Internetworking

Internet Protocol

Network Layer

IP 位于网络层，网络层实体存在于每个路由器和端主机上。IP 提供无连接的服务。

网络层的主要协议有

- 路由协议：如 RIP，OSPF，BGP 等
- IGMP：用于支持组播
- IP：用于寻址以及规定数据报格式
- ICMP：用于报告错误以及传递信息

Internet Addressing

Addressing Level

物理网络的地址用于在简单的物理网络中对 PDU 进行路由

互联网的地址用于在互相连接的网络之间对 PDU 路由

应用的地址用于在端主机标识进程

Addressing Scope

global address 用于在全局无歧义地标识一个路由器或主机，且一台设备可以有多个 global address

network attachment address 是对连接到某种特定网络上的主机的唯一地址，如以太网上的 MAC 与 ATM 网络上的地址

port address 是在网络层之上并且在一个端设备内唯一的地址（全局不一定唯一），如传输层的端口号

Address Mode

unicast address 单播地址，是一个设备或端口的地址

broadcast address 广播地址，是一个 domain 内所有实体的地址

multicast address 组播地址，是网络某个子集中所有实体的地址

anycast address 任播地址，子集中任意一个实体的地址（区分组播，任播只需要至少一个实体收到即可）

IP operations

Routing

IP 有路由的功能，主机和路由器维护路由表，标识不同数据报下一跳应该转发到哪个路由器，可以是静态的也可以是动态的

路由有不同的策略，如距离向量（Distance Vector），连接状态（Link State），路径向量（Path Vector）等，对应着不同的具体路由算法

也有源路由的做法，由发送方标明路由路径

Datagram Lifetime

网络中的数据报可能会无限循环（路由器的网络信息可能已经过时），因此需要为数据报维护 lifetime，即 IP 中的 TTL (Time To Live)，当 lifetime 结束时则丢弃该包。TTL 一般每一跳减 1

Fragmentation and Re-assembly

有时候从上层传来的数据可能会超过下层网络的 MTU，此时需要对数据分片，使其不超过路径上的 MTU 最小值

分片工作在端主机和路由器均可完成，而重组一般只在端主机完成

IP 头部的信息用于分片后的重组，包括

- Data Unit Identifier：标识数据报的发送方和接收方以及上层的标识符
- Data Length：数据报的长度
- Offset：标识该分片在原始数据报中的位置，单位为 64 bit（8 octet）
- More Flag：用于标识是否是最后一个分片

在接收方进行重组时，必须预留好足够的缓冲区，当有着相同 ID 的报文到达时，将其插到正确的位置（根据 Offset 和 Length），直到整个数据报被重组完成

数据报的重组也有可能会失败，在开始重组时设置计时器，若超时，则丢弃不完整的数据

Error Control

IP 不保证交付，仅提供尽力而为的服务。如果路由器丢弃了某个包（校验和错误或是 TTL 超时），需要告知数据报的发送方，此时通过 ICMP 协议来传递错误信息，而发送方可能会通知更高层的应用或协议

Flow Control

流控制用于使路由器能够限制到达的包的速率，但是在无连接的网络中较难做到。

当路由器的缓冲满时，会丢弃新到来的包，此时可以通过 ICMP 告知发送方

IP header

IPv4 的报文头的典型长度为 20 字节，包括

- Version：4 bit，标记版本，对于 IPv4，其值为 4
- IHL：4 bit，Internet header length，单位是 32 bit，一般对于没有选项的 IPv4 其值为 5
- ToS： 8 bit，Type of Service，用于 QoS，包括
  - Precedence: 3 bits, 8 levels defined
  - Reliability: 1 bit, Normal or High
  - Delay: 1 bit, Normal or Low
  - Throughput: 1 bit, Normal or High
- total length： 2 字节，标识数据报的总长度，单位为 octet
- Identification：2 字节，为 IP 报文的序列号，和地址及上层协议一同唯一标记一个数据报
- Flags：3 bit，包括 More Flag（指示之后是否还有分片），Don't fragment（不能分片）
- Fragmentation offset：13 bit，用于分片
- Time to live： 8 bit，TTL
- Protocol：8 bit，用于指示上层协议
- Checksum：2 字节，整个头部的校验和，计算方法与 UDP 的校验计算方法相同
- Source address：4 字节，源地址
- Destination address：4 字节，目的地址
- 选项和 padding：可选项，padding 用于填充头部至 32 bit 的整数倍

IP 首部之后就是数据段，数据报最大长度（首部+数据）是 65535 octet

IP Addressing

IP address

IP 地址是每个网络接口（一个 router 可以有多个网络接口——可以有多个 IP 地址）的 32 bit 长的全局网络地址，可分为网络部分与主机部分。IP 地址主要用于互联网的寻址，物理网络内的寻址不需要 IP 介入

IP 地址可使用点分十进制记法，即将 32 bit 地址分为 8 bit * 4 部分，每个部分采用十进制表示，中间用点分隔

IP 地址中的网络部分唯一且由指定组织分配（ARIN, RIPE, APNIC），而主机部分则由网络地址的拥有者自行分配

IP address classes

Class A

最高位 0，之后是 7 bit 的网络地址和 24 bit 的主机地址

其中全 0 的网络地址和 01111111（127）的地址保留，127 开头的地址用于回环测试，故支持的网络地址共 126 个，已被全部分配

1.x.x.x - 126.x.x.x

A 类地址支持约 $2^{24} - 2$ 个主机地址，其中全 0 是网络号，全 1 用于广播

Class B

最高位 10，之后是 14 bit 的网络地址和 16 bit 的主机地址

支持 2^{14} 个网络地址，已全部被分配

128.0.x.x - 191.255.x.x

B 类地址支持约 $2^{16} - 2$ 个主机地址

Class C

最高位 110，之后是 21 bit 的网络地址和 8 bit 的主机地址

支持 2^{21} 个网络地址，已几乎被全部分配

192.0.0.x - 223.255.255.x

C 类地址支持 254 个主机地址

Subnet and Subnet Masks

子网是对 IP 网络的一种逻辑上的划分，属于同一子网的主机的 IP 地址的高位部分是相同的，这一部分称为子网部分，其余低位部分即主机部分，用于唯一标识子网内的主机。子网部分的长度即是子网掩码高位的 1 的个数。路由器可以在子网间提供路由。分类网络也可以看作是对整个互联网空间的一种子网划分（如 B 类地址可以看作子网掩码为 255.255.0.0）

子网可以简化路由，减少路由表中的表项。根据子网掩码可以确定一个 IP 地址属于哪个子网，然后向对应子网的边界路由转发即可

CIDR

CIDR 即 Classless Inter Domain Routing，无类别域间路由。在 CIDR 中一个 IP 地址表示为

$$A.B.C.D/n$$

其中地址的前 n 位称为 IP 网络的**前缀**，指示 IP 地址所属的网络，n 指示了子网掩码中高位有多少个 1，即子网部分的长度

CIDR 是为了解决分类路由中的不灵活（无类比路由相对于分类路由），一个 B 类地址对于一个组织来说太大了（有接近 65000 个主机地址），而一个 C 类地址过小了（只有 254 个主机地址），不同于分类网络中前缀长度固定，只能为 8，16，24，CIDR 提供了更灵活的网络划分，CIDR 基于可变长子网掩码（VLSM）进行任意长度的前缀分配。

CIDR 支持**路由聚合**，即使用单个前缀标识多个网络。考虑一个 ISP，其分配的前缀为 `200.23.16.0/20`，对于其负责管理的网络部分，可以再划分出 8 个前缀 `200.23.16.0/23` - `200.23.30.0/23`，供不同的组织使用，在连接到因特网时，可只提供一条路由信息，即匹配前缀 `200.23.16.0/20` 的包均转发至该 ISP 的边界 router，然后在网络内部再进行到 8 个不同子网的路由，即使用一个前缀标识了多个不同网络。

同理，16 个原来的 C 类地址（/24），可聚合为一个 /20 的网络地址，只要其网络部分的前 20 位相同。而 /20 的地址也可以进一步聚合。路由聚合减少了网络向外界通告的信息数量，有效地控制了路由表项的数量

NAT

即 Network Address Translation，NAT 使得内网和互联网可以使用不同的 IP 地址集，将私有的 IP 隐藏在互联网上合法的 IP 地址之后。NAT 也解决了 IP 地址不足的问题，私有网络可以不使用在互联网上合法的 IP 地址，而是自行分配。

只需在内网和互联网的边界路由上提供 NAT 服务即可

NAT 可以隐藏内网中设备的 IP 地址，防止来自外界的攻击，起到防火墙的作用，也可以使一个组织内网地址分配的选择空间更大。NAT 一般分为三种

- static NAT，即每个内网的 IP 地址对应一个公网的 IP，一般的应用场景是使得外界可访问内网中的服务器的同时又隐藏服务器的内部 IP 地址
- dynamic NAT，即 NAT 路由维护一个公网 IP 地址池，每当有内部设备要访问公网时，从中分配一个地址供其使用
- NAPT，即端口复用，将不同内网设备对外网的访问映射到 NAT 路由的不同端口，多台设备使用一个公网 IP 即可与外界通信，摆脱了 NAT 中公网 IP 数量的限制

NAT 的地址转换有时候也会带来问题，如 P2P 连接

IP protocol suits

ARP

Address Resolution Protocol，用于将 IP 地址转换为物理地址，仅仅工作在 LAN 内（不能跨 LAN），是一个处于 Layer 2 和 Layer 3 之间的协议

ARP 的过程可描述为

- 发送方
  - 先查询本地缓存有没有目的 IP 的 MAC 地址
  - 若没有则构造一个 ARP request，添加发送者的 IP, MAC 和目的 IP
  - 将 ARP request 在 LAN 内广播
  - 获取 MAC 后缓存在本地（带时间戳，随时间过期）
- 接收方
  - 收到广播的 ARP request，检查目的 IP 是不是本机
  - 如果是，构造 ARP reply，添加本机的 IP 和 MAC
  - 将 reply 发送给发送方（request 中有其 IP 和 MAC）
  - 将发送方的 IP，MAC 加入缓存（带时间戳）

一般本地 ARP 缓存在数分钟内过期，是为了应对网络的变化

ARP 不能跨子网工作，当一个包目的地址不在本地网络中时，ARP 的查询对象应当是网关路由器的 MAC，在发送给路由器后由其转发给下一跳网络

DHCP

Dynamic Host Configuration Protocol，用于给网络中的主机动态地分配 IP 地址，尤其是为拨号上网和 LAN 的用户，是 BOOTP 的扩展，基于 UDP 协议，用于传递配置 IP 地址的信息

DHCP 的过程可描述为

- client 在子网内广播 DHCP-DISCOVER
- 若有 DHCP server，回复 DHCP-OFFER
- client 选择其中一个 server，广播 DHCP-REQUEST，其中包含 server 的 IP，用于告知未被选择的 DHCP server 可收回其 offer
- 被选择的 server 回复 DHCP-ACK，其中包含了 IP 地址的配置信息，同时 server 将该主机和对应 IP 绑定
- 若 client 想要解除绑定，则发送 DHCP-RELEASE

DHCP 的绑定有时限，若 client 没有在超时前更新也会被解绑

ICMP

Internet Control Message Protocol，用于传递错误信息和控制信息，如使用 Echo request 和 Echo reply 确认网络可达性，或是传达 TTL 超时，网络不可达等错误信息

ICMP 头部包含 8 bit 的 Type 与 8 bit 的 Code，这两个字段共同决定 ICMP 携带何种信息，而之后有 16 bit 的校验和，以及用于信息传递的其余数据（如 Ping 包含标识符和序列号）

ICMP 可用于测试网络可达性，源主机发送 Echo request，若目的设备能接收到 request，会回复 Echo reply，除了测试网络可达性以外，ping 还可以用于测量 RTT，以及计算从源到目的地需要经过多少跳（使用 TTL）

ICMP 计算路径的方法是向目的地连续发送 Echo request，其 TTL 设置为 $1, 2, \ldots$，直到收到回复。每当一个包 TTL 超时，丢弃该包的路由器便会发送回一个 TTL expired 的错误信息，通过从 1 开始递增 TTL 发送包即可根据收到的错误信息确认从本机到目的地的路由路径，这就是 trace route 软件的工作原理

ICMP 也可以用于测试路径上最小 MTU，原理为当一个设置为 Don't Fragment 的包大小超过路由器下一跳的 MTU 时会发回 parameter unintelligible 的错误信息，故从最大的包开始，利用二分法即可找到路径上最大的 MTU

Mobile IP

Need for mobile IP

Mobile IP 的产生是随着越来越多的 PDA 和 laptop 的使用。其原因是数据包传递时根据目的 IP 地址寻址，而许多应用基于 TCP 连接，TCP 基于 IP 地址和端口号寻址。当一个可移动设备改变位置时，其 IP 地址也会动态改变，IP 的改变会导致 TCP 连接的重启。mobile IP 即是为了解决这一问题。mobile IP 不改变 TCP 协议的内容，即对于 TCP 连接，假装移动设备的 IP 没有发生变化

Mobile IP Entities

Mobile Node (MN)：一个可能会更换接入点的主机

Correspondent Node (CN)：向 mobile node 发送信息的主机

Home Agent (HA)：在 home network 维护一个 mobile node 列表的节点

Foreign Agent (FA)：在 foreign network 帮助一个 mobile node 传递数据的 router

Triangle Routing

1. CN 建立起与 MN 的连接，并向其 home address 发送信息
2. HA 截取数据并将其转发给 MN（代理功能）
3. MN 直接回复给 CN

上述路由过程在 CN，HA 和 MN 间形成一个三角，故称为 triangle routing

Protocol

mobile IP 的协议分为三部分，Discovery，Registration，Tunneling

Discovery 阶段，FA 使用 ICMP 周期性告知其存在，而 MN 选择一个 FA 向其发送 request

Registration 阶段，MN 从 FA 得到一个 **Care-of-Address**，然后要求其 HA 将发给 MN 的数据转发给 FA，即

- MN 将 registration request 发送给 FA （包含 COA，MN 地址，HA 地址）
- FA 将其转发给 HA （根据 HA 地址）
- HA 向 FA 发送 registration reply，并将 MN 与其 COA 绑定
- FA 将其转发给 MN

由此在 HA 将 MN 注册，之后发送给 MN 的包都会被转发到 COA

Tunneling，即在 HA 和 COA 间建立起了 tunnel，HA 发送 ARP request 将 MN 的 IP 与其 MAC 绑定，然后将发送给 MN 的包通过 tunnel 转发给 FA，而 CN 认为 MN 的地址没有改变（MN 直接回复 CN），维持了 TCP 连接

在 Tunneling 的过程中，HA 将 CN 发送的数据报整个封装在一个新的 IP 报文中发给 FA，而 FA 将其取出后再转发给 MN

IPv6

IPv6 产生的动机是由于 IPv4 的地址空间快要用尽，32 bit 的地址空间很快就被完全分配。除此之外的动机还有

- 新的首部格式有助于加速处理和转发
- 首部改变以更好服务 QoS
- 在路由器不再分片
- 新的地址模式

IPv6 header

IPv6 的首部包括

- Version：4 bit，在 IPv6 中该字段值为 6
- Traffic Class： 8 bit，用于 QoS
- Flow Label：20 bit，用于标识处于同一个 Flow 的数据报
- Payload Length：16 bit，指示数据报总长度
- Next Header：8 bit，标识下一个首部的类型，用于扩展首部或是上层的首部
- Source/Destination Address：128 bit

IPv6 的扩展首部会依序出现（如果有）

- Hop-by-Hop Options：要求在每个中继路由器上处理
- Routing：源路由
- Fragment：源主机进行的分片
- Authentication：认证
- Encapsulating security payload：加密后的负载，和上面的 Authentication 一样是 IPSec 的内容，在 IPv6 中 IPSec 强制实现
- Destination options：在目的主机处理

IPv6 Enhancements

IPv6 扩展了地址空间，从 32 bit 扩展到 128 bit

IPv6 提升了可选机制，将扩展首部放在 IPv6 首部和传输层首部之间，大部分对中继路由不可见，这些首部使得扩展选项更容易

移除了校验和，用于提升路由器的处理速度

IPv6 提升了地址的灵活性，支持任播，即一个集合中只要有至少一个节点收到信息，也扩展了组播地址，可以自动配置地址

IPv6 支持资源分配，使用 traffic class 字段，对于属于不同流的数据报，可以提供尽力而为之上的服务

IPv6 Flow

在 IPv6 中，Flow 是从某个发送者到某个接收者的一系列包。

从端设备视角，流由某个应用产生，并且对于传递有着同样的要求（时延/吞吐量/丢包率），一个流可能包括多个 TCP 连接，而一个应用也有可能产生一个或多个流

从路由器视角，流的属性影响路由器处理流中包的原则，包括路由，资源分配，丢包原则，安全性等。

一个流被 Flow Label 和源/目的地址 唯一标识，在流开始前为其定义服务需求和分配 Label，而路由器处理时仅仅根据 Flow Label 查表，不检查首部其余部分

IPv6 Address

IPv6 的地址长度为 128 bit，且与 interface 关联，但允许一个 interface 关联多个单播地址。IPv6 中有三种地址：单播，组播，任播。表示地址时将 128 bit 分为 16 bit * 8 部分，使用 16 进制表示每一部分，中间用 ":" 连接

From v4 to v6

不可能在同一时间内全世界的设备从 IPv4 升级到 IPv6，故网络必将在 v4 和 v6 混合的情况下运行，此时有两种解决方案

Dual Stack：某些路由器有两套协议栈，可以分别处理 IPv4 和 IPv6，并进行两者间的转换。但是转换过程中可能会丢失某些 IPv6 的功能

Tunneling：在不支持 IPv6 的路由器间，将 IPv6 封装在 IPv4 内进行传播

Internet Routing

至今为止学习的 routing 都是在非常理想的情况下，即所有路由器都是相同的，且网络是一个“平面”，然而这在实际中是不可能的。实际的互联网有大约 2 亿的设备，这些地址不可能全部存储在路由表中，而按照路由算法进行信息交换会占满整个网络的容量。

Hierarchical Routing

互联网是层次网络，是 network of network，故可以将互联网分为不同的区域，由每个区域网络的所有者决定网络内的路由

将路由器聚合为一个区域，称为自治系统（AS, Autonomous System）。AS 由全局唯一的 AS number 标识

同一个 AS 内的路由运行相同的路由协议，称为 **Intra-AS routing protocol**，不同的 AS 中的路由器可运行不同的 Intra-AS routing protocol

负责路由目的地不在本 AS 内的路由器称为 Gateway router，Gateway router 和其他 Gateway router 运行 **Inter-AS routing protocol**，同时也与 AS 内的路由器运行 Intra-AS routing protocol

AS 也是分层级的，其中有 transit 和 stub 之分。transit AS 承载着经过其的流量，而 stub AS 只能由目的地在其中的流量进入，目的地不在其中的流量不会经过 stub AS

IGP and EGP

IGP (Interior Gateway Protocol) 用于 Intra-AS routing

- IGP 仅在 AS 内交换路由信息
- IGP 可以专注于路由性能
- 不同 AS 的 IGP 可以不同

EGP (Exterior Gateway Protocol) 用于 Inter-AS routing

- 运行 EGP 的路由器需要一些 AS 之外的网络信息
- EGP 比起性能更关注可达性
- EGP 可能受到非技术原因影响（政治，经济，etc.）

常用的 IGP 协议有 RIP（基于 DV），OSPF（基于 LS），IGRP

常用的 EGP 协议有 BGP（基于 PV）

> Distance Vector: 首代路由算法，每个节点与邻居节点交换距离向量，距离向量的内容是与其相连的网络的链路代价以及到每个目的地的估计代价和下一跳地址，路由表的建立通过 DV 信息的交换进行
>
> Link State: 第二代路由算法，路由器初始化时将与其直接相连的链路代价广播到全网，随着所有路由器的广播结束，每个路由器都有整个网络中所有链路代价的信息，根据此使用算法得到路由表（SSSP 算法，如 Dijkstra），若发生重大变化，重新广播链路代价
>
> Path Vector: 用于 AS 之间，不关注链路代价，仅仅将 AS 可达的 AS 及其中继 AS 包含在 PV 中传递出去，每个 gateway router 根据实际情况选择路由路径

RIP and OSPF

RIP

Routing Information Protocol，基于 DV 实现，用跳数来衡量距离，最大值为 15 跳

RIP 使用组播向运行 RIP 的 router 发送信息

运行 RIP 的路由器每 30s 与邻居交换 DV 信息，若一个邻居 180s 以上没有传递 DV 信息，则认为其断线。每个 DV 信息中最多包含 25 个目的网络，封装在 UDP 中传输

之后队列长度代替了跳数成为了链路代价的度量标准

OSPF

Open Shortest Path First，基于 LS 实现，取代了 RIP

OSPF 维护链路状态信息，并且每 10s 向整个 AS flooding。报文直接封装在 IP 报文中

在每个路由器中，网络的拓扑信息存储为一个有向图，包括了 router node 和 network node（分为 transit 和 stub），边的权值即为链路代价，使用 Dijkstra 算法得出最短路径构建路由表

OSPF 相较于 RIP，优点在于

- 安全：OSPF 的报文信息都需要经过认证，防止恶意攻击
- 允许多条相同代价的路径
- 对于每条链路，基于 ToS 定义不同的代价
- 整合了 unicast 和 multicast
- 对于大型的 AS，提供层次 OSPF

层次 OSPF 是将 AS 划分为一个个 Area，每个 Area 以一个 32 bit 的 Area-ID 唯一标识，Area 内的路由仅知道 Area 的网络拓扑，将路由信息的 flooding 限制在 Area 内。由 Area Border Router (ABR) 综合通往其他 Area 的路由信息。

每个 Area 必须连接到 backbone area

可以将路由器分为

- Internal Router：仅知道 area 内部的路由信息
- Area Border Router：将本 area 的路由信息总结后发送给其他 ABR
- Backbone Router：在 backbone area 内的 router
- AS boundary (ASB) Router：位于 AS 边界，运行 EGP 的 router

RIP vs OSPF

对于 RIP

- 配置简单，适用小型网络
- 可分布式实现
- 最短路径收敛速度慢

- 网络是平面的

对于 OSPF

- 收敛速度块
- 支持不同服务类型
- 支持认证
- 支持层次型网络，可以用于大规模复杂网络
- 集中式算法
- 每个节点要维护全局的拓扑信息
- 配置复杂

BGP

Border Gateway Protocol，每个 AS 使用 BGP 向互联网告知其存在。

可以使用 eBGP 从邻居 AS 获取网络可达性信息

可以使用 iBGP 在 AS 内部的 BGP 路由间传递可达性信息

根据可达性和具体策略选择最佳路径

BGP router 之间通过 TCP 连接交换信息：BGP session。路由信息通过 CIDR 标准的前缀传达，在此过程中可进行路由聚合

BGP 信息可分为

- OPEN：用于建立连接
- UPDATE：传递新路径/撤销旧路径
- KEEP-ALIVE：回复 OPEN/保持在没有 UPDATE 信息时的连接
- NOTIFICATION：错误报告/断开连接

BGP 的过程为

- 一个 BGP router 给邻居发送一个 OPEN，若其收到则会回复 KEEP-ALIVE
- 两个 router 定期发送 KEEP-ALIVE 或 UPDATE
- 每个路由器维护一个数据库，包含其可达的网络和途中经过的所有 AS
- 每当数据库内容发生变化时发送 UPDATE 信息（增量更新）

在 BGP 的路径信息中包含了经过的所有 AS 和下一跳信息，路由器如果在经过的 AS 中发现了本 AS，则会拒绝这条信息，防止产生环路

IP Multicasting

Multicast，组播，即仅通过一次发送将数据报发送到多个 Host。组播地址是 IPv4 中的 D 类地址，代表分布在一个或多个网络上的一组主机

组播需要生成一颗组播树，将成员连接起来，仅仅在树之上复制并转发

Multicast Service Model

组播的实现是间接的。主机将 IP 数据包发到组播组（目的地址为组播地址），路由器将其转发给组播组成员。

组播地址是 D 类地址，最高位是 1110，之后是 28 bit 的组播 ID，对于 IPv6 来说，8 bit 前缀，4 bit flag，4 bit scope，112 bit 组播 ID

> 224.0.0.0～224.0.0.255 为预留的组播地址（永久组地址），地址224.0.0.0保留不做分配
>
> 224.0.1.0～224.0.1.255是公用组播地址，可以用于Internet
>
> 224.0.2.0～238.255.255.255为用户可用的组播地址（临时组地址），全网范围内有效
>
> 239.0.0.0～239.255.255.255为本地管理组播地址，仅在特定的本地范围内有效。

在 IP 层面，要将组播地址转换成包含组播成员所在网络的列表。在 MAC 层面，将组播 IP 转换为组播 MAC

> 组播 MAC 高 24 bit 为 0x01005e，低 23 bit 为组播 IP 的低 23 bit

对于维护一个组播组，局部网络可以使用 IGMP 协议，广域网可以使用 DVMRP，MOSPF，PIM 等协议

IGMP

用于主机和 router 在 LAN 上交换组播信息

主机向路由发送 report 以加入或退出一个组播组。主机不用显式退出组播组

路由向主机定期发送 query，属于某个组播组的主机必须回复

对于 IGMP，有两个特殊的组播地址

- 224.0.0.1：子网上的所有组播组
- 224.0.0.2：子网上的所有路由

在 LAN 上，选出一个 router 作为 querier，定期向 224.0.0.1 发送一个 TTL 为 1 的 Query 信息（不会被转发）。主机在随机间隔后发送其属于的组播组的信息作为 reply（随机间隔是为了防止碰撞），TTL 同样为 1，不被转发。同一组的成员若监听到该回复则不再发送。这样路由器就获得了本局域网内所有的组播组的信息，一定时间内没有回复则认为该组播组所有成员

均已退出。

在 IGMPv2 中路由器可以 query 某个特定的组播组，主机也可以显式退出一个组播组而不用等待到超时

IGMPv1 和 v2 均不能记录组播内容的发布者（任意成员均可以在组播组内发送内容），但是在 v3 中可以让主机选择允许的发送者，并且将名单之外的流量阻拦在路由器之外

在 IPv6 网络中使用 ICMPv6 进行组播管理

Multicast Routing

需要找到一个生成树，连接所有组播成员。可以分为两种

- Shared-Tree：所有成员共用同一棵树
- Source-Based：根据发送方的不同而不同的树

Source-based tree

一种方案是使用 **Shortest Path Tree**，即从源到接收者的最短路径组成的树，可以通过 Dijkstra 算法生成

另一种方案是 **Reverse Path Forwarding**，即基于从**接收者**到**发送者**的最短单播路径，其思想可描述为

- 如果组播数据报从**通向发送方的最短路径的上一跳**而来，则将其 flooding 到所有出链路
- 否则忽视

其结果是一颗反向的 SPT，但是如果链路代价不对称，结果会很差

在转发时也可以进行剪枝，若子树中已经没有组播成员了，则反向发送 prune 信息，表示从这个链路向下已经没有组播成员了，不需要进一步转发

Shared-Tree

一种方案是 **Steiner Tree**，即计算出连接所有有组播成员的路由器的最小代价树，问题本身是 NP-完全的，但是存在很好的启发式算法

Steiner Tree 并未应用在实际网络中，原因有

- 计算代价过大
- 需要整个网络的信息
- 不灵活，当有路由加入/离开时需要重新计算

实际中采用的方案是 **Center-based Tree**，即选定一个 router 为 center，其余路由器向 center 发送 join 信息，则该信息或是到达现有的树的一枝或是到达 center，无论何种情况，join 信息经过的路径会成为树的新的一枝

Multicasting Routing Protocols

DVMRP (Distance Vector Multicast Routing Protocol)，采用 source-based tree 的 reverse path forwarding 方案。每过一分钟重置剪枝状态，让到达的组播数据包再次 flooding

PIM (Protocol Independent Multicast)，不依赖于特定的单播算法，分为 Sparse 和 Dense 两种模式。Sparse 模式使用 center-based tree，Dense 模式和 DVMRP 类似

MPLS

Multiprotocol label switching

基本思想为使用定长的 Label 进行高速的 IP 转发（相比于基于 IP 地址的转发），借鉴了虚电路的思想，是两种方案的中和