计算机网络

5. 网络层

网络层的功能

网络层：在发送主机和接收主机之间传输报文。

发送端：将报文封装为数据报文。

接收端：将接收到的报文交给传输层。

路由器：对经过的数据分组，检查头部信息，为其选择合适的输出端口并决定移动的路径。

网络层的主要功能：**转发**forwarding，路由器将分组由输入端口移送到合适的输出端口。**路由**routing，决定分组由源主机到目的主机的路径。

OSI的网络层：提供面向连接的虚电路服务，和无连接的数据报服务。

虚电路VC：从源到目的通路的类似于电话网的电路，性能优异。在数据传输前建立呼叫，结束后拆除电路；每个分组携带VC标识，每个路由器维护源主机到目的主机的连接状态，链路和路由器资源分配给VC，专用。

VC呼叫需要信令协议，用于建立、维持和拆除VC。

互联网Internet不采用VC。

VC的信息：由源主机到目的主机的路径；VC标识，在每条路径的每个链路上唯一，属于某个VC的分组，携带该VC的VC标识；路径上各个路由器，保存转发表，说明带各个VC标识的分组如何转发；每段链路上的VC标识可以改变，新的VC标识来自于转发表。

逻辑信道：一条链路上可以建立多条逻辑信道，一条VC由多段链路上的逻辑信道级联形成。VC标识可以使用**逻辑信道号LCN**表示，由于LCN只有局部意义，并且可以存放在路由器中，因此分组头开销和处理复杂度奖励。

数据报：Internet模式的网络层，没有呼叫建立的过程，没有网络层“连接”的概念；路由器不维护端到端的连接状态，而是基于目的地址进行路由选择，相同的源地址-目的地址的分组，可以选择不同的路径。

数据报的特点：各个分组独立选择路径，有利于充分利用网络资源，遇到节点或链路故障也容易解决。分组头需要包含地址字段，开销变大。各个分组路径不同，可能出现先发后到的现象。分组必须具有生存期，超过生存期则抛弃，避免在网络内死转耗费资源。

	VC服务	数据报服务
想法	可靠通信应当由网络来保证	可靠通信应当由用户主机来保证
连接	必须建立连接	不需要连接
地址	仅仅在连接建立阶段使用，此后每个分组使用一个短的虚电路号	每个分组都必须有完整的地址字段
路由方式	属于同一VC的分组由同一路由转发	每个分组独立选择路由进行转发
路由器或链路故障	所有通过故障节点或链路的VC均故障	故障节点可能丢失分组，一些分组的路由可能发生变化
分组顺序	总是按照发送顺序到达	不一定按照发送顺序到达
差错控制和流量控制	可以由网络负责，也可以由用户主机负责	由用户主机负责

路由器

路由器：路由器是一个具有多个端口输入，多个端口输出的专用计算机，其任务是转发分组，即由输入端口收到分组后，按照分组的目的地，从合适的输出端口转发给下一跳路由器。

路由器结构：路由器在分组转发的层面上，具有多个输入和输出端口，同时可以对使用转发表，对分组进行处理；在路由选择的层面上，路由器具有路由选择处理机，其中具有路由选择协议的程序，以及路由表的数据结构。

转发：路由器根据转发表，将用户的分组从适当的端口转发出去。

路由选择：按照分布式算法，根据从相邻路由器得到的关于网络拓扑的变化情况，动态地改变要选择的路由。

$$\text{路由选择算法} \Rightarrow \text{路由表} \Rightarrow \text{转发表}$$

输入端口：收到分组后，首先在链路层剥去帧首部和尾部，之后将其送至网络层的队列中排队，等待在交换结构中查表转发。

输入队列溢出：路由器处理分组的速率低于分组进入队列的速率，使得队列的存储空间减少到零，最终后续分组被直接丢弃。输入队列溢出是造成分组丢失的重要原因之一。

输出端口：先用队列缓存交换接收接收的分组，之后再链路层加上帧首部和尾部，再交由物理层发送到外部线路。

输出队列溢出：路由器处理分组的速率高于分组出转发队列的速率，使得队列的存储空间减少到零，最终后续分组被直接丢弃。输出队列溢出是造成分组丢失的重要原因之一。

交换结构：常见的交换结构包括通过存储器，通过总线，和通过互联网络进行。

通过内存交换：输入分组被复制到系统内存中，经过处理再从内存进行输出。存储器总线带宽限制了速度，若总线带宽为B，则转发吞吐量为$\frac{B}{2}$

$$\begin{array}{c}\text{内存}\\ \text{输入} \Updownarrow \text{输出}\\ \text{输入端口} \Longleftrightarrow \text{输出端口}\end{array}$$

通过总线交换：输入分组经过共享总线到达输出端口。由于总线需要竞争，因而交换速度受限于总线速度。

通过网络实现交换：用于克服总线带宽的限制。使用阵列结构，使得任何一个输入端口可以在几乎任何时刻向任何一个输出端口进行交换。

线路前部（HOL）阻塞：排队的分组必须等待通过交换结构。假设输入端口1和2的等待队列分别为{A,...}和{A,B,...}，他们的队首分组竞争输出端口A，假定端口1的A分组竞争得到输出端口，则端口2的B分组必须等待A分组，B被HOL阻塞。

高速交换机制：将变长的分组切割成固定长度的单元（cell），在发送到输出线路前，再将cell成组为分组。

虚拟输出队列VOQ：输入端对不容的输出建立FIFO，到达的cell在对应输出排队。每个时隙开始时，中心调度算法检测所有输入端口队列，找到不冲突的输入输出端口匹配。

优先级机制：根据紧急度设定优先级，优先级高的有线接入阵列交换；使用协议限制高优先级的业务流量，使得高优先级的业务量小。

加速原理：零阵列板的运行速度高于外部连线速率，N倍于外部线速率，则称之为加速N。

对于N端口的设备，若要保证没有数据分组在输入端排队，则至少需要加速N。

缓冲区长度：平均缓存量B = 平均往返时延RTT × 链路容量C

近期建议为：若N个流经过链路，则$B = RTT \times C/\sqrt{N}$

路由算法

路由选择：根据分组的目的地址选择路径。数据报方式下，每个分组都要在途径的节点上单独选路，VC方式下，在建立连接时进行选路。

路由选择性能评估：一般要求路由选择代价最小，最简单的性能评估准则是最小跳数。

泛洪Flooding：不需要网络信息。源节点直接将分组发送给邻近节点，节点接收分组后，在除接收链路之外的所有链路上转发；最终分组的多个备份将到达目的节点，每个分组由唯一序号以消除重复分组；节点可以记录哪个分组曾经被泛洪转发过，从而控制负载。

Flooding尝试所有可能的路由，非常稳健；可以获得最小跳数路由；可以用于建立VC；网络负载很重。

Dijkstra最短路径算法：集中式，路由器有全部的拓扑及链路代价的完整信息，可以通过链路状态广播获得，所有节点具有相同的信息。

$c(x,y)$：节点x到节点y的链路代价

$c(i,i) = 0$

$c(i,j) = \infty$，说明两节点没有直连

$c(i,j) > 0$，说明两节点直连

$D(v)$：从源结点到节点v的最小代价路径的代价，迭代得到

$p(v)$：从源节点到节点v演最小代价路径的前一个节点

N'：算法处理的节点集合，若节点v的最小路径已知，则$v \in N'$

```
Dijkstra算法，以u为源节点计算最短路径
N' = {u}
FOR v IN 所有节点 DO
    IF v为u的邻居节点 THEN D(v) = c(u,v)
    ELSE D(v) = infinity
LOOP
    找到D(w)最小，且不再N'中的节点w，将w加入N'
    FOR v in 与w相邻但不再N'中的节点 DO
        D(v) = min(D(v), D(w)+c(w,v))
UNTIL 所有节点都在N'中
```

距离矢量B-F算法：异步，分布式，路由器仅有邻居节点的局部信息，各个节点的信息不同，并且独立计算。

$$
\begin{aligned}
d_x(y)&：从\ x到\ y的最小代价路径的代价\\
BF方程&：d_x(y)=min_v\{C(x,v)+d_v(y)\},\ v是\ x的邻居节点\\
D_v&：经过\ x的邻接点\ v到\ y的距离矢量，\ D_v=[D_v(y)|y\in N]
\end{aligned}
$$

```
B-F算法，对每个节点
LOOP FOREVER
    等待从邻居节点获得本地链路代价的消息，获得Dv
    更新本地距离矢量，Dx(y)←minv{c(x,y)+Dv(y)|y∈N}
    若距离矢量变化，通知所有邻居节点
```

路由协议

路由信息协议RIP：基于距离矢量的分布式路由协议。定义路由器到直连网络的距离为1，到非直连网络的距离为经过的路由器数目加1，即距离为跳数，跳数最大值为15，为16时表示不可达。相邻路由器交换路由表信息（包括目的网络，跳数，和下一跳路由器），按固定的时间间隔进行路由交换。

RIP距离矢量算法

```
收到相邻路由器地址为X的RIP报文
修改此RIP报文：将下一跳路由器字段改为X，将所有距离字段加1
对RIP报文中的每一行，进行如下操作
    IF 表项的目的网络不在路由表 THEN 将其加入路由表
    ELSE IF 下一跳路由器地址相同 THEN 用收到的表项替换原表项
    ELSE IF 收到表项中的跳数更小 THEN 更新路由表
    ELSE 忽略
IF 超时（3min）还未收到相邻路由器的更新路由表 THEN
    将其标记为不可达，距离置为16
```

RIP协议要求网络不超过15跳，适合中小型网络。

距离矢量路由DV：RIP为其的一种特殊形式。每个路由器定期向其邻居节点发送距离矢量用来更新路由表；每个路由器收到所有邻居节点的距离矢量后，取$\min_i\{c(x,i)+D_v(i)\}$更新自己的路由表。

距离路由矢量，存在好消息（加入节点或节点启动）传得快，坏消息（节点停机）传得慢的问题。若没有最大跳数限制，则某节点突然停机后，所有路由器都将无限更新下去。当网络拓扑结构变化，距离矢量路由需要很长时间才能达到收敛并稳定下来。

解决方案：禁止路由器向邻居返回一个从邻居获得的最佳路径（去除成环的更新），或将无穷大路由器跳数设置为最大跳数加1。

链路状态路由LS：每个路由器完成如下五个工作，1) 发现邻居节点，2) 测量链路代价，3) 构造链路状态分组，4) 分发链路状态分组，5) 计算新的路由。

1．发现邻居节点：路由器启动时，在每一条点到点线路上发送HELLO数据包，对端路由器回复其全局唯一的名字。
2．测量链路成本：可以自动设置成ISP配置的度量，或者发送ECHO消息获得往返时延作为链路成本。
3．构造链路状态分组：链路状态分组中包括本路由器的唯一标识的名字，递增序号，年龄Age，以及邻居列表，邻居列表中每一项都是本路由器的邻居的唯一标识名和其链路成本的元组。
4．分发链路状态分组：使用泛洪法发送链路状态分组，每个路由器收到状态分组后检查分组序号和年龄，若年龄为0则丢弃，若序号小于最近一次收到的来自于同一个节点的状态分组序号，则说明收到过时分组，丢弃；一般，所有链路状态数据分组都要被确认以保证正确性；完成接收后，路由器向除接收端口外的所有端口发送状态分组，并将发送的状态分组年龄减1。
5．计算新路由：路由器累积了整个网络的链路状态分组后后，便可以构建完整的网络拓扑图，从而可以使用Dijkstra等最短路径算法得到路由选择的最短路径。

开放最短路径优先OSPF：LS的一种变体。其分组类型有Hello，链路状态描述，链路状态请求，链路状态更新和链路状态确认五种。

1．确定可达性阶段：发送Hello分组
2．链路状态同步阶段：泛洪扩散链路状态描述，构建网络拓扑图。
3．新情况下的同步阶段：每隔一段时间（30min），通过链路状态请求+链路状态更新+链路状态确认，更新链路状态和网络拓扑图。

OSPF数据报很短，路由信息量少；互联网规模很大时，效果明显优于RIP；没有无穷计算问题（坏消息传得慢）。

IP和ARP

IP分组格式：首部+数据。其中首部包括固定部分和可变部分，固定部分占20字节，包括版本，首部长度，服务类型，总长度，表示，生存时间，首部校验和，源地址，目的地址等字段。

校验和计算：将首部按16位字分割，逐字算数求和，结果取低16位加上溢出各位，再对16位字取反，得到首部校验和。

校验和检验：同计算相同，将校验和和首部各字求和，结果取低16位加上溢出各位，但不进行取反，若结果为0则通过校验，否则出错。

IP地址：连接在因特网上的主机/路由器的唯一标识。IPv4占32位，IPv6占128位。IP地址通过因特网域名和地址分配机构ICANN管理。IP地址编址有分类IP地址，子网划分，构成超网等方法。

IP地址标志了主机与链路的结构，路由器至少连接两个网络，因此具有两个以上的IP地址。

分类IP地址：将IP地址分为A~E五类，每类地址都由固定长度的网络号和主机号字段组成。

$$
\begin{aligned}
&IP\text{地址} \leftarrow \{< \text{网络号} >, < \text{主机号} >\} \\
&A\text{类地址：首位为}0\text{，网络号}8\text{位，主机号}24\text{位} \\
&B\text{类地址：首位为}10\text{，网络号}16\text{位，主机号}16\text{位} \\
&C\text{类地址：首位为}110\text{，网络号}24\text{位，主机号}8\text{位} \\
&D\text{类地址：首位为}1110\text{，特殊的多播地址} \\
&E\text{类地址：首位为}1111\text{，特殊的保留地址（留作以后使用）}
\end{aligned}
$$

常用三类IP地址：A类地址中，最大的网络数为126 (2^7-2)，因为0.x.x.x和127.x.x.x都是特殊地址，不能使用；B类地址中，最大的网络数为$2^{14}-1$；C类地址中，最大的网络数为$2^{21}-1$。**三类IP中，主机号全1均表示广播。**

同一网络上的主机和路由器，其IP地址网络号必须相同，主机号必须互不相同。路由器具有至少两个IP地址（视接口数目而定），每个接口的IP地址网络号互不相同。

特殊IP地址：0.0.0.0，表示本网络的本主机，用于DHCP；0+主机号，表示本网络中的主机；全1+全1，表示在本网络上进行广播；网络号+全1，表示对该网络所有主机广播；127+非全0或非全1，用于本机软件环回测试。

直接交付和间接交付：主机A向B发送分组时，检查B是否与其位于同一网络，若是则直接交付给主机B；否则，间接交付，将分组发送给本网络路由器，由路由器负责转发。

发送主机A判断是否位于同一网络，判断的标准是A和B的网络号是否相同。

IP地址和MAC地址：网络层之上传输IP分组，使用IP地址；链路层上传送数据帧，使用MAC地址。必须建立IP地址和MAC地址的映射关系。

地址解析协议ARP：解决同一局域网上主机的IP地址与网卡的MAC地址之间的映射关系，使得可以根据IP地址找到其对应的MAC地址。

```
1. 主机广播ARP请求分组到整个网络，要求查询某IP的MAC地址。
2. 相应主机向请求主机发送相应分组，回答其MAC地址。
```

反向地址解析协议RARP：已知主机的MAC地址，而要找到IP地址。

ARP高速缓存：主机储存IP地址到MAC地址的映射表，减少广播ARP请求带来的代价。

ARP解决同一个网络上主机或路由器IP地址和MAC地址的映射问题，在不同网络之间不需要使用ARP。

划分子网：A类和B类网络中，主机数过多，会存在ARP广播风暴的问题。对其进行修改，在IP地址中增加子网号字段，子网号是对网络本地部分的再次划分。

$$IP_{\text{地址}} \leftarrow \{<\text{网络号}>,<\text{子网号}>,<\text{主机号}>\}$$

子网掩码：为1的部分表示子网地址，为0的部分表示主机地址，也可以用/n表示，n为子网地址的位数。eg. 255.255.255.0，其可写作/24掩码。

无分类编址CIDR：消除了A、B、C类地址及划分子网的概念，使用网络前缀代表地址中的网络号和子网号。CIDR使用斜线计法如下，斜线左侧为IP地址，右侧为子网掩码。

$$IP_{\text{地址}} \leftarrow \{<\text{网络前缀}>,<\text{主机号}>\}$$
$$\text{斜线记法：} x.x.x.x/n$$

路由聚合：将网络前缀相同的连续的IP地址组合成CIDR地址块，从而极大地压缩路由表。路由聚合也称作**构成超网。**

前缀匹配的转发表：由于转发表表项很多，因而使用前缀匹配的方式压缩转发表项，加快查表速度。

最长匹配原则：选择最长的匹配前缀项。

路由器的分组转发算法

```
提取分组的目的IP地址D，得到目的网络地址N
IF N与此路由器直连 THEN 直接交付
ELSE IF 路由表中有D的特定路由 THEN 转发至路由表项标明的下一跳
ELSE IF 路由表中有到达N的路由 THEN 转发至路由表项标明的下一跳
ELSE IF 路由表中存在一个默认路由 THEN 转发至默认路由的下一跳
ELSE 使用ICMP报告转发分组出错
```

使用子网掩码的分组转发算法

```
提取分组的目的IP地址D
IF 存在网络的子网掩码与D匹配 THEN 直接交付
ELSE 间接交付
    IF 路由表中有目的地址为D的特定路由 THEN 转发至路由表项标明的下一跳
    ELSE IF 存在路由表中子网掩码与D匹配 THEN 转发至路由表项标明的下一跳
    ELSE IF 路由表中存在一个默认路由 THEN 转发至默认路由的下一跳
    ELSE 使用ICMP报告转发分组出错
```

IP控制协议

DHCP协议：允许计算机加入新的网络，并且自动获取IP地址。存在DHCP服务器，专职分配IP地址；主机启动时，发送广播报文DHCP DISCOVER寻找DHCP服务器，服务器以DHCP OFFER进行相应，之后二者通过DHCP REQUEST和DHCP ACK和DHCP NACK等报文进行通信。DHCP服务器在数据库中查找主机的配置信息，若找到则返回其IP地址，否则从IP地址缓存区分配一个地址给该主机。

DHCP代理：DHCP代理知道DHCP服务器的IP地址等，主机向DHCP代理广播DHCP DISCOVER，DHCP代理通过单播方式将其转发至DHCP服务器。

地址租期：分配给主机的IP地址具有有效期，逾期则失效。

网关：路由器的IP地址

全球地址：全球唯一的IP地址，必须向ICANN申请才能获得。

专用地址：仅在内部使用的IP地址，无需申请。专用地址只能用于内部通信，路由器不对目的地址是专用地址的分组进行转发。

网络地址转换NAT：专用地址经过NAT路由器后，被转换成NAT路由器所有的全球地址。将若干个专用地址转换为一个全球地址，从而实现多个专用地址对一个全球地址的共享，缓解了IP资源紧缺的压力。

```
NAT，主机X使用专用地址IPx，经过NAT路由器（IPg），同主机Y（IPy）通信
1. NAT路由器将源地址IPx转换为IPg，利用IPx和TCP端口得到NAT表索引N。
2. 替换分组中的源地址为IPg，TCP端口号为N，重新计算IP头和TCP头的校验和，发送至Internet。
3. NAT路由器接收到主机Y发回的数据报，通过TCP端口N，查表进行逆转换。
4. 转发给内部主机X，完成通信。
```

隧道技术：路由器将一种网络层协议报文封装到另一个网络层协议报文中。发送路由器进行封装，接受路由器解开分组并取出原始分组；传输过程中中间路由器并不在意封装的分组是什么。

隧道技术可以用于实现虚拟专用网，组播网络，IPv4和IPv6互联（将IPv6分组转入IPv4分组中）。

避免分段的策略：使用非透明分段，需要尽量发现路径上的MTU，使得传输的分组数降低。

最大传输单元MTU：分组的最大长度。与硬件，OS，协议，期望的分组出错和重传次数，避免分组占用信道时间过长。

透明分段：路由器对分组分段，并重组。路由器开销很大。

非透明分段：路由器在传输超过其MTU的分组时对其分段，目的主机负责重组。增加了传输的分组数，需要更多的传输时间。

$$分段头部：分组号 + 该分段在分组中的偏移量 + 分段结束标志$$

标志位：IP首部中的3bit的字段，最低位为MF（More Fragment），表示还有后续分片，中间位为DF（Don't Fragment），标志不能进行分片。

网络拥塞：因网络过载，而导致性能严重下降。发生拥塞时，信道利用率降低，缓存队列长度变长甚至溢出，分组丢失率上升。

流量调节：具有向源主机发送抑制报文，显式拥塞通知ECN，逐跳后压，随机早期检测RED等方法。

向源主机发送抑制报文：路由器产生抑制报文，沿数据流反向传送到源节点，之后源节点限制传输速率。该方法需要的时间长。

逐跳后压：在拥塞沿路传输抑制报文，路由器减缓发送速率。快速缓解拥塞，但会消耗路由器缓存。

显式拥塞通知ECN：在TCP协议层等加入一个通知字段，用于显式地告知是否发生拥塞，从而使得源节点限制传输速率。

随机早期检测RED：计算平均队列长度$E[N_q]=(1-\alpha)E[N_q]+\alpha N_i$，根据$E[N_q]$选择是否丢包。RED和ECN本质相同。

$$\begin{aligned}
&若\ E[N_q]>N_{max}，则丢弃新包\\
&若\ E[N_q]<N_{min}，则新包入队列\\
&若\ E_{min}<E[N_q]<N_max，则新包以概率p丢弃\\
&p=\frac{\delta}{1-c\delta},\ \delta=\frac{E[N_q]-N_{min}}{N_[max]-N_{min}}
\end{aligned}$$

因特网报文控制协议ICMP：允许主机或路由器报告差错情况并提供差错报告。ICMP报文作为IP层数据报的数据，加上数据报首部，组成IP分组发送。

> ICMP差错报告报文：目的地不可达，超时，参数问题（无效的头字段），源站抑制（拥塞控制），路由重定向（发生错误路由）。
> ICMP询问报文：回显请求和回显应答（检查主机是否工作），时间戳请求和应答（带时间戳的回显），路由器询问和通告（发现附近的路由器）。

使用两个IP地址的问题：当改变IP地址时，应用程序会与网络连接中断，因为原有的socket套接字失效；若采用DHCP动态分配IP，则IP改变会出现通信中断-再连接的问题。

IP组播

IP组播：实现一点到多点的，实时的信息交付，减少多个单点通信的网络资源的消耗。eg. 网络电视。

D类组播IP地址：D类IP开头为1110，剩余28位可供分配，这其中前5位不用来组成以太网硬件地址，因此可供分配的组播地址为23位。

以太网硬件组播地址：以太网地址为48位，首字节为01表示组播，第23位直接来自于D类IP地址。

网际组管理协议IGMP：使路由器获得组播组的成员信息。使用三类报文，主机发送的报告和离开报文，以及组播路由器发送的探寻报文。

主机加入或离开：主机加入时，向组播组对应的某个D类IP发送IGMP成员报告报文，本地组播路由器收到IGMP保温后，将组成员关系转发给其他组播路由器；主机离开组播组时，发送IGMP成员离开报文。

组播路由器维护成员信息：组播路由器周期性发送组探询报文，查询组成员状态，某个组中只要有一个主机响应，则认为该组活跃。

组播路由选择：找出以源主机为根节点的组播转发树。不同的组播组对应于不同的组播转发树，同一个组播组对不同的源点也有不同的组播转发树。

基于生成树的组播转发：路由器使用泛洪法转发组播数据报，采用**反向路径转发**RPF策略避免兜圈子。

RPF：路由器收到组播数据报时，检查是否是从源点经过最短路径送来的，若是，则向进入方向之外的其余方向Flooding，否则丢弃。若有多条最短路径，则选择IP最小的路由作为最短路径。

基于核心树的组播转发：所有路由器都同意将某个路由器作为核心，成员发送数据分组以建立整棵生成树。若发送者距离核心较近，则基于核心树的分发最优；若发送者距离核心较远，则将发送者作为核心最优。

移动主机路由

移动IP：允许移动主机可以在任何地方使用home IP，不允许修改固定主机的软件，不允许改动路由器软件和各个表，发给移动主机的大多数数据包不应该绕道，移动主机在home时不应该有开销。

MA：移动站点。HA：home站点。FA：移动主机外出，当前所在位置附近的站点。

代理：移动主机外出时，通过FA与HA取得联系；移动主机在home时，HA截获本地分组。

代理发现：通过ICMP广播消息，HA和FA通告移动业务。MA发现后，可以进行注册。

隧道技术：HA与移动主机通信的报文直接封装在HA与FA通信的报文中，有FA进行中间传递。

自组织网络路由

Ad Hoc网络：没有固定的基础结构，网络拓扑结构可变，所有节点均可以移动；由主机、移动节点组成的无线网络；节点间路由多跳；设备小型化低功耗，通信能力有限。

AODV路由协议：假设链路双向对称。S节点要发送分组给D，但没有节点D的路径，则启动路由发现过程，源节点泛洪RREQ（Route Request）分组；其他节点收到RREQ后再广播，同时建立它到源节点的反向路径；目的节点D收到RREQ时，响应RREP（Route Reply），RREP沿着RREQ转发过程中建立的反向路径进行转发。

目的地序列号DSN：确定路径信息新旧并进行选择更新，在RREQ，RREP和RERR中均有DSN。S到D发送新的RREQ，为其分配一个更大的DSN；若中间节点有到D的路由，若其DSN比RREQ小则不能发送RREP，否则可以发送。

超时：RREQ转发过程中建立的路由表项维持的是D到S的临时的反向路径，超时则应该直接删除。路由表项维持的S到D的前向路径，如果在active_route_timeout内没有被使用，也直接删除。超时机制降低了保存路由表的内存开销。

激活节点：如果在active_route_timeout间隔内，通过路由表的相应端口转发分组，则认为对应的节点是一个相邻节点，并且处于激活状态。

链路中断：节点的路由表的下一跳节点链路中断，则使用RERR（Route Error）通知所有激活节点。产生RERR时节点X根据缓存的目的节点D的DSN，加一设置RERR；各个节点收到RERR后，转发并删除路由表项；S收到RERR后启动一次新的RREQ转发，DSN再次加1；D收到新的RREQ，DSN更大，设置新的DSN。

6. 传输层

传输层的功能和服务

网络层：分组交换，数据报方式存在不可靠性。

传输层：为应用层提供面向连接的，或无连接的服务，并进行差错及流量控制。

应用层：提供各类业务的服务，要求保证可靠性和实时性。

传输层的功能和服务：寻址，定位应用程序；复用，在一个主机上支持多个任务；建立连接和释放，提供面向连接的服务；差错和流量控制；拥塞控制；崩溃恢复。

寻址：每个主机仅有一个NSAP（标识主机）但具有多个TSAP（标识应用程序），应用程序。

复用和反向复用：主机只有一个地址，但有多个应用，因而必需应用复用地址；若主机有多条网络露营，而应用网络需求很高，则必须轮询使用，一个应用反向复用各条网络路径。

并发操作：服务器为多个用户提供服务。有固定监听和查号台等方案。

可靠信道：传输层通过TCP协议，向应用层提供全双工可靠信道。

不可靠信道：传输层通过UDP协议，向应用层提供不可靠信道。

UDP：在传送数据之前不需要建立连接，传输层收到UDP报文后，也不需要发送确认。

TCP：提供面向连接的可靠的服务，需要确认消息。开销大，算法复杂。

端口：端口号为16bit，在进程间通信中用于标识进程。

熟知端口：0-1023，默认用于一些常见的服务。一半端口：随时分配给请求通信的进程。

套接字：IP地址与端口组成了socket套接字，在网络编程时使用。

$$Socket \leftarrow \{IP_{\text{地址}}, _{\text{端口号}}\}$$

TCP连接：一个TCP连接唯一地被通信两端的两个套接字所确定。

$$TCP \leftarrow \{Socket1, Socket2\} = \{(IP_1, Port_1), (IP_2, Port_2)\}$$

UDP

用户数据报协议UDP：在IP数据报之上增加端口和差错检测功能，没有流量和拥塞控制，没有重传机制。

UDP的优点：不需要建立连接，简单易于实现；网络拥塞时也不降低发送速率，适合传输实时业务，短时突发业务；首部开销小，只有8byte。

UDP用户数据报结构：首部8字节，包括2字节源端口，2字节目的端口，2字节长度，2字节校验和。其中**首部校验和时需要用伪首部和首部和数据段拼接在一起计算**。

伪首部：12字节，包括4字节源IP，4字节目的IP等。伪首部不在用户数据报结构中，仅仅是计算首部校验和时使用。

DNS系统

域名：由于地址不方便记忆，使用不方便，如果业务更换了服务器，IP地址也会改变，因此使用方便理解记忆和使用的域名。一个主机上可以有多个域名，多个服务器也可以共用一个域名用于负载平衡。

域名系统DNS：域名是主机的别名，采用层次结构，规则如下

$$\cdots.\text{三级域名}.\text{二级域名}.\text{顶级域名}$$

域名服务器：采用分布式结构，由若干个域名服务器负责域名到IP地址的解析。

顶级域名：.cn，.us，.com，.net，.org等。

域名服务器层状结构：根节点为根域名服务器，其下为顶级域名服务器，包括org域名服务器，com域名服务器等；再向下为权限域名服务器，例如baidu.com域名服务器；最下层为本地域名服务器，作为代理直接接收并转发主机的DNS查询。

根域名服务器：知道所有顶级域名服务器及其IP地址，全球共有13个（受UDP限制，最多13个根域名服务器），名字是a-m，域名分别为a.rootservers.net，...，m.rootservers.net。

DNS查询：当本地DNS对域名无法解析时，首先求助根域名服务器；根域名服务器把下一步要找的顶级域名服务器的IP地址告知本地DNS；之后本地DNS迭代查询，直到完成域名解析。

递归查询：主机向本地DNS的查询是递归查询，本地DNS代替主机处理域名解析，直到得到完整答案再返回。本地DNS到根也可以采用递归查询，但因为很多DNS服务器不支持递归查询，因而很难实现。

迭代查询：本地DNS到根DNS的查询是迭代查询，根DNS收到本地DNS的迭代查询时，给出IP地址或下一步要查询的域名服务器。根DNS的查询只返回部分答案，并移动到下一次查询地过程，有本地DNS继续发起下一次请求。

DNS缓存：减轻网络负载和DNS服务器压力。每个DNS服务器都维护一个cache，存放最近使用过的名字，以及从何处获得名字映射信息的记录，该记录具有计时器，并且会设置合适的超时时间。

DNS记录：一条DNS记录，包括name，value，type和TTL四个字段。TTL是超时时间，若超时则本记录删除。

$$\begin{aligned}
Type = A &\Rightarrow name = \text{主机名}, value = IP \\
Type = CNAME &\Rightarrow name = \text{别名}, value = \text{该别名对应的规范化名字} \\
Type = NS &\Rightarrow name = \text{域名}, value = \text{负责该域名解析的} DNS \text{服务器的域名} \\
Type = MX &\Rightarrow name = \text{邮件服务器别名}, value = \text{规范化名字}
\end{aligned}$$

NSLOOKUP：实现域名解析的命令行工具。

可靠传输和面向连接

链路层：相邻节点间的链路，存在错帧丢帧问题，但没有错序问题。

链路层连接：两次握手连接请求CR+收到确认ACK，即可保证节点之间状态同步。

链路层差错和流量控制：帧校验和帧序号，定时和超时重传，滑动窗口协议。

网络层：非相邻节点之间的分组交换。

网络层VC方式：两次握手，建立连接，之后数据沿同一路径传输，保证无差错。

网络层数据报方式：无连接服务，没有呼叫建立过程，基于目的地址对分组进行路由选择。保证无差错，但存在拥塞、丢失、错序等问题。

数据报方式建立连接：使用序号区分重复的段，三次握手保证序号同步。

> 第一次握手：主机1发送CR，初始序号为X
> 第二次握手：主机2回复ACK，确认收到主机1的序号X，并告知自己的初始序号为Y
> 第三次握手：主机1回复DATA，初始序号为X，并确认主机2序号为Y

序号回绕：对分组使用序号进行标记，当序号用尽时回到最小，并再次递增，若在序号生存期内，出现无法区分序号是用尽前还是用尽后重启的序号的情况，则出现序号回绕。

避免序号回绕：序号数目S，发送速率C，生存期T0，满足下式则不会出现序号回绕。

$$T_0 < \frac{S}{C}$$

基于时间戳的序号：使用使用设置初始序号，容易受到基于时钟预测的攻击，猜对初始序号即可建立伪造连接。最初的TCP方案。

伪随机初始序号：现有TCP采用的方案，保证一段时间内的序号一定不重复。

数据报方式释放连接：任何一方发送连接释放请求，等待另一方确认。

> 第一次挥手：主机1发送DR，请求释放连接，同时启动定时器，超时则再次发送DR，重发若干次后，关闭连接
> 第二次挥手：主机2回复DR，确认释放连接，启动定时器，超时则关闭连接
> 第三次挥手：主机1回复ACK，确认连接释放，主机1和主机2连接断开

TCP

TCP：端到端的，可靠的，按序的字节流的，全双工的，面向连接的连接；在其上可以进行流量和拥塞控制。

TCP首部：固定长度为10byte，包括16bit源端口和16bit目的端口，32bit序号和32bit确认号，首部长度（单位是32位字），6bit标志位，16bit窗口，16bit检验和，16bit紧急指针。此外还有长度可变的选项。

序号：报文段中发送数据的第一个字节的序号。

确认号：期望收到的后续报文段数据的第一个字节的序号。

URG标志：置1表示报文段为紧急数据。

ACK标志：置1表示确认号有效。

PSH标志：置1表示尽快交付，不要等到缓存满后再交付。

RST标志：置1表示TCP连接出错，需要重新建立连接。

SYN标志：置1表示连接请求或链接接受。

FIN标志：置1表示释放连接。

窗口：收方设置发方的窗口，单位为字节。

校验和：校验首部，数据和**伪首部**。

紧急指针：报文段中紧急数据放在报文段数据最前面，紧急指针说明其字节数。

选项：最多40byte，32位字对齐。包括MSS（最大报文段数），用于增大报文，提高传输效率；大窗口，用于设置大时延带宽积的信道上的大窗口，用于提高效率；时间戳，用于计算RTT防止序号回绕；SACK，用于告知发送方已经接受的序号范围。

TCP连接的三阶段：连接建立，数据传送，连接释放。

连接建立：要使每一方都知道对方的存在，要允许双方协商参数，要对传输实体分配资源。三次握手

```
第一次握手：主机A主动连接，处于SYN_SENT状态，发送SYN，SEQ=x
第二次握手：主机B确认接收序号，处于SYN_RCVD状态，发送SYN+ACK，SEQ=Y，ACK=x+1
第三次握手：主机A确认接受序号，处于ESTABLISHED状态，发送ACK，SEQ=x+1，ACK=y+1
连接建立：主机B主语ESTABLISHED状态，连接建立。
```

连接断开：要使双方都知道对方离开，要释放传输实体占用的资源。四次挥手

```
第一次挥手：主机A主动关闭，处于FIN_WAIT_1状态，发送FIN，SEQ=u
第二次挥手：主机B确认收到，处于CLOSE_WAIT状态，发送ACK，SEQ=v，ACK=u+1，主机A收到后处于FIN_WAIT_2状态
第三次挥手：主机B关闭连接，处于LAST_ACK状态，发送FIN+ACK，SEQ=v，ACK=u+1，
第四次挥手：主机A确认收到，处于TIME_WAIT状态，发送ACK，SEQ=u+1，ACK=v+1
连接断开：主机A处于TIME_WAIT状态，主机B处于CLOSED状态，连接断开。
```

流量控制：滑动窗口机制，接收端窗口为0时，发端被阻塞。阻塞状态下，发端等待窗口不为0（有死锁危险），或者发送紧急消息或1字节段（强制收端通告窗口）。

低效窗口问题：收端缓存区每有变化，就会通告窗口；而当数据块很大时，没读1byte，就会通告一次窗口为1，影响效率。

Clark解决方案：禁止收端通告1byte窗口更新，而只有当窗口为建立连接时所通告的最大数据段或收端缓存的1/2时，才通告窗口更新。

TCP提高效率的方案：发端不发送太短的数据段，收端不通告太小的窗口。

选择确认SACK：从累计确认中无法判断字节流之中哪些已达而那些丢失，SACK允许给出3个已经接收的字节范围。eg. ACK2 SACK3,4,6表明丢失了2和5。

TCP拥塞控制

拥塞原因：源主机太多，并且向网络发送数据的速率过高，导致网络内部拥塞。

理想的带宽分配：满足**最大最小公平性**，分配一个流的带宽时，若不减少另一个流的带宽则无法增加该流的带宽。

最大最小公平算法：给每个用户分配其期望且可满足的最小资源，然后将资源均分给需要更多资源的用户。保证每个用户得到的不会比其需要的更多，即使需求不被满足，也不会比其他用户里最多的资源还少。

```
用户1,...,n需要资源x1,...,xn，且x1<=...<=xn，服务能力C：
均分C/n给每个用户，记作C1,...,Cn
FOR Ci IN {C1,...,Cn} DO
    IF Ci > xi THEN
        FOR Cj IN {C(i+1),...,Cn} DO
            Cj <- Cj + (Ci - xi) / (n - 1)
RETURN C1,...,Cn
```

拥塞检测：拥塞时会发生分组丢失（路由器上队列溢出）和时延增加（数据在路由器缓存排队）。

拥塞控制：发端维护流量控制窗口rwnd和拥塞窗口cwnd，真正的发送窗口为$swnd = \min\{rwnd, cwnd\}$，需要机制设置cwnd避免网络拥塞。

TCP Tahoe：慢启动（指数增加），拥塞避免（线性增加），发生拥塞时拥塞门限减半（乘性减少）。

```
慢启动阶段，指数增加
    若cwnd<=ssthreash（慢启动门限）则为慢启动阶段
    每次收到ACK，cwnd增加一个报文段长度，窗口指数增长
拥塞避免，线性增加
    每收到确认，cwnd=cwnd+(MSS*MSS/cwnd)，窗口加性增加
以丢失报文段来指示网络拥塞，定时器超时，门限减半
    ssthresh = cwnd / 2，cwnd = 1，进入慢启动过程
```

TCP Reno：增加快重传（用3次重复的ACK指示报文丢失）和快恢复（快重传之后的临时模式）。

```
启动拥塞的条件，分组丢失
    定时器超时 OR 收到重复的ACK（3次）
发生分组丢失，减少cwnd
    IF 定时器超时 THEN
        ssthresh = cwnd/2
        cwnd = 1
        进入慢启动过程
    ELSE 重复ACK  // 快重传
        ssthresh = cwnd/2   // 快恢复
        cwnd = cwnd/2 + 3*MSS
        n = 收到的重复的ACK数目
        IF n > 3 THEN
            cwnd = ssthresh + n*MSS
        IF 发送窗口还允许发送报文段 OR 收到新的ACK
            cwnd = ssthreash
            进入拥塞避免
数据发送成功，增加cwnd
    慢启动阶段，指数增加；拥塞避免，线性增加
```

TCP NewReno：Reno超时会出现在ACK丢失，或首次报文段丢失之后的重复ACK少于3。快重传之后收到新的ACK应该视为再次发生了分组丢失，不需要等待到超时，立刻发送即可（NewReno快恢复）。

TCP Vegas：cwnd为期望速率与实际数据速率之差，目的是控制网络缓存中的数据量。TCP Vegas根据延迟变化主动响应拥塞。

$$
\begin{aligned}
cwnd&：拥塞窗口\\
rtt^*&：无拥塞时最小RTT\\
rtt&：有拥塞时实际RTT\\
diff&：队列中缓存数据的估计\\
\alpha&：diff的低门限，diff>\alpha\\
\beta&：diff的高门限，diff>\beta\\
diff&=(\frac{cwnd}{rtt^*}-\frac{cwnd}{rtt})\times rtt^*\\
&IF\ diff<\alpha\ THEN\ cwnd+=1\\
&ELIF\ diff>\beta\ THEN\ cwnd-=1\\
&ELSE\ (\alpha<=diff<=\beta)\ cwnd=cwnd
\end{aligned}
$$

TCP定时器：每发送一个报文段，就对其设置一个重传定时器；若超时前收到ACK，则取消定时；否则超时重传。超时会指示网络拥塞。

网络往返时延：链路上的延迟均值是可以估计的，因为链路层上链路的往返延迟时延方差很小；网络层上IP分组由多条链路传送，往返时延变化会很大。

RTT估计：使用迭代的方式进行估计。RTT为估计值，rtt为本次发送的测试值。α越接近1，则新的rtt对RTT影响越弱；典型的$\alpha = \frac{7}{8}$

$$RTT \leftarrow RTT \times \alpha + rtt \times (1 - \alpha)$$

超时重传时间RTO：β越接近1，对超时和拥塞的测试条件越严苛，越容易出现超时误判。TCP标准推荐$\beta = 2$。

$$RTO = \beta \times RTT,\ \beta > 1$$

由于往返时延震荡很强，因此仅仅使用RTT作为期望估计是不足的，也需要对方差进行估计，并对RTO进行修正。推荐$\alpha = \frac{3}{4}$

$$D \leftarrow \alpha \times D + (1 - \alpha) \times |RTT - M|$$
$$RTO = RTT + 4 \times D$$

Karn算法：在时刻1发送报文段，没有收到ACK，在时刻2重传，收到ACK，如何确认该RTT式哪一个？以及rtt如何计算？Karn算法表示计算RTT时，若报文段重传，则不采用新的时延值。

修正的Karn算法：报文段每重传一次，就增大RTO，直到报文段不再重传，才根据rtt更新RTT和RTO。推荐$\gamma = 2$。

$$RTO = \gamma \times RTO$$

无线TCP拥塞控制

无线链路特点：RTT更大（慢启动），误码率更高（容易超时），移动性和切换网络（存在潜在的丢帧问题）。

拥塞控制算法：Reno算法，Hybla算法（引入相对RTT参数），Bic/Cubic算法，Westwood+算法等。

卫星链路上，使用Westwood+和Hybla改进TCP性能更好。

7. 应用层

流媒体应用和协议

流媒体：音频/视频数据，边下载边播放，下载结束后用户磁盘上不保留播放内容。

特点：数据量大，产生的数据与时间有关，需要有效利用数据相关性和人的感觉的不灵敏性对数据进行压缩。

数字音频：语音信号的数字表示。

采样：对语音信号在时间轴上分段，每秒采样次数定义为采样率。

量化：用一定的比特数表示样本幅度，将表示每个样值的比特数定义为量化位数。

奈奎斯特定理：采样率为信号最高频率的两倍时，采样后的数字信号完整地保留了原始信号中的信息。

静默期：没有声音，无信息，可以不进行数据传输。

激活期：有声音，传输间隔固定。

数字图像：将图像分割成像素（点阵），对像素的颜色/亮度进行编码，保留每个像素值。

数字视频：在时间轴上连续的数字图像（帧）。

JPEG有损编码：用于24位RGB数字图像压缩，利用了人眼对亮度信号敏感度高于颜色信号的特性。首先需要进行信号变换$(R,G,B)\rightarrow(Y,Cb,Cr)$，其中Y为亮度分量，Cb和Cr为色度分量，之后对在Cb和Cr矩阵上进行2x2均值pooling，将数据量降为1/4；在Y，Cb和Cr矩阵上对每个8x8像素构成的块，进行离散余弦变换（DCT），对DCT系数进行量化（除以量化表对应的权值），缩小高频分量值；最后按Z形排列64个元素值，使得更多的0连续排列，对排列的数值采用Huffman码游程编码。

DCT：变换后(0,0)为该块均值，其他为空间频率的功率值，能量会集中于低频部分。

MPEG压缩：同时压缩音频和视频，利用图像的空间和时间相关性进行压缩。空间相关性同JPEG，时间相关性为对相邻帧求差值后在采用JPEG。

I帧：帧内编码。

P帧：预测帧与前一帧的逐块差值。

B帧：双向帧与前一帧和后一帧的逐块差值。

存储的媒体：媒体存储在信源端，需要传递给客户。

流：客户接收到数据后，按照原始记录的时序连续播放。

流媒体：客户端在播放此前储存的报文信息的同时，还要及时接收来自服务器的报文。

实况流媒体：弱交互，可以暂停播放，回放，快进等，要求必须根据原始记录时序实时播放。

实时流媒体：交互式，例如IP电话和视频协议，对延迟的要求很高。

从WWW服务器下载文件的方法：客户机通过浏览器GET视频/音频文件，WWW服务器进行响应，将文件下载至客户机本地，最后客户机使用媒体播放器将视频/音频文件进行播放。

带有元文件的WWW服务器：客户机通过浏览器GET元文件，WWW服务器响应并将元文件下载到客户机本地，客户机再通过媒体播放器打开元文件，利用元文件信息从WWW服务器GET视频/音频文件，得到服务器响应，直接在媒体播放器上播放。

元文件：一种非常小的文件，用来描述或指明其他文件的一些重要信息。

媒体服务器：从WWW服务器获取元文件，之后媒体播放器解析元文件，从媒体服务器获取视频/音频文件。媒体服务器与WWW服务器分离。

实时传输协议RTP：为实时应用提供端到端传输服务，多媒体数据经过RTP封装后，交给**UDP**的Socket接口发送。

RTP首部：有效载荷类型，指明信源编码类型（MPEG，JPEG等）；序号，接收端用序号检测丢包并回复RTP报文时序，每发送一个则序号加1；时间戳，RTP数据中第一个字节的采样时间戳；同步源标识SSRC，标识RTP信源，在TRP会话中每个流有唯一的SSRC。

RTCP：RTP控制协议。负责服务质量检测和反馈，流媒体间同步，多播组成员识别。RTP会话的参与者会定期发送RTCP。**UDP传输**。

RTP会话：一个多播组，所有属于该会话的RTP/RTCP都使用组播地址，不同会话的RTP和RTCP通过端口号进行区分，会话参加者根据参加者数量增减RTCP数据量。

结束报文BYE：关闭一个数据流。

特定应用报文APP：定义新的分组类型。

接收端报告RR：接收端以多播方式周期地发送。包括分组丢失比例，最新序号，平均延迟变化等。

发送端报告SR：发端使用多播方式周期地发送。包括RTP流的SSRC，当前时间，发送分组数，发送字节数等。提供RTP流同步的时间戳信息。

信源描述SDES：描述会话参加者。包括与RTP流有关的SSRC，建立SSRC到用户/主机名的映射。

实时流媒体协议RTSP：媒体播放器与媒体服务器之间的控制协议，使用554端口，与媒体流传输端口不同。控制消息采用**UDP或TCP**传输。客户端有如下命令，每个命令都会得到服务器端的RESPONSE。

DESCRIBE：获取多媒体参数。

SETUP：建立播放器和服务器之间的逻辑信道。

PLAY：服务器向客户端发送多媒体流。RECORD：服务器接收客户端的多媒体流。

PAUSE：暂停发送。

TEARDOWN：拆除逻辑信道。

SIP协议：用于IP电话的信令和服务质量，同RTP和RTCP配合。其具有两个构件——用户代理和网络服务器。SIP消息可以采用**TCP或UDP**发送。SIP使用名字或E-mail地址唯一标识用户。

用户代理：发起呼叫或接受呼叫。

网络代理：重定向服务器转发用户呼叫请求，最终到达被叫用户；重定向服务器通知下一跳代理服务器的地址。

建立呼叫：被叫知道谁在呼叫，主被叫协商媒体类型和编码方式。

```
1. Alice发出SIP邀请，指示端口号，IP地址，接收编码类型
2. Bob回复200 OK，指示端口号，IP地址，期望的编码类型
3. 编码协商：Bob不支持Alice的编码，应答606标识不接受，之后Alice可以发送新的邀请，并更换其他编码
4. 拒绝呼叫：Bob可以使用各种原因主动拒绝呼叫，也可能因为媒体采用RTP等协议而拒绝呼叫。
```

名字翻译和用户定位：主叫呼叫被叫，但只有被叫的名字或E-mail时，需要得到被叫当前主机的IP地址。

SIP注册：启动SIP客户端时，客户端发送SIP REGISTER给该用户注册的代理服务器。

SIP代理：用户向其代理服务器发送邀请消息，代理服务器将SIP路由到被叫，被叫响应进行同样方式的路由发回主叫。代理服务器类似本地DNS。

媒体播放器：通过支持RTSP提供交互功能；提供用户图形界面；可以解压缩；可以消除错误；缓存多媒体数据以消除抖动。

消除错误：利用FEC进行简单纠错。每发送4个报文，就构造一个按位异或的检验报文，与原报文一起发送，可以进行简单纠错检错。

消除抖动：播放器通过缓存来自服务器的输入，增加时延，但消除抖动。

QoS

QoS：服务质量，满足业务需求的必需的技术指标，包括带宽，延迟，抖动，丢帧率等。

RTP发送时是等间隔的，经过互联网转发后变为不等间隔的。

时延抖动：由于网络延时的变化导致分组到达速率的变化，严重时会导致视频/音频时快时慢，用户无法忍受。

IntServ：为每个应用会话提供服务质量保证，提供资源预留和呼叫建立。提供比尽力而为更好的服务，适用于所有业务模型。

资源预留：路由器需要知道为某一个会话预留的多少资源。

呼叫建立：需要服务质量保证的会话必须首先在源到目的的路径上的每个路由器都预留足够的资源。

R-Spec：定义需要的QoS，声明其QoS需求。

T-Spec：定义网络传输的业务流特征。

信令协议RSVP：携带R-Spec和T-Spec给路由器，以预约请求。**UDP**。

发端和收端在RSVP之外完成，加入一个多播组（发端不必需）。

发端到网络的信令：路径信息，保证发端知道所有的路由器；路由拆除，从路由器上删除发端路径状态信息。

收端到网络的信令：预留信息，预约由发端到收端的资源；预约拆除，删除接收端预约。

路径消息Path：沿数据路径从发端主机出发，并记录路径上每个节点的的路径状态 。

预留消息Resv：由接收方沿着反向路径发送到发送方；每个节点上，预留消息的IP目的地址将会改成反向路径上下一节点的地址，同时IP源地址将会改成反向路径上前一节点的地址；路径上各个路由器按照Resv中资源预留的设置进行资源预留，直到达到发端。

RSVP作用于路由器的接纳控制和通信量控制器上，其最终作用于分组调度器，用于调度分组转发。

多服务等级：提供服务类型等级的划分，对不同等级采取不同的网络处理方式。IP分组头中的ToS（服务类型）字段可以标识业务类型。

区分服务DiffServ：在路由器中增加区分服务的功能；将网络划分为许多DS域；边界路由器；根据流的DS值将若干流聚合成更少的流。

区分服务码点DSCP：IP头部的服务类型中子字段，定义服务类型。DSCP表示了四级业务和三种丢弃优先级。

服务等级SLA：ISP与用户协商服务等级，约定服务类别及业务量。

DS域：网络被划分为很多DS域，每个DS域的内部服务器尽可能简单，有利于快速转发，DS域的边界服务器实现复杂功能。

边界路由器：边界路由器实现分类，标记，整型，测量等功能。

聚合：根据流的DS值将若干流聚合成少量流，路由器对相同DS值的流按相同的优先级转发，简化内部路由器转发即止。

迅速转发：路由器提供大于某阈值的发送速率，提供保证带宽的端到端服务。

确保转发：DSCP表示的四级业务，每一级提供最低带宽和缓存空间；对三种丢弃优先级，发生拥塞时有限丢弃较高丢弃优先级的分组。

调度：选择在链路上要发送的下一个分组。

FIFO调度：按到达队列顺序发送。丢弃策略有丢尾（丢弃新到分组），优先级（丢弃低优先级分组），随机（随机丢弃）。

优先级调度：优先发送优先级高的分组。高优先级分组有可能抢占低优先级分组的资源。

循环调度RR：最具公平性的调度，按类别排队，循环扫描各类别队列，每个类别服务一次。

加权公平调度：在RR基础上，为每个类别的队列分配一个服务权重，按权重分配服务时间。

流量整形：将输入流的速率控制在有效带宽之内，避免拥塞和分组延迟，需要控制平均速率，限制峰值速率，控制突发长度。

令牌桶：漏桶每秒注入r个令牌，最多盛放b个令牌；分组到达后在队列中等待令牌，只有漏桶中有令牌时可以取走令牌并进入网络。流的长期速率由r限制，短期突发长度由b限制。

漏桶：B=0，强行限制数据速率，固定为r，不允许任何突发。

令牌桶和漏桶的结合：第一级使用令牌桶，r和b都较大，用于平滑流量；第二级使用漏桶，禁止突发，调节流的速率。

其他网络技术

Web页面：HTML，超文本，超链接。

URL：统一资源定位符。通过URL浏览器可以定位和获取网络上的内容资源。

HTTP：HTTP1.0具有多个连接和系列请求；HTTP1.1允许连接重用，支持持续连接和流水线式请求。

HTTP请求：GET获取网页，HEAD获取网页头部，POST返回表单等。

HTTP状态码：1xx表示消息，2xx表示成功，3xx表示重定向，4xx表示客户端错误，5xx表示服务器端错误。

内容分发CDN：利用最靠近每位用户的服务器，更快、更可靠地将网络资源发送给用户。

选择最好的CDN服务器：CDN建立映射表，指示了ISP到CDN节点的距离；请求到达了DNS服务器时，服务器查询用户请求来自哪个ISP，利用映射表决定选择哪个CDN。

多协议标记交换MPLS：用面向连接的方式代替IP的无连接分组交换，利用更快捷的查找方法，而非最长前缀匹配的方法来查找路由表。

MPLS标记：分组打上固定长度的标记，之后标记交换路由器（**LSR**）之间利用标记分配协议**LDP**交换报文，找到标记交换路径**LSP**，各个LSR根据LSP构造分组转发表。

MPLS域：标记生效的区域为MPLS域。分组进入MPLS域时，入口节点打上标记；离开时，出口节点去除标记。

FEC转发等价类：要求路由器按同样的方式对待的分组的集合。传统路由协议可能会导致某条路经过载，通过MPLS设置FEC标记，可以分散和平衡负载。

信息传输安全

信息安全：保证网络传输信息的机密性，认证性，完整性和不可否认性。

机密性：保证信息为授权者使用，不会泄露给没有授权者。

认证性：接收到敏感信息时，确认通信对方是谁，并保证信息从真实的发送者传送到真实的接收者。

完整性：信息传递过程中没有被他人修改。

不可否认性：发送者不能否认对信息进行的任何行为。

安全问题：窃取，中断，篡改，伪造。

数据加密模型：$D_K(Y = E_K(X)) = X$，所有算法同开，只有秘钥保密。

置换密码：加密秘钥和解密秘钥相同，例如凯撒密码。可以通过密码的频率统计进行部分破解，之后构造试探性密文得到其余密码。

替换密码：选择不重复的单词或短语或简单的数字串作为密钥，明文与密码对齐按行写，之后按密钥的排列顺序有小到大按列读出。

DES对称加密：64位秘钥，其中包括8位奇偶校验。对明文按64位分组，各组迭代加密后得到64位密文组，拼接后得到密文。

公钥密码体系：加密E解密D算法公开，具有公钥PK和私钥SK，SK由PK决定，但根据PK不能计算SK，公钥用于加密，私钥用于解密。

RSA算法：选择两个大素数p和q，计算$n=p\times q$和$z=(p-1)\times(q-1)$，选择与z互素的d，找到e使其满足$e\times d=1\ mod(z)$

$$加密：C=p^e\ mod(n)$$
$$解密：P=c^d\ mod(n)$$

数字签名：要求接受者能够核实发送者身份，接受者不能伪造报文，发送者事后不能诋毁报文内容。

数字签名实现：A进行签名，A使用私钥SK进行加密签名，B使用公钥PK进行解密核实签名。若A抵赖，则B出具密文和明文有第三方验证；若B伪造，则其不能出具密文。

保密的数字签名：在签名和核实之间，A利用B的公钥对签名完成的密文再加密，B利用其私钥对密文解密，之后再进行签名核实。

报文完整性：利用数字完整性认证报文。可以保证不可伪造和不可否认。

报文摘要MD：A利用散列函数，得到报文摘要，之后利用私钥签名，将签名的报文摘要追加在报文之后；B收到报文，分离出签名的报文摘要，利用公钥核实签名，若报文摘要与利用相同的散列函数得到的相同，则断定报文由A产生。

访问控制认证：验证通信对端不是假冒者。

重放攻击：假冒者记录真实的通信对端的分组，之后发送该分组用于认证身份。明文密码，对称加密的密码均不能防止重放攻击。

不重数：避免重放攻击，使用通信双方的共享秘钥加密不重数用于认证，并且之后很长时间内不再使用该不重数。

公钥加密的不重数：利用不重数认证，并且采用公钥加密。

中间人攻击：中间人拦截通信两端的所有分组，中间人替换双方私钥并索要双方公钥，完成认证。

认证中心CA：CA将某个实体的公钥与其绑定，每个实体具有CA发放的证书，内含公钥及其拥有者的标识，此证书经过CA数字签名。用户从可信的第三方获取CA公钥，用来验证某个公钥是否被某实体所有，中间人无法伪造证书。