今日干饭背诵（1.5）

【背诵】一些历史知识之类的

【概念】重点的概念，可能会考选填，要记清楚关键词

【计算】要掌握，可能会出小题

【大题】一些算法

【其他】不在考纲里，但感觉可能也会涉及（比如干扰项），放在最后面大家有空看看就行

Chap7 路由选择和网络层

（注：P129-176讲路由器体系结构和关键技术的部分属于扩展内容，考试不考）

网络层概述

【背诵】

网络层定义：网络层为**一个网络连接的两个传送实体**间交换**网络服务数据单元**（Network SDU）提供功能和规程的方法，它使传送实体独立于**路由选择和交换**的方式。

> 回忆：chap2
>
> - 点到点通道（交换式通信）的关键技术：路由选择
> - 协议数据单元PDU=SDU+PCI（对等实体之间）

网络层地位：**通信子网的最高层**；位于数据链路层和传输层之间，使用数据链路层（下层）提供的服务，为传输层（上层）提供服务。

提供的服务（为传输层）：

- 面向连接的服务（传统电信的观点，在网络层保证可靠性）
- 无连接服务（互联网观点，在传输层完成复杂功能，而非通信子网）

> 回忆：chap5 数据链路层为网络层提供的服务有 无确认无连接 / 有确认无连接 / 有确认有连接 三类

功能：实现**不同类型网络的互连**（屏蔽了差异，例如可以在ATM子网上运行TCP/IP，这样ATM网络能和异质网络互连）；了解通信子网的拓扑结构，选择路由，实现报文的网络传输。

网络层的内部结构（数据报子网、虚电路子网）

【概念】

内部结构：（具体带宽、状态、服务质量、健壮性、可扩展性等方面的比较如图）

- **数据报子网**：通信子网采用数据报分组交换方式，分组带地址被独立转发，每个分组都要做路由选择
- **虚电路子网**：通信子网采用虚电路分组交换方式，分组沿虚电路顺序转发，只需要在建立连接时做一次路由选择（之后都沿着建立好的虚电路走）

- 虚电路子网与数据报子网的比较
  - **带宽与状态的权衡**
    - **数据报子网中，每个数据报都携带完整的目的/源地址，开销大，浪费带宽** IPv4是32位，IPv6是128位
    - **虚电路子网中，路由器需要维护虚电路的状态信息；** 开销小，但需要维护端到端的连接状态（N个节点→N^2，扩展性存在问题）
  - **地址查找时间与连接建立时间的权衡**
    - **数据报子网对每个分组的路由查找过程复杂** 最长前缀匹配
    - **虚电路子网需要在建立连接时花费较长的路由查找时间**
  - **可靠性与服务质量的权衡**
    - **数据报不太容易保证服务质量QoS(Quality of Service)，但是对于通信线路的故障，适应性很强** 有竞争，不容易做资源的预留；但健壮性强
    - **虚电路方式很容易保证服务质量，适用于实时操作，但比较脆弱**
  - **可扩展性**
    - **数据报子网具有更好的可扩展性**

路由算法（网络层协议的一部分）

基础：静态 / 动态算法

【背诵】

路由算法：分为静态（非自适应）、动态两类，根据最优化原则找出并使用汇集树（从所有源结点到一个目的结点的最优路由的集合）

> 最优化原则：如果路由器 J 在 I 到 K 的最优路由上，那么从 J 到 K 的最优路由会落在同一路由上。

最短路径路由算法（Shortest Path Routing）：Dijkstra算法（P16），给拓扑求最短路径。

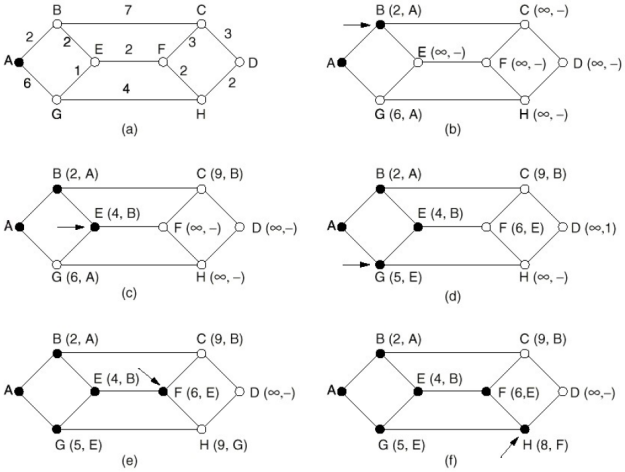


Fig. 5-6. The first five steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

【计算】

- 静态：
  - 洪泛算法（避免产生**loop**，有带计数器的宽搜 / 记录路径等措施）
  - 选择性洪泛算法（改进版本，将分组仅发送到**与正确方向接近**的线路上）
    - 特点：资源会浪费，但有极好的健壮性，可以作为**衡量标准**评价其他路由算法
  - 基于流量的路由算法（既考虑拓扑，又兼顾网络负载）
    - 前提：每对结点间平均数据流相对稳定可预测，因此网络延迟可以**提前离线计算**

        - 计算需要的信息：网络拓扑结构、通信量矩阵、线路带宽矩阵
- 动态：【重要】
    - 距离向量路由算法（Distance Vector Routing，以下简称DV，例题见P24）
        - 最初用于**ARPANET**，被**RIP**协议采用
        - 【步骤】每个路由器向邻居发送自己到所有路由器的距离表，本路由器到 i 的距离利用邻居路由器 X 、自己到X的距离m更新，为 $D_i = \min_{X \in nbr}(X_i + m_X)$
        - 注意：本路由器中的老路由表在计算中不被使用，更新是取所有邻居的最小值
        - 【缺点】**无穷计算**问题，选择路由时没有考虑**链路带宽**，路由**收敛速度慢**，路由报文**开销大**（不是增量更新，每次用全部的重算一遍），**不适合大规模网络**（例如采用它的RIP协议最大支持15跳）
        - 改进 - **水平分裂算法**（A告诉我的消息我不告诉A）以避免无穷计算，虽然广泛使用，但有时候会失败
    - 链路状态路由算法（Link State Routing，以下简称LS，例题见P32）
        - 使用的协议：**OSPF**，**IS-IS**；步骤如下：
        - 启动时，通过HELLO两次握手发现邻居节点；**简化拓扑**， n 个路由器连在一个LAN时，引入**人工节点DR**（代表路由器），原本 C_n^2 条路由之间的连线被简化成了 DR与路由器们的 n 条
        - 测量每个邻居节点的延迟或开销（用ECHO分组的往返时间/2，或者根据带宽）
        - 将学习到的**邻居信息封装成分组**（链路状态声明LSA），内容是sender ID、seq、age、邻居结点list（邻居-延迟）——【注意】分组定期创建，或发生重大事件创建
        - 把分组发送到**全网路由器**，采用洪泛方式
            - 控制洪泛中产生的重复包，分组含序号seq，同一个路由器尽量使自己发出的不同分组序号不同；其他路由接收到时可以判断是新分组 / 重复分组 / 过时分组，只处理新分组
问题	序号循环使用	路由器重启后序号重置 / 出错
改进	用32位序号	增加age域，计数超过Max丢弃
            - 其他改进
                1. 分组到达后延迟一段时间，多接收一些分组，把来源相同的进行比较，丢弃重复的，保留新的（抑制抖动）；
                2. 对分组进行应答，发送ack
        - 根据Dijkstra算法计算最短路径，把邻居的信息汇总成图

【概念】

路由算法中DV和LS的比较：

-

- 路由信息的复杂性
  - LS
    - 路由信息向全网发送　把自己对邻居的认识（准确）发送给全网
    - N个节点，E个链路的情况下，发送O(NE)个报文
  - DV
    - 仅在邻居节点之间交换　把自己对全网的认识（不一定准确）发送给邻居
  - 注意
    - LS发送的是链路信息，DV发送的是到所有结点的向量信息
    - LS信息定期创建（30分钟）或发生重大事件时创建，DV定期创建（30秒钟）
    - LS发布增量信息，DV发布全部信息
- 收敛（**Convergence**）速度
  - LS　收敛比较快
    - 使用最短路径优先算法，算法复杂度为O(nlogn)
      - n个结点（不分组括源结点），需要n*(n+1)/2 次比较
      - 使用更有效的实现方法，算法复杂度可以达到O(nlogn)
    - 可能存在路由振荡（oscillations）
  - DV　Bellman-Ford算法
    - 收敛时间不确定
      - 可能会出现路由循环
      - count-to-infinity问题
- 健壮性：如果路由器不能正常工作会发生什么？
  - LS
    - 结点会广播错误的链路开销　（到邻居的开销）
    - 每个结点只计算自己的路由表
  - DV
    - 结点会广播错误的路径开销　（到全网的开销）
    - 每个结点的路由表被别的结点使用，错误会传播到全网
      例如，一个人把自己的路由器接入校园网并宣称自己到其他路由距离最短，于是所有路由器都转发给他了，但他的路由不转发——变成一个路由黑洞

- 【DV有但LS没有的优点】（因此DV现在还在使用）
  DV的开销计算尺度是一致的（可以把带宽、延迟等因素加权计算出统一的），但LS可能各个路由算各自的，互连时难以度量。因此在不同国家/运营商之间（往往不便透露路由策略）网络互连使用DV（距离向量）算法。
- 相应使用的协议：
  - DV：RIP

    > 相似 - 路径向量算法PV被BGP采用（域外路由协议EGP）

  - LS：OSPF、IS-IS（都是域内路由协议IGP）

【其他】

由于网络规模、应用场景的变化，许多问题需要新的解决方案，例如：

- 分层路由（解决路由表规模过大的问题）
- 移动主机的路由（解决移动主机如何定位的问题）

分层路由

【概念】

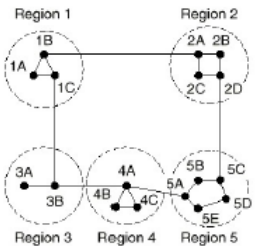
分层路由：网络规模增长导致路由表规模、路由选择时长和收敛速度等问题的一种解决方案。采用分而治之的思想，将网络分成若干域，使路由表规模大幅减少。

- 自治系统AS，AS之间互相记录
- AS内划分为区域，区域之间互相记录
- 区域内有各个路由器，路由器之间互相记录

【问题】分层后计算得到的路由不一定是最优路由。

【计算】

x 个AS（自治系统 / 簇），每个AS内 y 个区域，每个区域中 z 个路由器，则路由表项有 $x+y+z-2$ 个。



(a)

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

域内

域间

(c)

移动主机的路由

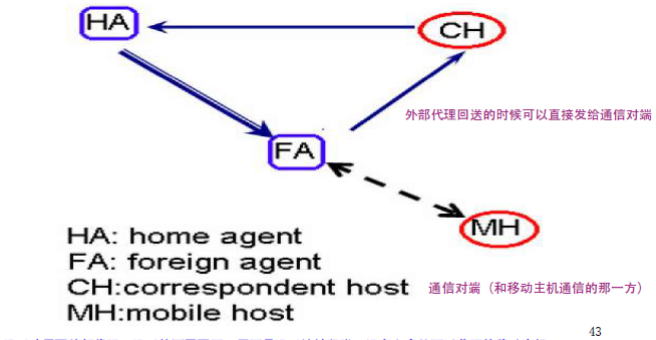
【概念】

- 移动用户
- 家乡位置（每个移动用户都有一个永久的家乡位置，用地址来标识）
- 家乡代理（记录不在家的移动用户）
- 外部代理（记录正在访问该区域的移动用户）

过程：

- 移动用户进入新区域后向**外部代理**注册
- 外部代理联系家乡代理完成确认后注册成功（具体过程见P41）
- 当一个分组发给移动用户时，首先被转发到用户的**家乡局域网**
- 包到达家乡局域网被**家乡代理**接收，家乡代理查询移动用户的新位置和与其对应的外部代理的地址，采用**隧道技术**，将收到的分组作为净负荷封装到一个新分组中，发给外部代理
    - Optional：家乡代理告诉发送方，后续分组作为净负荷直接发给外部代理（发送方要修改协议栈）
- 外部代理收到分组后，将**净负荷**封装成数据链路帧（同一LAN内才能使用MAC地址）发给移动用户

【注意】IPv4才需要外部代理，IPv6不需要。



IPv4才需要外部代理，IPv6就不需要了，原因是IPv4地址紧张，没有多余的可以分配给移动主机；
但IPv6地址很多，可以对MH自身的信息（家乡等）分配全球唯一的地址，CH发给HA后，HA可以直接发给MH

拥塞控制算法（主要在网络层和传输层TCP进行）

【背诵】

拥塞：网络上分组过多，性能下降

拥塞产生的原因：

- 设备问题 —— 网络设备处理器性能低
- I/O问题 —— 高速端口输入，低速端口输出；多个输入对应一个输出

拥塞控制 vs. 流控制的差别：

- 拥塞控制是**全局**问题，解决通信子网和数据流问题（涉及主机、路由器等）
- 流控制是**局部**的点到点传输问题，一般基于反馈控制，解决快发慢收问题（例如滑动窗口）

【概念】

1、拥塞控制基本方法：

- 开环控制：提前设计，不考虑网络某次运行时的当前状态
- 闭环控制：基于反馈，监控拥塞-传输消息-调整解决

2、影响拥塞的网络设计策略（Policies that affect congestion，P50）：

- 传输层：重传、缓存、确认（ack）、流控、超时等机制
- 网络层：虚电路 / 数据报子网的设计等
- 数据链路层：重传、缓存、确认（ack）、流控等机制

3、拥塞控制算法：（框架性的理解）（部分内容来自《计算机网络》第5版 5.3-5.4）

- 网络供给（provision）：增加设备资源
- 流量感知的路由（traffic-aware routing）：提高利用率
- 准入控制：降低负载，拒绝新连接的建立（广泛用于虚电路子网中的拥塞控制）
    - 要想确定是否接受新连接的建立，需要**流说明**（速度和形状等描述）
    - 流说明通常采用的描述符是**漏桶** / **令牌桶**（它们用于流量整形），这样就包含了数据流平均速率和瞬时突发流量大小这两个参数
- 流量调节：参与源端的反馈循环，通知src减缓速度（闭环控制）
    - **抑制分组** / **逐跳抑制分组**是其中的两种方法，可用于数据报 & 虚电路子网
- 负载丢弃：努力失败，只能丢弃（闭环控制）
    - **随机早期检测**（RED，Random Early Detection）是其中一种方法，为了确定何时开始丢弃数据包，路由器要维护一个运行队列长度的平均值。当某条链路上的平均队列长度超过某个阈值时，该链路就被认为即将拥塞，因此路由器随机丢弃一小部分数据包。随机丢弃的数据包使得快速sender发现丢包的可能性更大。（见《计算机网络》第5版，P325）

4、服务质量保障的方法：（框架性的理解）

- 【流量整形】属于开环控制，强迫分组以一种可预测的速率发送，避免突发流量高峰，典型算法有
    - **漏桶**算法：漏桶存放数据分组（满了就丢），转变成平滑的数据分组流

      【特点】不够灵活，不允许空闲主机积累发送权
    - **令牌桶**算法：漏桶存放令牌（固定间隔产生，满了就丢），分组传输之前必须获得一个令牌，传输完成后删除令牌

      【特点】允许空闲主机积累发送权（最大为桶的大小）
    - （以上两种都可用于①固定分组长的协议如ATM；②可变分组长的协议如IP）
- 【包调度】数据包调度算法，决定包何时得到路由器的服务
    - FIFO
    - **公平队列**
    - **加权公平队列**（WFQ）
- 【准入控制】

======框架结束，下面按照课件顺序列出了一些知识（漏桶和令牌桶在上面）======

虚电路子网中的拥塞控制：

1. 流说明：描述发送数据流的模式和希望得到的服务质量的数据结构
    - 子网和接收方可以做出三种回复：同意、拒绝、其它建议
2. 准入控制：根据流说明和网络资源分配情况，进行准入控制
    - 可以在解决拥塞前不允许建立新的虚电路
    - 也可以允许建立新虚电路，但要绕开拥塞地区
3. 资源预留：建立虚电路时主机与子网达成协议，子网根据协议在虚电路上为连接预留资源

抑制分组/逐跳抑制分组：（P61的示意图）

- 抑制分组：只对发送方起作用（向源主机发送抑制分组，使源主机减少发向特定的发生拥塞的目的地址的流量）
- 逐跳抑制分组：对它经过的每个路由器都起作用（高速、长距离网络中源主机响应慢，这个可以快速缓解拥塞，但是要求上游路由器有更多的缓冲区）。

公平队列算法（Fair Queueing）：路由器的每个输出线路有多个队列；路由器循环扫描各个队列，**发送队头的分组**；所有队列**优先级相同。**【改进】变长分组可以由逐分组轮询改成按字节轮询。

加权公平队列（Weighted FQ）：不同队列优先级不同，优先级高的队列在一个轮询周期内获得更多的时间片。

负载丢弃（Load Shedding）：上述算法不能消除拥塞时的路由器丢弃策略，针对不同服务可以不同——

- 文件传输：丢弃新的，wine（认为酒越老越香）；
- 多媒体服务：丢弃老的，milk（牛奶越新鲜越好）。

网络互联

【背诵】

互连网络（internet）：两个或多个网络构成互连网络（异构网络不能通过data link层互连，因此需要network层）

网络互连设备：

- 中继器：物理层，在**电缆段**之间**拷贝**比特，对弱信号进行放大/再生以延长传输距离

  > 集线器（hub）也工作在物理层。

- 网桥：数据链路层，在**局域网**之间**存储转发**帧，网桥**可以改变帧格式**
- 多协议路由器：网络层，在网络之间**存储转发**分组。必要时，做**网络层协议转换**
- 传输网关：传输层转发字节流
- 应用网关：应用层实现互连

无连接网络互连（Internetworking）：

- 路由工作过程：类似数据报子网，互连网络中每个分组单独路由
- 路由设备：**多协议路由器**为连接的不同子网进行协议转换（分组格式、地址等）
    - 一种技术：隧道技术，适用于一种常见的特殊情况（见【概念】）

互连网络路由（Internetwork Routing）：

- 路由工作过程：类似单独子网，只是更复杂
- 路由算法：两级
    - 内部网关协议（IGP, Interior Gateway Protocol），自治系统AS内部
        - RIP、OSPF、IS-IS
    - 外部网关协议（EGP, Exterior）
        - BGP

分片：解决网络互连时最大分组长度不同的措施（见【其他】）

防火墙：防止信息泄露或不好的信息渗透，在**网络边缘**设置防火墙。

- 早期配置是两个路由器中间夹着一个应用网关。

【概念】

隧道技术：**源和目的主机所在网络类型相同**，连接它们的是一个不同类型的网络，这种情况下可以采用隧道技术。

【注意】对于头尾的多协议路由器，经过后会加上header（没有头的话就只是翻译）

- 例如[IPv6 packet] -> [IPv4 [IPv6 packet]] -> IPv6 packet，详见P72

【其他】

分片：发生在不同网络的最大分组长度（最大传输单元MTU）不同的时候，大分组经过小分组网络时，网关要把大分组分成若干片段（fragment），每个片段作为独立的分组传输

分片重组策略：

- 对其他网络透明：使每个片段经过同一出口网关，在那里重组（receiver看不见分片过程，拿到的是完整的）

  【问题】

    - 出口网关要知道何时片段到齐；

    - 所有片段必须从同一出口网关离开；
    - 大分组经过一系列小分组网络时反复分片重组，开销大。
- 对其他网络不透明：中间网关不重组，由目的主机完成

    【问题】

    - 对主机要求高（得能够重组）
    - 每个片段都有一个分组头，网络开销增大

标记片段的方法：

- 树形标记法。例子：分组0分成三段，分别标记为0.0, 0.1, 0.2，片段0.0构成的分组被分成三片，分别标记为0.0.0, 0.0.1, 0.0.2。
    - 【问题】段标记域要足够长，分片长度前后要一致
- 偏移量法，基本片段长度，分组头包括原始分组seq，第一个基本片段的offset，最后片段指示位（例子见P79）

网络层协议

网络层中互联网可以看作自治系统（AS）的集合，是由网络组成的网络。

网络之间互连的纽带是**IP协议**。

IP协议（Internet Protocol）

IP头：20个字节的固定部分 + 0-40个字节的变长部分。Version，IHL（**注意单位是32-bit word，即4 bytes**），Type of Service，Total length，Identification，DF，MF（若分片，除最后一个片段外都要置More Fragments位），Fragment offset（除最后一个片段外的所有片段的长度必须是**8字节**的倍数），TTL（最大值为255），Protocol，Header Checksum，Source address，Destination address，Options

IP地址：网络号 + 主机号

- 有类地址（A类0，B类10，C类110，D类1110，E类11110）
- ABC的主机号都按字节对齐，D没有网络号 + 主机号，表示多播，E是保留
- 【注意】全0表示本网络 / 本主机，全1表示广播地址，因此进行IP地址分配时要保留这两个，不用于分配

子网：子网掩码高len位全1，与IP地址做AND得到网络地址。

ICMP协议（Internet Control Message Protocol）

主要用来报告错误和测试，ICMP报文封装在**IP**分组中

ARP协议（Address Resolution Protocol）

解决网络层地址（IP地址）与数据链路层地址（MAC地址）的映射问题

- 若目的主机在同一子网内，用目的IP地址在ARP表中查找，否则用缺省网关的IP地址在ARP表中查找
- 若未找到，则发送广播分组，目的主机收到后给出应答，ARP表增加一项
- （每个主机启动时会广播自己的IP-MAC地址，ARP表项中的动态ARP有生存期）

ARP攻击：攻击者发出伪造的ARP响应，更改目标主机ARP缓存中的IP-MAC表项，造成网络中断 / 中间人攻击；**存在于局域网**

RARP：见课件

RIP协议（Routing Information Protocol）

属于内部网关协议（IGP），封装在**UDP分组**中，采用**距离向量算法**（DV）

故障处理：

- 180s未收到邻居路由声明，则认为其失效，链路失效信息迅速传播到全网
- 使用**毒性反转**（从A学到的消息还告诉A，但是计算开销会发现是不可达的）

OSPF协议（Open Shortest Path First）

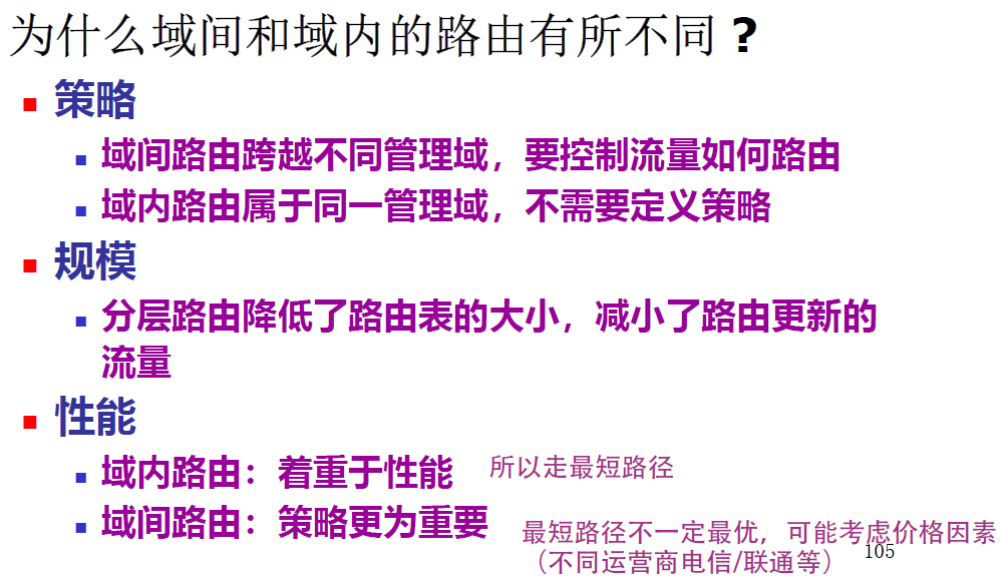
属于内部网关协议（IGP），采用**链路状态算法**（LS）因此支持多种距离衡量尺度，支持分层路由。

BGP协议（Border Gateway Protocol）

属于外部网关协议（EGP），封装在**TCP分组**中，采用**路径向量算法**（类似距离向量，每个BGP网关向邻居广播所有通往目的地的路径；网关W收到了邻居网关X的路径，
$Path(W,Z)=w,Path(X,Z))$

BGP消息：Open，Update，KeepAlive，Notification

【区别】域间路由 vs. 域内路由



无类域间路由CIDR（Classless InterDomain Routing）

提出背景：IPv4分配完毕，基于分类的组织浪费了大量地址（罪魁祸首是B类地址）

基本思想：将剩余的C类地址分成大小可变的地址空间

方法：路由表中增加掩码域mask，采用**最长前缀匹配原则**（需要遍历路由表，找到网络地址匹配且mask的len最大的），可以用于所有IP地址，地址格式为a.b.c.d/x，x为地址中网络号的位数（即mask中连续高位1的长度len）

【大题】CIDR地址分配（见2020网原课堂习题汇总，P116也有例题）

【注意】地址后分配的需要保证与前面分配的不交，如果后分配的地址掩码长度短于前面分配的（即子网地址个数更大），需要注意把前面的地址空间做padding，否则会出错。（如例题中Oxford从194.24.16.0开始，把上一个Cambridge的194.24.7.255填充到了194.24.15.255，原因是Oxford地址空间是Cambridge的2倍，掩码长度少1位）

IPv6

20世纪90年代由IETF讨论形成。

特点：IPv6与IPv4不兼容，但与其他Internet协议（TCP、UDP、OSPF、BGP、DNS）等兼容，然而实际上还是需要开发另外一套协议栈。

目标：减少路由表大小、提供安全性、支持组播（太多了，详见P119）

主要变化：

- 地址从32位变成128位（16字节）
- IP头由13个域减少为7个域

  > 例如分组头定长，就取消了IHL域，还有protocol, fragment相关, checksum域都无了

- 安全性提高
- 更好地支持选项功能

IPv6地址表示：16字节地址表示为用冒号 `:` 隔开的8组，每组4个16进制位（2 byte），例如 `8000:0000:0000:0000:0123:4567:89AB:CDEF`

- 优化表示：省略开头的0，0123写成123
- 多组16个0可以被一对冒号 `::` 代替（只能出现一次），例如上面的地址可以写成 `8000::123:4567:89AB:CDEF`
- IPv4地址可以表示成一对冒号 concat 用 `.` 分隔的十进制数，例如 `::192.31.20.46`

IPv4与IPv6过渡期的互连

解决办法：

1、双栈：实现IPv4/v6两套协议栈，主机根据DNS返回的结果或对方发来报文的版本号决定采用哪个协议，路由器根据收到IP分组的版本号决定采用哪个协议

2、翻译（多协议路由器）：有些路由器实现IPv4/v6两套协议栈，在两套之间进行**协议翻译和地址翻译**——已经被NAT-64等协议代替

3、隧道：IPv6的报文作为**IPv4报文的净负荷**在IPv4网络中传输

- 【注意】隧道模式只适用于两端网络一样的情况，如果一端是v4，一端是v6，就只能采用翻译