都有一个序号，该字段的作用则是指本报文所发送的数据的第一个字节的序号
- **确认号：期望收到对方写一个报文段的第一个字节的序号，如果确认号为N，则序号N-1之前的所有数据均已正确收到**

- 数据偏移(首部长度)：指TCP报文段的数据起始处距离报文段的起始处的距离多远
- 紧急URG：URG=1表示当前报文段中有紧急数据，应该尽快发送
- **确认ACK：当ACK=1的时候，确认字段才有效**
- 推送PSH：接收TCP收到PSH=1的报文后就应该尽快推送向前交付接收应用进程，不要等到缓存满后再交付
- 复位RST：RST=1表示TCP连接中出现严重错误，必须释放连接并且重新建立运输连接
- **同步SYN：SYN=1表示这是一个连接请求或者连接接受的报文段，当SYN=1，ACK=0时，表明这是一个连接请求报文段，当SYN=1，ACK=1时，表明这是一个链接接受报文段**
- **终止FIN：用来释放一个连接，FIN=1表示该报文段发送数据发送完毕，要求释放运输连接**
- 窗口：告诉对方从本报文的确认号算起，接收方目前允许对方发送的数据量，窗口值经常在动态变化
- 检验和：检验范围是首部和数据，需要在报文段前面加上12字节的伪首部
- 紧急指针：在URG=1的时候指出紧急数据在次报文段的字节数，指出了紧急数据末尾在报文段中的位置
- 最大报文段长度MSS：是TCP报文段中的数据字段的最大长度
- 时间戳：用于计算往返时间RTT或者用于吧新报文段和迟到的旧文段区分

---

## #六、TCP可靠传输的实现

1. 以字节为单位的滑动窗口
   **TCP使用流水线传输和滑动窗口协议实现高效可靠的传输**
   TCP的滑动窗口是**以字节为单位**的
   发送方和接收方分别维持一个发送窗口和接收窗口
   发送窗口：在没有收到确认的情况下，发送方可以连续将窗口内的数据全部发送出去，凡是已经发送过的数据，**在未收到确认之前都必须暂时保留，以便超时重传的时候使用**
   接收窗口：只允许接收落入窗口内的数据
   注意事项：
   - 第一，发送窗口是根据接收窗口设置的，但在同一时刻，发送窗口并不总是和接收窗口一样大（因为有一定的时间滞后）。
   - 第二，TCP 标准没有规定对不按序到达的数据应如何处理。通常是先临时存放在接收窗口中，等到字节流中所缺少的字节收到后，再按序交付上层的应用进程。
   - 第三，TCP 要求接收方必须有累积确认的功能，以减小传输开销。接收方可以在合适的时候发送确认，也可以在自己有数据要发送时把确认信息顺便捎带上。但接收方不应过分推迟发送确认，否则会导致发送方不必要的重传，捎带确认实际上并不经常发生。
2. 超时重传时间的选择
   TCP发送方在规定的时间内没有收到确认的就要重传已经发送的报文段
   重传时间设置不能太短。否则会引起很多报文段的不必要重传，使得网络的负荷增大，不能过长会使网络的空闲时间增大，降低了传输效率

采用加权平均往返时间$RTT_s$

$$\text{新的 RTT}_S = (1-\alpha) \times (\text{旧的 RTT}_S) + \alpha \times (\text{新的 RTT 样本}) \quad (5\text{-}4)$$

**其中，$0 \leq \alpha < 1$。**
**若 $\alpha \to 0$，表示 RTT 值更新较慢。**
**若 $\alpha \to 1$，表示 RTT 值更新较快。**
**RFC 6298 推荐的 $\alpha$ 值为 1/8，即 0.125。**

重传报文段后，无法判定确认报文段是对原来报文段的确认还是对重传报文段的确认，因此l在计算平均往返时间 RTT 时，只要报文段重传了，就不采用其往返时间样本。
超时重传时间：

- RTO (Retransmission Time-Out) 应略大于加权平均往返时间 $RTT_S$。
- RFC 6298 建议 RTO：

$$\text{RTO} = \text{RTT}_S + 4 \times \text{RTT}_D \quad (5\text{-}5)$$

其中：$RTT_D$ 是 RTT 偏差的加权平均值。

- RFC 6298 建议 $RTT_D$：

$$\text{新的 RTT}_D = (1-\beta) \times (\text{旧的 RTT}_D) + \beta \times |\text{RTT}_S - \text{新的 RTT 样本}| \quad (5\text{-}6)$$

其中：$\beta$ 是个小于 1 的系数，其推荐值是 1/4，即 0.25。

修正后的Karn算法：

- 报文段每重传一次，就把 RTO 增大一些：

$$\text{新的 RTO} = \gamma \times (\text{旧的 RTO})$$

- 系数 $\gamma$ 的典型值 = 2。

- 当不再发生报文段的重传时，才根据报文段的往返时延更新平均往返时延 RTT 和超时重传时间 RTO 的数值。

3. 确认选择SACK
收到的报文段没有差错，只是没有按照序号，中间还缺少一些序号的数据，使用确认选择SACK来实现针对缺少的数据的重传
根据序号确认字节的边界，左边界=第一个字节的序号，右边界=最后一个字节的序号+1

---

#七、TCP的流量控制

1. 利用滑动窗口实现流量控制
流量控制：让发送方的发送速率不要太快，利用滑动窗口机制可以很方便地实现在TCP连

接上对发送方流量的控制
A向B发送数据，在连接建立后B告诉A：**B的接收窗口rwnd的大小，随后根据接收窗口的大小通知A允许A发送数据的大小**
为了防止B对A的报文段丢失，A又持续等待B发送通知，就设置一个持续计时器，只要乙方的TCP连接收到对方的0窗口通知就启动该计时器，持续时间到就发送一个零窗口探测报文段，对方在确认该探测报文段后给出当前的窗口值，如果窗口值是0，则收到报文段的一方就重新设置持续计时器，否则就打破死锁僵局

2. TCP的传输服务
控制TCP发送报文段的时机：三种机制：
   - TCP 维持一个变量，它等于最大报文段长度 MSS。只要缓存中存放的数据达到 MSS 字节时，就组装成一个 TCP 报文段发送出去。
   - 由发送方的应用进程指明要求发送报文段，即 TCP 支持的推送 (push) 操作。
   - 发送方的一个计时器期限到了，这时就把当前已有的缓存数据装入报文段（但长度不能超过 MSS）发送出去

---

#八、TCP的拥塞控制

1. 拥塞控制的一般原理
出现拥塞的原因：
   - 节点缓存容量太小；
   - 链路容量不足；
   - 处理机处理速率太慢；
   - 拥塞本身会进一步加剧拥塞；
   出现拥塞的条件：

   $$\sum \text{对资源需求} > \text{可用资源} \qquad (5\text{-}7)$$

   拥塞控制与流量控制的区别：
   - 拥塞控制：防止过多的数据注入网络中，避免网络中的路由器或链路过载；是一个全局性的过程，涉及到所有的主机，路由器，以及与降低网络传输性能有关的所有因素
   - 流量控制：抑制发送端发送数据的速率，使得接收端来得及接收；点对点通信量的控制是一个端到端的问题
   开环控制与闭环控制：
   - 开环控制：在涉及网络的时候，实现考虑周全，力求**工作时候避免发生拥塞**，一旦系统运行的时候就不再中通进行改正
   - 闭环控制：**根据当前的运行状态采取响应的控制措施，消除拥塞**。TCP拥塞控制属于闭环控制
2. TCP的拥塞控制方法
TCP采用基于滑动窗口的方法进行拥塞控制，属于闭环控制方法
TCP的发送方维持一个拥塞窗口cwnd
拥塞窗口的大小取决于网络的拥塞程度，并且是动态变化的

发送方利用拥塞窗口根据网路的拥塞情况来调整发送的数据量
发送窗口的大小不仅取决于接收方窗口，还取决于网络的拥塞状况
真正的发送窗口值=Min(接收方通知的窗口值，拥塞窗口值)
发送方判断拥塞的方法：超时重传计时器超时，收到3个连续重复的确认
TCP拥塞控制算法流程：

- **慢开始**：从小到大逐渐增大注入网络中的数据字节，从小到大逐渐增大拥塞窗口的数值
  初始值设置为1个最大报文段MSS，设置慢开始门限ssthresh
  拥塞窗口增大：每收到一个新的报文段的确认，每经过一个传输轮次RTT，就把拥塞窗**口增加一倍变为原来的二倍**，窗口大小按照指数增加
- 拥塞避免：当拥塞窗口达到慢开始门限就开始拥塞避免算法，**让拥塞窗口缓慢增大，每经过一轮拥塞窗口cwnd值加1，按照线性规律增长**
  当拥塞避免算法执行过程中出现了网络发生拥挤，则重新启动慢开始算法
- 快重传：当发送方**连续收到三个重复的确认就立即对缺失的报文段进行重传**
- 快恢复：当执行快重传算法后，不执行慢开始算法，而是执行快恢复算法，将慢开始的门限设置为此时cwnd的一半，即ssthresh=cwnd/2，新的拥塞窗口为慢开始门限，然后执行拥塞避免算法

3. 主动队列管理AQM
   对TCP拥塞控制影响最大的是路由器的分组丢弃策略，采用的是先进先出的处理规则与尾部丢弃的策略
   主动队列管理AQM最流行的就是随机早期检测RED
   路由器队列维持两个参数：队列长度最小门限与队列长度最大门限。对每一个到达的分组都计算平均队列长度，
   如果平均队列长度小于最小门限，则将其放入队列进行排队；
   如果平均队列长度大于最大门限长度，则将新到达的分组丢弃；
   如果平均队列长度位于最小门限与最大门限之间则按照一定的概率p将新到达的分组丢弃

---

#九、TCP运输连接管理
TCP是面向连接的协议
TCP的连接建立有三个阶段：
- 连接建立
- 数据传输
- 连接释放
TCP的连接需要解决的三个问题：

- 要使每一方能够确知对方的存在
- 要允许双方协商一些参数
- 能够对运输实体资源进行分
  TCP连接的建立采用的是客户服务器方式

1. TCP的连接建立
   TCP建立连接的过程叫做握手
   采用三次报文握手：在客户和服务器之间交换三个TCP报文段，以防已经失效的连接请求报文段突然传送到了，从而产生TCP的连接建立错误
   - 服务器B的TCP进程首先创建传输控制模块TCB，准备接受客户进程的连接请求
   - **第一次握手**：A的TCP想B主动发送连接请求报文段，其中首部中的同步位SYN=1，并选择序号seq=x，表明传输数据的时候第一个数据字节序号是x，其中SYN报文段不能携带数据，但是需要消耗一个序号
   - **第二次握手**：B的TCP收到连接请求报文段之后，如果同意，需要发回确认，在确认报文段中使得SYN=1，ACK=1，确认号ack=x+1，自己的序号为seq=y，该报文段不携带数据，但是同样消耗一个序号
   - **第三次握手**：A收到B的报文段后向B给出确认，ACK=1，确认号ack=y+1，该报文段可携带数据，如果不携带数据则不消耗序号。此时A的TCP通知上层应用进程，连接已经建立
   - B收到A的确认后，也通知上层的应用进程，TCP连接双方已经确立，可以开始数据传送
2. TCP的连接释放
   数据传输结束后，通信的双方可以释放连接
   TCP释放连接的过程是四次报文握手：
   - A的应用进程先向TCP发送连接释放报文段，停止发送数据，主动关闭TCP连接
   - **第一次挥手**：A把链接释放报文段的首部的FIN=1，序号seq=u，等待B的确认，FIN报文段不携带数据也消耗一个序号
   - **第二次挥**手：B发出确认，ACK=1，确认号ack=u+1，报文段序号seq=v，TCP服务器进程通知高层应用进程，从A到B的连接就释放了，TCP连接处于半关闭状态，B发送数据，A仍然需要接受
   - **第三次挥手**：如果B没有向A发送的数据，则应用进程通知TCP释放连接，发送连接释放报文段，FIN=1，ACK=1，seq=w，确认号ack=u+1
   - **第四次挥手**：A收到连接释放报文段之后，必须发出确认，ACK=1，确认号ack=w+1，自己的序号seq=u+1
   - 此时的TCP连接还没有释放掉，经过时间等待计时器设置的2MSL之后，A释放TCP连接

---

#十、课后作业

1. **IP地址和端口号**可以唯一确定一个在互联网上通信的进程
2. **0~1023范围**内的端口号被称为”**熟知端口号**“并限制使用，这意味着这些端口号是为常用的应用层协议如FTP和HTTP等保留的
3. A与B建立了TCP连接，当A收到确认号为100的确认报文时，表示**未字节序号99的报文段**已收到
4. 在TCP协议中，发送方的窗口大小取决于**接收方允许的窗口和拥塞窗口**

5. A和B之间建立了TCP连接，A向B发送了一个报文段，其中序号字段seq=200，确认号字段ack=201，数据部分有2个字节，那么B对该报文的确认报文字段中seq=201，ack=202
    - A向B发送的报文段，seq=200说明A发送的报文的序号是200，确认号字段是201指的是A希望收到B序号为201的确认报文
    - B向A发送确认报文，seq=201，表示序号为201，ack=202，表示B已经收到A发送的所有数据，并且期望收到的下一个数据字节序号为202
6. 计算机网络的最基本的功能是**数据通信**
7. 计算机网络的资源主要是指：**计算机硬件，软件与数据**
8. 广域网的拓扑结构通常采用**网状**
9. 协议指的是**不同结点对等实体**之间进行数据通信的规则或约定
10. 局域网和广域网的差异不仅在于它们所覆盖的范围不同，还主要在于它们所使用的**协议**不同
11. **服务，接口，协议**是计算机网络中OSI参考模型的3个主要概念
12. 当数据由端系统A传送到端系统的时候，不参与封装工作的是**物理层**
13. OSI参考模型中，自下而上第一个提供**端到端服务的层次是传输层**
14. TCP/IP参考模型的网络层提供的是**无连接不可靠的数据传输服务**
15. DNS系统的作用是**将域名转换为IP地址**