computer network review

ooooo

1 Lec1 概述

电路交换网络：电话网络

存储转发网络 中间节点收到报文全部内容后，才将报文发向下一个节点.

大消息占用时间长, 会阻塞其它消息

电路交换 发送发和接收方之间建立起一条专用物理链路用于交换，可时分，频分复用

以用户会话为调度单位

存储转发 (报文交换) 通信双方交换报文借助中间 1(or more) 个中间节点转发

以消息为调度单位

包交换 (分组交换) 通信双方交换的报文按照网络规定划分成若干个小包, 采用存储转发技术

以数据包为调度单位, 粒度更细, 连接共享特性更好, 效率更高

同一个消息不同包可同时在不同连接上传输, 小消息不必等待大消息.

报文交换与分组交换的主要区别 报文一整段是有意义信息, 而包没有完整意义.

2 Lec2 网络体系结构和模型

计算机网络 由若干地理上分散的, 具有独立功能的计算机系统利用各种通信系统互相连接起来而形成的计算机系统集合

网络协议要素 语法, 语义, 时序

层次结构特点 具有一定层次; 层次间单向依赖; 上层隐藏下层细节, 统一下层差异

水平通信 (对等实体) 虚通道

垂直通信 (相邻通道) 实际通道

服务分类

- 有连接服务 (面向连接服务)
- 无连接服务
  - 无确认
  - 有确认
  - 确认-应答

原语 (primitive) 通知服务者采用某些动作或报告服务用户对等实体采取某个动作

IOS 标准定义四种原语:*request, indication, response, confirm*

参数 用来传递数据和控制信息

服务与协议 服务: 垂直; 协议: 水平

标准化时机 在两个驼峰 (研究, 投资) 之间的低谷阶段制定标准

定得太早：在研究工作完成以前，人们对这一主题还缺乏理解，产生可能不好.

定得太晚：很多公司已经对不同的模式进行了大量的投资，实际上会无人遵守。

OSI 七层 应用层, 表示层, 会话层, 传输层, 网络层, 链路层, 物理层

TCP/IP 五层 应用层, 传输层, 网络层, 链路层, 物理层

3 Lec3 网络分类和网络性能

发送, 传输时延 发送时延 = 数据帧长度 (b)/发送速率 (b/s)= 数据块长度/信道带宽

传播时延 传播时延 = 信道长度 (m)/信道上传播速率 (m/s)

电路交换

- 优点
  - 实时性好
  - 数据传输稳定
  - 无信道访问延迟
- 缺点
  - 不能充分利用传输介质
  - 长距离连接建立过程长
  - 扩展性差

报文转发

- 优点
  - 减轻网络通信拥挤
  - 有效利用信道资源
  - 提供异步通信
- 缺点
  - 不满足实时性
  - 不适合交互式通信
  - 硬盘要大

包交换

- 优点
  - 数据包小, 实时性较好
  - 数据流分路, 更好利用带宽资源
  - 不怕链路故障
- 缺点
  - 存储-转发延迟, 排队延迟
  - 包丢失

时延带宽积 时延带宽积 = 传播时延 × 带宽

4 Lec4 物理层

分贝 (dB)

$$D = 10 \log_{10} \frac{P_1}{P_2}$$

P1:destination
P2:source

信噪比

$$R_{S/N} = 10 \log_{10} \frac{S}{N}$$

S: 平均噪声功率
N: 噪声功率

奈奎斯特 (Nyquist) 准则 离散无噪声数字信号, 信道容量:

$$C = 2W \log_2 L$$

W: 信道带宽 (Hz)
L: 代码采用进制数

香农 (Shannon) 定理 带宽受限, 且有高斯白噪声干扰的信道极限无差错信息传输速率:

$$C = W \log_2(1 + \frac{S}{N})$$

W: 信道带宽
S: 信道内信号平均功率
N: 信道内高斯噪声功率

不归零 (NRL-L) 0,1 高低

不归零反转 (NRZI) 1 变 0 不变

曼彻斯特编码 中间跳变,1 升 0 降

5 Lec5 物理层 2

多路复用

- 频分复用
- 时分复用
- 码分复用

6 Lec6 链路层 1

帧 (frame) 在数据链路上交换数据的单位。

形成数据帧 字节计数法, 字节填充的首尾定界法 (转义符 Esc), 位填充的首尾定界法 (0x7E)

流量控制 限制发送方发送速度的机制, 使发送速率不能超过接收方处理速度

停-等流量控制 源实体发送一个帧，目标实体收到后发回一个对该帧的确认，表示同意接收下一个帧；源实体必须等待收到确认后才能发送下一个帧。

线路利用率

$$U = \frac{t_{frame}}{2t_{prop} + t_{frame}}$$

t_{frame}: 发送一帧所需的时间 (传输时间)
t_{prop}: 传播延迟

线路的最大利用率

$$U = \frac{1}{1 + 2a}$$

a = 传播延迟/传输时间
其中:
传播延迟 = 链路距离/传播速率 = $\frac{d}{V}$
传输时间 = 帧长度/数据速率 = $\frac{L}{R}$

$$a = \frac{dR}{VL}$$

滑动窗口控制 利用窗口控制连续发送的数据量。
n = 1+2 传播延迟/传输时间

滑动窗口流量控制 肯定确认 (RRn), 否定确认 (RNRn)

管道化技术 发送端为达到信道最大效率必须连续不断地发送数据。

差错控制 指对传输的数据信息进行错误检测，并加以恰当的处理。

7 Lec7 链路层 2

顺序收发方式 接收方只能按照帧的序号接收数据帧。

回退 N ARQ 的发送窗口 最大发送窗口为:$W_T = 2^n - 1 = m - 1$

选择重传 ARQ(乱序收发方式) 只重发错误帧
收发窗口的最大值:
$m - 1 \geq 2w - 1$
$w \leq m/2 = 2^{n-1}$

汉明距离 一个码组集合中任意两个码组间的最小码距。

- 为了检出 e 个错码, 要求码集的汉明距离 $d \geq e+1$
- 为了纠正 t 个错码, 要求码集的汉明距离 $d \geq 2t+1$

编码效率 指一个码组中信息所占的比重，用 R 表示

$$R = \frac{k}{n} = \frac{k}{k+r}$$

k: 信息位长度
r: 校验位长度
n: 编码后码组总长

奇偶校验编码 先将所要传送的数据码元分组。在各组的数据后面附加一位校验位，使得该组码连校验位在内的码字中：

- 1 的个数为偶数, 偶校验
- 1 的个数为奇数, 奇校验

循环冗余校验码 (CRC)

8 Lec8 链路层 3

PPP 协议帧

0x7E(1)|0xFF(1)|0x03(1)|Protocol(2)| 信息部分 ($\leq$ 1500)|FCS(2)|0x7E(1)

协议字段

- 0x0021, 信息字段是 IP 数据报
- 0xC021, 信息字段是 PPP 链路控制数据
- 0x8021, 信息字段是是网络控制数据

字符填充法 将信息字段中出现的每一个 0x7E 字节转变成为 2 字节序列 (0x7D, 0x5E)。

若信息字段中出现一个 0x7D 的字节, 则将其转变成为 2 字节序列 (0x7D, 0x5D)。

若信息字段中出现 ASCII 码的控制字符（即数值小于 0x20 的字符），则在该字符前面要加入一个 0x7D 字节，同时将该字符的编码加以改变。

零比特填充 在发送端，只要发现有 5 个连续 1，则立即填入一个 0。接收端对帧中的比特流进行扫描。每当发现 5 个连续 1 时，就把这 5 个连续 1 后的一个 0 删除

介质访问控制 将传输介质带宽有效地分配给网上各站点用户的方法。

帧时 发送一个标准长度的帧所需时间

信道效率 所有已发送帧中有多少避开碰撞正确到达

纯 ALOHA 协议 不按时间槽, 不监听, 随机重发

分槽 ALOHA 协议 各用户节点只能在下一时间槽的起始时刻发送信息

载波侦听多路访问 (CSMA) 想要传输的站点首先听一听介质上是否有其他站点在传输,if 忙, 等待,else 传输

CSMA 协议的三种形式

- 1——坚持: 若介质空闲, 传输; 若介质, 一直侦听直到空闲马上传输; 若发生冲突, 等待一个随机长的时间.
- 非坚持: 若介质空闲, 传输; 若介质忙, 随机等一段时间, 重复此算法.
- P——坚持: 若介质空闲, 则以概率 p 传输, 以概率 (1-p) 把此次传输推迟一个时间槽; 若介质忙, 等待一个时间槽.

9 Lec9 链路层 4

CSMA/CA 如果介质为空, 则节点传输帧; 如果介质为忙, 则等待直到当前传输完全结束;

指数后退算法 竞争窗口初始化为某个最小值，发生冲突时加大窗口，直到达到最大值。

后退过程 当空闲时间 >=IFS, 立即传输; 当介质忙, 延迟直到当前传输结束 +IFS 时间; 开始随机后退过程: 选择一个随机数, 使用侦听确定每个时间槽是否有活动, 如果没有活动则减少 backoff 时间; 后退过程中如果介质为忙则挂起 backoff 过程; 在当前帧传输结束后恢复后退过程

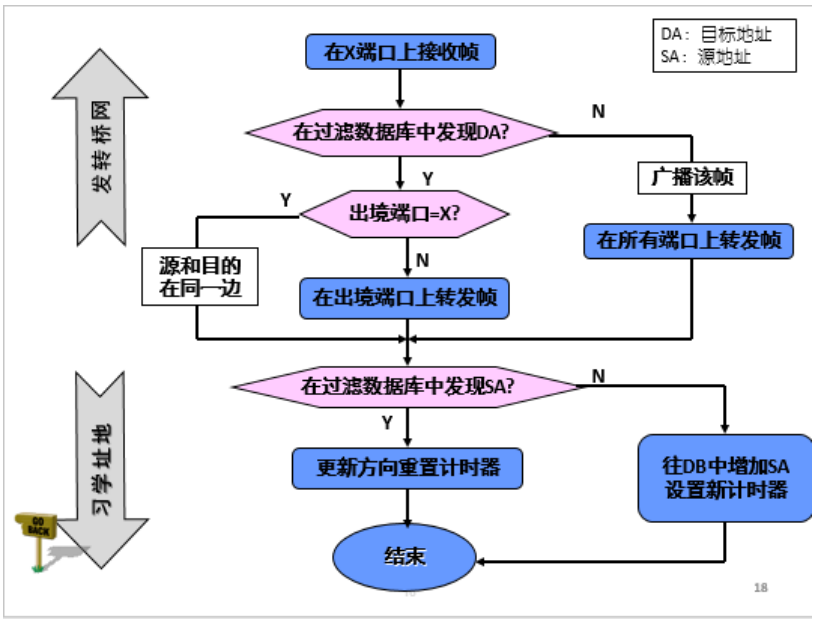
带有 RTS/CTS 的扩展 DCF

发送者发送 RTS, 接收者用 CTS
侦听到 RTS 发送者在附近
侦听到 CTS 接受者在附近

10 Lec10 链路层 5

网桥基本功能 过滤, 转发

过滤数据库 列出了 port 号与该 port 在同一边的站的地址信息。



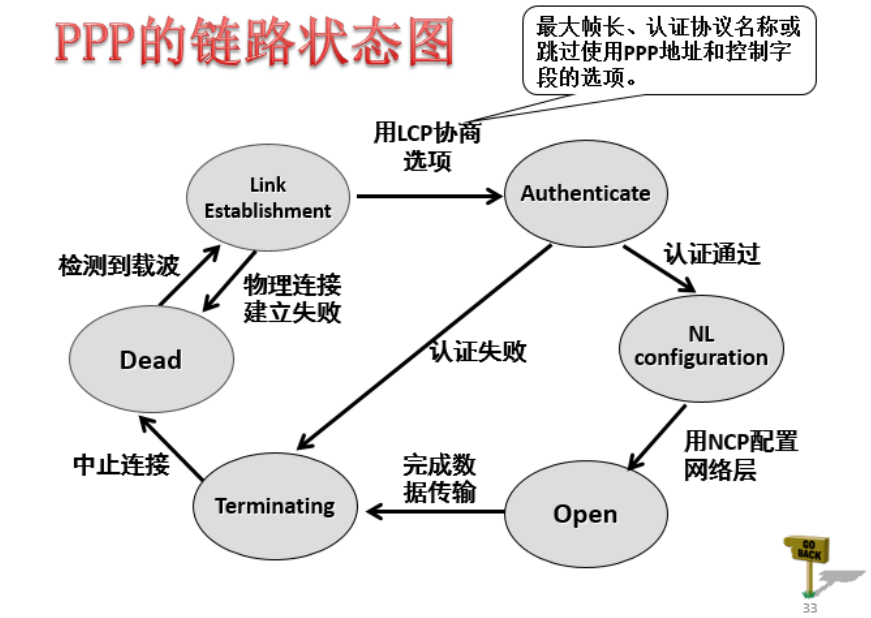
转换图 LAN 对应于节点；桥对应于边

PPP 一个可用于调制解调器、比特串行线路、SONET 和其他物理层的多协议成帧机制。

字节填充技术 在信息字段，若出现 flag 模式则在前面加一个转义字符 01111101；对于信息字段出现转义字符也作相同的处理。

LCP 帧 用来协商最大帧长、认证协议等。

NCP 帧 协商报头是否压缩；协商 IP 地址



11 网络层概述

功能 交换, 路由选择, 呼叫建立, 拥塞控制

端端通信 两个计算机系统传输实体之间的通信。

网络层是处理端端数据通信的最底层。

向传输层提供的服务：无连接的服务，面向连接的服务。

数据报子网 (实现) 每个数据报必须包含目的地的完整地址；路由器用一张表指出通向目的地的出境线路；当一个分组入境时，路由器查找路由表并将分组沿出境线路发出，无需修改分组中的任何内容。

虚电路子网 在传输数据之前需要有一个建立虚电路的过程，建立虚电路时选择一个当前未用的最低虚电路号。

中继器 放大电子信号的低级设备。它将来自一个接口的比特简单广播到所有的其他接口。用于连接以太制式的总线式网络，起到扩展网络连接距离的作用。

Hub “多端口中继器”将所有介质 (多段粗、细电缆或双绞线) 连接到一个中央位置的设备。

网络互连

- *物理层*: 中继器
- *链路层*: 网桥
- *网络层*: 路由器
- *传输层, 应用层*: 网关

桥路由 可同时作为网桥使用的路由器

协议网关 用于使用不同协议的网络间进行协议转换，由三种工作方式，分别对应于可在链路层、网络层、或二、三层之间进行转换。

分段技术 将大的包分成网络能容纳的一系列段，将每一段作为一个独立的包发送。

透明分段 前面的分段对后面的网络透明

不透明分段 任一中间网关都不重组，必要时只进行分段, 仅在目标主机进行一次重组

虚电路子网的互联 数据包沿着特定路径发送, 每个路由器负责中继包并在需要时进行包格式和虚电路号转换

数据报子网的互联 不同包选择的路由可能不同, 包的到达次序可能与发送次序不同

隧道技术 在两个端点建立传输数据报的虚拟管道，使所传输的数据报不为途径的节点所知, 采用封装技术

12 网络层路由

源路由 路由决策由源站点而非网络做出。

路由选择算法 给定一组路由器及连接路由器的链路，找出一条从源端到目标端的“好”路径。

静态路由算法 Dijstra 算法,flooding 算法

矢量距离算法 (DV)

$$D^x(Y,Z) = c(X,Z) + min_w\{D^z(Y,W)\}$$

节点 X 经过 Z 到达 Y 的距离= X 到邻居 Z 的距离+ Z 到 Y 的最短距离

主要数据结构：每个节点维护的距离表

一个节点能得到的信息：与其直接相连的链路的成本，来自邻接节点的路由信息。

链路状态路由算法 (LS) 每个节点都有网络的完整拓扑，每个节点维护到每个邻居的连通性与链路成本。利用 Dijkstra 算法计算最短距离。

RIP 内部网关协议, 基于 DV,UDP

要点

- 仅和相邻路由器交换信息
- 交换的信息是当前本路由器所知道的全部信息，即自己的路由表
- 按固定的时间间隔交换路由信息，例如，每隔 30 秒

OFPS 要点

- 向本自治系统中所有路由器发送信息 (flooding)
- 发送的信息是与本路由器相邻的所有路由器的链路状态
- 只有当链路状态发生变化时，路由器才用泛洪法向所有路由器发送此信息

BGP 不同自治系统的路由器之间交换路由信息的协议

边界网关协议 BGP 只能是力求寻找一条能够到达目的网络且比较好的路由（不能兜圈子），而并非要寻找一条最佳路由

13 网络层 IP

IP 无连接、不可靠的包传递服务

协议地址 协议软件定义一个与底层物理地址无关的编址方案，给每台主机分配一个唯一的地址。

IP 地址的层次结构使得路由算法针对网络进行路由成为可能。

IP 地址并不标识一台特定的主机，而是标识一台主机与网络的一个连接。

子网编址 在分类体系中增加一级，将主机号进一步划分成子网号和主机号。

子网掩码 用来确定子网划分的特殊比特模式；

无类域路由 没有地址分类以及子网划分概念的地址分配方法。

路由聚合 将具有相同网络前缀的地址合并成 CIDR 地址块

分段 当数据报的尺寸大于网络 MTU 时，路由器将数据报分成若干个较小部分—称为段 (fragment)。

重组 在所有段的基础上重新产生原始数据报的过程。

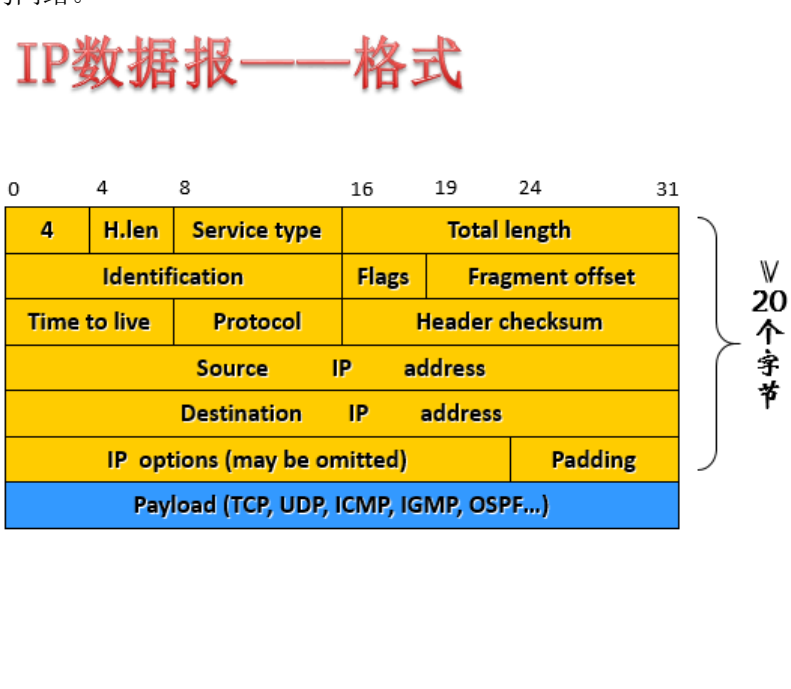
封装 将 IP 数据报装进一个帧的数据区，网络硬件像对待普通帧一样对待包含着数据报的帧。

严格源路由 两个相邻地址必须处在同一个网络上（规定了完整路径）

松散源路由 允许两个地址之间跳过多个网络（规定了部分路径）。

直接投递 指在一个物理网络上，数据报从一台机器直接被传送到另一台机器。

简介投递 发送方必须把数据报发送给某个路由器，由该路由器把数据报转发到目的网络。



14 网络层控制解析分配翻译

ICMP

当数据报产生差错时 ICMP 向数据报的源端回报差错情况

源端必须把差错交给一个应用程序或采取其他措施来纠正

特点:ICMP 不是高层协议,ICMP 不具备可靠性和优先级, 携带 ICMP 报文的 IP 数据报传递出现差错时不再报告, 携带 ICMP 报文的 IP 数据报与携带用户数据的 IP 数据报具有完全相同的路由选择

地址解析协议 (ARP)

ARP 的高速缓存: 用来存放最近获得的 IP 地址与硬件地址的绑定

ARP 优化: 在回答 ARP 请求后将请求消息中的发送方地址绑定信息加入自己的高速缓存

动态主机配置协议 (DHCP) 允许计算机快速、动态的获取 IP 地址

内联网 不与 Internet 相连的企业内部网络

虚拟专用网 (VPN) 利用隧道技术将内联网包封装成 Internet 上的 IP 包

NAT 协议 在私有地址和全局地址之间转换的协议

地址转换表 将处境包的 (源 IP 地址、Port 号) 替换成 (NAT IP、Port 号), 对于入境包，NAT 以目的 port 号作为索引查找转换表，以对应的源 IP/Port 置换回去, 转换表中的条目动态加入并在空闲超时后删除

Multi-Protocol Label Switching(MPLS) 使用定长标签进行高速 IP 转发

15 网络层移动, 组播

端系统移动处理 间接路由, 直接路由

Mobile IP 间接路由

组播 一次“发送”操作就可以把数据包从一个发送者传送到多个接收者

应用层组播 网络层没有组播功能, 发送者对每个接收者都采用一条单播传输

网络层组播 发送主机仅发送一个包, 一旦这个包需要转发到多条出境链路上时，网络路由器就复制该包的副本

IP 组播基本思想

1 组播发送者向组播地址发送数据包

2 组成员告知本网段的路由器他们需要接收哪些数据包

3 发送者和接收者之间的路由器构造组播树，确保组播数据包到达正确的接收者网络

组管理协议 (IGMP) 用户进程通过该协议提出加入/退出某个组的请求；组播路由器通过该协议了解本地哪些主机加入了哪些组。

组播路由算法 路由器之间共享组信息，为组播数据报的分发提供更好的路由

共享树 (ST) 在共享组播路由树中定义一个中心点 (或称为核心), 具有组播组成员的路由器向中心节点单播 join 控制报文

源树 (SPT) Dijkstra 算法计算从源到所有其他目标的最短路径，这些路径的集合形成一棵最小成本路径树

逆向路径转发 (RPF) 广播路由技术: 如果在通往 S 的正确路径上收到来自 S 的包，则转发到其他所有处境路线

在构造组播树的过程中，路由器接收到一个数据包，对它执行逆向路径转发检查, 路由器可以确保自己在组播树中入境的路径只有一条，并且是到发送者最优的那一条, 从全局来看则保证了构造的结果没有环路，是一棵树

修剪 (pruning) 一个接收到组播包的边缘 (叶子) 组播路由器，若其附接的主机没有属于该组播组，则给其上行流路由器发送一个 prune 消息, 如果一个路由器从其每个下行流路由器收到了一个修剪消息，则转发 prune 消息到上行流

嫁接 (grafting) 一个组播路由器收到其附接的主机发出的请求加入某个组的 IGMP 消息，则给其上行流路由器发送一个 graft 消息, 一个已在组播树中的路由器回复一个确认消息；不在组播树中的路由器则转发 graft 消息到上行流

组播协议 DVMRP(Distance-Vector Multicast Routing Protocol) 具有逆向路径转发、嫁接和修剪的“基于源的组播树”

修剪, 嫁接

组播协议 PIM(Protocol Independent Multicast)

域内组播协议 自治系统内部用来转发组播报的树

域间组播协议 自治系统之间用来传输组播数据报的树

16 传输层概述,UDP

传输层提供应用进程/程序间端到端的逻辑通信

传输层协议能提供应用的多路复用/分用服务、可靠数据传送、带宽保证及延迟保证等

传输层 根本原因在于网络不可靠

为什么要分层？ 网络层提供了主机之间的逻辑通道, 传输层提供了应用程序之间的端到端连接

服务

面向连接，可靠的服务 (TCP)

无连接，不可靠的服务 (UDP)

socket(套接字) 和 port(端口)

socket: 通信的标识

port: 传输层的服务访问点 (TSAP),0-65535, 知名端口 0-1023

多路复用/分用 (Multiplexing/demultiplexing) 传输层完成从主机通信到应用通信的过渡

发送方: 处理来自多个 socket 的数据段，添加相应的传输层协议头

接收方: 根据头信息将接收到的数据段分发给正确的 socket

面向连接的复用/分用

$$< Src\_IP, Src\_Port, Dest\_IP, Dest\_Port >$$

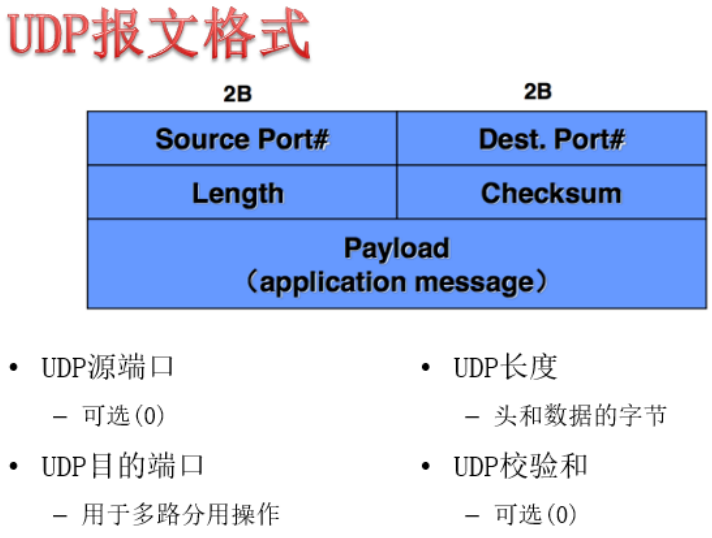
用户数据报协议 (UDP) UDP(User Datagram Protocol) 提供了不可靠的无连接传输服务。它使用 IP 传输报文，但增加了对给定主机上多个目标进行区别的能力。

最大 UDP 数据报长度:65535-20-8

特点

- 没有确认机制

- 不对报文排序
- 没有超时机制
- 没有反馈机制控制流量



校验和 加上伪头

UDP 端口的管理 TCP/IP 采纳了一种混合方式对端口地址进行管理：

- 对某些知名端口进行统一分配
- 为应用程序留下了很大的端口取值范围

17 传输层 TCP

rdt1.0 底层信道理想可靠

rdt2.0 基于误码数据通道 (差错检测, 接收端反馈, 重发机制),ACK 和 NAK 在传输过程中不会误码, 不会丢失

rdt2.1 处理误码的 ACK/NAK, 但不会丢失

rdt2.2 去除 NAK

rdt3.0 基于误码和丢包的数据通道, 滑动窗口, 回退 N, 选择重传

基于不可靠数据的连接建立：三次握手

序号选择的限制：序号不能进入禁止区域，发送太快从下方进入，发送太慢从上方进入。

连接建立时初始序号的选择:

基于时钟方法: 每台机器的始终采用二进制计数器的形式, 连接建立时用时钟的低 k 位作为初始序号

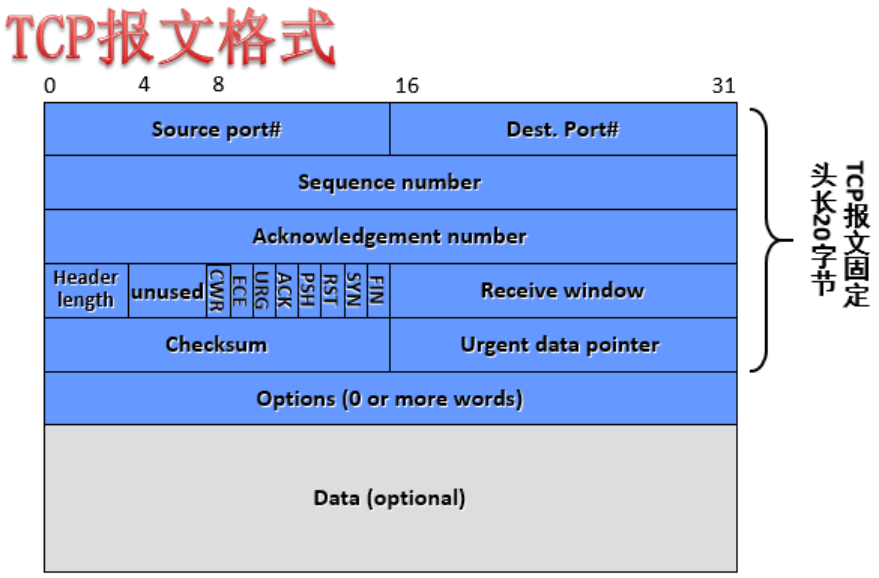
解决方法: 确保两个序号相同的报文永远不会同时有效, 主机恢复后等待 T 秒 (T 是报文生存期的倍数，用来确保报文发出去 T 时间后不再存在), 限制对序号的使用

两军对垒问题 最后发出信息的蓝军指挥官永远无法确定信息是否安全到达对方。

释放 三次握手

传输控制协议 (TCP) TCP(Transmission Control Protocol)：可靠的面向连接的端-端字节流传输协议。特点

- 面向连接
- 全双工
- 连接是点-点
- 有缓冲的发送
- 无结构的数据流



TCP 的接收数据特性 为每个字节编号, 确认号为等待的下一个字节, 采用累计确认, 缓存到达的乱序数据

MSS(maximum segment size) 连接两端位于同一物理网络, 选择的 MSS 应使 IP 数据报的大小与网络 MTU 适应; 连接两端位于不同物理网络, 最好设置为途径网络的最小 MTU 或省缺值 (536 字节); 连接两端必须协商 MSS

窗口扩大因子 移位数表示窗口大小扩大的位数 (至多 14), 收到窗口通告时要左移才能获得实际的窗口大小, 发送窗口通告时要右移, 该选项必须在连接建立时协商, 每个方向上的扩大因子可不同

TCP 连接建立 采用“三次握手”方法, 采用基于时钟的序号产生方案, 双方协商本次连接的初始序号

TCP 连接释放 采用“三次握手”方法, 每个方向连接单独释放, 超时值设定为 2 倍的 MSL

TCP 可靠性 TCP 没有否定确认机制，接收端只能通过重复确认来报告出错情况。

快速重传 发送端检测到三个重复 ACK 立即重传该 ACK 所指的段而不是等待该段超时后再重传。

流量控制 TCP 采用大小变化的滑动窗口协议, 由接收端通过 window size 字段反馈当前可接收的字节数

$$LastByteSent–LastByteAckd \leq RcvWindow$$

就能保证发送端不会淹没接收端

$$RcvWindow = RcvBuffer - [LastByteRcvd–LastByteRead]$$

优化

Nagle 算法 如果已传数据未确认之前发送端应用程序又生成了额外的数据，则照常将数据放入输出缓冲区，但并不发送；直到缓冲区中的数据足够填满一个 MSS；如确认到达后发送端仍处于等待状态，则发送缓冲区中累积的所有数据。

Clark 方法 通告 0 窗口后，要等到缓冲区可用空间达到总空间的一半或 MSS 之后才发送“窗口更新”通告。

18 传输层 TCP 拥塞控制

结论一：当包的到达率接近链路容量时将产生长的排队延迟。

结论二：发送端必须重传因缓存溢出而丢失的包

结论三：延迟增大发送端重发被延迟的包导致链路容量消耗在转发重复包上

结论四：当一个包在传输路径上被丢弃，所有转发过该包的路由器所做的工作都是白费的。

拥塞 太多的发送源端给网络发送太快太多的数据, 当路由器接收包的速度大于它们转发包的速度，便会发生子网拒绝额外的包进入拥塞区域, 被拥塞的路由器可丢弃队列中的包以便腾出空间存放新到达的包

包交换网络本质是排队网络

流量控制与拥塞控制

流控只与发送者和接收者之间的点-点通信有关

拥塞控制是全局问题

控制论的拥塞控制

开环: 通过良好设计避免拥塞的发生。

闭环: 通过反馈获得网络当前状况做恰当的调整。

拥塞控制机制的分类

端 - 端的拥塞控制: 网络层不提供对传输层拥塞控制的显式支持, 端系统必须由网络行为推断拥塞发生的发生

网络协助拥塞控制: 路由器在检测到拥塞时为发送端提供反馈信息

拥塞控制

显示通知:

抑制包:HOP-HOP 抑制包, 每一跳都降速

公平队列: 路由器为每个发送端设置一个发送队列，依次从每个队列中取出一个包发送

卸载: 当路由器被包所淹没时只能将包丢弃

早期丢弃 Random Early Discard (RED): 每当到达一个包，RED 算法就计算平均队列长度 avg, 如果 avg 低于某些低阈值，就假定发生拥塞的概率很小或者说不存在拥塞，将该包排入队列, 如果 avg 大于某些高阈值，则认为拥塞很严重，丢弃该包, 如果 avg 介于两个阈值之间，可认为拥塞正在形成，计算拥塞发生的概率

通信量整形: 将进入网络的通信量“整理形状”的途径，以此来预防（因突发流量造成）拥塞的发生。

接收窗口（RcvWin）：接收方根据自己缓冲区大小通告接收窗口的大小信息。

拥塞窗口（CongWin）：大小体显了网络的承受能力。

有效窗口兼顾网络和接收能力：min(接收窗口，拥塞窗口)

TCP 拥塞控制 TCP 发送端通过调整 CongWin 来限制发送速率

基本思想: 当发生丢失事件时，降低发送速率（减少拥塞窗口 CongWin 的大小）

所有通过壅塞区域的发送端都降低发送速率, 注入拥塞路径的流量减少, 由此缓解壅塞状况

缓速启动 TCP 连接建立时，拥塞窗口 CongWin 初始为一个 MSS, 初始的发送速率 = CongWin/RTT. 在初始阶段按指数增长速度 (每个 RTT 后 CongWin 大小加倍) 加大发送速率直到发生“丢包事件”. 发生“丢包事件”后将 CongWin 窗口减半并按线性速度增大 (每个 RTT 后 CongWin 加大一个 MSS).

AIMD 算法 逐步递增: 每当收到一个 ACK 就将 CongWin 窗口增大一个 MSS; 加倍递减: 一旦发现丢失段立即将拥塞窗口（CongWin）大小减半 (最后减到 1).

拥塞避免（congestion avoidance） TCP 拥塞控制协议的线性增加阶段。

超时 发送端进入“慢速启动”阶段，拥塞发送窗口置为 1 个 MSS, 按指数增长直到 CongWin = 发生超时时的一半, 按线性增长

三个重复 ACK 拥塞窗口减半, 按线性增长

19 应用层域名流媒体传输

Client − Server (CS) 服务器始终运行 (服务进程)，拥有永久地址, 客户端发起和服务器的通信，可能动态地址, 客户端之间不直接通信,request/response

Peer − to − Peer （P2P） 没有特定的服务器角色，节点地位对等, 节点之间互相请求服务，在客户端和服务器的角色间转换

域名查询

递归查询: 当主机/名字服务器 A 向 B 查询，B 将代表 A 执行查询请求，并将响应结果返回给 A

迭代查询: 当主机/名字服务器 A 向 B 查询，B 将返回下一个 DNS 名字服务器的 IP 地址给 A

前向纠错 适用于应用程序, 定期构造奇偶包 (P)：由数据包生成

交错编码 发送前将媒体流混合或者交叉编码, 接收端做相反操作还原原始媒体流

消除抖动 抖动：报文到达的延迟差, 应对网络延迟的变化方式

RTP 每个发送端可指定自己的独立 RTP 流, 大多数编码技术可将音频和视频数据编码在一个流中, 可用于“一对多”或者“多对多”通信

实时传输控制协议 RTCP 将当前 RTP 会话带宽的 5% 用于 RTCP 消息, 其中全部发送端占用 25%，全部接收端占用 75%

20 应用层流媒体,QoS

Internet 流量分类

弹性流量：指那些延迟和吞吐量变化很大仍然能满足应用需求的流量

非弹性流量：不能适应延迟和吞吐量变化的流量

调度机制

FIFO

优先级队列

公平队列

监管机制

漏桶: 将主机用户进程输出的不规则包流转换为输入网络的匀速包流, 主机与网络的接口为一个漏桶, 漏桶就是一个有限的内部队列

令牌桶: 只有在桶不为空时才能发送包, 每发送一个包桶内令牌数减 1, 桶满时新产生的令牌将被丢弃。

综合服务 IntServ 对单个的应用会话提供服务质量的保证 (资源预留, 呼叫建立)

有保证的服务 保证一个分组在通过路由器时的排队时延有一个严格的上限。

受控负载的服务 可以使应用程序得到比通常的“尽最大努力”更加可靠的服务。

资源预留协议 RSVP IntServ 的信令协议。

区分服务 DiffServ (DS) 针对不同类别的流量做不同的处理
特点: 可扩展性, 服务方式灵活

服务等级协定 SLA(Service Level Agreement) 指明了被支持的服务类别（可包括吞吐量、分组丢失率、时延和时延抖动、网络的可用性等）和每一类所容许的通信量。

边界路由器 分类器, 通信量调节器 (标记器, 整形器, 测定器)

每跳行为 PHB(Per-Hop Behavior) 每跳”是强调这里所说的行为只涉及到本路由器转发的这一跳的行为，而下一个路由器再怎样处理则与本路由器的处理无关。

迅速转发 (Expedited Forwarding) 指明离开一个路由器的通信量的数据率必须等于或大于某一数值。因此 EF PHB 用来构造通过 DS 域的低丢失率、低时延、低时延抖动、确保带宽的端到端服务。像点对点连接或“虚拟租用线”，又称为 Premium 服务。

确保转发 (Assured Forwarding) AF 用 DSCP 的比特 0 2 将通信量划分为四个等级，并给每一种等级提供最低数量的带宽和缓存空间。对于其中的每一个等级再用 DSCP 的比特 3 5 划分出三个“丢弃优先级”。当发生网络拥塞时，对于每个等级的 AF，路由器首先把“丢弃优先级”较高的分组丢弃。

21 网络安全

主动攻击 更改信息和拒绝用户使用资源的攻击 (中断, 篡改, 伪造)

被动攻击 截获信息的攻击 (截获)

安全四个方面 保密性, 真实性, 完整性, 不可否认性

替代密码 例: 错位密码

置换密码 按照某一规则重新排列消息中的比特或字符顺序。

序列密码 序列码体制是将明文 X 看成是连续的比特流 (或字符流)x1x2…，并且用密钥序列 K = k1k2…中的第 i 个元素 ki 对明文中的 xi 进行加密，即

$$E_K(X) = E_{k1}(x1)E_{k2}(x2)\dots$$

分组密码 将明文划分成固定的 n 比特的数据组，然后以组为单位，在密钥的控制下进行一系列的线性或非线性的变化而得到密文。这就是分组密码。

对称密钥算法 加密密钥与解密密钥是相同的密码

数据加密标准 DES 在加密前，先对整个明文进行分组。每一个组长为 64 bit。然后对每一个 64 bit 二进制数据进行加密处理，产生一组 64 bit 密文数据。最后将各组密文串接起来，即得出整个的密文。使用的密钥为 64 bit（实际密钥长度为 56 bit，有 8 bit 用于奇偶校验)。

高级加密标准 AES

公开密钥算法 公开密钥密码体制使用不同的加密密钥与解密密钥，是一种“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。

RSA 公开密钥密码体制 RSA 公开密钥密码体制所根据的原理是：根据数论，寻求两个大素数比较简单，而将它们的乘积分解开则极其困难。

加密算法: 若用整数 X 表示明文，用整数 Y 表示密文（X 和 Y 均小于 n），则加密和解密运算为：

$$Y = X^e \mod n$$
$$X = Y^d \mod n$$

密钥的产生:

1 计算 n。用户秘密地选择两个大素数 p 和 q，计算出 n = pq。n 称为 RSA 算法的模数。明文必须能够用小于 n 的数来表示。实际上 n 是几百比特长的数。

2 计算 $\phi(n)$。用户再计算出 n 的欧拉函数 $\phi(n) = (p-1)(q-1)$

3 选择 e。用户从 $[0, \phi(n)-1]$ 中选择一个与 $\phi(n)$ 互素的数 e 作为公开的加密指数。

4 计算 d。用户计算出满足下式的 d

$$ed = 1 \mod \phi(n)$$

作为解密指数。

5 得出所需要的公开密钥和秘密密钥：

公开密钥（即加密密钥）PK = {e, n}

秘密密钥（即解密密钥）SK = {d, n}

报文鉴别 (message authentication) 在信息的安全领域中，对付被动攻击的重要措施是加密，而对付主动攻击中的篡改和伪造则要用报文鉴别。

报文摘要 MD (Message Digest) 发送端将报文 m 经过报文摘要算法运算后得出固定长度的报文摘要 H(m)。然后对 H(m) 进行加密，得出 $E_K(H(m))$，并将其追加在报文 m 后面发送出去。

密钥分配 设立密钥分配中心 KDC (Key Distribution)，通过 KDC 来分配密钥。

链路加密

端到端加密

安全套接层 SSL(Secure Socket Layer)

IP 安全 (Security) 协议 IPsec 鉴别首部 AH (Authentication Header), 封装安全有效载荷 ESP (Encapsulation Security Payload)

安全关联 SA(Security Association) 在使用 AH 或 ESP 之前，先要从源主机到目的主机建立一条网络层的逻辑连接。此逻辑连接叫做安全关联 SA。