

# 网络基础

## TCP/IP

1. 应用层：直接面向用户，为应用程序或用户、网络提供接口，为用户提供各种网络服务；将数据进行编码及格式转换，使一个系统的应用层发送的数据能被另一个系统的应用层识别；提供一种标准表示形式将计算机内部的多种数据格式转换成通信中采用的标准表示形式；提供数据压缩和加密等转换功能；接受来自传输层的数据，创建、管理、维护和终止表示层实体之间的通信会话，支持它们间的数据交换
2. 传输层：建立主机端到端的链接，为会话层和网络层提供端到端的数据传输服务，不能保证数据包在该层总是安全可靠地传输；提供一项UDP（用户数据报协议）的选择
3. 网络层：通过路由选择算法，为报文或通信子网选择最适当的路径；控制数据链路层与传输层之间的信息转发；建立、维持和终止网络的连接；将数据链路层的数据转换为数据包，然后通过路径选择、分段组合、顺序、进/出路由等控制，将信息从一个网络设备传送到另一个网络设备
4. 数据链路层：接受物理层位流形式的数据，封装成帧传送到网络层；接受网络层的数据帧，拆装成位流形式传送到物理层；建立和管理节点间的链路，通过各种控制协议，将有差错的物理信道变为无差错的、能可靠传输数据帧的数据链路
5. 物理层：为数据链路层提供物理连接，实现相邻计算机节点之间比特流的透明传输，屏蔽具体传输介质和物理设备的差异

## TCP

1. TCP连接只在端系统中进行，对中间路由器无效
2. 最大报文段长度MSS（报文段里应用层数据的最大长度）：由最大传输单元MTU设置以适应单个链路层帧
3. TCP报文段结构：
  1. 源端口号、目的端口号
  2. 序号字段、确认号字段-可靠数据传输
    1. 报文段序号-报文段首字节的字节流编号
    2. 确认号：期望从对方收到的下一字节的序号
      - 提供累积确认-只确认该流中第一个丢失字节
  3. 接收窗口字段-流量控制
  4. 首部长度字段
  5. 检验和字段
  6. 选项字段-协商MSS
4. 标志字段
5. 捎带：对客户到服务器的数据的确认被装载在一个承载服务器到客户的数据的报文段中
6. TCP仅为传输一次的报文段测量样本RTT（SampleRTT）
7. 新EstimatedRTT（均值RTT）=  $(1 - a) * \text{旧EstimatedRTT} + a * \text{SampleRTT}$   
 $a=0.125$
8. DevRTT(RTT偏差) =  $(1 - b) * \text{DevRTT} + b * (\text{SampleRTT} - \text{EstimatedRTT})$

b=0.25

15. Timeoutinterval ( 推荐超时间隔 ) = EstinMrtedRTT + 4\*DevRTT

初始为1s

16. 隐式NAK：收到三个冗余ACK就会在超时前触发对后面报文段的重传

17. 超时间隔加倍-定时器过期后超时间隔加倍，但收到上层应用的数据或收到ACK中的任意一个事件启动时定时器重新计算：提供拥塞控制

18. 快速重传：三个冗余ACK触发

19. TCP的差错恢复机制是GBN与SR协议的混合体

20. 流量控制服务：消除发送方便接受方缓存溢出的可能性

1. 接收方把当前的空闲空间值放入发给发送方接收窗口字段中
2. 接收方接收窗口为0时，发送方继续发送只有一个字节数据的报文段，接收方确认后，缓存开始清空后，确认报文中将包含非0的空闲值
3. UDP不提供此项服务

21. 拥塞控制：防止IP网络的拥塞

22. 三次握手：

1. 客户端TCP向服务器端TCP发送SYN报文段（一个特殊的TCP报文段），报文段首部一个标志位被置为1，客户端随机选择初始序号并将此编号置于该SYN报文段的序号字段中-封装到IP数据报中
2. 服务器提取SYN报文段-向客户端发送允许连接的报文段（SYNACK报文段）
3. SYN比特：1
4. 确认号字段：客户端初始序号+1
5. 序号字段：服务器选择自己的初始序号
6. 客户向服务器发送另外一个报文段，SYN比特：0

23. 关闭TCP连接：

1. 客户向服务器发送特殊TCP报文段，FIN比特：1
2. 服务器返回确认报文段
3. 服务器发送终止报文段，FIN比特：1
4. 客户返回确认

24. 接收到不接受连接的端口的连接请求时，发送特殊重置报文段，RST标志位为1

25. namp可以侦查打开的TCP、UDP端口；防火墙及其配置；应用程序的版本和操作系统

26. 拥塞控制方法：

1. 端到端拥塞控制
2. 网络辅助的拥塞控制：路由器向发送方提供关于网络中拥塞状态的显示反馈信息-拥塞分组/标记某个字段

27. TCP拥塞控制：

1. 拥塞窗口：对TCP发送方向网络中发送流量的速率进行限制
2. 未确认的数据量上限同时由流量控制、拥塞控制决定
3. 拥塞指示：丢包事件（超时/三个冗余ACK）
4. 自计时：TCP使用确认触发增大它的拥塞窗口长度
5. 确认报文段增加发送方的速率

6. 带宽探测-增加发送速率直至出现丢包事件

28. TCP拥塞控制算法-加性增，乘性减：

1. 慢启动-起始慢，但在慢启动阶段发送速率指数增长直至丢包事件发生，重新慢启动或进入拥塞避免模式或快速重传后进入快速恢复状态
2. 拥塞避免
3. 拥塞窗口cwnd大约是上次遇到拥塞时的一半，每个RTT只将cwnd的值增加一个MSS；
4. 超时，cwnd的值被设置为1个MSS；
5. 丢包事件出现时，慢启动门限（ssthresh）的值被更新为cwnd值的一半
6. 三个冗余ACK：cwnd值减半，将ssthresh的值记录为cwnd值的一半，进入快速恢复状态
7. 快速恢复-对收到的每个冗余的ACK，cwnd的值增加一个MSS
8. 丢失报文段的ACK到达：进入拥塞避免
9. 超时：cwnd的值被设置为1个MSS，进入慢启动
10. 丢包：cwnd的值被设置为1个MSS，并且ssthresh的值设置为cwnd值的一半

29. TCP会导致多条链接平等共享带宽-实际情况中，具有较小RTT的连接能享有更高的吞吐量

30. TCP套接字编程：

#### 客户端

1. 导入套接字模块
2. 设置服务器、通信端口
3. 指示地址簇、套接字类型
4. 创建TCP连接
5. 执行三次握手
6. 发送分组（无需附上目的地址）
7. 等待服务器响应
8. 关闭套接字

#### 服务器端

1. 导入套接字模块
2. 设置通信端口
3. 创建TCP套接字
4. 绑定端口
5. 创建欢迎之门（面向所有用户）
6. 聆听
7. 客户敲门后创建专属套接字
8. TCP连接创建成功
9. 响应客户

## URL

1. 协议protocol
2. 主机host：port
3. 端口号port：HTTP协议默认端口号为80
4. 文件路径path
5. 请求参数query
6. 定位位置fragment：定位网页元素