概述

计算机网络

将分布在不同地理位置上的具有独立工作能力的主机、终端及其附属设备，用通信设备和通信链路连接起来，并配置网络软件，以实现计算机资源共享的系统

Internet

是网络和网络之间串成的庞大网络，这些网络以一组标准的网络协议族相连，连接全球几十亿个设备，形成逻辑上单一的巨大国际网络

协议（Protocol）

协议，是网络协议的简称，网络协议是通信计算机双方必须共同遵从的一组约定。他定义了再两个或者多个通信实体之间交换报文的格式和顺序，以及报文接收或发送报文或其他事件所采取的动作

三要素是语义、语法、语序

因特网服务供应商（ISP）

即向广大用户综合提供互联网接入业务、信息业务、和增值业务的电信运营商

网络边缘

通常把和因特网相连的计算机和其他设备称为端系统，它们位于因特网的边缘

接入网

是指将端系统物理连接到其边缘路由器的网络。

边缘路由器是端系统到任何其他远程端系统路径上的第一台路由器

常见接入有：DSL（数字用户线），电缆，FTTH（光纤到户），拨号和卫星

物理媒体

引导型媒体：电波沿着媒体传输

非引导型媒体：电波在空气或者外层空间传播

有：双绞铜线、同轴电缆、光纤、陆地无线信道、卫星无线电信道

网络核心

网络核心是指由因特网端系统的分组交换机和链路构成的网状网络，主要功能是建立传输的通道

分组交换（packet Switch）

分组：为了从源端系统向目的端系统发送一个报文，源将长报文划分为较小的数据块，被称为分组（packet）

在源和目的地之间，每个分组都要通过通信链路和分组交换机（packet switch），常见分组交换机有：路由器和链路层交换机

存储转发传输（store-and-forward transmission）：是指交换机能够开始向输出链路输出该分组的第一个比特之前，必须接收到整个分组。即不能传输已经接收的比特，而是要将这些分组的比特缓存起来，仅当路由器接收到该分组的全部比特之后，开始向输出链路传输

电路交换（circuit switch）

在电路交换网络中，端系统间通话期间，需预留端系统间沿路径通信所需要的资源。在发送方能够发送信息之前，该网络必须在发送方和接收方之间建立连接，该连接被称为 **电路（circuit）**

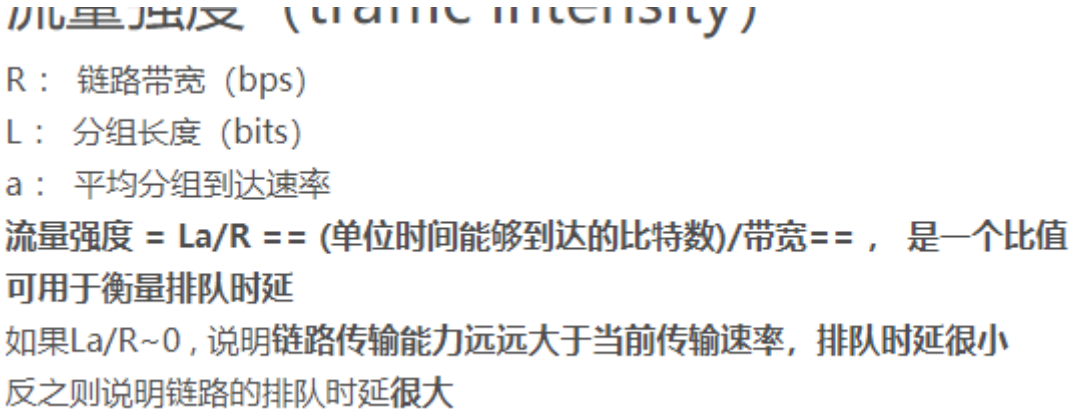
传输时延（transmission delay）

将所有分组的比特从交换机推向链路所需要的时间

传播时延（propagation delay）

比特在链路上传播所需的时间

流量强度（traffic intensity）



吞吐量（throughput）

吞吐量是指对网络、设备、端口、虚电路或其他设施，单位时间内成功地传送数据的数量

因特网协议栈及其功能

> **二、协议栈各层次的主要任务**
>
> **① 应用层：是网络应用程序及其应用层协议存留的层次。该层包括了所有与网络相关的高层协议，如文件传输协议（FTP）、超文本传输协议（HTTP）、远程终端协议（Telent）、简单邮件传送协议（SMTP）、因特网中继聊天（IRC）等。**
>
> **② 传输层：使源端主机和目标端主机上的对等实体可以进行会话。该层有两种服务质量不同的协议：传输控制协议（TCP）和用户数据报协议（UDP）。**
>
> **③ 网络层：通过路径选择把分组发往目标网络或主机，进行网络拥塞控制以及差错控制，是整个TCP/IP协议栈的核心。**
>
> **④ 链路层：负责网络层和物理层之间的通信，将网络层接收到的数据分割成特定的可被物理层传输的帧，并让物理层进行实际的数据传送。**
>
> **⑤ 物理层：将帧中的一个个比特从一个节点移动到下一个节点。该层的协议仍与链路相关，并进一步与链路的实际传输媒体相关。**

应用层报文、运输层报文段、网络层数据报、链路层帧

应用层报文：应用程序想发送和通过传输层的数据；

运输层报文段：由传输层生成并且封装有传输层头信息的应用层报文

网络层数据段：封装有网络层头信息的运输层报文段

链路层帧：封装有链路层头信息的网络层数据段

OSI七层模型

应用层，表示层，会话层，运输层，网络层，链路层，物理层

应用层

HTTP

超文本传输协议，用TCP作为支撑运输协议，端口号为80

因为HTTP服务器并不保存关于客户的任何信息，所以被称为无状态协议

持续连接和非持续连接

持续连接：每个请求/响应报文都在同一个TCP连接上传输

非持续连接：每个请求/响应报文都在不同的TCP连接上传输

往返时间（RTT）

round trip time：指分组从客户端发送到服务器，再返回客户端所花费时间

HTTP报文格式

HTTP请求报文：有请求行（方法字段，URL字段，版本字段）、首部行、实体体（使用GET方法时为空，使用POST方法时才使用）

HTTP响应报文：有状态行（协议版本字段，状态码，状态信息），首部行，实体体

---

cookie

cookie有四个组件：HTTP请求报文中的cookie首部行，HTTP响应报文中的cookie首部行，用户端系统本地的cookie文件，位于Web站点后端的数据库

首先，该用户在第一次与Amazon.com联系，当请求报文发送到Amazon服务器的时候，该Web站点便会产生一个唯一的识别码（ID 1678）作为一个索引在后端数据库产生一个表项。接下来Web站点发送一个包含了Set-cookie：的首部行的响应报文进行响应，其中含有着该唯一识别码

当用户的浏览器接收到了这个响应报文时，会看到Set-cookie中包含的唯一识别码，那么就会在他本地管理的cookie文件中加一行，该行包含着服务器主机名和识别码（amazon:1678）。当客户今后每次访问该Web页面的时，其浏览器就会查询cookie文件并抽取对这个网站的识别码，并放到HTTP请求报文中。比如发往Amazon服务器的每个请求报文都含有(Cookie:1678)

于是，Amazon服务器就可以追踪客户在Amazon站点的活动，可以确切的知道该用户在什么时间什么地点访问了哪个页面，比如说常见的购物车就是这个道理

Web缓存

Web缓存器（Web cache），也叫代理服务器（proxy server）

过程：浏览器先创建到Web缓存器的TCP连接，发出HTTP请求；如果Web缓存器有该对象的副本，则返回响应报文；如果没有，则创建一个该对象初始服务器的TCP连接，得到响应报文后在本地缓存，再响应回客户浏览器

内容分发网络（CDN）

条件GET

请求报文中包含一个If-Modified-Since首部行，如果没有修改过，则响应报文返回304 not MODIFIED，并且实体体为空

因特网电子邮件系统

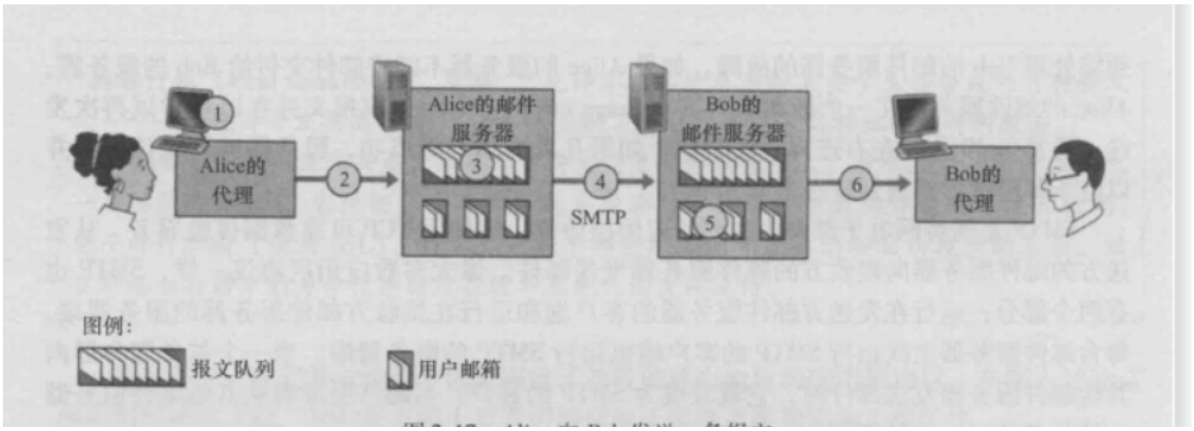
有三个组成部分：用户代理（user-agent），邮件服务器（mail server），**简单邮件传输协议（SMTP）**

SMTP

端口号25，支撑运输协议是TCP，运行在发送方和接收方的邮件服务器上

它限制所有邮件报文的体部分只能采用简单的7比特ASCii表示

- Alice 调用她的邮件代理程序并提供 Bob 的邮件地址（例如 bob@ someschool. edu），撰写报文，然后指示用户代理发送该报文。
- Alice 的用户代理把报文发给她的邮件服务器，在那里该报文被放在报文队列中。
- 运行在 Alice 的邮件服务器上的 SMTP 客户端发现了报文队列中的这个报文，它就创建一个到运行在 Bob 的邮件服务器上的 SMTP 服务器的 TCP 连接。
- 在经过一些初始 SMTP 握手后，SMTP 客户通过该 TCP 连接发送 Alice 的报文。
- 在 Bob 的邮件服务器上，SMTP 的服务器端接收该报文。Bob 的邮件服务器然后将该报文放入 Bob 的邮箱中。
- 在 Bob 方便的时候，他调用用户代理阅读该报文。



不使用中间邮件服务器

邮件报文格式

From：

To：

Subject

邮件访问协议

对于接收方，从邮件服务器获取邮件时使用的协议

- 第三版邮局协议（POP3）：端口号110，
- 因特网邮件访问协议（IMAP）：
- HTTP：电子邮件从A浏览器发送到他的邮件服务器，使用的是HTTP而不是SMTP；而邮件服务器之间是用SMTP

FTP

File transfer Protocol：文件传输协议

运行在TCP上，用两个并行的TCP连接来传输文件，是带外控制

控制连接：端口号21

数据连接：端口号20

FTP服务器必须在整个会话期间保留用户的 **状态（state）**

控制连接一直存在，数据连接在每次传输不同文件，都会重建。

DNS

因特网目录服务：运行在UDP上，端口号53

是由①一个由分层的DNS服务器实现的分布式数据库②一个使得主机能够查询分布式数据库的应用协议组成

分类：根DNS服务器，顶级DNS服务器（TLD）、权威DNS服务器、本地DNS服务器

递归查询 和 迭代查询：

1. 主机向本地服务器的查询一般都是采用递归查询。

> 递归查询是一定要给出准确结果的。因为主机是一定要获得准确结果的。

2. 本地DNS服务器向根DNS服务器查询采用迭代查询。

> 迭代查询不用给出准确结果。

解析过程：

主机-->本地DNS 递归查询。
本地DNS-->根DNS，根给出一个顶级DNS，
本地DNS-->顶级DNS，顶级DNS给出权威DNS，
本地DNS-->权威DNS，权威DNS给出查询结果，
本地DNS--> 主机

2.5.5 DNS记录和报文

共同实现DNS分布式数据库的所有DNS服务器存储了 **资源记录（Resource Record，RR）**RR提供了主机名到IP地址的映射

资源记录为一个4元组

`(Name,Value,Type,TTL)`

TTL是该记录的生存时间，Name和Value根据Type而定：

- Type为A时，Name是主机名，Value就是IP地址
- Type为NS时，Name是个域，Value是个知道如何获得该域中主机IP地址的权威DNS服务器的主机名
- Type为CNAME，则Value是别名为Name的主机对应规范主机名
- Type为MX，则Value是别名为Name的邮件服务器的规范主机名

P2P

对等网络，即对等计算机网络，每个对等方能够向任何其他对等方重新分发它已经接收到的该文件的任何部分

分发时间：所有N个对等方得到该文件的副本所需要的时间

- 对于CS结构：因为服务器需要向N个对等方每个都传送一个文件的副本，所以共需要传输NF比特，则

$$D_{cs} = max\{\frac{NF}{u_s}, \frac{F}{d_{min}}\}$$

- 对于P2P结构，D为

$$D_{p2p} = max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=i}^{N} u_i}\}$$

运输层

周知端口号

0~1023范围内的端口号

常见的有：HTTP80，FTP20（数据）21（控制），DNS（53），SMTP（25），DHCP（68），Telnet 23

多路复用和多路分解

多路复用：在源主机从不同的套接字中收集数据块，并给每个数据块封装上首部信息从而生成报文段，然后将报文段传递到网络层

多路分解：把运输层报文段中的数据交付到正确的套接字的工作

套接字（socket）

是应用程序通过网络协议进行通信的接口，是应用程序与网络协议栈进行交互的接口

UDP的套接字由一个二元组全面标识，包含目的IP地址和目的端口号

TCP的套接字由一个四元组全面标识，包含目的IP地址，目的端口号，源IP地址，源端口号

- 两个具有不同源IP地址或者源端口号的到达TCP报文会被定向到不同的套接字，除非是创建连接的请求

UDP

全程用户数据报协议，是一种无连接的传输层协议

UDP的优点

- 关于发送什么数据以及何时发送的应用层控制更加精细
- 无需连接建立
- 无连接状态
- 分组首部开销小

UDP报文段结构

UDP首部有8个字节，4个字段（源端口号，目的端口号，长度，检验和）

可靠数据传输

rdt1.0：经完全可靠信道的可靠数据传输

rdt2.0/2.1：经具有比特差错信道的可靠数据传输

- 引用了肯定确认ACK和否定确认ACK
- 引用了序号（避免ack丢失）

rdt3.0：经具有比特差错的丢包信道的可靠数据传输

TCP

全称为传输控制协议，是一种面向连接的、可靠的、基于字节流的传输层通信协议

提供的是全双工服务：即若主机AB之间存在一条TCP连接，数据从A流向B的同时，也可以从B流向A

是点对点的

三次握手

客户首先用一个特殊的TCP报文发送给服务器，服务器再用一个特殊的TCP报文进行响应，最后客户再用第三个特殊报文段做响应，前两个报文段不承载有效载荷，第三个报文段可以承载有效载荷，这种建立连接的过程叫做三次握手

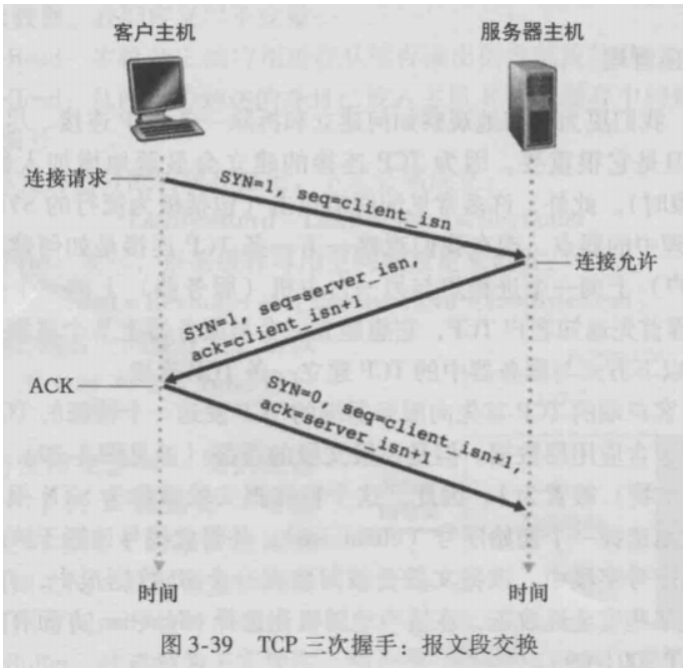


图 3-39　TCP 三次握手：报文段交换

四次挥手

由于TCP连接是全双工的,断开一个TCP连接,需要客户端与服务器发送四个包来确认连接的断开

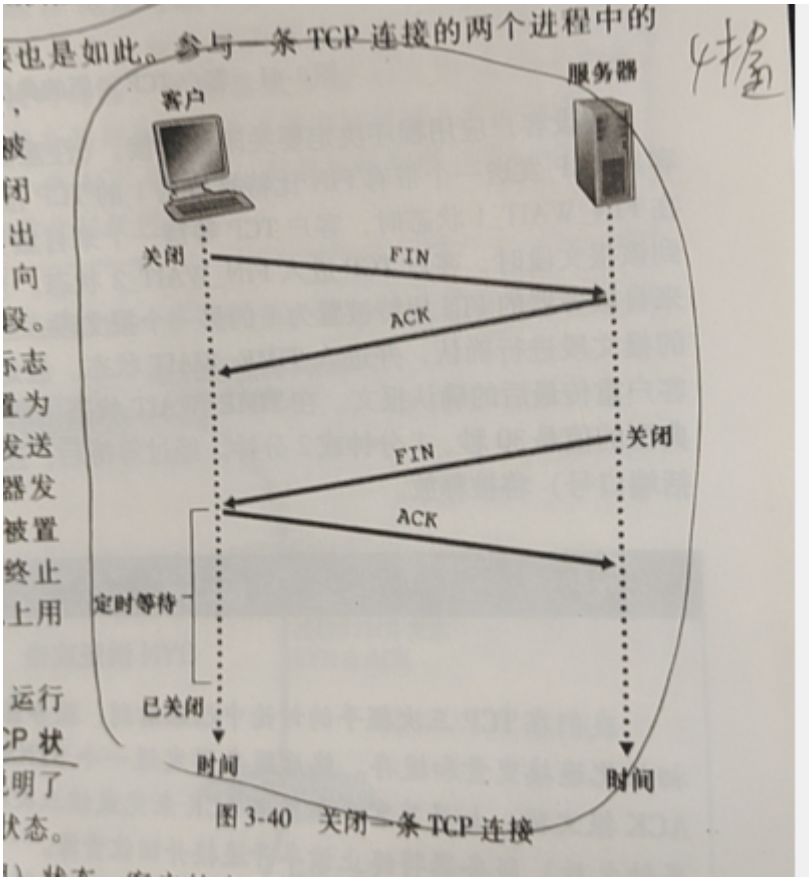


图 3-40　关闭一条 TCP 连接

MSS

最大报文段长度：表示TCP传输的最大数据的长度

MTU

最大传输单元：链路层帧的最大长度

TCP报文段

一般是20字节，包括源端口号，目的端口号，序号，确认号七七八八的

往返时间估计和超时

SampleRTT：样本RTT，指某报文从被发出到对该报文段的确认被收到之间的时间量

EstimatedRTT：均值RTT，公式为(1-a)EstimatedRTT+aSimpleRTT(a一般取0.125)

DevRTT：用于估算SampleRTT偏离EstimatedRTT的程度

$$DevRTT = (1-\beta) \times DevRTT + \beta \times |SampleRTT - EstimatedRTT|$$

注意到DevRTT是一个SampleRTT和EstimatedRRT差值的一个EWMA，β的推荐值为0.25

TimeoutInterval：超时间隔，

$$TimeoutInterval = EstimatedRTT + 4 \times DevRTT$$

超时时隔加倍

当发生超时时间以后，超时间隔会直接加倍，而不是计算出的TimeoutInterval

快速重传

当TCP发送方接收到对相同数据的3个冗余ACK，则马上进行重传，即在该报文段的定时器过期之前重传丢失报文段

TCP的差错恢复机制

TCP的差错恢复机制是介于GBN和SR之间的。TCP是累积确认，同时它也会选择重传

TCP的流量控制

发送方维护了一个接收窗口rwnd

- LastByteRead：主机B上的应用进程从缓存读出的数据流的最后一个字节的编号
- LastByteRcvd：从网络到达的并且已经放到主机B缓存中的数据流的最后一个字节的编号

  因为TCP不允许已分配的缓存溢出，所以必须满足

$$LastByteRcvd - LastByteRead <= RcvBuffer$$

  接收窗口用rwnd表示，根据缓存可用空间的数量设置

$$rwnd = RcvBuffer - [LastByteRcvd - LastByteRead]$$

TCP拥塞控制

看以前写的吧

网络层

转发（forwarding）

将分组从一个输入链路接口转移到适当的输出链路接口的路由器本地动作

路由选择（routing）

是指确定分组从源到目的地所采取的端到端的网络范围处理过程

最长前缀匹配规则（longest prefix matching rule）

掌握用此规则建立转发表的过程

交换

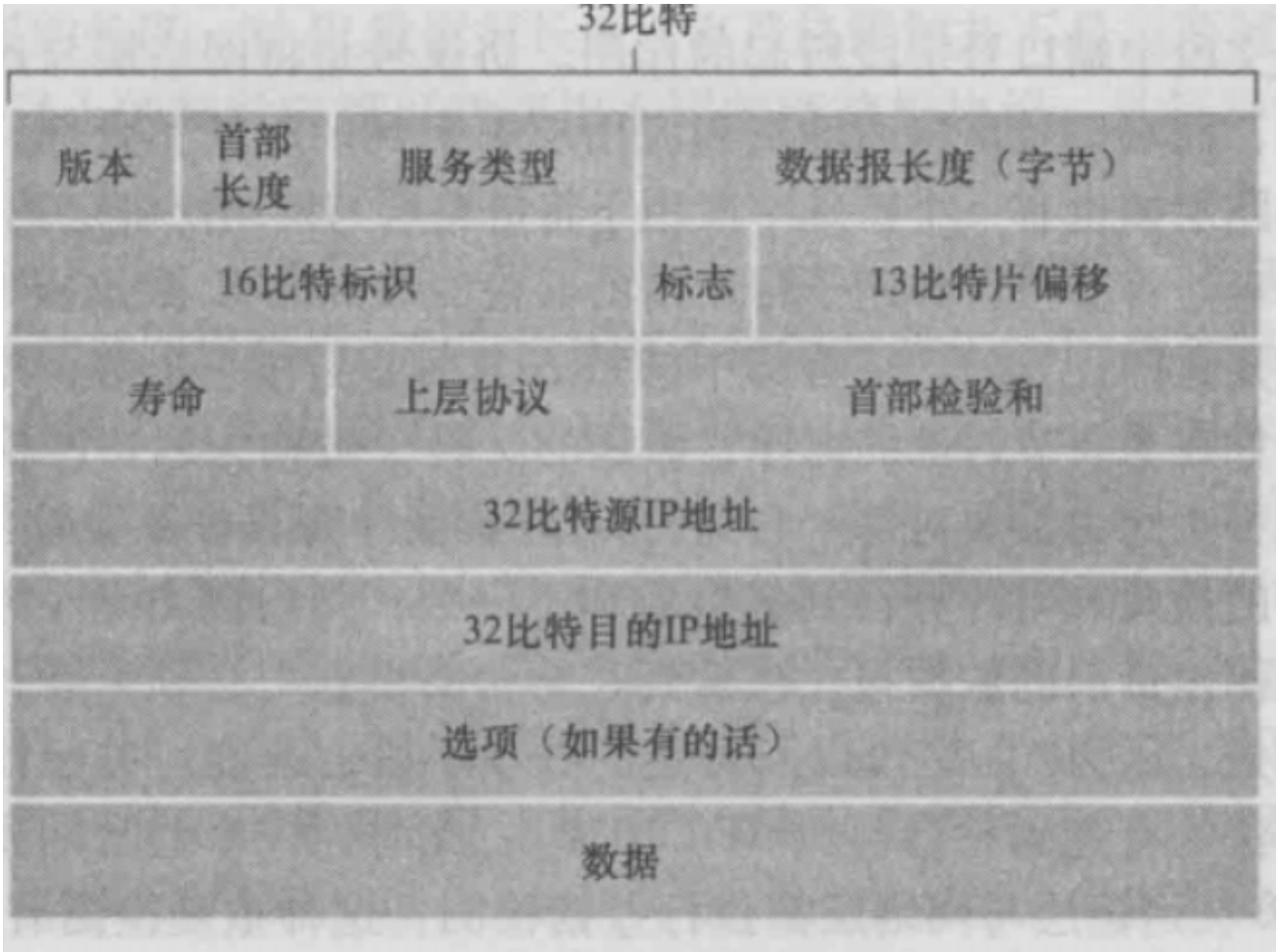
有经内存交换、经总线交换、经互联网络交换

分组调度

有先进先出（FIFO，也称先来先服务FCFS）、优先权排队（将分组排队队列分为高优先级和低优先级，在同优先级采取FIFO）、循环加权公平排队

IPv4数据报格式

IP数据报具有至少20字节的首部



数据报分片

一个链路层帧能承载的最大数据量称之为 **最大传输单元（Maximum Transmission Unit，MTU）**

由于链路的MTU比IP数据报的长度小，所以需要将IP数据报中的数据分片成多个较小的IP数据报，每个这些较小的数据报被称为片（fragment）

每个片也要加上IP数据报的首部

计算片偏移量时，每个片的数据要除以8

IPv4编址

主机和物理链路之间的边界叫做 **接口（interface）**

一个IP地址和一个接口关联，而不是和包括接口的主机或者路由器关联

每个IP地址的长度为32比特，用点分十进制法标记

CIDR

无类别域间路由选择

形式为a.b.c.d/x的记法，有时称为子网掩码（network mask），指示了32位比特中最左侧x位为子网地址，地址的x最高比特位被称为IP地址的网络位，且经常被称为该地址的前缀（prefix）

分类编址

具有8 、16 、24比特的子网地址的子网被称为A、B、C类网络

比如一个C类（/24）子网能够容纳$2^8 - 2 = 254$台主机（全0和全1用于特殊用途）

255.255.255.255被称为广播地址，当目的地址为它是，会被发送到同网络的所有主机

ICANN

因特网名字和编号分配机构

DHCP

动态主机配置协议，也经常被称为 **即插即用协议（play-and-play protocol）**或者 **零配置协议（zeroconf)**

网络管理员可以通过DHCP来进行主机地址的配置，可以给某给定主机每次与网络连接时能得到一个相同的IP地址，或者某主机将分配一个临时IP地址，同时可以让主机得知他的子网掩码、第一条路由器地址、本地DNS服务器地址

DHCP是客户端-服务器协议（用UDP承载，端口号68），如何从DHCP服务器获得IP地址呢：

- DHCP服务器发现：客户端通过广播目的地址255.255.255.255，广播到同一个网络的DHCP服务器
- DHCP服务器提供：当服务器接收到一个DHCP发现报文的时候，使用DHCP提供报文，该报文向子网所有节点广播
- DHCP请求
- DHCP ACK

一旦客户端收到DHCP ACK，那么客户端就能在租期内使用DHCP分配的IP地址

集中式路由选择算法

用完整的、全局的网络知识计算出从源到目的地之间的最低开销路径。拥有全局状态信息的算法通常称之为 **链路状态（Link State，LS）**算法

比如Dijkstra算法

分散式路由选择算法

路由器以迭代、分布式的方式计算出最低开销路径，每个节点没有关于网络的全部信息，它只要拥有与其直接相连链路的开销知识便可工作

比如 **距离向量算法（Distance-Vector，DV算法）**

Bellman-Ford方程：

我们先要了解一个存在于最低开销路径的开销之间的一种重要关系，也就是Bellman-Ford方程

$$d_x(y) = min_v\{c(x,v) + d_v(y)\}$$

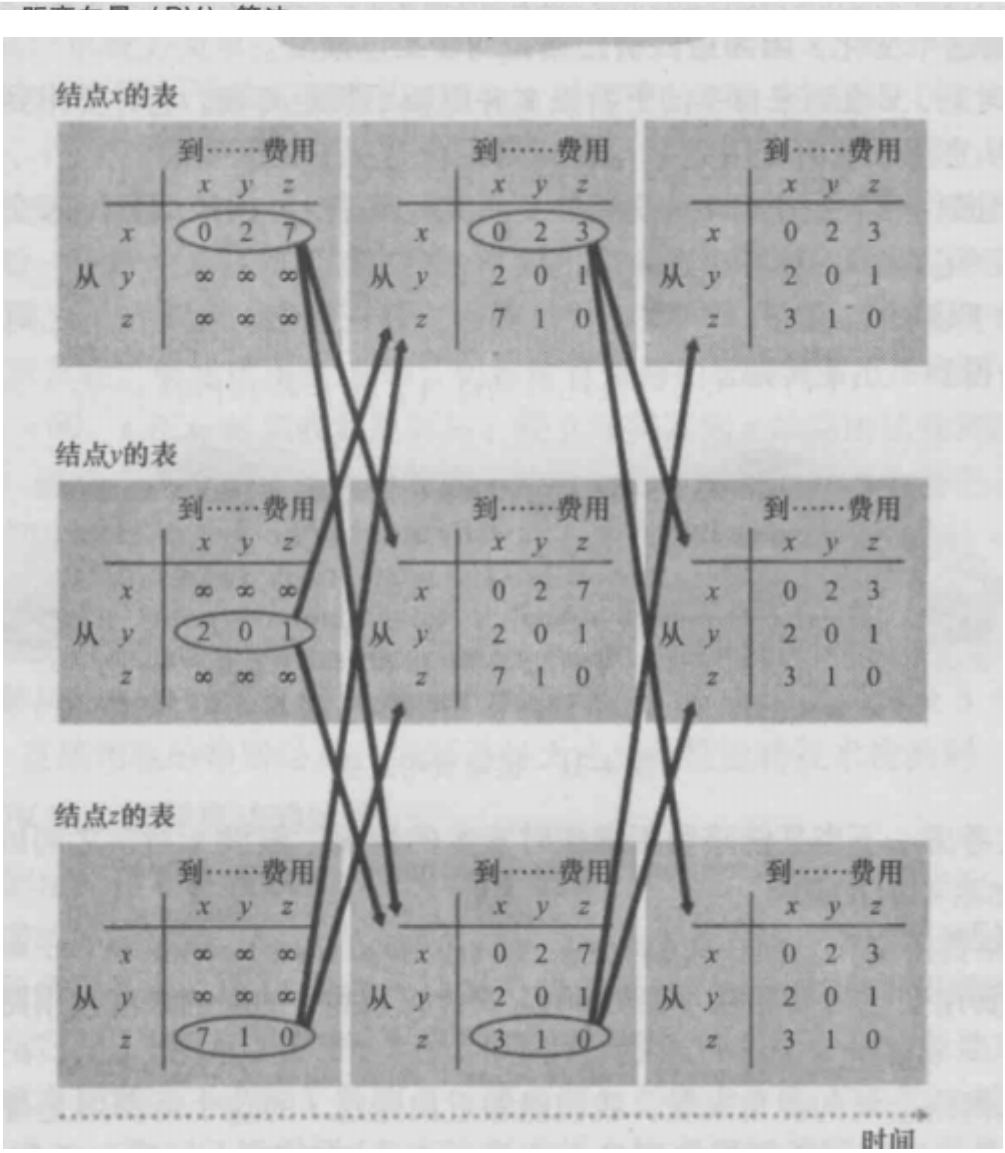
其中$d_x(y)$表示从节点x到节点y的最低开销，min_v表示对x 的所有邻居，这是种非常逆天的迭代，看得懂就不言传了

基本思想如下：

- 首先，每个节点x以$D_x(y)$开始，对在N中的所有节点y，估计一个从x到y的最低开销路径的开销。令$D_x = [D_x(y) : y \in N]$是节点x的距离向量，该向量是从x到N中所有其他节点的开销估计向量，里边包含着x到所有目的地开销的估计值
- 接着使用DV算法，每个节点x维护下列路由选择信息
  - 对于每个邻居v，从x直接相连v的开销为c（x,v）
  - 节点x的距离向量D_x
  - 它的每个邻居v的距离向量，即$D_v = [D_v(y) : y \in N]$
- 然后，在该分布式、异步算法中，每个节点不时地向它的每个邻居发送它的距离向量副本，当节点x收到了它任何一个邻居v的新距离向量，它会保存起来，然后用Bellman-Ford方程进行更新

$$D_x(y) = min_v\{c(x,v) + D_v(y)\} \quad 对N中的每个节点$$

- 如果节点x的距离向量因为这个更新步骤而改变，那么它又会对它的邻居发送其更新后的距离向量，这进而让他的邻居们更新他们自己的距离向量，每个开销估计$D_x(y)$收敛到$d_x(y)$即为节点x到节点y实际的最低开销路径



可能会出现路由选择环路导致的无穷计数问题，用毒性逆转

LS和DV算法的比较

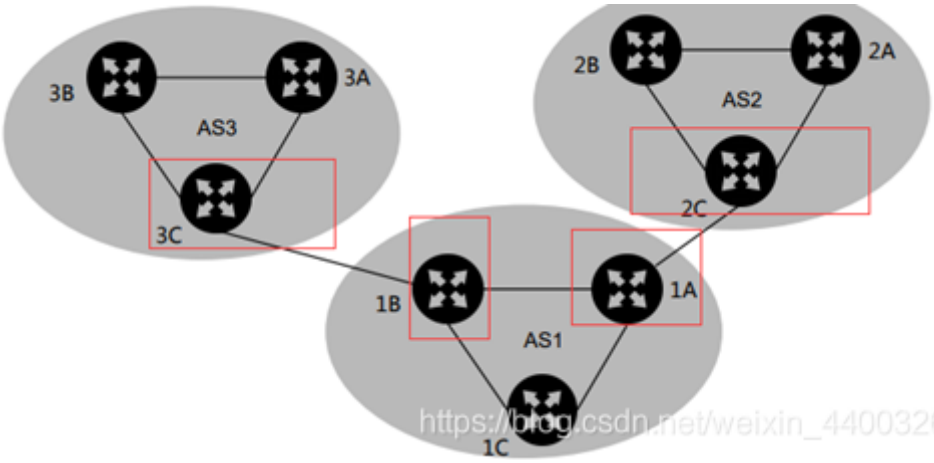
DV算法每个节点仅需要知道它直接相连的邻居的信息，LS需要全局信息

收敛速度上，DV算法较慢，且可能出现无穷计数问题

健壮性上，LS算法健壮性更强，DV算法可能会将错误信息广播到全部链路上

自治系统（autonomous system，AS）

互联网按组织边界划分为多个 自治系统。每个自治系统由运行相同路由协议和路由选择算法的路由器组成。如图3个AS组成

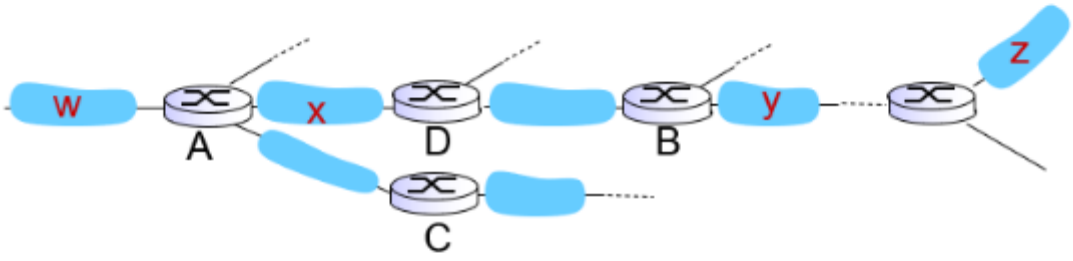


AS可能有复杂结构，该独立机构要负责保证其内部的路由信息的一致性和可用性在AS内的路由器，可以自由地选择寻找路由、广播路由、确认路由以及检测路由的一致性的机制。

RIP

路由信息协议，AS内部的路由选择协议，是一种距离向量协议，使用跳数作为其费用测度，即**每条链路的费用为1**。任何一台路由器的距离向量是从这台路由器到该AS中子网的最短路径距离的当前估计值

- 距离度量值：跳数（最大15跳），每个链路的开销均为1
- 邻居之间每30s通过RIP响应报文（亦称之为“通告”）交换一次距离向量
- 每次通告：至多25条到达目的子网的列表



routing table in router D

destination subnet	next router	# hops to dest
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

以该图为例，目的子网是Z，从D出发，需要7跳

本质和前边的DV一样，都用了那个不等式迭代计算最短路径

开放最短路优先（OSPF）

AS内部的路由选择协议，是一种链路状态协议，使用洪泛链路状态信息和Dijkstra算法。

使用OSPF的时候，路由器会向AS内全部的路由器广播路由选择信息

边界网关协议（Broder Gateway Protocol，BGP）

是一种分布式、异步的协议

在BGP中，分组并不是路由到一个特定的目的地址，相反是路由到一个CIDR化的前缀，每个前缀表示一个子网或者一个子网的集合，一个目的地可以采取138.16.68/22的形式，其路由器转发表的表项为（前缀，接口号）

网关路由器（Gateway router）：位于AS的边缘，直接连接到其他AS中的一台或者多台路由器

内部路由器（internal router）：仅连接在他自己AS中的主机和路由器

前缀及其属性被称为 **路由（route)**，其中两个重要的属性为AS-PATH（包含了通告已经通过的AS的列表）和NEXT-HOP(是AS-PATH起始的路由器接口的IP地址)

跨越两个AS之间的BGP连接叫做外部BGP连接（eBGP），在相同AS中的两台路由器的BGP会话叫做内部BGP连接（iBGP）

ICMP

因特网控制报文协议，用途是差错报告，如果IP路由器不能找到一个通往HTTP请求中所指定的主机的路径，该路由器就会向你的主机生成并发出一个ICMP报文提示错误

链路层

节点（node）

把运行链路层协议的任何设备都称之为节点，比如主机、路由器、交换机等

链路（link）

沿着通信路径连接相邻节点的通信信道称之为链路

媒体访问控制协议（MAC）

规定了帧在链路上的传输规则

网络适配器（network adapter）

是链路层的主体部分，也称为网络接口卡（network interface card，NIC），其核心是链路层控制器

差错检测和纠正技术

差错检测和纠正比特（Error-Detection and Correction，EDC）

- 奇偶校验法（parity bit）：发送方只需要加上一个附加的比特，让这d+1比特中1的总数是偶数（偶校验）。但是只能检查出 **奇数个**比特差错
- 二维奇偶校验：把d比特分为i行j列，每列和每行末尾加1比特，对每行每列计算奇偶值。他对单个比特差错具有检测和纠正能力，也能检测但是不能纠正一个分组中两个比特差错的任何组合
- 校验和方法：和UDP校验和一样，将k个比特整数加起来，用得到的和作为差错检测比特
- 循环冗余检测（Cyclic Redundancy Check，CRC）编码。也被称为多项式编码（polynomial code）。发送方和接收方协商的一个r+1的比特模式被称为生成多项式。计算过程不许不懂，

点对点链路（point to point link）

由链路一端的单个发送方和链路另一端的单个接收方组成

比如点对点协议（PPP）、高级数据链路控制（high-level data link control,HDLC）

广播链路（broadcast link)

它能让多个发送方和接收节点都链接到相同的，单一的，共享的广播信道上

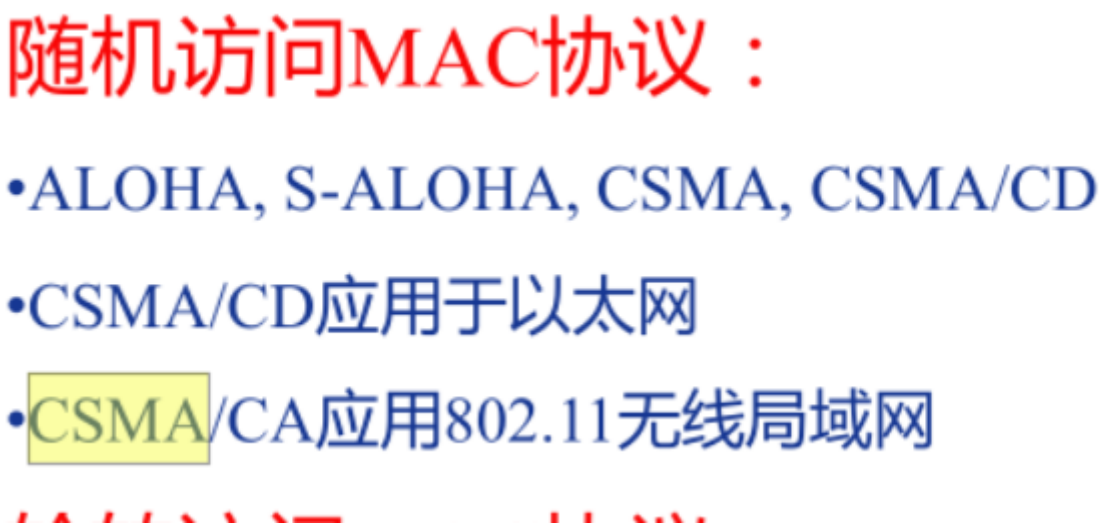
多路访问协议（multiple access Protocol），用来规范节点在他们共享的广播信道上的行为，希望拥有①当一个节点活跃的时候，该活跃节点具有Rbps的吞吐量（ALOHA和CSMA）；②当有M个节点活跃的时候，每个节点的吞吐量接近R/M（轮流协议）

- 信道划分协议（channel partitioning protocol）：
- 随机接入协议（random access protocol）：
- 轮流协议（taking turns protocol）

信道划分协议

- 时分复用TDM：把时间划分为时间帧（time frame），并把每个帧化为N个时隙（slot），每个节点在每个时间帧里的专用的传输速率是R/N bps
- 频分复用FDM
- 码分多址：CDMA

随机接入协议



- 时隙ALOHA：所有帧有L比特组成，时间被分为长度为L/R的时隙。时隙ALOHA是高度分散的，每个节点检测碰撞并独立的选择什么时候重传（需要对时隙同步）。其成功时隙的概率为$p(1-p)^{N-1}$。效率 1/e=0.37
- 无时隙ALOHA：高度分散无时隙，效率为时隙ALOHA的一半
- 载波侦听多路访问（CSMA）和具有碰撞检测的CSMA（CSMA/CD）
    - 载波侦听：一个节点在传输前，会先听信道，如果来自另一节点的帧正在信道上发送，则节点等直到检测到没有传输
    - 碰撞检测：一个传输节点在传输时一直侦听信道，如果他检测到另一个结点正在传输干扰帧，则停止传输

以太网 CSMA/CD 算法

- 网卡从网络层获得数据报，生成链路层帧
- 如果网卡侦听到信道空闲，开始传输帧。如果网卡侦听到信道正忙，则等待，直到侦听到没有信号能量才开始自己的传输
- 如果网卡传输整个帧没有检测到其他的传输（检测信号能量），网卡就完成了该帧的传输。
- 如果网卡传输帧过程中检测到其他的传输（检测信号能量），网卡会中止当前帧传输
- 中止传输后，网卡等待一个随机时间量，然后返回 b

效率

D_{prop}：信号能量在任意两个适配器之间传播需要的最大时间

D_{trans}：传输一个最大长度的以太网帧的实际（10M以太网，近似为1.2ms）

效率 $$效率 = \frac{1}{1 + 5d_{prop}/d_{trans}}$$

比ALOHA性能更好：简单，成本更低，且是非集中式的

轮流协议

轮流协议（taking-turns Protocol）主要的有两种：**轮询协议（polling protocol）**和 **令牌传递协议（token-passing protocol）**

- 轮询协议：
  - 要求这些结点之一被指定为主结点
  - 主结点以循环的方式 **轮询（poll）** 每个结点
  - 比如主结点先向结点1发送一个报文，告诉他能够传输帧的最大数量；当节点1传输了某些帧后，再告诉结点2他能传输最大帧数量
  - 但是他引入了轮询时延，即告诉结点他可以传输所需的时间；同时如果主结点故障，整个信道就不能操作了
- 令牌传递协议：

第二种轮流协议是**令牌传递协议**（token-passing protocol）。在这种协议中没有主结点。一个称为**令牌**（token）的小的特殊帧在结点之间以某种固定的次序进行交换。例如，结点 1 可能总是把令牌发送给结点 2，结点 2 可能总是把令牌发送给结点 3，而结点 N 可能总是把令牌发送给结点 1。当一个结点收到令牌时，仅当它有一些帧要发送时，它才持有这个令牌；否则，它立即向下一个结点转发该令牌。当一个结点收到令牌时，如果它确实有帧要传输，它发送最大数目的帧数，然后把令牌转发给下一个结点。令牌传递是分散的，并有很高的效率。但是它也有自己的一些问题。例如，一个结点的故障可能会使整个信道崩溃。或者如果一个结点偶然忘记了释放令牌，则必须调用某些恢复步骤使令牌返回到循环中来。经过多年，人们已经开发了许多令牌传递协议，包括光纤分布式数据接口

MAC地址

链路层地址并不是主机或者路由器拥有，而是他们的适配器（网络接口）具有的

链路层地址也可被称为：LAN地址，物理地址

长度为6字节，用16进制表示法表示，比如11-44-AB-98-22-B1

每一块适配器的MAC地址都是独一无二的

MAC广播地址：FF-FF-FF-FF-FF-FF

地址解析协议（Address Resolution Protocol，ARP）

任务是把网络层地址和链路层地址进行转换

他的功能和DNS很像，DNS就是将主机名转化为IP地址，他们之间的区别就在于：DNS可以在因特网的任何地方的主机解析主机名，而ARP只能为在同一个子网上的主机和路由器接口解析IP地址

每台路由器或者主机在其内存里有一张 **ARP表** 这张表包含了IP地址到MAC地址的映射关系，该ARP表也包含了一个寿命（TTL）值，他表示表中删除每个映射的时间。需要注意的是该表不必为该子网的每台主机都包含一个表项