普通高等教育“十五”国家级规划教材

通信网络基础

李建东　盛　敏　编著

高等教育出版社

内容提要

《通信网络基础》是普通高等教育“十五”国家级规划教材，主要介绍通信网络的基本原理。

全书共分7章。第1章主要讨论通信网络的基本构成和协议体系、本书所需的数学基础及通信网络的基本理论问题；第2章详细讨论了链路层、网络层和传输层的端到端传输协议：包括组帧、差错检测、自动请求重发（ARQ）、协议的初始化、差错控制和流量控制等；第3章首先描述了单个排队系统的基本时延性能，接着描述了多个排队队列组成的网络的时延性能，给出的分析模型是常用的网络时延模型；第4章分析了多个用户共享一个信道的问题——多址技术，重点研究随机多址的基本特征（时延、通过量和稳定性）及其改进的方法；第5章研究如何为数据分组选定合适的传输路径问题——路由算法，给出了常用的最短路由算法，并讨论了路由信息的广播；第6章讨论了维持网络正常运行的基本手段——流量和拥塞控制，重点研究了窗口式和漏斗式流量和拥塞控制方法。第7章简要讨论了通信网的拓扑设计。本书文字流畅、内容取材恰当，难度适宜，说理清楚，体现了作者丰富的教学和科研经验，适合教学。

本书可供普通高等学校工科电子信息工程、通信工程、信息工程专业及理科电子信息科学专业本科学生作为教材使用，也可供相关专业硕士研究生、工程技术人员作为参考书使用。

图书在版编目（CIP）数据

通信网络基础／李建东，盛敏编著．—北京：高等教育出版社，2004.8
ISBN 7-04-014568-5

Ⅰ.通… Ⅱ.①李…②盛… Ⅲ.通信网-高等学校-教材 Ⅳ.TN915

中国版本图书馆CIP数据核字（2004）第053909号

策划编辑 刘激扬 责任编辑 刘激扬 封面设计 刘晓翔
责任绘图 朱 静 版式设计 张 岚 责任校对 金 辉
责任印制

出版发行 高等教育出版社 购书热线 010-64054588
社 址 北京市西城区德外大街4号 免费咨询 800-810-0598
邮政编码 100011 网 址 http://www.hep.edu.cn
总 机 010-82028899 http://www.hep.com.cn

经 销 新华书店北京发行所
印 刷

开 本 787×960 1/16 版 次 年 月第1版
印 张 15 印 次 年 月第 次印刷
字 数 270 000 定 价 19.10元

前　言

近年来,通信网络在传统的电话交换网、分组交换网、计算机通信网的基础上得到了飞速发展,出现了多种新型的网络和技术,例如,宽带综合业务网(BISDN)、Internet、帧中继、千兆以太网、第三代移动通信系统(IMT－2000)等等。目前正在向下一代 Internet、全光网络、第 4 代移动通信等方向发展。尽管这些网络在形式上千差万别,但它们许多基本的原理都是相同的。本书的主要目的就是讨论这些网络的共性原理。希望通过本课程的学习,使读者能够理解当今各种新型通信网络的设计原理和依据,同时为读者设计和构思其他新型的通信网络打下理论基础。

本书共分七章。第 1 章主要讨论通信网络的基本构成和协议体系、本书所需的数学基础及通信网络的基本理论问题;第 2 章详细讨论了链路层、网络层和传输层的端到端传输协议:包括组帧、差错检测、自动请求重发(ARQ)、协议的初始化、差错控制和流量控制等;第 3 章首先描述了单个排队系统的基本时延性能,接着描述了多个排队队列组成的网络的时延性能,给出的分析模型是常用的网络时延模型;第 4 章分析了多个用户共享一个信道的问题——多址技术,重点研究随机多址的基本特征(时延、通过量和稳定性)及其改进的方法;第 5 章研究如何为数据分组选定合适的传输路径问题——路由算法,给出了常用的最短路由算法,并讨论了路由信息的广播;第 6 章讨论了维持网络正常运行的基本手段——流量和拥塞控制,重点研究了窗口式和漏斗式流量和拥塞控制方法。第 7 章简要讨论了通信网的拓扑设计。

本书在参照国外"Data Network"(第二版,作者 D. Bertsekas 和 R. Gallager)等多本教材的基础上,根据当前通信网络的最新发展动态,结合作者教案和教学经验、在原研究生教材《信息网络理论基础》基础上修改和撰写而成。其中部分内容曾在西安电子科技大学通信工程、电子信息工程本科班和信息与通信系统研究生班试用。本书可安排 56 个学时(全部内容)或 46 个学时(仅前五章)进行讲授,可根据具体情况调整,书中部分理论性较强的内容可作为选学(如排队网络等)。

本书第 1～第 3 章由李建东同志撰写,第 4～第 7 章由盛敏同志撰写,李建东同志对全书进行了统稿。本书得到了国家自然科学基金(60372048 和 60390540)和教育部优秀青年教师科研教学奖励计划的资助。同时得到了西安电子科技大学综合业务理论及关键技术国家重点实验室和信息科学研究所的

资助。

感谢西安电子科技大学信息科学研究所宽带无线通信实验室的支持，感谢刘增基教授给予的指导和帮助。感谢杨家玮、陈彦辉、徐展琦等同志提供的支持和帮助。特别感谢西安交通大学朱世华教授审阅了全书，并提出了许多宝贵的修改意见。

由于水平有限，书中难免有缺点和错误，敬请读者来信（jdli@mail.xidian.edu.cn 或 msheng@mail.xidian.edu.cn）批评指正。

作 者

2004 年 6 月于西安

目　录

第1章 通信网络概论及数学基础

在通信原理中已清楚地了解了任意两个节点(终端或用户)之间通过一个给定的信道进行信息传输的过程。可以通过适当的信源编码方式来表示信源和降低信源的信息速率,通过适当的信道编码来消除或减轻信道错误的影响,通过适当的调制方式来运载信息,以适应信道的特征。

在现实生活中,如果在任意两个用户之间都建立一条物理传输通道,就可以解决相互通信问题。但共需要 N^2 条物理传输通道(N为通信的用户数)。这种方式的缺点是成本昂贵,且极难扩展,每条物理传输通道的利用率极低。由此可以想像:通信网络的基本问题就是如何以尽可能低的成本有效地解决处于任何地理位置的任意两个用户之间即时信息传递问题。它完全类似于当今的交通运输系统解决人和物资流动的问题。

随着通信网络和计算机网络技术的飞速发展,人们现在不仅可以享用传统的电信业务,如电话、传真等,还可以享用多种信息服务,如电子邮件(E-mail)、网上浏览、信息搜索、电子商务、网上娱乐等。如果希望在任何时间、任何地点都可以享用自己所需的信息服务,就必须要有一个通信网络作支撑。该网络能使用户通过多种传输手段连接到网络之中,并以高速骨干网为基础,实现多种类型网络的互连、互通,为不同要求的用户提供不同速率、不同服务质量、不同类型的信息传输通道。

本章首先讨论通信网络的基本构成,接着讨论网络的分层结构,最后讨论通信网络应解决的基本理论问题。

1.1 通信网络的基本构成

一个基本的通信网络通常由物理传输链路(通道)和链路的汇聚点(网络节点)组成,如图1-1所示。在该网络中,网络节点可以是交换设备也可以是路由器,其主要功能是将多个用户的信息复接到骨干链路上或从骨干链路上分离出用户的信息。通过网络节点的汇聚作用,使得用户可以低成本地共享骨干链路,进而低成本地实现任意用户之间的信息交换。

根据用户类型(移动或固定),业务的种类(电话、计算机数据),传输媒介(有线、无线),节点采用的技术体制(ATM交换体制、电路交换体制、分组转发体制)或作用等可以将已有的通信网络分类,如固定电话网、移动通信网、ATM网络、

局域网等。

图 1－1　一个基本的通信网络示意图

当前实用的覆盖全国乃至全球的通信网中，通常由多种不同类型的网络互连互通而构成。一个可能的网络如图 1－2 所示。它包括的网络（通常称为子网）有：ATM（Asynchronous Transfer Mode，异步转移模式）网络、X.25 分组数据网、PSTN（Public Switched Telephone Network，公用电话交换网）/ISDN（Integrated Service Digital Network，综合业务数字网）、移动通信网/卫星通信网、FDDI（Fiber Distributed Data Interface，光纤分布式数据接口）环网、局域网及高速骨干核心网等。整个网络通过以 WDM（Wavelength Division Multiplexing，波分复用）链路作为核心路由器的高速通道，形成高速信息传输平台，将上述各子网互连互通，可形成一个无缝覆盖的网络。

路由器是网络互连的核心设备，它负责分组（分组是由若干数据比特组成的可进行独立传输的数据块，其长度为几十个字节到几千个字节）的转发和为各个分组选择适当的传输路径。正是由于路由器的存在，任一用户（用户 A）的分组可以通过一个最优的路由（用户 A→路由器 R13→路由器 R1→路由器 R3→ATM 网络→路由器 R14→用户 G）转发给任一目的用户（用户 G）。

在上述网络中，以分组作为载体来运载不同类型的业务。这些业务可以是话音、图像、视频，也可以是电子邮件、Web 业务等等。为了向用户提供不同的服务，除通信网络本身以外，网络中还挂有不同类型的服务器，如 S1、S2、S3 等。因此在通信网中，通信的双方可以是人与人、机器与机器、人与机器等，通信的形式既可以是一个用户对一个用户，也可以是一个用户对多个用户或多个用户对多个用户。

1.1.1 数据传输链路

所谓**数据传输链路**是指在物理传输媒介（如双绞线、同轴电缆、光纤、微波传输系统、卫星传输电路等）上利用一定的传输标准（它通常规定了电气接口、调制解调的方式、数据编码的方式、比特同步、帧格式和复分接的方式等）形成的传输规定速率（和格式）的数据比特通道。

例如：以拨号上网使用的在电话线路上利用 Modem（调制解调器）构成的数据传输链路为例。

它规定了计算机与 Modem 的电器接口为 RS－232C 接口，传输媒介为电话线（双绞线）。在该线路上，利用不同的频带分割和不同的调制方式可以实现 300、1200、2400、4800、9600、19200 b/s 以及更高的传输速率。在 CCITT V.22 标准规定的 1200 b/s 双工调制解调器标准中，将话音信道分为上、下两个子信道，一个子信道占有 900～1500 Hz 的频带，另一个占有 2100～2700 Hz 的频带，已调信号的载频分别为 1200±0.5 Hz 和 2400±1 Hz。调制方式为四相相移键控（QPSK）。主叫（发起通信方）调制解调器在低频段信道发送数据，在高频段信道接收数据，从而实现全双工操作。

数据传输链路分为两大类：一类是用户到网络节点（路由器或交换机）之间的链路（简称为**接入链路**），另一类是网络节点（路由器或交换机）到网络节点（路由器或交换机）之间的链路（简称为**网络链路**）。

接入链路有多种形式，如 Modem（调制解调器）链路、xDSL 链路、ISDN 链路、无线链路（移动通信和卫星通信链路）、局域网链路等。

Modem 链路是利用传统 PSTN 的电话线路，在用户侧和网络侧分别添加 Modem 设备来实现数据传输的，其传输速率可以为 300 b/s 到 56 kb/s。

xDSL 链路是通过数字技术，对 PSTN 端局（本地电话交换机）到用户终端之间的线路进行改造而成的数字用户线 DSL（Digital Subscriber Line）。DSL 的前缀 x 表示不同的传输方案。例如：ADSL（Asymmetric DSL，非对称数字用户线）的上行速率为 640 kb/s～1 Mb/s，下行速率为 6～8 Mb/s；HDSL（High Speed DSL）上下行速率相同，均为 768 kb/s 或 1.5 Mb/s；VDSL（Very High Speed DSL）下行速率 12.96 Mb/s、25 Mb/s、52 Mb/s，上行速率为 1.6～2.3 Mb/s。

ISDN 链路也是利用传统的电话线路实现数据传输，它提供两个基本的信道：B 信道（64 kb/s）和 D 信道（16 kb/s 或 64 kb/s）。供用户使用的基本接口速度为 2B＋D（144 kb/s），一次群速率接口为 23B＋D（1.544 kb/s）或 30B＋D（2.048 kb/s）。

无线链路中最典型的链路就是数字蜂窝移动通信的链路，根据使用的空中接口标准（如：GSM、IS－95，IMT－2000 等）的不同，支持的数据传输速率为十

几 kb/s到 2 Mb/s。对于卫星链路,根据其星座结构(低轨道、中轨道和静止轨道)和卫星数目几百颗(如 Teledesic 系统 288 颗),几十颗(如 Iridium 系统 66 颗)几颗或单颗(Spaceway)的不同,支持的数据传输速率从几 kb/s到几百 kb/s。如果采用固定的无线接入技术(如 LMDS 或近距离的无线接入技术,如无线局域网)其速率可达几 Mb/s至几十 Mb/s。

局域网链路中最典型的是以太网,链路在双绞线上可传输的峰值数据速率为 10 Mb/s、100 Mb/s、1000 Mb/s。这是在办公室环境下,最常用的接入方式。

网络链路也有多种形式,如 帧中继、SDH、WDM 等。

SDH(Synchronous Digital Hierachy,同步数字系统)是在美国贝尔实验室提出的 SONET(Synchronous Optical Network,光同步数字网)的基础上制定的技术标准,它具有一套标准化的结构等级 STM-N(N=1,4,16,64),它们的传输速率分别为 STM-1(155.520 Mb/s),STM-4(622.080 Mb/s),STM-16(2.488.320 Mb/s),STM-64(9.953.280 Mb/s)。

光波分复用(WDM,Wave Length Division Multiplexing)技术是在一根光纤中能同时传输多个波长光信号的一种技术。在发端将不同波长的光信号组合(复用)起来。在接收端又将组合的光信号分开(解复用)并送到不同的终端。目前在一根光纤上可提供的数据速率为 4×2.5 Gb/s(第一个数字 4 为波长数,第二个数字 2.5 为每一波长上的速率,后同)、16×10 Gb/s,160×2.5 Gb/s、128×10 Gb/s 等,理论上可达 5 Tb/s的传输速率。

根据上面的知识,对照图 1-2 可以看出:通信网中的任意两个用户根据他们在网络中所处的位置不同,他们之间的信息传输所经过的传输链路是多种多样的。例如,局域网 B 中的任意两个用户之间是相同特性的链路,而用户 D 与用户 F 之间要经过多种不同特性的链路,即无线链路、SDH 链路、WDM 链路和 X.25 链路等。

1.1.2 数据传输网络

数据传输网络的基本功能是通过网络中的交换机(或路由设备)为运载用户业务的分组,选择合适的传输链路,从而使这些分组迅速可靠地传送到目的用户。(一个分组经过的所有传输链路的集合称为一条路径)。

在数据传输网络中,要传送的基本内容称为消息(message)。根据不同的应用场合,消息可有不同的含义。比如,消息可以是一份电子邮件(E-mail),一份文件,一幅图像等。在要进行交互操作的场合,如:A 可以发一个消息给 B,B 可以发一个应答给 A,双方需要交互多次才可完成信息交换的过程,或者说,双方需要按一定的顺序交换大量的消息。这样一个消息的序列称为一个会话过程(session)。数据传输网络必须保证每一个会话过程可靠、及时、高效地完成。典

型的数据传输网络有分组交换网和 ATM 网等。

1 分组交换网

在分组交换网中，将消息分成许多比较短的、格式化的数据块称为分组（packet）进行传输和交换。每一个分组由若干比特的数据组成。每一个分组通常包括一个附加的分组头。分组头指明该分组的目的节点及其他网络控制信息。在每一个网络节点中采用存储转发的工作方式来将输入的分组送到选定的输出链路上。这种按照一定的规则（路由算法）将输入分组送到选定的输出链路上的过程称为交换。分组交换网如图 1－3 所示。

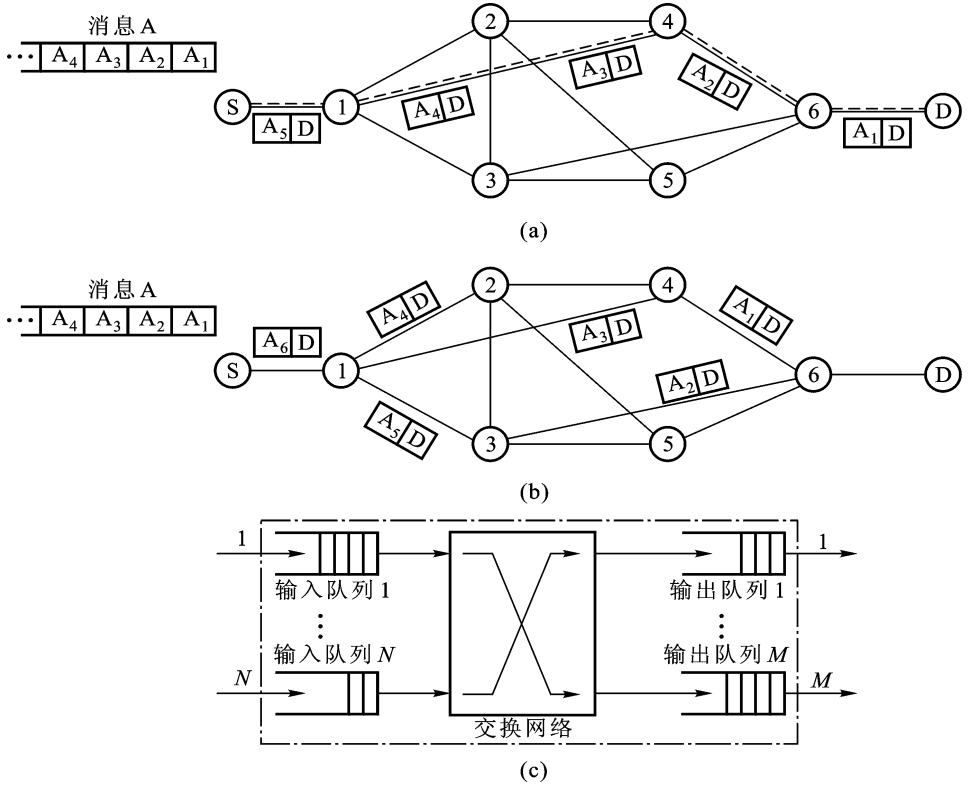


图 1－3 分组交换网

对于图 1－3 所示的分组交换网而言，需要完成三个基本的过程：

(1) 分段和重装的过程。在发端需将一条消息分成规定长度的分组，在收端需要将分组重新装配，恢复原始的消息。

(2) 选择传输路径（确定路由）的过程。在图 1－3 中，节点 S→D 之间可有多种选择，如：路径 A S→1→4→6→D，路径 B S→1→2→4→6→D，路径 C S→1→3→6→D 等。如果路径 A 具有最短的时延和最少的中转次数（或满足其他最佳准则），则选择路径 A。在实际分组传输过程中，有两种最基本的选择路由的

方式：一种是虚电路方式，另一种是数据报方式。在虚电路方式中，在一个会话过程开始时，确定 S→D 的一条逻辑通路（即实际分组传输时才占用物理链路，无分组传输时不占用物理链路。此时物理链路可用于其他用户分组的传输）。会话过程中所有的分组都沿此逻辑通道进行。例如图 1-3（a）的 S→D 之间经过路径 A 建立了一条逻辑通路。每一条逻辑通路中的逻辑链路可用一个虚电路号（VCn）来表示。在数据报方式中，为会话过程中的每一个分组独立地选择路由，也就是 S→D 之间一次会话过程中的分组可以独立地选择路径 A，路径 B 或路径 C 或其他路径。因而，到达目的节点 D 的分组所经过的链路可能各不相同，如图 1-3（b）所示。

（3）各网络节点的交换过程，该交换过程如图 1-3（c）所示。每个交换节点有 *N* 条输入链路和 *M* 条输出链路，每一条输入输出链路可以配置相应的缓冲区，来存储未及时处理的分组。交换网络的作用是根据选定的路由将输入队列的分组送到指定的输出队列中。

2. ATM 网络

ATM（Asynchronous Transfer Mode）是在传统电话网使用的电路交换[1]以及分组交换网基础上发展起来的一种交换技术，可以较好地支持不同速率、不同种类的宽带信息交换。它与分组交换的差别是采用一个全网统一的固定长度的分组（称之为信元）进行传输和交换。ATM 网络中，信元的长度为 53 个字节，其中 5 个字节为信元头，48 个字节用来运载信息。由于信元长度和格式固定，可用硬件电路对信元进行处理，因而缩短了每一个信元的处理时间。它采用面向连接（即虚电路）方式，以提高信息传送的实时性。ATM 设计是以光纤传输为基础，因此在传输链路上采用了非常简单的差错控制和流量控制等措施，提高了信元在网络中的传输速率。

ATM 网络的接口、信元格式和信元交换的过程如图 1-4 所示。

ATM 用户（终端）到 ATM 交换机（网络）之间的接口称为 UNI（User Network Interface，用户网络接口）。交换机（网络节点）之间的接口称为 NNI（Network Node Interface，网络节点接口）。接口 UNI 和 NNI 如图 1-4（a）所示。

ATM 的信元格式如图 1-4（b）所示。UNI 接口和 NNI 接口的信元格式除前 12 个比特格式不同外，其余格式完全相同。GFC（Generic Flow Control）是流

---

[1] 电路交换（或称线路交换）是指根据用户的呼叫请求，将输入物理电路直接与输出物理电路相连接的一种交换技术。电路交换机在功能上等效为一个开关矩阵，当某一输入电路要与某一输出电路相连时，则将对应的开关闭合，形成直接的通路。在电路交换网中，在双方通信之前，需要在双方建立一条直接的物理通路，在通信结束后，要拆除该物理通路。在通信过程中，收发双方独占该道物理通路。在电路交换方式中，信息的传输具有很短的时延，且可以保持收发双方严格的同步关系。但与分组交换相比，链路使用效率相对较低。

量控制比特。VPI/VCI用来表示信元传递的路径。其中,VPI(Virtual Path Identifier)是虚通道标识;VCI(Virtual Channel Identifier)是虚信道标识。从发端到收端的一条虚电路由若干段独立的VPI/VCI链路组成。例如,图1-4(c)中从A到B的虚电路(A→交换机X→交换机Y→交换机Z→B)是由A→X之间的(VPI=8,VCI=17),X→Y之间的(VPI=6,VCI=35),Y→Z之间的(VPI=9,VCI=26)和Z→B之间的(VPI=2,VCI=55)等逻辑链路组成。每个交换机要根据信元应经过的虚电路,完成输入VPI/VCI到输出VPI/VCI的转换。PT(Payload Type,负荷类型)用来区分该信元是用户信息还是控制信息。

CLP(Cell Loss Priority,信元丢失优先级)指示信元的丢失优先级。它用来表示当网络发生拥塞时如何处理该信元。例如,CLP=1,表示信元丢失优先级高,则网络负荷很重时,可以丢弃该信元,以减缓网络可能出现的阻塞。HEC(Header Error Control,信元头差错控制)提供信元头四个字节的差错控制,可进行多个比特的检错和单个比特的纠错。

ATM信元通常是在SDH链路上进行传输的。为了支持不同类型的业务,ATM网络向用户提供四种类别的服务,即A类、B类、C类和D类。服务类别是根据业务的比特率是固定的还是可变的、源节点到目的节点是否需要同步以及是面向连接还是无连接来划分的。对于不同类别的业务,ATM网络采用不同的适配方法(即如何将业务比特流分段,形成数据协议单元(CS-PDU),再如何将CS-PDU分成信元,最后如何有效地传输这些信元)。这些适配的方法称为AAL1~AAL5。AAL1支持A类:恒定比特流、收发需要定时关系、面向连接的业务。AAL2支持B类:可变比特流、收发需要定时关系且面向连接的业务(如话音和图像等业务)。AAL3/4支持C类/D类业务,服务既可以是面向连接的也可以是无连接的,支持的业务可以是报文模式也可以是流模式。AAL5为面向连接的应用提供更为有效的运载措施。

以上仅仅列举了两个典型的数据传输网络,实际应用的网络还有多种多样,如:X.25分组交换网,ISDN快速中继网(帧中继和信元中继)、移动GPRS系统,以路由器+光纤传输系统构成的高速信息网,1000 Mb/s或1 Gb/s的以太网等。

1.1.3 网络的互连

前面两小节主要讨论的是一个子网内的问题,这时所有的链路具有相同特性,采用某种数据传输链路协议和寻址方式,通过交换设备来实现子网内的路由选择和信息交换。

当多个子网要互联互通构成一个大的网络时,需要采用路由器(设备)。如图1-5所示。路由器的基本功能有两个:一是根据路由表,将报文(datagram)发送到正确的目的地;二是维持和更新决定报文传输路径的路由表。路由表中

指明到达目的地应将报文送入与路由器相连的哪一个子网。路由器工作方式是：从接收报文中提取目的地址，并确定该地址中的网络号，查找路由表以获得与该目标网络相匹配的表项。该表项包括报文应到达的下一网络及到达下一网络的必要信息（如对应的路由器输出端口）。报文被封装在选定的输出端口的数据帧中，并由输出端口输出。

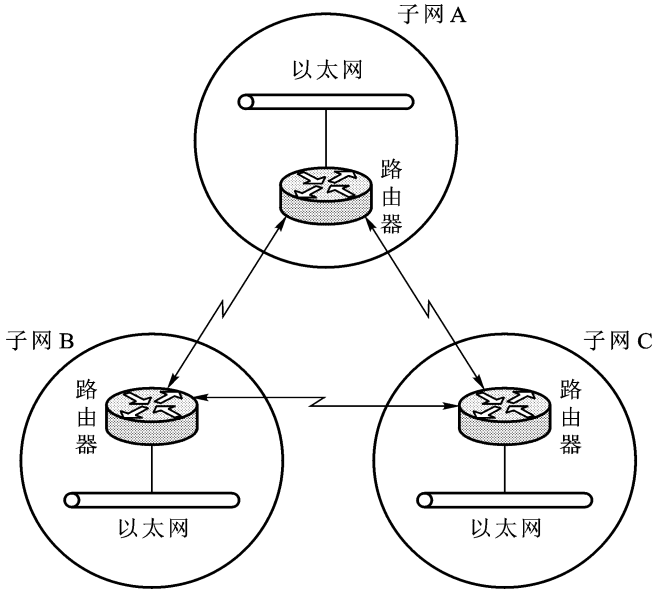


图1-5 网络互连示意图

路由器区别于交换机的关键特征是它可连接使用不同物理传输媒介、具有不同传输协议的数据链路。在一个典型的网络中，通常会有一种以上的局域网（LAN）和广域网（WAN）技术，而每个子网都有独立的数据链路传输协议和寻址方式。

为了实现全网互连需要两个基本条件：一是全网统一编址，二是路由算法。编址解决如何区分网络中的节点、用户终端等，例如，在Internet中是采用IP地址来区分路由路、服务器、用户计算机等。路由算法解决从源到目的地之间应经过的子网、路由器、网络节点等。例如，可以采用从源到目的地经过的路由器最少的原则来选择一条路由。

1.2 协议体系及分层的概念

上一节已经给出了多种可能的通信链路，仅仅有通信链路双方还不能进行有效的通信，还需要一系列的通信规则或协议方可进行通信。这里以一个通俗

的例子来说明通信协议的重要性。

假设有三支部队，红军1队、红军2队和蓝军部队。红军的两支部队被蓝军隔开，如图1-6所示。

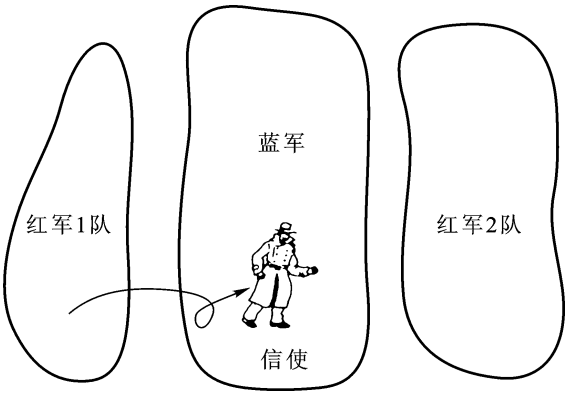


图1-6 红军和蓝军部队的部署情况

如果两个红军部队同时攻击蓝军，则红军胜；否则蓝军胜。两个红军之间通信的惟一手段就是信使（通信员），但信使必须要通过蓝军阵地。任一信使都有可能被蓝军抓获，导致信息丢失，这相当于通信链路不可靠。

红军为了取胜，他们想要两个部队同时进攻。但每一部队必须得到对方也想进攻的确认后，才会进行，否则任一方都不愿意进攻。下面来看能否设计一种协议确保双方同时进入进攻状态？

假设红军一方要通过信使向对方送一个信息“让我们在星期六晚上8:00同时进攻，如果同意，请回复”，并期待对方的回复。对方收到该消息后，会发回一条消息“我们同意。如果你们收到此信息，请回复”，并等待对方的确认。不难看出，如此往复下去将引起无穷多次信息的交换，也不可能使双方同时进入进攻的状态。这个问题出现的关键是：每一方很难相信自己是正确的，它要求双方的信息都必须严格正确。

如果把前面严格确认的条件放松，即要求同时进攻的概率很高，这样上面的问题就可以解决。解决的方法是：如果红军一方要在某个时间发起进攻，它就同时派出多个信使，并确信对方会以很大的概率获得该信息，而对方确信请求进攻方会发起进攻。这样双方取胜的可能性很大。

上述例子说明了通信协议（规则）的重要性，完善的通信协议应当保证通信的终端能高效地向用户提供所需的服务。通信协议通常可通过完善的协议体系来描述。为了描述协议体系，这里首先给出分层的概念。

1.2.1 分层的概念

通信网络的协议可按照分层的概念来设计。分层概念的基础是“模块”的概念。例如:在计算机系统中,一个模块就是一个过程或一台设备,它完成一个给定的功能;若干个模块组成一个完整的系统功能。模块提供的功能通常称之为“服务”。

采用模块概念的好处是:设计简单、可懂性好、标准化、互换性好,有大量现存的模块可以利用。对于模块设计人员,要关心该模块内部的细节和模块的操作。而对于模块使用人员,把模块当作一个黑盒子,只关心该模块的输入、输出以及输入-输出的功能关系,而不关心模块内部的工作细节。模块可以嵌套组成更大的模块,如图 1-7 所示。在该图中一个高层的模块由低层模块加上一些简单模块组成。

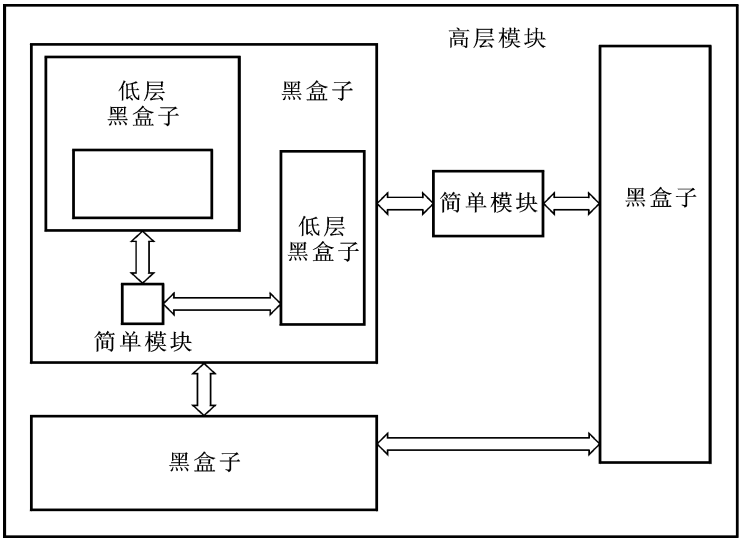


图 1-7 模块组成

通信网络的分层可以看成由一套模块组成的体系结构,除了最低层由物理通信链路组成以外,每一个高层模块是由低层黑盒子通信系统加一组简单的模块组成,如图 1-8 所示。

由于信息的交换必须在双方进行,通信的双方必须有相同(或相应)的功能块才能完成给定的功能,因此在每一层双方两个功能相对应的模块就称为对等(peer)模块或对等过程。如图 1-8 中的模块 H 和 H′,模块 L 和 L′都是对等模块。在该图中,低层模块(通信系统黑盒子)本身由更低层的对等模块和更低层的通信系统黑盒子组成。

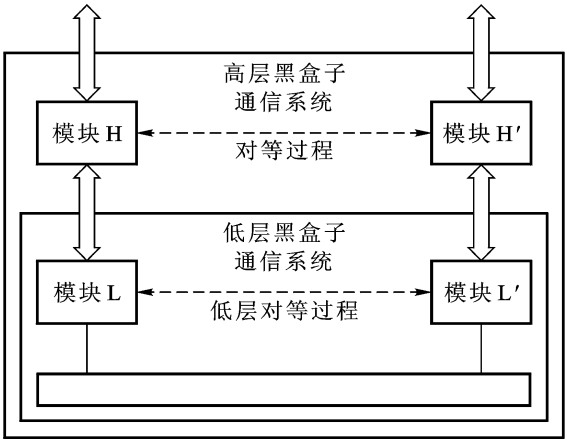


图 1-8 通信系统中模块组成

假设讨论的是第 n 层,那么一个节点中第 n 层对等模块与对方节点中第 n 层对等模块通过第 $n-1$ 层进行通信时,有两个非常重要的方面。第一方面是:需要有一个分布式算法 (或称为协议) 来供两个对等层相互交换消息,以便为高层提供所需的功能和业务。第二方面是第 n 层和第 $n-1$ 层之间的接口 (API),该接口对于实际系统的设计和标准化非常重要。

1.2.2 OSI协议的体系结构

国际标准化组织 (ISO) 将协议体系结构模型分为七个层次:应用层、表示层、会话层、运输层、网络层、数据链路层和物理层,并将它作为开发协议标准的框架。该模型被称为开放系统互连 (OSI) 参考模型,如图 1-9 所示。

各层的主要功能如下:

第一层:物理层 (physical layer)。在由物理通信信道连接的任一对节点之间,提供一个传送比特流 (比特序列) 的虚拟比特管道。在发端它将从高层接收的比特流变成适合于物理信道传输的信号,在收端再将该信号恢复成所传输的比特流。物理信道包括:双绞线、同轴电缆、光缆、无线电信道等。

第二层:数据链路层 (data link layer)。物理层提供的仅仅是原始的数字比特流传送服务,它并不进行差错保护。而数据链路层负责数据块 (帧) 的传送,并进行必要的同步控制、差错控制和流量控制。由于有了第二层的服务,它的上层可以认为链路上的传输是无差错的。

第三层:网络层 (network layer) 网络层的基本功能是把网络中的节点和数据链路有效地组织起来,为终端系统提供透明的传输通路 (路径)。网络层通常分为两个子层:网内子层和网际子层。网内子层解决子网内分组的路由、寻址和传输问题;网际子层解决分组跨越不同子网的路由选择、寻址和传输问题。它还

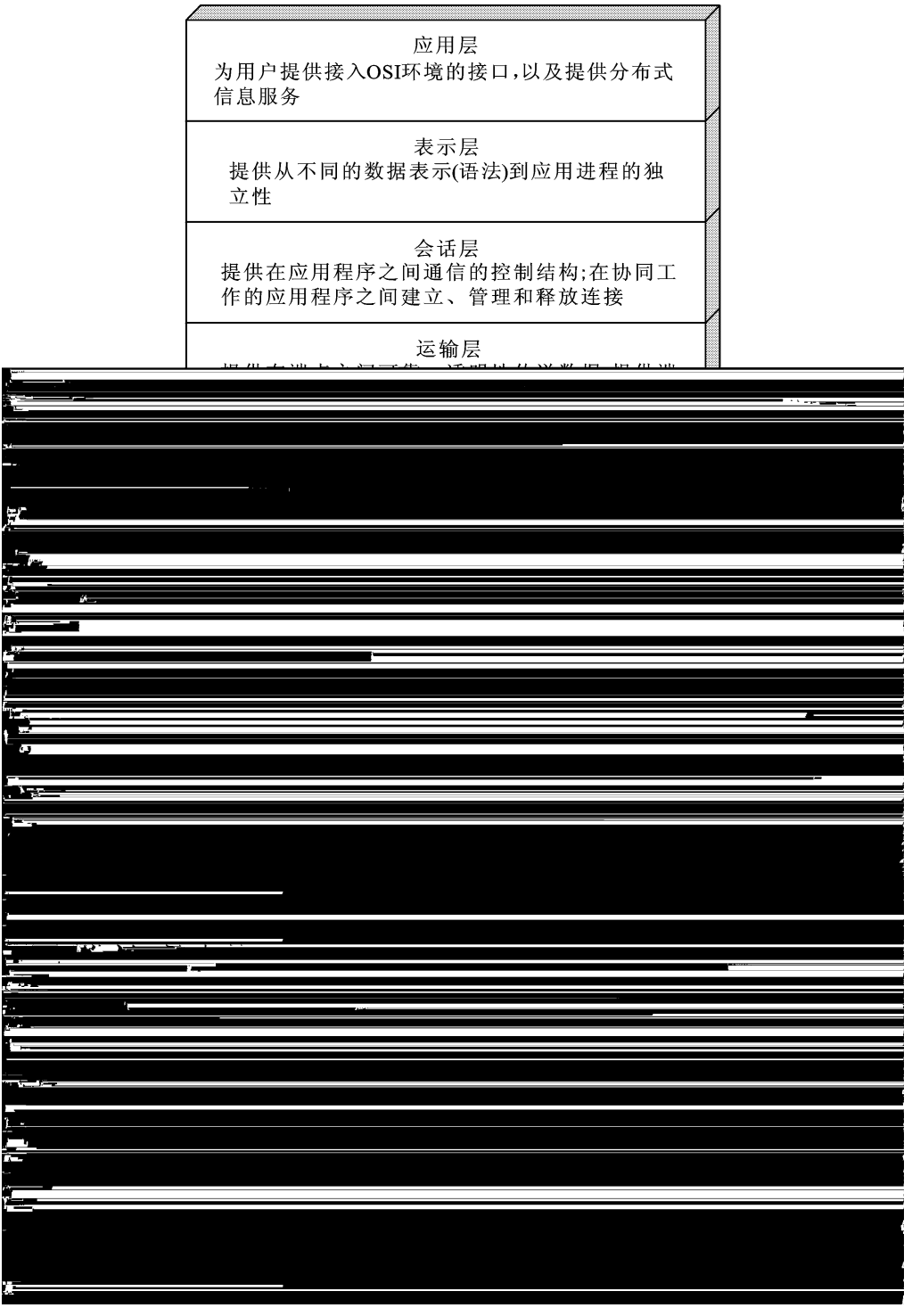

应用层
为用户提供接入OSI环境的接口,以及提供分布式信息服务
表示层
提供从不同的数据表示(语法)到应用进程的独立性
会话层
提供在应用程序之间通信的控制结构;在协同工作的应用程序之间建立、管理和释放连接
运输层

图1-9 OSI参考模型

包括不同子网之间速率匹配、流量控制、不同长度分组的适配、连接的建立、保持和终止等问题。

第四层：运输层（transport layer）。运输层可以看成是用户和网络之间的“联络员”。它利用低三层所提供的网络服务向高层提供可靠的端到端的透明数据传送。它根据发端和终端的地址定义一个跨过多个网络的逻辑连接（而不是第三层所处理的物理连接），并完成端到端（而不是第二层所处理的一段数据链路）的差错纠正和流量控制功能。它使得两个终端系统之间传送的数据单元无差错，无丢失或重复，无次序颠倒。

当高层需要高的吞吐量时，运输层可以产生多个网络连接。当产生和维护一个网络连接很贵时，运输层可以将多个运输层的连接复接到一个网络连接中。当用户终端系统运行多个 session 时，运输层需要建立多个连接，并正确区分不同连接中不同 session 的消息。

第五层：会话层（session layer）。会话层负责控制两个系统的表示层（第六层）实体之间的对话。它的基本功能是向两个表示层实体提供建立和使用连接的方法，而这种表示层之间的连接就叫做“会话”（session）。除此之外，会话层还可以提供一些其他服务，例如提供不同的对话类型（两个方向同时进行，两个方向交替进行，或单方向进行等），以及遇到故障时的对话恢复（同步）。（利用在对话中插入一系列检查点，一旦故障发生，会话层可以从故障发生前的一个检查点开始，重新传送所有数据）。即会话层除向高层提供连接外，还考虑了对话的规则和连续性。例如：一个 session 可允许一个用户登录到一个远端的分时系统，或允许两个机器之间进行文件传输。

第六层：表示层（presentation layer）。表示层负责定义信息的表示方法，并向应用程序和终端处理程序提供一系列的数据转换服务，以使两个系统用共同的语言来进行通信。表示层的典型服务有：数据翻译（信息编码、加密和字符集的翻译），格式化（数据格式的修改及文本压缩）和语法选择（语法的定义及不同语言之间的翻译）等。

第七层：应用层（application layer）。应用层是最高的一层，直接向用户（即应用进程 AP）提供服务，它为用户进入 OSI 环境提供了一个窗口。应用层包含了管理功能，同时也提供一些公共的应用程序，如文件传送，作业传送和控制，事务处理，网络管理等等。它包括了其他 6 层未包括的功能。应用层与其他层的差别是：应用层仅完成自己的功能（特定应用规定的特定任务），而其他各层完成的是满足许多不同应用要求的总任务中一部分。

以上七层功能又可按其特点分为两类，即低层功能和高层功能。低层功能包括了第一至第三层的全部功能，其目的是保证系统之间跨越网络的可靠信息传送；高层功能指第四至第七层的功能，是一些面向应用的信息处理和通信

功能。

在 OSI 七层模型中，许多层具有类似的功能，因此在实际系统中，并不一定需要所有七层的功能，可以根据需要仅有其中部分层次，以提高网络效率。

1.2.3 TCP/IP 协议的体系结构

TCP/IP (Transmission Control Protocol/Internet Protocol) 协议族是在美国国防远景研究规划局 (DARPA) 所资助的实验性分组交换网络 ARPARNET 上研究开发成功的。TCP/IP 协议族的通信任务组织成五个相对独立的层次：应用层、运输层、互连网层、网络接入层、物理层。（它没有 OSI 七层模型中的表示层和会话层）。

TCP/IP 协议族重点强调应用层、运输层和互连网层，而对网络接入层只要求能够使用某种协议来传送互连网层的分组。

TCP/IP 协议族与 OSI 七层模型的对应关系如图 1－10 所示。

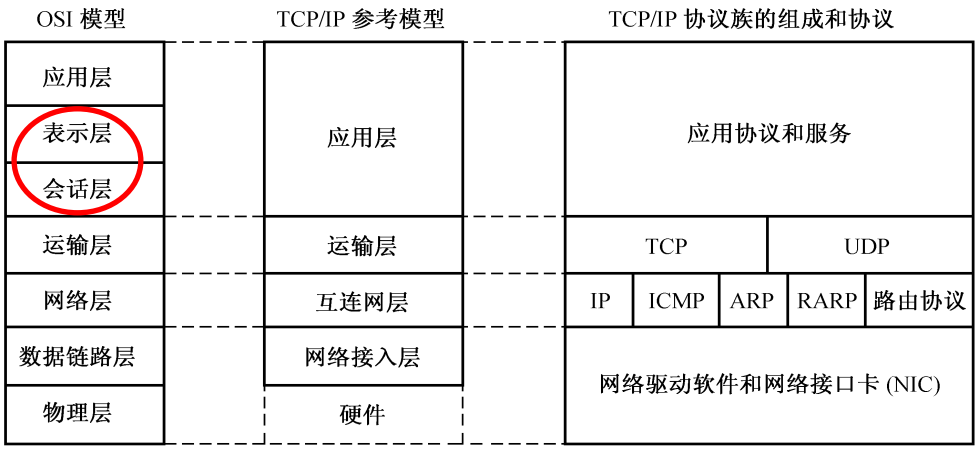


图 1－10 TCP/IP 协议族与 OSI 七层模型的对应关系

网络接入层的主要功能是解决与硬件相关的功能，向互连网层提供标准接口。从网络的角度来讲，它是解决在一个网络中两个端系统之间传送数据的问题，以及一个端系统（计算机）和它连接的网络之间的数据交换。发端计算机必须给网络提供目的计算机的地址，以便网络将数据传到相应的目的地。发端可以充分利用网络提供的服务。

如果两台设备连在两个不同的网络上，要使数据穿过多个互连的网络正确地传输，这是互连网层（网际层）要完成的功能。该层采用的协议称为互连网协议（IP），它提供跨越多个网络的选路功能和中继功能。IP 解决了网络互连问题，但它是一个不可靠的传输协议。在传输过程中可能会出现 IP 报文的错误、

丢失和乱序等问题。与 IP 一起工作的还有 ICMP (Internet Control Message Protocol)、ARP (Address Resolution Protocol)、RARP (Reverse ARP) 等协议。(如图1-10 所示。)

TCP/IP 协议族中的运输层 (也称为主机到主机层) 采用两种不同的协议：TCP 和 UDP。TCP 为应用程序之间的数据传输提供可靠连接,它是面向连接的传输控制协议。UDP (User Datagram Protocol) 为应用层提供无连接的尽力(best effort) 服务,它并不保证一定传到,也不保证按顺序传输以及不重复传送。

TCP/IP 协议族中应用层的主要协议有：文件传送协议 (FTP, File Transfer Protocol)、简单邮件传送协议 (SMTP, Simple Mail Transfer Protocol)、远程登录协议 (TELNET),域名服务 (DNS, Domain Name Service)、网络新闻传送协议 (NNTP, Network News Transfer Protocol)、超文本传送协议 (HTTP, Hyper Text Transfer Protocol)、简单网络管理协议 (SNMP, Simple Network Management Protocol) 等。

1.2.4 混合的分层协议体系

由于现代通信网络的低层基本都是参照 OSI 的模型设计的,而 TCP/IP 协议随着 Internet 的飞速发展而被广泛采用,因而通常采用混合的分层协议体系来描述一个信息网络,如图 1-11 所示。它包括应用层、运输层、网络层、数据链路层和物理层。

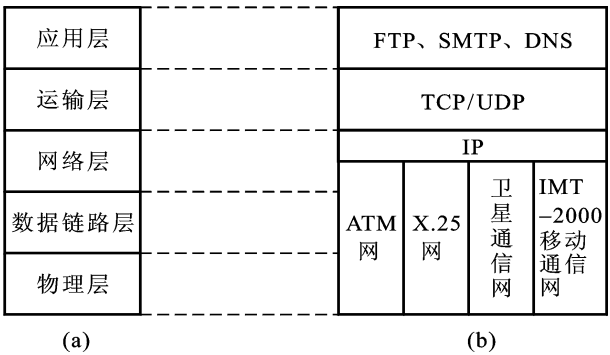


图 1-11 混合分层协议

1.3 通信网络中的数学基础

为了定量地描述通信网络的运行过程、设计通信网络的体系结构和评估通信网络容量、时延和服务质量等,需要了解网络中每个链路、节点、交换机/路由

器，用户终端等设备的输入、输出业务流的行为特征和处理过程。描述这些行为特征和处理过程的基本数学基础是随机过程和排队论，描述网络结构的基本方法是图论。本节主要讨论常用的随机过程和图论基础，在第三章中将详细讨论排队论的基本内容。

1.3.1 随机过程的基本概念

随机过程是随机变量概念在时间域上的延伸。直观地讲，随机过程是时间 t 的函数的集合，在任一个观察时刻，随机过程的取值是一个随机变量。或者说，依赖于时间参数 t 的随机变量所构成的总体称为随机过程。

随机过程是用来描述在一个观察区间内某一实体的随机行为。例如在通信系统中的噪声就是一个典型的随机过程。有很多方法可以获取该随机过程的观察值或样本函数。（在某一时间区间内观察到的该过程取值的一个时间函数称为随机过程的一个样本函数。）如可以用若干台相同的记录仪同时记录通信系统中噪声值的大小或用同一记录仪在不同的观察区间重复记录通信系统中的噪声大小。（不同观察方法的准确性，取决于随机过程本身的特性）。通过获取足够多的样本函数，就可以得到随机过程的统计特性。

设 $X(t)$ 是一个随机过程，则可以从两个方面来描述 $X(t)$ 的特征：一是在任意时刻 t_1，随机变量 $X(t_1)$ 的统计特征，如一维分布函数、概率密度函数、均值和方差等；二是同一随机过程在不同时刻 t_1 和 t_2 对应的随机变量 $X(t_1)$ 和 $X(t_2)$ 的相关特性，如多维联合分布函数、相关函数、协方差矩阵等。

随机过程 $X(t)$ 的一维分布函数定义为

$$F_t(x) = P\{X(t) < x\} \tag{1-1}$$

式中 $P\{\}$ 表示概率。

如果 $F_t(x)$ 对 x 的微分存在，则 $X(t)$ 的一维概率密度定义为

$$f_t(x) = \frac{F_t(x)}{x} \tag{1-2}$$

通常一维分布函数不能完全描述随机过程的特征，需要采用 n 维联合分布函数。对于给定的 n 个时刻 $t_1, t_2, \cdots, t_n$，随机变量 $X(t_1), X(t_2), \cdots, X(t_n)$ 的联合分布函数为

$$F_{t_1, t_2, \cdots, t_n}(x_1, x_2, \cdots, x_n) = P\{X(t_1) < x_1, X(t_2) < x_2, \cdots, X(t_n) < x_n\} \tag{1-3}$$

若 $\int_{-\infty}^{+\infty} |X| \mathrm{d}F_t(x) < +\infty$，则随机过程 $X(t)$ 的均值函数为

$$m_X(t) = E[X(t)] = \int_{-\infty}^{+\infty} x \mathrm{d}F_t(x) \tag{1-4}$$

对任给的时刻 t_1 和 t_2，若下列函数

$$C_X(t_1,t_2)=\operatorname{cov}[X(t_1),X(t_2)]=E[(X(t_1)-m_X(t_1))(X(t_2)-m_X(t_2))] \tag{1-5}$$

存在，则称 $C_X(t_1,t_2)$ 为 $X(t)$ 的**协方差函数**

$$D_X(t)=D[X(t)]=E[(X(t)-m_X(t))^2] \tag{1-6}$$

为 $X(t)$ 的**方差函数**。

若对任意给定的时间 t_1 和 t_2，$R_X(t_1,t_2)=E[X(t_1)X(t_2)]$ 存在，则 $R_X(t_1,t_2)$ 为 $X(t)$ 的**自相关函数**。

协方差函数、自相关函数、均值函数有下列关系

$$C_X(t_1,t_2)=R_X(t_1,t_2)-m_X(t_1)m_X(t_2) \tag{1-7}$$

下面介绍几类典型的随机过程。

1. 独立随机过程

设有一个随机过程 $X(t)$，如果对任意给定的时刻 $t_1,t_2,\cdots,t_n$，随机变量 $X(t_1),X(t_2),\cdots,X(t_n)$ 是相互独立的，也就是其 n 维分布函数可以表示为

$$F_{t_1,t_2,\cdots,t_n}(x_1,x_2,\cdots,x_n)=\prod_{i=1}^{n}F_{t_i}(x_i) \tag{1-8}$$

则称 $X(t)$ 是独立随机过程。该过程的特点是任意一时刻的状态与其他任何时刻的状态无关。

2. 马尔可夫 (Markov) 过程

设有一个随机过程 $X(t)$，如果对于一个任意的时间序列 $t_1<t_2<\cdots<t_n$，$n\geq 3$，在给定随机变量 $X(t_1)=x_1$，$X(t_2)=x_2$，$\cdots$，$X(t_{n-1})=x_{n-1}$ 的条件下，$X(t_n)=x_n$ 的分布可以表示为

$$F_{t_n,t_1,t_2,\cdots,t_{n-1}}(x_n\mid x_1,x_2,\cdots,x_{n-1})=F_{t_n,t_{n-1}}(x_n\mid x_{n-1}) \tag{1-9}$$

则称 $X(t)$ 为马尔可夫过程或简称为马氏过程。该过程的基本特点是无后效性。即当该过程在 t_0 时刻的状态为已知的条件下，则该过程在 $t(>t_0)$ 所处的状态与该过程在 t_0 时刻之前的状态无关。

式 (1-9) 右端的条件分布函数可以写成

$$F_{t,t'}(x\mid x')=P\{X(t)<x\mid X(t')=x'\},\qquad t>t' \tag{1-10}$$

该式称为马氏过程的转移概率。

3. 独立增量过程

设 $X(t_2)-X(t_1)=X(t_1,t_2)$ 是随机过程 $X(t)$ 在时间间隔 $[t_1,t_2)$ 上的增量，如果对于时间 t 的任意 n 个值 $0\leq t_1<t_2<\cdots<t_n$，增量 $X(t_1,t_2)$，$X(t_2,t_3)$，$\cdots$，$X(t_{n-1},t_n)$ 是相互独立的，则称 $X(t)$ 为独立增量过程。该过程的特点是：在任一时间间隔上过程状态的改变并不影响未来任一时间间隔上状态的改

变。可以证明独立增量过程是一种特殊的马尔可夫过程。

4. 平隐随机过程

如果对于时间 t 的任意 n 个值 $t_1, t_2, \cdots, t_n$ 和任意实数 ε，随机过程 X(t) 的 n 维分布函数满足关系式

$$F_{t_1,t_2,\cdots,t_n}(x_1, x_2, \cdots, x_n) = F_{t_1+\varepsilon,t_2+\varepsilon,\cdots,t_n+\varepsilon}(x_1, x_2, \cdots, x_n) \quad (1-11)$$

则称 X(t) 为平稳随机过程或简称为平稳过程。该过程的特点是随机过程的统计特性不随时间的平移而变化。

按照上述定义的随机过程通常称为严（狭义）平稳过程。而在实际应用中更关心这样一类过程：其 $E[|X(t)|^2] < \infty$（称为二阶矩过程），且满足下列条件：(1) 均值为常量（与时间 t 无关）；(2) 对于任意时刻 s 和 t，其相关函数满足 $R_X(s,t) = R_X(t-s)$，即相关函数仅与时差 $t-s$ 有关，而与 s，t 的取值无关，称这类过程为宽（广义）平稳过程。在实际应用中所指的随机过程通常是宽平稳过程。

对于平稳过程中一个重要特征就是是否具有各态历经性。为了说明各态历经性，在时间轴上定义下列两种平均：

$$\langle X(t) \rangle = \lim_{T\to+\infty} \frac{1}{2T}\int_{-T}^{+T} X(t)\,dt \quad (1-12)$$

$$\langle X(t)\,X(t+\tau) \rangle = \lim_{T\to+\infty} \frac{1}{2T}\int_{-T}^{+T} X(t)\,X(t+\tau)\,dt \quad (1-13)$$

为随机过程 X(t) 的时间均值和时间相关函数。

如果 X(t) 是一个平稳过程，如果 $\langle X(t) \rangle = E[X(t)] = m_X$ 依概率 1 成立（即对所有样本都成立），则称随机过程 X(t) 的均值具有各态历经性；如果 $\langle X(t)\,X(t+\tau) \rangle = E[X(t)\,X(t+\tau)] = R_X(\tau)$ 依概率 1 成立，则称过程 X(t) 的自相关函数具有各态历经性。如果 X(t) 的均值和自相关函数都具有各态历经性，则称 X(t) 是（宽）各态历经过程，或者说 X(t) 是各态历经的。

1.3.2 Poisson 过程

在日常生活中，如果观察顾客进入商店、银行或其他公共服务场所的过程，会发现若把一位顾客的到达看成一个“随机点”，则这是一个源源不断出现随机点的过程。在这一过程中任一段时间内到达的顾客数也是随机的。这类描述到达顾客数及其特征的过程通常称为计数过程。如果考察一个交换局中电话呼叫到达（人们拨打电话的行为中拿起电话并拨出对方号码的动作称为一次电话呼叫到达）的过程也具有类似的特征。

设一个随机过程为 $\{A(t), t \geq 0\}$，A(t) 的取值为非负整数，如果该过程满足下列条件，则称该过程为到达率为 λ 的 Poisson 过程：

(1) $A(t)$ 是一个计数过程，它表示在 $[0,t)$ 区间内到达的用户总数，$A(0)=0$，$A(t)$ 的状态空间为 $\{0,1,2,\cdots\}$。如图 1-12 所示。任给两个时刻 s 和 t，且 $s<t$，则 $A(t)-A(s)$ 即为 $[s,t)$ 之间到达的用户总数。

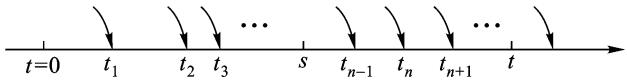


图 1-12 Poisson 到达过程示意图

(2) $A(t)$ 是一个独立增量过程。即在两个不同时间区间（区间不重叠）内到达的用户数是相互独立的。

(3) 任一个长度为 τ 的区间内，到达的用户数服从参数为 $\lambda\tau$ 的 Poisson 分布，即

$$P(A(t+\tau)-A(t)=n)=e^{-\lambda\tau}\frac{(\lambda\tau)^n}{n!} \qquad n=0,1,2,\cdots \tag{1-14}$$

其均值和方差均为 $\lambda\tau$。由于在 τ 区间内平均到达的用户数为 $\lambda\tau$，则 λ 即为单位时间平均到达的用户数或称为到达率。

Poisson 过程的基本特征有：

(1) 到达时间间隔 $\tau_n=t_{n+1}-t_n$ 相互独立，且服从指数分布，其概率密度函数为

$$p(\tau_n)=\lambda e^{-\lambda\tau_n} \tag{1-15}$$

其分布函数为

$$p(\tau_n<s)=1-e^{-\lambda\tau} \qquad s\geq 0 \tag{1-16}$$

该特性说明 Poisson 过程的到达间隔服从指数分布。相反，如果一个计数过程的到达间隔序列是相互独立同分布，其分布是参数为 λ 的指数分布，则该过程是到达率为 λ 的 Poission 过程。因此，说顾客到达过程为到达率为 λ 的 Poisson 过程与说顾客到达间隔相互独立且服从参数为 λ 的指数分布是等价的。

(2) 对于一个任意小的区间 $\delta\geq 0$，将 Poisson 分布用 Taylor 级数展开，即利用

$$e^{-\lambda\delta}=1-\lambda\delta+\frac{(\lambda\delta)^2}{2}-\cdots \tag{1-17}$$

可得

$$P\{A(t+\delta)-A(t)=0\}=1-\lambda\delta+o(\delta) \tag{1-18}$$

$$P\{A(t+\delta)-A(t)=1\}=\lambda\delta+o(\delta) \tag{1-19}$$

$$P\{A(t+\delta)-A(t)\geq 2\}=o(\delta) \tag{1-20}$$

式中，$o(\delta)$ 表示 δ 的高阶无穷小，即 $\lim\limits_{\delta\to 0}\dfrac{o(\delta)}{\delta}=0$。

公式 (1-18) ～ (1-20) 的含义是：在一个充分小的时间间隔内，没有用户到

达的概率为 $1-\lambda\delta$，在一个充分小的时间间隔内，有一个用户到达的概率为 $\lambda\delta$；在一个充分小的时间间隔内有两个或两个以上用户到达是几乎不可能的。

(3) 多个相互独立的 Poisson 过程之和 $A=A_1+A_2+\cdots+A_k$ 仍是一个 Poisson 过程，其到达率为 $\lambda=\lambda_1+\lambda_2+\cdots+\lambda_k$，式中 λ_k 是 Poisson 过程 A_k 的到达率。

(4) 如果将一个 Poisson 过程的到达以概率 p 和 $1-p$ 独立地分配给两个子过程，则这两个子过程也是 Poisson 过程。注意这里是将到达独立地进行分配。如果把到达交替的分配给两个子过程，即两个子过程分别由奇数号到达和偶数号到达组成，则这两个子过程就不是 Poisson 过程。

例 1.1 有红、绿、蓝三种颜色的汽车，分别以强度为 λ_R、λ_G、λ_B 的 Poisson 流到达某哨卡，设它们是相互独立的。把汽车合并成单个输出过程（假设汽车长度为 0）。

(1) 求两辆汽车之间的时间间隔的概率密度函数；

(2) 求在 t_0 时刻观察到一辆红色汽车，下一辆汽车将是：(a) 红的，(b) 蓝的，(c) 非红的概率；

(3) 求在 t_0 时刻观察到一辆红色汽车，下三辆汽车是红的，然后又是一辆非红色汽车将到达的概率。

解 (1) 由于独立的 Poisson 过程之和仍为 Poisson 过程，且其强度为 $\lambda_C=\lambda_R+\lambda_G+\lambda_B$。设 Z_C 为两辆汽车到达的时间间隔，则其概率密度函数为

$$P_{z_C}(z)=\begin{cases}\lambda_C e^{-\lambda_C z}, & z\geqslant 0\\ 0, & z<0\end{cases}$$

(2) 设 Z_R、Z_G、Z_R 分别为两辆红色、绿色、蓝色汽车到达的时间间隔，Z_X 为红色与非红色汽车到达的时间间隔。由于红色车辆的到达与非红色车辆的到达相互独立。因此，Z_X 仅与非红色汽车到达间隔有关。非红色汽车是绿和蓝色 2 种汽车的复合流，因此 Z_X 的概率密度函数为

$$p_{Z_X}(z)=\begin{cases}(\lambda_B+\lambda_G)e^{-(\lambda_B+\lambda_G)z}, & z\geqslant 0\\ 0, & z<0\end{cases}$$

由于 Z_X 与 Z_R 相互独立，故下一辆是红色汽车的概率为

$$P\{\text{下一辆是红色汽车}\}=P\{Z_R<Z_X\}=\int_0^\infty \lambda_R e^{-\lambda_R z_R}\mathrm{d}z_R\int_{z_R}^\infty \lambda_X e^{-\lambda_X z_X}\mathrm{d}z_X$$

$$=\frac{\lambda_R}{\lambda_R+\lambda_X}=\frac{\lambda_R}{\lambda_R+\lambda_G+\lambda_B}$$

令 Z_Y 是从 t_0 算起的非蓝色汽车的到达时刻，则同理可得：

$$P\{\text{下一辆是蓝色汽车}\}=P\{Z_B<Z_Y\}=\frac{\lambda_B}{\lambda_R+\lambda_G+\lambda_B}$$

$$P\{\text{下一辆是非红色汽车}\}=1-\frac{\lambda_R}{\lambda_R+\lambda_G+\lambda_B}=\frac{\lambda_G+\lambda_B}{\lambda_R+\lambda_G+\lambda_B}$$

(3) 来到的是三辆红色汽车然后是一辆非红色汽车同时发生的概率为

$$p=\left(\frac{\lambda_R}{\lambda_R+\lambda_G+\lambda_B}\right)^3\frac{\lambda_G+\lambda_B}{\lambda_R+\lambda_G+\lambda_B}$$

1.3.3 马尔可夫链

马尔可夫(Markov)链是最简单的马氏过程——即时间和状态过程的取值参数都是离散的马氏过程。将可列个发生状态转移(变化)的时刻记为 $t_1,t_2,\cdots,t_n,\cdots$,在 t_n 时刻发生的转移称为第 n 次转移;并且假定在每一个时刻 t_n $(n=1,2,\cdots)$,$X_n=X(t_n)$ 所有可能的状态的集合 S 是可数的,即可表示为 $S=\{0,1,2,\cdots\}$。对应于时间序列 $t_1,t_2,\cdots,t_n,\cdots$,马氏链的状态序列为 $i_1,i_2,\cdots,i_n,\cdots$。这时对应于式(1-9)有

$$P\{X_n=i_n\mid X_{n-1}=i_{n-1},\cdots,X_1=i_1\}=P\{X_n=i_n\mid X_{n-1}=i_{n-1}\}\tag{1-21}$$

该式表示在 $X_{n-1}=i_{n-1}$ 的条件下,第 n 次转移出现 i_n 的概率。如该概率与 n 无关(即与哪一次转移无关,仅与转移前后的状态有关),则该马氏链为齐次马氏链;否则称为非齐次马氏链。本书仅讨论齐次马氏链。此时式(1-21)可以表示为

$$P_{ij}=P\{X_n=j\mid X_{n-1}=i\}\tag{1-22}$$

上式称为马氏链的(一步)转移概率。P_{ij} 满足下列条件:

$$P_{ij}\geq 0,\qquad \sum_{j=0}^{\infty}P_{ij}=1,\qquad i=0,1,\cdots\tag{1-23}$$

相应的转移概率矩阵可以表示为

$$P=\begin{matrix}P_{00} & P_{01} & P_{02} & \cdots\\ P_{10} & P_{11} & P_{12} & \cdots\\ \cdots\cdots & & & \\ P_{i0} & P_{i1} & P_{i2} & \cdots\\ \cdots\cdots & & & \end{matrix}\tag{1-24}$$

例 1.2 设一盲人(或看成一个随机点)在如图 1-13 所示的线段上游走,其步长为 l。假定他只能在 $a_1=2l,a_2=l,a_3=0,a_4=-l,a_5=-2l$ 这5个点上停留,且只在 $t=1,2,\cdots$ 时刻上发生游走。游走的规则是:如果游走前他在 a_2,a_3,a_4 这几个点上,那么就分别以 1/2 的概率向左或向右走动一步;如果游走前他在 a_1 (a_5) 上,那么就以概率 1 走到点 a_2 (a_4) 上。

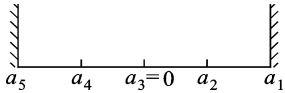


图 1-13 随机游走过程

以 $X_n = a_i$, i=1,2,3,4,5 表示在时刻 $t=n$ 盲人位于停留点 a_i 处，则容易看出 $X_1, X_2, \cdots$ 是一个马氏链，且他游走的转移概率矩阵为

$$P=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

如果用图来表示这一转移过程，则可得图 1−14。图中的圆圈表示马氏过程的状态，图中带箭头的弧线表示状态的转移，弧线上的数字表示一步转移概率。该图称为马氏过程的状态转移图。

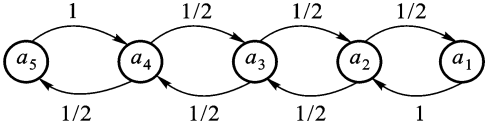


图 1−14 状态转移图

在考察马氏过程中还经常用到 n 步转移概率，即

$$P_{ij}^{n} = P\{X_{n+m} = j \mid X_m = i\} \tag{1-25}$$

它表示当前（第 m 步）的状态为 i，经过 n 步转移后（第 n+m 步）系统的状态为 j 的概率。系统中 $n=m_1+m_2$ 步状态转移概率可用下式来求解

$$P_{ij}^{m_1+m_2} = \sum_{k=0}^{\infty} P_{ik}^{m_1} P_{kj}^{m_2}; \quad n,m \geq 0; \quad i,j \geq 0 \tag{1-26}$$

该公式称为 Chapman−Kolmogorov 等式。

根据 n 步转移概率，就可以来定义马氏链状态转移的特性。

如果马氏链的两个状态 i 和 j 有下列特性：即存在整数 n 和 n′有 $P_{ij}^{n}>0$ 及 $P_{ji}^{n'}>0$（也就是从状态 i（j）经过 n（n′）步转移到状态 j（i）的概率大于 0），则称 *i 和 j 是互通的。如果马氏链的所有状态都是互通的，则该马氏链是不可约的*（irreducible）。

如果马氏链的状态 i 有下列特性：即存在某个整数 m≥1，使 $P_{ii}^{m}>0$ 且存在某个整数 d>1 并仅当 *n 为 d 的整倍时有* $P_{ii}^{n}>0$，则状态 i 是有周期性的。如果马氏链中没有一个状态是有周期性的，则称该马氏链为非周期的。本书仅考虑非周期不可约的马氏链。

上面讨论了状态之间的转移概率，同时还需关心过程处于某一状态的稳态概率。

对于马氏链，其状态的稳态概率定义为：如果有

$$p_j = \sum_{i=0}^{\infty} p_i P_{ij} \qquad j=0,1,\cdots \tag{1-27}$$

则称概率分布 $\{p_j \mid j \geq 0\}$ 是马氏链的稳态分布。对于概率分布有 $\sum_{j=0}^{\infty} p_j = 1$。稳态概率反映了系统达到稳态后，系统处于某一状态的可能性（概率）。

稳态分布可以表示为：

$$p_j = \lim_{n \to \infty} P\{X_n = j \mid X_0 = i\} \qquad i = 0,1,\cdots \tag{1-28}$$

即过程从初始状态 $X_0 = i$ 出发，最终转移到状态 $X_n = j$ 的概率。显然 p_j 与初始状态 $X_0 = i$ 无关。稳态分布也可以表示为：（以概率 1 成立）

$$p_j = \lim_{k \to \infty} \frac{\text{前 } k \text{ 次转移中访问状态 } j \text{ 的次数}}{k} \tag{1-29}$$

因此 p_j 表示该过程中访问状态 j 的时间比例或频率，且该频率与初始状态无关。

由于 $\sum_{i=0}^{\infty} P_{ji} = 1$，则结合式 (1-27) 有

$$p_j \sum_{i=0}^{\infty} P_{ji} = \sum_{i=0}^{\infty} p_i P_{ij} \qquad j = 0,1,\cdots \tag{1-30}$$

该式称为全局平衡方程（Global balance equations）。它表示在稳态情况下，从一个状态 j 转移出去的频率［式 (1-30) 左边］等于转移进入状态 j 的频率（式 (1-30) 的右边）。该方程提供一种典型的求解稳态概率分布的方法。

例 1.2（续） 在例 1.2 中已知各种状态的转移概率矩阵，设状态 $a_5, \cdots, a_1$ 的稳态概率为 p_5, p_4, p_3, p_2 和 p_1，则根据式 (1-27) 可得稳态概率的方程为

$$(p_5, p_4, p_3, p_2, p_1) \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} = (p_5, p_4, p_3, p_2, p_1) \tag{1-31}$$

由于 $\{p_i \mid i = 1,2,\cdots 5\}$ 是稳态概率分布，则有

$$\sum_{i=1}^{5} p_i = 1 \tag{1-32}$$

求解式 (1-31) 和式 (1-32) 组成的方程组，得稳态概率分布为

$$(p_5, p_4, p_3, p_2, p_1) = \left(\frac{1}{8}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}\right) \tag{1-33}$$

式 (1-30) 是对一个状态而言的，它反映的是从一个状态出发和返回到该状态的转移频率。可将它推广到一组状态，即从一组状态出发和返回该组状态的

转移频率。

设 S 是状态空间的一个子集。将式 (1－30) 对 S 中的所有状态 (即对所有 $j\in S$) 相加,得

$$\sum_{j\in S}p_j\sum_{i\overline{|}S}P_{ji}=\sum_{i\overline{|}S}p_i\sum_{j\in S}P_{ij} \tag{1-34}$$

该式表明转出状态子集 S 的频率等于转入状态子集 S 的频率。

在实际中经常遇到这样一类特殊的马氏过程:仅有相邻状态的转移,而没有其他状态的转移 (即如果 $|i-j|>1$,则 $P_{ij}=0$),如图 1－15 所示。这一类过程通常称为生灭 (birth-death) 过程。它表示一个群体 (动物、生物等) 总数为 n 的特殊状态转移过程。该群体总数的状态转移只有三种可能:或者从 n 增加一个 (群体出生一个),或者从 n 减少一个 (死亡一个),或者群体的总数 n 保持不变。而其他所有可能的转移相对前三种转移都是高阶无穷小,因而可以忽略不计。该群体状态的转移概率取决于群体的总数 n (状态)。

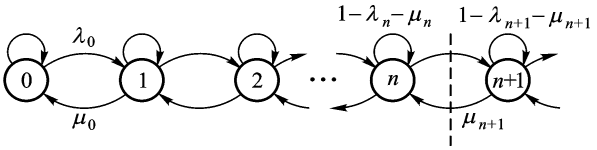


图 1－15 生灭过程状态转移图

令 $S=\{0,1,\cdots,n,n+1,\cdots\}$,则应用式 (1－30) 可得

$$p_nP_{n,n+1}=p_{n+1}P_{n+1,n}\qquad n=0,1,\cdots \tag{1-35}$$

它表示在稳态的情况下,从状态 n 转移到状态 n+1 的频率等于从状态 $n+1$ 转移到状态 n *的频率,或在状态转移图中设置一个虚拟的平面* (图 1－15 中的虚线),则进入该平面的频率等于退出该平面的频率,即该平面是系统的一个稳定点。

式 (1－35) 可推广到一个更一般的形式

$$p_jP_{ji}=p_iP_{ij} \tag{1-36}$$

该等式称为详细平衡方程 (detailed balance equation)。如果对于一个过程,有式 (1－36) 成立,则意味着有式 (1－27) 全局平衡方程存在。但并不是任何马尔可夫链都必须有式 (1－36) 成立。但在很多实际应用中,式 (1－36) 是成立的。因此实际中可以先假设式 (1－36) 成立,然后通过它们求解稳态概率 p_j, $j\geq 0$。如果求得的结果满足 $\sum_j p_j=1$,则求得的分布 $\{p_j, j\geq 0\}$ 就是满足式 (1－27) 的稳态分布。

例 1.3 设一个特殊的生灭过程的状态转移概率为 $P_{n,n+1}=p_R$, $P_{n+1,n}=p_L$,试求其稳态分布。

解 利用式(1-35)[或假设式(1-36)成立]有

$$p_n p_R = p_{n+1} p_L$$

从而有

$$P_{n+1} = p_n \frac{p_R}{p_L} = \rho p_n$$

式中

$$\rho = \frac{p_R}{p_L}$$

经过递推得

$$p_{n+1} = \left(\frac{p_R}{p_L}\right)^{n+1}$$

$$p_n = \rho^{n+1} p_0$$

显然只有在 $\rho = \frac{p_R}{p_L} < 1$ 的情况下,下式才有可能成立

$$\sum_{n=0}^{\infty} \rho^n p_0 = \frac{p_0}{1-P} = 1$$

从而可得 $p_0 = 1-\rho$。*进而可得*

$$p_n = \rho^n (1-\rho)$$

综上可得,在 $p_R < p_L$ 的情况下,该生灭过程的稳态分布为 $\{p_n = \rho^n (1-\rho), n \geq 0\}$。

1.3.4 图论基础

图论是一个新的数学分支,也是一门很有实用价值的学科,它在很多领域都得到了广泛的应用。近年来它受到计算机科学蓬勃发展的影响,发展极其迅速。应用范围也不断拓广。在通信网络中,许多问题的描述都是基于图论的,因此下面对图论的一些基本概念进行讨论。

1. 图的概念

一般几何上将图定义成空间中一些点(顶点)和连接这些点的线(边)的集合。

图论中将图定义为 G=(V,E),其中 V 表示顶点的集合,E 表示边的集合。这样如图 1-16 所示的图可以表示为

$V=\{v_1, v_2, v_3, v_4\}$,$E=\{e_1, e_2, e_3, e_4, e_5, e_6\}$

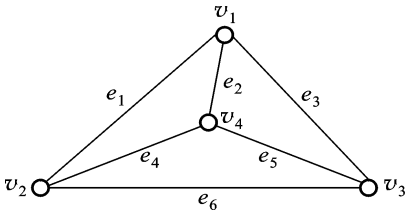


图 1-16 一个简单的无向图

也可以用边的两个顶点来表示边。如果边 e 的两个顶点是 u 和 v,那么 e 可写成 e=(u,v),这里(u,v)表示 u 和 v 的有序对。如果有(u,v)和(v,u)同时存在,它表达了以*u,v为端点的一条无向边,如果图中的所有*边都是无向边,则称该图为无向图。这样也可以将无向图 1-16 表示为:

$$G=(V,E),\ V=\{v_1,v_2,v_3,v_4\},$$

$$E=\{(v_1,v_2),(v_1,v_3),(v_1,v_4),(v_2,v_3),(v_2,v_4),(v_3,v_4)\}$$

或 $E=\{(v_2,v_1),(v_3,v_1),(v_4,v_1),(v_3,v_2),(v_4,v_2),(v_4,v_3)\}$

一般图 $G=(V,E)$ 的顶点数用 $n(=|V|)$ 表示，边的数目用 $m(=|E|)$ 表示。若 $|V|$ 和 $|E|$ 都是有限的，则称图 G 是**有限图**，否则称为**无限图**。本书只讨论有限图的情况。

在实际应用中，图中每条边可能有一个方向是很自然的（它反映了信息或物质的流向）。当给图 G 的每一条边都规定一个方向，则称该图为**有向图**。对有向图 $G=(V,E)$，有向边 e 用与其关联的顶点 (u,v) 的有序对来表示，即 $e=(u,v)$，它表示 u 为边 e 的起点，v 为边 e 的终点。那么图 1－17 所示的有向图可表示如下：

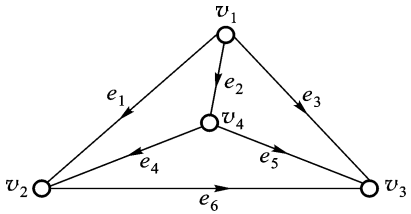


图 1－17 一个简单的有向图

$$G=(V,E),\quad V=\{v_1,v_2,v_3,v_4\},$$

$$E=\{(v_1,v_2),(v_1,v_3),(v_1,v_4),(v_4,v_2),(v_4,v_3),(v_2,v_3)\}$$

如果顶点 v 是边 e 的一个端点，则称为 e 和顶点 v **相关联**（incident）；对于顶点 u 和 v，若 $(u,v)\in E$，*则称 u 和 v 是邻接的*（adjacent）。在图 1－16 中，边 e_2,e_4,e_5 都与顶点 v_4 相关联，v_4 分别与 v_1,v_2,v_3 相邻接。若两条边有共同的顶点，则称这两条边是邻接的。在图 1－16 中，边 e_1,e_2,e_3 两两相邻接。

对图 $G=(V,E)$ 和 $G'=(V',E')$ 来说，若有 $V'\subseteq V$ 和 $E'\subseteq E$，*则称图 G′ 是图 G 的一个*子图；若 $V'\subset V$ 或 $E'\subset E$，*则称图 G′ 是图 G 的一个*真子图。

2. 路径与回路

定义：图 $G=(V,E)$ 的一些顶点和边的交替序列 $\mu=v_0e_1v_1\cdots v_{k-1}e_kv_k$，且边 e_i 的端点为 v_{i-1} 和 v_i，$i=1,2,\cdots,k$，则称 μ 为一条路径（path），v_0 和 v_k 分别为 μ 的起点和终点。如果 μ 中所有的边均不相同，则称其为**简单路径**。以 v_0 为起点，v_k 为终点的路径称为 v_0-v_k 路径。

如果路径 μ 中有 $v_0=v_k$，则 μ 为**回路**（或环，cycle），回路中没有重复边时称为**简单回路**。

对于有向图也有类似定义路径、回路的概念，只不过此时需要考虑方向性。

例 1.4 在图 1－18 中，$S=\{v_1e_2v_2e_3v_3e_6v_4\}$ 是一路径，$C=\{v_1e_2v_2e_3v_3e_6v_4e_4v_1\}$ 是一回路。

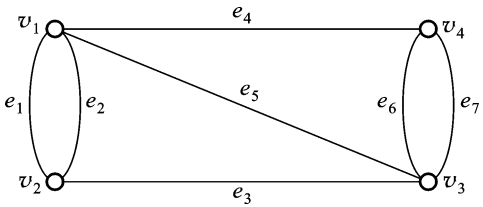


图 1－18 例 4 的图

定义：对图 $G=(V,E)$ 来说，若 G

的两个顶点 u，v 之间存在一条路径，则称 u 和 v 是连通的，若图 G 的任意两个顶点都是连通的，则称图 G 是**连通的**；否则是**非连通的**。非连通的图可分解为若干连通的子图。

图 1－19 的无方向图中，图 (a) 中任意两个顶点之间都有路径，所以该图是连通的；图 (b) 中顶点 3 和图中其他顶点之间没有路径，所以该图是非连通的；图 (c) 则是一个孤立的节点

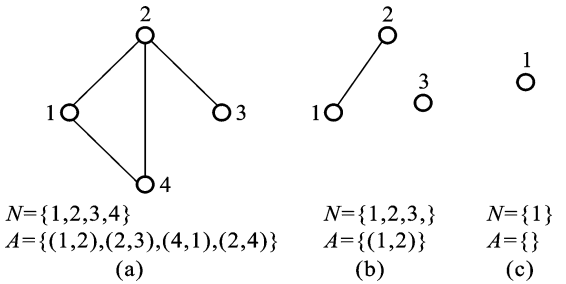


图 1－19 无方向图

(a) 连通图 (b) 非连通图 (c) 孤立节点

对于有向图，若去掉方向后边是连通的，则称该图为连通的**有向图**。若对于有向图的任意两个顶点 u 和 v 之间存在 u 到 v 的路径和 v 到 u 的路径时，称该图为强**连通的**。图 1－20 (a) 的有向图是一个连通的有向图，但不是**强连通的**。因为顶点 2 和顶点 3 之间不存在双向的路径；图 1－20 (b) 是一个强连通的图，该图中任意两个顶点之间都存在双向的路径。

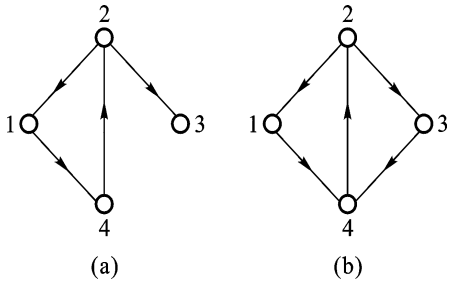


图 1－20 方向图

(a) 连通的方向图 (b) 强连通的方向图

3. 生成树和最小重量生成树

为了介绍生成树和最小重量生成树，首先给出树的概念。

定义：不包括回路（环）的连通图，称为树。

定义：对于图 G＝（V，E），包含了图 G 中所有顶点的树称为生成树（Spanning Tree）。

在图 1－21 中，图 (b)、(c) 和 (d) 都是树。而图 (a) 由于有回路，所以不是树。在图 1－21 中，图 (b) 和图 (c) 都是图 (a) 的生成树。

对于一个给定的图 G＝（V，E），其生成树的构造算法如下：

(1) 令 n 是 V 中的任意一个顶点，构造子图 G′＝（V′，E′），其中 V′＝{n}，E′＝ě（空集）；

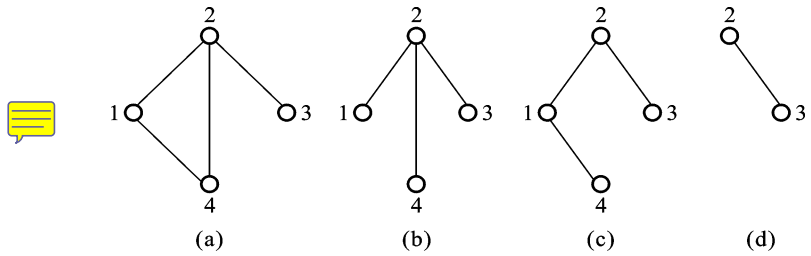


图 1－21 树和生成树

(2) 如果 $V'=V$ *则停止。此时* $G'=(V',E')$ 就是一个生成树。否则进行第 (3) 步；

(3) 令 $(i,j)\in E$，*其中* $i\in V'$，$j\in V-V'$，*并采用下列方式更新* V' *和* E'：$V':=V'\cup\{j\}$，$E':=E'\cup\{(i,j)\}$，转到第 (2) 步。

该算法是从仅有一个顶点、0 条边的子图开始，以后每执行一次第 (3) 步就增加一个顶点和一条边。这就意味着最终生成的树有 $|V|$ 个节点，$|V|-1$ 条链路。通常对于一个连通图，其边的条数大于等于 $|V|-1$。如果一个图 G 的边的数目等于 $|V|-1$，则上述算法将使用该图中所有的边，因而有 $G=G'$，即此时图 G 本身就是一个树。

一般而言，对于一个图可以有很多个生成树。对于通信网络来说，利用生成树来实现广播是比较经济的，但每一条边的成本或时延通常是不相同的，这时就要考虑树中各边的重量（成本或时延）。通常用 W_{ij} 表示边 (i,j) 的重量。

最小重量生成树 (MST，Minimum Weight Spanning Tree)：是指边的重量和最小的生成树。

MST 的任一个子树称为一个树枝 (fragment)。（注意：一个顶点本身就是自己的一个树枝）。

输出链路 (Outgoing Arc) 是指该链路的一个端点在 fragment 内，而另一个端点不在该 fragment 内。这里所谓的链路和图的边的概念是等效的。

下面讨论对于一个给定的图 $G=(V,E)$，如何构造该图的最小重量生成树 (MST)。

定理 1 给定一个 fragment F，令 $\alpha=(i,j)$，$i\in F$，$j\notin F$ 是 F *的最小重量*输出链路，将 F 扩展一条链路 α 和一个顶点 j，仍是一个 fragment。

由这一定理可知如何从 MST 的一个子树生成一个完整的 MST。

根据该定理可以有下列三种构造 MST 的方法（以图 1－22 (a) 为例）。

(1) Prim－Dijkstra 算法

选择任意一个顶点作为一个 fragment，然后根据定理 1，每次选择一条具有最小重量的输出链路来扩展 fragment，最终即可生成 MST，如图 1－22 (b) 所示。

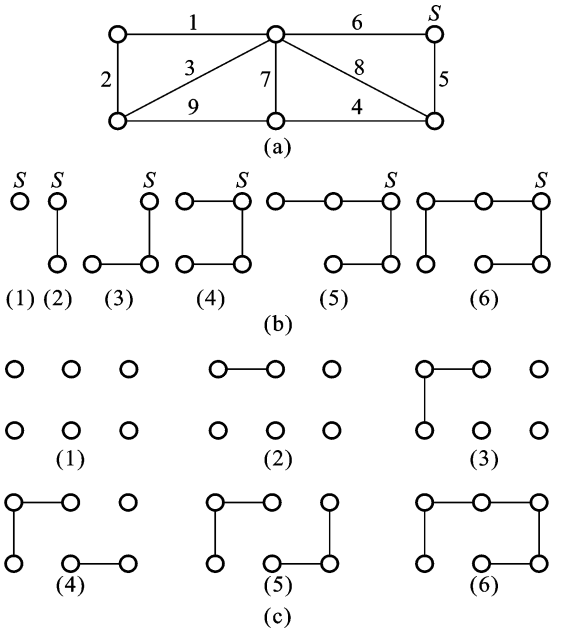


图 1-22 最小重量生成树的构造

(a) 网络拓扑 (b) Prim-Dijkstra 算法 (c) Kruskal 算法

(2) Kruskal 算法

选择所有的顶点作为单顶点的 fragment，在所有的链路中选择具有最小重量且不会形成回路（环）的链路添加到当前的 fragment 中。每次迭代仅添加一条链路。最终即可生成 MST，如图 1-22 (c) 所示。

(3) 分布式构造 MST 算法（假定网络中仅有惟一一个 MST）

从一个 fragment 集（例如：该集就是图中的某一个顶点）开始：

① 每个 fragment 决定它自己的最小重量输出链路，将该链路添加到自身的 fragment 中，并通知该输出链路的另一个端点。

② 只要连接两个 fragment 的链路的确是某个 fragment 的最小重量输出链路，则所有时间内，算法都维持着 MST 的 fragment 集，并且不会形成回路（环）。

③ 继续算法，添加新的链路，直至没有新的输出链路，且只有一个 fragment（即 MST）为止。

该算法不要求各 fragment 同步，只要有合适的确定最小重量输出链路的方法，就可以求得所需的 MST。

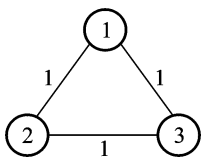


图 1-23 简单的图

上述分布式算法要求 MST 是惟一的，如果不惟一，则可能会引起闭环。

例如，有一个网络如图 1-23 所示。从三个顶点开

始构造 MST,则可能会出现三个顶点同时加入(1,2)、(2,3)和(3,1)三条链路,从而形成一个闭环。这个例子中,三条链路的重量是不可区分的。

定理 2 如果图 *G* 中所有链路的重量是不同的,则仅有惟一的一个 MST。

在实际的网络中,对于具有相同重量的链路,可以采用链路重量以及链路关联的两个顶点的标号来共同区分链路。例如,有两条链路(i, j)和(k, l)具有相同的重量,如果 $i<j, k<l$ 且有 $i<k$,则选定链路(i, j)是具有最小重量的链路。

4. 割集

设图 G=(V, E)是连通图,$S\subseteq E$,若从图 G 中消去属于 S 的所有的边,则图 G−S 称为一个非连通的图;而如果去掉属于 S 的任何真子集中的边,图仍然保持连通,则称 S 是图 G 的一个割集。也就是说割集 S 是使连通图 G 失去连通性的最小的边的集合。所谓饱和割集是指链路利用率(负荷)非常高的割集。例如图 1−24 中与虚线对应边是一个割集,它同时也是一个饱和割集(图中链路旁的数字表示该链路的利用率)。

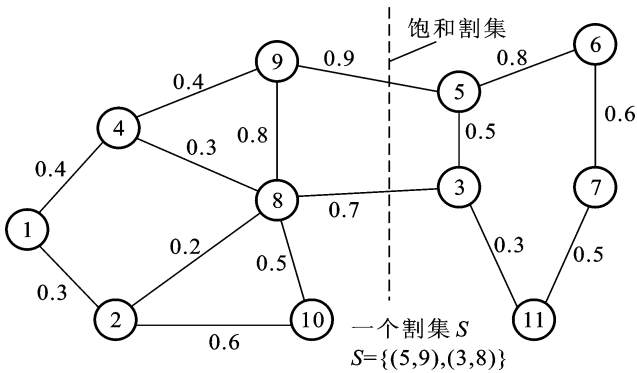


图 1−24 割集和饱和割集示意图

1.4 通信网络的基本理论问题

在一个由多种类型的物理网络构成的通信网络中,对每个用户而言,所关心的问题主要是如何将其消息快速而准确地传到对方。用户的信息通常要跨越多个网络,例如,图 1−2 中用户 D 到用户 G 之间要跨越移动通信链路、SDH 链路、WDM 链路、ATM 链路、以太网等。因此,不仅要关心两个相邻节点之间链路的传输可靠性和有效性,而且还要关心同一种物理媒介网络中任意两个节点之间的链路的传输可靠性和有效性。此外,还要关心穿越不同物理媒介网络的任意两个节点之间的传输可靠性和有效性。以上是从用户的观点来看待一个网络。

若从网络设计的观点来看待网络中的基本问题,所碰到的第一个最基本的问题是众多的用户如何共享一个物理媒介,即多址问题。第二个问题是如何设置网络的接入点和网络节点,使得众多的用户能够方便地接入到网络之中,经济地共享高速大容量的骨干链路和网络。这是网络拓扑设计和网络覆盖问题。第三个问题是如何为用户的消息或分组选择最佳的传输路径,从而使得用户的消息或分组在一个子网内或跨越多个网络时能快速、可靠地传送到对方。这是路由问题。第四个问题是如何保证网络稳定运行,即如何避免网络中某条链路、某个子网或整个信息网络发生拥挤和阻塞,它包括用户接入网络的业务流量和网络内部的流量管理和控制。这是流量控制问题。

根据上述讨论,本书的主要内容安排如下:

第2章:端到端的传输协议。主要讨论错误检测,自动请求重传协议,组帧的主要问题以及网络层和运输层的端到端的协议。

第3章:网络的时延模型。主要介绍排队论的基本知识,它是后面章节讨论的基础。

第4章:多址协议。本章以共享传输媒介的网络(如局域网和无线通信网)为基础,讨论随机多址的基本问题、冲突分解方法及其性能改进的方法。

第5章:路由算法。主要讨论最短路由算法、路由信息的广播等。

第6章:流量控制。主要讨论接入允许控制、窗口流量控制和闭环流量控制等。

第7章:网络的拓扑设计。主要讨论根据用户的需求,如何最优设置用户接入点和网络节点。

习　题

1.1　通信网络由哪些基本要素组成?试举例列出五种常用的通信网络。

1.2　常用的通信链路有哪些?其主要特征是什么?

1.3　试简述分组交换网的要点。

1.4　什么叫做虚电路?它与传统电话交换网中的物理链路有何差异?

1.5　ATM信元与分组有何差别?ATM网络是如何支持不同种类业务的?

1.6　分层的基本概念是什么?什么是对等层?

1.7　试述OSI七层模型和TCP/IP协议体系的区别和联系。

1.8　一个典型的通信网络可由哪些物理子网构成?路由器在该网络中的作用是什么?

1.9　通信网络要研究的基本理论问题有哪些?

1.10　设随机过程 $X(t)$ 定义为 $:X(t)=2\cos(2\pi t+Y)$，其中 Y 是离散随机变量，且 $P\{Y=0\}=\frac{1}{2}$，$P\left\{Y=\frac{\pi}{2}\right\}=\frac{1}{2}$。试求该过程在 $t=1$ 时的均值，和 $t_1=0, t_2=1$ 时的自相关函数值。

1.11　设随机过程 $X(t)$ 是一个随机相位信号，即 $X(t)=A\cos(\omega_c t+\theta)$，式中 A 和 ω_c 为常量，θ 是一个均匀分布的随机变量，其概率密度函数为 $f(\theta)=\frac{1}{2\pi}$，$-\pi<\theta<\pi$。试求 $X(t)$ 的均值函数和自相关函数。并讨论其平稳性和各态历经性。

1.12　试求 Poisson 过程的均值函数、方差函数和相关函数。

1.13　设到达某商店的顾客组成强度为 λ 的 Poisson 流，每个顾客购买商品的概率为 p，各顾客是否购买商品与其他顾客无关，分别用 $\{Y(t), t>0\}$ 和 $\{Z(t), t>0\}$ 表示购买商品顾客和未购买商品顾客的顾客流过程，请证明他们分别是强度为 $p\lambda$ 和 $(1-p)\lambda$ 的 Poisson 流。

1.14　设某办公室来访的顾客数 $N(t)$ 组成 Poisson 流，平均每小时到访的顾客数为 3 人，求：(1) 上午 (8 到 12 点) 没有顾客来访的概率；(2) 下午 (2 点到 6 点) 第一个顾客到达的时间分布。

1.15　设有三个黑球和三个白球，把这六个球任意分给甲乙两人，并把甲拥有的白球数定义为该过程的状态，则有四种状态 0, 1, 2, 3。现每次从甲乙双方各取一球，然后相互交换。经过 n 次交换后过程的状态记为 X_n，试问该过程是否是马氏链？如是，试计算其一步转移概率矩阵，并画出其状态转移图。

1.16　分别利用 Prim－Dijkstra 算法和 Kruskal 算法求解图 1－25 中的最小重量生成树。

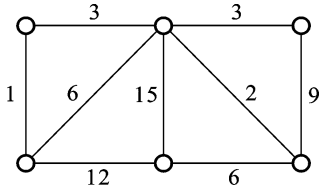


图 1－25　习题 1－16 图

第2章 端到端的传输协议

第1章已提到,物理层是为链路层提供一组虚拟的比特管道。本章将讨论在这样的比特管道上如何形成一条可靠的业务通道为上层提供可靠的服务。为了形成一条可靠的业务通道,首先解决如何标识高层送下来的数据块(分组)的起止位置,接着就要解决如何发现传输中的比特错误,最后要解决的就是发现错误后,如何消除这些错误。根据通信双方所处的相对位置不同,所采用的解决方法(协议)也不同。若通信双方之间是通过一条物理链路直接相连,这时需采用链路层的协议来解决数据帧的传输错误;若通信双方是在一个通信子网内通过多条数据链路形成的通路相连,这时需要采用网络层的端到端传输协议来解决分组的传输错误;当通信双方处于不同的通信子网时,需要采用运输层的端到端传输协议来解决报文的传输错误。本章首先讨论组帧技术、差错检测、自动请求重发的协议和典型的数据链路层协议;接着讨论网络层的端对端传输协议、IP协议;最后讨论运输层的端到端传输协议。

2.1 组帧技术

物理层通常仅负责比特的传输,而不对比特的含义和作用进行区分。因此当数据链路层将网络层的分组连续送到物理层进行传输时,就有一个问题要解决,如何决定什么时刻是一帧(链路层传送的一个数据比特块(数据单元)称为一帧,每一帧通常运载网络层的一个分组)的开始时刻?什么时刻是一帧的结束时刻?哪一段传输的是用于差错校验的比特?这就是组帧技术需解决的问题。

有三种组帧的方式:一是面向字符的组帧技术,二是面向比特的组帧技术,三是采用长度计数的组帧技术。

2.1.1 面向字符的组帧技术

所谓面向字符的组帧技术是指物理层传输的基本单元是一个字符(通常用一个字符可用一个字节来表示),并在此基础上形成具有一定格式的字符串。

在物理层中,可以有多种方式来实现字符的传输,例如可以采用RS-232C异步串行接口协议。该协议在传送每个字符(如一个字符由8个比特D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0组成)前后分别加上起始位($D_{起}$)、停止位($D_{止}$),以便区分不同的字符,如图2-1所示。

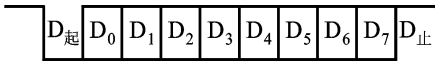


图2－1 RS－232C串行通信字符传输格式示意图

Internet中常用的面向字符的组帧技术的协议有SLIP(Serial Line Internet Protocol)和PPP(Point to Point Protocol)。

SLIP的帧格式如图2－2所示。SLIP帧运载的是高层IP数据报。它采用两个特殊字符：END(十六进制C0H,这里H表示十六进制)和ESC(十六进制DBH)。END用于表示一帧的开始和结束。为了防止IP数据报中出现相同的END字符而使收端错误地终止一帧的接收，SLIP中使用了转义字符ESC。当IP数据报中出现END字符时，就转换成ESC和ESC－END(其中ESC－END＝DCH)两个字符。当IP数据报中出现ESC时，就转换成为ESC和ESC－ESC(其中ESC－ESC＝DDH)两个字符。这样收端只要收到END字符即表示一帧的开始或结束。每当遇到ESC字符就进行字符转换，恢复IP报文中的原有的END和ESC字符。这样就可以完全以一个IP数据报的形式向IP层提交数据。

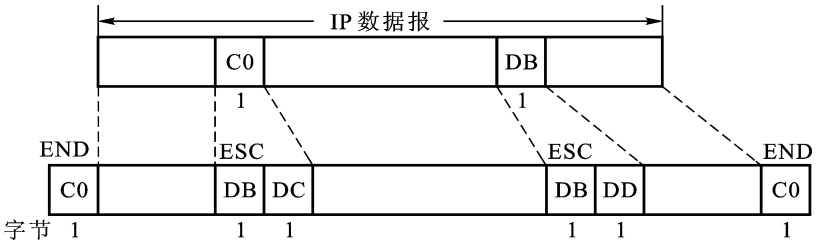
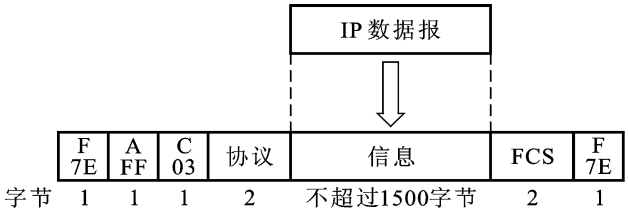


图2－2 SLIP协议的帧格式

PPP的帧格式如图2－3所示，该格式与后面要讨论的HDLC协议的格式相同。PPP中，采用7EH作一帧的开始和结束标志(F)；其中地址域(A)和控制域(C)取固定值(A＝FFH，C＝03H)；协议域(两个字节)取0021H表示该帧运载的信息是IP数据报，取C021H表示该帧的信息是链路控制数据，取8021H表示该帧的信息是网络控制数据；帧校验域(FCS)也为两个字节，它用于对信息域的校验。若信息域中出现7EH，则转换为7DH，5EH两个字符。当信息域出现7DH时，则转换为7DH，5DH。当信息流中出现ASCII码的控制字符(即小于20H)，即在该字符前加入一个7DH字符。



一种用于 ARPANET 同步通信的面向字符帧格式如图2－4所示。图中，DLE STX 两个字符表示帧开始，DLE ETX 两个字符表示帧结束，CRC 为校验字符。当分组数据中出现 DLE 字符时，转换为 DLE DLE 两个字符。图中，SYN 字符是在两帧之间无数据传输时，为维持物理链路同步而插入的同步空闲字符。

注:SYN=同步,STX=开始发送,DLE=数据链路转义字符,ETX=发送结束。

图2－4 用于 ARPANET 的面向字符帧格式

理解为黑匣子

上述三种帧格式均支持数据的透明传输，这些帧结构在处理时非常简单，但缺点是效率较低，插入了许多转义字符。另外，数据长度必须以字节为单位。

2.1.2 面向比特的组帧技术

在面向比特的组帧技术中，通常采用一个特殊的比特串（称为 Flag），如 01^60（1^j 表示连续 j 个“1”）来表示一帧的正常结束和开始。这里与面向字符的组帧技术面临相同的问题，即当信息比特流中出现与 Flag 相同的比特串（如连续出现 6 个“1”）如何处理？这里采用的办法是比特插入技术，发端信息流中，每出现连续的 5 个“1”就插入一个“0”，如图 2－5 所示。这样被插“0”的信息比特流中就不会有多于 5 个“1”的比特串。接收端在收到 5 个“1”以后，如果收到的是“0”就将该“0”删去 如果是“1”就表示一帧结束。

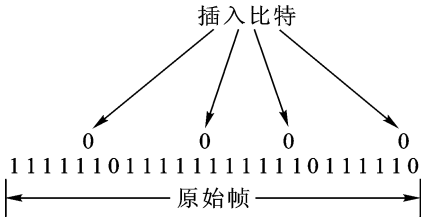


图2－5 比特插入技术

采用比特插入技术，除了消除信息帧中出现 Flag 的作用以外，它还带来其他作用，如要丢弃或中止一帧，则可连续发送 7 个或 7 个以上的“1”。当链路连续出现 15 个“1”则认为链路空闲。因此 01^6 是一个结束标志，如果 01^6 后面是 0 表示正常结束，如果 01^6 后面是 1 表示非常中止。

针对该面向比特的组帧技术，我们希望知道：Flag 的长度是多少时，帧的开销最小。

设输入的信息比特流是独立同分布的二进制变量，其“0”和“1”等概出现。假定采用 01^j 作为结束标志，现在来求 j 为多少时效率最高或插入比特开销最小。

首先考察在第 i 位要插入“0”的情况及相应概率。如果原始数据从 $i-j+1$ 位到 i 位（$i \geq j$）的比特为 01^{j-1}，则在第 i 位后面将要插入一个“0”，其概率为 2^{-j}。如果原始数据从 $i-2j+2$ 到 i 位（$i \geq 2j-1$）的比特为 01^{2j-2}，则也将在

第 i 位后面插入一个“0”，其概率为 2^{-2j+1}。可以继续考察在第 i 位前连续出现 $n\ (j-1)$ 个“1”的情况及相应的概率。在后面的讨论中，将忽略第二次插“0”及更长连续“1”的插“0”情况。这里需要注意的是：如果输入比特流的前 j－1 个比特均为“1”，则将要在第 j 位插入一个“0”，其概率为 2^{-j+1}。

设原始数据的长度为 k，k≥ j－1 时的平均插入“0”的数目为

$$1\cdot 2^{-j+1}+[k-(j-1)]\cdot 2^{-j}=(k-j+3)2^{-j}$$

加上一个结束标志，总的开销为

$$OV=(k-j+3)2^{-j}+j+1 \qquad (2-1)$$

对上式取均值得

$$E\{OV\}=(E\{k\}-j+3)2^{-j}+j+1 \qquad (2-2)$$

通常 $E\{k\}\gg j$，所以上式可以用一个上界来表示，即 $(j>3)$

$$E\{OV\}\le E\{k\}2^{-j}+j+1 \qquad (2-3)$$

上式右边对应的最小 j 值为

$$j=\mathrm{Int}[\mathrm{lb}\{E\{k\}\}] \qquad (2-4)$$

式中，Int［x］表示取 x 的整数部分，lb 表示以 2 为底的对数（lb＝$\log_2$）。

将式（2－4）代入式（2－3）得

$$E\{OV\}\le \mathrm{lb}\{E\{k\}\}+2 \qquad (2-5)$$

例如：$E\{k\}=1000$ bit 时，最佳的 j＝9，平均一帧的开销小于 12 bit。而在标准的 DLC 协议，取 j＝6，平均一帧的开销约为 23 bit。因此，从理论上讲，应当改变现有标准 DLC 的帧标志位的长度。

2.1.3 采用长度计数的组帧技术

组帧技术的关键是正确地表示一帧何时结束，除前面采用 Flag 和特殊字符外，还可以采用帧长度来指示一帧何时结束，如图 2－6 所示。如果最大长度为 K_{max}，则长度域的比特数至少为 $\mathrm{Int}[\mathrm{lb}\{K_{max}\}]+1$。长度域的比特数通常是固定的。例如，DECNET 就采用了该组帧技术。

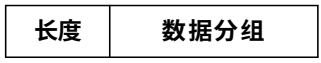


图 2－6 采用长度的组帧技术

与式（2－5）相比，可以看出采用该技术的开销与采用 Flag 的开销类似。

2.2 链路层的差错控制技术

2.2.1 差错检测

链路层差错检测的目的是：如何有效地发现一帧数据比特经过物理信道传

输后是否正确。

常用的检错方法有两类：一类是奇偶校验，另一类是循环冗余校验 CRC（Cyclic Redundancy Check）。其基本思路是发端按照给定的规则在 K 个信息比特后面增加 L 个按照某种规则计算的校验比特，在接收端对收到的信息比特重新计算 L 个校验比特。比较接收到的校验比特和本地重新计算的校验比特，如果相同则认为传输无误，否则认为传输有错。

1. 奇偶校验码

奇偶校验的种类很多，这里给出一个奇偶校验码的例子，如表 2-1 所示。这里信息序列长 K=3，校验序列长 L=4。输入信息比特为 $\{S_1, S_2, S_3\}$，校验比特为 $\{C_1, C_2, C_3, C_4\}$。校验的规则为：$C_1 = S_1 \oplus S_3$，$C_2 = S_1 \oplus S_2 \oplus S_3$，$C_3 = S_1 \oplus S_2$，$C_4 = S_2 \oplus S_3$。（注：⊕为模 2 加法）。

异或---****

表 2-1 奇偶校验码

S_1	S_2	S_3	C_1	C_2	C_3	C_4	校验规则
1	0	0	1	1	1	0	$C_1 = S_1 \oplus S_3$
0	1	0	0	1	1	1	$C_2 = S_1 \oplus S_2 \oplus S_3$
0	0	1	1	1	0	1	$C_3 = S_1 \oplus S_2$
1	1	0	1	0	0	1	$C_4 = S_2 \oplus S_3$
1	0	1	0	0	1	1	
1	1	1	0	1	0	0	
0	0	0	0	0	0	0	
0	1	1	1	0	1	0	

例如：设发送的信息比特为 {100}，经过奇偶校验码生成的校验序列为 {1110}，则发送的信息序列为 {1001110}。若经过物理信道传输后，接收的序列为 {1011110}，则本地根据收到的信息比特 {101} 计算出的校验序列应为 {0011}。显然该序列与接收到的校验序列 {1110} 不同，表明接收的信息序列有错。

如果 L 取 1，$C = S_1 \oplus S_2 \oplus S_3 \oplus \cdots \oplus S_K$，则该方法即为最简单的单比特的奇偶校验码，它使得生成的码字（信息比特＋校验比特）所含“1”的个数为偶数。该码可以发现所有奇数个比特错误，但是不能发现任何偶数个错误。

在实际应用奇偶校验码时，每个码字中 K 个信息比特可以是输入信息比特流中 K 个连续的比特，也可以在信息流中每个一定的间隔（如一个字节）取出一个比特来构成 K 个比特。为了提高检测错误的能力，可将上述两种取法重复使用。

2. CRC 校验

CRC（循环冗余校验）是根据输入比特序列（$S_{K-1}, S_{K-2}, \cdots, S_1, S_0$）通过下列 CRC 算法产生 L 位的校验比特序列（$C_{L-1}, C_{L-2}, \cdots, C_1, C_0$）。

CRC 算法如下：

将输入比特序列表示为下列多项式的系数

$$S(D) = S_{K-1} D^{K-1} + S_{K-2} D^{K-2} + \cdots + S_1 D + S_0 \qquad (2-6)$$

式中：D 可以看成为一个时延因子，D^i 对应比特 S_i 所处的位置。

设 CRC 校验比特的生成多项式（即用于产生 CRC 比特的多项式）为

$$g(D) = D^L + g_{L-1} D^{L-1} + \cdots + g_1 D + 1 \qquad (2-7)$$

则校验比特对应下列多项式的系数

$$C(D) = \text{Remainder}\ \frac{S(D) \cdot D^L}{g(D)} = C_{L-1} D^{L-1} + \cdots + C_1 D + C_0 \qquad (2-8)$$

式中：Remainder [·] 表示取余数。式中的除法与普通的多项式长除相同，其差别是系数是二进制，其运算以模 2 为基础。［例如，$(D^5 + D^3) / (D^3 + D^2 + 1)$ 的商为 $D^2 + D$，余数为 $D^2 + D$。］最终形成的发送序列为（$S_{K-1}, S_{K-2}, \cdots, S_1, S_0, C_{L-1}, \cdots, C_1, C_0$）。

生成多项式的选择不是任意的，它必须使得生成的校验序列有很强的检错能力。常用的几个 L 阶 CRC 生成多项式为：

CRC－16（L＝16）

$$g(D) = D^{16} + D^{15} + D^2 + 1 \qquad (2-9)$$

CRC－CCITT（L＝16）

$$g(D) = D^{16} + D^{12} + D^5 + 1 \qquad (2-10)$$

CRC－32（L＝32）

$$g(D) = D^{32} + D^{26} + D^{23} + D^{22} + D^{16} + D^{12} + D^{11} + D^{10} + D^8 + D^7 + D^5 + D^4 + D^2 + D + 1 \qquad (2-11)$$

其中，CRC－16 和 CRC－CCITT 产生的校验比特为 16 比特，CRC－32 产生的校验比特为 32 比特。

例 2.1 设输入比特序列为（10110111），采用 CRC－16 生成多项式，求其校验比特序列。

解 输入比特序列可表示为：

$$S(D) = D^7 + D^5 + D^4 + D^2 + D^1 + 1, \quad (K=8)$$

因为 $g(D) = D^{16} + D^{15} + D^2 + 1, \quad (L=16)$

所以

$$C(D) = \text{Remainder}\ \frac{S(D) \cdot D^L}{g(D)} = \text{Remainder}\ \frac{D^{23} + D^{21} + D^{20} + D^{18} + D^{17} + D^{16}}{D^{16} + D^{15} + D^2 + 1}$$

$$= \text{Remainder}\frac{(D^7+D^6+D^4+D^3+D)(D^{16}+D^{15}+D^2+1)+D^9+D^8+D^7+D^5+D^4+D}{D^{16}+D^{15}+D^2+1}$$

$$= D^9+D^8+D^7+D^5+D^4+D$$

$$= 0\cdot D^{15}+0\cdot D^{14}+0\cdot D^{13}+0\cdot D^{12}+0\cdot D^{11}+0\cdot D^{10}+1\cdot D^9+1\cdot D^8+1\cdot D^7+0\cdot D^6+1\cdot D^5+1\cdot D^4+0\cdot D^3+0\cdot D^2+1\cdot D^1+0$$

由此式可得校验比特序列为：(0000001110110010)。最终形成的经过校验后的发送序列为 (1011011100000001110110010)。

在接收端，将接收到的序列

$$R(D)=r_{K+L-1}D^{K+L-1}+r_{K+L-2}D^{K+L-2}+\cdots+r_1D+r_0 \quad (2-12)$$

与生成多项式 g(D) 相除，并求其余数。如果 $\text{Remainder}\frac{R(D)}{g(D)}=0$，则认为接收无误。$\text{Remainder}\frac{R(D)}{g(D)}=0$ 有两种情况：一是接收的序列正确无误；二是 *R(D) 有错，但此时的错误使得接收序列等同于某一个可能的发送序列。出现后一种情况称为漏检*。

注意

当信息比特序列长度小于 2^{L-1} 比特时，由式 (2-9)、(2-10) 和 (2-11) 生成的 CRC 校验码可以检测所有单个和两个错误比特，能够检测出长度不超过 L 的突发错误。在二进制对称信道 (BSC) 下其漏检概率小于等于 2^{-L}。

2.2.2 ARQ 协议

2.2.1 节解决了如何发现传输帧的错误问题，这一节要解决当接收端发现传输帧有错时处理的方法。出错的最简单处理方法是 (收端) 自动请求发端重发 (ARQ，Automatic Retransmission Request)。即收端收到一帧后，经过 CRC 检验，如果发现该帧传输有误，则通过反馈信道 (该信道可以与前向传输相同，也可以不同) 以某种反馈规则通知发端重复上述过程，直到收端收到正确的帧为止。对反馈规则和重传规则的设计，要保证整个自动重传协议的正确性和有效性。

为了研究 ARQ 协议，对物理比特管道 (物理链路) 作如下假定：

(1) 在物理信道上传输的帧到达接收端前被时延了一个任意可变的时间；

(2) 帧在传输过程中可能会丢失，也可能出错；

(3) 帧到达的顺序与发送的顺序相同。

有四种不同形式的 ARQ 重传协议：停等式 ARQ、返回 *n*－ARQ、选择重发式 ARQ 和并行等待式 ARQ，下面分别进行讨论。

1. 停等式 ARQ

停等式 ARQ (Stop and Wait ARQ) 的基本思想是在开始下一帧传送以前，

必须确保当前帧已被正确接收。假定A发,B收。具体的传送过程如下:A发送一帧后,B如果接收正确,则B和A返回一个肯定的应答(ACK);B如果接收错误,则B向A返回一个否定应签(NAK)。A必须在收到B的正确ACK后,方可发送下一帧。如果A发送一帧后(并给定时器设置一个初值),在一个规定的时间内(定时器溢出),没有收到对方的ACK,则重发该帧;如果收到了NAK,也要重发该帧。

由于A到B之间的双向链路都可能出错,如何保证该协议能够正确工作呢?基本的方法是在传输的帧中增加发送序号(SN)和接收序号(RN),如图2-7所示。

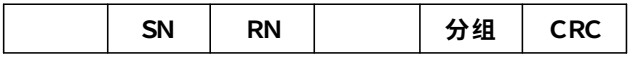


图2-7 ARQ帧结构

假设没有发送序号SN,看该ARQ能否正确工作。A发送一帧给B后,B接收正确,B返回给A一个ACK。如果ACK在传输过程中丢失或被时延,则A会重发刚才的帧。B接收正确后,就会出现B接收到两个完全相同的帧。这时B将无法区分这两个帧是不同的帧还是重复帧。这说明发端必须增加一个发送序号。

假设没有接收序号RN,看该ARQ能否正确工作。假设B返回的ACK被时延,可能出现的情况如图2-8所示。

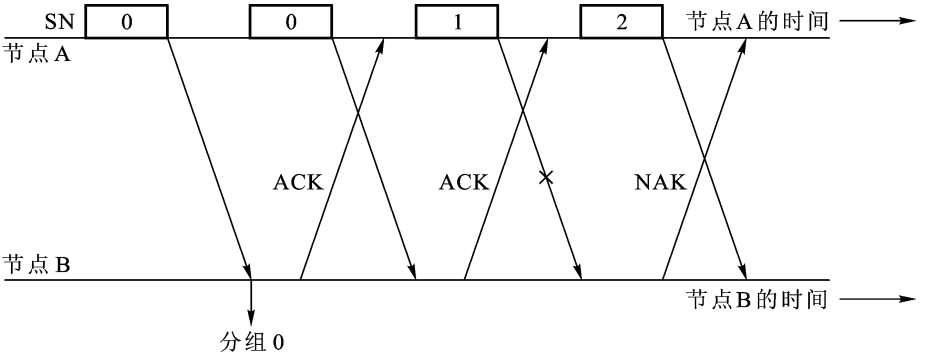


图2-8 无接收序号时停等式ARQ的工作过程

A发送的序号为0的帧被B正确接收后,由于在定时器溢出时,ACK未到达A,A重发SN=0的帧。B的第一次应答ACK到达A后,A认为SN=0帧已被正确接收,发送SN=1的帧。B对重发的SN=0帧的应答ACK到达A后,A认为这是对SN=1的帧的应答,因而认为SN=1的帧已被正确接收(A无法区分该ACK是对SN=0还是对SN=1帧的应答)。A将发送SN=2的帧。如果SN=1的帧传输出错,A会在发送SN=2的帧后收到NAK应答,此时A端会认

为 SN=2 的帧传输出错,因而会重发 SN=2 的分组。这就导致了 SN=1 的帧丢失。解决该问题的方法是:收端不是简单地回答 ACK 或 NAK,而是告诉对方正在等待接收的下一帧的序号。(当然也可以是正确接收帧的最大序号。)通常 A 到 B 之间双向都有帧要传输,因而可将接收序号(RN)附带在反向传输数据的帧头中,以提高传输效率。

综上所述,在 ARQ 帧中,必须有 RN 和 SN。

等待式 ARQ 协议的严格描述如下:

假定 A 向 B 发送分组(A→B),节点 A 的发送算法如下:

发送的规则;A发送到B,则A需要遵循的规则。

(1) 置 SN=0。

(2) 如果从高层接收到一个分组,则将 SN 指配给该分组;如果没有高层分组则等待。

(3) 将发送序号为 SN 的分组装入的物理帧中发送给接收节点 B。

(4) 如果从 B 接收的 RN>SN,则将 SN 加 1,返回(2)。如果在规定的有限长时间内,没有从 B 接收到 RN>SN 的帧(应答),则返回(3)进行重传。

节点 B 的接收算法如下:

(1) 置 RN=0。

(2) 无论何时从 A 正确接收一个 SN=RN 的帧,将该帧中的分组送给高层,并将 RN 加 1。

(3) 在接收到该分组后的一个规定的有限长时间内,将 RN 放入一帧的 RN 域中发给 A。返回(2)。

在上述算法中,规定时间通常是采用定时器来确定的。RN 通常是附带在反向数据帧中传给对方的。如果节点 B 没有数据传送给对方,则应单独传送一个包含 RN 的无数据帧给 A。

从上面的叙述可以看出,反向业务流的存在对停等式 ARQ 的机制没有任何影响,它仅对应答的时延有所影响,因此在后面的讨论中将忽略反向业务流。

对于上述停等式 ARQ 协议或其他类似的协议,要从两个主要方面对其进行评估,一是算法(协议)的正确性,二是算法(协议)的有效性。

所谓算法的正确性是指算法始终能够正常工作,正确接收输入的数据和状态,产生正确的动作和正确的输出结果。对于不同的算法,其正确性有不同的描述。对于停等式 ARQ 协议而言,所谓"该算法是正确的"是指:A 能够始终不断地从高层接收数据分组,B 能够始终按照发端的顺序向 B 的高层呈送接收到的分组,既不重复,也不丢失。

算法的正确性证明可分为两部分:一是稳妥性(Safety),即算法决不会产生不正确的结果(在这里是指提交给上层分组的顺序不对);二是活动性(Liveness),即算法会永远不停地产生结果(不会产生死循环,一旦进入死循环,

将不会进行进一步的处理),在这里指在 A 节点能够不停地接收高层的分组,在 B 节点能够不停地将这些分组呈送给高层。

在停等式 ARQ 中,稳妥性是很明显的。在算法开始时,SN=0,RN=0,节点 B 等待 RN=0 的帧因而仅会有 SN=0 帧中的分组送上高层。由于 RN 是将要接收的下一个帧的序号,接收到一个新的帧后 RN 加 1,这样重发帧的序号小于 RN,因而其中的分组不可能送给高层。所以,节点 B 能按顺序将分组呈现给高层。

为了考察该算法的活动性,假定 A 在 t_1 时刻开始发送给定的分组 i,B 在 t_2 时刻正确接收到该分组,A 在 t_3 时刻正确接收到 B 的应答。显然,如果 $t_2=\infty$,则表示 B 收不到 A 的分组 如果 $t_3=\infty$,则表示 A 收不到 B 的应答。因而,要证明该算法的活动性就要证明 $t_1<t_2<t_3$ 且 $t_3<\infty$。

为了证明该协议的正确性,假定:

① 所有错误的分组都能被 CRC 检出;

② 一个分组能够被正确接收的概率为 $q, q>0$;

令 t 时刻的 RN 和 SN 为 RN(t) 和 SN(t),显然 RN(t) 和 SN(t) 是时间 t 的非降函数。由于 SN(t) 是 t 时刻以前从 B 接收到的最大请求序号。所以 $SN(t)\leq RN(t)$。由于分组 i 在 t_1 时刻开始第一次传输 (A 在 t_1 以前从未传送过分组 i),则根据稳妥性 (算法不会产生不正确的结果) 有 $RN(t_1)\leq i$。*又因为* $SN(t_2)=i$,*所以有* $SN(t_2)=RN(t_2)=i$。

又根据假定:在 $t=t_2$ 时,$RN(t_2)=i+1$;在 $t=t_3$ 时,$SN(t_3)=i+1$。利用 RN(t) 和 SN(t) 的非降特性和 $SN(t)\leq RN(t)$,*则有* $t_2<t_3$。

根据算法,A 将会重复发送分组 i,其重发的间隔是有限的。又由于分组被成功接收的概率 q>0,因而分组 i 会在有限次传输后被正确接收。也就是说,从 t_1 至 t_2 的间隔是有限的。由于 A 不停地以有限长的间隔发送分组 i,这将导致节点 B 以有限的间隔发送应答 (包括 RN 的分组)。由于 B 成功传输的概率 $q>0$,所以应答在有限的时间内到达 A,即 t_3 是有限的。

综上所述有 $t_1<t_2<t_3<\infty$,所以算法是活动的。所以,停等式 ARQ 协议是正确的。

在前面的讨论中,假定 SN 和 RN 是随时间任意增加的。这样就需要很大的比特域来表示发送序号。但实际上是不需要的,可以采用一个整数的模值 (mod *N*) *来表示,如* SN mod8,SN mod128 等。很容易看出对于停等式 ARQ,取模 2 就足够了。

算法的有效性可以从三个主要方面来表述:一是**吞吐量 (或通过量)**,二是**链路的利用率**,三是**分组延迟**。所谓吞吐量就是在给定的物理信道和输入分组流的条件下,接收端能够呈送给高层的分组速率 (分组/秒或比特/秒)。所谓链路

的利用率就是指物理层（比特管道）的传输容量中用于有效分组传输所占的比例。在信道中，如果被等待、重传和其他不必要的传输所占的比例越少，则说明信道利用率越高。分组延迟是指链路层从发端收到高层的分组开始到收端将该分组呈送给高层为止所需的时间。

设数据帧是固定帧长，其传输时间为 T_D（秒），肯定和否定应答帧长为 T_{ACK}（秒），物理信道的传播时延为 T_P（秒），则在忽略算法的处理时延的情况下，一帧的传输周期为（$T_D + T_P + T_{ACK} + T_P$），假定任意一个数据帧平均需要发送 N_T 次（一次初发，$N_T - 1$ 次重发）才能成功。则该帧平均一共需要 N_T 个传输周期，如图 2-9 所示。由此可以看出，物理链路的**最大平均利用率**为（即物理链路以最大可能的负荷在传输时的**平均利用率**）：

$$U = \frac{T_D}{N_T(T_D + T_P + T_{ACK} + T_P)} = \frac{1}{N_T(1 + 2T_P/T_D + T_{ACK}/T_D)} \tag{2-13}$$

令 $\alpha = T_P/T_D$，忽略应答帧的传输时间（认为 $T_{ACK} = 0$），则有

$$U = \frac{1}{N_T(1 + 2\alpha)} \tag{2-14}$$

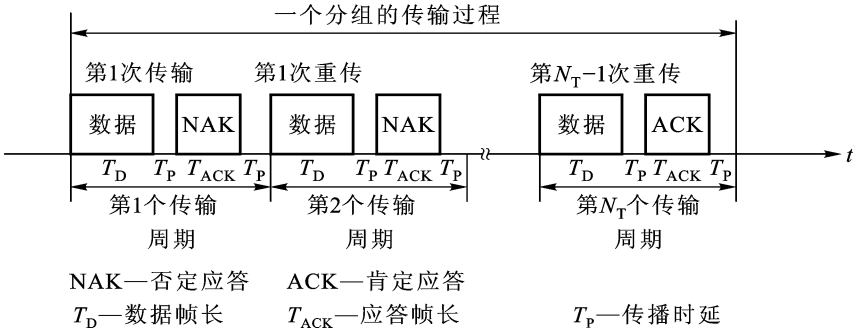


图 2-9 传输周期示意图

又假定数据帧的误帧率为 $p = 1 - q$，*应答帧因长度很短，其出错的可能性*可以忽略，即认为应答帧总是可以正确传输，则一个数据帧发送 i 次成功的概率为 $p^{i-1}(1-p)$，*从而有*

$$N_T = \sum_{i=1}^{\infty} ip^{i-1}(1-p) = \frac{1}{1-p} \tag{2-15}$$

将上式代入式（2-14）得链路的最大平均利用率为

$$U = \frac{1-p}{1+2\alpha} \tag{2-16}$$

从上式可以看出，误帧率越高，链路的利用率越低，链路的传播时延越大，链路的利用率也越低。即使在 $p = 0$ 时，如果 $\alpha > 0.5$（帧长度小于传播时延的两

倍），则链路利用率小于50%。

根据图2－9可得，停待式ARQ的最大平均吞吐量为

$$S=\frac{1}{N_T(T_D+T_P+T_{ACK}+T_P)} \quad (分组/秒)$$

$$\approx\frac{1-p}{T_D(1+2a)} \quad (分组/秒) \qquad (2-17)$$

停待式ARQ的平均分组延迟为

$$D=组帧时延+(N_T-1)(T_D+T_P+T_{ACK}+T_P)+(T_D+T_P)$$

$$\approx组帧时延+N_T T_D+(2N_T+1)T_P \qquad (2-18)$$

组帧时延是指从高层分组的第一比特到达链路层开始，到链路层将该分组的所有比特收齐，经过增加控制头（如帧起止标志发送序号，接收序号等）和校验比特（CRC）形成可传输的数据帧为止。它取决于网络层与链路层之间的接口速率和方式，以及链路层的处理速度和方式。例如：若网络层与链路层运行在相同的微处理器或计算机系统上，采用数据块传递的方式来传递分组，则组帧时延可以相当小，且可以忽略。如果网络层与链路层采用传输速率为R（比特/秒）的接口交换数据，则组帧的时延为K/R（这里K为分组的长度（比特数））。

例2.2 设有三种物理链路：一条是卫星链路，其信道速率为64 kbps，传播时延为T_P=270 ms，一条是经过电话网的呼叫接续构成的5000 km的链路，其信道速率为9600 bps，传输播时延T_P=25 ms；一条是由同轴电缆提供的长为500 m的链路，其信道速率为10 Mbps，传播时延T_P=2.5 μs。试求帧长为L=1000 bit和L=10000 bit时停等式ARQ的链路最大平均利用率U，最大平均吞吐量S（分组/秒）和平均分组时延D（ms）。

解 根据式（2－16）、（2－17）和（2－18）可得不同帧长情况，停等式ARQ的性能如表2－1所示。从表中可以看出传播时延相对于帧长的比例α越小，链路利用率越高，吞吐量也越大，时延也越小。由此可以看出：在卫星链路和电话网链路上采用停等式ARQ的效率很低。

表2－2 停等式ARQ的性能

性能参数 / 链路类型	传输速率	T_P	p	α		U		S		D	
				L=1000	L=10000	L=1000	L=10000	L=1000	L=10000	L=1000	L=10000
卫星链路	64 kbps	270 ms	10%	17.3	1.73	0.025	0.202	1.6	1.3	887	1043
电话网链路	9.6 kbps	25 ms	5%	0.24	0.024	0.766	0.906	7.4	1.0	187	1174
同轴电缆链路	10 Mbps	2.5 μs	1%	0.02	0.002	0.950	0.986	9500	986	0.1	1

2. 返回 n－ARQ

由于等停等式 ARQ 效率很低，人们想到发端在等待对方应答时，应当做更多的事情，这就提出了三种改进方案：返回 n－ARQ（Go Back n ARQ），选择重发式 ARQ（Selective Repeat ARQ）和并行等待式 ARQ（ARPANET ARQ）。

这里首先讨论返回 n－ARQ。它是一种运用最广泛的 ARQ 协议。例如，它已应用在 HDLC、SDLC、ADCCP 和 LAPB 等标准的 DLC 协议中。

返回 n－ARQ（有时也称为连续 ARQ）的基本思路是：发端在没有收到对方应答的情况下，可以连续发送 n 帧。收端仅接收正确且顺序连续的帧，其应答中的 RN 表示 RN 以前的所有帧都已正确接收。（这里收端不需要每收到一个正确的帧就发出一个应答，可对接收到的正确顺序的最大帧序号进行应答。）

这里 n 是一个重要参数，它叫做（滑动）窗口宽度，如图 2－10 所示。设窗口宽度为 5，则在开始时，发端可发送 0～4 号共 5 个帧（如图 2－10（a））。当收到对 SN＝0 帧的确认后，发端可发送 1～5 号共 5 个帧（图 2－10（b））。当收到对 3 号帧的确认后，发端可发送 4～8 号共 5 个帧（图 2－10（c））。随着应答的不断到达，发送窗口不断地向前滑动。

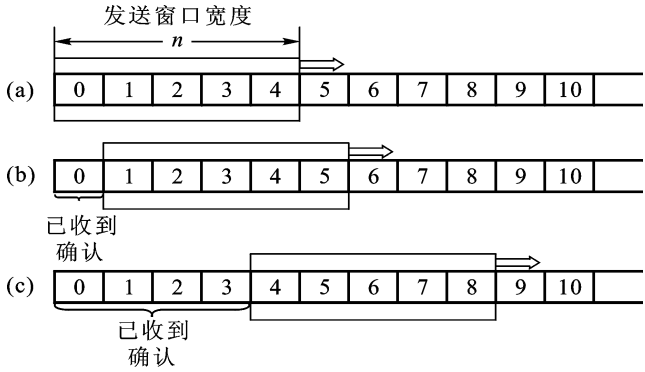


图 2－10　滑动窗口示意图

（a）开始时窗口所处的位置　（b）收到对 SN＝0 帧确认后窗口所处的位置

（c）收到对 SN＝3 帧确认后窗口所处的位置

从图中可以看出，如果收端能及时返回应答，则发端可连续不断地全速连续发送帧。（如果减缓应答返回的速率，则可以控制发端发送帧的速率，从而达到速率控制的目的。）

下面考察双向都有数据传输并且帧长度不等长时，发送端窗口滑动的情况。

图 2－11（a）显示了传输错误对发送窗口的影响。在该图的上面是发端发送的帧及其序号，下面是反向传输的帧。RN 嵌入在反向业务流中。因而 RN 的传输时延受反向业务流的限制。该图的窗口长度为 4。发端窗口开始时为

[0,3]，当收到 RN＝1 应答帧（表示 SN＝0 帧已正确接收）时，发端窗口变为 [1,4]。由于 SN＝1 的帧出错，在发送完 SN＝4 的帧后将等待对 SN＝1 帧的应答。在定时器溢出后，发端将从 SN＝1 的帧开始重发。当发端收到 RN＝2 的应答后，将窗口变为 [2,5]。在该图中，尽管 2,3,4 号帧传输正确，但它们的序号与接收端期望的序号不符而不能被正确接收，因而仍需要重传。

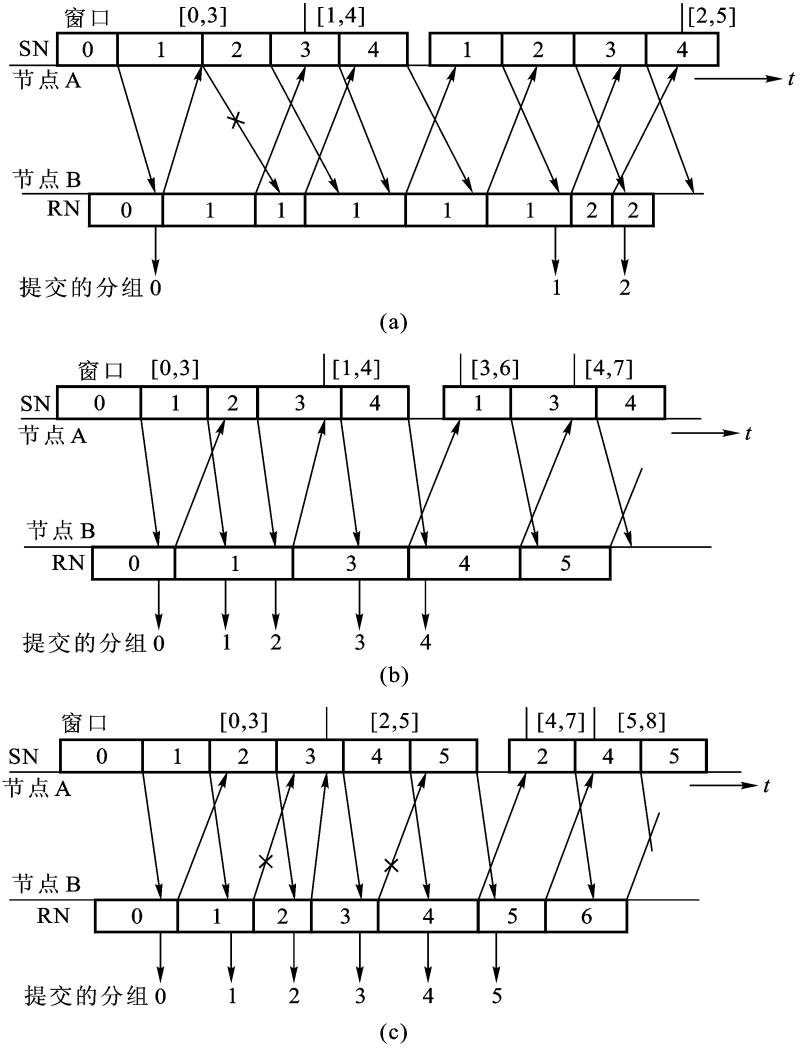


图 2－11 返回 *n*－ARQ 的帧传送过程

(a) 正向传输错误对发端窗口的影响 (b) 反向帧长对发端窗口的影响

(c) 反向传输出错对发端的影响

图 2－11 (b) 显示了反向帧长对发端窗口的影响。图中的窗口长度仍为 4。由于对 SN＝1 和 2 帧的应答包括在 RN＝3 的反向帧中，该帧被发端正确接收

时刻晚于发端对帧1设置的等待应答时限,这导致发端重传SN=1的帧。由于从SN=1帧开始重传的过程中已收到了对SN=2帧的应答,因而发端在重发SN=1的帧后,接着发送SN=3的帧,且发送窗口从[1,4]变为[3,6]。

图2-11(c)显示了反向帧出错对发端的影响。反向帧出错可能对发端无影响。如图中反向RN=1帧的传输出错,被反向RN=2帧的正确传输所补救,因而未对发端产生影响,发端收到反向RN=2的帧后,将窗口从[0,3]变为[2,5]。反向帧出错也可能对发端有影响。如图中反向RN=3帧的出错将导致发端的传输停顿,并在定时器溢出后重传SN=2的帧。

从上面的讨论可以看出,应答的超长时延或出错会导致发端重发。当然重发过程不一定是连续重发从某一需重发的帧开始的窗口内的所有帧,它取决于重发过程中收到应答的情况。

上面对返回 n-ARQ算法的基本过程进行了讨论,这里给出返回 n-ARQ算法的具体描述。(假定A为发端、B为收端。)

设发端使用SN_{min}表示A目前没有收到应答的帧中序号最小的帧(即发端窗口的低端),SN_{max}表示它将要指配给运载从高层新到达分组的帧序号。A节点将试图传送SN_{min}到$SN_{max}-1$之间的帧。

发端(A节点)的算法:

(1) 置$SN_{min}=0$,$SN_{max}=0$。

(2) 算法以任意顺序重复执行第(3)、(4)、(5)步。在每一步的条件满足时刻到该步被执行的时刻之间的时延是任意的,但是该时延是一个有限的值。

(3) 如果$SN_{max}<SN_{min}+n$,且上层有一个新分组到达,将SN_{max}指定给承载该分组的帧,并将SN_{max}加1。如何对帧进行编号

(4) 如果接收的$RN>SN_{min}$,则置$SN_{min}=RN$。接收到应答如何处理

(5) 如果$SN_{min}<SN_{max}$,且当前没有帧在传输,从$[SN_{min},SN_{max})$中选择一个或一组帧进行传输。当SN_{min}不再改变时,SN_{min}帧的重传间隔应当小于一个规定的有限值。帧如何正常传输和如何进行重传

收端(B节点)的算法:

(1) 置RN=0,重复执行(2)和(3)。

(2) 当接收到的SN=RN,将分组呈送给高层以,并将RN加1。

(3) 在接收到A的任何一个正确帧后,在一个有限的时间内,将收端的RN发给A。

在上述算法中,发端第(3)步是说明如何对帧进行编号,第(4)步是说明接收到应答如何处理,第(5)步说明帧如何正常传输和如何进行重传。收端第(2)步说明收到帧是如何处理的,第(3)步说明如何进行应答。

上述算法实际上是一个框架,它是指一类算法,其定时、操作顺序可以根据

需要来安排。例如,在重发 SN_{min}后,可以按顺序发送 $SN_{min}+1 \rightarrow SN_{max}-1$ 的帧,也可以根据重发过程中收到应答的情况,仅发送需重发的帧。又例如,在发送完 $SN_{min} \rightarrow SN_{max}-1$ 的帧后,立即返回再次发送 $SN_{min} \rightarrow SN_{max}-1$ 的帧,而不是等到应答定时器超时后再重发。

该算法的正确性证明类似于停等式 ARQ,其稳妥性证明完全相同,其活动性证明只要证明 $t_1 < t_3 < \infty$,$t_2 < t_3 < \infty$即可。其中,t_1 是 A 节点第一次发送 $SN_{min}=i$ *帧的时刻*,t_2 是 B 收到无差错的帧 i 并提交高层的时刻,t_3 是 SN_{min}增加至大于 i 的时刻。

返回 n-ARQ 的序号也可以用模为 m(m>*n*)*的整数来表示。例如,取模* 8 则可用 3 比特来表示序号,此时最大的窗口取值 n 只能为 7。如果 $n=m$,*则* 系统无法正常工作。其原因如下:假设 $n=m=8$,发端发送 8 帧后,收到了对方的所有确认,则将发送新的 8 帧,其序号为 0~7。如果发端发送 8 帧后,收端发送的应答未能到达发端,发端将重发这 8 帧,其序号仍为 0~7。由于这两种情况对收端而言是无法区分的,因而在接收到第二次序号为 0~7 的帧时,收端无法区分是新的帧还是重发的帧。

在序号取模 m(mod*m*)*的情况,返回* *n*-ARQ 的算法如下:(m>*n*)

发端 (A 节点) 的算法:

(1) 置模 m 变量 SN_{min}和 SN_{max}等于 0。

(2) 算法以任意顺序重复执行第 (3)、(4)、(5) 步。在每步的条件满足时刻到该步被执行的时刻之间的时延是任意的,但该时延是一个有限的值。

(3) 如果 $(SN_{max}-SN_{min}) \bmod m < n$,*且上层有一个分组到达,则将* SN_{max}指定给运载该分组的帧,并将 SN_{max}增加至 $(SN_{max}+1) \bmod m$。

(4) 如果接收的 RN 满足 $(RN-SN_{min}) \bmod m \leq (SN_{max}-SN_{min}) \bmod m$,*则* 置 $SN_{min}=RN$。

(5) 如果 $SN_{min} \neq SN_{max}$,且当前没有帧在传输,选择一个满足 $(SN-SN_{min}) \bmod m < (SN_{max}-SN_{min}) \bmod m$ *的* SN 帧进行传输。当 SN_{min}不再改变时,SN_{min}帧的重传间隔应当小于一个规定的有限值。

收端 (B 节点) 的算法:

(1) 置模 m 的变量 RN=0。重复执行第 (2) 和 (3) 步。

(2) 当接收到的 SN=RN,将分组呈送给高层,并将 RN 增加至 $(RN+1) \bmod m$。

(3) 在接收到 A 的任何一个正确帧后,在一个有限时间内,将收端的 RN 发给 A。

下面简要考察窗口长度 n 对返回 n-ARQ 效率的影响。

前面已提到导致返回 n-ARQ 效率下降 (重传或等待应答) 有三个方面的

原因。第一个原因是反向帧长过长,这就要求增加 n。增加 n 以后,只要 n 个正向传输帧长之和大于反向帧长的概率大大提高,就可以有效地减缓反向帧长过长带来的影响。第二个原因是反向应答出错,这也要求增加 n。增加 n 以后,出错的应答帧被后来应答帧所补救的概率增大,从而可以降低出错对效率的影响。第三个原因是正向传输出错。如果根据前两个原因的要求增加 n,这将导致正向传输出错后,反向应答到达发端的时延较长时需重传的帧数大大增加,这反而会导致系统效率下降。解决第三个问题的办法就是加快出错的反馈速度。即收端一旦接收到一个错误帧,立即返回一个短的应答帧(监控帧),使发端尽快返回重发。

为了分析的简单起见,假定数据帧长是一个固定值,且假定应答帧传输时间很小可以忽略。下面来讨论返回 n-ARQ 的效率。

返回 n-ARQ 的效率与链路的传输时延(T_P)、帧长(T_D)、窗口 n 等参数紧密相关,如图 2-12 所示。(图中:$d = T_D + 2T_P$)

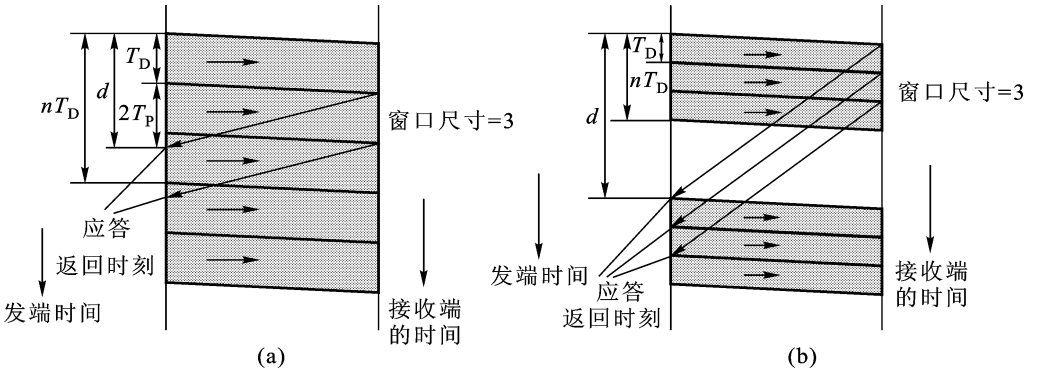


图 2-12 d、T_D、n 的相互关系

从图中可以看出,当 $nT_D > d$ 时,*应答帧可以及时返回,在传输帧无差错的*情况,链路的最大平均利用率为 1。发端可连续不断地向链路上发送帧。当 $nT_D < d$ *时链路的最大平均利用率为* nT_D/d。即在 $d = T_D + 2T_P$ 的时间内,发端最多可以发送 n 个帧。

设传输过程中的误帧率为 p,每出现一个错帧,发端就会重复 n 帧。若一帧经过 i 次传输成功(i=1 是第一次传输,i>1 表示重传),则表明经过了(i-1)次返回后该帧才传输成功,而每次返回需要传输 n 帧,因而总的所需传输的帧数为 1+(i-1)n,其出现的概率为 $p^{i-1}(1-p)$。*由此可得:成功传输一帧平*均所需传输的帧数 N_f 为

$$N_f = \sum_{i=1}^{\infty} [1 + (i-1)n] p^{i-1}(1-p) = 1 + \frac{np}{1-p} \tag{2-19}$$

利用该式可得返回 n－ARQ 方式中链路最大平均利用率为

$$U = \begin{cases} \dfrac{1}{N_f}, & nT \geq d \\ \dfrac{nTd}{N_f d}, & nT_D < d \end{cases} = \begin{cases} \dfrac{1}{N_f}, & n \geq (1+2a) \\ \dfrac{n}{N_f(1+2a)}, & n < (1+2a) \end{cases} \tag{2-20}$$

式中 $\alpha = \dfrac{T_P}{T_D}$，代入式 (2－19) 得

$$U = \begin{cases} \dfrac{1-p}{1+(n-1)p}, & n \geq (1+2a) \\ \dfrac{n(1-p)}{(1+2a)(1+(n-1)p)}, & n < (1+2a) \end{cases} \tag{2-21}$$

当 p=0.01 时，该式的结果如图 2－13 所示。从图中可以看出，链路利用率与相对传输时延 α、窗口 n 的大小紧密相关。当相对传输时延 α 相对较大 (如在卫星链路中) 时，为了达到较高的链路利用率，应选择较大的 n。从图中可以看出，当 n=「(1+2a)?（「x? 表示取大于等于 x 的最小整数) 时，链路利用率最高。也就是最佳的窗口宽度 (nT_D) 近似等于一帧的传输时间＋2 倍的传播时延。

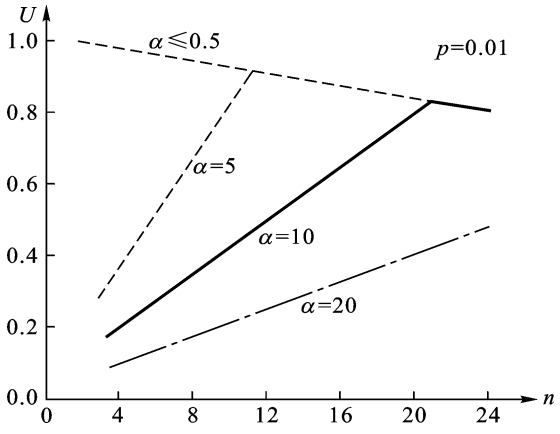


图 2－13 链路利用率与 n 的关系

3. 选择重发式 ARQ

选择重发式 ARQ (Selective Repeat ARQ) 是对返回 n－ARQ 的改进。

在返回 n－ARQ 中，如果前向传输的某一个帧出错，则在收到对方的否定应答后，该帧及其后续的帧都要重传，而不管这些后续是否传输正确。选择重发式 ARQ 的思路与返回 n－ARQ 相同，其窗口仍为 n，但仅仅重发有错的帧。

为了实现选择有错帧进行重发的目的，这就要求接收节点具有对分组排序的能力（因为这时接收到的分组是乱序的），并且在应答时除了应答 RN 以外，还要包括大于 RN 的那些帧已被正确接收的信息。接收节点可以用 RN＋k 个比特（每个比特的位置的取值对应于 RN 后面第 n 个帧的接收状态是 ACK 或 NAK）来进行应答。

当 RN 和 SN 采用模 m 来表示时，要求 $m \geq 2n$；*否则，如果取 $m > n$，会引*起接收数据的序号混淆。

选择重发式 ARQ 的链路利用率同样可用式 (2－20) 来表示，但这里重传的帧仅为出错的帧，即 N_f 可用式 (2－15) 来表示，$N_f = \dfrac{1}{1-p}$。从而可得选择重发式 ARQ 的链路利用率为

$$U = \begin{cases} 1-p, & n \geq 1+2a \\ \dfrac{n(1-p)}{(1+2a)}, & n < 1+2a \end{cases} \qquad (2-22)$$

4. ARPANET ARQ

ARPANET ARQ 采用了 8 个并行等待式 ARQ，如图 2－14 (a) 所示。每一个等待式 ARQ 对应一个虚拟信道，8 个虚拟信道分别为 A－H。输入分组可以任意分配到空闲的虚拟信道 A－H 上。如果所有虚拟信道忙，分组将在 DLC 层外等待。处于忙状态的虚拟信道上的分组被复接到物理比特管道上传输。可以采用轮询的方法来循环查询各个虚拟信道，当轮询到某一忙信道时，如果应答还没有收到，则将该虚拟信道的分组再次发送到物理信道上。因此，该复接方式就不需要设置定时器来计算等待应答的时间。

该协议的收发序号的表示方法如图 2－14 (b) 所示。由于采用 mod 2 方式，仅需一位就可以表示 SN 和 RN。为了区分不同的虚拟信道，在帧头中还增加了 3bit 的虚拟信道号。由于发送方式采用了轮询的方式，因而反向信道的应答也是重复发送的。因此，在反向信道的传输错误仅会引起很少的重发，而只有前向信道的传输错误时才需重发。图 2－14 (c) 中给出了仅有 A、B 两个信道时的传输过程。

使用该协议的一个问题 DLC 层不负责排序，因而高层应当具有分组排序功能。

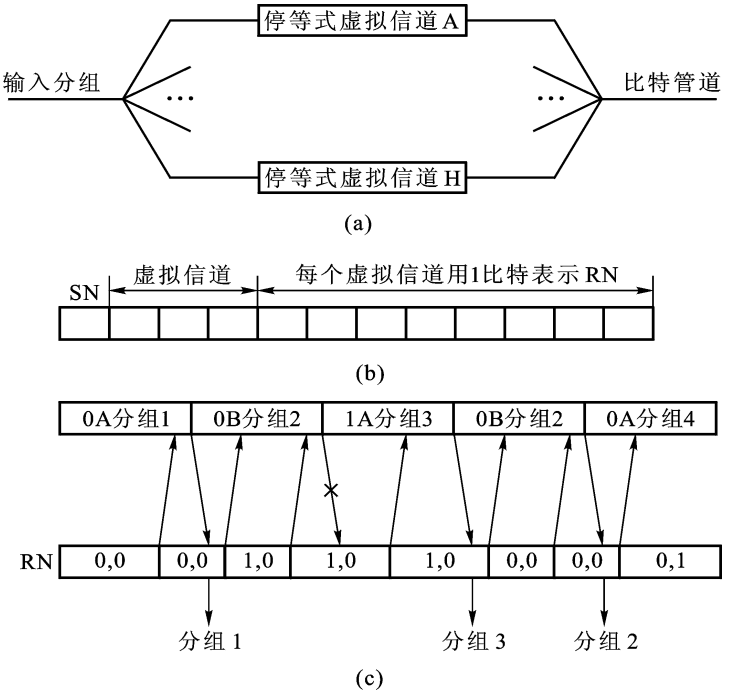


图 2-14 ARPANET ARQ

2.2.3 最佳帧长

我们需从两个方面来考察最佳帧长,一个方面是在一条链路上使传输效率最高的最佳帧长,另一个方面是在多条链路构成的传输路径上,使得传输效率最高的最佳帧长。

首先考察一条链路上的最佳帧长。

在实际传输过程中,每一帧数据(长为 l_f bit)通常包括 l_d bit 的数据负荷和 l_h bit 的控制信息(即 $l_f = l_d + l_h$)。如果帧长较短,控制比特所占用的比例较大,因而链路利用率下降。如果帧长较长,在数据帧传输过程中,因信道误码的存在而导致帧传输错误的概率较大,重传的次数将增大,这也会导致链路利用率的下降。因此存在一个最佳帧长,使链路利用率最高。

设链路的误比特率为 p_b,在随机错误的(如卫星信道)条件下,数据帧的差错率或误帧率 p 为

$$p = 1 - (1 - p_b)^{l_f} \tag{2-23}$$

当 p_b 很小时,上式可近似为

$$p \approx l_f p_b \ll 1 \tag{2-24}$$

以停等式 ARQ 为例,并利用式(2-16)可得链路的有效利用率为

$$U_e = \frac{l_d}{l_f}\frac{(1-l_f p_b)}{1+2\alpha} \tag{2-25}$$

式中 T_b 为比特宽度，$\alpha = \frac{T_P}{T_D} = \frac{T_P}{l_f T_b} = \frac{\alpha_0}{l_f}$，$\alpha_0 = \frac{T_P}{T_b}$。将 $l_f = l_d + l_h$ 代入并经整理得

$$U_e = \frac{(1-l_h p_b)\, l_d - p_b l_d^2}{l_d + (l_h + 2\alpha_0)} \tag{2-26}$$

该式对 l_d 求导，并令其为零，可得最佳数据帧长度为

$$l_d = \frac{4p_b^2 (l_h+2\alpha_0)^2 + 4p_b(1-l_h p_b)(l_h+2\alpha_0) - 2p_b(l_h+2\alpha_0)}{2p_b} \tag{2-27}$$

$$\approx \frac{(l_h+2\alpha_0)}{p_b} - (l_h+2\alpha_0)$$

例如：当 $p_b = 10^{-4}$，$l_h = 48$ bit，$2t_p = 100$ ms，$\frac{1}{T_b} = 4.8$ kbps，则 $2\alpha_0 = 480$，$l_d \approx 1770$ bit。

下面讨论在分组经过多次中转才能到达目的节点时，能使得网络开销最小和时延最小情况下的最佳帧长。

一条消息分成不同长度的分组经过中转到达目的节点的过程如图 2－15 所示。从图中可以看出，图 (b) 中分组的传输时延为 2 倍的分组长度。在图 (c) 中将分组的长度减少一半，则此时的时延仅为原来分组长度的 1.5 倍。因此从降低时延的角度，分组的长度应尽可能小。下面再看分组长度较小会带来其他什么问题。

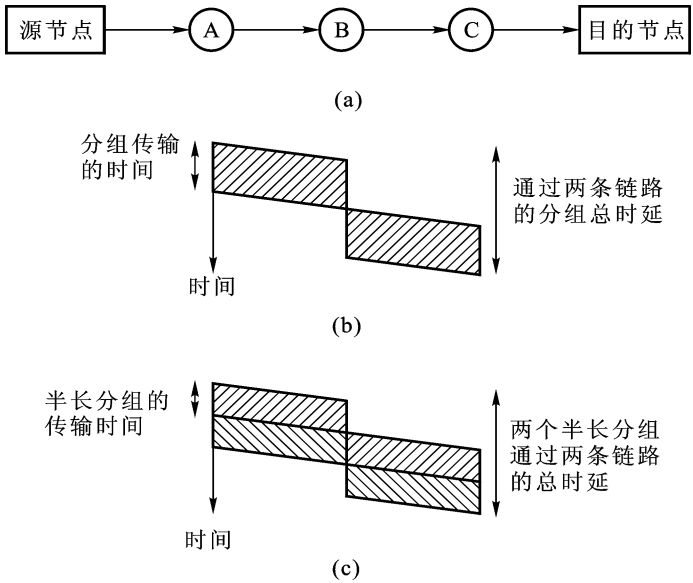


图 2－15 分组的中转过程

(a) 网络举例 (b) 分组中转时延 (c) 分组长度减半时的中转时延

设消息的长度为 M,分组的长度为 K,通常每一帧都包含固定的开销 V(含头和尾),这样每一条消息要分成 $\mathrm{Int}\left[\frac{M}{K}\right]^{+}$ 分组,$\mathrm{Int}[x]^{+}$ 表示大于或等于 x 的最小整数。在消息的传输过程中,前 $\mathrm{Int}\left[\frac{M}{K}\right]$ 个分组均有 K 个比特,而最后一个分组的比特数在1到 K 之间。一条消息要传输的总比特数为 $M+\mathrm{Int}\left[\frac{M}{K}\right]^{+}\times V$。

帧长 K 减小,会导致帧数增加,这会引起传输开销增加和网络处理负荷增加,因此应当增加帧长。在 $M\to\infty$ 时,开销所占的比例为 $\frac{V}{V+K}$。因此,增加帧长会降低开销。

综合考虑时延和开销两个方面,就存在一个最佳帧长。

设每条链路的容量为 C(b/s),将一条消息经过中继传到目的节点的总时间为 T,在忽略各节点的处理和缓存时延的情况下有:

T=消息在最后一条链路上的传输时间+(j−1)条链路引起的时延

$$T=\frac{M+\mathrm{Int}\left[\frac{M}{K}\right]^{+}V+(j-1)(K+V)}{C} \tag{2-28}$$

对 M 求均值,得

$$E\{T\}=\frac{1}{C}(K+V)(j-1)+E\{M\}+E\,\mathrm{Int}\left[\frac{M}{K}\right]^{+}V \tag{2-29}$$

用 $E\,\mathrm{Int}\left[\frac{M}{K}\right]^{+}\approx E\left[\frac{M}{K}\right]+\frac{1}{2}$ 代入上式,并使得上式最小的最佳帧长为

$$K_{\mathrm{opt}}\approx\sqrt{\frac{E\{M\}V}{j-1}} \tag{2-30}$$

在上面的讨论中未讨论打包的时延,即只关心消息进入系统后到达目的节点的时延。而对于某些流型业务(如语音),将要关心给定的某一个比特进入网络到该比特离开网络的时延,这时就必须考虑打包的时延。

设输入的比特速率为 R,分组长度为 K,收发之间各链路的容量分别为 C_1,C_2,…(均大于 R),分组的开销为 V 个比特,则一个比特的时延为

$$T=\text{打包时延}+\text{各链路的传输时延}=\frac{K}{R}+\sum_i\frac{K+V}{C_i} \tag{2-31}$$

从上式可以看出,当链路速率 C_i 提高后,T 主要由 $\frac{K}{R}$ 决定。例如:对于 64 kb/s 的数字语音,通常要求打包时延小于 10 ms,因此 K 通常取 500 bit 或更小。

上面都是从单个用户的角度来讨论网络处于轻负荷状态下,最佳帧长如何选取的问题。但是对于网络而言,各个用户的最佳长度各不相同。这就会出现

当不同长度的分组在一条链路上传输时,长度很长的分组传输时延很大,从而阻碍了短分组的快速传递。这就像一条单行线上,一辆慢速行驶的货车,阻塞了后面小车的快速行驶。这种现象称为"show trunk effect"(慢速货车效应)。因此,最佳的帧长度应当由网络来设定,而不是由各终端自行设计。

上面讨论的系统中,帧长是可变的。另外一种方式就是采用固定帧长,典型的是 ATM 信元,其长度为 53 字节。采用固定帧长的优点是便于硬件实现,适应于不同类型的业务种类。

2.3 标准数据链路控制协议及其初始化

前面 2 节研究了数据链路如何进行有效地帧传输及出错如何处理,本节将讨论如何在实际的协议中应用这些结果。

2.3.1 标准的数据链路控制协议

目前常用的标准数据链路控制(DLC)协议有 IBM 提出的 SDLC,ISO 建议的 HDLC,ANSI 规定的 ADCCP 和 CCITT 建议的 LAPB 等。其中,HDLC 与 ADCCP 功能相同,SDLC 是 HDLC 的一个功能子集。LAPB 也是 HDLC 的一个子集。

HDLC(ADCCP)是为多种物理链路设计的。这些链路包括多址链路、点对点链路、全双工和半双工链路[1]。它包括三种工作模式:正常响应模式(NRM)、异步响应模式(ARM)和异步平衡模式(ABM)。

正常响应模式(NRM)用于主从式链路。即链路的一端是主站(节点),另一端是从站。主站负责控制和协调双方的通信过程。典型的应用场合是一个计算机与多个外设之间的链路。采用轮询(polling)机制,实现主站与从站之间的通信。

异步响应模式(ARM)也是采用主从模式,但对从站没有严格的限制,该方式未被广泛使用,后面将不再讨论。

异步平衡模式(ABM)用于全双工点对点的链路,链路两端的节点具有相同的责任进行链路控制。这是应用最广泛的协议之一。

LAPB 仅使用 ABM 模式,SDLC 使用 NRM 和 ARM 模式。

上述四种协议及三种工作方式都使用如图 2-16 所示的帧格式。

---

[1] 通信双方链路的工作方式通常分为三类:全双工、半双工和单工方式。单工方式指通信双方仅有单向数据流动,一方为发端,另一方为收端,链路上仅为一条单向物理信道;全双工是指通信双方可以同时向对方发送和接收对方的数据流,链路上同时有两条不同流向的物理信道;半双工通信指通信双方可以向对方发送或从对方接收数据流,但不可以同时收发,任一方在发送时不能接收,在接收时不能发送。

8	8	8	?	16	8
Flag	地址	控制	分组	CRC	Flag

图 2－16 标准 DLC 的帧结构

图 2－16 中采用面向比特的帧结构。Flag＝$01^{6}0$，采用比特插 0 的技术来消除帧中可能出现的与 Flag 相同的比特串。CRC 校验采用 CRC－CCITT，其生成多项式 $D^{16}+D^{12}+D^{5}+1$。在具体计算 CRC 校验码时，将地址域和控制域取反后与分组数据一起参与 CRC 计算，将求得的余数取反后，形成 CRC 比特（或者说在传输时将 CRC 比特取反后进行传输）。这种实现 CRC 的方法，可以避免全零的序列满足 CRC 校验的可能性，也可以避免在一帧的头或尾增加或删去几个零而满足 CRC 校验的可能性。

地址域在常用情况下为一个字节（8 bit），它用于多用户共享一条链路时区分不同的节点。在不同的工作方式下，地址域的功能可以不同。例如：在 NRM 方式中，地址域总是从站的地址。当用于点对点通信时，地址域没有作用。

控制域用来区分不同的帧类型，其格式如图 2－17 所示。它有三种格式：信息帧（I）、监控制（S）和无编号帧（U）。信息帧采用模 8 的返回 *n*－ARQ 方式进行传输，它对应的控制包括 SN 和 RN。监控帧用于在无数据传输时返回 ARQ 的信息（如：RN）或加速返回 ACK 和 NAK 信息。无编号帧用于链路的建立和终止以及附加信息的传输。

	1	2	3	4	5	6	7	8
信息帧	0	SN			P/F	RN		
监控帧	1	0	类型		P/F	RN		
无编号帧	1	1	类型		P/F	类型		

图 2－17 控制域的格式

控制信息种类的区分是靠控制域的第 1 和 2 比特来区分的。第 1 比特为 0 表示为信息帧，第 1 和 2 比特为“10”表示监控帧，第 1 和 2 比特为“11”表示为无编号帧。控制域中的第 5 个比特为查询/结束（P/F）比特，在查询时称为 P 比特，在响应时称为 F 比特。在不同的工作模式中，P/F 比特的用法不同。例如：在 NRM 模式中，主站查询从站时，置 P＝1。从站响应如果有多个帧，则应将最后一个响应帧 F 置 1，其余各响应帧置 F＝0。在 ABM 方式中，任一站均可主动发送 S 和 I 帧，此时将 P 置 1，对方收到 P＝1 的帧后，将 F 置 1 进行应答。

监控帧（S）有四种类型：RR——接收准备好，RNR——接收未准备好，

REJ——拒绝,SREJ——选择拒绝。它们通过控制域中的第 3 和 4 个比特来区分。“00”对应 RR,“10”对应 RNR,“01”对应 REJ,“11”对应 SREJ。

RR 是正常的响应帧,它用于反向没有数据帧时,携带 RN 对发端进行应答。RNR 是用于收端暂不能接收更多帧(缓冲区满)时给对方的提示,RNR 中也包括 RN。REJ 用于刚接收的帧出错或者帧的序号与期望的序号不符时,给发端的应答。发端将重发序号为 RN 及其后面的分组,同时表明 RN 以前的分组已正确接收。REJ 可以改善返回 n－ARQ 的效率。SREJ 是一种简单的选择重发方式,它要求发端重发序号为 RN 的分组。

无编号帧(U)用于链路的建立、拆除和特殊控制。无编号帧是采用第 3,4,6,7,8 个比特来区分不同类型的。目前,只定义了 15 种无编号帧。无编号帧可用于设置工作模式,如置正常响应模式 SNRM(Set NRM),置异步平衡模式 SABM(Set ABM)等。例如:节点 A 和 B 之间有一条链路,当 A 发出 SNRM 后,B 用无编号帧 UA(Unnumbered Acknowledgment)予以响应。当对方通信结束后,用无编号帧发出拆线 DISC(DISConnect)命令。

下面举例说明数据链路控制(DLC)协议的工作过程。

为了简化描述,采用下列命令格式

X(Y)Z

其中:X 表示地址;Y 表示信息帧的 SN 和 RN,或监控帧的类型及参数,或无编号帧的类型;Z 表示 P/F 位是否置位。

例 2.3　NRM(正常响应模式)的工作过程。

设 A 是主站,B、C 是两个从站,如图 2－18 所示。

图 2－18 中的主要步骤:A 先与 B 建立连接,再与 C 建立连接,A 与 B、C 传输数据分组的过程以及 A 与 B 拆除连接的过程。具体步骤如下:

A 首先发出 B(SNRM)P 的命令(即接收地址为 B,置正常响应模式,置位查询位)来初始化从 A 到 B 的链路。B 收到该命令后,采用 UA 帧进行应答,其应答帧的格式为 B(UA)F。接着 A 以同样的方式来初始化 A 到 C 的链路。假定 A 到 C 的第一次命令 C(SNRM)P 传输出错,A 经过一定的等待时延后,重新进行初始化。当 A 成功初始化 A 到 B 及 C 的链路后(双方的 SN＝0,RN＝0),A 向 B 发送数据,其格式为 B(SN,RN)。A 发出的帧为:B(0,0),B(1,0)和 B(2,0)P。B(2,0)P 表示 A 已传输结束,询问 B 是否有数据。B 收到此格式的数据后,开始发送自己的数据,其格式为 B(SN,RN)。B 发出的帧为:B(0,1)和 B(1,1)F。B 置位 F 表示数传结束。然后 A 查询 C,并与 C 之间进行数据传输。

从图中可以看到,A→B 及 B→A 之间的数据传输出现了错误。A 在收到 B(1,1)F 的帧后,发现 B 希望接收的帧序号为 RN＝1,因而 A 将重发 SN＝1 及后续(SN＝2)帧。当 B 收到 A 重发的 B(2,0)P 后,发现对方未能收到自己 SN＝0

的帧，因而重发 SN＝0 及 1 号帧。此时 B 又有新的帧到达，可接着发送该 SN＝2 的帧，并告知对方传输结束。假定 A 决定结束 A 到 B 之间的连接，则 A 向对方 B 发出 RNR 帧，并同时应答对方 B 已发送的帧；此外，置位 P 还要求 B 对此命令帧进行应答，此帧的帧格式为 B(RNR,3) P。A 得到对方的确认（收到 B(RR,3) F 帧）后，A 发出拆除连接的命令，命令格式为 B(DISC) P，并要求对方应答，格式为 B(UA) F。当 A 收到对方的拆线应答后，A 到 B 之间的连接正式断开。

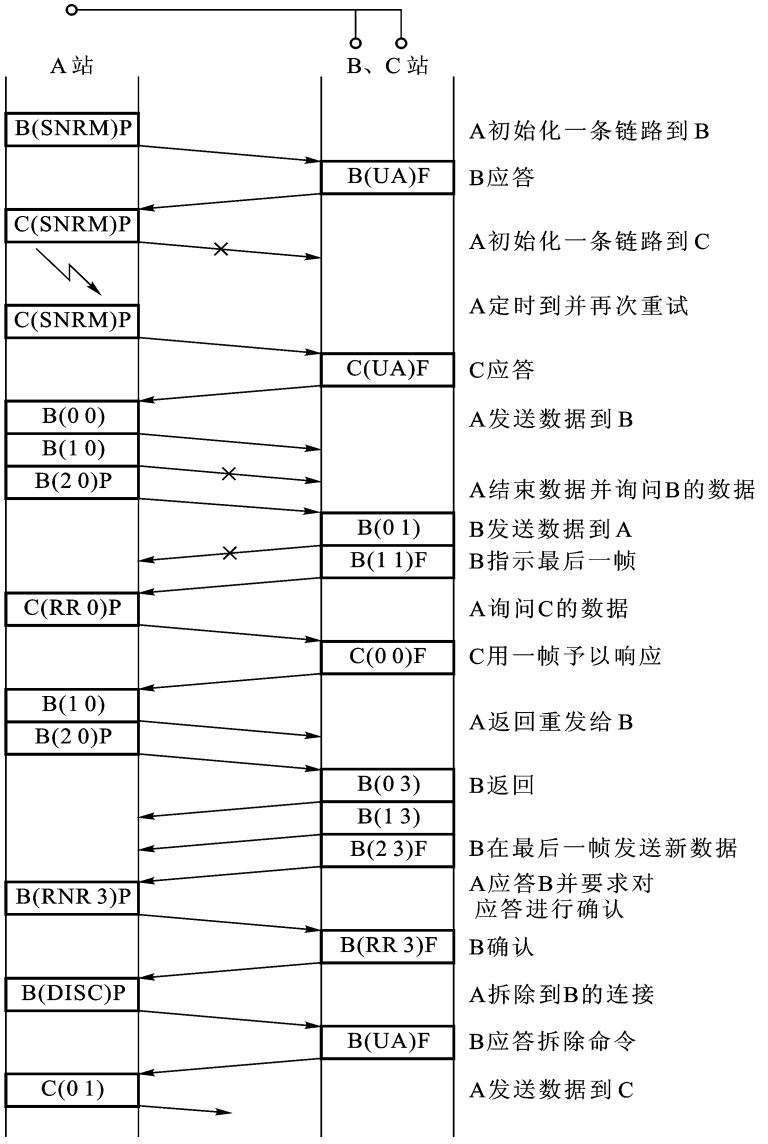


图 2－18 NRM 的工作过程

注意在图2-18中,地址域总是从站的地址,主站(A)在发出命令和数据时,使用从站(B或C)的地址,从站(B或C)用自己的地址予以响应或传输数据。

例2.4 ABM(异步平衡模式)的工作过程。

ABM的工作过程如图2-19所示。

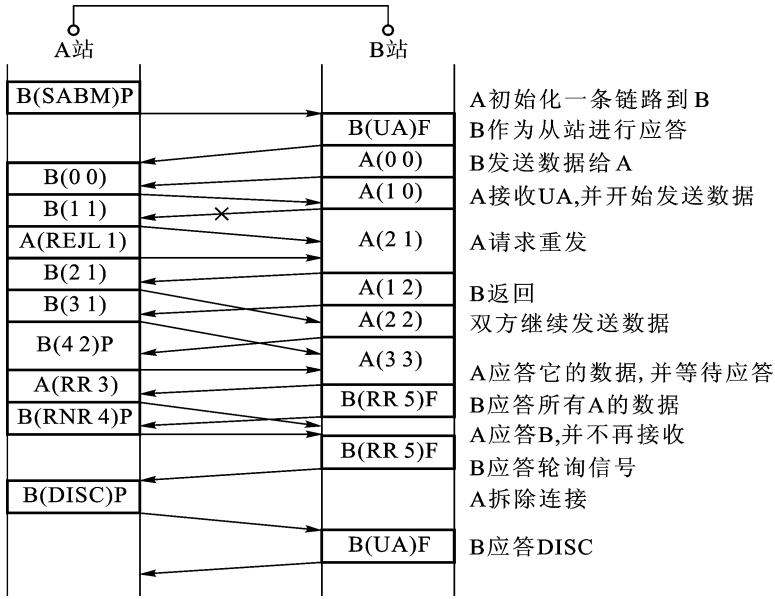


图2-19 ABM的工作过程

在ABM中,每一个节点既可以作为主站,也可以作为从站。当一个节点发送需要对方应答的所有信息帧和所有无编号帧时,它作为主站。此时发送的帧中使用对方的地址,P/F比特为轮询(P)比特。当一个节点发送运载应答或响应的所有无编号帧和发送应答主站的大多数监控帧时,它作为从站。此时在发送的帧中使用自己的地址,P/F比特为结束(F)比特。

图2-19分为三个主要的过程:A与B建立连接,A和B同时传输数据,A拆除连接。在这些过程中,任何一方发送数据和要求对方应答时使用对方的地址,应答时使用自己的地址。值得注意的是拆线的过程。A首先发送RNR监控帧[B(RNR,4)P],表示不再接收对方的分组。在该帧中包括了A已接收到的B的分组序号(RN),从而可使B明确A已接收到那些分组。B通过RR帧[B(RR,5)F]予以应答,并同时包括B已接收到的A的分组(RN),从而可使A明确B已接收到的分组。这样双方对链路状态有一个明确的认识。此时A再发出拆线命令,B收到后予以应答。到此通信过程结束。

2.3.2 数据链路层协议的初始化

通信双方对使用的通信协议进行初始化是通信过程中的基本问题,不仅在

链路层，而且在网络层、传输层及其他许多协议中都需要初始化。如果在通信过程，无异常情况（如节点或链路无故障），则网络的初始化是比较简单的。在前面的 ARQ 协议的讨论过程中，假定通信双方都已正确的初始化，即线路上无分组在传输，双方的 SN 和 RN 均为 0。但是如果有链路故障或节点故障（或因系统掉电后重新起动）存在，初始化问题是比较复杂的。

以链路故障为例，当链路出现故障一段时间后，为了保证端到端的传输可靠性，网络层或传输层通常会采取一定的措施，另外选择一条新的链路来传输在旧链路上未传送的分组。当旧链路恢复工作以后，高层会在该链路中建立一条新的通路来传输新的分组流。链路正常工作时称为 UP 状态。链路故障时称为 DOWN 状态。因此在一条链路上会出现 UP 和 DOWN 状态交替出现的情况，如图 2－20 所示。

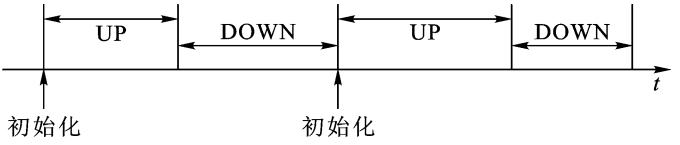


图 2－20 链路的等效工作状态

数据链路控制协议（DLC）需在每一个 UP 状态开始点进行初始化。在每个 UP 周期的末尾会出现一些分组已进入 DLC 进行传输，但并没有被对方接收和提交给高层。因此，DLC 的正常工作应当能保证在每一个 UP 周期，接收端提交给高层的分组流是对方（发端）从高层接收到的分组流中的最前头一段的分组流。

当链路状态 UP 和 DOWN 交替时，要使 DLC 能正常工作，就必须使双方对当前链路的状态有一致的看法。下面讨论不同工作模式的初始化过程。

1. 主从模式下的链路初始化

在主从模式（与 NRM 相对应）下，简化的链路初始化的过程如图 2－21 所示。为了讨论简便起见，采用 mod 2 的停等式 ARQ 协议。

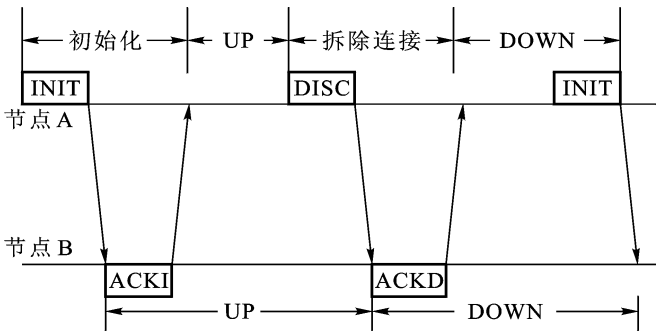


图 2－21 主从协议的初始化过程

图2-21中,A节点为主站,B节点为从站。A决定何时进行建立连接和拆除连接。A首先发送初始化命令(INIT(SN=1))进行链路初始化,B收到INIT后用ACKI(RN=0)予以应答。只有当A收到B的应答后,初始化才能结束。A如果在规定的时间内未收到B的应答,A会重发INIT命令,直至收到B的应答。当A决定拆除链路时,A发拆线命令DISC(SN=0),B收到A的命令后,用ACKD予以应答。

从图2-21中可以看出,对于节点A来说,从节点A发出INIT命令到节点A收到ACKI时为止是初始化阶段。从A收到ACKI到A发送DISC命令时为止,A认为链路处于UP状态。从A发出DISC命令到A收到ACKD为止是拆线阶段。从A收到ACKD到A再次发送INIT命令为止,A认为链路处于DOWN状态。在DOWN状态下,A到B将不会传送任何数据。对于节点B来说,从B收到INIT命令到B收到的DISC命令为止,B认为链路处于UP状态。从B收到DISC命令到B再次收到INIT命令为止,B认为链路处于DOWN状态。在B认为链路处于DOWN状态时,B不会发送任何数据。根据停等式ARQ的证明,我们很简单地就可证明,链路可能正确地初始化。

在HDLC协议中,SNRM帧对应于上述协议中的INIT,DISC帧对应上述DISC,ACK帧对应于上述ACKI和ACKD。

由于在HDLC中,对SNRM和DISC都是采用相同的UA帧予以应答,因而无法区分对SNRM的应答还是对DISC的应答,因而可能导致不正确的操作(如分组丢失)。假定HDLC的初始化、数据传输和拆除连接的过程如图2-22所示。

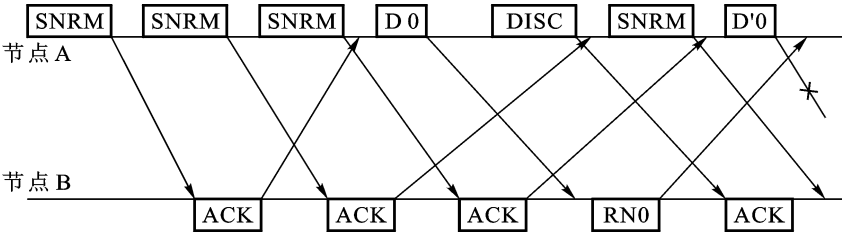


图2-22 HDLC的初始化、数据传输和拆除连接的过程

在2-22图中,假定传输时延大于发端等待应答的时延。在图中,A发送三次SNRM才收到对方B的应答。A收到B的应答后,发送数据序号为SN=0的分组D0。A在发送D0后等待一段时间后仍未收到对方的应答,则认为链路已不工作,进而发送DISC命令拆除连接。第二个ACK(对应于第二次SNRM的应答)到达A,使A认为是对DISC的应答,从而使A认为链路已拆除。A在一定的时延后,决定重新初始化该链路,发出SNRM命令。第三个ACK(对应于第三次SNRM的应答)使得A认为链路初始化结束,进而发送新的SN=0的

数据分组 D′0,此时对分组 D0 的应答 (RN=0) 到达 A,使 A 认为 D′0 已被正确接收。如果 D′0 传输出错,将导致 D′0 的丢失。该图从概念上说明由于 HDLC 应答机制的不完善,将有可能导致链路传输出错。因此在进行通信协议设计时,必须对初始化问题进行认真考虑,它可能使你设计的协议不能正常工作。

2. 平衡模式下的初始化

平衡模式中,通信双方是平等的,即当 A 站发送数据时,A 是主站,B 是从站;当 B 站发送数据时,B 是主站,A 是从站。因而这相当于有两个主从协议在工作。

平衡模式下的初始化过程如图 2-23 所示。该过程中,应答是嵌入在发送命令之中的。图中符号的含义与图 2-21 中符号的含义相同。链路 UP/DOWN 状态 (即 A→B 及 B→A 的状态) 是由 A 和 B 共同确定的。例如 B 在收到对方的 INIT 命令并发出初始化 INIT 命令后,只有收到对方 (A) 发来的 ACKI 时,才能认定链路为 UP 状态。又如,当 A 发出 DISC 后,只有在收到对方的 DISC 命令后,才能认定链路为 DOWN 状态。而在前面的主从协议中,链路的状态是由主站确定的。

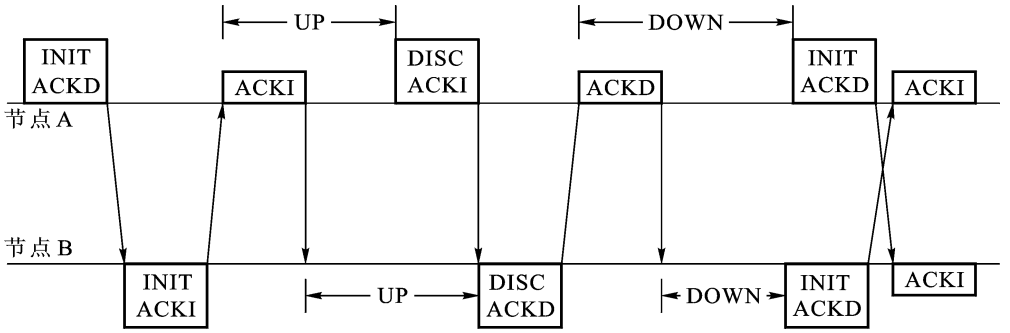


图 2-23 平衡模式下的初始化过程

3. 在有节点故障时的初始化

节点故障意味着所有与之相连的链路都出现故障。节点有故障时,不接受任何输入,也不产生任何输出,不发生任何操作。如果节点故障时能记忆其状态,则可以看成数据传输丢失,或高层已将数据改走其他链路,它与链路故障情况相同。当节点恢复工作时,所有相邻链路都需要进行初始化。如果节点故障时状态丢失,则此时的问题要复杂得多。

假定采用主从式初始化协议,节点故障时丢失其状态信息,每次节点从故障状态恢复时,都进行初始化。假定节点故障的时间和恢复工作的时间与来回传输时延属同一个量级。节点故障及其初始化的过程如图 2-24 所示。在该图中,A 节点多次出现短时间的故障状态,B 无法判定 A 是否故障。A 开始发送 INIT 后出现故障。A 从故障恢复后再次发送 INIT 进行初始化,并收到 B 的应

答(该应答是对第一次 INIT 的应答)后,A 认为对方已收到本次 INIT 命令,便开始发送数据分组 D0。A 再次出现故障,当 A 从故障再次恢复后,又发送 INIT 命令。A 在收到 B 的应答(该应答是对第二次 INIT 的应答)后,发送数据分组 D'0。当 A 收到对 D0 的应答后,会误认为是对 D'0 的应答,如果 D'0 传输出错,则会导致 D'0 的丢失。

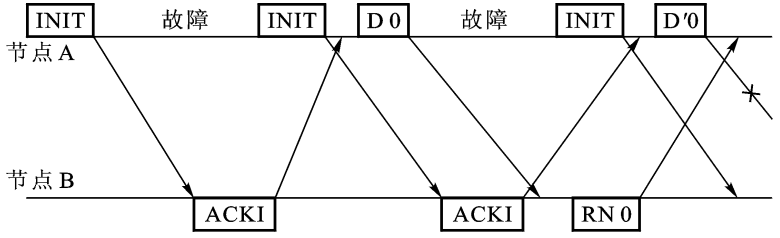


图 2-24 有节点故障情况下的初始化过程

在上面的讨论中假定链路的传播时延是没有上限的,因此,无论从什么状态开始以及采用何种协议,都有可能发生错误。

为了解决节点故障后的初始化问题,可以采用以下几种办法:

(1) 采用非易失性的存储器来保存链路的工作状态。

(2) 如果链路有一个最大的传播时延,则可以设计一个足够长的定时器,来避免上述初始化问题。

(3) 采用一个随机数的方法来区分不同正常运行期的操作,从而使得发生不正常操作的概率很小。

2.4 网络层和运输层的点对点传输协议

前一节研究了两个相邻节点的数据传输的协议,这里要讨论一个会话过程(session)跨越一个网络中多条链路,或跨越多个网络的不同传输链路时的分组编号、差错恢复、流量控制和编址等问题,以保证任意两个网络节点或两个应用进程之间可靠的数据传输。

2.4.1 网络层(子网层)的点对点传输协议

1. 会话过程和分组的编号

在 2.2 节讨论 ARQ 的帧格式时指出,为了使得数据帧能正确传输,必须对发送的数据帧和应答帧进行编号。每一帧运载的内容是网络层的一个分组。对于网络中的一条链路而言,它通常被通过该链路的若干个会话过程所共享。也就是说,不同会话过程的分组要共享同一链路。如果要将装载在物理帧中的分

组送达不同的目的地或区分来自不同源的分组,这就必须对不同会话过程的分组进行标识。标识分组的方法有两种。对于数据报方式,通常在分组头中应包括:源节点地址和目的节点的地址以及相同节点中不同会话过程的标识。利用这些信息,就可以将任一节点中的任一会话过程中的分组送到任一节点中相应的会话过程。对于虚电路(VC)方式,对于不同的会话过程,采用虚电路号进行标识,即每个分组含有一个虚电路号。在这两种标识方式中,数据报方式中的分组头开销较大;而虚电路方式中分组头的开销较小,但需要有虚电路的建立过程。在虚电路方式中,不同会话过程的区分方法如图2-25所示。图中节点3与7之间同时存在两个session(A和B),session A由链路(3,5)中的VC7,链路(5,8)中的VC4和链路(8,7)中的VC11组成;session B由链路(3,5)中的VC13,链路(5,8)中的VC7和链路(8,7)中的VC6组成。节点6与2之间有一个session C,它由链路(6,5),(5,8)和(8,2)中的VC3、VC3和VC7组成。

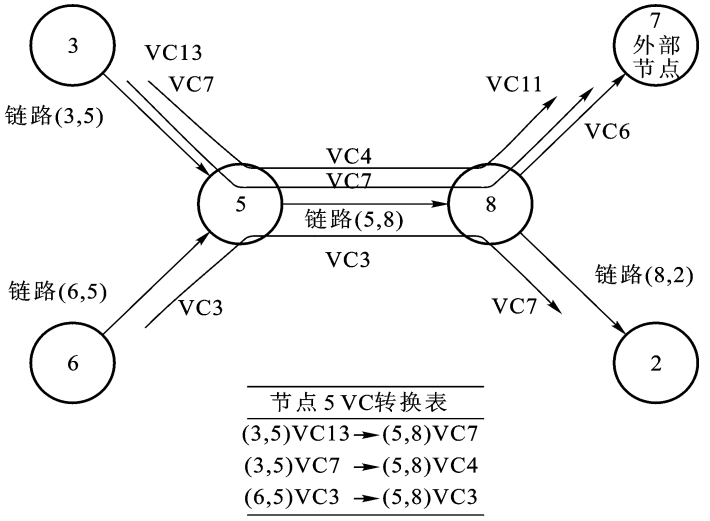


图2-25 不同会话过程的标识方法

不同session的分组可以在不同的帧中独立传输,也可以将多个session的分组复接在一帧中进行传输。不同session的分组复接在一帧中传输的示意图如图2-26所示。它包括帧头和各个session的分组及相应的session头。session头用于区分不同session的分组。

帧 头	session 1 头和分组	……	session *N* 头和分组

图2-26 不同session的分组复接在帧中传输

前面讨论了如何区分不同 session 的分组，那么对于同一 session 中的分组是否需要进行标识或编号呢？

在数据报方式中，同一 session 分组可能会经过不同的路径，这样到达目的节点的顺序就会不同于源节点发出分组的顺序，另外一方面分组在传输的过程中，因链路拥塞、传输错误、节点或链路故障等原因会引起分组丢失。因此，就必须提供一种方式来使目的节点发现上述问题。解决方法就是对同一 session 发送的分组进行编号。

在虚电路方式中，可能会有下列原因导致分组丢失或传输出错：

⑴ 虚电路号错误导致不正确的帧通过了 CRC 校验，而把不正确帧误认为是一个正确的帧；

⑵ 当数据分组中的传输错误未能被 CRC 检查出来；

⑶ 节点或链路故障可能导致部分分组丢失，如果没有分组编号，目的节点就不可以发现丢失的分组。

因此，在虚电路方式中，同样需要对同一 session 中发送的分组进行编号。

分组编号的大小可以采用 K 个比特表示，用 $\mathrm{mod}\ 2^K$ 的方式对分组进行循环编号。它可以表示一个 session 中的不同分组，也可以表示一个分组传输的第一个比特、第一个字符或第一个字（每个字由多个字节组成）在一个 session 中的编号。

2. 网络层的差错控制

网络层的差错控制方式与数据链路层的差错控制方式类似，采用 ARQ 方式，发端有发送序号 SN，收端应答有接收序号 RN。ARQ 的方式可为返回式 ARQ 或选择重发式 ARQ。

网络层的差错控制与数据链路层差错控制的主要差别在于：

⑴ 使用的位置不同。数据链路层的差错控制是用于一条物理链路的两端，而网络层的差错控制是用于网络中的任意两节点之间。通常网络中的任意两个节点之间的传输路径会由多条链路串联而成。

⑵ 分组编号的方式不同。在网络层是对一个 session 中的分组（或者字节，或者消息）进行统一编号。而在链路层上是对不同 session 中所有分组进行顺序编号。

⑶ 传输顺序的差别。在链路层，所有的帧都是按顺序传输的；而在网络层中，相同源和目的节点的分组可能会经过不同的路径，分组的传输可能会出现乱序现象。

⑷ 时延不同。在链路层，传输时延（包括传播时延、处理时延、帧传输的时延）在小范围内变化；而在网络层，传输时延会在大范围内变化。

当然在链路层和网络层都会出现帧和分组的丢失。

由于分组可能会经过不同的路径,会出现丢失、乱序和任意的时延,因而不能保证网络层(和运输层)的差错恢复100%正确。例如:假定分组要用mod m的编号方式,则序号为l mod m和$(l+m)$ mod m的分组在接收端是无法区分的。假定这里使用数据报方式,由于分组的时延和丢失,有时必须采用端到端的重传,即发端在一个给定的时间内没有收到接收端的应答,发端就会重发某一分组(设其序号为l mod m)。而该重发分组的前一个拷贝,可能并没有丢失,可能会因网络的任意时延而潜伏在网络之中。发端继续发送新的分组直至$(l+m)$ mod m分组时,如果潜伏在网络中的序号为l mod m的分组的一个拷贝先于序号为$(l+m)$ mod m分组到达接收节点将会导致不可纠正的错误。尽管这种情况出现的可能性很小,但它说明在网络层(或传输层)的差错控制不能确保正确地工作。

解决上述问题的办法有:

(1) 网络层最好采用虚电路方式;

(2) 分组编号的模值应足够大,使得上述错误出现的可能性足够地小,达到可以接受的范围;

(3) 给每个分组规定一个最大生存时间,在该区间内,使得分组的序号在使用mod m时不可能出现一个循环。

在后面讨论的TCP协议中,采用了后两种方法的组合。

前面已经提到,在链路层为了检测传输错误,使用了CRC校验序列;在网络层为了检测出子网中的传输错误和链路层中的未检测出来的错误,也需要使用某种形式的校验序列。校验序列可以是CRC校验,也可以采用简单的校验方法。如将发送信息的16 bit作为一个字,先求发送序列的每一个字的补码,再将这些补码相加求和,再求该和的补码作为校验序列。(该序列的长度为16 bit)。

在分组通过不同的节点时,通常需要对分组头作某些变动(如更改虚电路号),这样每个节点都要重新计算校验序列。这样既增加了节点的负荷,也使得校验失去意义。通常的做法是将分组头中的可变部分(如虚电路号)不参加校验,这样每个节点就不需要计算校验和。此外,为了防止分组头的出错,导致分组被错误地送到其他节点,使得该节点发生接收错误。可将源节点和目的节点共知的信息(如地址码等)作为信息比特的附加部分一起参与校验比特的计算(但这些信息不进行传输),从而可以防止目的节点出错。

3. 网络层的流量控制

前面讨论的ARQ协议(返回n-ARQ和选择重发式ARQ等)主要用于点对点的差错控制。这里讨论ARQ协议如何用于流量控制。

ARQ协议的基本机制是采用滑动窗口(窗口宽度为n)来限制在一个

session 中发送节点向网络发送的分组数。即第 j 个分组能够发送给网络的条件是第 $j-n$ 个分组已经被应答。如果网络发生拥塞或传输时延增加，则应答将被时延，这样信源的发送速率就会降低。因此，利用 ARQ 协议的这种特性，目的节点如想减缓接收分组的速率，则可以将含有 RN 的应答分组适当延迟后再发送即可。这种控制信源速率的方法称为端到端流量控制。

上述采用延迟应答来减小拥塞的方法，对于减少子网内部和目的节点处的拥塞都是非常有效的。这种减少拥塞的方法是将流控功能附在应答的传输过程中。但这也带来了一些限制，即发端无法区分下列三种情况。

(1) 如果由于传输出错，而导致应答未到达发端，发端会在超时后重发对应的分组。

(2) 如果应答因网络拥塞而被时延，则发端会在超时后重发。但实际上此时发端不应重发，否则会加重网络拥塞。

(3) 如果因为目的节点处于过载状态而导致应答被推迟发送，则发端也会在超时后重发，这样也会加重网络的拥塞。

解决上述问题的方法是设法将应答和流控的功能适当分开。一个思路是不减缓应答，而使发端减慢发送速度。具体做法是采用类似于 DLC 中 RNR (接收未准备好) 帧来应答对方，这样既应答了对方，又表示目的节点处于忙或拥塞的状态，从而达到了降速的目的。另一个思路是采用许可机制 (permit scheme)。具体做法是：目的节点向源节点发送两个反馈信息：RN + Permit (j)。其中，RN 属于常规的应答，许可证 (Permit) 通知目的节点还准备接收多少个分组，即允许源节点发送的分组序号为 RN 到 RN + $j-1$。该方法实际上改变了窗口的大小。

这里仅仅给出了拥塞控制的基本概念，后面将详细讨论拥塞控制的机制和性能。

4. X.25 网络层标准

X.25 标准是由 CCITT (现称为 ITU-T) 制定的外部设备 (称为数据终端设备——DTE) 到网络节点 (称为数据通信设备——DCE) 之间的标准接口。其物理层标准称为 X.21，其 DLC 层标准称为 LAPB。X.25 网络层标准有时称为分组层标准。

X.25 分组有两种：一种是数据分组，另一种是控制分组。其通用格式和数据分组格式如图 2-27 (a) 和 (b) 所示。其分组头由三个字节组成：第一个字节的高四位是总格式标识 (GFI)，第一个字节的低四位和第二个字节是虚信道号；第三个字节是分组类型标识。

X.25 数据分组的格式如图 2-27 (b) 所示。分组头中的第三个字节与 DLC 层的格式类似，RN 和 SN 用于表示一个 session (虚电路) 内的分组序号。其他控

制比特的含义如下：

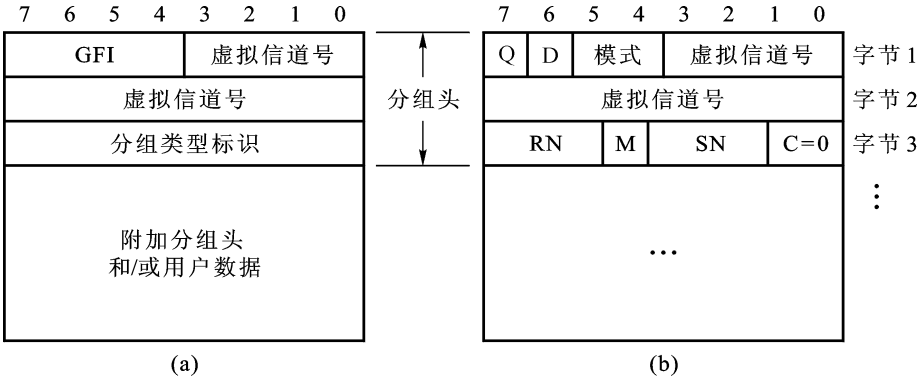


图2－27 X.25的分组格式

C比特用于区分是数据分组还是控制分组，C＝0表示数据分组，C＝1表示控制分组。M比特表示一个session中是否还有后续分组，M＝1表示该session还有后续分组，它不是最后一个分组；M＝0表示这是一个session中的最后一个分组。虚信道号比特域（12比特）用于表示该分组的虚信道号。模式域（2比特）表示SN和RN采用的模值大小，即表示模8还是模128（"01"表示模8，"10"表示模128）。如果采用mod 128，则RN和SN要扩展为7比特。D比特表示应答的类别。D＝1表示端到端应答，即目的节点到源节点的应答，它与RN相结合，表示目的节点已正确收到RN以前的所有分组；D＝0表示每条物理链路段上收节点对发节点的应答。由于在DLC层已有应答，因而采用RN的应答功能是多余的，RN仅用于流控。Q比特是业务类型指标，Q＝1表示是来自传输层和高层的控制分组；Q＝0表示是数据分组和网络层的控制分组。

X.25的控制分组可以分为6组：呼叫建立、流量控制、监视、证实、诊断和中断。

呼叫建立分组有四类：呼叫请求、输入请求、接受呼叫和呼叫连接。这些分组在虚电路的呼叫建立阶段使用。呼叫分组应包括源节点和目的节点的地址长度及地址、session中最大的分组长度、窗口大小、吞吐量协商、逻辑信道号分配、付费等信息。

流量控制分组包括接收准备好（RR）、接收未准备好（RNR）和拒绝接收（REJ）三个控制分组，它类似于DLC的监控帧。

监视分组包括重新启动请求/指示、清除请求/指示和复位请求/指示分组。

证实分组包括重启动证实、清除证实、复位证实和中断证实，用于确认前面请求的执行情况。

诊断分组用于诊断错误，指示分组被拒绝的原因。

X.25的分组交换过程如图2-28所示。它分为呼叫建立阶段、数据传输阶段和呼叫清除阶段，其数据传输过程类似于DLC层。呼叫建立阶段用于确定通信的逻辑通路和通信使用的参数以及初始化双方的工作状态。呼叫清除阶段用于拆除逻辑通路。实际系统的工作情况要比图2-28复杂。例如，当子网内部因为负载太重而不能建立虚电路时，或者目的节点不愿接受呼叫时，这时网络或目的节点将会向主叫节点发送呼叫清除请求分组，此次呼叫被拒绝。

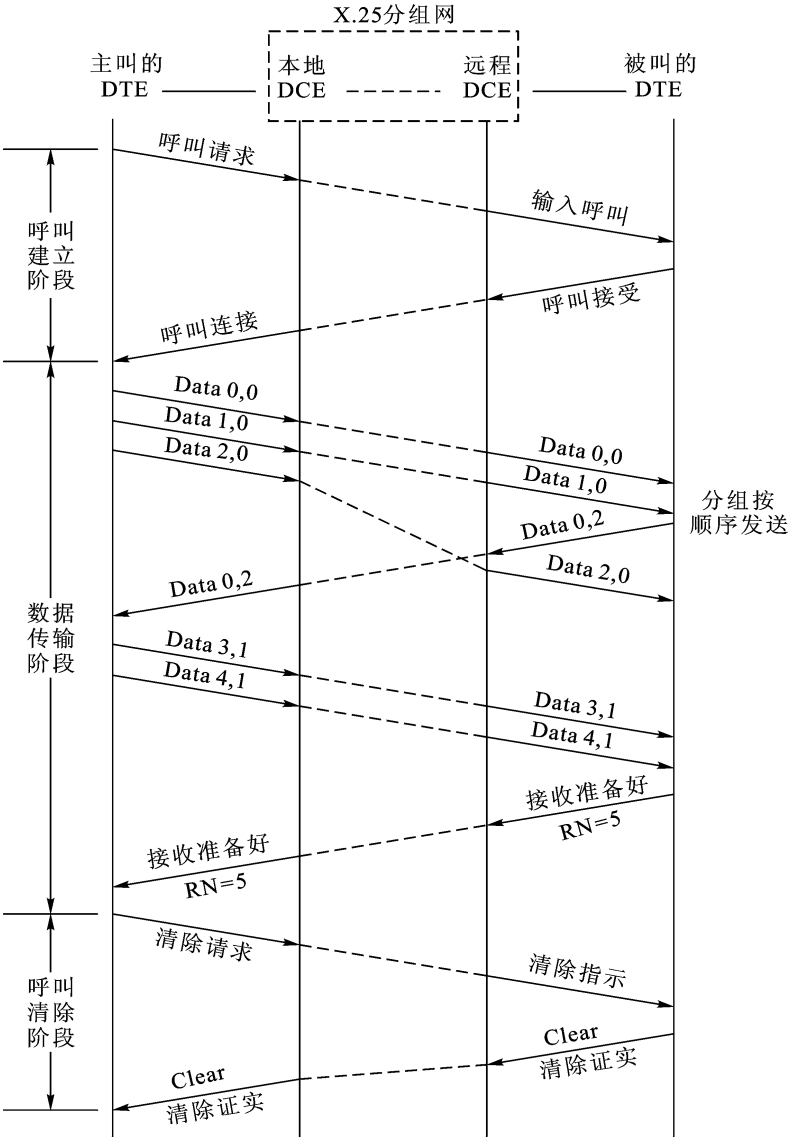


图2-28 X.25分组交换过程

2.4.2 网际层(互连层)的传输协议——IP协议

在第1章中已经看到,用户可以使用各种类型的接入网络,要实现任意两个用户之间的通信,就需要将这些网络互连起来。网络互连在一起进行通信,会遇到许多问题要解决,如:不同的寻址方案、不同的最大分组长度、不同的网络接入机制、不同的超时控制、不同的差错恢复方法、不同的状态报告方法、不同的路由选择技术、不同的用户接入控制、不同的服务方式(面向连接服务和无连接服务)、不同的管理与控制方式,等等。

目前全球最大的、开放的、由众多网络通过路由器互连而成的,采用TCP/IP协议族的网络称为因特网(Internet)。

Internet使用的网际协议称为IP(Internet Protocol)。

Internet的设计目标是使得所有的网络尽可能容易地加入到Internet中。尽管许多网络传输数据报的时延可能是任意的,数据报的传输可能会丢失、重复和乱序,但这些网络都允许加入到Internet中。由于Internet网假定实际的子网缺少可靠性的保障,所以IP采用了数据报协议。至于子网内部采用什么样的协议与IP无关。子网内部可以采用数据报方式,也可以采用虚电路方式来传输IP分组。(注:利用IP传输的一个数据单元称为数据报。一个数据报在子网内部可分为若干个分组来传输。)

IP的主要功能有:

(1) 为数据报通过Internet提供路由。

(2) 寻址功能。为源和目的节点提供地址信息。

(3) 将数据报分段、重装以适应不同的网络对分组长度的限制。

路由的功能将在5章讨论,这里讨论寻址和分段得装的功能。

IP协议有两个主要版本,IPv4和IPv6。IPv4中的地址长度为32 bit,IPv6中的地址长度为128 bit。IPv4的报头是可变长的,而在IPv6中报头是固定长度的。

IPv4的地址是分层次的,它由网络号+主机号两部分组成,分为五类。不同规模的网络采用不同类型的地址。常用的地址类型是A、B和C类(如图2-29所示);D类地址用于广播;E类地址保留,主要用于实验。地址的格式为W.X.Y.Z。A类地址的网络号为一个字节,用W表示,取值为0~126,其主机号用X.Y.Z表示,共有16 777 214个主机。B类地址的网络号为2个字节,用W.X表示,共有16 384个网络,其W的取值为128~191,主机用Y.Z表示,每个网有65 534个主机。C类地址的网络号为3个字节,用W.X.Y表示,共有2 097 152个网,其W的取值为192~223,主机用Z表示,每个网有254个主机。A类地址用于有很大量主机的网络,B类地址用于中、大规模的网络,C类地址

用于局域网。

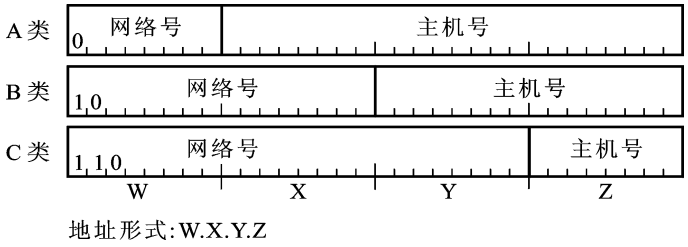


图 2-29 IPv4 的地址类型

在 IPv6 的地址长度为 128 位,采用十六进制表示,16 个比特作为一个基本单元,相互之间采用“:”相连。例如一个有效的 IPv6 的地址为:

4A3F:AE67:F240:56C4:3409:AE52:440F:1403

IPv6 的地址分为三类:单播地址(unicast address)、组播地址(multicast address)和任播地址(anycast address)。(单播就是点对点通信;组播是一点对多点通信,多个接收者构成一个组,组播将数据报发送到组内的每一位接收者;广播是组播的一种特殊形式,此时,接收者为所有网络成员;任播的目的节点也是一组接收者,但只将数据报交给该组中的一个成员,该成员通常是距发送节点最近的接收者。)

在 IP 中,数据报的原始长度主要是根据用户的方便来选择的,但不同的网络对分组的长度有不同的限制,因此要将一个数据报分成若干较小的段(fragment)。这些小的数据段在网络中自由地传输,到达目的节点后,再组装成 IP 分组上交给应用层。如果一个或多个数据段在传输过程中丢失,则该数据报中其他到达的数据段在一定时间后将被丢弃。整个数据段将由发端重发。尽管这种做法效率不高,但实现简单。如果要提高传输效率,网际层就必须采用差错恢复机制。

在 IP 分组中,分段有四个参量:

(1) 分段的起始位置[段偏移(fragment offset),以 8 个字节为单位]。

(2) 段长度(fragment length)。

(3) 表示当前段是否为数据报最后一段的标志(Flag)MF。MF=0 表示该段是数据报中的最后一段,MF=1 表示后面还有分段数据。与此相关的还有是否允许分段的标志 DF。DF=1 表示不允许分段,只有 DF=0 时才可以对数据报分段。

(4) 数据报的标识(ID)。

通过上述四个参数,就可以确定某一分段数据属于哪一个数据报以及它在数据流中所处的位置和长度,如图 2-30 所示。

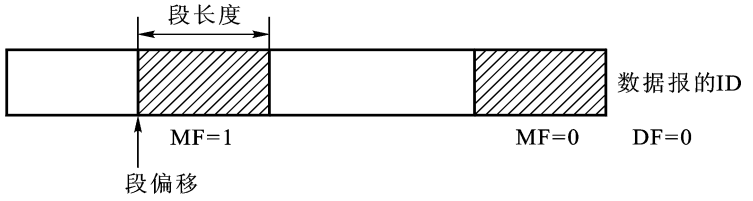


图 2-30 IP 分组的分段参数

根据上述讨论,构造的 IPv4 分组头如图 2-31 所示。

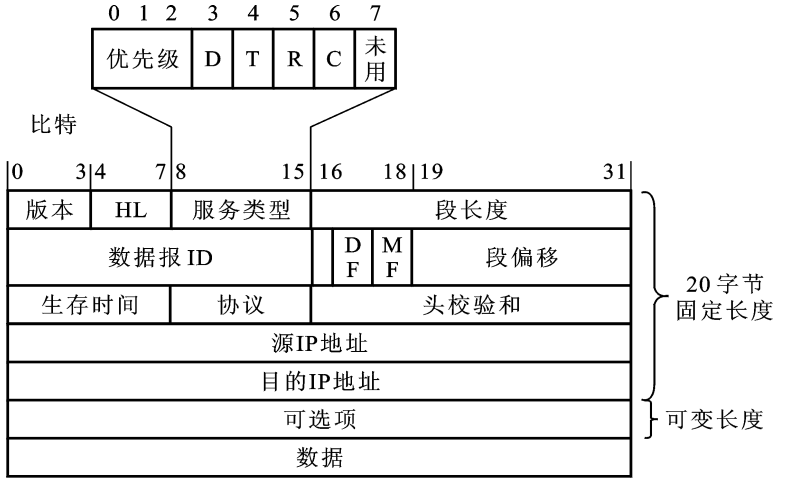


图 2-31 IP 数据报的格式 (IPv4)

在图 2-31 中,版本域用来表示 IP 的版本 (IPv4 或 IPv6) 。HL 表示分组头的长度,单位是 4 个字节。服务类型标明用户想得到的业务类型,其中优先级占三个比特,D 比特表示要求有更低的时延,T 比特表示要求更高的吞吐量,R 比特表示要求更高的可靠性,C 比特表示要求选择费用更低廉的路由。生存时间 (TTL, Time To Live) 表示 IP 分组在网络中可生存的时间,每经过一个路由器 TTL 将被减 1;如果 TTL=0,则 IP 分组将被丢弃。协议指示数据报携带的运输层数据是使用何种协议,如 TCP (6) 、UDP (7) 、ICMP (1) 、GGP (8) 、IGP (9) 、OSPF (89) 等。头校验和用于对头部的检验。校验的规则是将 IP 头看成 16 bit 字的序列,先将校验和置零,将前 5 个 16 bit 字相加后,将和的二进制反码写入头校验和字段。

与 IP 协议一起工作的还有三个协议:地址解析协议 ARP (Address Resolution Protocol) ,逆地址解析协议 RARP (Reverse Address Resolution Protocol) 和 Internet 控制报文协议 ICMP (Internet Control Message Protocol) 。

ARP 用于 IP 地址到子网设备 (如网卡) 地址的转换。在一个子网中是依靠

子网内的地址实现节点之间的相互通信,因此要将 IP 分组从一个子网内的节点传到另一个子网内的节点,就必须将源 IP 地址和目的 IP 地址转换为子网内相应的源节点地址和目的节点地址。由于用户的 IP 地址和子网内设备的地址都有可能更换,因此,ARP 要完成 IP 地址到子网内地址的动态转换。

RARP 用于确知自己的子网地址的情况下,获取自己的 IP 地址。

ICMP 允许主机或路由器报告 IP 数据报传输的差错情况和提供有关异常情况的报告。ICMP 报文可分为两类:一类是 ICMP 差错报文,另一类是 ICMP 询问报文。常用的 ICMP 差错报文有:通知主机改变路由的 ICMP 改变路由(redirect)报文;目的主机出现拥塞时向源节点发送的 ICMP 源站抑制报文,该报文使源站暂停发送报文;以及目的节点不可达时报告的报文。而常用的 ICMP 查询报文有 ICMP Echo 请求和应答报文(常用于两个主机间的连通性测试);ICMP 时间戳(timestamp)请求和应答报文;ICMP 地址掩码(address mask)请求和应答报文等。

尽管 IP 数据报的传输不保证不丢失,但与 ICMP 相配合,可以减少数据报的丢失。

2.4.3 运输层的点对点传输协议

运输层的功能是将消息分成分组。如果网络层没有合适的差错控制,则运输层提供必要的差错恢复。如果网络层没有流控,则运输层进行流量控制。对 session 进行复接和分接。目前运输层有两个典型的标准协议:TCP 和 OSI TP Class0～TP Class4。这里以 TCP 为例进行介绍。

1. TCP 中的寻址和复接

通常在 TCP 之上有很多的用户(或进程),为了区分这些用户(或进程),就需要对它们进行编址。TCP 中将 TCP 之上的每一个用户(或进程)称为一个端口(port)(用 16 比特表示)。一些常用的 TCP 端口为:SMTP(25),FTP(21),TELNET(23);一些常用的 UDP 端为:RPC(111),SNMP(161)和 TFTP(69)。因此,在 TCP 中,一个完整的地址应当由三部分组成:网络号、主机号、端口号,它们被称为一个套接口(socket)。在 TCP 中,仅将端口号包括在 TCP 头中,而以参数的形式将网络号和主机号(IP 地址)告知 IP 层,IP 层将 IP 地址放入 IP 头中。对于具有相同源和目的节点的不同端口的消息将复接在 IP 分组中传输。而 IP 本身并不关心 TCP 提交的信息内容。

通过本章的讨论已经看出:寻址问题是通信网络中一个非常重要的问题。不同层有不同层的寻址方式。以 TCP/IP over X.25 网为例,在 TCP 层有 16 bit 长的端口地址,在 IP 层有 32 bit 长的 IP 地址,在 X.25 网层有 12 bit 的虚电路号,在链路层 LAPB 有 8 bit 长的地址。每一层都有相应的头。有时这些头中的

信息有些多余,但是,对于任一个通信网络来说,子网层头必须足以用来完成子网内的路由和流控,IP 层的头加上子网层的头应足以用来完成网关(或路由器)之间的路由,而运输层的头要足以区分复接在一起的不同session。

2. TCP 中的差错控制

TCP 中采用的差错控制方式为选择重发式 ARQ。SN 和 RN 的长度为 32 bit。这里 SN 和 RN 不是分组的编号,而是对数据字节进行编号。SN 表示所传输的分组数据是从当前 session 中第 SN 个字节开始的一段数据。RN 表示希望接收到的分组的第一个字节应为当前 session 中第 RN 个字节。如果当前的 SN = m,数据长度为 n,则下一分组中的 SN = $m+n$。

在 TCP 中,差错恢复主要解决两方面的问题,一是重传问题,二是连接建立和拆除时错误。

重传的问题主要是如何确定重传的间隔(time out)。这主要是由于 TCP 报文可能会经过不同类型不同速率的网络,因而传输时延的方差很大。如果重发间隔过小,则会导致很多报文过早地重发,给网络增加了不应有的负荷。若重发间隔过大,则网络的传输效率会降低很多。

TCP 中采用了一种自适应算法来确定重发间隔。该算法采用下式来确定报文的平均往返时延 T(往返时延指报文发送时刻到发端接收到相应应答的时刻之间的时延。)。

$$T = \alpha \cdot T_{old} + (1-\alpha) \cdot T_{new} \tag{2-32}$$

式中,T_{old}表示旧的往返时延,T_{new}表示新测得的往返时延,$0 \leq \alpha < 1$。α 表示新旧往返时延对平均往返时延 T 的影响程度。典型的 α 值是 7/8。

重发间隔应略大于 T,即

$$\text{重发间隔} = \beta \cdot T \tag{2-33}$$

式中,β 是一个大于 1 的系数。实际上,系数 β 是很难确定的。它必须在传输效率和增加的网络负荷之间取得平衡。原 TCP 中的推荐值为 β=2。该方法的难点在于发端对第一次发送的分组和重发的分组未加任何区分,收端发出的应答也是未加区分的。因此,当发端发出重发组分组后,由于网络的随机时延,它将无法确定所收到的应答是对哪一次发送的分组(第一次还是重发过程的某一分组),这样导致往返时延的不确定性。为了解决该问题,若分组已进行重发,就不再计算平均往返时延,即仅计算一次发送成功并收到应答的报文平均往返时延,并以此来计算重发间隔。而每重发一次就将重发间隔增加一次,即

$$\text{重发间隔} = \gamma \cdot \text{旧的重发间隔} \tag{2-34}$$

在系统中,γ 的典型值为 2。

在连接建立和拆除过程中,可能会出现如下一些情况:第一次连接拆除后仍有分组到达,这就可能导致在第二次连接建立后,会有第一次连接的分组到达。

建立连接的分组和拆除连接的分组可能混淆。当网络一个节点出现故障,无法跟踪某一连接时,而该连接仍存在于另一节点中,这也将导致混淆。

TCP 中为了解决上述问题,在每次连接开始时都进行初始化,使得 SN 和 RN 同步。具体实现的方法如下:源节点和目的节点都有一个 32 bit的时钟计数器,每 4 μs 增加一次(节点间的时钟计数器不同步)。当要建立连接时,发端(A 节点)以本地的时钟计数器的值作为初始阶段的第一命令分组的序号(SN)。收端(B 节点)在响应该指令时,对发端的序号进行响应,即将 B 节点 RN′置为 SN+1 发给对方(A 节点),B 采用的初始化 SN′为其本地的时钟计数值,B 在收到对方(A 节点)的应答前不能传送数据。如图 2-32 所示。

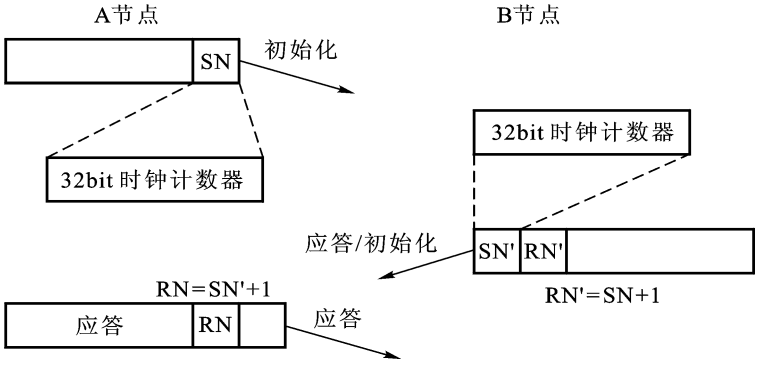


图 2-32 TCP 的初始化过程

采用这种随机地选择序号的方法,可以避免新旧连接的序号混淆。32 bit 的计数器循环一圈大约需 4.6 小时。

根据上面的讨论,可得 TCP 的报文格式,如图 2-33 所示。

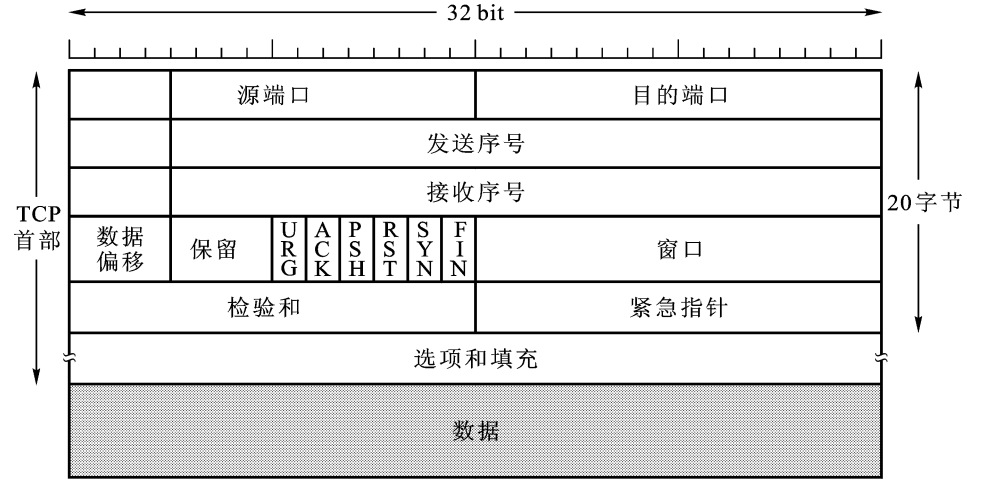


图 2-33 TCP 的报文格式

在图2－33中：源端口和目的端口是运输层向高层提供的服务接口，采用16 bit表示。数据偏移占4 bit，表示在该报文中数据开始点离TCP报文段起始点的距离，它实际上是TCP报文头的长度。控制域用6个比特表示，它用于建立和释放连接、应答和报文提交方式等动作的控制。其中：URG比特＝1表示此报文应尽快发送，它与紧急指针配合使用，指明紧急数据的长度；ACK比特＝1表示确认序号字段有意义；PSH比特＝1表示请求远端TCP将本报文段立即传送给其应用层；RST比特＝1表示要重新建立连接；SYN比特和ACK比特组合使用用于表示发送建立连接请求和应答，SYN比特＝1及ACK比特＝0表示建立连接的请求报文，SYN比特＝1及ACK比特＝1表示同意建立连接的应答报文；FIN比特＝1表示要释放一个连接。窗口占两个字节，表示收窗口的大小，即告诉对方在未收到应答前可发送的最大数据字节长度。校验和是对报文头的校验。紧急指针指明在报文段中，紧急数据的最后一个字节的序号。

3. TCP的流量控制

流量控制需要考虑两方面的问题：一是接收者的容量缓冲区大小，二是网络的容量及通过量。假定网络的容量和通过量较大而接收者的缓冲区相对较小，这时如果发端发送的业务量较大，就会导致接收缓冲区溢出而引起报文丢失。如果网络已经发生拥塞，通过量已很低，这时如果发端仍保持较高的发送速率，就会进一步加剧网络的拥塞，降低通过量，从而导致发端的报文丢失。

为了解决接收者缓冲区溢出问题，在TCP中采用了窗口允许机制。TCP报文格式中，采用了16 bit的窗口域，该窗口称为通知窗口（advertised window），它用来通知发端在未收到应答以前可以发送的最多字节数，即发端可以发送序号从RN到RN加窗口值之间的数据字节。

在网络传输过程中，有两种原因可能导致分组丢弃：一是传输出错导致分组被丢弃，二是因为网络拥塞引起的分组丢弃。由于发端很难区分这两种情况，所以Internet中的TCP协议假定分组丢弃都是由于网络拥塞引起的。分组的丢弃将导致发端超时重发。

为了控制网络的拥塞，在TCP中还引入了第二控制窗口——拥塞窗口（即发端一次可发送的字节数）。拥塞窗口长度是根据网络的拥塞情况动态调整的。发端真正使用的窗口（称为发送窗口）长度等于通知窗口和拥塞窗口的最小值，即发送窗口长度＝min（通知窗口长度，拥塞窗口长度）。

在TCP中采用了慢启动（slow－start）、拥塞避免（congestion avoidance）和加速递减等技术来进行拥塞控制。发端每传送一次（即发端将发送窗口中的报文段全部发完，并且收到了对该报文段的所有确认）调整一次发送窗口。为了实现控制过程，在TCP中引入了一个门限窗口。当拥塞窗口大于门限窗口时，发端要减缓发送速度，以避免网络拥塞。具体的拥塞控制算法举例如下。

设初始状态是通知窗口长度为 64 kB (千字节) ,门限窗口长度为 64 kB,拥塞窗口长度为64 kB。假定初始时就发生了传输超时,算法将第 0 次传输时拥塞窗口调整为 1 kB (最大的分段长度) ,门限窗口长度减少到拥塞窗口的一半为 32 kB。第 0 次传输后,每传送一次,发送窗口长度增加一倍 (即按指数增加窗口长度) ,当发送窗口长度等于门限窗口长度 (第 5 次) 后,每传送一次,窗口仅线性增加一个最大的分段长度。如果再次发生超时 (第 13 次传输) (拥塞) ,则下次传输时将门限窗口长度减少到拥塞时 (第 13 次) 窗口的一半即为 20 kB,并将拥塞窗口长度调整为 1 kB,如图 2－34 所示。

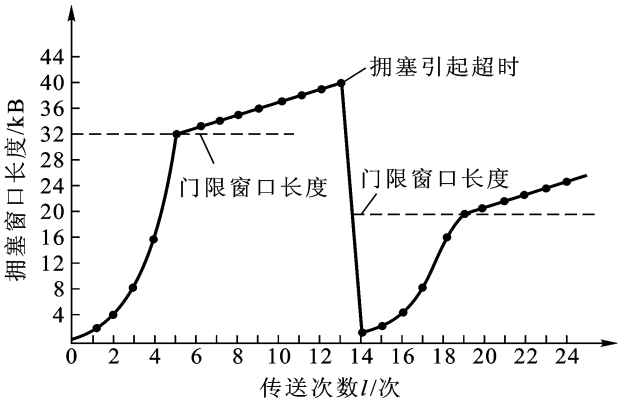


图 2－34　TCP的拥塞控制算法

图 2－34 中拥塞窗口长度从 1 kB 增加到门限窗口长度的过程为慢启动过程,即每次拥塞后,拥塞窗口长度都降为 1 kB 使报文慢慢注入网络。从门限窗口开始,拥塞窗口线性增加到最大值的过程称为拥塞避免过程。将前面慢启动过程中指数增加变为线性增加是为了避免网络再次出现阻塞。当网络再次出现阻塞,采用加速递减的方法,将门限窗口减少到拥塞时发送窗口的一半。

采用上述流量控制方法使得 TCP 的性能明显改善。但是当 TCP 用于无线信道时,由于经常会因传输错误而导致发端超时,这时如果仍采用上述流控方法,将会引起 TCP 的传输效率下降,此时需要对 TCP 进行适当的改进。

小　结

本章主要讨论了端对端的传输协议,它涉及到两种类型链路的可靠数据传输;一是在一条物理链路上如何进行有效和可靠的数据传输,二是针对跨越多条物理链路或多个网络的一条等效的链路之间如何进行可靠的数据传输。它要解决的基本问题是:

⑴ 将不可靠的链路变成一条可靠的链路；

⑵ 如何保证在各种异常情况下，收发双方协同工作；

⑶ 如何进行流量控制。

首先，针对一条物理链路，如何组成帧（framing）和确定最佳帧长，实际中常用 Flag 和帧长等方法来标识一帧的开始，讨论了如何进行检错（采用 CRC 等方法）；出错以后如何进行反馈重发（ARQ），采用的方法有停等式 ARQ，返回 n－ARQ 和选择重发式 ARQ；通过标准的 DLC 协议综合描述了前面各项技术的具体应用；还讨论了在正常和异常情况下，如何进行链路的初始化。

接着讨论了网络层的端对端传输协议，主要包括差错恢复、流量控制、虚电路传输协议以及 IP 协议等。

最后，讨论了运输层的端对端传输协议，主要包括 TCP 协议的寻址和复接、差错恢复和流量控制等。

习 题

2.1 常用的组帧方式有哪几种？哪一种方式的传输开销最小？

2.2 接收机收到了如下一个采用十六进制表示的字符串，C0 C0 10 36 87 DB DC DB DC DC DD DB DD C0 7C 8D DC DB DC C0，试根据 SLIP 帧格式恢复出接收的帧。

2.3 针对输入序列 01101111100111111010111111111101111010 应用 2.1.2 节的比特插入技术，给出相应的输出结果。如果接收到的序列为

01111110111111011001111100111110111111011000111111010111110

试移去插入的比特并指出 Flag 的位置。

2.4 假定 2.1.2 节帧中插“0”的规则修改为：仅在原始数据中出现 01^5 时插入一个 0。试详细描述这种变化后接收端去 0 的规则，并说明该规则是如何去掉下列比特串中的 0：01101111101111101111010111111110。（如果设计的规则正确，在该串中应去掉 2 个 0 并且仅有一个 Flag。）

2.5 设有一个奇偶校验码由 3 个数据比特和 4 个校验比特组成。假定 3 个码字分别为 1001011，0101101 和 0011110。试求产生该码的运算规则，并写出所有 8 个码字。并求出该码的最小距离（两个相同长度的码字之间的距离定义为两个码字比特取值不同的位置数）。

2.6 令 $g(D)=D^3+D^2+D+1$，$S(D)=D^3+D+1$，求 $\dfrac{D^4 S(D)}{g(D)}$ 的余数。

2.7 对于一个给定的 L 阶生成多项式 $g(D)$ 和一个给定的数据比特长度 K，假定输入序列除第 i 位为 1 以外，全部为 0，即 $S(D)=D^i$，$0\leq i\leq K-1$，其

对应的 CRC 结果为：$C^{(i)}(D) = C_{L-1}^{(i)} D^{L-1} + \cdots + C_1^{(i)} D + C_0^{(i)}$，试证明：

(1) 对于一个任意的数据多项式 $S(D)$，其 CRC 多项式 $C(D) = \sum_{i=0}^{k-1} S_i C^{(i)}(D)$；

(2) 令 $C(D) = C_{L-1} D^{L-1} + C_{L-2} D^{L-2} + \cdots + C_1 D + C_0$，则 $C_j = \sum_{i=0}^{K-1} S_i C_j^{(i)}$，$0 \le j < L$。上式说明每一个 C_j 是一个奇偶校验比特。也就是说，CRC 校验码也是一种奇偶校验码。

2.8　假定改变停等式 ARQ 的传输策略，在连续发送的分组中，不使用序号，而是改为：发送的 DLC 发送给定分组已被重传的次数，即帧格式为

j	分组	CRC

，这里 j 为重传的次数（$j=0$ 为分组第一次发送），接收 DLC 对每一个接收的帧采用 ACK 或 NAK 予以应答（应答中没有请求的序号）。试通过举例证明无论接收端 DLC 采用什么规则来接收分组，该传输策略都不能正确工作。

2.9　试证明停等式 ARQ 的序号可采用模 2 表示。

2.10　在停等式 ARQ 中，设重发分组之间的间隔为 T_r（包括分组传输时间、传播时延、等待应答时间和处理时延等），分组正确接收的概率为 ρ，试证明最大的可传送的分组到达率 $\lambda_{max} = \frac{\rho}{T_r}$。

2.11　在相同的帧长和相同的帧错误的情况，重画图 2-11(a) 和 (c)，考察分组从节点 B 到 A 的传输情况，即说明 SN 和节点 B 的窗口，以及 RN 和输出给 A 的分组。

2.12　设从节点 A 向节点 B 传输分组，令 γ 是节点 B 每成功接收一个分组，A 到 B 所发送帧的平均次数。令 p 是到达 B 的帧出错的概率（连续的帧相互独立）。假定 A 总是忙于发送帧，窗口长度 n 足够地大，在没有反馈的情况下，A 绝不会返回重发，但在听到一个等待应答的帧有错时，A 总会在下一帧返回重发。令 β 是一个给定帧从其传输开始到接收到与该帧对应的反馈帧（包括反馈帧到达时 A 正在传输的那一帧）之间从 A 到 B 平均发送的帧数。证明 γ 满足 $\gamma = 1 + p(\beta + \gamma)$。定义效率 η 为 $1/\gamma$，求出 η 与 β、p 的函数。

2.13　一条双向对称无误码的传输链路，链路传输速率为 64 kbps，单向传播时延为 15ms。设数据帧长为 3200 bits，确认帧长度为 128 bits，采用停等式 ARQ 协议，忽略处理时延。问：(1) 在仅有单向数据传输业务的情况下，在 820 s 内最多可以传输多少个数据帧？(2) 如果双向都有业务传输，且应答帧的传输只能跟在返向数据帧的尾部（格式为：

数据帧	应答帧

），问在 820 s 内每一个方

向最多可以传输多少个数据帧？ ⑶ 若采用返回 n－ARQ 且 $n=3$，重新计算 ⑴ 和 ⑵ 的结果。

2 14 在 2 2 2 节介绍的四种 ARQ 方式中，链路利用率与哪些参数有关？哪一种方式的链路利用率最高，请定量说明。

2 15 试画出 K_{opt} 与 $E(M)$，V 及 j 的关系曲线，并对该结果进行讨论。

2 16 HDLC 中是如何保证数据透明传输的？HDLC 有几种工作模式？

2 17 试解释例 2 4 中图 2－19 的详细工作过程。

2 18 ⑴ 假定采用 2 3 2 节的平衡初始化和拆除连接协议，节点 A 和 B 都认为它们之间的链路处于 UP 状态。假定在时刻 t 节点 A 开始拆除连接（即发送 DISC）。试证：在采用与 2 2 2 节相同的假定条件下，每个节点最终都会认为链路处于 DOWN 状态。

⑵ 假定节点 A 认为链路处于 UP 状态，节点 B 正在进行链路初始化，但还没有从 A 收到 ACKI。若 A 开始拆除连接，试证明 B 最终认为链路为 UP 状态。然后 B 开始拆除连接，在此以后，A 和 B 最终都会认为链路为 DOWN 状态。

2 19 ⑴ 在主从式初始化协议中，假定一个从节点故障恢复的节点开始处于 UP 状态，通过先发 DISC 再发 INIT 来进行初始化，举出一个类似于图 2－24 中不正确工作的例子。

⑵ 现假定一个任意的协议用于初始化。已知从故障中恢复的节点 A 将发送一条消息 X，节点 B 从故障中恢复并收到消息 X 后将发送消息 Y，在这些条件下当节点 A 收到 Y 时，节点 A 将能被正确初始化。试构造一个故障和时延序列说明该协议不能正常工作。

2 20 一个通信子网内部采用虚电路方式，沿虚电路共有 n 个节点交换机，在交换机中为每一个方向设有一个缓冲区，可存放一个分组。在交换机之间采用停止等待协议，并采用以下的措施进行拥塞控制。节点交换机在收到分组后再发回确认，但条件是：⑴ 接收端已成功地收到该分组；⑵ 有空闲的缓冲区。设发送一个分组需 T(s)（数据或确认），传输的差错可忽略不计，用户（DTE）和节点交换机（DCE）之间的数据传输时延也可忽略不计。试问：分组交付给目的用户（DTE）的速率最快是多少？

2 21 有 AB 和 BC 两条链路。A 经过 B 向 C 发送数据。B 收到 A 发来的数据时，可以先向 C 转发再向 A 发确认，也可以把这顺序反过来。也就是说，B 要做的三件事的顺序是：收数据→转发→发确认，或：收数据→发确认→转发。现假定 B 在做完第二件事后处理机即现故障，内存中所存信息全部丢失，但很快又恢复了工作。试证明：只有采用端到端发确认信息的方法（即从 C 向 A 发确认信息），才能保证在任何情况下数据都能从 A 经 B 正确无误地交付到 C。

2 22 两个用户（U_1 和 U_2）通过他们的主机 H（DTE）同 X 25 网建立了虚

电路连接。图 2－35 的时序图表示分组到达网络层的过程。这里 P_{ij}为从第 i 个用户（$i=1,2$）来的第 j个分组。网络层将虚信道号 VC 与发送序号 P(S)（即网络层的 SN）插入到网络层分组头中。假设 U_1 的 VC 为 5，U_2 的 VC 为 17。而后所有分组采用多路复用方式发往数据链路层。数据链路层按顺序将发送序号 N(S)（即链路层的 SN）插入到帧头的其他参数中。画出分组在主机 H 与网络间的接口上传送的时序图。按顺序说明每一分组的 N(S)、VC、P(S) 的值。

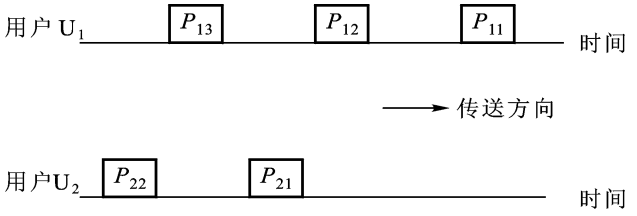


图 2－35　分组到达网络层的时序图

2 23　一个 TCP 连接使用 256 kb/s 链路，其端到端时延为 128 ms。经测试，发现吞吐量只有 120 kb/s。试问窗口是多少？

2 24　设 TCP 的拥塞窗口长度置为 18 kB。网络发生了超时，TCP 使用慢启动、加速递减和拥塞避免。设报文段的最大长度为 1 kB，试问：拥塞窗口从最小值经过 6 次变化后是多少？

2 25　网络层差错控制与数据链路层差错控制的主要差别是什么？

2 26　ARQ 协议用于差错控制和流量控制有何异同？

第3章 网络的时延分析

衡量网络传输能力的重要指标之一是将一个分组从源节点传到目的节点的时延。对时延的考虑将会影响网络算法和协议(如多址协议、路由算法、流控算法等)的选择。因此必须了解网络时延的特征和机制,以及网络时延取决于哪些网络特征。

网络中的时延通常包括四个部分:处理时延、排队时延、传输时延和传播时延。处理时延是指分组到达一个节点的输入端与该分组到达该节点输出端之间的时延。若节点的传输队列在节点的输出端,则排队时延是分组进入传输队列到该分组实际进入传输的时延。若节点的输入端有一个等待队列,则排队时延是指分组进入等待队列到分组进入节点进行处理的时延。传输时延是指发送节点在传输链路上开始发送分组的第一个比特至发完该分组的最后一个比特所需的时间。传播时延是指发送节点在传输链路上发送第一个比特的时刻至该比特到达接收节点的时延。

本章将讨论用于网络时延特性分析的主要定理和模型,包括 Little 定理、M/M/m 排队系统、M/G/1 及其推广型排队系统、排队系统的网络等内容。

3.1 Little 定理

排队是日常生活中最常见的现象,如去食堂排队就餐,去银行办理存取款等,可用图3-1的模型来描述一个排队的过程。描述该排队模型有三个方面:

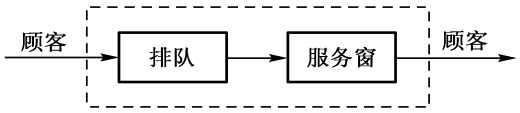


图3-1 排队模型

一是顾客到达的规则或行为。它由顾客到达的数目(可以是有限或无限)、到达间隔(可以是确定值或随机值)以及到达的方式(顾客是独立到达或是成批到达)等参数特征决定。

二是排队规则,即等待制还是损失制。等待制是指系统忙时,顾客在系统中等待。损失制是指顾客发现系统忙时,立即离开系统。典型的损失制系统就是日常使用的电话通信系统。当用户打电话时,发现系统忙(占线)时,立刻会放下

电话离开系统。

三是服务规则和服务时间。服务的规则可以是无窗口（如自选商场）、单窗口和多窗口。服务的时间可以是确定的，也可以是随机的。

在不同的传输网络中，顾客和服务时间可能是各不相同的。例如，在分组交换网络中，顾客即为分组，服务时间即为分组传输时间。在电路交换网络中，顾客即为呼叫，服务时间即为呼叫持续的时间。

在排队系统中，已知量有两个：一是顾客到达率（指单位时间内进入系统的平均顾客数，也称为单位时间内进入系统的“典型”顾客数，“典型”是指时间平均）；二是服务速率（指系统处于忙时单位时间内服务的典型（平均）顾客数）。要求解的量也有两个：一是系统中的平均顾客数（它是在等待队列中和正在接受服务的顾客数之和的平均数）；二是每个顾客的平均时延（即每个顾客等待所花的时间加上服务时间之和的平均值）。

现在讨论排队系统中的基本定理——Little定理。

3.1.1 Little定理

令 $N(t)$ = 系统在 t 时刻的顾客数，N_t 表示在 $[0,t]$ 时间内的平均顾客数，即

$$N_t = \frac{1}{t}\int_0^t N(t)\,\mathrm{d}t \tag{3-1}$$

系统稳态时的平均顾客数为

$$N = \lim_{t\to\infty} N_t$$

令 $\alpha(t)$ = 在 $[0,t]$ 内到达的顾客数，则在 $[0,t]$ 内的平均到达率为

$$\lambda_t = \frac{\alpha(t)}{t} \tag{3-2}$$

稳态平均到达率为

$$\lambda = \lim_{t\to\infty} \lambda_t$$

令 T_i = 第 i 个到达的顾客在系统内花费的时间（时延），则在 $[0,t]$ 内顾客的平均时延为

$$T_t = \frac{\sum_{i=0}^{\alpha(t)} T_i}{\alpha(t)} \tag{3-3}$$

稳态的顾客平均时延为

$$T = \lim_{t\to\infty} T_t$$

N、λ、T 的相互关系是

$$N = \lambda T \tag{3-4}$$

这就是 Little 定理(公式)。该公式表明:系统中的用户数(顾客数) = [用户(顾客)的平均到达率] × [用户(顾客)的平均时延]。

该公式与日常生活中的感性认识是一致的。例如,在相同的顾客到达率的情况下(λ相同),快餐店由于服务时间短(T较小),因而店中的用户数较少,仅需要较小的等待厅;而正常的饭店由于其服务时间长(T较大),店中的用户数较多,因而需要较大的服务厅。又如对于一个拥塞的系统,通常意味着有较大的用户时延。

上述讨论是针对时间平均的结论。对于统计平均有相同的结论,即

$$\overline{N} = \overline{\lambda} \cdot \overline{T} \tag{3-5}$$

式中,$\overline{N}$、$\overline{\lambda}$、$\overline{T}$ 分别表示系统中用户数、用户到达率和用户时延的统计平均值。

式(3-5)成立的基本要求就是系统具有各态历经性,并可以达到稳态。例如,令 $p_n(t)$ = 在 *t 时刻系统顾客数为 n 的概率。系统可以达到稳态的要求是*:$\lim_{t\to\infty} p_n(t) = p_n$, n=0,1,2,…。令 N(t) 统计平均为 $\overline{N(t)} = \sum_{n=0}^{\infty} np_n(t)$,则各态历经性的要求是指以概率1成立:$N = \lim_{t\to\infty} N_t = \lim_{t\to\infty} \overline{N(t)} = \overline{N}$。

下面以一个特例来说明 Little 定理的正确性。如图3-2所示,设系统的初始状态为 N(0) =0,在 t 时刻到达系统的总用户数为 α(t),离开系统的总用户数为 β(t),则停留在系统中的用户数为 $N(t) = \alpha(t) - \beta(t)$。*设用户 i 在系统*中停留时间为 T_i,则从图中可以看出

$$\sum_{i=1}^{\beta(t)} T_i \le \int_0^t N(\tau)\,\mathrm{d}\tau \le \sum_{i=1}^{\alpha(t)} T_i \tag{3-6}$$

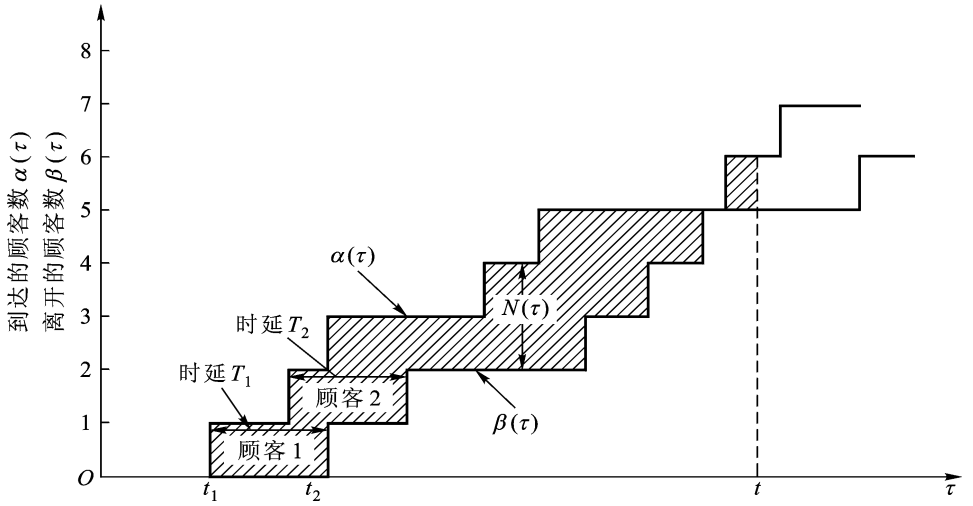


图3-2 Little 定理的正确性说明

式(3-6)中间部分表示图3-2中阴影部分的面积。将上式除以t得

$$\frac{1}{t}\sum_{i=1}^{\beta(t)} T_i \le \frac{1}{t}\int_0^t N(\tau)\,\mathrm{d}\tau \le \frac{1}{t}\sum_{i=1}^{\alpha(t)} T_i \tag{3-7}$$

因为

$$\frac{1}{t}\sum_{i=1}^{\beta(t)} T_i = \frac{\beta(t)}{t}\cdot\frac{\sum_{i=1}^{\beta(t)} T_i}{\beta(t)} = \delta_t T_t \tag{3-8}$$

式中，δ_t为离开系统的平均速率，T_t为用户的平均时延。

$$\frac{1}{t}\sum_{i=1}^{\alpha(t)} T_i = \frac{\alpha(t)}{t}\cdot\frac{\sum_{i=1}^{\alpha(t)} T_i}{\alpha(t)} = \lambda_t T_t \tag{3-9}$$

式中，λ_t为到达系统的平均速率。所以，式(3-7)可以写成

$$\delta_t T_t \le N_t \le \lambda_t T_t \tag{3-10}$$

在系统达到稳态的情况下，进入系统的顾客数等于离开系统的顾客数，从而有：

$$\delta T \le N \le \lambda T \tag{3-11}$$

$$\delta = \lambda \tag{3-12}$$

所以有

$$N = \lambda T$$

3.1.2 Little定理的应用

现在通过几个例子说明如何应用Little定理。

例3.1 考察一个分组流通过一个节点在一条链路上的传输过程。假定分组到达率为λ，分组在输出链路上的平均传输时间为$\bar{x}$，在该节点中等待传输(不包括正在传输)的分组的个数(队长)为N_Q，分组在节点中等待的时间(不包括传输时间)为W。如果仅把节点中等待的队列作为考虑的对象，则可以应用Little定理，有

$$N_Q = \lambda W \tag{3-13}$$

如果仅把输出链路作为考虑对象，则可以应用Little定理，有

$$\rho = \lambda\bar{x} \tag{3-14}$$

*ρ表示在传输链路上的平均分组数。由于该链路上最多有一个分组在传输，因*此ρ表示信道处于忙的时间所占的比例，即信道利用率。

例3.2 求解一个网络中分组的平均时延。假定一个网络有n个节点，节点i的分组到达率为λ_i，i=1,…,n，节点中的平均分组数为N_i，平均时延为T_i，则对节点i应用Little定理有$N_i=\lambda_i T_i$。设网络中的平均分组数为N，则网络中每个分组的平均时延为

$$T = \frac{N}{\lambda} \tag{3-15}$$

式中，$\lambda = \sum_{i=1}^{n} \lambda_i$ 是网络中的分组到达率。

例3.3 假定一个服务大厅有 K 个服务窗口，该服务大厅最多可容纳 N 个顾客（$N \geq K$）。又假定服务大厅始终是客满的，即离开一个顾客将会有一个新顾客立刻进入大厅。设每个顾客的平均服务时间为 $\overline{X}$，问顾客在大厅内停留的时间 T 为多少？

解 设进入大厅的顾客到达率为 λ。对整个系统而言，应用 Little 定理有

$$N = \lambda T] \quad T = \frac{N}{\lambda} \tag{3-16}$$

对服务窗口应用 Little 定理有

$$K = \lambda \overline{X}] \quad \lambda = \frac{K}{\overline{X}} \tag{3-17}$$

将式（3-17）代入式（3-16）得

$$T = \frac{N\overline{X}}{K} \tag{3-18}$$

例3.4 现在改变例3.3中顾客到达方式。假定顾客到达时发现服务窗口被占满就立即离开系统（即顾客被阻塞或丢失）。设顾客的到达率为 λ，问顾客被阻塞的概率 β 为多少？

解 因为顾客是随机到达的，则系统有时满，有时空。平均而言，平均处于忙的窗口数为 $\overline{k}$（$\overline{k} \leq K$）。则系统中的平均用户数为

$$\overline{k} = (1-\beta)\lambda\overline{X} \tag{3-19}$$

式中，$(1-\beta)\lambda$ 表示没有被阻塞部分（或被正常服务部分）的顾客到达率。

$$\beta = 1 - \frac{\overline{k}}{\lambda\overline{X}} \geq 1 - \frac{K}{\lambda\overline{X}} \tag{3-20}$$

上式给出了系统阻塞概率的下限。

假设一个电话交换机同时可以服务 $K=300$ 个用户的呼叫，每个用户的平均通话时间为 3 min，设该交换机服务区内有 3000 个用户。如果在忙时，每个用户至少半小时打一次电话，则每分钟的呼叫到达率 $\lambda \geq 100$ 次/分钟，此时根据式（3-20）可知，肯定会出现打不通电话的情况。

3.2 M/M/m 型排队系统

“M/M/m”是排队系统的通用表示法。第一个字母表示到达过程的特征，M 表示是无记忆的 Poisson 过程。第二个字母表示服务时间的概率分布，M 表示

指数分布，G表示一般分布，D表示确定性分布。第三个字母表示服务员的个数。有时还有第四个字母，表示系统的容量的大小。如果没有第四个字母，则表示系统的容量是无限大的。本节将讨论M/M/1，M/M/m，M/M/∞，M/M/m/m等排队模型。

3.2.1 M/M/1排队系统

M/M/1排队系统的示意图如图3-3所示。其到达过程为Poisson过程，到达率为λ；系统允许排队的队长可以是无限的（系统的缓存容量无限大）；服务过程为指数过程，服务速率为μ，平均服务时间为$\frac{1}{\mu}$，服务员的数目为1，到达过程与服务过程相互独立。

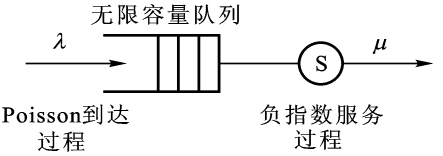


图3-3 M/M/1排队系统

令系统的状态为系统中的用户数$N(t)$，可以用状态转移概率来描述该系统的行为。将时间轴离散化（对$N(t)$进行采样，采样间隔为δ，δ为大于0的任意小常数），则显然该系统可用马氏链来描述。

假定考察的区间为$I_k=(k\delta,(k+1)\delta)$。在该区间内的状态$N_k=N(t=k\delta)$的状态转移概率为：

$$P_{00}=P\{\text{在 } I_0 \text{ 内没有到达}\}=e^{-\lambda\delta}=1-\lambda\delta+o(\delta) \tag{3-21}$$

$$P_{ii}=P\{\text{在 } I_k \text{ 内有0个到达且有0个离开}\}=e^{-\lambda\delta}e^{-\mu\delta}$$
$$=1-\lambda\delta-\mu\delta+o(\delta) \quad i\geq 1 \tag{3-22}$$

$$P_{i,i+1}=P\{\text{在 } I_k \text{ 内仅有1个到达且0个离开}\}=(\lambda\delta)e^{-\lambda\delta}e^{-\mu\delta}$$
$$=\lambda\delta+o(\delta) \quad i\geq 0 \tag{3-23}$$

$$P_{i,i-1}=P\{\text{在 } I_k \text{ 内有0个到达且仅有1个离开}\}=e^{-\lambda\delta}\mu\delta e^{-\mu\delta}$$
$$=\mu\delta+o(\delta) \quad i>1 \tag{3-24}$$

$$P_{1,0}=P\{\text{在 } I_k \text{ 内仅有0个到达且一个离开}\}=e^{-\lambda\delta}(1-e^{-\mu\delta})$$
$$=\mu\delta+o(\delta) \tag{3-25}$$

在上述表达式中，$e^{-\lambda\delta}$是I_k内没有用户到达的概率，$e^{-\mu\delta}$表示在I_k内没有用户离开的概率，$\lambda\delta e^{-\lambda\delta}$是在$I_k$内仅有一个用户到达的概率，$\mu\delta e^{-\mu\delta}$表示在$I_k$内仅有一个用户离开的概率，$(1-e^{-\mu\delta})$为在$I_k$内正在服务的用户将结束服务的概率。

对于任一个状态而言，有

$$P_{i,i+1}+P_{i,i-1}+P_{ii}=1+o(\delta) \tag{3-26}$$

利用上述结果，可以画出系统的状态转移图如图3-4所示。图中忽略了高阶无穷小项$o(\delta)$。

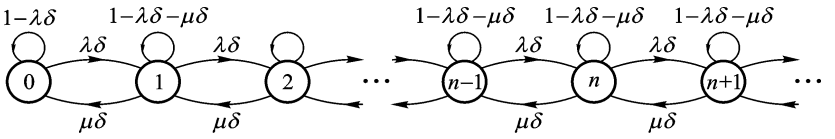


图3−4 M/M/1系统状态转移图

设系统状态的稳态概率为

$$p_n = \lim_{k\to\infty} P\{N_k = n\} = \lim_{k\to\infty} P\{N(t) = n\} \tag{3-27}$$

在系统能够达到稳态的情况下，系统从状态 n 转移到状态 n+1 的频率必然等于系统从状态 *n+1* 转移到状态 *n* 的频率。即有

$$p_n P_{n,n+1} = p_{n+1} P_{n+1,n} \tag{3-28}$$

否则系统不可能稳定。

将式 (3−23) 和式 (3−24) 代入上式得

$$p_n\lambda\delta + o(\delta) = p_{n+1}\mu\delta + o(\delta) \tag{3-29}$$

令 $\delta\to 0$，则有：

$$p_n\lambda = p_{n+1}\mu \tag{3-30}$$

$$p_{n+1} = \frac{\lambda}{\mu}p_n = \rho p_n \tag{3-31}$$

式中，$\rho=\frac{\lambda}{\mu}$，式 (3−30) 被称为全局平衡方程。

通过递推可得

$$p_{n+1} = \rho^{n+1} p_0 \qquad n = 0,1,2,\cdots \tag{3-32}$$

此外，由于 p_n 为状态 n 的概率，因而必有 $\sum_{n=0}^{\infty} p_n = 1$。

因此，在 $\rho<1$ 的条件下，有

$$\sum_{n=0}^{\infty} \rho^n p_0 = \frac{p_0}{1-\rho} = 1 \tag{3-33}$$

即 $p_0 = 1-\rho$。*将其代入式* (3−32) 得系统的稳态概率为

$$p_n = \rho^n(1-\rho) \qquad n = 0,1,2,\cdots \tag{3-34}$$

系统中的平均用户数为

$$N = \sum_{n=0}^{\infty} np_n = \sum_{n=0}^{\infty} n\rho^n(1-\rho) = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda} \tag{3-35}$$

由于 $\rho=\frac{\lambda}{\mu}$是到达率与服务速率之比，它反映了系统的繁忙程度。当 ρ 增加时，N 将随之增加；当 ρ 趋于 1 时，N 将趋于∞。$\rho=1-p_0$ 实际上是用户等待的概率 (P_Q)，即当用户到达系统时，发现系统中用户数不为 0 的 (用户数≥1) 的概率 $P_Q = \sum_{n=1}^{\infty} p_n$。如果 $\rho>1$，则系统将来不及服务，必然会导致系统中的

用户数趋于无穷大。

利用 Little 定理,可求得用户的平均时延为

$$T = \frac{N}{\lambda} = \frac{\rho}{1-\rho} \cdot \frac{1}{\lambda} = \frac{1}{\mu - \lambda} \tag{3-36}$$

通过简单的证明,可以求得用户的时延是服从均值为 T 的指数分布。

由于每个用户的平均服务时间为$\frac{1}{\mu}$,则每个用户的平均等待时间为

$$W = T - \frac{1}{\mu} = \frac{\lambda}{\mu} \cdot \frac{1}{\mu - \lambda} = \frac{\rho}{\mu(1-\rho)} \tag{3-37}$$

利用上式,可得到平均时延 T 的另一个表达式

$$T = W + \frac{1}{\mu} = \frac{1}{\mu} + \frac{\lambda}{\mu} \frac{1}{\mu - \lambda} = \frac{1}{\mu} + \frac{P_Q}{\mu - \lambda} \tag{3-38}$$

系统中的平均队长为

$$N_Q = \lambda W = \frac{\lambda^2}{\mu(\mu - \lambda)} = \frac{\lambda}{\mu} \cdot \frac{\rho}{1-\rho} \tag{3-39}$$

例 3.5　设某学校有一部传真机为全校2万名师生提供传真服务。假定每份传真的传输时间服从负指数分布,其平均传输时间为 3 min,并假定每个人发送传真的可能性相同。如果希望平均排队的队长不大于 5 人,试问平均每人间隔多少天才可以发送一份传真?

解　假定要发送的传真服从 Poisson 到达,则该传真服务系统可用 M/M/1 队列来描述。已知$\frac{1}{\mu}$=3min, N_Q=5 人,要求解 λ(份/天)。

根据式(3-39)有:

$$N_Q = \frac{\lambda}{\mu} \frac{\rho}{1-\rho} = \frac{\rho^2}{1-\rho} = 5 \text{ 人}$$

$$\rho = \frac{\lambda}{\mu} = \frac{3\sqrt{5} - 5}{2} \approx 0.854$$

系统总的可以发送的传真速率为

$$\lambda = \frac{\rho}{\frac{1}{\mu}} \approx \frac{0.854}{3} \approx 0.285 \text{ 份/分钟} \approx 410 \text{ 份/天}$$

则平均每个用户要隔$\frac{20000}{410} \approx 49$天才可以发送一份传真。如果提供传真服务的时间不是每天 24 小时开放,如每天开放 12 小时,则间隔的时间要增加一倍。

例 3.6　设有一个分组传输系统。其分组到达过程是到达率为 λ 的 Poisson 过程,分组长度服从指数分布,其均值为$\frac{1}{\mu}$。如果将 k 个这样的分组流统计复接在一个高速信道上来传输,即将输入到达率提高 k 倍,并将信道速率提高 k

倍，即服务时间变为$\frac{1}{k\mu}$，这相当于将 k 个平行的低速传输的信道统计复接到一个高速信道上。试比较两种情况下的传输时延。

解 原系统中的平均分组数和平均时延为：

$$N=\frac{\lambda}{\mu-\lambda}$$

$$T=\frac{1}{\mu-\lambda}$$

统计复接后系统中的平均分组数和平均时延为：

$$N'=\frac{k\lambda}{k\mu-k\lambda}=\frac{\lambda}{\mu-\lambda}$$

$$T'=\frac{1}{k\mu-k\lambda}=\frac{1}{k}\frac{1}{\mu-\lambda}=\frac{T}{k} \qquad (3-40)$$

从上式中可以看出，采用统计复用后，系统的平均时延降低到原来平均时延 T 的$\frac{1}{k}$。

现在反过来考虑该例子。将一个高速的分组传输信道分解成为 k 个信道（即采用 FDM/TDM 方式来传送各用户的分组）。设该高速信道的分组到达率为 λ，服务时间为$\frac{1}{\mu}$，现将信道分成 k 个并行的子信道，各个子信道的分组到达率为$\frac{\lambda}{k}$，其服务时间为$\frac{k}{\mu}$ 因为子信道的传输速率为高速信道的$\frac{1}{k}$，试比较分解前后的传输时延。

高速信道的传输时延为

$$T=\frac{1}{\mu-\lambda}$$

分解后各子信道的传输时延为

$$T'=\frac{1}{\frac{\mu}{k}-\frac{\lambda}{k}}=\frac{k}{\mu-\lambda}=k\cdot T \qquad (3-41)$$

从该式可以看出，将一个高速信道分解为 k 个低速信道后，传输时延将增加 k 倍。这样分解的另一个问题是，当各个低速信道的到达率不同时，出现忙闲不均，有的信道很闲，有的信道不足以满足用户的需求。这种分解的优点是当子信道的容量与用户到达相匹配时，各信道没有等待时延和等待队列；而在高速信道中，尽管传输的时延减少了，但各用户的等待时间及时延的变化都会增加。

3.2.2 M/M/m 排队系统

在 M/M/m 排队系统中，服务员为 m 个。设系统的到达率为 λ，每个用户

的服务速率为 μ。当系统中的用户数 n> *m* 时,*用户离开的速率为 mμ* (因为只有 m个服务员),当 n≤ *m* 时,*用户离开的速率为 nμ* (因为顾客数小于服务员数)。此时的系统状态(即系统中的用户数)转移图如图 3-5 所示。

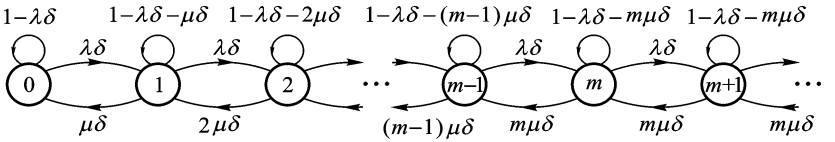


图 3-5 M/M/m 排队系统的状态转移图

在该状态图中,如果 m→∞,即为 M/M/∞排队系统。如果仅有 0~ *m* 状态,则为 M/M/m/m 排队系统(系统中的容量仅为 m)。

采用与 M/M/1 相同的分析方法,当 δ→0 时,系统的稳态全局平衡方程为(令 p_n 为系统状态的稳态概率)

$$\begin{aligned} \lambda p_{n-1} &= n\mu p_n \qquad n \le m \\ \lambda p_{n-1} &= m\mu p_n \qquad n > m \end{aligned} \tag{3-42}$$

利用与 M/M/1 排队系统相同的推导过程,可得系统状态的稳态概率为

$$p_n = \begin{cases} p_0 \dfrac{(m\rho)^n}{n!} & n \le m \\ p_0 \dfrac{m^m \rho^n}{m!} & n > m \end{cases} \tag{3-43}$$

式中:

$$\rho = \frac{\lambda}{m\mu} < 1 \tag{3-44}$$

$$p_0 = \left[1 + \sum_{n=1}^{m-1} \frac{(m\rho)^n}{n!} + \sum_{n=m}^{\infty} \frac{(m\rho)^n}{m!} \frac{1}{m^{n-m}}\right]^{-1} \tag{3-45}$$

下面对上述结果进行讨论:

(1) 用户到达系统必须等待的概率(发现所有的服务员都在忙的概率)

$$P_Q = \sum_{n=m}^{\infty} p_n = \frac{p_0 (m\rho)^m}{(1-\rho)\, m!} \tag{3-46}$$

这就是等待制系统中需等待的概率,该公式称为 Erlang C 公式。它说明在具有 *m 条线路的电路交换系统中,用户到达时发现 m 条线路都忙的概率。*

(2) 正在排队的用户数

$$N_Q = \sum_{n=0}^{\infty} n p_{n+m} = \frac{p_0 (m\rho)^m}{m!} \sum_{n=0}^{\infty} n\rho^n \tag{3-47}$$

利用 $\sum_{n=0}^{\infty} n\rho^n = \dfrac{\rho}{(1-\rho)^2}$ 和式(3-46)得

$$N_Q = P_Q \frac{\rho}{1-\rho} \tag{3-48}$$

(3) 用户的平均等待时延(利用 Little 定理)

$$W = \frac{N_Q}{\lambda} = \frac{\rho}{\lambda(1-\rho)} P_Q \tag{3-49}$$

每个用户的平均时延为

$$T = \frac{1}{\mu} + W = \frac{1}{\mu} + \frac{P_Q}{m\mu - \lambda} \tag{3-50}$$

系统中的平均用户数为

$$N = \lambda T = \frac{\lambda}{\mu} + N_Q = m\rho + \frac{\rho P_Q}{1-\rho} \tag{3-51}$$

例 3.7 假定有 *m* 个信道,到达率为 λ 的分组流动态共享这 *m* 个信道,每个信道的服务时间为 $\frac{1}{\mu}$,试求分组的平均时延 T,并将该平均时延与到达率为 λ 的分组流在服务速率为 mμ (输入分组在一个高速信道上传输) 的单信道上传输的平均时延 $\hat{T}$进行比较。

解 该例题的前一部分为一个服务速率为 μ 的 M/M/m 排队系统,后一部分为服务速率为 mμ 的一个 M/M/1 排队系统。因此:

$$T = \frac{1}{\mu} + \frac{P_Q}{m\mu - \lambda}$$

$$\hat{T} = \frac{1}{m\mu} + \frac{\hat{P}_Q}{m\mu - \lambda}$$

式中,$\hat{P}_Q$ 为排队(等待)概率。

现在将 T 和 $\hat{T}$作一比较。

(1) 在轻负荷的情况下(ρ≪1),有 $P_Q \approx 0$,$\hat{P}_Q \approx 0$。根据 T 和 $\hat{T}$的表达式有 $\frac{T}{\hat{T}} \approx m$。*也就是说,在轻负荷的情况下,分组的时延主要由分组的传输时延决定*,m 个信道时的传输时延是单信道高速传输时延的 m 倍。

(2) 在重负荷的情况下(ρ 接近于 1),有 $P_Q \approx 1$,$\hat{P}_Q \approx 1$。因为 $\rho = \frac{\lambda}{m\mu}$,所以有 $\frac{1}{\mu} \gg \frac{1}{m\mu - \lambda}$,进而有 $\frac{T}{\hat{T}} \approx 1$。也就是说,在重负荷的情况下,分组的时延主要由分组的等待时延所决定,此时两者时延基本相等。

M/M/m 排队系统可进行下列两种形式的推广。$m \to \infty$ 时的 M/M/∞,以及限定系统容量为 m 时的排队系统 M/M/m/m。

对于 M/M/∞排队系统,系统的平衡方程仅取式(3-42)的上半部分,即

$$\lambda p_{n-1} = n\mu p_n \qquad n = 1,2,\cdots \tag{3-52}$$

可以推出

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!} \qquad n = 1,2,\cdots \tag{3-53}$$

根据 $\sum_{n=0}^{\infty} p_n = 1$，可得

$$p_0 = e^{\frac{-\lambda}{\mu}} \tag{3-54}$$

结合式 (3-53) 和式 (3-54) 有

$$p_n = \left(\frac{\lambda}{\mu}\right)^n \frac{e^{\frac{-\lambda}{\mu}}}{n!} \tag{3-55}$$

该式是一个参量为$\frac{\lambda}{\mu}$的 Poisson 分布，因而系统中的平均用户数为

$$N = \frac{\lambda}{\mu} \tag{3-56}$$

利用 Little 定理，得分组的平均时延为

$$T = \frac{N}{\lambda} = \frac{1}{\mu} \tag{3-57}$$

上式表明，在 m=∞的排队系统中，没有等待时延，仅有服务时延。

对于 M/M/m/m 排队系统，系统中的容量为 m。当用户进入系统时，发现 *m 个服务员全忙时，就立刻离开系统 (或丢失)。这种情况主要用于电路交换系* 统。比如，打长途电话时，假定仅有 m 条线路可用，如果发现线路全忙，就会过一会再打或以后再打，这就相当于离开系统。这是一种呼损制系统，而不像 M/M/m 是一个等待制系统。

在呼损制系统中，感兴趣的主要参数是呼损率 (阻塞概率，blocking probability)。所谓呼损率就是新到用户发现系统所有线路都忙的概率，也就是他的呼叫被拒绝的概率。

M/M/m/m 系统的状态转移图如图 3-6 所示。

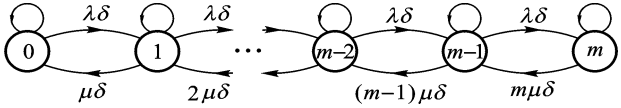


图 3-6 M/M/m/m 排队系统的状态转移图

系统的平衡方程仅取式 (3-42) 的上半部分，即：

$$\lambda p_{n-1} = n\mu p_n, \qquad n = 1,2,\cdots,m \tag{3-58}$$

$$p_n = p_0 \left(\frac{\lambda}{\mu}\right)^n \frac{1}{n!}, \qquad n = 1,2,\cdots,m \tag{3-59}$$

根据 $\sum_{n=0}^{\infty} p_n = 1$，可得

$$p_0 = \left[\sum_{n=0}^{m} \frac{\lambda}{\mu}^n \frac{1}{n!}\right]^{-1} \tag{3-60}$$

呼损率就是所有 m 个服务员都忙的概率，即

$$B = p_m = \frac{\frac{\lambda}{\mu}^m}{m!} \Big/ \sum_{n=0}^{m} \frac{\frac{\lambda}{\mu}^n}{n!} \tag{3-61}$$

该公式成为 Erlang B 公式。它不仅适用于服务时间指数分布的系统，而且同样适用于服务时间服从均值为$\frac{1}{\mu}$的一般性分布的系统。

利用该公式还可以确定每个服务员的繁忙程度（如果每个服务员对应一个物理信道，则繁忙程度就对应信道利用率）η

$$\eta = \frac{(1-B)}{m}\frac{\lambda}{\mu} = (1-B)\rho \tag{3-62}$$

由于式（3-61）计算复杂，可以用表格的形式给出。m、ρ、B、η的关系的简表如表3-1所示。

表3-1 m、ρ、B、η的关系表格

B	1%		5%		10%		20%	
m	ρ	η	ρ	η	ρ	η	ρ	η
1	0.010	1%	0.053	5%	0.111	10%	0.25	20%
5	0.272	1.66%	0.444	42.2%	0.576	51.9%	0.802	64.16%
10	0.446	44.2%	0.622	59.1%	0.751	67.6%	0.969	77.5%
15	0.541	53.5%	0.708	67.2%	0.832	74.9%	1.041	83.4%
20	0.602	59.5%	0.762	72.9%	0.858	79.3%	1.081	86.5%

例3.8 假定系统的服务员数分别为 $m_1=10$ 和 $m_2=20$，每次呼叫的平均时间为3min，要求系统的呼损率小于5%，试求系统支持的最大呼叫到达率和服务员的繁忙程度。

解 根据表3-1可以查出，$m_1=10$ 时对应的 $\rho_1=0.622$，$\eta_1=59.1\%$，$m_2=20$时对应的 $\rho_2=0.762$，$\eta_2=72.9\%$。

因为：$$\rho=\frac{\lambda}{m\mu},\lambda=m\mu\rho=\frac{m\rho}{\frac{1}{\mu}}$$

所以：$$\lambda_1=\frac{10\times0.622}{3}\approx2.07 \text{ 呼叫/分钟}$$

$$\lambda_2=\frac{20\times0.762}{3}\approx5.08 \text{ 呼叫/分钟}$$

从上例和表3-1可以看出，如果系统要求呼损率越小，则系统可承担的负荷越小，各服务员的繁忙程度就越低。在相同的呼损率条件下，服务员越多，各服务员的繁忙程度越高，因而系统承担的负荷越大。这也反映了统计复用带来的好处。

注意在M/M/m/m系统中$\rho=\frac{\lambda}{m\mu}$可以大于1，此时系统不能服务的到达都将丢失。例如，在m=20，B=20%时，$\rho=1.081$，但此时只有80%*的到达业务*得到服务。

3.3 M/G/1型排队系统

M/G/1排队系统与M/M/1系统的主要差别是服务时间为一般性的独立同分布的服务时间。本节将首先讨论M/G/1排队系统的基本概念，然后讨论M/G/1队列的多种推广形式。另外，本章给出的基于剩余服务时间的概念对M/G/1队列的平均等待时间求解和第四章多址协议的性能分析具有重要的作用。

3.3.1 M/G/1排队系统

假定第 *i个用户的服务时间为* X_i，X_i是独立同分布的，并且与到达间隔相互独立。令$X=\{X_1,X_2,\cdots\}$，则服务时间的均值和二阶矩为：

$$\text{平均服务时间}=\overline{X}=E\{X\}=\frac{1}{\mu}$$

$$\text{服务时间的二阶矩}=\overline{X^2}=E\{X^2\}$$

下面将证明，M/G/1排队系统的平均等待时间为

$$W=\frac{\lambda\overline{X^2}}{2(1-\rho)} \tag{3-63}$$

式中，$\rho=\frac{\lambda}{\mu}=\lambda\overline{X}$。该式称为P-K(Pollaczek-Khinchin)公式。

根据上述P-K公式，可以得该系统的平均时延为

$$T=\overline{X}+W=\overline{X}+\frac{\lambda\overline{X^2}}{2(1-\rho)} \tag{3-64}$$

应用 Little 定理,有:

平均队列长度为

$$N_Q = \lambda W = \frac{\lambda^2 \overline{X^2}}{2(1-\rho)} \tag{3-65}$$

系统中的平均用户数为

$$N = \lambda T = \lambda \overline{X} + \frac{\lambda^2 \overline{X^2}}{2(1-\rho)} \tag{3-66}$$

从上述结果可以看出:

(1) 如果 G=M,即服务时间服从指数分布,则有 $\overline{X^2} = \frac{2}{\mu^2}$,进而有

$$W = \frac{\rho}{\mu(1-\rho)} \tag{3-67}$$

该式与 M/M/1 队列的结果式 (3-37) 完全相同。

(2) 如果服务时间是常量,即 $\overline{X} = \frac{1}{\mu}$, $\overline{X^2} = \frac{1}{\mu^2}$,则有

$$W = \frac{\rho}{2\mu(1-\rho)} \tag{3-68}$$

该式与 M/D/1 系统的结果相同。其等待时间是 M/M/1 系统的一半。

由于在均值相同的分布中,确定性分布的 $\overline{X^2}$ 最小,因此,M/D/1 队列的 W、T、N_Q、N 等参数的取值是 M/G/1 系统相应参数的下限。

P-K 公式的证明过程如下。

证明 P-K 公式证明的思路是基于对平均剩余服务时间 (mean residual service time) 的求解。设第 i 个用户到达系统时,第 l 个用户正在接受服务,其剩余服务时间为 R_i,此时等待队列中有 N_i 个用户,如图 3-7 所示。

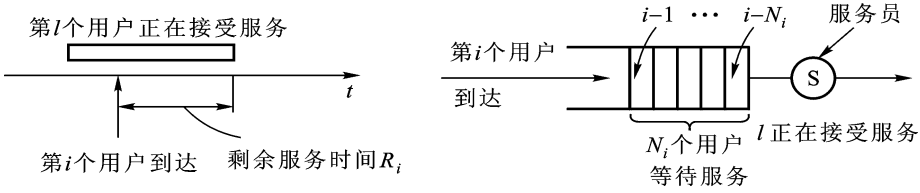


图 3-7 第 i 个用户到达的系统状态

(a) 用户到达的时刻 (b) 用户到达时的状态

设第 k 个用户的服务时间为 X_k,则由图 3-7 可知,用户 i 的等待时间为

$$W_i = R_i + N_i \text{个用户的服务时间} = R_i + \sum_{k=i-N_i}^{i-1} X_k \tag{3-69}$$

对上式求平均得

$$\overline{W_i} = E\{R\} + E\left\{\sum_{k=i-N_i}^{i-1} X_k\right\} = E\{R\} + \overline{X} \cdot E\{N_i\} \tag{3-70}$$

式中 X_k 和 N_i 均为随机变量。X_k 的均值为 $\overline{X}$；N_i 的均值为 $E\{N\}$，表示平均排队长度。

令 $i \to \infty$，$W = \lim_{i\to\infty} \overline{W_i}$，有

$$W = R + \overline{X} N_Q = R + \frac{1}{\mu} N_Q = R + \frac{1}{\mu}\lambda W = R + \rho W \tag{3-71}$$

式中：$R = \lim_{i\to\infty} E\{R\}$，$N_Q = E\{N\}$，$\rho = \frac{\lambda}{\mu}$。整理上式得

$$W = \frac{R}{1-\rho} \tag{3-72}$$

由该式可以看出，只要求得平均剩余服务时间 R，就可得到 W。

假定系统有稳态解，且具有各态历经性，则剩余服务时间 $r(\tau)$ 可用图3-8来表示。

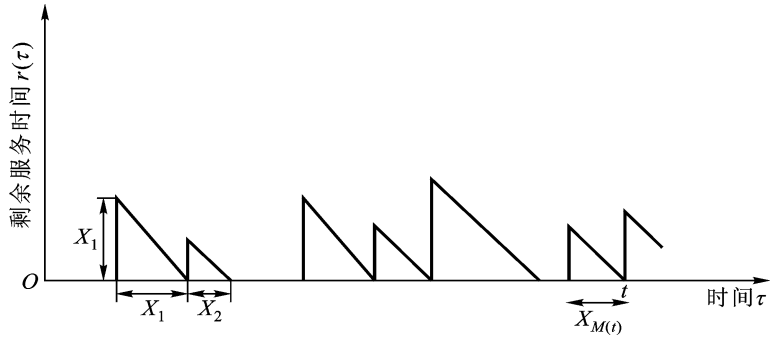


图3-8 剩余服务时间的变化曲线

为了方便起见，取 t 为 $r(t) = 0$ 的时刻，则 $[0, t]$ 区间的平均剩余服务时间为

$$R_t = \frac{1}{t}\int_0^t r(\tau)\,\mathrm{d}\tau = \frac{1}{t}\sum_{i=1}^{M(t)} \frac{1}{2} X_i^2 \tag{3-73}$$

式中 $M(t)$ 表示 $[0, t]$ 区间内已服务的用户数。

上式可以写成

$$R_i = \frac{1}{2} \cdot \frac{M(t)}{t} \cdot \frac{\sum_{i=1}^{M(t)} X_i^2}{M(t)} \tag{3-74}$$

式中第二项为平均到达率，第三项为 X_i 的二阶矩。

令 $t \to \infty$，得

$$R = \frac{1}{2}\lambda \overline{X^2} \tag{3-75}$$

将上式代入式(3-72)得P-K公式

$$W = \frac{R}{1-\rho} = \frac{\lambda \overline{X^2}}{2(1-\rho)} \tag{3-76}$$

从P-K公式可以看出,M/G/1队列的一个重要特征:$W \propto \overline{X^2}$。即使$\rho < 1$,如果$\overline{X^2} \to \infty$,则$W \to \infty$。它说明,如果有少量用户有非常长的服务时间,一旦这些用户被服务,将导致队列非常长。

例3.9 返回n-ARQ系统的时延性能分析

设返回n-ARQ系统中的分组到达过程是速率为λ的Poisson过程,分组(帧)长度相同,为一个单位,等待应答的最长时间(重传间隔)为n-1个分组长度。因此,任一分组传输出错后,将在n-1个分组后重传,如图3-9所示。

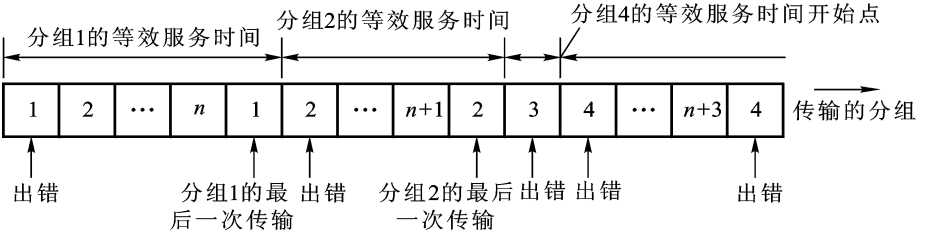


图3-9 返回n-ARQ系统的传输过程

在该系统中,一个分组(帧)的服务时间,不仅仅是一个分组的一次传输时间,而且还应当包括分组的重传时间。因此,这里的服务时间为一个等效服务时间,它等于从一个分组的第一次传输开始,到该分组的最后一次传输结束时刻之间的时间长度,它服从一般性分布,因而该系统可用M/G/1队列模型来描述。

假定分组传输错误的概率为p。如果分组重传次数为k,则等效服务时间为$1+kn$,其概率为$(1-p)p^k$(一次正确传输,k次传输错误,共k+1次传输),即等效服务时间X_k的概率分布为

$$P(X_k = 1 + kn) = (1-p)p^k \tag{3-77}$$

其一阶矩和二阶矩分别为:

$$\overline{X} = \sum_{k=0}^{\infty} (1+kn)(1-p)p^k = 1 + \frac{np}{1-p} \tag{3-78}$$

$$\overline{X^2} = \sum_{k=0}^{\infty} (1+kn)^2 (1-p)p^k = 1 + \frac{2np}{1-p} + \frac{n^2(p+p^2)}{(1-p)^2} \tag{3-79}$$

该公式的推导过程中,利用了下列结果:

$$\sum_{k=0}^{\infty} p^k = \frac{1}{1-p}, \quad \sum_{k=0}^{\infty} kp^k = \frac{p}{(1-p)^2}, \quad \sum_{k=0}^{\infty} k^2 p^k = \frac{p+p^2}{(1-p)^3}$$

利用P-K公式可得,分组的平均等待时延和分组的平均时延分别为:

$$W = \frac{\lambda \overline{X^2}}{2(1-\lambda \overline{X})} \tag{3-80}$$

$$T = \overline{X} + W \tag{3-81}$$

假定 $p = 10^{-3}$，$n = 8$，$\lambda = 0.5$、0.8、0.9 和 0.95 分组/单位时间，则有 $\overline{X} = 1.008$，$\overline{X^2} = 1.08$，$W \approx 0.54$，2.23，5.24 和 11.29 个单位分组时间。由此可以看出，当分组的到达率接近系统容量时，分组的平均等待时间急剧上升。因此，为了保证分组的最大等待时间低于某一门限，应当限制分组的到达率。

3.3.2 服务员有休假的 M/G/1 排队系统

服务员有休假的 M/G/1 (M/G/1 queues with vacations) 排队系统是指在每一个忙周期后 (分组传输结束后)，服务员需要休假 [休假对应于服务员 (通信节点) 要进行其他处理，如存储数据、信令交换等]，在服务员休假期内到达的用户，要等待服务员休假结束后，才能被服务。如服务员休假期满后，没有用户到达，服务员进入另一个休假期。如图 3-10 所示。

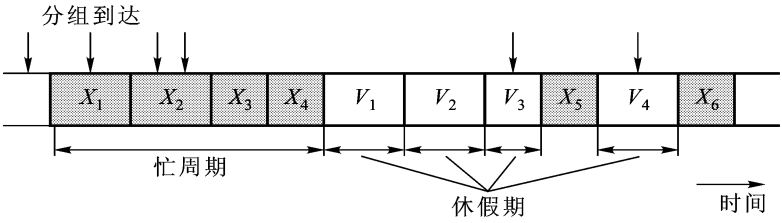


图 3-10 服务员有休假的 M/G/1 排队系统的分组传输过程

该服务员的休假期用 V_1，V_2，…表示，V_i 是独立同分布的随机变量，并且与到达间隔相互独立，其均值为 $\overline{V}$。用户 (分组) 的到达是速率为 λ 的 Poisson 过程，每个用户的服务时间也是独立同分布的随机变量，其平均服务时间为 $\frac{1}{\mu}$。

利用前一节的分析思路，一个新用户到达系统时，它可能会遇到两种情况：一种是当前有一个分组正在接受服务，另一种是服务员正在处于休假期。前一种情况的剩余服务时间与标准的 M/G/1 队列相同，后一种情况的剩余服务时间为服务员休假的剩余服务时间。如图 3-11 所示，图中阴影部分表示分组传输的剩余服务时间 X_i，其余部分是服务员休假期的剩余服务时间 V_i。

令 $M(t)$ 和 $L(t)$ 分别为 $[0, t]$ 区间内到达的用户数和服务员休假的次数。平均剩余服务时间为

$$\frac{1}{t}\int_0^t r(\tau)\,\mathrm{d}\tau = \frac{1}{t}\sum_{i=1}^{M(t)} \frac{1}{2} X_i^2 + \frac{1}{t}\sum_{i=1}^{L(t)} \frac{1}{2} V_i^2$$

$$= \frac{1}{2}\frac{M(t)}{t}\frac{\sum_{i=1}^{M(t)} X_i^2}{M(t)} + \frac{1}{2}\frac{L(t)}{t}\frac{\sum_{i=1}^{L(t)} V_i^2}{L(t)} \tag{3-82}$$

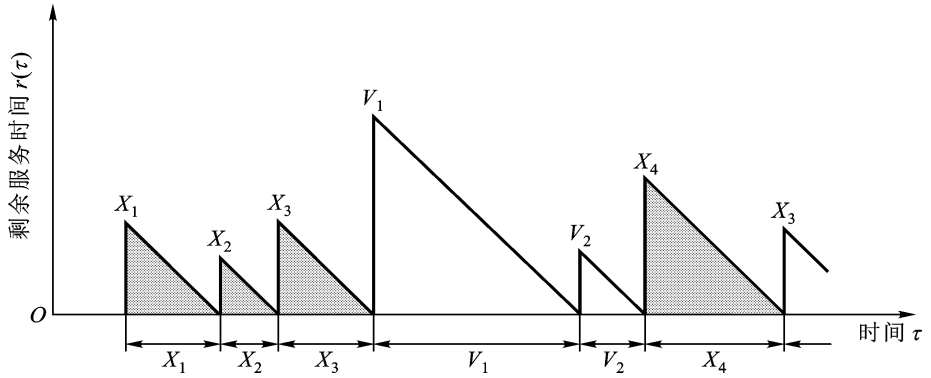


图 3-11 服务员有休假的 M/G/1 队列的剩余服务时间

式中：$\frac{M(t)}{t}$为平均分组到达率，$\frac{L(t)}{t}$为平均休假期的到达率，$\frac{\sum_{i=1}^{M(t)} X_i^2}{M(t)}$ 为 X_i 的二阶矩，$\frac{\sum_{i=1}^{L(t)} V_i^2}{L(t)}$ 为 V_i 的二阶矩。

由于分组传输和休假期的交替到达占满整个时间轴，则在单位时间，分组所占的比例为 $\frac{\lambda}{\mu} = \rho$，休假期占的比例为 $1-\rho$。因而有休假到达率为 $\frac{1-\rho}{\overline{V}}$。

根据上式，并令 $t \to \infty$，得平均剩余服务时间为

$$R = \frac{1}{2}\lambda\overline{X^2} + \frac{1}{2}\frac{1-\rho}{\overline{V}}\overline{V^2} \tag{3-83}$$

将上式代入式 (3-72 (b))，得平均等待时间为

$$W = \frac{R}{1-\rho} = \frac{\lambda\overline{X^2}}{2(1-\rho)} + \frac{1}{2}\frac{\overline{V^2}}{\overline{V}} \tag{3-84}$$

从该式可以看出式 (3-84) 比式 (3-76) 多出了一个第二项，它反映了由于服务员休假带来的额外等待时间。

例 3.10 时隙 FDM 和 TDM 系统的性能比较。

设基本的 FDM (Frequency Division Multiplexing) 系统有 m 个信道，每个信道的分组到达率为 $\frac{\lambda}{m}$，每个分组的传输时间为 m 个单位时间，即服务时间 $\frac{1}{\mu} = m$。在该系统中，只要分组到达时信道空闲，该分组就会立即得到服务。显然每

个信道是标准的 M/D/1 排队系统，$\rho=\frac{\frac{\lambda}{m}}{\mu}=\frac{\lambda/m}{1/m}=\lambda$，其等待时间为

$$W_{FDM}=\frac{\rho}{2\mu(1-\rho)}=\frac{\lambda m}{2(1-\lambda)} \tag{3-85}$$

假设 m 个信道变成以时隙为基础，时隙的宽度为 m 个单位，所有的分组都在时隙的开始点进行传输。如果在时隙开始时，没有分组到达，信道将空闲一个时隙（对应于服务员休假一次）。该系统被称为 SFDM 系统。其服务过程如图 3-12(a) 所示。该系统是服务员有休假的 M/D/1 系统，此时，$\rho=\lambda$，$\overline{X^2}=m^2$，$\overline{V}=m$，$\overline{V^2}=m^2$。利用式 (3-84) 有

$$W_{SFDM}=\frac{\lambda m}{2(1-\lambda)}+\frac{1}{2}m=W_{FDM}+\frac{1}{2}m=\frac{m}{2(1-\lambda)} \tag{3-86}$$

该公式直观上的理解是一致的，即在 SFDM 系统中，分组到达后平均要比 FDM 系统多等半个时隙。

对 TDM (Time Division Multiplexing) 系统，一帧由 m 个时隙组成，每个分组的传输时间为一个时隙，时隙的宽度为一个单位时间。由于信道速率提高了 m 倍，故分组的传输时间降低为原传输时间的 $\frac{1}{m}$。由于每个用户（分组流）在一帧中仅占一个时隙，如果在时隙开始时刻用户无分组到达，则该用户必须等到下一帧中相同的时隙才可能开始传输分组。因此，对于每个用户来说，信道将暂停（休假）一帧（m 个时隙）。在 TDM 系统中，剩余服务时间与 SFDM 系统的完全相同，即当该用户在某一帧有分组传送时，他等待的剩余服务时间不是一个时隙的剩余时间，而是一帧中的剩余时间，如图 3-12(b) 所示。

因此，TDM 系统中每个用户的等待时间为

$$W_{TDM}=W_{SFDM}=\frac{m}{2(1-\lambda)} \tag{3-87}$$

上述三种系统的分组总的时延分别为：

$$T_{FDM}=m+\frac{\lambda m}{2(1-\lambda)} \tag{3-88}$$

$$T_{SFDM}=T_{FDM}+\frac{m}{2} \tag{3-89}$$

$$T_{TDM}=1+\frac{m}{2(1-\lambda)}=T_{FDM}-\frac{m}{2}-1 \tag{3-90}$$

比较上面三式可知，TDM 系统的平均时延最小。

3.3.3 采用不同服务规则的 M/G/1 排队系统

在前面的讨论中没有考虑服务规则和分组的优先级，来一个分组服务一个

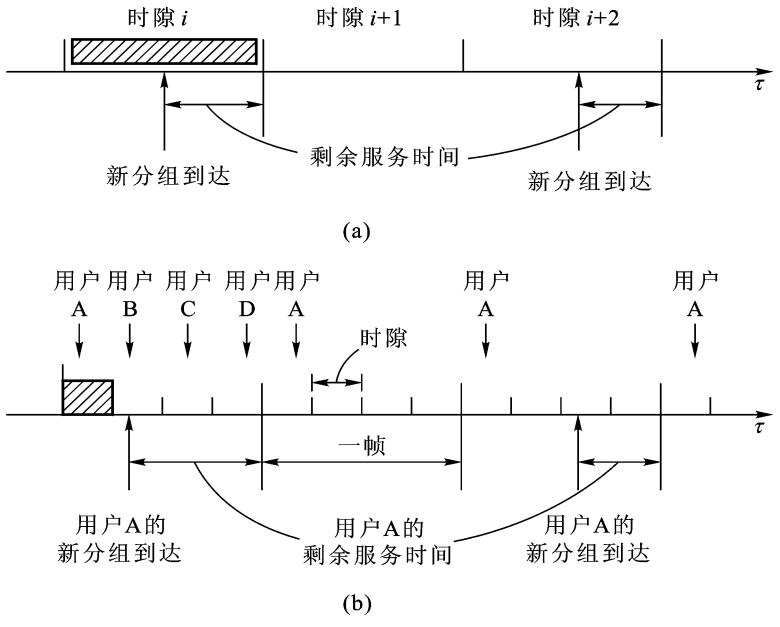


图 3-12 SFDM 和 TDM 系统的比较

(a) SFDM (b) TDM

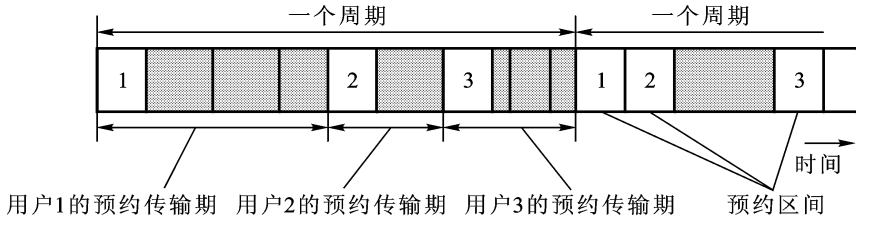
分组。但当多个用户共享一个统计复用的系统时,就必须采用适当的服务规则,如预约或轮询,对不同优先级的分组采用不同的服务方式等。

1. 采用预约方式的 M/G/1 排队系统

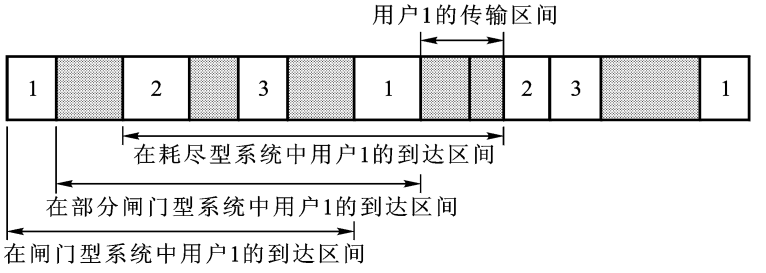
假设系统采用周期性预约的方式(类似于 LAN 中的令牌环协议),第一个用户预约并传输一定数量的分组后,由第二个用户进行预约和传输,依次类推,所有用户按循环的次序预约传输,如图 3-13(a)所示,所有用户预约传输一次构成一个周期。

由于在每次预约分组和数据分组的传输期内(称为预约传输期)会有新的分组到达,对于这些新的分组有三种不同的处理方式:一是在每个用户的预约传输期内,仅传送预约分组传输前到达的分组,该系统称为闸门型系统(gated system);二是在每个用户的预约传输期内,将上次预约传输期结束到本次预约传输期结束前到达的分组都在本预约传输期内传输,或者说在预约传输期内,将内存中所有到达的分组都传送完毕,该系统称为耗尽型系统(exhaustive system);三是在每个用户的预约传输期内,仅传送预约分组传输结束前到达的所有分组,该系统称为部分闸门型系统(partially gated system)。在三种不同的处理方式下,用户 1 的到达区间如图 3-13(b)所示。

假设所有(m 个)用户的分组到达过程都是相互独立的到达率为 $\frac{\lambda}{m}$的



(a)



(b)

图3-13 采用预约方式的M/G/1排队系统

(a) 所有用户的预约传输过程 (b) 在三种不同处理方式下,用户1的到达区间

Poisson过程,分组传送时间的均值为 $\overline{X}=\frac{1}{\mu}$,二阶矩为 $\overline{X^2}$,$\rho=\frac{\lambda}{\mu}$。

(1) 单用户系统。首先来看单用户闸门型系统的性能。

该系统的分组到达和传输过程如图3-14所示。

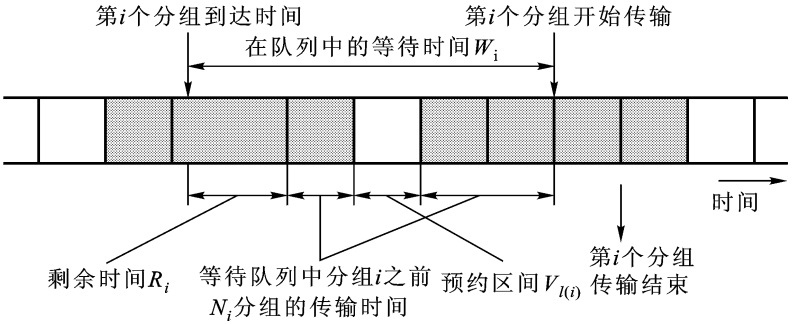


图3-14 单用户闸门型系统的分组到达和传输过程

假定第 l 个预约分组的长度为 V_i,V_i 是相互独立同分布的随机变量,其均值为 $\overline{V}$,二阶矩为 $\overline{V^2}$。从图3-14可以看出,第 i 个分组到达时等待时间包括剩余服务时间,N_i 个等待分组的传输时间和预约分组传输区间 $V_{I(i)}$。即

$$E\{W\} = E\{R\} + \frac{1}{\mu}E\{N\} + E\{V_{I(i)}\} \qquad (3-91)$$

当 $i\to\infty$ 时,有:

$$\lim_{i\to\infty} E\{N\} = N_Q = \lambda W$$

$$\lim_{i\to\infty} E\{V_{l(i)}\} = \overline{V}$$

$$\lim_{i\to\infty} E\{R\} = R$$

$$\lim_{i\to\infty} E\{W\} = W$$

式(3-91)变成

$$W = R + \frac{\lambda}{\mu} W + \overline{V} = R + \rho W + \overline{V} \tag{3-92}$$

经整理得

$$W = \frac{R}{1-\rho} + \frac{\overline{V}}{1-\rho} \tag{3-93}$$

式中，R的取值与服务员有休假的M/G/1排队系统相同，即

$$R = \frac{1}{2}\lambda\overline{X^2} + (1-\rho)\frac{\overline{V^2}}{2\overline{V}}$$

上式代入式(3-93)得，单用户闸门系统的平均等待时间为

$$W = \frac{\lambda\overline{X^2}}{2(1-\rho)} + \frac{\overline{V^2}}{2\overline{V}} + \frac{\overline{V}}{1-\rho} \tag{3-94}$$

如果预约分组长度为一个常量A，则有

$$W = \frac{\lambda\overline{X^2}}{2(1-\rho)} + \frac{A}{2}\cdot\frac{3-\rho}{1-\rho} \tag{3-95}$$

(2) 多用户系统。对于多用户系统来说，与单用户的差别主要是新到达分组进入不同用户的队列(系统共有m个队列)时，将会遇到不同数量的预约时隙数(服务休假次数)。设系统有m个用户，当第i个分组到达系统时正好是第l个用户的预约传输期。如果第i个分组属于l+1个用户的队列，则它只会遇到一个预约分组的传输。如果第i个分组属于第l−1个用户的队列，则它会遇到$m-1$个预约分组的传输。

设每一用户的预约分组的传输时间为V_l，其均值为$\overline{V}_l$，则对于耗尽型系统来说，第i个分组在第l个用户的预约传输期到达，但属于$(l+j) \bmod m$个用户，则要等待的预约分组传输时间之和为$Y_i = V_{(l+1) \bmod m} + \cdots + V_{(l+j) \bmod m}$，$j \geq 1$。因此，在耗尽型系统中的平均等待时间为

$$E\{W\} = E\{R\} + \frac{1}{\mu}E\{N\} + E\{Y\} \tag{3-96}$$

式中，第一项是平均剩余服务时间，第二项是平均等待分组的传输时间，第三项是平均等待的预约分组的传输时间(参见图3-15)。

令$i\to\infty$，有

$$W = R + \rho W + Y \tag{3-97}$$

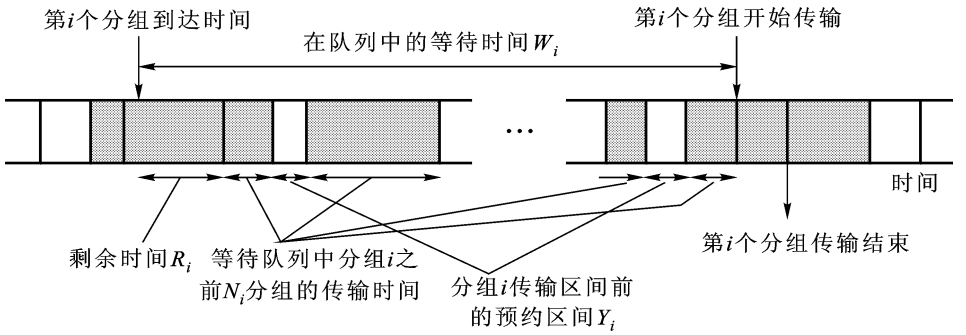


图3-15 多用户预约排队系统

经整理得

$$W=\frac{R+Y}{1-\rho} \tag{3-98}$$

利用式(3-83)得

$$R=\frac{1}{2}\lambda\overline{X^2}+\frac{1-\rho}{2}\frac{\overline{V^2}}{\overline{V}}=\frac{1}{2}\lambda\overline{X^2}+\frac{1-\rho}{2}\frac{\frac{1}{m}\sum_{l=0}^{m-1}\overline{V_l^2}}{\frac{1}{m}\sum_{l=0}^{m-1}\overline{V_l}}$$

$$=\frac{1}{2}\lambda\overline{X^2}+\frac{1-\rho}{2}\frac{\overline{V^2}}{\overline{V}} \tag{3-99}$$

式中：

$$\overline{V}=\frac{1}{m}\sum_{l=0}^{m-1}\overline{V_l} \tag{3-100}$$

$$\overline{V^2}=\frac{1}{m}\sum_{l=0}^{m-1}\overline{V_l^2} \tag{3-101}$$

另外，可以证明，耗尽型系统的平均等待的预约分组传输时间为

$$Y=E\{Y\}=\frac{(m-\rho)\overline{V}}{2}-\frac{(1-\rho)\sum_{i=0}^{m-1}\overline{V_l^2}}{2m\overline{V}} \tag{3-102}$$

令

$$\sigma_V^2=\frac{\sum_{l=0}^{m-1}(\overline{V_l^2}-\overline{V^2})}{m} \tag{3-103}$$

该式表示所有的预约分组长度的方差。

将式(3-99)和(3-102)代入式(3-98)，可得耗尽型系统的平均等待时间为

$$W=\frac{\lambda\overline{X^2}}{2(1-\rho)}+\frac{(m-\rho)\overline{V}}{2(1-\rho)}+\frac{\sigma_V^2}{2\overline{V}} \tag{3-104}$$

对于部分闸门性系统，它与耗尽型系统的差别是：在部分闸门型系统中，当分组到达自己的数据传输时，要等到下一次本节点的预约传输期才能传输，即要多增加 $m\overline{V}$ 时延，发生该事件的概率为 $\frac{\rho}{m}$。因此，Y 部分要增加 $\frac{\rho}{m}\cdot m\overline{V}=\rho\overline{V}$，代入式 (3-98) 可得部分闸门型系统的平均等待时间为

$$W=\frac{\lambda\overline{X^2}}{2(1-\rho)}+\frac{(m+\rho)\overline{V}}{2(1-\rho)}+\frac{\sigma_V^2}{2\overline{V}} \tag{3-105}$$

对于闸门型系统，当分组到达自己的预约分组传输期，比部分闸门型系统还要增加 $m\overline{V}$，发生该事件的概率为 $\frac{1-\rho}{m}$，因此 Y 部分要增加

$$\frac{1-\rho}{m}\cdot m\overline{V}=(1-\rho)\overline{V} \tag{3-106}$$

可以推出闸门型系统的平均等待时间为

$$W=\frac{\lambda\overline{X^2}}{2(1-\rho)}+\frac{(m+2-\rho)\overline{V}}{2(1-\rho)}+\frac{\sigma_V^2}{2\overline{V}} \tag{3-107}$$

2. 具有优先级的排队系统

设 M/G/1 系统的到达流具有 *n 个优先级，第* 1 类的分组流具有最高优先级，第 2 类的分组流具有次高优先级，第 n 类的分组流具有最低的优先级。从最高优先级到最低优先级的分组流的到达率分别为 $\lambda_1,\lambda_2,\cdots,\lambda_n$，服务时间的均值分别为 $\frac{1}{\mu_1},\frac{1}{\mu_2},\cdots,\frac{1}{\mu_n}$，服务时间的二阶矩分别为 $\overline{X_1^2},\overline{X_2^2},\cdots,\overline{X_n^2}$。$N_Q^k$、$W_k$ 和 ρ_k 分别表示第 k 类分组的等待队长、等待时间和利用率。

当一个分组正在传输，而有一个高优先级分组到达时，有两种处理方法：一种是允许当前正在传输的分组传输结束后再传送高优先级的分组（称为非强插优先级，nonpreemptive priority）；另一种是高优先级分组要中断当前低优先级分组的传输，待高优先级分组传输结束后，再接着传输低优先级的分组（该方法称为强插优先级，preemptive priority）。

假设各类到达过程是相互独立的 Poisson 过程，且与服务时间独立。下面分别讨论上述两种处理优先的方法。

(1) 非强插优先排队系统，设系统能够处理所有种类的业务，即

$$\rho_1+\rho_2+\cdots+\rho_n<1 \tag{3-108}$$

平均剩余服务时间为 R，仿效 P－K 公式的推导过程，来求解各类业务的等待时间。

对于最高（第 1 类）优先级队列有

$$W_1 = R + \frac{1}{\mu_1} N_Q^1 = R + \frac{1}{\mu_1} \cdot \lambda_1 W = R + \rho_1 W \qquad (3-109)$$

经整理得

$$W_1 = \frac{R}{1-\rho_1} \qquad (3-110)$$

对于第2类优先级队列有

$$W_2 = R + \frac{N_Q^1}{\mu_1} + \frac{N_Q^2}{\mu_2} + \frac{\lambda_1 W_2}{\mu_1} \qquad (3-111)$$

式中,等式右边第二项为最高优先级分组的传输时间,第三项为第2优先级队列的传输时间,第四项是已到达分组在等待过程中,新到达的最高优先级分组的传输时间,利用Little公式,上式可以写成

$$W_2 = R + \rho_1 W_1 + \rho_2 W_2 + \rho_1 W_2$$

经整理得

$$W_2 = \frac{R + \rho_1 W_1}{1-\rho_1-\rho_2} = \frac{R}{(1-\rho_1)(1-\rho_1-\rho_2)} \qquad (3-112)$$

类似地有

$$W_k = \frac{R}{(1-\rho_1-\rho_2-\cdots-\rho_{k-1})(1-\rho_1-\cdots-\rho_k)} \qquad (3-113)$$

参见图3-8,可得平均剩余服务时间为

$$R = \frac{1}{2}\sum_{i=1}^{n} \lambda_i \overline{X_i^2} \qquad (3-114)$$

代入(3-113)得

$$W_k = \frac{\sum_{i=1}^{n} \lambda_i \overline{X_i^2}}{2(1-\rho_1-\rho_2-\cdots-\rho_{k-1})(1-\rho_1-\cdots-\rho_k)} \qquad (3-115)$$

$$T_k = \frac{1}{\mu_k} + W_k \qquad (3-116)$$

利用上式可得各类分组的平均时延为

$$T = \frac{\lambda_1 T_1 + \lambda_2 T_2 + \cdots + \lambda_n T_n}{\lambda_1 + \lambda_2 + \cdots + \lambda_n} \qquad (3-117)$$

利用上述结果就可以解决如何给不同的业务分配不同的优先级,使系统的平均时延最小。在实际系统中,通常给服务时间短的用户分配高的优先级,这样可得到较好的性能。

例如:有两类业务A和B,它们的到达率分别为 λ_A 和 λ_B。其服务速率分别为 μ_A 和 μ_B,各类业务的平均时延为 $T=\frac{\lambda_A T_A + \lambda_B T_B}{\lambda_A + \lambda_B}$。如果 $\mu_A > \mu_B$,则应给

A分配较高的优先级,利用 T 的表达式,可得采用这种优先级的分配方法可使时延 T_1 小于给 B 分配高优先级时的时延 T_2。

例 3.11 设一个分组交换网发送两种类型的分组:数据分组(其到达率为 λ_2)和控制分组(其到达率为 λ_1)。数据分组用于运载业务数据,控制分组用于传送网络有关的控制信息,如:阻塞通知、故障通知和路由变更等信令分组。控制分组长度(l_1)通常大大小于数据分组(l_2)。为了保障网络畅通,通常控制分组的优先级(第1级)高于数据分组的优先级(第2级)。假设该分组网传输链路容量为 9600 bit/s,到达该链路总的分组流是到达率为 $\lambda(=\lambda_1+\lambda_2)=6.0$ 分组/秒的 Poisson 过程,且 $\lambda_1=0.2\lambda$, $\lambda_2=0.8\lambda$;$l_1=48$ bit(固定长度);l_2 的平均长度为 960 bit。试求控制分组和数据分组的平均等待时间。

解 该分组网传输链路可用具有两个优先级的非强插优先排队系统来描述。

控制分组服务时间的均值为:$\frac{1}{\mu_1}=\frac{48\text{ bit}}{9600\text{ bit/s}}=0.005\text{ s}$,其方差为 0,二阶矩为 $\overline{X_1^2}=2.5\times10^{-6}$;

数据分组服务时间的均值为:$\frac{1}{\mu_2}=\frac{960\text{ bit}}{9600\text{ bit/s}}=0.1\text{ s}$,其方差为 $2\left(\frac{1}{\mu_2}\right)^2=0.02$,二阶矩为 $\overline{X_2^2}=0.03$。

因为:
$$\rho_1=\frac{\lambda_1}{\mu_1}=0.2\times0.005\times6=0.006$$
$$\rho_2=\frac{\lambda_2}{\mu_2}=0.8\times0.1\times6=0.48$$

所以合成业务强度:$\rho=\rho_1+\rho_2=0.486$,其二阶矩为
$$\overline{X^2}=0.2\,\overline{X_1^2}+0.8\,\overline{X_2^2}\approx0.024$$

利用式(3-114)得平均剩余服务时间为
$$R=\frac{1}{2}\lambda_1\,\overline{X_1^2}+\frac{1}{2}\lambda_2\,\overline{X_2^2}\approx0.072$$

利用式(3-110)得控制分组等待时间为
$$W_1=\frac{R}{1-\rho_1}\approx0.0724\text{ s}=72.4\text{ ms}$$

利用式(3-112)得数据分组等待时间为
$$W_2=\frac{R}{(1-\rho_1)(1-\rho_1-\rho_2)}\approx140.9\text{ ms}$$

如果控制分组和数据分组传输不分优先级,则利用式(3-76)得总的平均时延为
$$W=\frac{R}{1-\rho}=140.1\text{ ms}$$

上面的计算结果表明:设定优先级后,控制分组的平均等待时延可以从140.1ms降为72.4ms,缩短了近一半;而数据分组的平均等待时延,在降低优先级后仅比原来增加了0.8ms,相比之下,这仅仅是一个微小的变化。

显然利用优先级排队系统,使高优先级信息的性能有了明显的改进,但这种改进是以牺牲低优先级信息的性能为代价的。系统总体性能需遵守守恒定律:系统的加权总等待时间等于各优先级加权等待时间之和,即

$$\sum_{k=1}^{n} \rho_k w_k = \rho W \qquad (3-118)$$

式中 W为不分优先级时系统的平均等待时间。

(2) 强插优先级排队系统。此时较高优先级业务要抢先接受服务、低优先级业务的服务被中断。只有当所有到达的高优先级业务被服务完毕后,才能继续被中断的低优先级业务的服务。强插优先级的一个典型的例子就是ATM传输系统。当多种优先级业务流的分组复接在一个ATM传输系统上传输时,一个分组通常被分为若干个ATM信元。当正在传输的一个低优先级分组的ATM信元流时,有一个高优先级分组到大,这是将暂停低优先级分组的ATM信元的传输,在高优先级分组的信元流传输结束后,再恢复低优先级分组的信元流传输。

在该系统中,直接考虑每一优先级的平均时延 T_k。

T_k 由两部分构成:①顾客的平均服务时间 $1/\mu_k$,②顾客的平均等待时间。

到达的第k优先级顾客的平均等待时间 W_k 包括两部分:一是当第k优先级用户到达时,已在系统中的第1到 *k优先级顾客需要服务时间* (W_{old}^k),二是新到达的第k优先级用户在等待过程中,新到达的第1到 $k-1$ 优先级顾客需要服务的时间 (W_{new}^k)。

$$T_k = \frac{1}{\mu_k} + W_k = \frac{1}{\mu_k} + W_{old}^k + W_{new}^k \qquad (3-119)$$

W_{old}^k 可以按照普通的无优先级M/G/1队列来求解,其到达包括第1到 *k优先级* 的用户,而忽略第k+1到 *n的用户。即*

$$W_{old}^k = \frac{R_k}{1-\rho_1-\rho_2-\cdots-\rho_k} \qquad (3-120)$$

式中

$$R_k = \frac{1}{2}\sum_{i=1}^{k} \lambda_i \overline{X_i^2} \qquad (3-121)$$

W_{new}^k 包括了第k优先级用户到达后新到达的第1到 $k-1$ 优先级用户的服务时间。

$$W_{new}^k = \sum_{i=1}^{k-1} \frac{1}{\mu_i} \cdot \lambda_i T_k = \sum_{i=1}^{k-1} \rho_i T_k \qquad k>1 \qquad (3-122)$$

经整理得：

$k=1$ 时有 $$T_1=\frac{\frac{1}{\mu_1}(1-\rho_1)+R_1}{1-\rho_1} \tag{3-123}$$

$k>1$ 时有 $$T_k=\frac{\frac{1}{\mu_k}(1-\rho_1-\cdots-\rho_k)+R_k}{(1-\rho_1-\cdots-\rho_{k-1})(1-\rho_1-\cdots-\rho_k)} \tag{3-124}$$

3.4 排队网络

前面讨论的都是单个队列，并且假定到达过程和服务间隔相互独立，而在实际的数据通信网中，每节点都有一个队列，各节点的队列组成一个排队的网络。如图3-16所示。在该图中仅给出输出链路的排队情况，而忽略了节点内部的排队。在排队的网络中，每个节点的分组到达过程与前一个队列的服务间隔（分组传输时间）紧密相关，因而就不能采用M/M/1和M/G/1的结果进行严格和有效的分析。

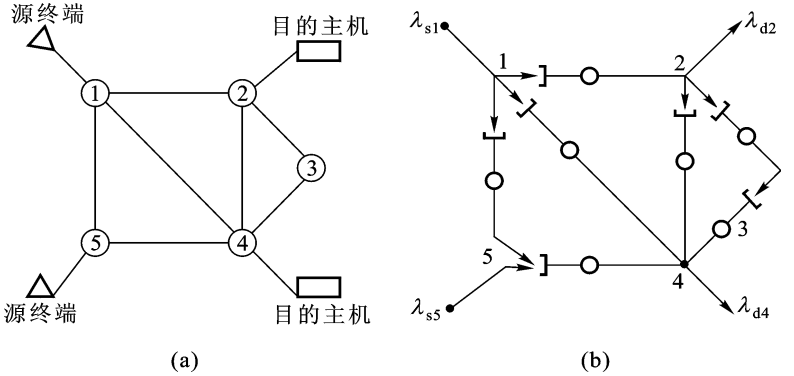


图3-16 一种5节点网络及其排队网模型
(a) 5节点网 (b) 排队网

假设有两个节点组成一个串行的网络，如图3-17所示。假定链路的容量和节点的处理能力相同。假定节点1的输入是到达率为 λ 的Possion过程。假定在前一个分组传输结束以后的一个时刻有一个长分组到达。在该长分组传输时间有一个短分组到达，则第一个节点的输出间隔将取决于分组的到达间隔和服务时间。

从图3-17可以看出，如果所有分组具有相同的长度，则节点1的队列可用M/D/1来描述。第二个队列的到达过程完全取决于第一个节点队列的输出，其

到达间隔大于$\frac{1}{\mu}$(分组的传输时间),并且在下一个分组到达时间以前,分组已传输结束,因而节点2不可能有等待队列,这就使得基于 Possion 假设的模型不适合用于第二个节点的队列。

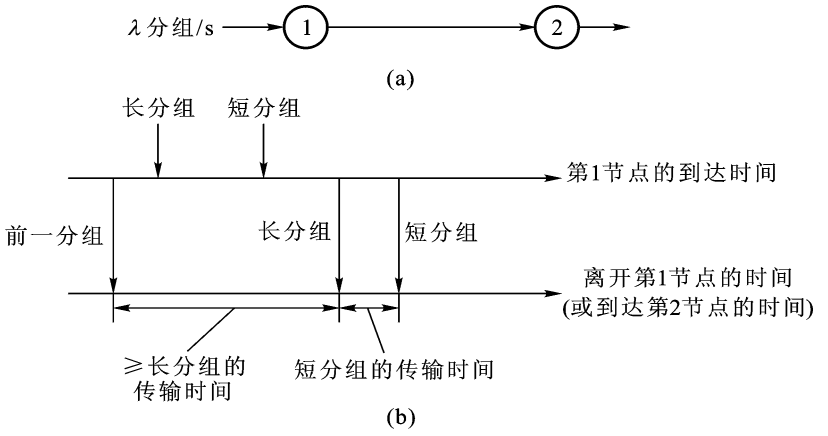


图 3-17 两个队列的串行网络

本节将讨论如何消除节点输出过程对下一节点的到达过程的影响,进而求解排队网络的性能。

3.4.1 Kleinrock 独立性近似

对于任一个网络来说,假定进入网络的分组流是服从 Possion 分布,经过网络传输后,节点输入过程的到达间隔与前一节点分组传输间隔紧密相关,从而破坏了到达过程和服务间隔相互独立的假设,这样就不能使用前面分析的 M/M/1 队列的有关结果,为了解决该问题,需要采用 Kleinrock 建议的独立性近似方法。

假设有一个网络如图 3-18 所示。分组流(某一虚电路上的分组)用 s 来表示,对于经过任一条链路(i,j)的分组到达率 λ_{ij} 由经过该链路的各分组流的到达率组成,即

$$\lambda_{ij}=\sum_{\text{所有经过}(i,j)\text{的分组流}s} x_s \qquad (3-125)$$

式中,x_s 是分组流 s 的到达率(分组/秒)。(注意,对于数据报型网络,任一对源节点和目的节点之间)。

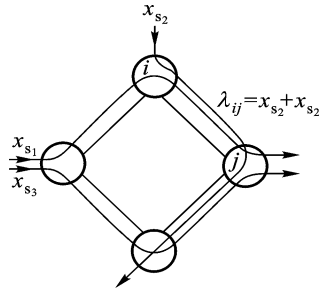


图 3-18 采用虚电路的排队网络

Kleinrock 建议,几条分组流合成的一个分组流,类似于部分恢复了到达间隔和分组长度的独立性。如果合成的分组流数目 n 较大,则到达间隔与分组长

度的依赖性将很弱。这样就可以采用 M/M/1 模型来描述每条链路,而不管这条链路上的业务与其他链路上业务的相互作用,这就是 Kleinrock 独立性近似,这对于中等到重负荷的网络是一个很好的近似。

利用 M/M/1 模型,在链路 (i, j) 上的平均分组数为

$$N_{ij} = \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} \tag{3-126}$$

式中,$\frac{1}{\mu_{ij}}$是链路 (i, j) 上的平均分组传输时间。

网络中的平均分组数为

$$N = \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} \tag{3-127}$$

应用 Little 公式,可得平均分组时延为

$$T = \frac{1}{\gamma} \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} \tag{3-128}$$

式中 γ 为系统总的到达率,即

$$\gamma = \sum_{s} x_s \tag{3-129}$$

如果各链路的处理时延和传播时延之和 d_{ij}是不可忽略的,则上式需改写为

$$T = \frac{1}{\gamma} \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} + \lambda_{ij} d_{ij} \tag{3-130}$$

对于任给一条路径 p,在该路径中的总的平均时延为

$$T_p = \sum_{p\text{上所有链路}(i,j)} \frac{1}{\mu_{ij}} \cdot \frac{\lambda_{ij}}{(\mu_{ij} - \lambda_i)} + \frac{1}{\mu_{ij}} + d_{ij} \tag{3-131}$$

式中括号内的第一项是等待时间 W_{ij},第二项是传输时延,第三项是处理时延和传播时延之和。

上面以虚电路型网络为基础对 Kleinrock 独立性近似进行了讨论。现在通过一个例子来看 Kleinrock 独立性近似在数据报网络中的适用程度。

例 3.12 假定有一个网络如图3-19所示。假定 A 沿两条链路 L_1 和 L_2 向 B 发送数据分组,链路的服务速率为 μ。节点 A 的分组到达过程是速率为 λ 的 Possion 过程,分组长度服从指数分布,且与到达间隔相互独立。试问应如何在 A 和 B 之间的两条链路上分配业务流?

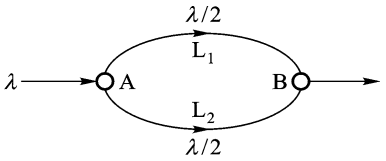


图 3-19 数据网络举例

解 假定采用两种方法:一是随机的方式 (randomization),即 A 通过扔硬币的方法来分配分组流;二是采用计量的方法 (metering),节点 A 将分组送入比特长度最短的队列 (它等于各排队分组比特长

度之和)。

在随机方式中,很容易证明 L_1 和 L_2 上的分组到达流都是 Poisson 流,且与分组长度无关。这样每一条链路都是到达率为 $\lambda/2$ 的 M/M/1 队列。利用 M/M/1队列结果,可得分组的平均时延为

$$T_R = \frac{1}{\mu - \frac{\lambda}{2}} = \frac{2}{2\mu - \lambda} \tag{3-132}$$

这种情况与 Kleinrock 的独立性近似是一致的。

在计量的方式中,到达的分组进入比特长度最短的队列,这时系统相当于一个 M/M/2 的系统,总的到达率为 λ,每条链路是一个服务员,利用前面的结果得分组的平均时延为

$$T_M = \frac{2}{(2\mu - \lambda)(1 - \rho)} \tag{3-133}$$

式中 $\rho = \frac{\lambda}{2\mu}$。

比较两种方法的平均分组时延可以发现,采用计量的方法,可使时延减少到随机方式的 $\frac{1}{(1+\rho)}$,这也进一步反映了统计复用的好处。

如果在实际系统中,要将一个比特流分解成几个可选的路由,应当采用计量的方法。

但是,采用计量的方法破坏了各个队列的 Poisson 特性,这时每条链路的到达间隔不再是指数分布,且与前面的分组长度相关。这时若采用 M/M/1 的近似,其准确度较差。

上述例题说明,采用不同的服务法则,可能会影响采用 Kleinrock 独立性近似的准确度。

3.4.2 Burke 定理

根据 Little 定理,如果能够求得网络中的平均分组数 N,就可以求得分组的平均时延。在后面的讨论中,将重点研究在排队的网络中如何求解网络中的用户数。

本小节首先讨论一个关于 M/M/m 型系统中输出过程和排队状态的定理——Burke 定理。

Burke 定理 对具有到达率为 λ 的 M/M/1, M/M/m, M/M/∞ 系统,假定系统开始时就处于稳态(或初始状态是根据稳态分布而选定的),有下列结论:

(1) 系统的离开过程是速率为 λ 的 Poisson 过程。

(2) 在时刻 t,系统中顾客数独立于 t 时刻以前用户离开系统的时间序列。

该定理说明了该类排队系统的两个特性：一是输出过程（或离开过程）仍服从 Poisson 过程；二是系统中当前顾客数与离开系统的顾客流之间的关系，它们之间相互独立。

Burke 定理的（2）直观的感觉是非常不同的。我们可能会认为：系统最近有一个非常繁忙的离开流，意味着系统现在会有大量顾客在排队，是一个繁忙系统。然而，Burke 定理表明这是不正确的，该定理指出一个繁忙的离开流，没有说明任何当前系统的状态信息。下面通过一个例题来看 Burke 定理的应用。

例 3.13 求解两个 M/M/1 队列串联后系统的状态概率。该系统的到达过程是到达率为 λ 的 Poisson 过程。这两个队列的服务时间相互独立（即相同的分组在两个节点的服务时间不同），服务时间与到达过程相互独立，如图 3－20 所示。（注意与图 3－17 的区别，在图 3－17 中两个队列的服务时间是相同的。）

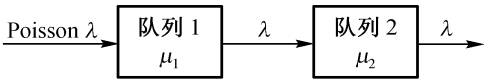


图 3－20 两个独立的M/M/1队列的串联

解 由于队列 1 是 M/M/1 队列，因此在该队列中，用户数为 n 的概率为

$$p_{n1} = \rho_1^n(1-\rho_1) \qquad (3-134)$$

式中 $\rho_1 = \lambda/\mu_1$。

由 Burke 定理的（a）可知，队列 1 的输出是速率 *λ的* Poisson 过程，并根据假定知，队列 2 的服务时间与到达过程独立，因此队列 2 可以看成为孤立的 M/M/1 队列，因而有队列 2 中用户数为 *m的概率为*

$$p_{n2} = \rho_2^m(1-\rho_2) \qquad (3-135)$$

式中 $\rho_2 = \dfrac{\lambda}{\mu_2}$。

由 Burke 定理的（b）可知，队列 1 当前的用户数与过去的离开过程相互独立，也就是与队列 2 的过去到达过程无关。所以，队列 1 当前的用户数与队列 2 当前的用户数无关。因此有：系统中队列 1 有 *n 个用户，队列* 2 有 *m 个用户的* 概率为

$$\begin{aligned} &P\{\text{队列1中有 } n \text{ 个用户，队列2中有 } m \text{ 个用户}\} \\ &= P\{\text{队列1中用 } n \text{ 个用户}\} \cdot P\{\text{队列2中用 } m \text{ 个用户}\} \\ &= p_{n1} \cdot p_{n2} = \rho_1^n(1-\rho_1)\rho_2^m(1-\rho_2) \end{aligned} \qquad (3-136)$$

由该公式可知两个串联的队列，只要满足独立性的要求，就可以看成是两个完全独立的具有相同到达率的 M/M/1 队列。

可以将上述结果作进一步的推广。

3.4.3 Jackson 定理

由前面的讨论可知，当一个分组的到达过程通过网络的第一个队列以后，分组到达将与它们的长度相关。如果这种相关性可以消除或采用随机的方法将分组分成若干个不同的路由，那么系统中的平均分组数可以通过将网络中的每个队列看成是 M/M/1 队列而导出。这是 Jackson 定理的基本结果。

设一个网络由 k 个先进先出单服务员队列组成。从网络外进入第 i 个队列的顾客流是到达率为 r_i 的独立 Poisson 过程，且在网络中至少有一个 i，使 $r_i>0$，即该网络是有外部业务输入的。一个顾客在队列 Q_i 服务结束后，将以 P_{ij} 的概率进入队列 Q_j，以 $1-\sum_{i=1}^{k} P_{ij}$ 的概率离开网络。每个队列 j 的总到达率 λ_j 是 r_j 与来自其他队列的顾客到达率之和。即

$$\lambda_j = r_j + \sum_{i=1}^{k} \lambda_i P_{ij} \qquad j=1,\cdots,k \tag{3-137}$$

对于非环形网络，λ_j 是很容易计算的（对于一般性的网络，如果某一顾客（分组）访问某一相同队列两次的概率大于 0，则需基于式（3-137）进行更复杂的运算）。式（3-137）表示具有 k 个未知数 λ_j，$j=1,\cdots,k$ 的线性系统。在给定 λ_j 和 P_{ij}（$i,j=1,\cdots,k$）的情况下，需保证式（3-137）能够有惟一解，要作一个很自然的假定，即每个顾客以概率 1 最终离开系统。

假定顾客是在第 j 个队列 Q_j 的服务时间服从均值为 $\frac{1}{\mu_j}$ 的独立的指数分布，且与该队列的到达过程独立。令 $\rho_j=\frac{\lambda_j}{\mu_j}$，$j=1,\cdots,k$。令网络的状态 n 为 $(n_1, n_2,\cdots,n_k)$。其中 n_i 表示在第 i 个队列 Q_i 中的顾客数，即 $n=(n_1,n_2,\cdots,n_k)$，$P(n)=P(n_1,n_2,\cdots,n_k)$ 表示网络状态的稳态分布，则有如下 Jackson 定理。

Jackson 定理 对上述网络，假定 $\rho_j<1$，$j=1,\cdots,k$，则对所有的 $n_1, n_2,\cdots,n_k\geq 0$ 有：

$$P(n) = P_1(n_1)\,P_2(n_2)\cdots P_k(n_k) \tag{3-138}$$

$$P_j(n) = \rho_j^{n_j}(1-\rho_j) \qquad n_j\geq 0 \tag{3-139}$$

Jackson 定理说明了若排队网络满足下列两个条件：(1) 进入网络的到达过程是 Poisson 过程；(2) 各队列的服务时间是独立的指数分布，则在数值上系统的顾客数由 k 个独立的 M/M/1 队列决定。注意该定理并没有要求到达各队列的到达过程是独立的 Poisson 过程。

下面看 Jackson 定理如何应用，再讲述各队列的到达过程不一定是 Poisson 过程的情况。

例3.14 求图3-21所示计算机系统的总任务数 N 和任务的平均时延 T。该系统是一个具有输入/输出(I/O)反馈的中心处理器(CPU)系统。任务到达过程是速率为 λ 的 Poisson 过程。假定所有的服务时间相互独立,包括相同的任务再次经过(CPU)和I/O的时间也是相互独立的。CPU 处理完的任务以概率 p_1 离开系统。

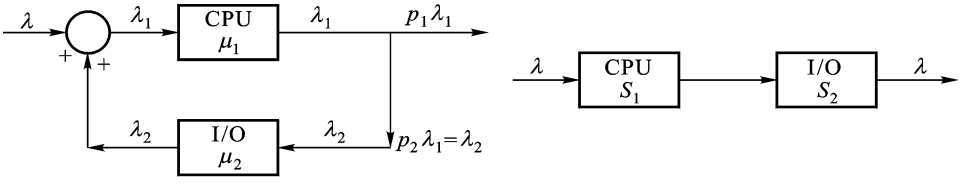


图3-21 例3-14中计算机系统

(a) 原始系统 (b) 数值上等效的系统

解 如图3-21(a)所示,$\lambda_1 = \lambda + \lambda_2$,$\lambda_2 = (1-p_1)\lambda_1 = p_2\lambda_1$,则 $\lambda_1 = \lambda/p_1$,$\lambda_2 = \lambda p_2/p_1$。

令 $\rho_1 = \dfrac{\lambda_1}{\mu_1}$,$\rho_2 = \dfrac{\lambda_2}{\mu_2}$,利用 Jackson 定理有

$$P(n) = P(n_1, n_2) = \rho_1^{n_1}(1-\rho_1) \cdot \rho_2^{n_2}(1-\rho_2) \qquad (3-140)$$

该公式在形式上等效为两个 M/M/1 队列,因而 CPU 和 I/O 队列中的平均任务数分别为:

$$N_1 = \frac{\rho_1}{1-\rho_1}$$

$$N_2 = \frac{\rho_2}{1-\rho_2}$$

系统中的总 T 任务数为

$$N = N_1 + N_2 = \frac{\rho_1}{1-\rho_1} + \frac{\rho_2}{1-\rho_2} \qquad (3-141)$$

系统中任务的平均时延为

$$T = \frac{N}{\lambda} = \frac{\dfrac{\lambda_1}{\mu_1}}{\lambda\left(1-\dfrac{\lambda_1}{\mu_1}\right)} + \frac{\dfrac{\lambda_2}{\mu_2}}{\lambda\left(1-\dfrac{\lambda_2}{\mu_2}\right)}$$

$$= \frac{\dfrac{\lambda}{p_1\mu_1}}{\lambda\left(1-\dfrac{\lambda}{p_1\mu_1}\right)} + \frac{\dfrac{\lambda p_2}{p_1\mu_2}}{\lambda\left(1-\dfrac{\lambda p_2}{p_1\mu_2}\right)}$$

$$= \frac{1}{S_1 - \lambda} + \frac{1}{S_2 - \lambda} \qquad (3-142)$$

式中：

$$S_1 = p_1 \mu_1$$

$$S_2 = \frac{p_1 \mu_2}{p_2} \qquad (3-143)$$

从上式可以看出，平均时间从数值上讲，它可以看成是两个服务速率为 S_1 和 S_2 的 M/M/1 队列的串联［如图 3-21(b) 所示］。但在实际队列中，任务在系统中停留的时间的分布与上述串联等效队列中停留时间的分布是完全不同的。为了说明该问题，设上例中 $p_1 = p_2 = \frac{1}{2}$，$\mu_1 \ll \mu_2$ 即 CPU 的服务速率比 I/O 的服务速率快得多。

在这种情况下，原始队列的半数任务不需要任何 I/O 处理，它们的平均时间要比另一半任务的平均时间要小得多。而在等效的串联队列中，CPU 的服务时间很短，系统的服务时间主要由等效的 I/O 队列来确定，显然，在实际系统和等效系统中，任务在系统中停留的时间的分布是不一样的。

Jackson 定理表明了在求解系统中的顾客数时，可以把系统看成 k 个独立的 M/M/1 队列。它要求进入网络的到达过程是 Poisson 过程，但是每一个队列的总到达过程不一定需要是 Poisson 过程。例如：如图 3-22 所示，外部到达过程是速率为 λ 的 Poisson 过程，队列的服务速率 $\mu \gg \lambda$，*经过队列的分组以 $1-p$ 的概率离开系统*。假定 p 接近于 1。对于队列而言，当有一个到达后，将会以很大的概率在很短的时间内又有一个到达（反馈到达）。对于网络而言，当有一个到达后，在很短的时间内又有新到达的概率非常小。也就是说，由于队列输出反馈到队列输入的概率很高，网络的一个到达，将触发队列有一批到达（或者说一个突发的到达串），显然，队列的到达间隔不是独立的，其总的到达过程不是 Poisson 到达。

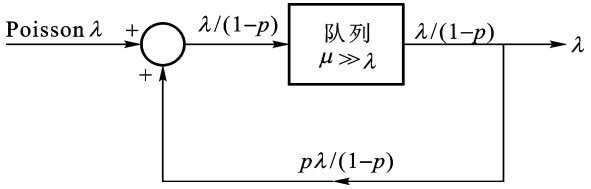


图 3-22　队列的总到达率过程不是 Poisson 过程举例

小　结

本章主要讨论了信息网络中常用的时延模型，这些模型常用于多种网络的

性能分析和评估。

首先讨论了排队系统中的基本定理——Little 定理 ($N=\lambda T$) *及其多种变形*表示和应用。

接着讨论了单一排队模型。包括两类排队模型：一类是 M/M/1, M/M/m, M/M/m/m, M/M/∞这一类排队模型。采用的基本分析方法是根据 Markov 链的状态转移图和平衡方程，求解系统的状态概率，进而求出 W、T 和 N_Q 等参量；另一类是 M/G/1 排队模型，采用的基本分析方法是利用推导 P－K 公式的分析法——平均剩余服务时间的求解法。

最后讨论了排队网络的性能，其基本出发点是如何消除各队列相关性，主要的方法是 Kleinrock 和 Jackson 定理。

习 题

3.1 设顾客到达一个快餐店的速率为每分钟 5 人，顾客等待他们需要的食品的平均时间为 5 min，顾客在店内用餐的概率为 0.5，带走的概率为 0.5。一次用餐的平均时间为 20 min。问快餐店内的平均顾客数是多少？

3.2 设有两个通信节点 1 和 2 向另一个节点 3 发送文件。文件从 1 和 2 到 3 所需的平均传输时间分别为 R_1 和 R_2 个单位时间。节点 3 处理节点 i (i=1,2) 的文件所需的平均时间为 P_i 个单位时间，在处理结束后再向节点 1 或 2 请求另一个文件。(具体选择节点的规则未定)。如果 λ_i 是节点 i 以每单位时间内发送的文件数表示的通过率，试求所有可行的通过率对 (λ_1, λ_2) 的区间？

3.3 一个健忘的教授将与两个学生的会谈的时间安排在相同时间，设会谈的区间是独立的，服从均值为 30 min 的指数分布。第一个学生准时到达，第二个学生晚到 5 min，问第一个学生到达时刻到第二个学生离开的平均间隔是多少？

3.4 一个通信链路分成两个相同的信道，每一个信道服务一个分组流，所有分组具有相等的传输时间 T 和相等的到达间隔 R (R> T)。*假如改变信道的*使用方法，将两个信道合并成一个信道，将两个业务流统计复接到一起，每个分组的传输时间为 $\frac{T}{2}$。试证明一个分组在系统内的平均时间将会从 T 下降到 $\frac{T}{2}\sim\frac{3T}{4}$，分组在队列中等待的方差将会从 0 变为 $\frac{T^2}{16}$？

3.5 一个通链路的传输速率为 50 kb/s，用来服务 10 个 session，每个 session 产生的 Poisson 业务流的速率为 150 分组/分钟，分组长度服从指数分布，其均值为 1000 bit。

⑴ 当该链路按照下列方式为session服务时,对于每一个session,求在队列中的平均分组数,在系统中的平均分组数,分组的平均延时。

①10个相等容量的时分复用信道;

②统计复用。

⑵ 在下列情况下重做⑴:

①5个session发送的速率为250分组/分钟;

②另5个session发送的速率为50分组/分钟。

3.6 考察一个到达率及服务速率与服务系统状态相关的类似于M/M/1的系统。设系统中的顾客数为 n,除了到达率为 λ_n,服务速率为 μ_n 外与M/M/1完全相同,试证明 $P_{n+1}=(\rho_0\cdots\rho_n)\ p_0$,式中 $\rho_k=\frac{\lambda_k}{\mu_{k+1}}$ 及 $p_0=1+\sum_{k=0}^{\infty}(\rho_0\cdots\rho_k)^{-1}$。

3.7 考察一个离散型M/M/1系统,该系统的到达间隔和服务时间均为整数值,即顾客在整数时刻到达或离开。令 λ 是一个到达发生在任何时刻 k 的概率,并假定每次最多仅有一个到达。一个顾客在 $k+1$ 时刻被服务结束的概率为 μ。试求以 λ 和 μ 表示的系统状态(顾客数)概率分布 p_n。

3.8 设有一个M/M/∞队列,其服务员分别标有1,2,…。现增加一个限制,即一个顾客到达时将选择一个空闲的且具有最小编号的服务员。试求每一个服务员是忙的时间比例。如果服务员数目是有限的,答案有无变化?

3.9 假定在M/M/2队列中,两个服务员具有不同的服务速率,试求系统的稳态分布。(当系统为空时,到达的顾客分配到服务速率较快的服务员。)

3.10 设有 M 个顾客,m 个服务台,缓冲器的容量为 K 的排队系统,到达速率和服务速率为:

$$\lambda_k=\begin{cases}\lambda(M-k) & 0\le k\le K-1\\ 0 & \text{其他}\end{cases}$$

$$\mu_k=\begin{cases}k\mu & 0\le k\le m\\ m\mu & k\ge m\end{cases}$$

假设到达过程为Poisson过程,服务时间为指数分布,且 $M\ge K=m$。画出状态转换图。求该排队系统中顾客数的稳态分布,平均时延和阻塞概率。

3.11 M/M/m/m排队是在电路交换应用中产生的。这里设呼叫到达过程为Poisson过程,它由最大值为 m 个指数分布的服务台服务。当系统中有 m 个呼叫时,第 $(m+1)$ 个呼叫被阻塞。设系统的状态 l 表示当前正在进行的呼叫数。到达和服务速率为

$$\lambda_l=\begin{cases}\lambda & l<m\\ 0 & l>m\end{cases}$$

其中，$\mu_l = l\mu$, $l=1,2,\cdots,m$，求系统中呼叫个数的稳态分布，阻塞概率 B_l 和呼叫等待时间的期望值 $E(W)$。

3.12 设一条传输链路有 *m 个等容量的电路组成，有两种类型的* session，其 Poisson 到达率分别为 λ_1 和 λ_2。当所有电路都忙时，一个到达的 session 将被拒绝而离开系统，否则一个到达的 session 被分配到任一个空闲的电路。两种类型的服务时间（保持时间）服从指数分布，其均值分别为 $\frac{1}{\mu_1}$ 和 $\frac{1}{\mu_2}$，求该系统的稳态阻塞概率。

3.13 试利用平均剩余服务时间的概念证明 M/D/1 系统的等待时间为 $W=\frac{\rho}{2\mu(1-\rho)}$。

3.14 在 M/G/1 系统中，试证明：

$$P\{\text{系统空闲}\}=1-\lambda\overline{X}$$

$$\text{忙区间之间的平均长度}=\frac{1}{\lambda}$$

$$\text{忙区间的平均长度}=\frac{\overline{X}}{1-\lambda\overline{X}}$$

$$\text{在一个忙区间内服务的平均顾客数}=\frac{1}{1-\lambda\overline{X}}$$

3.15 考察一个有单一休假期的 M/G/1 系统，即在每个忙区间后跟有一个休假期。一旦这个休假期结束，到达的顾客进入空闲系统立即得到服务。假定休假的区间是独立同分布的且与用户的到达间隔和服务时间独立。试证明队列中的平均等待时间为

$$W=\frac{\lambda\overline{X^2}}{2(1-\rho)}+\frac{\overline{V^2}}{2I}$$

这里 I 是空闲周期的平均长度。试说明如何计算 I。

3.16 考察一个服务受限的系统，对于闸门型和部分闸门型，试证明：

(1) 在一个预约区间内一个分组到达的稳态概率为 $1-\rho$。

(2) 一个预约期间后跟一个空闲数据区间的稳态概率为 $\frac{1-\rho-\lambda\overline{V}}{1-\rho}$。

3.17 有一个网络如图 3-23 所示，有四个 session ACE, ADE, BCEF 和 BDEF，它们发送的 Poisson 业务的速率分别为 100, 200, 500 和 600 分组/分钟，分组的长度是均值为 1000 bit 的指数分布，所有传输链路的容量均为 50 kb/s。每条链路的传输时延为 2 ms，利用 Kleinrock 的独

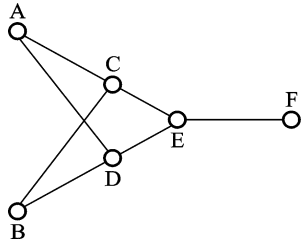


图 3-23 习题 3.17 中的网络

立性近似。试求解系统中的平均分组数，分组的平均时延（不区分 session），以及每个 session 中分组的平均时延。

3.18　设有一个 CPU 连接到 m 个 I/O 设备，如图 3-24 所示，任务进入系统是服从速率为 λ 的 Poisson 过程，通过 CPU 后分别以概率 p_i，$i=1,\cdots,m$ 分送到第 i 个 I/O 设备，而以概率 p_0 离开系统。任务在 CPU 和第 i 个 I/O 设备内的服务时间分别服从均值为 $\frac{1}{\mu_0}$ 和 $\frac{1}{\mu_i}$ 的指数分布。假定在所有队列中所有任务的服务时间是相互独立的，试求系统的稳态状态概率分布，并构造一个具有相同分布的“等效”的 m+1 个队列级联的系统。

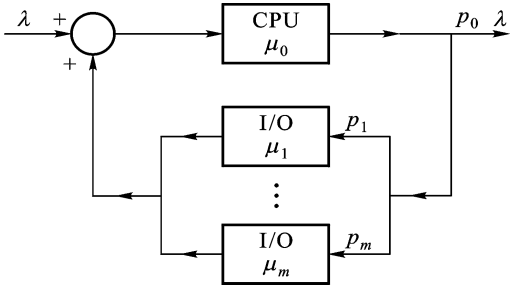


图 3-24　习题 3.18 的图

第4章 多址技术

4.1 多址协议概述

网络中的终端设备通过通信子网来访问网络中的资源。当多个终端同时访问同一资源（如共享的通信信道）时，就可能会产生信息碰撞，导致通信失败。典型的共享链路有：卫星链路和蜂窝移动通信系统的链路、局域网、分组无线电网等，如图4-1所示。在卫星和蜂窝移动通信系统中，多个用户采用竞争或预约分配等方法向一个中心站（卫星或移动通信系统中的基站）发送信息，中心站通过下行链路（中心站到用户的链路）发送应答信息。在局域网中，一个用户发送，所有用户都可以接收到，它是一个全连通的网络，其典型网络是以太网（Ethernet）。在分组无线电网络中，用户分布在一个很广的范围内，每个用户仅能接收到其通信范围以内的信息，任意两个用户之间可能需要多次中转才能相互交换信息，它是一个部分连通（或称为多跳）的网络。在上述网络中，如果多个用户同时发送时，就会发生多个用户的帧在物理信道上相互重叠（即碰撞），可能使得接收端无法正确接收。

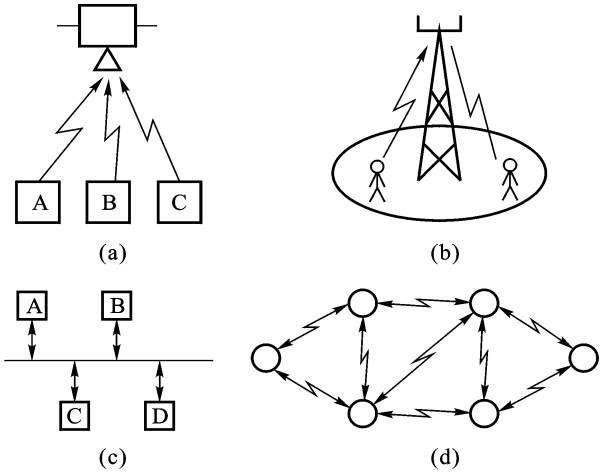


图4-1 共享链路网络

(a) 卫星通信系统 (b) 移动通信 (c) 局域网 (d) 分组无线电网络

为了有效地进行通信，就需要有某种机制来决定资源的使用权，这就是网络

的多址接入控制问题。所谓多址接入协议(multiple access protocol)就是在一个网络中,解决多个用户如何高效共享一个物理链路资源的技术。

4.1.1 MAC层在通信协议中的位置

从分层的角度来看,多址技术是数据链路层的一个子层。多址接入控制MAC层(Medium Access Control)在通信协议中的位置如图4-2所示。它处于数据链路逻辑控制层下方,物理层的上方。MAC层将有限的资源分配给多个用户,从而使得在众多用户之间实现公平、有效地共享有限的带宽资源;实现各用户之间良好的连通性,获得尽可能高的系统吞吐量以及尽可能低的系统时延。逻辑链路控制(LLC)子层为本节点提供了到其邻节点的"链路",而如何协调本节点和其他节点有效地共享带宽资源,是媒质接入控制子层——MAC层的主要功能。

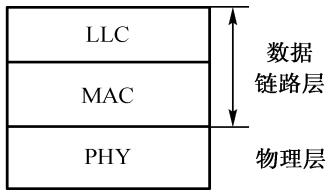


图4-2 MAC层在通信协议中的位置

4.1.2 多址协议的分类

多址协议主要分为固定分配多址接入协议、随机分配多址接入协议和基于预约方式的多址接入协议。所谓固定分配多址接入是指在用户接入信道时,专门为其分配一定的信道资源(如频率、时隙、码字或空间),用户独享该资源,直到通信结束。所谓随机多址接入是指用户可以随时接入信道,并且可能不会顾及其他用户是否在传输。当信道中同时有多个用户接入时,在信道资源的使用上就会发生冲突(碰撞)。因此,对于有竞争的多址接入协议如何解决冲突,从而使所有碰撞用户都可以成功进行传输是一个非常重要的问题。所谓基于预约的多址接入协议,是指在数据分组传输之前,先进行资源预约。一旦预约到资源(如频率、时隙),则在该资源内可进行无冲突的传输。可用图4-3来描述多址接入协议的分类。

4.1.3 系统模型

从排队论的观点出发,多址信道可以看成一个多进单出的排队系统(即该系统有多个输入而仅仅有一个输出)。每一个节点都可以独立的产生分组,而信道则相当于服务员,它要为各个队列服务。由于各个排队队列是相互独立的,各节点无法知道其他队列的情况,服务员也不知道各个队列的情况,所以增加了系统的复杂性。如果可以通过某种措施,使各个节点产生的分组在进入信道之前排列成一个总的队列,然后由信道来服务,则可以有效地避免分组在信道上的碰撞,大大提高信道的利用率。如图4-4所示。

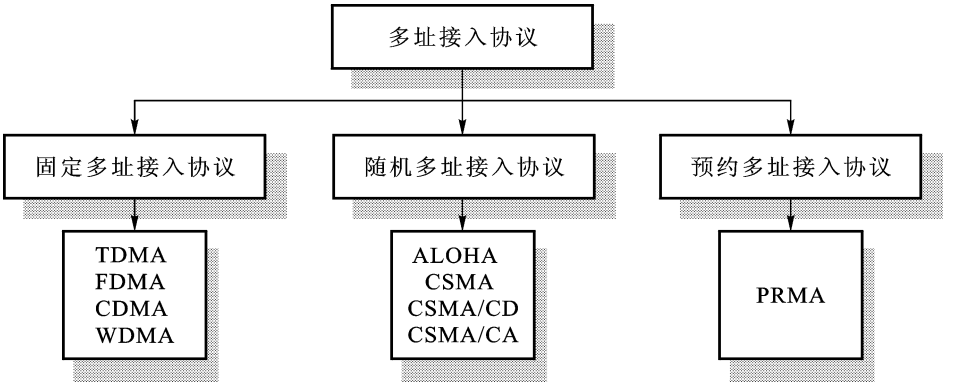


图 4-3 多址接入协议的分类

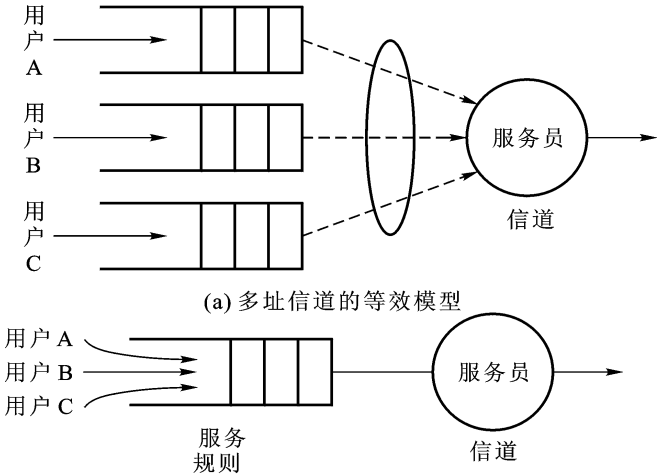


图 4-4 多址技术示意图

为了能够有效地分析多址接入协议,必须根据应用环境做一些假设。在讨论每种多址协议时,应该考虑下列问题:

1 网络的连通特性。通常将网络按其连通模式分为:单跳、两跳及多跳网络。所谓单跳网络是指网络中所有的节点都可以接收到其他节点发送的数据;所谓两跳网络是指网络中的部分节点之间不能直接通信,需要经过一次中继才能通信;而所谓多跳网络是指网络中源节点和目的节点之间的通信可能要经过多次中继。多跳网络既可以是有线网络,也可以是无线网络。在无线通信网络中,通信节点之间的有效通信距离是由发端的发送功率、节点之间的距离以及接收机灵敏度等条件决定的。本章主要讨论对称的信道,即任意两个在通信距离内的节点都可以有效的和对方进行通信。

2 同步特性。通常用户是可以在任意时刻接入信道,但也可以以时隙为基

础接入信道。在基于时隙的系统中,用户只有在时隙的起点才能接入信道。在这种系统中,要求全网有一个统一的时钟,同时将时间轴划分成若干个相等的时间段,称之为时隙。系统中所有数据的传输开始点都必须在一个时隙的起点。

3. 反馈和应答机制。反馈信道是用户获得信道状态的途径。在本章的讨论中,假设用户(节点)可以获得信道的反馈信息,即信道是空闲、碰撞还是进行了一次成功传输。

4. 数据产生模型。所有的用户都按照泊松过程独立地产生数据。

4.2 固定多址接入协议

固定多址接入协议又称为无竞争的多址接入协议或静态分配的多址接入协议。固定多址接入为每个用户固定分配一定的系统资源,这样当用户有数据发送时,就能不受干扰地独享已分配的信道资源。固定多址接入的优点在于可以保证每个用户之间的"公平性"(每个用户都分配了固定的资源)以及数据的平均时延。典型的固定多址接入协议有:频分多址(FDMA)、时分多址(TDMA)、码分多址(CDMA)及空分多址(SDMA)等。在本节中重点讨论时分多址和频分多址系统。

4.2.1 频分多址接入

频分多址——Frequency Division Multiple Access(FDMA)是把通信系统的总频段划分成若干个等间隔的频道(或称信道),并将这些频道分配给不同的用户使用,这些频道之间互不交叠,见图4-5。

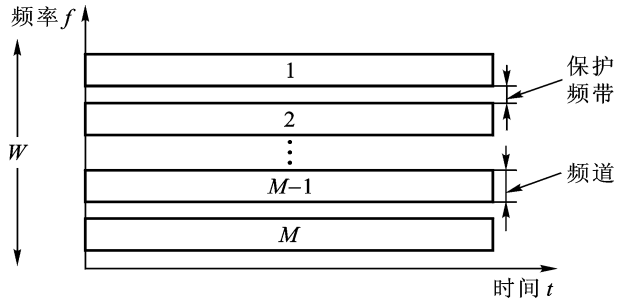


图4-5 频分多址的基本原理

FDMA的最大优点是相互之间不会产生干扰。当用户数较少且数量大致固定、每个用户的业务量都较大时(比如在电话交换网中),FDMA是一种有效的分配方法。但是,当网络中用户数较多且数量经常变化,或者通信量具有突发性的特点时,采用FDMA就会产生一些问题。最显著的两个问题是:当网络中

的实际用户数少于已经划分的频道数时，许多宝贵的频道资源就白白浪费了；而且当网络中的频道已经分配完后，即使这时已分配到频道的用户没有进行通信，其他一些用户也会因为没有分配到频道而不能通信。

4.2.2 时分多址接入

时分多址——Time Division Multiple Access (TDMA) 也是一种典型的固定多址接入协议。TDMA 多址接入协议将时间分割成周期性的帧，每一帧再分割成若干个时隙（无论帧或时隙都是互不重叠的），然后根据一定的时隙分配原则，使每个用户只能在指定的时隙内发送。其工作原理如图 4-6 所示。

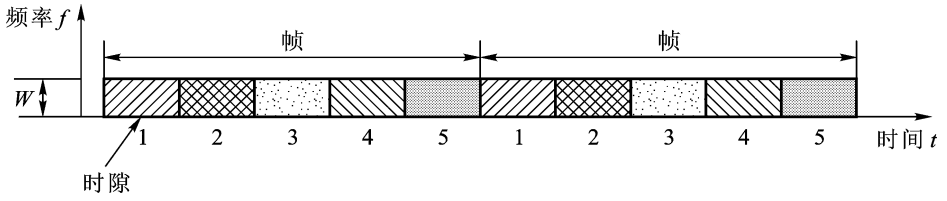


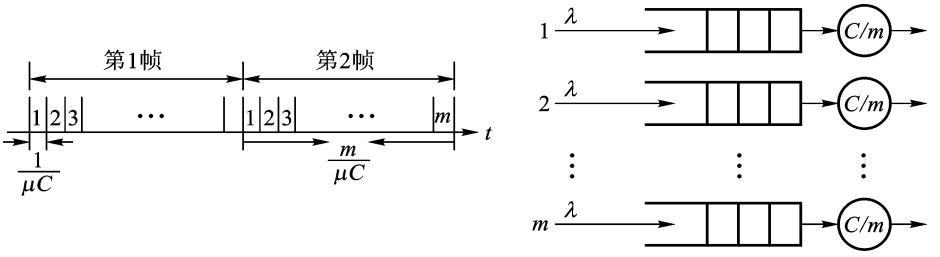
图 4-6 TDMA 时隙分配

在时分多址的系统中，用户在每一帧中可以占用一个时隙，如果用户在已分配的时隙上没有数据传输，则这段时间将被浪费。

4.2.3 固定多址接入协议的性能分析

FDMA 和 TDMA 同属于固定的多址接入技术，两者的工作原理和系统性能基本相似。首先从 TDMA 着手分析其性能，然后再讨论两者的差别。

我们讨论一个由 m 个用户组成的 TDMA 系统。设共享信道的总容量为 C (bit/s)，每个用户的分组到达率为 λ (分组/秒)，分组的固定长度为 $\frac{1}{\mu}$ (bit)。如图 4-7 (a) 给出了每帧的时隙分配，图 (b) 给出了系统的模型。



(a) (b)

图 4-7 时分多址系统的性能分析模型

(a) 帧的时隙分配 (b) m 个 M/D/1 排队模型

因为每时隙等长,所以系统构成了 m 个独立的 M/D/1 排队模型。如果分组经过系统的时延为 T,则 T 由三部分组成:

① 分组的发送时间 $\tau=\frac{1}{\mu C}$,即一帧内一个时隙的宽度;

② 分组的排队时延 W;

③ 等待帧内服务时隙的时延。对于泊松到达,在稳态下该时延可取为半个帧长,即 $\frac{m\tau}{2}$。

由排队论的 M/D/1 系统分析可知,分组在队列中的排队时延为

$$W=\frac{\rho}{2(1-\rho)\mu C_0} \tag{4-1}$$

注意,这里等效的信道容量 C_0 等于 $\frac{C}{m}$,代入上式可得

$$W=\frac{\rho m\tau}{2(1-\rho)} \tag{4-2}$$

其中

$$\rho=\frac{m\lambda}{\mu C}=m\lambda\tau \tag{4-3}$$

定义系统的吞吐量 *S为在单位时间内系统实际传输业务量与信道允许的*最大业务量之比。则 m 个用户总的平均数据到达率为 $m\lambda$,则信道允许的最大业务量为 μC,则有

$$S=\frac{m\lambda}{\mu C}=m\lambda\tau=\rho \tag{4-4}$$

可见,该系统的吞吐量等于系统的总业务强度,最大可达 100%。

将式 (4-4) 代入式 (4-1),可得

$$W=\frac{mS\tau}{2(1-S)} \tag{4-5}$$

因此,可得分组的平均时延为

$$T=\tau+\frac{m\tau}{2}+\frac{mS\tau}{2(1-S)} \tag{4-6}$$

为了方便后面比较性能,用 τ 对 T 进行归一化,得归一化的时延 D 为

$$D=1+\frac{m}{2}+\frac{mS}{2(1-S)} \tag{4-7}$$

当 m 很大 (譬如 $m\geqslant 20$),式 (4-7) 可简化为

$$D=m\left(\frac{1}{2}+\frac{S}{2(1-S)}\right)$$

可见,不管 S 取值多少 ($S\leqslant 1$),数据分组时延随 m 增大而上升。

为了获得与上述 TDMA 系统相对应的 FDMA 系统参数,将信道容量 (最大

数据速率）C折算成信道总带宽 R。m路数据源分别固定使用一个子信道，每个子信道带宽为$\frac{R}{m}$。这样，FDMA 系统也构成如图 4-7(b) 所示的 m个M/D/1系统。与 TDMA 不同的是，TDMA 的每一路信号占一个时隙，而 FDMA 的每一路信号占一个子频带。如果在两种系统中对数据信号采用相同的调制方式，则按上述折算方法所得的时分和频分两种复接系统的资源是完全等价的。在相同的输入条件，FDMA 系统在两个方面与 TDMA 有差别：①FDMA 没有半个帧的等待服务时延；②FDMA 的每个分组传输时间比 TDMA 大 m 倍，即 $\tau=\frac{m}{\mu C}$。由此可得到 FDMA 的分组时延为

$$T = m\tau + \frac{m\tau S}{2(1-S)} \tag{4-8}$$

仍用 τ 归一化，可得 FDMA 的归一化时延公式为

$$D = m\left(1+\frac{S}{2(1-S)}\right) = \frac{m(2-S)}{2(1-S)} \tag{4-9}$$

比较式 (4-7) 和式 (4-9)，可以得出

$$D_{\mathrm{FDMA}} = D_{\mathrm{TDMA}} + \frac{m}{2} - 1 \qquad (m \geq 2) \tag{4-10}$$

上式说明：当 m≥2，FDMA 系统的分组时延总是大于 TDMA 系统的一个固定值 $\frac{m}{2}-1$，它与网络负荷无关。其结果如图 4-8 所示。由图 4-8 可以看出，D 的最小值是 2。当 m=2 时，TDMA 与 FDMA 的性能相同，两曲线重合。m 值越大，两者的差别就越大。

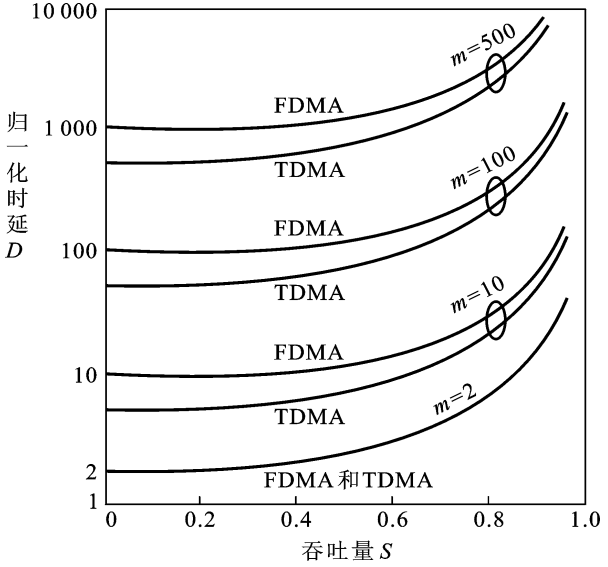


图 4-8 FDMA 和 TDMA 的时延-吞吐量特性

从上面的讨论和分析可以看出,传统的固定多址接入协议不能有效地处理用户数量的可变性和通信业务的突发性,因此,将进一步讨论随机接入的多址协议。

4.3 随机多址接入协议

随机多址协议又叫做有竞争的多址接入协议。网络中的节点在网络中的地位是等同的,各节点通过竞争获得信道的使用权。随机多址接入协议又可细分为完全随机多址接入协议(ALOHA 协议)和载波侦听型多址接入协议。不论是哪种随机多址接入协议,主要关心两个方面的问题:一个是稳态情况下系统的通过率和时延性能,另一个是系统的稳定性。

4.3.1 ALOHA 协议

ALOHA 协议是 20 世纪 70 年代 Hawaii 大学建立的在多个数据终端到计算中心之间的通信网络中使用的协议。其基本思想是:若一个空闲的节点有一个分组到达,则立即发送该分组,并期望不会和其他节点发生碰撞。

为了分析随机多址接入协议的性能,假设系统是由 m 个发送节点组成的单跳系统,信道是无差错及无捕获效应的信道,分组的到达和传输过程满足如下假定:

(1) 各个节点的到达过程为独立的参数为 $\frac{\lambda}{m}$ 的 Poisson 到达过程,系统总的到达率为 λ。

(2) 在一个时隙或一个分组传输结束后,信道能够立即给出当前传输状态的反馈信息。反馈信息为"0"表明当前时隙或信道无分组传输,反馈信息为"1"表明当前时隙或信道仅有一个分组传输(即传输成功),反馈信息为"e"表明当前时隙或信道有多个分组在传输,即发生了碰撞,导致接收端无法正确接收。

(3) 碰撞的节点将在后面的某一个时刻重传被碰撞的分组,直至传输成功。如果一个节点的分组必须重传,则称该节点为等待重传的节点。

(4) 对于节点的缓存和到达过程作如下假设:

① 假设 A:无缓存情况。在该情况下,每个节点最多容纳一个分组。如果该节点有一个分组在等待传输或正在传输,则新到达的分组被丢弃且不会被传输。在该情况下,所求得的时延是有缓存情况下时延的下界(Low Bound)。

② 假设 B:系统有无限个节点($m=\infty$)。每个新产生的分组到达一个新的节点。这样网络中所有的分组都参与竞争,导致网络的时延增加。因此,在该假设情况下求得的时延是有限节点情况下的时延上界(Up Bound)。

如果一个系统采用假设A或假设B分析的结果类似,则采用这种分析方法就是对具有任意大小缓存系统性能的一个很好的近似。

1. 纯ALOHA协议

纯ALOHA协议是最基本的ALOHA协议。只要有新的分组到达,就立即被发送并期望不与别的分组发生碰撞。一旦分组发生碰撞,则随机退避一段时间后进行重传。图4-9给出了一个简单的纯ALOHA协议的例子。

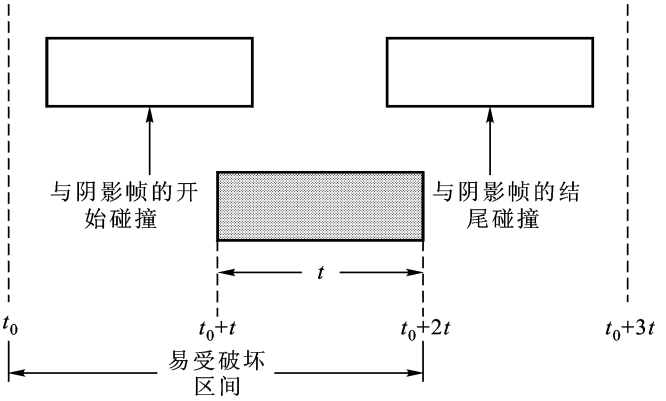


图4-9 纯ALOHA协议工作原理图

如果从数据分组开始发送的时间起点到其传输结束的这段时间内,没有其他数据分组发送,则该分组就不会和其他分组发生碰撞。如图4-9所示,在什么情况下图中阴影部分表示的数据分组(在 t_0+t 时刻产生的分组)可以不受任何干扰的发送呢?为了便于分析,假设系统中所有分组的长度相等,传输数据分组所需的时间定义为系统的单位时间,为了简化描述,令该值等于t,并在下面的分析中令其等于1)。从图中可以看到,如果在 t_0 到 t_0+t 时间内,其他用户产生了数据分组,则该分组的尾部就会和阴影分组的头部碰撞;同样,在 t_0+t 和 t_0+2t 之间产生的任何分组都将和阴影分组的尾部发生碰撞。时间区间 $[t_0, t_0+2t]$ 称为阴影分组(在 t_0+t 时刻产生的分组)的易受破坏区间。

很显然,在纯ALOHA协议中,只有在数据分组的易受破坏区间内没有其他分组传输,则该分组可以成功传输。为了分析方便,设系统有无穷多个节点(假设B),假定重传的时延足够随机,重传分组和新到达分组合成的分组流是到达率为 G 的Poisson到达过程。则在纯ALOHA系统中,一个分组成功传输的概率就是在其产生时刻前一个时间单位内没有分组发送,并且在该分组产生时刻的后一个时间单位内也没有分组发送的概率,即在该分组产生时刻前后两个时间单位内没有其他分组发送的概率。

根据Poisson公式,在单位时间内,产生 k 个分组的概率是

$$P(k) = \frac{e^{-G}G^k}{k!} \tag{4-11}$$

则根据上面的分析可以得到在纯 ALOHA 系统中，分组成功传输的概率

$$P_{succ} = P\{\text{在两个时间单位内没有其他分组发送的概率}\} = P(0)$$

$$= \frac{e^{-2G}(2G)^0}{0!} = e^{-2G} \tag{4-12}$$

因此，系统的通过率

$$S = G \cdot P_{succ} = Ge^{-2G} \tag{4-13}$$

对上式求最大值，可得系统的最大通过率为$\frac{1}{2e}\approx 0.184$，对应的 $G=0.5$。

2. 时隙 ALOHA 协议

从前面的描述中可以看到，在纯 ALOHA 协议中，节点只要有分组就发送，易受破坏区间为两个单位时间。如果缩小易受破坏区间，就可以减少分组碰撞的概率，提高系统的利用率。基于这一出发点，提出了时隙 ALOHA 协议。

时隙 ALOHA 系统将时间轴划分为若干个时隙，所有节点同步，各节点只能在时隙的开始时刻才能够发送分组，时隙宽度等于一个分组的传输时间，如图 4-10所示。从图 4-10 中可以看出，当一个分组到达某时隙后，它将在下一时隙开始传输，并期望不会与其他节点发生碰撞。如果在某时隙内仅有一个分组到达（包括新到达的分组和重传分组的到达），则该分组会传输成功。如果在某时隙内到达两个或两个以上的分组，则将会发生碰撞。碰撞的分组将在以后的时隙中重传。很显然，此时的易受破坏区间长度减少为一个单位时间（时隙）。

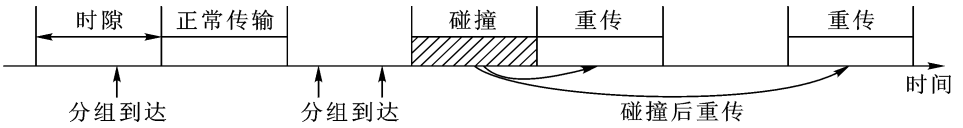


图 4-10 时隙 ALOHA 系统

利用前面的假设条件，并假定系统有无穷多个节点（假设 B）。从图 4-10 中可以看出在一个时隙内到达的分组包括两个部分：一部分是新到达的分组，另一部分是重传的分组。设新到达的分组是到达率为 λ（分组数/时隙）的 Poisson 过程。假定重传的时延足够随机化，这样就可以近似地认为重传分组的到达过程和新分组的到达过程之和是到达率为 $G(>\lambda)$ 的 Poisson 过程。则在一个时隙内有一个分组成功传输的概率为 Ge^{-G}，它被定义为系统的通过率（S）（或离开系统的速率），即

$$S = Ge^{-G} \tag{4-14}$$

其结果如图 4-11 所示。如果分组的长度为一个时隙宽度，则系统的通过率就是指在一个时隙内成功传输所占的比例（或有一个分组成功传输的概率）。其最

大通过率为$\frac{1}{e}\approx0.368$,对应的 $G=1$。很明显,时隙 ALOHA 的最大通过率是纯 ALOHA 系统最大通过率的 2 倍。

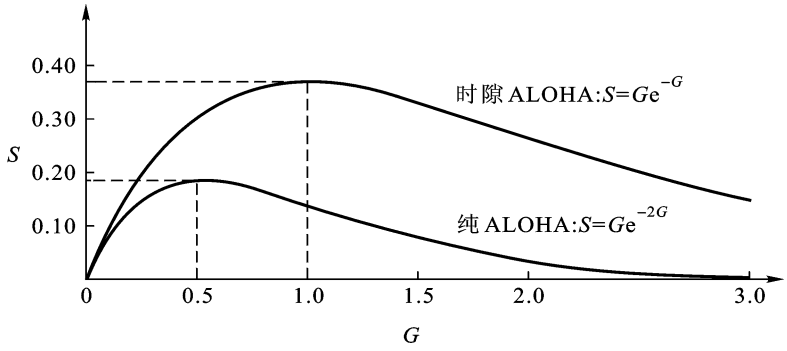


图 4-11 ALOHA 协议的通过率曲线

例 4.1 若干个终端用纯 ALOHA 随机接入协议与远端主机通信。信道速率为 2.4 kb/s。每个终端平均每 3 min 发送一个帧,帧长为 200 bit,问系统中最多可容纳多少个终端?若采用时隙 ALOHA 协议,其结果又如何?

解 设可容纳的终端数为 N。*每个终端发送数据的速率是*$\frac{200}{3\times60}$ bit/s$\approx$1.1bit/s,由于纯 ALOHA 系统的最大系统通过率为$\frac{1}{2e}$,则有 $N=\frac{2400\times\frac{1}{2e}}{1.1}\approx$396 (个)。

若采用时隙 ALOHA 协议,因为时隙 ALOHA 系统的最大系统通过率为$\frac{1}{e}$,则有 $N=\frac{2400\times\frac{1}{e}}{1.1}\approx793$ (个)。

3. 时隙 ALOHA 协议稳定性分析

从图 4-11 可以看出,对于时隙 ALOHA 系统,当 $G<1$ 时,系统空闲的时隙数较多;当 $G>1$ 时,碰撞较多,从而导致系统性能下降。因此,为了达到最佳的性能,应当将 G 维持在 1 附近变化。

当系统达到稳态时,应该有新分组的到达率等于系统的离开速率,即有 $S=\lambda$。则将 $S=\lambda$*的曲线与对应的通过量曲线相交,可以看到在对应的* Ge^{-G} 曲线上有两个平衡点 (如图 4-12 所示)。

由前面的讨论无法判定这两个平衡点中哪个是稳定的、哪个是不稳定的。因此,将通过对时隙 ALOHA 系统动态行为的分析,来了解系统的稳定性及其控制方法。

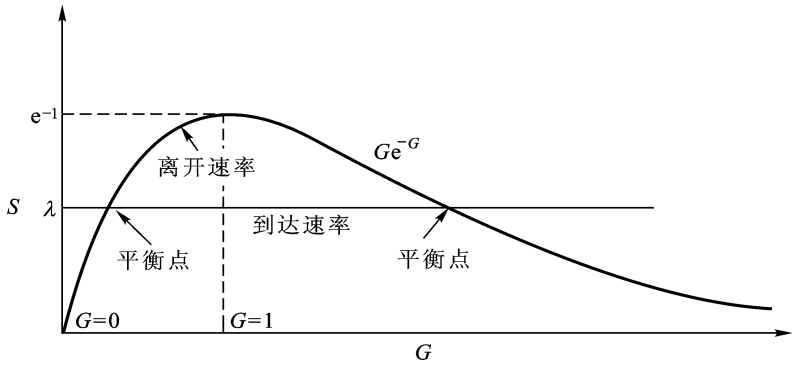


图 4-12 时隙 ALOHA 协议稳态时的平衡点

为了分析系统的动态行为,先采用假设 A(无缓存的情况)来进行讨论。时隙 ALOHA 的行为可以用离散时间马尔可夫链来描述,其系统的状态为每个时隙开始时刻等待重传的节点数。令

q_r:在碰撞后等待重传的节点在每一个时隙内重传的概率;

n:在每个时隙开始时刻等待重传的节点数;

m:系统中的总用户数;

q_a:每个节点有新分组到达的概率;

λ:m 个节点的总到达率,即每个节点的到达率为 $\frac{\lambda}{m}$,其单位为分组数/时隙;

$Q_r(i,n)$:n 个等待重传的节点中,有 i 个节点在当前时隙传输的概率;

$Q_a(i,n)$:在当前时隙中,$m-n$ 个空闲节点中有 i 个新到达的分组传输的概率。

显然,每个节点有新分组到达的概率 $q_a=1-e^{-\frac{\lambda}{m}}$。在给定 n 的条件下,有:

$$Q_r(i,n)=\binom{n}{i}(1-q_r)^{n-i}q_r^i \tag{4-15}$$

$$Q_a(i,n)=\binom{m-n}{i}(1-q_a)^{m-n-i}q_a^i \tag{4-16}$$

令 $P_{n,n+i}$ 表示时隙开始时刻有 n 个等待重传的节点,到下一时隙开始点有 $n+i$ 个等待重传节点的转移概率。其状态转移概率为

$$P_{n,n+i}=\begin{cases}Q_a(i,n) & 2\le i\le m-n \quad (4-17a)\\ Q_a(1,n)[1-Q_r(0,n)] & i=1 \quad (4-17b)\\ Q_a(1,n)Q_r(0,n)+Q_a(0,n)[1-Q_r(1,n)] & i=0 \quad (4-17c)\\ Q_a(0,n)Q_r(1,n) & i=-1 \quad (4-17d)\end{cases}$$

式(4-17a)表示有 $i \geq 2$ 个新到达的分组进行传输,此时必然会导致碰撞。从而不论原来的所有处于等待重传状态的节点是否进行传输,都将使 $n \to n+i$。式(4-17b)表示在 n 个等待重传节点有分组传输的情况下,空闲节点中有一个新到达的分组进行传输,此时也必然产生碰撞,并且使 $n \to n+1$。式(4-17c)包含了两种情况:第一种是仅有一个新到达分组进行传输,所有等待重传的分组没有分组进行传输的情况,此时新到达的分组将成功传输,即式(4-17c)中第一项表示新到达分组成功传输的概率;第二种情况是没有新分组到达,等待重传节点没有分组传输或有两个及两个以上分组传输的情况,即式(4-17c)中第二项表示在等待重传节点没有分组传输或有两个及两个以上分组传输的概率。不论在哪种情况下,网络中处于等待重传状态的节点数都不会变化,此时有 $n \to n$。式(4-17d)表示等待重传的节点有一个分组成功传输的概率。该系统的状态转移图如图 4-13 所示。

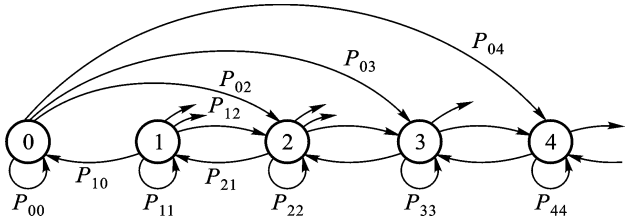


图 4-13 马尔可夫链的状态转移图

从图中可以看出,系统不会出现 0→1 的状态转移,这是因为此时系统中仅有一个分组,必然会传输成功。而且,每次状态减少的转移中只能减少 1,这是因为一次成功传输只能有一个分组。在稳态情况下,对于任一状态 n 而言,从其他状态转入的频率应当等于从该状态转移出去的频率,即有

$$\sum_{i=0}^{n-1} p_i P_{i,n} + p_n P_{n,n} + p_{n+1} P_{n+1,n} = \sum_{j=n-1}^{m} p_n P_{n,j} = p_n \sum_{j=n-1}^{m} P_{n,j} \tag{4-18}$$

由于从 n *转移到各种可能状态的概率之和为* 1 $\sum_{j=n-1}^{m} P_{n,j} = 1$,从而有

$$p_n = \sum_{i=0}^{n+1} p_i P_{i,n} \tag{4-19}$$

再利用 $\sum_{i=0}^{m} p_i = 1$ 和式(4-17)就可以求出 p_0 和 p_n。

从前面的讨论可以看到,如果重传的概率 $q_r \approx 1$,将会导致出现大量的碰撞,从而使系统中的节点长时间处于等待重传状态。为了进一步了解系统的动态行为,定义系统状态偏移量为

D_n = 当系统状态为 n 时，在一个时隙内等待重传队列的平均变化量

= (在该时隙内平均到达的新分组数) − (在该时隙内平均成功传输的分组数)

$$= (m-n)q_a - P_{succ} \tag{4-20}$$

式中

$$P_{succ} = Q_a(1,n)\cdot Q_r(0,n) + Q_a(0,n)\cdot Q_r(1,n) \tag{4-21}$$

式中的第一项是一个新到的分组传输成功的概率，第二项是重传队列中有一个分组传输成功的概率。系统状态偏移量反映了系统状态变化的趋势。如果 $D_n<0$，表明系统状态转移的整体趋势是向左的，系统将趋于稳定。如果 $D_n>0$，则表明系统状态转移的整体趋势是向右的，系统将趋于不稳定。

再定义当系统状态为 n 时，一个时隙内平均传输的分组数为 $G(n)$，则有

$$G(n) = (m-n)q_a + nq_r \tag{4-22}$$

并将式 (4−15) 和式 (4−16) 代入式 (4−21)，化简可得

$$P_{succ} = \frac{(m-n)q_a}{1-q_r} + \frac{nq_r}{1-q_r}(1-q_a)^{m-n}(1-q_r)^n$$

$$\approx G(n)e^{-q_a(m-n)}e^{-q_r n}$$

$$\approx G(n)e^{-G(n)} \tag{4-23}$$

分组到达率与系统状态 n 的关系曲线 ($(m-n)q_a \sim n$) 和分组离开率与一个时隙内平均传输的分组数的关系曲线 [$P_{succ} \sim G(n)$] 如图 4−14 所示。图中的横轴有两个坐标：一个是系统状态 n，另一个是一个时隙内平均传输的分组数 $G(n) = (m-n)q_a + nq_r$。

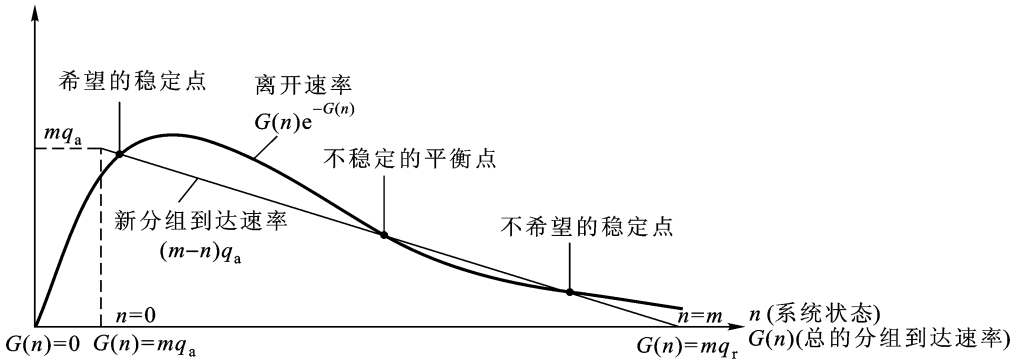


图 4−14 时隙 ALOHA 的动态性能曲线

从图中可以看出，D_n 就是分组到达速率 (到达率) 曲线与分组离开速率 (离开率) 曲线之差。两条曲线有三个交叉点即三个平衡点。在第一个交叉点与第二个交叉点之间由于系统的离开率大于分组的到达率，所以 D(n) 为负值，因而

会导致系统的状态减少。或者说，$D(n)$ 的方向为负，因而对第二交叉点的任何负的扰动都会导致系统趋于第一个交叉点。而在第二个交叉点与第三个交叉点之间的 $D(n)$ 为正值，即系统的分组到达率大于分组离开率，因而在该区域内的状态变化会导致系统的状态趋于第三个交叉点。因此可以得出以下结论，即第一和第三个交叉点是稳定的平衡点，而第二个交叉点是不稳定的平衡点。从图4-14还可以看到，对于第一个交叉点有较高的通过率，而第三个交叉点的通过率很低。因此，第一个交叉点是希望的稳定平衡点，而第三个交叉点是不希望的稳定平衡点。

如果将重传概率 q_r 增加，则重传的时延将会减小。对应于图4-14中的横坐标，若 n 保持不变，则 $G(n)=(m-n)q_a+nq_r$ 的取值将增加，$G(n)$ 对应的曲线 $G(n)e^{-G(n)}$ 的值将会下降，即曲线向左压缩，第二个交叉点向左移。这样，退出不稳定性的可能性增加，但到达不稳定点的可能性增大。因为此时很小的 n 值，都可能使系统进入不稳定区域。

如果 q_r 减小，则重传时延将会增加。若保持图4-14的横坐标 *n 不变，则 $G(n)$ 取值下降，曲线 $G(n)e^{-G(n)}$* 的值将会增加，即曲线向右扩展。在向右扩展一定程度后，系统将仅有一个稳定点。

下面讨论在假设 B 的情况下，系统的稳定性能。

在假设 B 的情况下，由图4-12可知，当到达率为常量时，不希望的稳定点消失，只有一个希望的稳定点和一个不希望的稳定点。当系统状态超过不稳定点时，系统的通过率趋于0，时延将会趋于∞。

4. 稳定的时隙 ALOHA 协议——伪贝叶斯算法

对于一个多址协议还需要讨论它是否是一个稳定的多址协议。所谓稳定的多址协议是指对于给定的到达率，多址协议可以保证每个分组的平均时延是有限的。或者说对于给定的到达率，系统是稳定的。使系统稳定的到达率的最小上界称为系统的最大稳定通过率。很显然由上述定义，普通的时隙 ALOHA 协议对于任何大于0的到达率都是不稳定的，也就是说，最大稳定的通过量为0。

从上一节的讨论可以看出，P_{succ} 近似等于 $G(n)e^{-G(n)}$，并且当 $G(n)=1$ 时，获得最大的系统通过量。如果可以动态的改变 q_r，使 $G(n)$ 总是处于1，则系统可以一直获得最大的通过量。由于 $G(n)$ 是 n 的函数（n 为系统中处于重传状态的分组数），因此只要能正确估计 n 的值，就可以使 $G(n)=1$。由于 n 是未知的值，所以只能通过反馈信息来估计。

假定 n 可以准确估计且 $G(n)=1$，则根据 Poisson 到达的近似，可得成功的概率为 $\frac{1}{e}\approx 0.368$，空闲的概率为 $\frac{1}{e}\approx 0.368$，碰撞的概率为 $1-\frac{2}{e}=\frac{e-2}{e}\approx 0.264$。因此，在调整重传概率 q_r 时，应使碰撞的概率小于空闲的概率。

伪贝叶斯算法(pseudo-Bayesian algorithm)是一种稳定的时隙 ALOHA 算法。它的核心思想是:尽可能地使 $G(n)=1$,从而使系统的通过率达到最大值。其基本思路是:假定系统有无穷多个节点(假设 B),新到达的分组立即被认为是等待重传的分组(这是与普通时隙 ALOHA 协议的差别),即所有的分组都以相同的方式处理。根据时隙开始点状态(等待重传的节点数)的估计值 $\hat{n}_k$ 确定重传概率 $q_r(\hat{n}_k)$,并根据当前时隙的传输状态(空闲、成功或碰撞)来估计下一时隙开始点的状态 $\hat{n}_{k+1}$。在理想情况下,假定在时隙开始处有 n 个等待重传的分组,则当前时隙总的传输速率为 $G(n)=nq_r$,其成功传输一个分组的概率是 $\binom{n}{1}q_r(1-q_r)^{n-1}=nq_r(1-q_r)^{n-1}$。根据 $G(n)=nq_r=1$ 的要求,应有 $q_r=\frac{1}{n}$。

伪贝叶斯算法的具体步骤如下:

(1) 估计当前时隙(第 k 个时隙)开始点的等待重传的节点数 $\hat{n}_k$,则各个节点在第 k 个时隙发送的概率 $q_r(\hat{n}_k)=\min\left(1,\frac{1}{\hat{n}_k}\right)$(即要求 $q_r\leq 1$)

(2) 根据第 k 个时隙的传输结果,估计第 $k+1$ 个时隙开始点的等待重传的节点数 $\hat{n}_{k+1}$

$$\hat{n}_{k+1}=\begin{cases}\max\{\lambda,\hat{n}_k+\lambda-1\} & \text{第 } k \text{ 个时隙空闲或成功}\\ \hat{n}_k+\lambda+(e-2)^{-1} & \text{第 } k \text{ 个时隙碰撞}\end{cases} \tag{4-24}$$

式中加入 λ 是考虑到新的到达,取 max 表示对 $\hat{n}_{k+1}$ 的估计不会小于新到达分组的贡献。若传输成功,则新的估计要从 $\hat{n}_k$ 中减去 1。若碰撞则新的估计要将 $\hat{n}_k$ 增加 $(e-2)^{-1}$。增加 $(e-2)^{-1}$ 的目的是适当地减少重传的概率。若时隙空闲,将 $\hat{n}_k$ 减去 1,这样适当地增加重发的概率,以免有太多的空闲时隙。这样可以维持系统的真实状态取值 n 和估计值 $\hat{n}_k$ 之间的平衡。也就是说,在空闲和碰撞的情况下,平均队长不应该改变。因为在 Poisson 近似下,空闲的概率为 $\frac{1}{e}$,碰撞的概率为 $\frac{e-2}{e}$,则这两种情况下平均队长的改变为 $-\frac{1}{e}+\frac{1}{e-2}\cdot\frac{e-2}{e}=0$。

该伪贝叶斯算法对于任何 $\lambda<\frac{1}{e}$ 的到达率都是稳定的。在 n 较大时,如果有 $n=\hat{n}$,则有 $q_r=\frac{1}{n}$,$G(n)=1$,其成功的概率为 $\frac{1}{e}$。根据式(4-19)的定义有:$D(n)=\lambda-\frac{1}{e}$,当 $\lambda<\frac{1}{e}$ 时,则有 $D(n)<0$,此时系统是稳定的。当系统状

态估计的初值 $\hat{n}_k$ 与实际系统的 n 相差较大时，系统也会进入稳态。因为当 $n \ll \hat{n}$，$\hat{n}_k$ 较小，系统碰撞概率很大，必然导致 $\hat{n}_k$ 迅速增加，从而有 $\hat{n}$ 趋于 n。当 $\hat{n} \gg n$ 时，$\hat{n}_k$ 较大，系统空闲的概率很大，成功传输的概率很高，必然导致 $\hat{n}_k$ 迅速减少，从而有 $\hat{n}$ 趋于 n。

要准确地分析时隙ALOHA协议的平均时延是很困难的，这里仅给出纯ALOHA协议和时隙ALOHA协议的平均传输时延和吞吐量的关系曲线。如图4-15所示。这是在忽略传播时延，且重发间隔在(1,5)个单位时间内均匀分布的条件下得到的。从两条曲线的对比可以看出，当吞吐量很小时，纯ALOHA的性能要稍好些。但当吞吐量增大时(尤其是当S接近于0.18时)，纯ALOHA的时延会急剧上升，而时隙ALOHA却可以在更高的吞吐量下工作。

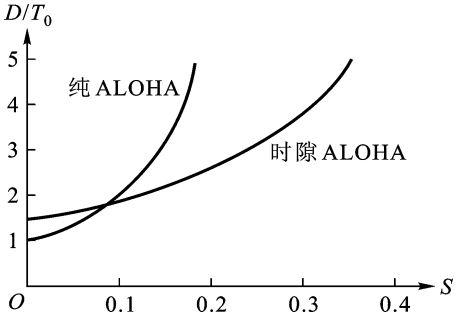


图4-15 ALOHA协议中帧的平均传输时延与吞吐量的关系曲线

4.3.2 载波侦听型多址协议

在前面讨论的ALOHA协议中，网络中的节点不考虑当前信道是忙还是闲，一旦有分组到达就独自决定将分组发送到信道。显然这种控制策略存在盲目性。即使是稍有改进的时隙ALOHA协议，其最大吞吐率也只能达到约0.368。若要进一步提高系统吞吐率，还应进一步设法减少节点间发送冲突的概率。为此，除了缩小易受破坏区间外(这也是有限度的)，还可以从减少发送的盲目性着手，在发送之前先观察信道是否有用户在传输(或进行“载波侦听”)来确定信道忙闲状态，然后再决定分组是否发送。这就是被广泛采用的载波侦听型多址接入协议CSMA(Carrier Sense Multiple Access)。CSMA是从ALOHA协议演变出的一种改进型协议，它采用了附加的硬件装置，每个节点都能够检测(侦听)到信道上有无分组在传输。如果一个节点有分组要传输，它首先检测信道是否空闲，如果信道有其他分组在传输，则该节点可以等到信道空闲后再传输，这样可以减少要发送的分组与正在传输的分组之间的碰撞，提高系统的利用率。

CSMA协议可细分为几种不同的实现形式：非坚持型(non-persistent)CSMA，1-坚持型CSMA和 p-坚持型CSMA。所谓非坚持型CSMA是指当分组到达时，若信道空闲，则立即发送分组；若信道处于忙状态，则分组的发送将被延迟，且节点不再跟踪信道的状态(即节点暂时不检测信道)，延迟结束后节点再次检测信道状态，并重复上述过程，如此循环，直到将该分组发送成功为止。所谓1-坚持型CSMA是指当分组到达时，若信道空闲，则立即发送分组；若信道

处于忙状态,则该节点一直坚持检测信道状态,直至检测到信道空闲后,立即发送该分组。所谓 p-坚持型 CSMA 是指当分组到达时,若信道空闲,则立即发送分组;若信道处于忙状态,则该节点一直检测信道的状态,在检测到信道空闲后,以概率 p 发送该分组。在下面的讨论中将重点讨论非坚持型 CSMA 的性能。

众所周知,由于电信号在介质中的传播时延,在不同的观察点上监测到同一信道的出现或消失的时刻是不相同的。因此,在 CSMA 多址协议中,影响系统性能的主要参数是(信道)载波的检测时延(τ)。它包括两部分:发送节点到检测节点的传播时延和物理层检测时延(即检测节点开始检测到检测节点给出信道是忙或闲所需的时间)。设信道速率为 C(bit/s),分组长度为 L(bit),则归一化的载波侦听(检测)时延为 $\beta=\tau\cdot\frac{C}{L}$。

1. 非时隙 CSMA 多址协议

非时隙 CSMA 协议的工作过程如下:当分组到达时,如果信道空闲,则立即发送该分组;如果信道忙,则分组被延迟一段时间后,重新检测信道。

如果信道忙或发送时与其他分组碰撞,则该分组变成等待重传的分组。每个等待重传的分组将重复地尝试重传,重传间隔相互独立且服从指数分布。其具体的控制算法描述如下:

(1) 若有分组等待发送,则转到第(2)步,否则处于空闲状态,等待分组到达。

(2) 监测信道:若信道空闲,启动发送分组,发完返回第(1)步;若信道忙,放弃监测信道,选择一个随机时延的时间长度 t 开始延时(此时节点处于退避状态)。

(3) 延时结束,转至第(1)步。

图 4-16 给出了整个过程的示意图。

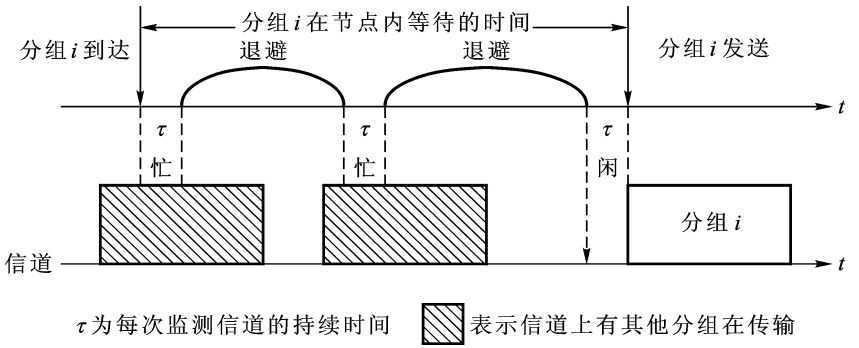


图 4-16 非坚持型非时隙 CSMA 协议的控制过程示意图

非坚持型非时隙 CSMA 多址协议的主要特点是在发送数据前先监测信道,

一旦监测到信道忙时，能主动的退避一段时间（暂时放弃监测信道），其系统通过率为

$$S=\frac{Ge^{-\beta G}}{G(1+2\beta)+e^{-\beta G}} \tag{4-25}$$

2. 时隙 CSMA 多址协议

时隙 CSMA 协议把时间轴分成宽度为 β 的时隙（注意：时隙 ALOHA 中时隙的宽度为一个分组的长度，这里的时隙宽度为载波检测时间）。如果分组到达一个空闲的时隙，它将在下一个空闲时隙开始传输［如图 4－17（a）］。如果某节点的分组到达时，信道上有分组正在传输，则该节点变为等待重传的节点，它将在当前分组传输结束后的后续空闲时隙中以概率 q 进行传输，如图 4－17（b）所示。

可以用马尔可夫链来分析时隙 CSMA 协议的性能。设分组长度为 1 个单位长度，其总的到达过程是速率为 λ 的 Poisson 到达过程，网络中有无穷多个节点（假设 B）。信道状态 0、1、e 的反馈时延最大为 β。又设系统的状态为每一个空闲时隙结束时刻等待重传的分组数 n，则相继两个状态转移的时间间隔为 β 或 $\beta+1$，如图 4－17（c）所示。

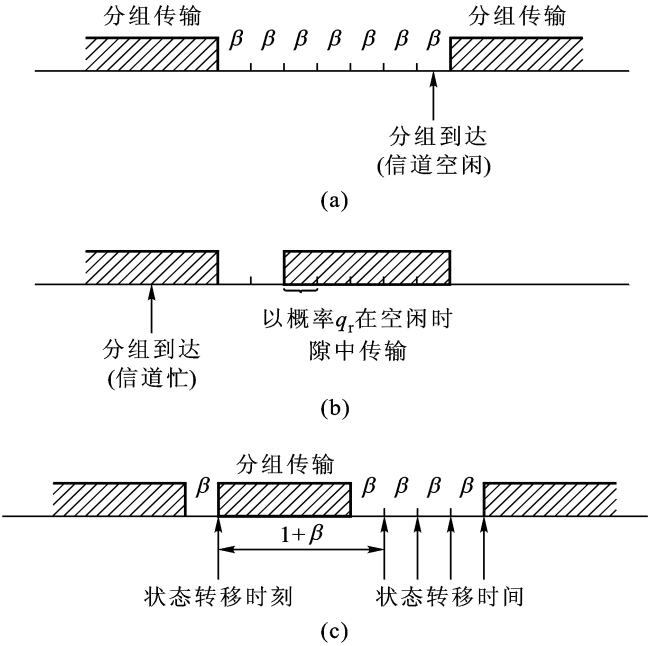


图 4－17 时隙非坚持 CSMA 协议

（a）分组到达空闲时隙 （b）分组到达时信道忙 （c）状态转移时刻

类似于式（4－20），定义在一个状态转移间隔内 n 的平均变化数为

$$D_n = E\{\text{状态转移间隔内到达的分组数}\} - P_{succ}$$
$$= \lambda \cdot E\{\text{状态转移间隔}\} - P_{succ} \quad (4-26)$$

这里

$$E\{\text{状态转移间隔}\} = \beta \cdot P(\text{时隙空闲}) + (1+\beta)(1-P(\text{时隙空闲}))$$
$$= \beta + 1 - P(\text{时隙空闲})$$
$$= \beta + 1 - e^{-\lambda\beta}(1-q_r)^n \quad (4-27)$$

时隙空闲对应于 *β间隔内无分组到达，以及 n个等待重传的节点没有分组发送*。分组成功传输的条件是：在 β 内有一个分组到达且 n 个等待重传的节点没有分组发送，或在 β 内没有新分组到达但 n 个等待重传的节点有一个分组传输。因此有

$$P_{succ} = \lambda\beta e^{-\lambda\beta}(1-q_r)^n + e^{-\lambda\beta} n q_r (1-q_r)^{n-1}$$
$$= \left(\lambda\beta + \frac{q_r}{1-q_r} n\right) e^{-\lambda\beta}(1-q_r)^n \quad (4-28)$$

将式 (4-27) 和式 (4-28) 代入式 (4-26) 得

$$D_n = \{\lambda[\beta + 1 - e^{-\lambda\beta}(1-q_r)^n]\} - \left(\lambda\beta + \frac{q_r}{1-q_r} n\right) e^{-\lambda\beta}(1-q_r)^n \quad (4-29)$$

当 q_r 较小时，有 $(1-q_r)^{n-1} \approx (1-q_r)^n \approx e^{-q_r n}$，进而有

$$D_n \approx \lambda(\beta + 1 - e^{-g(n)}) - g(n) e^{-g(n)} \quad (4-30)$$

式中，$g(n) = \lambda\beta + q_r n$，*它反映的是在一个状态转移间隔内到达分组数和重传分组数之和*，即在一个状态转移间隔内试图进行传输的总分组数。

使 D_n 为负的条件为

$$\lambda < \frac{g(n) e^{-g(n)}}{\beta + 1 - e^{-g(n)}} \quad (4-31)$$

由于在一个状态转移间隔内到达的总分组数（新到达的和重传的分组数之和）是服从参数为 $g(n)$ 的 Poisson 分布，所以式 (4-31) 右边的分子为每个状态转移区间内平均成功传输的分组数，分母为平均状态转移区间的长度，两者相除表示单位时间内的平均离开率（通过率）。它与 $g(n)$ 的关系如图 4-18 所示。从图中可以看出，最大的通过率为 $\frac{1}{1+2\beta}$，它对应于 $g(n) = 2\beta$。CSMA 协议与 ALOHA 协议一样，也存在着稳定性的问题。图 4-19 给出了几种典型的随机多址接入协议的性能曲线。从图中可以看出，非坚持型 CSMA 协议可以大大减少碰撞的机会，使系统的最大吞吐量达到信道容量的 80% 以上，时隙非坚持型 CSMA 协议的性能则更好。1-坚持型 CSMA 由于毫无退避措施，在业务量很小时，数据的发送机会较多，响应也较快。但若节点数增大或总的业务量增加

时，碰撞的机会急剧增加，系统的吞吐量特性急剧变坏，其最大吞吐量只能达到信道容量的53%左右。但总的说来，CSMA协议的性能优于ALOHA协议的性能。

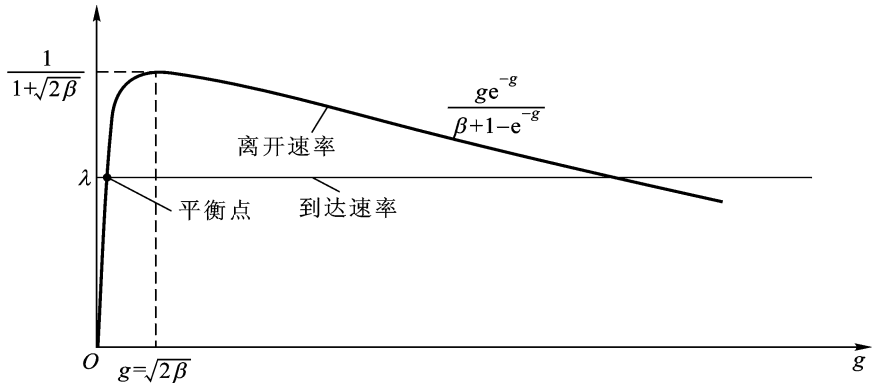


图4-18 时隙CSMA协议的平均离开率（通过率）

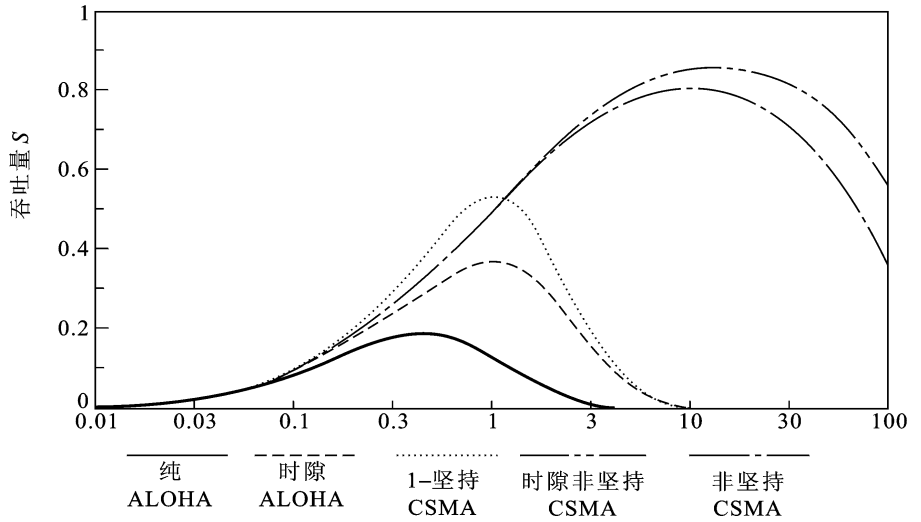


图4-19 典型的随机接入多址协议性能曲线

3. 稳定的时隙CSMA多址协议

假定所有新进入系统的分组立即变成等待重传的分组。设每个状态转移时刻的等待重传分组数为n。n的估计值为$\hat{n}$，在每个空闲时隙结束时，每个等待重传的分组独立地以概率q_r发送，q_r是$\hat{n}$的函数。稳定的时隙CSMA协议的基本出发点是根据n如何确定q_r，使得$g(n)=2\beta$，从而使通过量达到最大。在给定n的条件下，在当前时隙开始发送的平均分组数为$g(n)=nq_r$，根据

$g(n)=2\beta$得 $q_r=\frac{2\beta}{n}$。

在给定 n 的一个估计值 $\hat{n}$ 的情况下，q_r 应这样选择

$$q_r(\hat{n})=\min\left(\frac{2\beta}{\hat{n}},\ 2\beta\right) \tag{4-32}$$

式中取极小值是为了防止 $\hat{n}$ 较小时，导致 $q_r(\hat{n})$ 太大。更新 $\hat{n}$ 的规则如下：

$$\hat{n}_{k+1}=\begin{cases}\hat{n}_k[1-q_r(\hat{n}_k)]+\lambda\beta & ;\ \text{时隙空闲}\\ \hat{n}_k[1-q_r(\hat{n}_k)]+\lambda(1+\beta) & ;\ \text{成功传输}\\ (\hat{n}_k+2)+\lambda(1+\beta) & ;\ \text{碰撞}\end{cases} \tag{4-33}$$

上式中，等号后的第一项反映了等待重传队列中分组数变化情况的估计，第二项反映了新到达的分组数。可以证明，只有当 $\lambda<\frac{1}{1+2\beta}$，该算法才是稳定的。

4. 有碰撞检测功能的载波侦听型多址协议 (CSMA/CD)

前面讨论的 CSMA 协议由于在发送之前进行载波监听，所以减少了冲突的机会。但由于传播时延的存在，冲突还是不可避免的。只要发生冲突，信道就被浪费一段时间。CSMA/CD 比 CSMA 又增加了一个功能，这就是边发送边监听。只要监听到信道上发生了冲突，则冲突的节点就必须停止发送。这样，信道就很快空闲下来，因而提高了信道的利用率。这种边发送边监听的功能称为冲突检测。

CSMA/CD 的工作过程如下：当一个节点有分组到达时，它首先侦听信道，看信道是否空闲。如果信道空闲，则立即发送分组；如果信道忙，则连续侦听信道，直至信道空闲后立即发送分组。该节点在发送分组的同时，监测信道 δ 秒，以便确定本节点的分组是否与其他节点发生碰撞。如果没有发生碰撞，则该节点会无冲突地占用该总线，直至传输结束。如果发生碰撞，则该节点停止发送，随机时延一段时间后重复上述过程。(在实际应用时，发送节点在检测到碰撞以后，还要产生一个阻塞信号来阻塞信道，以防止其他节点没有检测到碰撞而继续传输。)

总的来说，CSMA/CD 接入协议比 CSMA 多址接入协议的控制规则增加了如下三点：

(1) “边说边听”——任一发送节点在发送数据帧期间要保持侦听信道的碰撞情况。一旦检测到碰撞发生，应立即中止发送，而不管目前正在发送的帧是否发完。

(2) “强化干扰”——发送节点在检测到碰撞并停止发送后，立即改为发送一小段“强化干扰信号”，以增强碰撞检测效果。

(3) “碰撞检测窗口”——任一发送节点若能完整的发完一个数据帧，则停

顿一段时间（两倍的最大传播时延）并监听信道情况。若在此期间未发生碰撞，则可认为该数据帧已经发送成功。此时间区间即称“碰撞检测窗口”。

上述第（1）点保证尽快确知碰撞发生和尽早关闭碰撞发生后的无用发送，这有利于提高信道利用率；第（2）点可以提高网络中所有节点对于碰撞检测的可信度，保证了分布式控制的一致性；第（3）点有利于提高一个数据帧发送成功的可信度。如果接收节点在此窗口内发送应答帧（ACK 或 NAK）的话，则可保证应答传输成功。

下面分析 CSMA/CD 协议的性能。为了简化分析，首先假定一个局域网（LAN）工作在时隙状态下，以每个分组传输的结束时刻作为参考点，将空闲信道分为若干个微时隙，用分组长度进行归一化的微时隙的宽度为 β。所有节点都同步在微时隙的开始点进行传输。如果在一个微时隙开始点有分组发送，则经过一个微时隙后，所有节点都检测到在该微时隙上是否发生碰撞。如果发生了碰撞，则立即停止发送。

这里仍然用马尔可夫链的方法分析。分析的方法与时隙 CSMA 协议相同。设网络中有无穷多个节点，每一个空闲时隙结束时的等待重传的分组数为 n，每个等待重传的节点在每一个空闲时隙后发送的概率为 q_r。在一个空闲时隙发送分组的节点数为 $g(n)=\lambda\beta+q_r n$。*在一个空闲时隙后可能有三种情况：一是*仍为空闲时隙，二是一个成功传输（归一化的分组长度为 1），三是一个碰撞传输。它们所对应的到达下一个空闲时隙结束时刻的区间长度分别为 β、$1+\beta$ *和* 2β。*因此，两个状态转移时刻的平均间隔为*

$$E\{\text{状态转移时刻的间隔}\}=\beta+1\cdot g(n)e^{-g(n)}+\beta\cdot\left[1-(1+g(n)e^{-g(n)}\right] \tag{4-34}$$

式中第一项表示在任何情况下基本的间隔为 β，第二项是成功传输对平均间隔的额外贡献，第三项是碰撞对平均间隔的额外贡献。

定义在一个状态转移区间内，n 的变化量为

$$D_n=\lambda\cdot E\{\text{状态转移时刻的间隔}\}-P_{succ} \tag{4-35}$$

式中

$$P_{succ}=g(n)e^{-g(n)} \tag{4-36}$$

要使 $D_n<0$，则有

$$\lambda<\frac{g(n)e^{-g(n)}}{\beta+g(n)e^{-g(n)}+\beta[1-(1+g(n)e^{-g(n)})]} \tag{4-37}$$

从上式可以看出，不等式的右边为分组离开系统的概率，其最大值为 $\frac{1}{1+3.31\beta}$，它对应的 $g(n)=0.77$。因此，如果 CSMA/CD 是稳定的（如采用伪贝叶斯算法），则系统稳定的最大分组到达率应小于 $\frac{1}{1+3.31\beta}$。

5. 有碰撞避免功能的载波侦听型多址协议(CSMA/CA)

CSMA/CA是有冲突避免(collision avoidance)的载波侦听型多址接入协议。它是对CSMA的另一种改进方法。通常在无线系统中,一台无线设备不能在相同的频率(信道)上同时进行接收和发送,因而不能采用碰撞检测(CD)技术。因此,只能通过冲突避免的方法来减少冲突的可能性。在IEEE802.11无线局域网(WLAN)的标准中,就采用了CSMA/CA协议。它不仅支持全连通的网络拓扑,同时支持部分连通的网络拓扑。

IEEE802.11中CSMA/CA的基本工作过程是:一个节点在发送数据帧之前先对信道进行预约。假定A要向B发送数据帧,发送节点A先发送一个请求发送帧RTS(Request To Send)来预约信道,所有收到RTS帧的节点将暂缓发送。而真正的接收节点B在收到RTS后,发送一个允许发送的应答帧CTS(Clear To Send)。在RTS和CTS帧中均包括要发送分组的长度(在给定信道传输速率及RTS和CTS长度的情况下,各节点就可以计算出相应的退避时间,该时间通常称为NAV(Network Allocation Vector))。CTS帧有两个作用:一是表明接收节点B可以接收发送节点A的帧,二是禁止B的邻节点发送,从而避免了B的邻节点的发送对A到B的数据传输的影响。RTS和CTS帧很短,如它们分别可为20和14个字节。而数据帧最长可以达到2346字节。相比之下,RTS和CTS引入的开销不大。RTS/CTS的传输过程如图4-20所示。

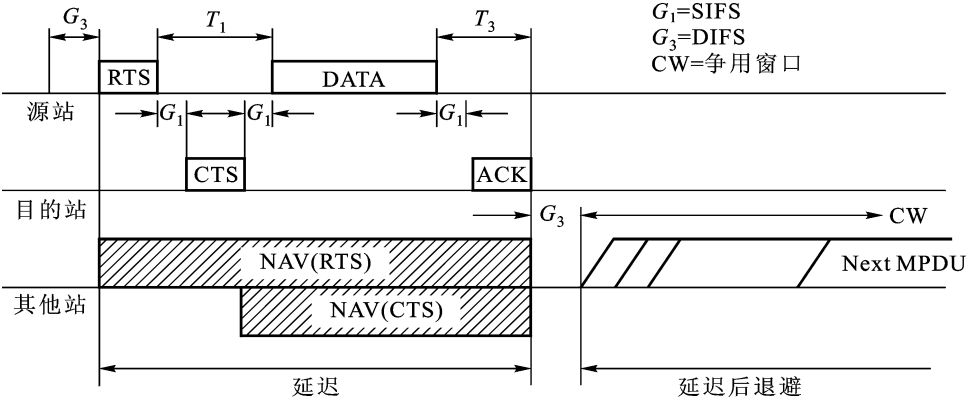


图4-20 RTS/CTS的传输过程

为了尽量避免冲突,IEEE802.11标准给出了三种不同的帧间间隔IFS(Inter Frame Space),它们的长短各不相同。参照图4-20,给出只使用一种IFS(假定对图中的SIFS、DIFS不加以区分)时的CSMA/CA接入算法。

(1) 发送帧的节点先侦听信道。若发现信道空闲,则继续侦听一段时间IFS,看信道是否仍为空闲。如是,则立即发送数据。

(2) 若发现信道忙(无论是一开始就发现,还是在后来的IFS时间内发现),

则继续监听信道,直到信道变为空闲。

(3) 一旦信道变为空闲,此节点延时另一个时间IFS。若信道在时间IFS内仍为空闲,则按二指数退避算法(所谓二进制指数退避算法,即对一个分组的重发退避时延的取值范围与该分组的重发次数构成二进制指数关系。也就是说,随着分组遭碰撞而重发次数的增加,其退避时延的取值范围按2的指数增大)延时一段时间。只有在退避期间信道一直保持空闲,该节点才能发送数据。这样做可使在网络负荷很重的情况下,发生冲突的机会大为减小。

IEEE802.11协议中定义了三种不同的帧间间隔:

SIFS,即短帧帧间间隔(Short IFS),典型的数值只有10 μs。

PIFS,即点协调功能中的帧间间隔,比SIFS长,在点协调(PCF)方式轮询时使用。

DIFS,即分布协调功能中的帧间间隔,是最长的帧间间隔,典型数值为50 μs。在分布式协调(DCF)方式中使用。

4.4 冲突分解算法

对于有竞争的多址接入协议如何解决冲突从而使所有碰撞用户都可以成功传输是一个非常重要的问题。从前面的讨论可以看出,通过调整对等待重传队列长度的估值,改变重传概率,可以进一步减缓碰撞。而另一种更有效的解决冲突的方式就是冲突分解(Collision Resolution)。

冲突分解的基本思想是:如果系统发生碰撞,则让新到达的分组在系统外等待,在参与碰撞的分组均成功传输结束后,再让新分组传输。

下面针对ALOHA协议进行讨论,以两个分组碰撞的情况来简要说明冲突分解的过程和好处。

例4.2 设两个分组在第 i 个时隙发生碰撞,若每个分组独立的以 $\frac{1}{2}$ 的概率在第 $i+1$ 和 $i+2$ 时隙内重传。求在这次冲突分解过程的通过率。

解 在第 $i+1$ 个时隙内有一个分组成功传输的概率为 $\frac{1}{2}$。如果成功,另一个分组在第 $i+2$ 个时隙内成功传输,此时需2个时隙解决碰撞。如果第 $i+1$ 个时隙空闲或再次碰撞,则每个分组再独立地以概率 $\frac{1}{2}$ 在第 $i+2$ 和 $i+3$ 时隙内重传。这样在第 $i+2$ 个时隙内有一个分组成功传输的概率为 $\frac{1}{4}$ 如果成功,另一个分组在第 $i+3$ 个时隙成功传输,此时共需3个时隙解决碰撞。依此类推,需要 k 个时隙完成冲突分解的概率为 $2^{-(k-1)}$。设一个分组成功传输所需的平均时隙数为 $E[t]$,由于每个分组需传输 i 次才能成功(其中 $i-1$ 次重传,1

次正确传输),而分组正确传输的概率是$\frac{1}{2}$,所以有

$$E[t]=\sum_{i=1}^{\infty} i\times\left(\frac{1}{2}\right)^{i-1}\times\frac{1}{2}=\sum_{i=1}^{\infty} i\times\left(\frac{1}{2}\right)^{i}=2$$

一旦一个分组成功传输,则另一个分组在下一时隙必然成功传输,所以平均需要3个时隙才能成功发送2个分组。因而在冲突分解的过程中,通过率为$\frac{2}{3}$。

例4.2说明通过冲突分解可以有效的提高系统的通过率。有很多方法可以决定碰撞节点如何进行重传。下面给出两种具体的冲突分解算法:树形分裂算法(Tree Splitting Algorithm)和先到先服务的分裂算法(FCFS Splitting Algorithm)。

4.4.1 树形分裂算法

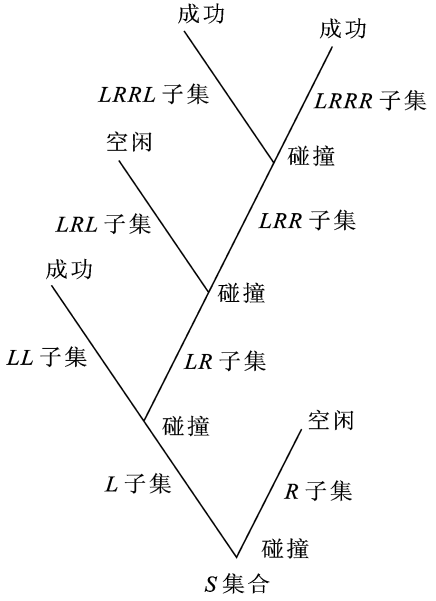
假设在第k个时隙发生碰撞,碰撞节点的集合为S。*所有未介入碰撞的节*点进入等待状态。S被随机地分成两个子集,用左集(L)和右集(R)表示。左集(L)先在第k+1时隙中传输。如果第k+1时隙中传输成功或空闲,则R在第k+2个时隙中传输。如果在第k+1时隙中发生碰撞,则将L再分为左集(LL)和右集(LR),LL在第k+2个时隙中传输。如果第k+2时隙中传输成功或空闲,则LR在第k+3个时隙中传输。以此类推,直至集合S中所有的分组传输成功。从碰撞的时隙(第k个时隙)开始,直至S集合中所有分组成功传输结束的时隙称为一个冲突分解期(CRP)。

例4.3 *一个有三个节点在第k个时隙发生碰撞后的分解过程如图*4-21所示,图中集合的分割是采用随机的方式,即在每次集合分割时,集合中的节点通过扔硬币的方法决定自己属于左集还是右集。

该图中用了8个时隙完成了冲突分解。

该算法中,在给定每个时隙结束时立即有(0,1,e)反馈信息的情况下,各个节点能构造一个相同的树,并确定自己所处的子集和确定何时发送自己的分组。具体的方法如下:树形算法中的发送顺序可对应于一个数据压入堆栈的顺序。当一个碰撞发生后,碰撞节点的集合被分为子集,形成的每一个子集作为一个元素压入堆栈。在发送时,堆栈最顶端的子集从堆栈中移出并进行发送。每个节点采用一个计数器来跟踪它的分组所在的当前子集处于堆栈中的位置。如果该子集处于堆栈的顶端,则立即发送。当该节点的分组传输发生碰撞(冲突分解开始),计数器的初值置0或1(取决于该分组被放在哪个子集中,显然如果该分组被放入左子集,则初值被置为0;而如果该分组被放入右子集,则初值置为1)。在冲突分解过程中,当计数器的值为0时,则发送该分组。如果计数器为非0,则在冲突分解过程中,每次时隙发生碰撞,计数器值加1,每次成功传输或时隙

空闲，计数器值减1。



时隙	发送集合	等待集合	反馈
1	*S*	—	*e*
2	*L*	*R*	*e*
3	*LL*	*LR,R*	1
4	*LR*	*R*	*e*
5	*LRL*	*LRR,R*	0
6	*LRR*	*R*	*e*
7	*LRRL*	*LRRR,R*	1
8	*LRRR*	*R*	1
9	*R*	—	0

图4-21 树形冲突分解算法举例

在冲突分解期（CRP）中，处理的分组是介入碰撞的分组。而在CRP中，还会不停地有新分组到达。对于CRP中新到达的分组有两种处理方法。方法一是在当前CRP结束后立即开始一个新的CRP，该新CRP所处理的分组就是当前CRP中到达的新分组。这种方法的问题是，如果当前CRP到达了很多分组，则在新的CRP中，可能要碰撞很长时间才能通过分解得到一个很小的子集。方法二是在当前CRP结束时刻，立即将到达的分组分为 j 个子集（j 的选择应使每个子集中的分组数稍大于1），然后对每一个子集进行冲突分解。该方法的最大通过率可以达到每个时隙0.43个分组。

通过仔细观察树形算法，可以发现，如果在一次碰撞（如第 k 个时隙）以后，下一个时隙（第 $k+1$ 时隙）是空闲的，则第 $k+2$ 个时隙必然会再次发生碰撞。这表明将碰撞节点集合中的所有节点都分配到了右集（R），自然会再次发生碰撞。改进的方法是：当碰撞后出现空闲时隙，则不传送第二个子集（R）中的分组，而是立即将 R 再次分解，然后再传输分解后的第一个子集（RL），如果再次空闲，则再次进行分解，然后传送 RLL 集合中的分组，以此类推。通过这样的改进可以使每个时隙的最大通过率达到0.46个分组。

4.4.2 FCFS分裂算法

先到先服务（FCFS）分裂算法的基本思想是根据分组到达的时间进行冲突

分解，并力图保证先到达的分组最先传输成功。

设 $T(k)$ 以前到达的分组都已发送完毕，现在需确定从 $T(k)$ 开始，长度为 $\alpha(k)$ 区间内到达的分组在第 k 个时隙中传输。该区间被称为指配区间(allocation interval)。从 $T(k)+\alpha(k)$ 至当前传输时刻称为等待区间。该算法的主要功能是根据冲突分解的情况，动态地调整指配区间的长度和起始时刻。FCFS 分裂算法如图 4-22 所示。

如图 4-22(a)所示，在 $[T(k), T(k)+\alpha(k)]$ 内到达的分组在第 k 个时隙内传输。如果第 k 个时隙发生了碰撞，则将指配区间分为两个相等部分，左集(L)和右集(R)。其中，$L=[T(k+1), T(k+1)+\alpha(k+1)]$，$T(k+1)=T(k)$，$\alpha(k+1)=\frac{1}{2}\alpha(k)$，且 L 首先在第 $k+1$ 个时隙内传输。如果第 k+1 个时隙空闲，则必然在第 k+2 个时隙内碰撞。这里采用前面树形算法的改进方案。由于 R 区间必定包括 2 个以上的分组，则在第(k+1)时隙结束时刻，立即进行分解，得 RL 和 RR 两个相等的区间。其中 $RL=[T(k+2), T(k+2)+\alpha(k+2)]$，$T(k+2)=T(k+1)+\alpha(k+1)$，$\alpha(k+2)=\frac{\alpha(k+1)}{2}$，这样如果在第 k+2 时隙内传输成功，相应的第 k+3 时隙内也传输成功。此时一个 CRP 结束。

如果第 k+1 时隙发生碰撞，这说明 L 区间至少有 2 个分组。(由于 L 区间的碰撞没有给出任何关于 R 区间的信息，因此，将 R 区间划入到等待区间内，这次碰撞分解不再考虑 R 区间)。将 L 分为 LL 和 LR。其中 $LL=[T(k+2), T(k+2)+\alpha(k+2)]$，$T(k+2)=T(k+1)$，$\alpha(k+2)=\frac{1}{2}\alpha(k+1)$。在该例中[图 4-22(b)]中，LL 和 LR 在第(k+2)和第(k+3)时隙都会成功，一个 CRP 结束。

4.5 预约多址接入协议

在前面介绍的几种随机多址接入技术中，可以看到它们共同的关键技术是如何最大限度地减少发送冲突，从而尽量提高信道利用率和系统吞吐率。本节要讨论的预约多址协议的要点就是最大限度地减少或消除随机因素，避免发送竞争所带来的对信道资源的无秩序竞争，使系统能按各节点的业务需求合理地分配信道资源。所以，预约方式有时又被称为按需分配方式。

预约方式要求在网络节点之间“隐式”或“显式”地交换预约控制信息。依据这些信息，各网络节点可以执行同一个控制算法，以达到分布式控制操作的协调。预约信息的传输需要占用信道资源。因此，预约信息的多少反映了多址协议开销的多少。依据这种开销形式的不同，预约方式可分为隐式预约方式和显式预约方式。

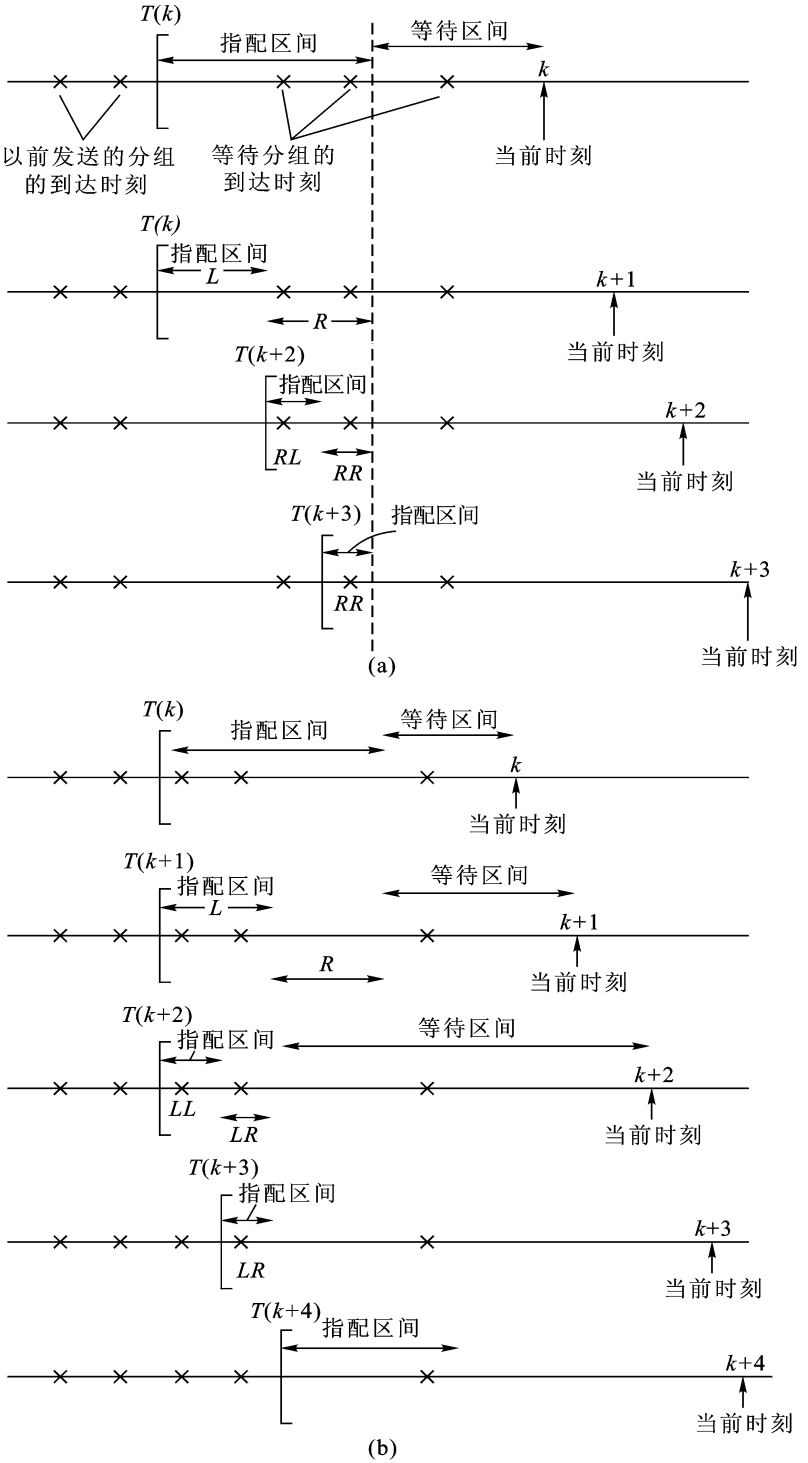


图 4-22 FCFS分裂算法示意图

(a) 第 k+1 时隙空闲 (b) 第 k+1 时隙碰撞

在随机多址协议中，当数据分组发生碰撞时，整个分组都被破坏。如果分组较长，则信道的利用率较低。当数据分组较长时，可以在数据分组传输之前，以一定的准则发送一个很短的预约分组，为数据分组预约一定的系统资源。如果预约分组成功传输，则该数据分组在预约到的系统资源（频率、时隙等）中无冲突的传输。由于预约分组所浪费的信道容量很少，因而提高了系统效率。

设数据分组的长度为1，预约分组的长度为 v（通常 $v \ll 1$）。设在预约期内，预约分组的最大通过量为 S_r，则在单位时间内数据分组的最大通过量为 $S=\frac{1}{1+\frac{v}{S_r}}$。

4.5.1 时隙预约多址协议

一种基于时隙的预约多址协议的帧格式如图 4-23(a) 所示。它采用与 TDMA 类似的帧结构。在一帧当中有一个预约的区间（其长度为 $A=mv$），该预约区间由 m 个小的预约时隙组成。在该区间中，为每个节点固定分配一个预约时隙。一帧中的另一部分为数据区间，它由若干个数据分组的传输时隙（时隙的长度等于每个分组的长度）组成。这种多址协议常用于卫星通信系统中。在卫星通信系统中，传播时延很大，如果采用随机竞争的方式，将会导致冲突分解过程很慢。设卫星链路的来回传播时延为 2β，则最小帧长应大于 2β，这样在当前帧中传输的预约分组将用于预约下一帧中的数据分组传输的时隙。或者说，在当前帧中进行预约的分组，在下一帧中才能进行传输，如图 4-23(b) 所示。

假定每个预约的时隙仅预约一个分组传输的时隙，这时系统可达到的最大通过量为 $\frac{1}{1+v}$。如果假定每个预约的时隙可以预约多个分组传输的时隙，这时帧长较长，则预约区间所占的比例很小，因而系统的通过率将趋于1。

假定分组的到达是 Poisson 到达，分组的平均长度 $\overline{X}$ 为 1 $\frac{1}{\mu}=1$。若每个预约的分组可以在当前帧中预约多个分组的传输，则该系统等同于第3章中介绍的图3-14中采用预约方式的单用户闸门型系统。由于传播时延的影响，使得在卫星系统中当前帧中预约的分组只能在下一帧中进行传输，则在卫星系统中第 i 个分组的平均等待时延为

$$E\{W\} = E\{R\} + \frac{E\{N\}}{\mu} + 2A \tag{4-38}$$

式 (4-38) 中，R_i 为第 i 个用户到达时的剩余服务时间，N_i 为第 i 个用户到达时处于等待状态（排队）的分组数，$A=mv$ 为预约传输的平均区间。在式 (4-38) 中 $2A$ 表示在卫星通信系统中，分组必须经历两个预约期后才能进行传输。利

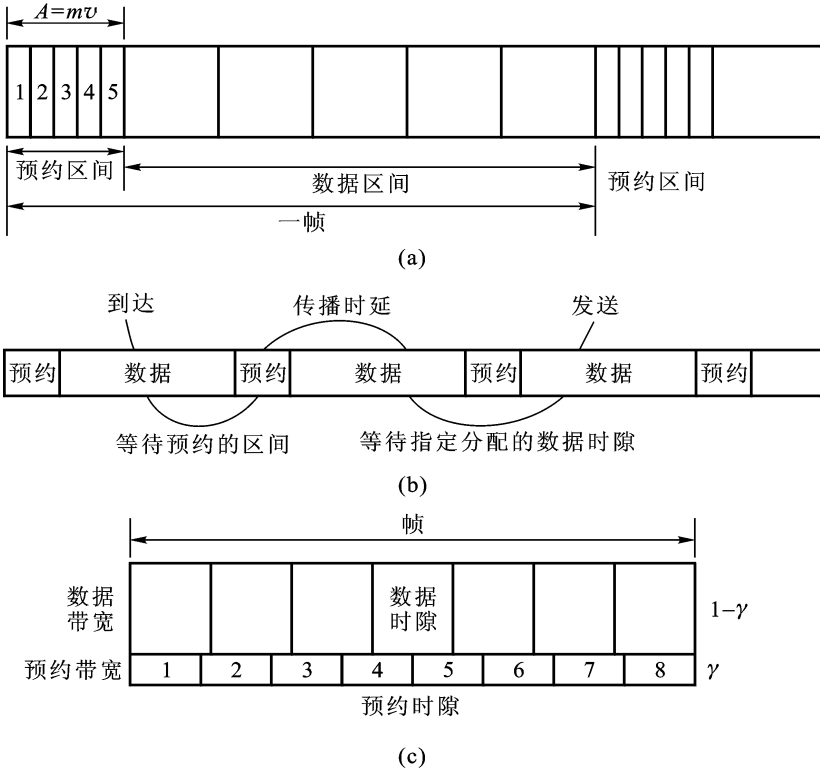


图 4−23 时隙预约多址协议

用第 3 章式 (3−94) 的结果,可以得到系统的平均等待时延为

$$W=\frac{\lambda\overline{X^2}}{2(1-\lambda)}+\frac{A}{2}+\frac{2A}{1-\lambda} \tag{4-39}$$

在式 (4−39) 的求解过程中忽略了卫星通信系统的来回传播时延 2β。*由于通常卫星信道的来回传播时延 2β 的值远远大于系统预约的平均区间 A 的取值*。所以,在式 (4−39) 的 W 值只有当 λ 非常接近于 1 时,W 的值才可能大于 2β $W=\frac{\lambda\overline{X^2}+(1-\lambda)A+4A}{2(1-\lambda)}$,只有当 λ 趋于 1 时,W 才可能得到一个比较大的值,此时认为该值可以和 2β *做比较,且大于* 2β 。然而在实际系统中,λ 值是不应该接近于 1 的。所以由式 (4−39) 求得的卫星通信系统的平均时延并不是一个很好的近似值。下面寻找该条件下的近似解。

上面的分析中采用了可变帧长的方案,该方案会带来两个方面的问题:一是如果某些节点在接收预约分配信息时,发生错误,则这些节点将无法跟踪下一个预约期,即出现不同用户之间的不同步,系统将无法工作;二是系统中会出现不公平的现象,非常繁忙的节点可能在每一帧中预约了很多时隙,从而使得帧长很长,这样会使许多节点无法接入系统。

为了解决上述问题，可采用固定帧长的方案，每一个节点仍在预约时隙中进行预约，每个节点有一个预约的时隙。如果在当前帧中分组传输不完，可推迟到下一帧中进行传输。这样从概念上讲，在系统中就形成了一个已获得预约的分组队列。该队列可以采用任何服务的规则进行服务。如先到先服务、轮询或优先等方式。只要已获得预约的分组队列不空时，在每一个数据时隙中就有分组传输，并且服务规则与分组长度无关，则平均时延与服务规则无关。

直接对该系统的性能进行分析仍是比较复杂的，对原模型稍作修改，可以得到一个很好的简单近似分析结果。

假定帧长为 2β，*则预约期占的比例为* $\gamma=\frac{mv}{2\beta}$。现在假定把系统的带宽分为两部分：一部分用于预约分组的传输，其所占带宽的比例为 γ；另一部分用于数据分组的传输，其所占带宽的比例为 $1-\gamma$ *如图* 4-23 (c) 所示。从图中可以看出，每个预约分组的传输时间为 $\frac{2\beta}{m}$，接收到预约分配信息（应答）所需的时延为 2β。*而一个新到达的分组平均要等待* $\frac{2\beta}{2}$ 才能开始进行预约分组的传输，因此一个分组从它到达系统至获得预约分配信息（或加入已获得预约的公共分组队列）的平均时延（称为预约时延）为 $\frac{2\beta}{2}+\frac{2\beta}{m}+2\beta=3\beta+\frac{2\beta}{m}$。

假定已获得预约的公共分组队列是 Poisson 到达，其平均分组的传输（服务）时间 $\frac{\overline{X}}{1-\gamma}=\frac{1}{\mu}$，则 $\rho=\frac{\lambda}{\mu}=\frac{\lambda}{1-\gamma}$。该队列可用 M/G/1 队列来描述。它对应的等待时延为 $\frac{\lambda\overline{X^2}}{1-\rho}$。因此，系统总的等待时延为该等待时延与预约时延之和。即

$$W=3\beta+\frac{2\beta}{m}+\frac{\lambda\overline{X^2}}{2(1-\rho)} \tag{4-40}$$

在这种多址协议中，采用了专门的预约区间，为每一个节点分配了一个预约时隙，这样就给节点数的增加和减少带来了困难。随着 m 的增加，预约将耗费较多的资源。可以采用下列改进方案：

(1) 在预约的小时隙中采用竞争的方式，用于预约的小时隙数比节点数 m 要少得多。这样，当 m 较大且 λ 较小时，可以降低时延。

(2) 不设专门的预约小时隙，当发送节点要发送一条消息（每条消息通常包括多个分组）时，发送节点在某一个空闲时隙以该消息的第一个分组作为预约分组进行预约，如果预约成功，则该节点就预约了后续各帧中相应的时隙，直至消息中的所有分组传输完毕。

(3) 在一帧中，为每一个节点预先固定地分配一个时隙。当该节点无分组传输时，其他节点可以通过竞争的方式使用该时隙。但当该节点要使用该时隙

时,它在“自己”的时隙上发送。如果发生碰撞,将不再允许其他节点使用该时隙。这种方法比较公平,但时延较大。

4.6 分组无线电网络

在分组无线电网络(PRNET,Packet Radio Network)中,每个节点不是与所有节点都直接相连的,或者说网络是部分连通的。这是分组无线电网与前面讨论的卫星通信、蜂窝移动通信、有线局域网的一个重要差别。

如果用图来描述一个网络,则一个分组无线网的网络拓扑图如图4-24所示。该拓扑可以用一个图 $G=(N,L)$ 来表示。其中,N是节点的集合(每一个节点用一个编号来表示),L是链路的集合。对于每一条链路(链路的起点为i,终点为j,用(i,j)来表示该链路),如果 $(i,j)\in L$,*则意味着 i 可以到达 j,用 $i\rightarrow j$ 表示(即 j* 可以直接收到i的分组)。在有些情况下有 $i\rightarrow j$,但没有 $j\rightarrow i$(*j 不能到达 i*),*因而* $(j,i)\notin L$,即 i 与j之间是单向链路(或非对称的)。一个具有对称链路的分组无线网的拓扑结构如图4-24所示。该图由5个节点组成,其节点的集合为 $N=\{1,2,3,4,5\}$,其链路的集合为 $L=\{(1,2),(1,3),(2,4),(3,4),(3,5)\}$。

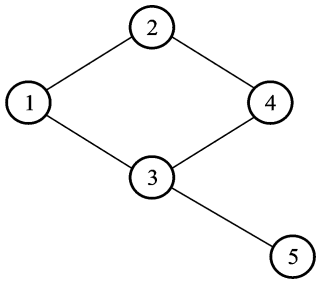


图4-24 分组无线电网络的拓扑结构

在PRNET中,*i 发送的分组能被 j 正确接收的充要条件是:*

(1) 存在 $i\rightarrow j$*的链路(即* $(i,j)\in L$*)*;

(2) 当i发送时,没有其他节点 k($(k,j)\in L$*)在发送*;

(3) i发送时,j本身未发送,即节点j处于接收状态。

由于PRNET的部分连通特性,在PRNET中就可以同时有多条链路在传输而相互不会影响。例如,在图4-24中,1→2和3→4可以同时传输。但某些链路不能同时发送,例如,2→4(同时有2→1)和3→1(同时有3→4)不能同时传输,否则会在节点1和4处产生碰撞。为了避免不同链路传输之间的碰撞,定义一个无冲突的链路集合,在该集合中,所有的链路都可以同时传输,且各接收节点没有碰撞。例如,[(1,2),(3,4)]和[(2,1),(5,3)]两个都是无冲突集。

如果把图 *G* 中的所有链路按任一顺序排列,并将无冲突链路集与该链路序列构成的矢量相联系。如果某一链路属于该无冲突链路集,则在该矢量中,该链路对应的分量取1。如果某一链路不属于该无冲突链路集,则在该矢量中,该链路对应的分量取0。这样构成的矢量称为无冲突矢量(*CFV*)。表4-1给出了与图4-24对应的几个无冲突矢量。

表 4－1　与图 4－24 对应的无冲突矢量（*CFV*）

链路	(1,2)	(2,1)	(1,3)	(3,1)	(2,4)	(4,2)	(3,4)	(4,3)	(3,5)	(5,3)
CFV_1	1	0	0	0	0	0	1	0	0	0
CFV_2	1	0	0	0	0	0	0	0	1	0
CFV_3	0	1	0	0	0	0	0	1	0	0
CFV_4	0	1	0	0	0	0	0	0	0	1

4.6.1　时分复用（TDM）在 PRNET 中的应用

在 PRNET 中，选择一组无冲突链路集（X_1，X_2，…，X），以 TDM 的方式，在一个帧中为每一个无冲突链路集分配一个时隙，如图 4－25（a）所示。在该方式中，每一个时隙内，传输的链路都不会发生碰撞，此时每条链路的利用率即为每条链路在这一组 CFV 中出现的次数除以 CFV 的数目（J）。

上述方法可以作进一步的推广，可以在一帧内为一个无冲突链路集分配多个时隙。例如，X_1 可占 2 个时隙，X_3 可占 3 个时隙等等，如图 4－25（b）所示。此时，每条链路的利用率即为一帧中每条链路出现的总次数除以一帧的时隙数。

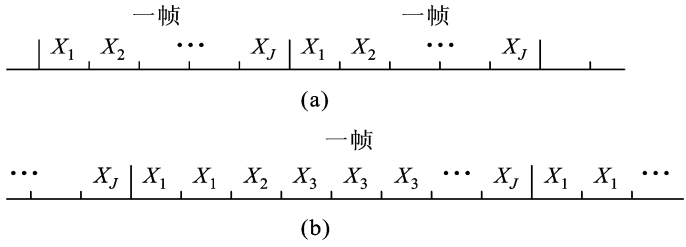


图 4－25　TDM 在 PRNET 中的应用

在 PRNET 中应用 TDM 方式的主要缺点是：当负载较轻时，时延较大。此外，在 PRNET 中，节点通常是移动的，或者说拓扑是动态变化的，这就会导致无冲突链路集经常改变，这就要求 TDM 的帧结构不断变化。要使 TDM 的帧结构能不断适应网络拓扑的变化，这通常是很困难的。另一方面，当网络节点增加时，无冲突链路集的数目随着节点数目增加呈指数增长，这将大大增加求解无冲突链路集的困难。

小　结

本章讨论的主题是多址接入协议，它主要解决多个用户如何共享信道的问

题。首先讨论了固定多址接入协议(TDMA、FDMA等)的特点并分析了它们的性能;然后讨论了最基本的随机多址接入协议ALOHA协议,主要针对它的稳态性能及其稳定性做了深入的研究,同时还利用伪贝叶斯算法构造了一个稳定的ALOHA协议;接着针对ALOHA协议利用率不高的原因,研究了载波侦听型的多址接入协议(CSMA协议),它可以有效地减少想接入信道的分组对正在传输的分组的影响。在CSMA协议基础上,还讨论了CSMA/CD协议和CSMA/CA多址接入协议。在随机多址接入协议的基础上,又讨论了冲突分解算法,给出了树形算法和FCFS算法。在研究了固定多址接入和随机多址接入协议之后,讨论了基于预约的多址接入协议。当要传输的分组较长时,可以用一个很短的分组进行预约,如果预约成功,则该分组将无冲突的进行传输。预约可以是显式的也可以是隐式的。例如在CSMA/CD中,以分组头部来进行预约,如果分组头部未与其他分组碰撞,则该分组将无冲突地进行传输。以上几种多址接入方式都是针对全连通的网络来讨论的。在本章的最后部分,还对部分连通的网络——分组无线电网络进行了讨论,并讨论了在PRNET中利用无冲突的矢量集来提高系统利用率的方法。

从前面讨论的基本协议出发,可以构造出多种类型的协议,其基本方法就是预约与冲突分组和固定分配相结合,所构造的多址接入协议不仅要支持单一的业务,而且还需支持多种不同类型的业务。这一方面仍然是多址接入协议需要研究的重点问题。

习 题

4.1 请讨论固定多址接入协议的优缺点是什么?

4.2 在ALOHA协议中,为什么会出现稳定平衡点和不稳定的平衡点,重传概率对系统的性能有何影响?

4.3 设信道数据速率为9600 bit/s,分组长度为804 bit。计算当 $G=0.75$ 时纯ALOHA系统负荷为多少。

4.4 n个节点共享一个9600 bit/s的信道,每个节点以每100 s产生一个1000 bit分组的平均速率发送数据分组。试求在纯ALOHA系统和时隙ALOHA系统中最大可容许的系统用户数 N的值。

4.5 什么叫稳定的多址接入协议?使用伪贝叶斯算法的时隙ALOHA协议是不是稳定的多址接入协议?如果是,其稳定的最大通过率是多少?

4.6 CSMA协议的基本原理是什么?与ALOHA系统相比,为什么CSMA系统有可能获得更高的系统吞吐率?

4.7 CSMA系统主要是在什么问题的处理决策上去区分三种不同类型的

CSMA 协议？说明它们各自的关键技术特点。

4 8　CSMA 方法有什么应用环境限制？在卫星信道上能采用 CSMA 接入方法吗？为什么？

4 9　假设有以下两个 CSMA／CD 网：

网络 A 是 LAN（局域网），传送速率为 5 Mbit／s，电缆长 1 km，分组长度 1000 bit；

网络 B 是 MAN（城域网），电缆长 50 km，分组长度 1000 bit。

那么，网络 B 需要多大的传送速率才能达到与网络 A 相同的吞吐量？

4 10　K 个节点共享 10 Mbit／s 的总线电缆，用 CSMA／CD 作为访问方案（即以太网 LAN）。总线长 500 m，分组长 L 比特，假设网络上的 K 个节点总有业务准备传送（重负荷情况）。P 是竞争时隙中一个节点发送分组的概率。令 $K=10$，传播速度是 3×10^8 m／s。求竞争周期的平均时隙数、竞争周期的平均持续时间及以下两种情况的信道利用率。

⑴　$L=100$ bit。

⑵　$L=1000$ bit。

4 11　试给出图 4－26 所示网络中的无冲突矢量集合。

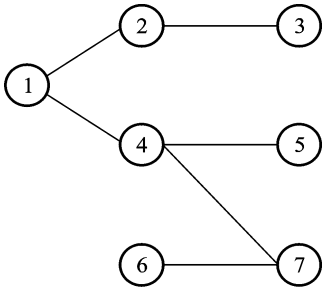


图 4－26　习题 4.11 图

第5章 路由算法

5.1 路由算法概述

在通信网络中,网络层主要负责将两个终端系统经过网络中的节点用数据链路连接起来,组成通信通路,实现两个终端系统之间数据的透明传送(所谓透明传输是指发端发送到网络接口的任何信息都会按照其原始的形式传送到接收端,网络不会修改其内容或将与该信息无关的内容送给接收者)。网络层的功能包括寻址和选择路由,建立、保持和终止网络连接等。路由算法是网络层的核心问题,其主要功能是指引分组通过通信子网到达正确的目的节点。它包括两个方面的功能:第一是为不同的源节点和目的节点对(SD)选择一条传输路径;第二是在路由选择好以后,将用户的消息正确地送到目的节点。

在分组交换网中,每个分组可以单独选路,也可以若干分组构成的序列选择相同的路径。如果子网内部采用数据报的传输方式,则对每一个分组都要重新作路由选择。因为对于相同源和目的节点的每个分组来说,上次选择的最佳路由可能由于网络拓扑、负荷和拥塞等情况的变化已被改变。然而,如果子网内部采用虚电路的传输方式,则仅需在建立虚电路时作一次路由选择,以后,数据就在这条建立起来的路由上传送。由于网络的拓扑结构在运行中可能发生变化,而且信息的流量更是随时在变化,所以一个能考虑网络当前运行情况来选择路径的适当算法,将在很大的程度上影响网络传输的可靠性和通信效率。

现在,网络设计者面临的问题是:采用什么策略来选择合适的路由?依据什么信息来进行这种选择?应该如何执行这种选择的策略?用什么标准来评判所选路径的好坏?这些都是后面需要讨论的问题。

关于路由选择的目的和要求,归纳起来大致有以下几点:

(1) 能正确、迅速、合理地传送分组(报文)信息。

(2) 能适应网络内节点或链路故障而引起的拓扑变化,使分组(报文)在有故障的条件下一般还能到达终点。在发生故障时,允许某些线路的通信量过载而增加时延。

(3) 能适应网络流量的变化,使各通路的流量均匀,整个网络的通信设备负荷平衡,充分发挥效率。

(4) 算法尽量简单,以减少网络开销。

下面通过一个简单的例子来看路由选择对网络性能的影响。

例 5.1 有一个如图 5－1 所示的网络。网络中有两个源节点和一个目的节点。所有链路的容量为 10 个单位，两个源节点 1 和 2 的输入业务量分别为 λ_1 和 λ_2，试讨论：① $\lambda_1=\lambda_2=5$ 单位时；② $\lambda_1=5$ 单位，$\lambda_2=15$ 单位时，路由选择对网络性能的影响。

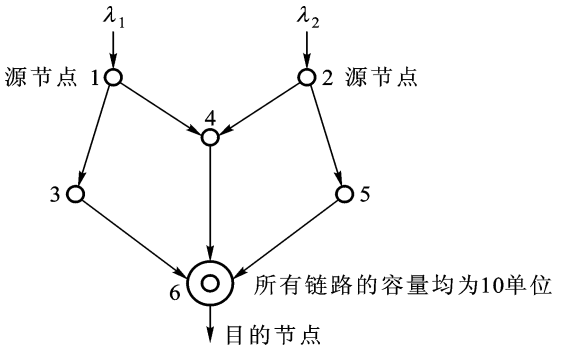


图 5－1 一个网络示意图

解 ① 如果节点 1 选择 1→3→6，节点 2 选择 2→5→6，则由于每条链路的业务量都只有信道容量的一半，因而时延很小。如果节点 1 选择 1→4→6，节点 2 选择 2→4→6，则链路 4→6 运载的业务量为 10 个单位，达到了链路的最大容量，因而时延会很大。

② 此时 $\lambda_1=5$，$\lambda_2=15$，节点 2 的输入业务量为 15 个单位。由于每条链路的容量仅为 10 个单位，在仅使用一条路径的情况下，节点 2 至少要丢弃 5 个单位的业务量。如果节点 2 将输入业务流量在 2→4→6 和 2→5→6 之间分摊，节点 1 选择 1→3→6，则每条链路上的业务流量都不超过链路容量的 75%，因而分组的时延较小。

从图中可以看出，当节点 1 和 2 输入的流量很大时，根据不同的路由选择方法，网络可接纳的最大通过量为 10～30 个单位。

由此可以看出：一个路由算法应当在高的业务负荷的情况下，在保证相同的时延条件下，可以增加网络的通过量；在轻负荷和中等负荷情况下，可以减少每一个分组的平均时延。

5.1.1 路由选择算法的分类

路由算法执行两项主要功能：源节点／目的节点对之间的路径选择，以及选定路径之后将分组传送到它们的目的地。第二项功能比较简单，它使用称为路由表的数据结构。所谓路由表通常记录了从源节点或本节点到达目的节点的路由信息，通常包括到达目的节点必须经过的下一个节点（或输出链路）以及该路

由的有关质量和利用率的度量值。一般是在每个节点设置一张路由表,用来决定该分组的输出路径。路由表应根据网络的运行情况随时加以修改、更新。每一个网络都有反映自己特色要求和决定修改路由表的原则,这些原则体现为一种算法。网络节点应根据所规定的算法,经过运算才能确定路由的选择。第一项功能通常包括一组在不同节点上运行的算法,这些算法相互之间交换必需的信息来互相支持,从而共同或单独决定一条传输路径。本章重点讨论路由算法的第一项功能。

路由算法的分类有多种方法,表5-1给出了路由选择算法的基本要素,这些要素可以用来对路由选择算法进行分类。(1)如果从路由选择算法能否随网络的业务量或者拓扑变化自适应地进行调整来划分,可分为两大类:非自适应的和自适应的。非自适应算法不根据实测或估计的网络当前业务量和拓扑结构来做路由选择。例如,从某一节点i到节点j的路由对于节点i和j都是事先计算好的,在网络启动时就下载到网络节点(路由器)中。这一过程也称作静态路由选择。这种策略的最大优点是简单和开销小。(2)如果按路由决策的方法来分,可分为:集中式和分布式。集中式路由算法是指网络的路由是由路由控制中心计算的,该中心周期性收集各链路的状态,经过路由计算后周期性地向各网络节点提供路由表。分布式路由是指网络中所有节点通过相互交换路由信息,独立地计算到达各节点的路由。(3)如果按应用场合来分,可分为:广域网路由和互联网路由。广域网中的路由主要是用来解决一个子网内的路由,而互联网中的路由主要解决不同子网之间的路由。

表5-1 路由选择算法的基本要素

分 类	要 素
决策地点	每一节点(分布式)
	中央节点(集中式)
	源节点
	节点子集
决策时间	分组(数据报)
	会话(虚电路)
性能准则	链路数
	设施代价
	时延
	吞吐量

续表

分　类	要　素
网络信息源 (与路由选择有关的信息)	无
	本地
	相邻节点
	路径上的节点
	所有节点
路由选择策略	静态——简单类算法
	自适应——更新时间:连续变更、周期性变更、主要负载改变时、拓扑改变时

5.1.2 对路由选择算法的要求

一个理想的路由算法应具有如下特点：

(1) 正确性:算法必须是正确的。即沿着各节点(交换机或路由器)中路由表所指引的路由,分组一定能够最终到达目的节点(交换机或路由器)。并且,分组到达目的节点后不会再向其他节点(交换机或路由器)转发该分组。

(2) 计算简单:算法应使用节点上最少的运行资源,这样可以节省开销、减少时延,而且应该尽量少使用节点间链路的带宽。如果为了计算合适的路由必须使用其他节点发来的大量状态信息,额外开销就会较大。

(3) 自适应性:又可称为"稳健性"或"鲁棒性"(robustness)。即算法能够适应网络业务量和拓扑的变化。当网络总的业务量发生变化时,算法能自适应地改变路由。当节点、链路出现故障或修复后重新开始工作时,算法应能及时找到一条替换的路径。

(4) 稳定性:算法必须收敛,当业务负载和拓扑变化时,没有过多的振荡。所谓振荡,是指算法得出的整个或部分路径是在多条可能路径之间来回不停地变化,而不会稳定在一条可能的路径上。

(5) 公平性:算法对所有的用户必须是等同的。例如,仅考虑使某一对用户的端到端时延为最小,它们就可能占用相对较多的网络资源,这样就明显不符合公平性的要求。

(6) 最优性:路由选择算法应该能提供最佳路由,从而使平均分组时延最小、吞吐量最大或可靠性最高。这里"最佳"可以是由多个因素决定的,如链路长度、数据率、链路容量、传输时延、节点缓冲区被占用的程度、链路的差错率、分组的丢失率等。显然不存在一种绝对最佳的路由算法,所谓"最佳"只能是相对于

某一种特定准则要求下得出的较为合理的选择而已。

实际上没有一个算法能全部满足上述要求,有的要求还可能是矛盾的。例如,要使吞吐量最大就可能会增加时延。然而,路由选择的效能可能影响到时延随吞吐量增加而增加的快慢程度,而且一个好的算法可能做到在一个较好的吞吐量门限以下,网络的时延较小,而在这个门限值以上时延会过大,这时就必须进行流控,以保证网络尽量工作在门限以下。

5.1.3 路由算法的实现 路由表

节点上的路由表指明该节点如何选择分组的传送路径,如图5-2所示的网络中,路由表的一个可能的例子如表5-2所示。

表5-2 节点上的路由表举例

节点1上的路由表						节点4上的路由表					
目的节点	2	3	4	5	6	目的节点	1	2	3	5	6
下一个节点	2	2	4	4	2	下一个节点	1	2	3	5	5

上述路径选择的原则是使到达目的节点的链路数(中转的次数或跳数hops)最少。当存在2条以上具有相同链路数的最少链路数路径时,可以选择其中任意一条。路由表对每个目的节点指出分组应发向的下一个节点(输出链路)。

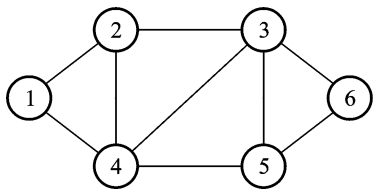


图5-2 网络拓扑举例

在分布式路由计算过程中,各节点中关于某一对节点的路由信息可能不一致。不一致的路由表可能导致乒乓(Ping Pong)效应,形成环路等现象。例如,节点 *i* 上的路由表指出到目的节点 m 的最佳路径是通过下一个节点 j,而节点 j 上的路由表又指出下一个最佳节点是 i,则分组就会在节点 i 和 j 之间来回发送。当采用分布式算法时,特别是在适应网络变化的过程中,很难消除暂时出现的路由环路。

当路由表建立起来之后,在进行路由选择时只是简单地查找路由表中的信息,无需再作计算。然而对自适应路由选择来说,会要求相当数量的计算来维持这张路由表。

通常路由表中还会包含一些附加信息,例如基于最少链路数准则的算法可能包括到达目的节点的估计链路数,这样表5-2所示的路由表要修改为表5-3所示的形式。

表 5-3 包含最少链路数的节点 1 上的路由表

目的节点	2	3	4	5	6
下一个节点	2	2	4	4	2
链路数	1	2	1	2	3

5.1.4 路由算法与流量控制的关系

路由选择算法确定数据从源节点到目的节点传送的路径,而流量控制算法是限制允许到达某些数据链路或网络某个部分的业务量,以防止这些链路或部分过分拥挤。这两个算法往往要分别加以研究,但实际上它们是密切相关的,因为若路由算法把太多的业务量引导到同一区域内,则可能引发拥挤,从而需要采用流量控制算法。

路由选择与流量控制 (简称为流控) 之间的一般关系可用图 5-3 来说明。如图所示,流控控制进入网络的吞吐量,进入网络的吞吐量影响到路由的选择,路由选择又影响到网络分组传输的时延,而这一时延又影响流控所允许进入网络的业务量,可能进一步影响到时延。路由选择和流控总是处于这种动态的交互协调之中。路由算法应当将网络中分组时延维持在很低的水平上。由于时延的存在,流控算法通过平衡通过量和时延的关系,采取必要措施来拒绝一些可能会引起网络阻塞的业务,好的流控算法应当允许更多的业务流进入网络。时延和通过量之间的严格平衡将由流控算法决定,而路由算法将决定不同的通过量对应的时延曲线。

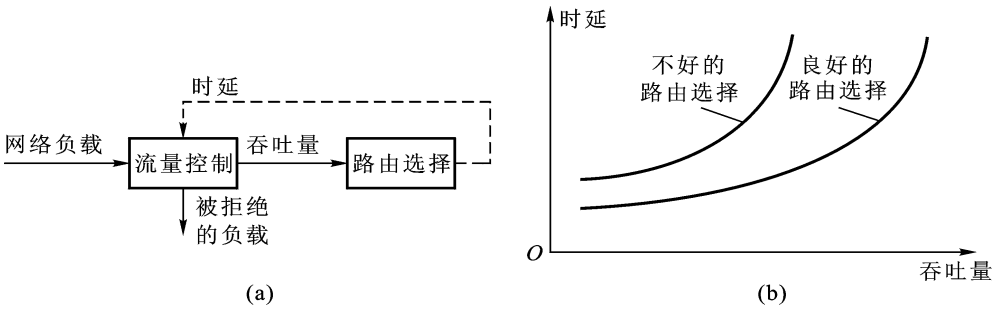


图 5-3 路由选择和流量控制之间的相互作用

5.2 常用的路由算法

不同的应用场合对路由算法有不同的要求,正如前面讨论的那样,广域网内的路由主要解决子网内分组的传输路径问题,它主要包括三种路由算法:广播、

最短路由和最佳路由。而在互连网中则主要采用分层的网络。目前还有一类网络是 Ad Hoc 网络,它是一种分布式的 PRNET 网络,该网络中使用了多种形式的路由算法。下面分别对这几种网络中的路由算法加以讨论。

5.2.1 广域网中的路由算法

1. 广播 (Broadcast)

广播是通信网中最常用的方式,它用来传播公共信息、拓扑变化信息 (包括节点和链路工作变化和故障等信息)。广播分组的接收节点通常是全网所有成员。如果接收节点仅为一个组或部分网络节点,则称为多播 (multicast)。广播时采用的路由算法可以有多种方法:如泛洪 (flooding) 路由、采用生成树 (spanning tree) 的广播方式等。当然也可以逐一地把要广播的分组按照点对点的路由算法 (unicast) 发送给每一个目的节点,但这种方法可能会浪费大量的网络资源,并且广播节点需要知道全网所有节点的路由信息。

泛洪路由的基本想法是源节点 (发起广播的节点) 将消息以分组的形式发给其相邻的节点,相邻的节点再转发给它们的相邻节点,继续下去,直至分组到达网络中所有的节点。为了限制分组的传输次数,需要两个附加规则:

(1) 若节点 B 是从 A 收到一个广播分组,则 B 不会将该广播分组再转发给 A;

(2) 每个节点仅将相同的广播分组转发给邻节点最多一次。具体的实现方法是:源节点广播的每一个分组都有一个标识符 (ID) 和序号,每发送一个新的分组,序号加 1。每个节点在收到一个广播分组后,要检查该分组的标识符和序号,如果该分组的序号大于记录中具有相同标识符分组的最大序号,则中转该分组并记录其标识符和序号,所有小于或等于记录序号的分组都被丢弃,而不会被中转,分组的广播过程如图 5-4 (a) 所示。图中箭头上的标号表示该分组被中转的次数。图中 A 是广播的发起节点。设 *L 为网络的链路数,该方法的分组传输次数在 L~2L 之间。为了减少广播分组传输的次数,可以采用图* 5-4 (b) 的方法,首先构造一个生成树,在该树上分组仅需传输 N-1 次 (N 为网络的节点数) 即可。

2. 最短路由 (shortest path routing)

许多实际的路由算法如 RIP (Routing Information Protocol),OSPF (Open Shortest Path First) 等都是基于最短路径这一概念。分组交换网络的各种路由算法实质上都是建立在某种形式的最小费用准则的基础上。譬如,把准则定为“最短路径”,那就有所谓“最短路径路由算法”这里所说的“最短路径”并不单纯意味着一条物理长度最短的通路,它可以是从发送节点到达接收节点的中转次数最少。

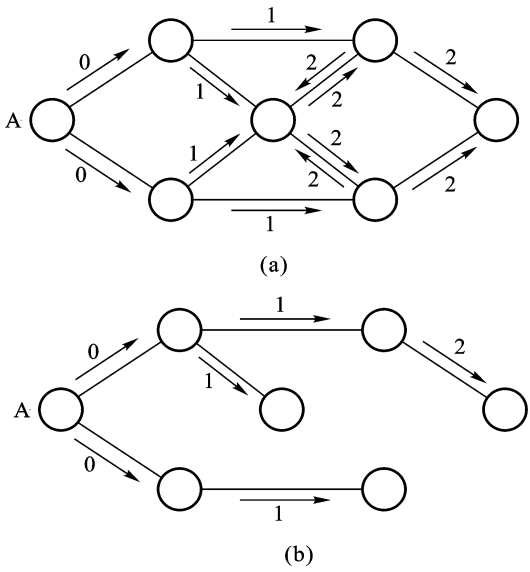


图5-4 广播的基本工作示意图
(a) 泛洪广播 (b) 生成树广播

最短路由的一个关键是如何定义“费用”。如果最关心分组的时延,则可以把“费用”与时延相关联。此时每条链路的“费用”明显地与两个参数有关:链路的物理长度和链路上的业务强度。前者决定信道的传播时延,后者决定分组的发送等待时延。因此,如果能将上述两个参数的值折算为该链路的费用或“长度”值(时延的大小),则最小费用算法也就等效为最小时延路由算法。所以,所谓“最短”取决于对链路长度的定义。长度通常是一个正数,它可以是物理距离的长短、时延的大小、各个节点队列长度等等。如果长度取1,则最短路由即为最小跳数(中转次数)的路由。其次,链路的长度随着时间可能是变化的,它取决于链路拥塞的情况。下一节将详细地讨论最短路由算法。

3. 最佳路由(optimal routing)

上述最短路由仅关心的是一个节点对之间的一条路径的选择和求解,因而有两个方面的缺陷:一是为每对节点之间仅提供一条路由,因而限制了网络的通过量;二是适应业务变化的能力受到防止路由振荡的限制。而最佳路由是从全网的范围寻找所有可能的传输路径,从而使得发送节点到达接收节点的信息流的时延最小、流量最大,而不是局限于一条所谓的最短路径。因此,采用最佳路由(基于平均时延最佳化)可以克服最短路径的上述缺陷,它可以将节点对之间的流量分配在多条路径上,从而可使网络的通过量最大,时延最小。

5.2.2 互连网中的路由算法

为了实现网络之间的互连,通常采用三种设备:网关、网桥和路由器。实现广域网(WAN)至广域网(WAN)之间的互连设备称为网关(gateway)。它完成相当复杂的网络层的任务,包括协议转换、路由功能等。它通常在网际子层。实现局域网(LAN)与局域网(LAN)之间在MAC层互连的设备称为网桥(bridge)。实现LAN与WAN或LAN与LAN之间互连的设备称为路由器(Router),它提供高级的路由功能。一个典型的互连网络如图5-5(a)所示。

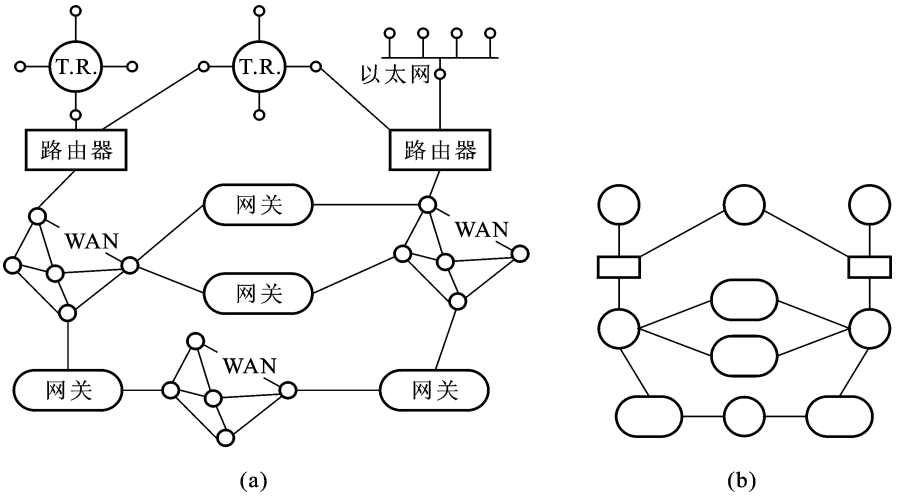


图5-5 互连的网络示意图

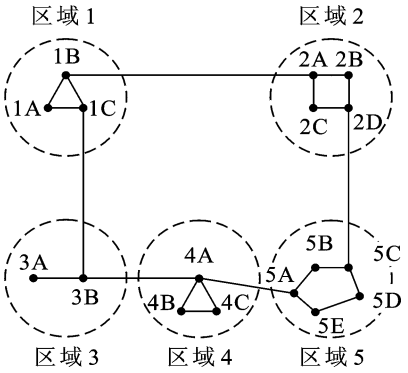
(a) 典型的互连网络 (b) 互连网络示意图

可以以两种观点来看待一个互连的网络,一是将互连的设备看成是一个附加的网络节点,它与网络中其他节点的地位等同,所有的节点组成一个更大的网络。网络中的每一个节点都维持一个到达各个节点的路由表(这个表通常会很大)。第二种观点是把每个子网看成是一个节点,如图5-5(b)所示。这样将网络分为两层,高层是由互连设备和子网组成的网络,低层是各子网的内部网络。在这种分层的方式中,Gatewav或Router只维持到达各个子网的路由表,各个子网仅维持子网内的路由表。这样路由表的维持和修正的负荷相对较小和易于操作。分层的缺点是所形成的路由对整个网络而言不一定是最佳的。

从上面的讨论中可以看到,随着网络的增大,路由表中存储的内容也将不断的增大。增大的路由表不仅占用路由器的内存,而且需要更多的CPU时间扫描表格,以及需要更大的链路容量来传送关于路由表的状态报告。因此,为了实现充分利用有限资源的同时,还可以实现网络扩展,必须进行分级路由选择。

所谓分级路由选择是指:将路由器划分为区域,每个路由器仅知道怎样在其所属区域内选择路由和知道分组在该区域要到达的目的端的全部细节,但并不知道其他区域的内部结构。当不同的网络相连时,很自然地将每个网络看作为独立的区域,以便让一个网络中的路由器免于知道其他网络的拓扑结构,从而有效地减少了每个路由表的存储内容。

对于大型的网络而言,通常需要分成多级。图5-6给出了在有五个区域的两级结构中作路由选择的一个例子。路由器1A的整个路由表共有17个表项,如图5-6(b)所示。在图5-6(c)中,所有其他区域都被抽象为一个单独的路由器,大大节省了存储空间。因此,到区域2的所有业务都经过1B-2A这条路径,其余的业务量都经过1C-3B这条路径。分级路由选择将路由器1A的路由表从17个表项减少到7个。如果区域数和区域内的路由器的比例增大时,节省的存储空间也会按比例增大。



1A的完整表

目的	线路	站段数
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

1A的分层表

目的	线路	站段数
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(a) (b) (c)

图5-6 分级路由

(a) 网络结构 (b) 不分级时的路由表 (c) 分级时的路由表

5.2.3 Ad Hoc网络中的路由算法

Ad Hoc网络并非一个全新的概念。实际上,20世纪70年代由美国国防高

级研究计划局(DARPA)提出并应用的分组无线网(PRNET)可以看作是Ad Hoc网络的先驱。但提出分组无线网络的初衷是用于军事目的,而Ad Hoc网络技术却扩展到更多的领域。传统的路由算法(如距离矢量算法和链路状态法等)基本上是为有线网络设计的,没有考虑到网络的动态特性。而且在传统的路由算法中,网络管理的开销随着网络的规模增大而迅速增长。上述这些因素恰恰是移动Ad Hoc网络中要关注的重要因素,而且移动Ad Hoc网络还面临着无线信道的不可靠性、高速移动环境下链路频繁中断、故障以及节点的能量有限等情况。很显然,上述这些传统的路由算法不可能直接应用到Ad Hoc网络中。更为重要的是,传统的路由算法中都存在着一些致命的缺陷,如路由闭环、收敛速度慢等问题,因此,必须研究新的路由策略来适应移动Ad Hoc网络的特殊性。为了适应移动Ad Hoc网络中对路由算法的新要求,目前MANET(Mobile Ad Hoc Network)已经在距离矢量算法和链路状态法的基础上,提出了许多改进型的路由协议。同时,也有许多协议是直接从有线网络继承改进得到的。总的来说,可以把目前单播的Ad Hoc路由算法分为以下三种(如表5-4所示):

表5-4 Ad Hoc网络路由算法分类

Ad Hoc路由协议			
平面式路由		分层路由	地理位置辅助的路由
Proactive (表驱动路由)	Reactive (按需路由)	ZRP HSR CGSR LANMAR	GeoCast LAR DREAM GPSR
DSDV WRP FSR FSLS	AODV DSR TORA SSR		

(1) 平面式路由(flat routing)算法。网络中的所有节点都处于同一层次上,各节点在网络中获得的路由信息基本相同。根据其设计的具体原则可进一步的将平面式路由分为proactive routing算法和reactive routing算法。

(2) 分层路由(hierarchical routing)算法。网络按一定的规则分为多个不同的层次,在不同层次中又可以有不同的路由策略。分层的路由策略比较容易进行网络规模的扩充。

(3) 地理位置辅助的路由(geographic position assisted routing)算法。网络中的节点可以获得节点的地理位置信息,通过这些信息可以有效地降低路由算法中路由建立或维护的开销。

5 3 最短路由算法

上节已经提到,许多实际的路由算法如 RIP,OSPF 等都是基于最短路由这一概念。这里首先要明确最短的含义,它取决于对链路长度的定义。长度通常是一个正数,它可以是物理距离的长短、时延的大小、各个节点队列长度等等。如果长度取 1,则最短路由即为最小跳数(中转次数)的路由。其次,链路的长度随着时间可能是变化的,它取决于链路拥塞情况。最短路由算法的理论基础是图论,本节将首先讨论图论的基本概念,然后讨论常用的最短路由算法。

每一个网络都可以抽象成一个图。一个图 *G 由一个非空的节点集合 N 和*节点间的链路 A 组成,即 G=(N,A)。链路可以是有方向的,也可以是无方向的。如果节点 i 和 j 之间仅有 i→*j 的链路,则称该链路是有方向的(或单向链*路)。如果节点 i 和 j 之间同时有 i→*j 及 i→j 的链路,则称该链路是无方向的*(或双向链路)。

通常在一个网络中,考虑业务流量问题时,必须要考虑到业务的流向问题。这样就需要使用方向图的概念。方向图 G=(N,A)对应的无方向图 G′=(N′,A),其中 N′=*N,如果(i,j)∈A 或(j,i)∈A 或两者同时成立,则(i,j)∈A′。*

定义几个术语:

关联(Incident):它表示链路与节点的关系。一条链路(n_1,n_2)关联的两个节点是 n_1 和 n_2。这里讨论的图不允许有一条链路关联的两个端点相同(节点不允许有一条自环的链路)。

方向性行走(Walk):是一个节点的序列(n_1,n_2,…,n_l),该序列中关联的链路(n_i,n_{i+1}),1≤*i*≤*l*−1 是 *G 中的一个链路。*

方向性 Path:指无重复节点的方向性行走。

方向性环(Cycle):指开始节点和目的节点相同的方向性 Path。这里需要注意的是:(n_1,n_2,n_1)在方向图中是一个环。但在无方向图中,如果(n_1,n_2)是一条无方向性链路,则(n_1,n_2,n_1)不可能是一个环。

强连通方向图:指对于每一对节点 i,j 都有一条方向性路径。

连通的方向图:指如果方向图对应的无方向图是连通的,则该方向图是连通的。

给每条链路(i,j)指定一个实数 d_{ij} 作为其长度,则一条方向性路径 *p*=(i,j,k,…,l,m)的长度就是各链路长度之和,即 $d_{ij}+d_{jk}+\cdots+d_{lm}$。最短路径问题就是寻找从 i 到 m 的最小长度方向性路径。

最短路径问题有着非常广泛的应用,例如:如果定义 d_{ij}=使用链路(i,j)的成本,则最短路径就是从 i 到 m 发送数据成本最低的路由。如果定义 d_{ij}=分

组通过链路(i,j)的平均时延,则最短路径就是最小时延路由。如果定义 $d_{ij}=-\ln P_{ij}$,P_{ij}是链路(i,j)的可用概率,则寻找最短路径问题就是寻找 $i\to m$ 最可靠的路径。

5.3.1 集中式最短路径算法

本节讨论三种标准的集中式最短路径算法:Bellman-Ford 算法,Dijkstra 算法和 Floyd-Warshall 算法。其中 Bellman-Ford 算法和 Dijkstra 算法是点对多点的最短路径算法,而 Floyd-Warshall 算法则是多点对多点的最短路径算法。

1. Bellman-Ford 算法

典型的 Bellman-Ford 算法(简记为 B-F 算法)是一种集中式的点到多点的路由算法,即寻找网络中一个节点到其他所有节点的路由。如图 5-7 所示的网络中,假定节点 1 是"目的节点",要寻找网络中其他所有的节点到目的节点 1 的最短路径。假设每个节点到目的节点至少有一条路径。用 d_{ij}表示节点 i 到节点 j 的长度。如果(i,j)不是图中的链路,则 $d_{ij}=\infty$。

定义:最短($\le h$)*行走*(Walk)是指在下列约束条件下从给定节点 i 到目的节点 1 的最短 Walk。

① 该行走(Walk)中最多包括 h 条链路,即 Walk 中包含的链路数至多为 h 条。

② 该行走(Walk)仅经过目的节点 1 一次。

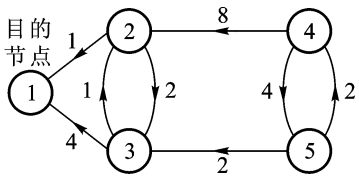


图 5-7 一个网络的示意图

最短($\le h$)*行走*(Walk)长度用 D_i^h 表示。(这样的 Walk 不一定是一条路径,它可能包含重复节点,但在一定的条件下,它将不包含重复节点。)对所有的 h,令 $D_i^h=0$。B-F 算法的核心思想是通过下面的公式进行迭代,即

$$D_i^{h+1}=\min_j [d_{ij}+D_j^h] \qquad \text{对所有的 } i\neq 1 \tag{5-1}$$

下面给出从 h 步 Walk 中寻找最短路由的算法。

第一步:初始化。即对所有 i($i\neq 1$),令 $D_i^0=\infty$。

第二步:对所有的节点 j($j\neq i$),*先找出一条链路的最短*($h\le 1$)的 Walk 长度;

第三步:对所有的节点 j($j\neq i$),*再找出两条链路的最短*($h\le 2$)的 Walk 长度;

依次类推:如果对所有 i 有:$D_i^h=D_i^{h-1}$(即继续迭代下去以后不会再有变化),则算法在 h 次迭代后结束。

例 5.2 请描述图 5-8 中节点 4 到节点 1 的路由迭代过程。

解　在第一步中，由于仅可使用一条链路，故 $D_4^1=\infty$。在第二步中，在仅使用两条链路的情况下，节点 4 通过节点 2 到达目的节点 1 的 $D_4^2=8+D_2^1=9$。在第三步中，节点 4 不能通过引入新的链路来减少 Walk 的长度，因此其路由不变。在第四步中，节点 4 通过节点 5 到达目的节点 1 的 $D_4^4=4+D_5^3=8$ 要小于节点 4 通过节点 2 到达目的节点 1 的 $D_4^3=9$，故节点 4 最终选择通过节点 5 作为到达目的节点 1 的路由。具体过程如图 5-8 所示。

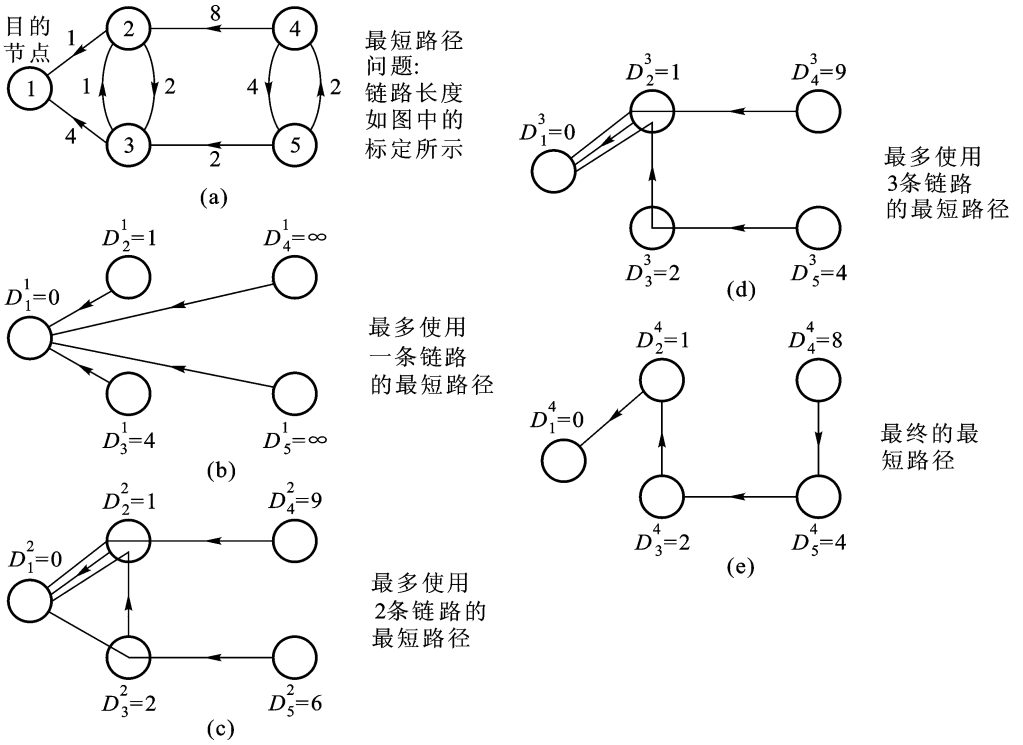


图 5-8　B-F 算法的迭代过程

(a) 网络举例　(b) 第一步的迭代结果　(c) 第二步的迭代结果

(d) 第三步的迭代结果　(e) 最终的迭代结果

上述算法计算的是最短 Walk 的长度。下面的定理 1 给出了最短 Walk 长度等于最短路径长度的充分必要条件。

定理 1：对于式 (5-1) 的 B-F 算法 (初始条件：对所有 $i\neq1$，有 $D_i^0=\infty$)，有：

(1) 由该算法产生的 D_i^h 等于最短 ($\leq h$) Walk 的长度；

(2) 当且仅当所有不包括节点 1 的环具有非负的长度，算法在有限次迭代后结束。此外，如果算法在最多 $k\leq N$ *次迭代后结束，则结束时* D_i^h 就是从 i 到 1 的最短路径长度。

定理 1 中 (1) 阐明了 D_i^h 与最短 ($\leq h$) Walk 的关系。(2) 阐明了算法何时结

束,结束时所得的结果是否是最短路径。

证明 采用归纳法证明(1)。

① 因为 $D_i^1 = d_{i1}$,所以显然有 D_i^1 等于最短(≤ 1)的 Walk 长度;

② 假定 D_i^h 是等于最短($\leq h$)的 Walk 长度,求证 D_i^{h+1} 是等于最短($\leq h+1$)的 Walk 长度。

从图5-8中已经体会到,从 i 到1的最短($\leq h+1$) Walk 包含的链路数有两种情况:一种情况是链路数小于 h+1,在此情况下,有 Walk 长度等于 D_i^h;另一种情况是链路数等于 h+1。在后一种情况下,有

$$\text{最短}(\leq h+1)\ \text{Walk 长度} = \min\{D_i^h, D_i^{h+1}\} = \min\{D_i^h, \min_{j\neq 1}[d_{ij} + D_j^h]\} \quad (5-2)$$

根据 D_i^h 等于最短($\leq h$) Walk 的假设,有 $D_j^k \leq D_j^{k-1}$,对所有的 $k \leq h$。[这是因为从 k 到1的($\leq k$) Walk 的集合包含了($\leq k-1$) Walk 的集合,在更大范围内求极小值必然越来越小。]因此有

$$D_i^{h+1} = \min_j [d_{ij} + D_j^h] \leq \min_j [d_{ij} + D_j^{h-1}] = D_i^h \quad (5-3)$$

代入式(5-2)得

$$\text{最短}(\leq h+1)\ \text{Walk 长度} = D_i^{h+1} \quad (5-4)$$

至此定理1的第(1)部分得到证明。

③ 如果 B-F 算法在 h 次迭代后结束,即有

$$D_i^k = D_i^h \qquad \text{对所有 } i \text{ 和 } k \geq h \quad (5-5)$$

则不可能通过添加更多的链路来减少最短的 Walk 长度。(否则,算法没有结束。)也就是不可能存在一个负长度的(不包括目的节点)环。因为这样的负长度的任意大次数的重复将使 Walk 的长度任意的小,这与式(5-5)相矛盾。

相反,假定所有的不包括1的环具有非负的长度。从最短($\leq h$) Walk 中删除这些环,会得到长度相同或更短的路径。因此,对每一个 i 和 h,总存在一条从 i 到1的最短($\leq h$) Walk,其相应的最短的路径长度等于 D_i^h。由于路径中没有环,路径可能包括最多 N-1 条链路。因此,$D_i^N = D_i^{N-1}$,对所有 i 成立。即算法在最多 N 次迭代后结束。

上面的讨论中,关心的主要问题是最短路径的长度。现在把重点移到如何直接来构造最短的路径。

假定所有不包括节点1的环具有非负的长度,用 D_i 表示从节点 i 到达目的节点1的最短路径长度。根据前面的讨论,当 B-F 算法结束时,有:

$$D_i = \min_j [d_{ij} + D_j] \qquad \text{对所有 } i \neq 1$$

$$D_1 = 0 \quad (5-6)$$

该式称为 Bellman 方程。它表明从节点 i 到达目的节点1的最短路径长度,等于 i 到达该路径上第一个节点的链路长度,加上该节点到达目的节点1的最短

路径长度。从该方程出发，只要所有不包括1的环具有正的长度（而不是0长度）的情况下，可以很容易地找到最短路径（而不是最短路径长度）。具体方法如下：

对于每一个节点 $i \neq 1$，选择一条满足 $D_i = \min_j [d_{ij} + D_j]$ 的最小值的链路 (i, j)，利用这些 $N-1$ 条链路组成一个子图，则 i 沿该子图到达目的节点1的路径即为最短路径。如图5-9所示。

利用上面的构造方法可以证明：如果没有0长度（或负长度）的环，则Bellman方程式(5-6)（它可以看成一个含有 N 个未知数的 N 个方程的系统）有惟一解。（如果有不包括节点1的环的长度为0，则Bellman方程不再具有惟一解。注意：路径长度惟一，并不意味着路径惟一。）

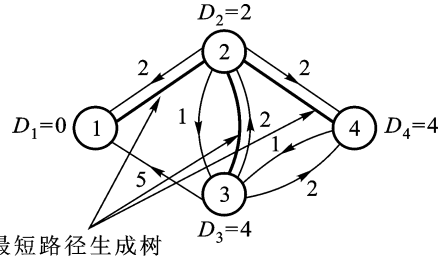


图5-9 最短路径生成树的构造

利用该结论可以证明：即使初始条件 $D_i^0, i \neq 1$ 是任数（而不是 $D_i^0 = \infty$），B-F算法都能正确工作，对于不同的节点，迭代的过程可以以任意顺序并行进行。

2. Dijkstra算法

Dijkstra算法也是一种典型的点对多点的路由选择算法，即通过迭代，寻找某一节点到网络中其他所有节点的最短路径。Dijkstra算法通过对路径的长度进行迭代，从而计算出到达目的节点的最短路径。其基本思想是按照路径长度增加的顺序来寻找最短路径。假定所有链路的长度均为非负。显然有：到达目的节点1的最短路径中最短的肯定是节点1的最近的邻节点所对应的单条链路。（由于链路长度非负，所以任何多条链路组成的路径的长度都不可能短于第一条链路的长度。）最短路径中下一个最短的肯定是节点1的下一个最近的邻节点所对应的单条链路，或者是通过前面选定的节点的最短的有两条链路组成的路径，依次类推。

Dijkstra算法通过逐步标定到达目的节点路径长度的方法来求解最短的路径。

设每个节点 i 标定的到达目的节点1的最短路径长度估计为 D_i。如果在迭代的过程中，D_i 已变成一个确定的值，称节点 i 为永久标定的节点，这些永久标定的节点的集合用 P 表示。在算法的每一步中，在 P 以外的节点中，必定是选择与目的节点1最近的节点加入到集合 P 中。*具体的*Dijkstra算法如下：

(1) 初始化，即 $P = \{1\}$，$D_1 = 0$，$D_j = d_{j1}$，$j \neq 1$。（如果 $(j,1) \notin A$，则 $d_{j1} = \infty$）。

(2) 寻找下一个与目的节点最近的节点，即求使下式成立的 i，$i \notin P$

$$D_i = \min_{j \notin P} D_j \tag{5-7}$$

置 $P=P\cup\{i\}$。如果 P 包括了所有的节点，则算法结束。

(3) 更改标定值，即对所有的 $j\notin P$，置

$$D_j=\min_i [D_j, d_{ji}+D] \tag{5-8}$$

返回第 (2) 步。

Dijkstra 算法的应用如图 5－10 所示。

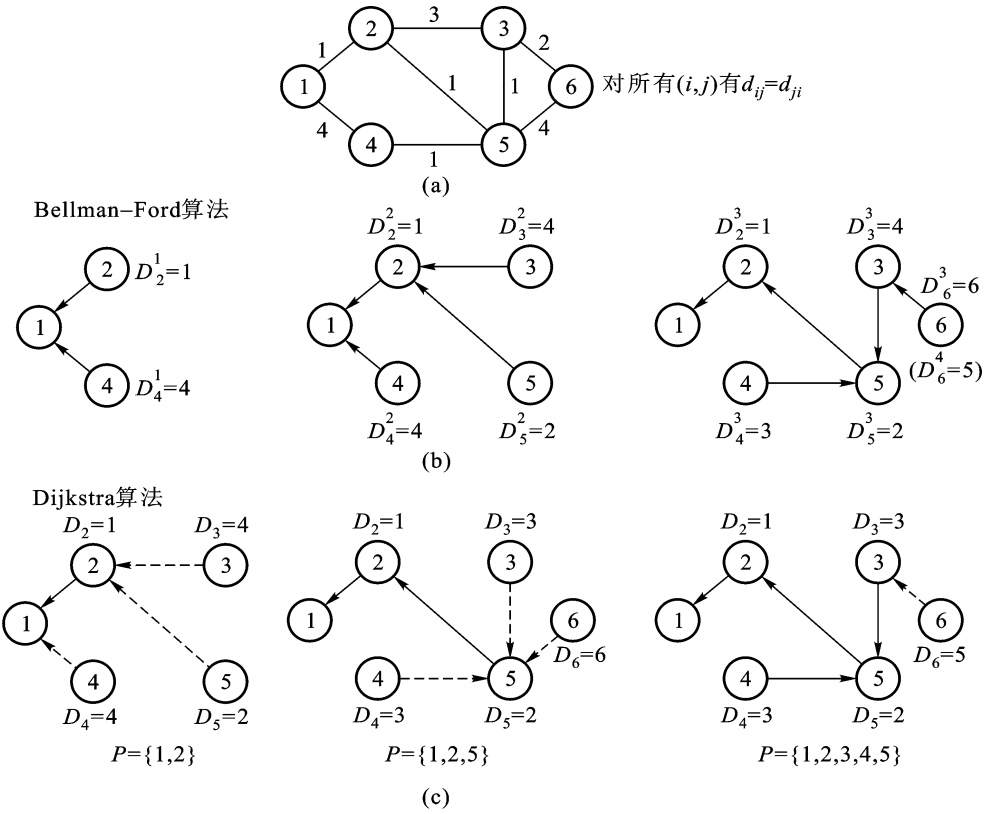


图 5－10 Dijkstra 算法和 B－F 算法应用举例

(a) 网络拓扑结构 (b) Bellman－Ford 算法 (c) Dijkstra 算法

Dijkstra 算法的迭代过程如图 5－10 (c) 所示，第一次迭代 (第一张图) 到达目的节点 1 的单条链路最近的是链路 (2,1)，$D_2=1$，$P=\{1,2\}$，其余的节点 (3,4,5) 相应地修改其标定值。第二次迭代 (第二张图)，下一个最近的节点是 5，$D_5=2$，$P=\{1,2,5\}$，其余的节点 (3,4,6) 相应的修正其标定值。第三次迭代 (第三张图) 下一个最近的节点是 3 和 4，$D_3=3$，$D_4=3$，$P=\{1,2,3,4,5\}$，还剩节点 6，$D_6=5$。再经过一次迭代，P 中将包括所有节点，算法结束。

在图 5－10 中还给出了 B－F 算法的迭代过程。很显然，在最坏的情况下，Dijkstra 算法的复杂度为 $O(N^2)$，而 B－F 算法的复杂度为 $O(N^3)$。即 Dijkstra

算法的复杂度要低于 B－F 算法。同时，从 Dijkstra 算法的讨论过程中可以看到：

① $D_i \leq D_j$，对所有 $i \in P, j \notin P$。

② 对于每一个节点 j，D_j 是从 j 到目的节点 1 的最短距离。该路径使用的所有节点（除 j 以外）都属于 P。

3. Floyd－Warshall 算法（F－W 算法）

前面讨论的 B－F 算法和 Dijkstra 算法都是求解所有节点到一个特定的目的节点之间的最短路径，而 F－W 算法则是多点对多点的路由选择算法。即 F－W 算法是寻找所有节点对之间的最短路径。其基本思想是在 $i \to j$ 的路径之间通过添加中间节点来减小路径的长度。

在 F－W 算法中，假定链路的长度可以是正或负，但不能具有负长度的环。F－W 算法开始时，以单链路（无中间节点）的距离作为最短路径的估计。然后，在仅允许节点 1 作为中间节点的情况下，计算最短路径，接着，在允许节点 1 和节点 2 作为中间节点的情况下计算最短距离，以此类推。其具体描述如下：

令 D_{ij}^n 是可以用 1,2,…,n 作为中间节点的从 i 到 j 的最短路径长度，则算法开始时 $D_{ij}^0 = d_{ij}$，对所有 i，j，$i \neq j$。

对于 n=0,1,…,N－1，有

$$D_{ij}^{n+1} = \min [D_{ij}^n, D_{i(n+1)}^n + D_{(n+1)j}^n] \quad \text{对所有 } i \neq j \qquad (5-9)$$

上式是已知 i 到 j 的最短路径 D_{ij}^n（以 1,2,…,n 作为中间节点）的条件，如何计算在 i 到 j 的最短路径上可添加节点 n+1 后的最短路径长度。在允许添加节点 n+1 的情况下，有两种可能性：一种是最短的路径将包含节点 n+1，此时的路径长度为 $D_{i(n+1)}^n + D_{(n+1)j}^n$；另一种可能是节点 n+1 不包括在最短路径中，此时路径长度等同于用 1,2,…,n 作为中间节点的路径长度。因此，最终的最短路径长度应取上述两种可能情况下的最小值，即有式（5－9）成立。F－W 算法的计算复杂度和 B－F 算法的一样，都是 $O(N^3)$。

前面讨论的三种最短路由算法的构造方法，都是通过迭代的过程求得最终结果，但其主要差别是迭代的内容不同。在 B－F 算法中，迭代的是路径中的链路数，即使用一条，两条，…，直到 N－1 条链路。在 Dijkstra 算法中迭代的是路径的长度，即最短长度，次短长度，…。而在 F－W 算法中，是对路径的中间节点进行的迭代，即一个中间节点，两个中间节点，…。

5.3.2 分布式最短路径算法

这种路由选择策略是每个节点周期性地从相邻的节点获得网络状态信息，同时也将本节点做出的决定周期性地通知周围的各节点，以使这些节点不断地根据网络新的状态更新其路由选择。所以整个网络的路由选择经常处于一种动

态变化的状态。各个节点的路由表相互作用，是这种路由选择算法的特点。当网络状态发生变化时，必然会影响到许多节点的路由表。因此，要经过一定的时间以后，各路由表中的数据才能达到稳定的数值。也就是说，分布式路由选择算法的核心思想是各个节点独立的计算最短路径。典型的分布式最短路径选择算法有距离矢量路由算法和链路状态路由算法。

1. 距离矢量路由算法

距离矢量路由算法（distance vector routing）算法是B－F算法的具体实现。它最初用于ARPANET路由选择算法，也用于Internet网中（称为RIP）以及用于DECnet和Novell的IPX的早期版本之中。AppleTalk和Cisco路由器中使用了改进型的距离矢量协议。

在距离矢量路由表中，每个路由器维护一张路由表，该表中记录了到网络中其他所有节点的路由信息。包括到该目的节点的下一跳节点（即本节点通过哪个邻节点到达指定的目的节点），和到达该目的节点所需的“距离”的估计值。以图5－11（a）所示的网络拓扑为例，节点J收到的邻节点矢量如图5－11（b）所示，J的新路由表如图5－11（c）所示。

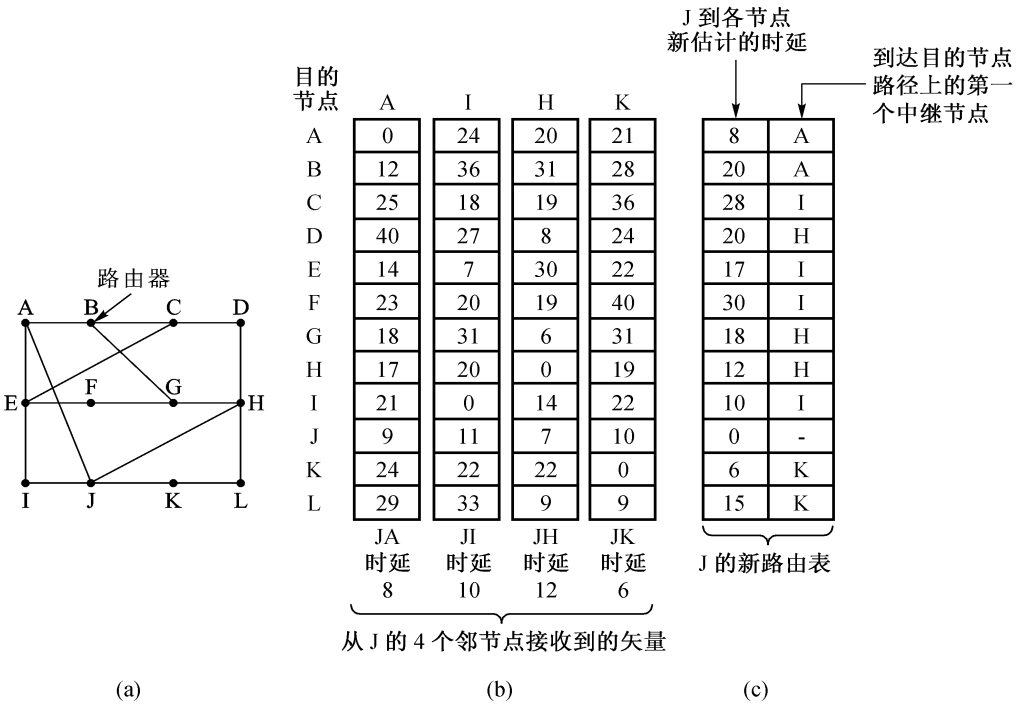


图5－11 距离矢量法应用说明

（a）网络拓扑结构 （b）从J的4个邻节点接收到的矢量 （c）J的新路由表

在表中使用的距离量度可以是跳数、时延(ms)、某一路径排队的总分组数或者其他类似的量度。每一个节点都确知它的每一个邻节点的距离。如果采用时延作为距离的量度,每个节点应当能够利用一个特殊的"回声"(ECHO)分组来直接测量该时延。接收节点收到"回声"分组后,只对它加上时间标记后就立即送回。

这里以时延作为距离的量度。每隔 T 秒,每个节点向它的每个邻节点发送一个路由信息分组,该分组包括了发送节点已知的目的节点的下一跳节点和时延估计值。同样,每一个节点都会收到它所有的邻节点发送来的路由信息分组,如图 5-11(b)所示。节点 J 收到了它四个邻节点 A,I,H,K 的路由信息分组。来自任一邻节点 X 的路由分组的某一项的取值 x_I,表明节点 X 到达目的节点 I 的时延为 x_I。如果本节点 W 确知到达该邻节点 X 链路(W,X)的时延为 m_X,则可以求得 W 通过 X 到达目的节点 I 的时延 $m_X + x_I$。本节点 W 可以比较不同邻节点(比如 A、I、H、K)到达相同目的节点(比如 G)的时延,其取值分别为(8+18),(10+31),(12+6)和(6+31),从中选择最短时延(即通过 H 的时延最短)的路径。

(1) 计数至无穷问题

距离矢量路由算法在理论上是可以正常工作的,但在实际运用中却有很大的缺陷。虽然它能得出正确的结论,但有可能太慢。特别是它对好消息的反应迅速,但对坏消息却反应迟钝。假定一个节点 I 到达一个目的节点 X 的最短距离很大。如果下一次收到的路由信息分组中,A 突然报告它到目的节点 X 的时延很短,则 I 会立即将最短路由切换到通过 A 的链路去往目的节点 X。即通过一次信息交换,好信息即被处理。假定有一个网络如图 5-12(a)所示,这里采用的距离为跳数,每一条链路的长度为 1 跳(图中标明了各节点到达目的节点 A 的跳数)。假定开始时节点 A 处于关闭状态,因而各节点到达 A 的跳数均为无穷大。当 A 开始正常工作以后,B 在收到 A 的一次路由信息分组后,会立即将到达 A 的距离置为 1。同样 C、D、E 在收到邻节点的路由信息分组后,均会立即将其到达目的节点 A 的最短距离修正到正确的取值。因此经过 4 次交换以后,好消息传遍整个网络。如图 5-12(b)所示。

下面讨论坏消息的传播速度。如图 5-12(b)所示。假定开始时,网络中各节点有到达目的节点 A 的正确路由。假定节点 A 故障或链路 A\ B 中断。在第一次路由信息交换的过程中,节点 B 没有听到节点 A 的任何消息,但节点 C 报告它有到达节点 A 的路由,距离为 2。由于节点 B 不知道节点 C 是通过本节点到达 A 的,B 可以认为节点 C 有多条独立的长度为 2 的到达 A 的路由。这样 B 就认为它可以通过 C 到达 A,其距离为 3。在第二次信息交换过程中,C 已发现它的每一个邻节点都认为到达 A 的长度为 3。因此,C 随机的选择一个邻节

点作为到达 A 的路由，并将到达 A 的距离修改为 4。后续的交换和路由修正过程如图 5－12 (b) 所示。从图中可以看出，坏消息传播很慢，没有一个节点会将其距离设置成大于邻节点报告的最小距离值加 1，所有的节点都会逐步地增加其距离值，直至无穷大。该问题称为“计数至无穷问题”(count to infinity) 问题或“坏消息现象”(bad news phenomenon)。在实际系统中，可以将无穷大设置为网络的最大跳数加 1。但是当采用时延作为距离的长度时，将很难定义一个合适的时延上界。该时延的上界应足够大，以避免将长时延的路径认为是故障的链路。

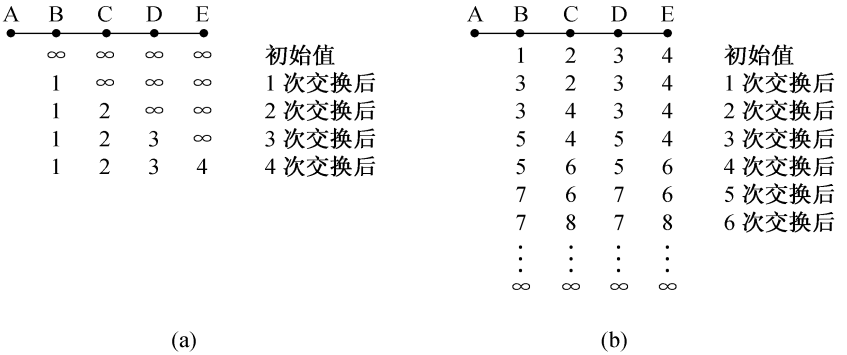


图 5－12 计数至无穷问题举例

(2) 水平分裂算法

理论上已经提出了许多解决计数至无穷问题的办法。但这些办法都比较复杂。这里介绍一种“水平分裂算法”(Split Horizon)。水平分裂算法与距离矢量算法的工作过程基本一样，不同之处在于如果节点 I 到达某一目的节点 J 的距离是通过节点 X 得到的，则节点 I 将不会向节点 X 报告有关节点 J 的信息 (即节点 I 向节点 X 报告的到达节点 J 的距离为无穷大)。例如，在图 5－12 (b) 中，C 告诉 D 它到 A 的真实距离，但它告诉 B 它到 A 的距离为无穷大。

水平分割法虽然能够解决一些计数至无穷问题，但有时候也不能正常工作。如图 5－13 所示的 4 个节点的子网，初始化时，A、B 到 D 的距离都为 2，到 C 的距离为 1。假设链路 CD 故障。如果采用水平分裂法，A 和 B 都会告诉 C 它们不能到达 D。因此，C 立即将得到结论，D 是不可达的。并通知 A 和 B。不幸的是，A 听到 B 有一条到 D 的长度为 2 的路径，因此，它认为能通过 B 经过 3 个节点到达 D。类似的，B 也认为能通过 A 到达 D，且将到 D 的距离置为 3。在以后的路由信息交换中，A 和 B 都会逐渐把到达节点 D 的距离不断增加直至无穷大。

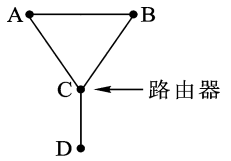


图 5－13 水平分割不能正常工作举例

2. 链路状态路由算法

在1979年之前,ARPANET都是采用的距离矢量路由算法。之后,就用链路状态路由算法(link state routing)取代了距离矢量路由算法。其主要原因有两个。第一,因为在距离矢量算法中,时延的度量是仅仅是队列的长度,而并没有考虑后来链路带宽的增长;第二,距离矢量算法的收敛速度比较慢,即使是采用了类似于水平分割这样的技术,也需要耗费过多的时间用于记录信息。

链路状态路由算法的思想非常简单,它包括以下五个部分:

(1) 发现邻节点,并获取它们的地址;

(2) 测量到达每一个邻节点的时延或成本;

(3) 构造一个分组来通告它所知道的所有路由信息:

(4) 发送该分组到所有其他节点;

(5) 计算到所有其他节点的最短路径。

事实上,完整的拓扑结构和所有的时延都已经分发到网络中的每一个节点。随后,每个节点都可以用Dijkstra算法来求得到其他所有节点的最短路径。下面将详细地讨论上述五个步骤。

(1) 发现邻节点

当一个路由器启动以后,它的第一个任务就是要知道它的邻节点是谁。具体实现的方法是:该路由器在每一个输出链路上广播一个特殊的Hello分组,在这些链路另一端的路由器将会发送回一个应答分组,告知它是谁。所有路由器的名字(地址)必须是全球惟一的。

当两个或多个路由器通过LAN互连时,如图5-14(a)所示,这时把LAN看成一个虚拟的节点N,如图5-14(b)所示。这时A到C的路由就可以看成是ANC。

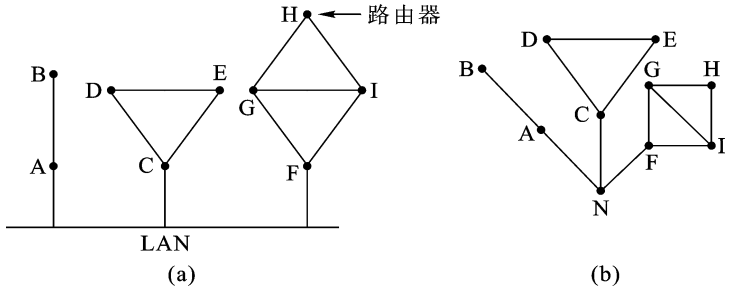


图5-14 节点通过LAN互连时的模型

(a) 通过LAN互连的网络 (b) 用虚拟节点N来等效的网络

(2) 测量链路时延或成本

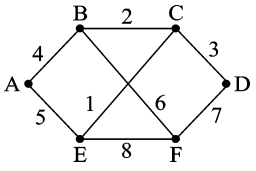
链路状态路由算法要求每个路由器确知到达每一个邻节点的时延或对该时

延有一个合理的估计。确定该时延的最直接的方法是发送一个特殊的ECHO分组给每个邻节点,并要求每个邻节点立即发回该分组。将测量的来回时延除以2就可以得到该链路时延的估计。为了得到较好的结果,可测量多次后取平均。

在测量链路时延时,既可以考虑排队的时延(链路的负荷),也可以不考虑排队的时延。考虑排队时延(链路负荷)的优点是可以获得较好的性能,但是可能会引起路由的振荡。

(3) 构造链路状态分组

每个节点都构造一个自己的链路状态分组,它包括发送节点的标号、该分组的序号和寿命,以及发送节点的邻节点列表及发送节点到这些邻节点的链路时延。链路状态分组如图5-15(b)所示。例如:A节点的链路状态分组中有两个邻节点B和E,A到它们的时延分别是4和5。图5-15(b)给出了每个节点的链路状态分组。



(a)

链路状态分组

A	
序号	
寿命	
B	4
E	5

B	
序号	
寿命	
A	4
C	2
F	6

C	
序号	
寿命	
B	2
D	3
E	1

D	
序号	
寿命	
C	3
F	7

E	
序号	
寿命	
A	5
C	1
F	8

F	
序号	
寿命	
B	6
D	7
E	8

(b)

图5-15 链路状态分组的格式

(a) 网络拓扑 (b) 链路状态分组的格式

构造链路状态分组是很容易的,困难的是何时构造这些分组。一种方法是周期性地构造这些分组;另一种方法是在链路状态变化(如故障、恢复工作或特性改变)时才构造这些分组。

(4) 分发链路状态分组

该算法的中最具技巧性的部分就是如何可靠地分发链路状态分组。当链路状态分组被发布后,首先得到该分组的路由器将改变其路由选择。同时,别的路由器可能还在使用不同的旧版本的链路信息,这样将导致各节点对当前网络拓扑的看法不一致性,从而计算出的路由可能出现死循环、不可达或其他问题。

链路状态分组分发的最基本方法是采用泛洪(flooding)方式。为防止每个

节点处理和中转过时的链路状态分组,在这些分组中引入了序号。每个节点仅中转序号大于已记录的最大序号的分组,为了防止序号出错,在分组中还引入了寿命,寿命每秒递减一次,如果寿命为0,则该分组将被丢弃。

为了提高传输的可靠性,所有链路状态分组都需要应答。为了处理链路状态分组在泛洪中需要发往哪些邻节点、需要对哪条链路的分组进行应答的问题,每个节点需构造一个如图5-16所示的分组存储数据结构。该图是图5-15拓扑中节点B的数据结构。图中每一行对应刚刚到达,但还没有完全处理的链路状态分组。由于节点B有三个邻节点A、C和F,所以发送标志指明应当发送给哪个邻节点,应答标志指明应答哪个邻节点。

源节点	序号	寿命	发送标志			ACK标志			数据
			A	C	F	A	C	F	
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

图5-16 链路状态分组的存储结构

在图5-16中,A的链路状态分组是直接到达B的,所以B必须发给C和F,并应答A,如图中的标志位所示。同样,来自F的链路状态分组必须发给A和C,并应答F。然而,当第三个来自节点E的分组到达时,情况则不同。由于分组已到达两次,一次是通过EAB,另一次是通过EFB,所以仅需发给C,但需同时应答A和F。最后一种情况是:如果来自节点C的链路状态分组仍在内存中(如图5-16的第四行所示),还没有中转,此时由C发出的链路状态分组的一个拷贝从F节点到达,这时需将这两个相同的分组合并处理,即将6个标志变成100011,表示仅需要发给A(而不再需要发给F),但要应答C和F。

(5) 计算新的路由

当每个节点获得所有的链路状态分组以后,它可以构造一个完整的网络拓扑,此时每个节点就可以运行Dijkstra算法来构造到达所有目的的节点的最短路由。

链路状态路由算法已广泛用于多种实际网络中,例如Internet中的OSPF采用了该算法,ISO的无连接网络层协议(CLNP)使用的IS-IS(Intermediate System-to-Intermediate System)协议也是采用该算法。IS-IS中交换的信息是用于计算最短路由的网络拓扑图(而不仅仅是链路状态分组),它还可以支持多种网络协议(如IP,IPX,AppleTalk等)。

5.4 自适应最短路由的稳定性分析

正如前面讨论的那样,路由算法可以根据不同的分类原则划分为不同的种类。但是总的来说,可以将其分为确定型策略(Deterministic Algorithm)和自适应型策略(Adaptive Algorithm)。确定型策略是基于网络拓扑和平均分组时延要求,以某一固定的准则来选择分组的路径,结果一般不受业务和拓扑变化的影响。这类策略包括:扩散式路由算法、随机式路由算法、固定式最佳路由算法等。而适应型策略是基于某个在时间上不固定的准则来选择在某一段时间内有效的路径,其结果尽量地适应业务和拓扑的变化。这类策略又可细分为:集中式自适应路由、孤立式自适应路由算法和分布式自适应路由算法。

适应型路由策略,不管是集中式、分布式或是孤立式,都是基于某一性能准则(例如最小跳数、最小平均报文时延等),根据网络当前运行中不断变化着的实际参数动态的决定各节点的路由选择方案。每个分组的传输路径是在传输期间动态的确定,以便及时自运适应当前的网络业务或其他条件(例如节点或链路故障等)的变化。对于这种适应型的路由算法,关心的一个问题是:如果在一段时间之后,算法最终是否可以有一个稳定的解,会不会振荡?这就是要讨论的算法稳定性问题。

下面针对以链路长度最短为路由选择原则的自适应路由算法来分析其稳定性问题。这里,讨论一种典型的情况,即链路的长度随着该链路的业务情况变化而变化。很显然,业务量比较重的链路将会被指定较大的链路长度。

下面通过实例来看看数据报网络的稳定性问题。

5.4.1 数据报网络的稳定性

假定有一个网络如图5-17所示,它由16个节点组成,其中节点16是惟一一个目的节点。

网络中的每一个节点都可能有两条路径到达目的节点:一条是顺时针方向到达目的节点,另一条是逆时针方向到达目的节点。设链路(i,j)的长度 d_{ij} 与链路上的流量 F_{ij} 成正比,这里取 $d_{ij}=F_{ij}$,采用自适应最短路由准则。

设节点1~7、9~15输入的业务量为1个单位,节点8输入的业务量为 $\varepsilon>0$(ε 是一个非常小的正数),网络每隔 T 秒更新一

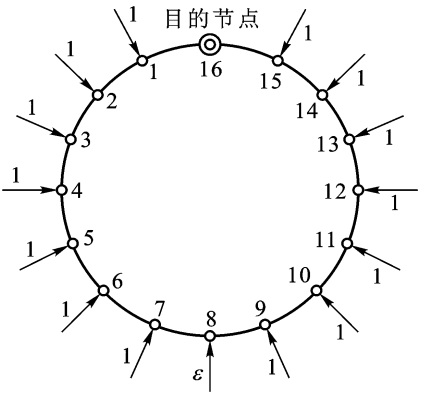
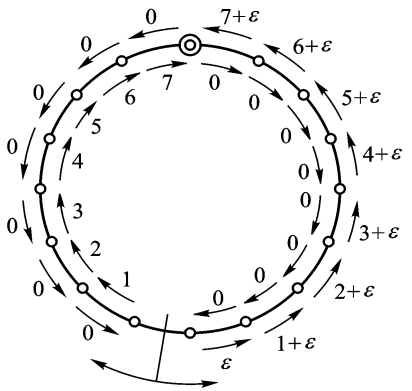
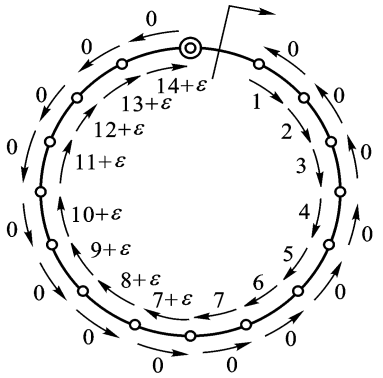


图5-17 拓扑图

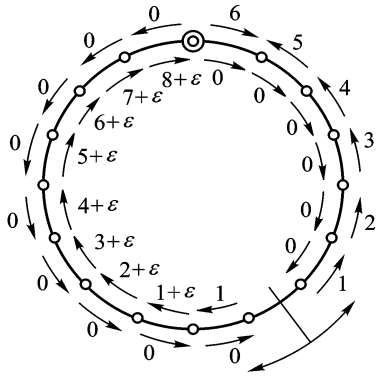
次最短路由，即每个节点用前一个 T 内的 F_{ji} 决定当前 T 秒的路由。假定第一次路由更新（t=0）的结果是 1～7 节点顺时针方向到达目的节点 16，8～15 节点逆时针方向到达目的节点 16 如图 5－18 所示。



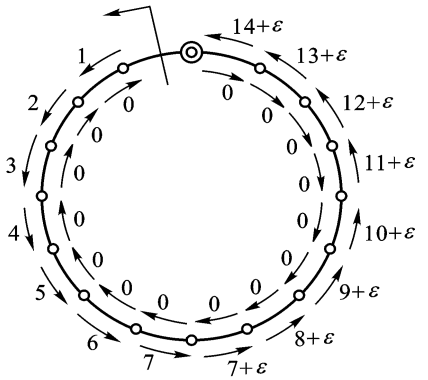
第1次路由



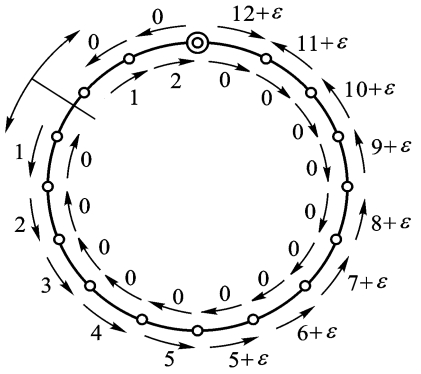
第4次路由



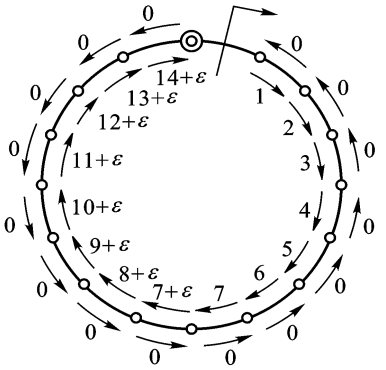
第2次路由



第5次路由



第3次路由



第6次路由

图 5－18　路由更新的结果

在 [0, T] 内节点 8 顺时针的路径长度为 28，逆时针的路径长度为 28+28ε（现在的路由），则 t= *T 时（第二次更新）根据最短路由准则，节点* 8 要改变路由。为了表示方便，令 P_n^L 表示节点 n 的顺时针的路径长度，P_n^R 表示节点 n 的逆时针路径的长度。在 [0, T] 内，P_9^L=28< P_9^R=28+7ε *（现在的路由），则 t= T* 时（第二次更新），节点 9 要改变路由。其他节点的路由不变。在 [T，2 *T] 区间* 内，P_1^L～ P_9^L 分别为 8+ ε、15+2ε、21+3ε、……、37+8ε，P_{10}^L～ P_{15}^L均为 37+8ε，P_1^R～ P_{10}^R均为 21，P_{11}^R～ P_{16}^R分别为 20、19、……、6。由于有 P_1^L< P_1^R，P_2^L< P_2^R，因此在 t=2 *T 时（第三次更新），节点* 1 和 2 的路由不变。由于从节点 n=3～15，均有 P_n^L> P_n^R，所以在 t=2 *T 时节点 n*=3～15 均采用逆时针方向路由。由于节点 10～15 原来就是逆时针方向路由，故仅有节点 3～9 改变了路由方向。在 [2 *T*，3 *T] 区间内，* P_1^L = 2；P_n^L = 2，n = 2，…，15；P_1^R = R_2^R = P_3^R = 83 + 8ε，P_4^R～ P_{15}^R分别为 82+8ε、…、12+ ε。*由于* P_k^R 均小于 P_k^L，k=1，…，15，故所有节点在 t=3 *T（第四次路由更新）时，均选择顺时针路由。*

在 [3 *T*，4 *T] 区间内，逆时针的路径长度均为* 0，故在 t=4 *T（第五次路由更* 新）时，全部选择逆时针方向。

在 [4 *T*，5 *T] 区间内，顺时针的路径长度均为* 0，故在 t=5 *T（第六次路由更* 新）时，全部选择顺时针方向。

以此类推，路由不断地振荡。

出现上述振荡的原因是各链路的长度（流量）取决于路由的选择，而路由的选择又取决于各链路的流量，这样就构成了反馈效应。在一般情况下，只要链路长度 d_{ij}是随着链路流量 F_{ij}的增加而单调增长，且当 F_{ij}=0 时有 d_{ij}=0，都会出现上述不稳定的振荡现象。

为了消除或减缓上述振荡，一种方法是给每条链路的长度都增加一个正的常量 α（α称为偏置因子）。当 F_{ij}=0 时有 d_{ij}= α>0。此时 α的选择就对路由的稳定性起着决定性的作用。如果 α相对于可能的链路长度足够大，则基本上是一个静态路由，且上述自适应最短路由就变成了一个最小跳数的路由。此时，网络振荡的可能性很小，但网络对链路的阻塞将不敏感。另一种方法是引入一种机制，将链路长度在多个最短路由更新周期上平均，这样将改进路由算法的稳定性，但也降低了网络对链路阻塞的敏感程度。

5.5 路由信息的广播

从前面讨论的讨论可以看出，路由信息的交换是路由算法的基础。特别是在分布式路由算法中，无论是距离矢量算法还是链路状态算法，都需要各个节点

相互交换路由信息,然后各节点再独立地计算各自的路由表。

在正常情况下,将路由信息从收集该信息的节点发送到需要该信息的节点是一个比较容易的过程。但是,当网络出现故障(例如:链路故障等)时,实施起来就非常困难了。这些困难主要体现在以下几个方面:

(1) 网络的拓扑信息必须通过链路来传输,而链路本身可能有故障。在有中心的网络中,这个问题尤为严重。因为,当网络的部分节点通过一条可能会故障的链路与网络中心相连时,如果该链路发生了故障,则网络中的部分节点将会与网络中心失去联系,从而导致这些节点无法获知网络的拓扑信息。

(2) 网络中会存在多次拓扑变化(如链路断开后,又在短时间内恢复正常),这就意味着必须对旧的路由信息和新的路由信息加以区分。

如果网络采用泛洪的方式来广播拓扑变化信息,当网络仅有一次拓扑变化时,泛洪方式可以正常工作,但当网络有多次拓扑变化时,泛洪的方法将不能正常工作。

假定有一个网络如图5-19所示,链路D→C开始处于连通(UP)状态,然后断开(DOWN),最后再恢复到正常连通状态。这样节点C就要产生两条拓扑修正信息,第一条信息(M_1)指明D→C处于DOWN状态,第二条信息(M_2)指明D→C恢复到UP状态。再假定这两条信息在C→B→A的链路上的传输时间小于在C→A链路上的传输时间。设M_1和M_2经过这两条链路到达节点A的时间顺序是:M_1首先经过C→B→A第一次到达A,接着,M_2经过C→B→A第一次到达A;然后是,M_1通过C→A第二次到达A。若此时C→A链路故障,则M_2将不能通过C→A第二次到达A。这样节点A就会认为链路D→C的状态是处于DOWN状态。而实际上的链路D→C的状态是处于UP状态。这里的难处在于,把过期的信息误认为是新的信息。

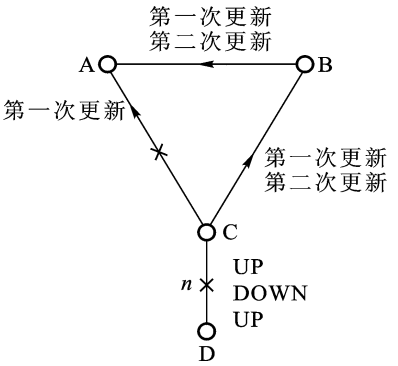


图5-19 一个网络的拓扑图

(3) 当正在执行最短路由算法时,新的拓扑信息到达,这时,要么算法能够处理这一变化,要么终止刚才进行的计算,开始重新计算。

(4) 当一条链路修复之后,它可能会引起分离的网络被重新连接,这时每一个分离的部分都可能会包括有关另一部分的过时拓扑信息。路由算法必须能够保证这两部分最终是一致的,并能够适应正确的网络拓扑。

由前面的讨论可知,网络中的每一个节点要在所有的时间内都得到正确的网络拓扑是不可能的。所以期望的最佳值是:算法能够在有限的时间内处理任意有限次的拓扑变化。

为了便于讨论作出如下假设,这些假设可以保证拓扑变化能够被正确的检测到。

⑴ 网络的链路能维持传输的顺序性和正确性。此外,各节点能够维持其存储器中数据的完整性。

⑵ 链路的故障是由链路的两个端点检测的,但不一定要同时检测到。这就意味着链路的一个端点认为链路断开后,另一个端点最终也会认为链路断开。必须注意的是,另一个端点认为链路断开必须在第一个点认为链路恢复之前。

⑶ 有完善的数据链路层协议,能够识别链路何时再次工作。如果链路的一个端点宣布链路恢复(UP),则在有限的时间内,另一个端点要么宣布链路恢复,要么首先再次宣布链路故障(DOWN)。

⑷ 节点可以有故障。这种情况下,与它关联的每一条链路的另一个端点在有限的时间内必须宣布链路故障(DOWN)。

下面以ARPANET网络中的泛洪广播为例,详细讨论一下在泛洪广播时需要注意的问题。

5.5.1 ARPANET的泛洪算法

泛洪算法又称为扩散式算法,其基本思想是:每个节点通过向其所有的邻节点发送消息的方式,将拓扑更新消息广播给所有的节点。每个相邻节点收到该信息后,再将其转发给它所有的邻节点,依此类推。

ARPANET泛洪算法有两个主要的特点:⑴能有效的防止拓扑更新信息在网络中无限次的循环;⑵能有效的防止序号出错带来的影响。

假定有一个网络如图5-20所示,如果网络采用泛洪的方式来广播拓扑更新消息,那么每个节点收到一条拓扑更新信息后,都将会中转给它的所有邻节点(发送该消息的节点除外)。这样链路(1,2) DOWN的消息到达节点3后,将在3→4→5→3之间无限次的循环。因此,这种简单的泛洪算法只能在没有环的网络中正常工作。

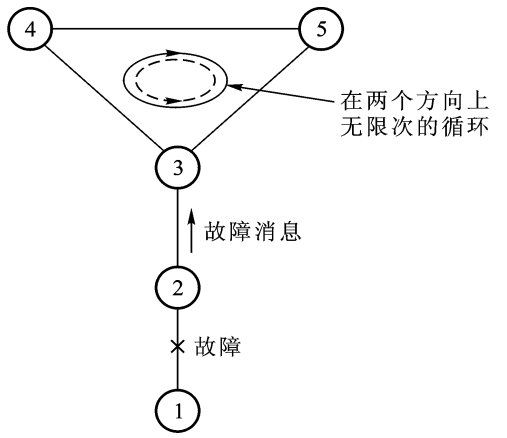


图5-20 泛洪广播举例

解决上述问题的方法是:在拓扑更新消息中,携带足够多的附加信息,使得每个节点仅发送该信息有限次,最好是一次。

在ARPANET中,每一消息都标定一个序号,节点*j*收到*i*的一个消息后,首先要比较该消息的序号是否大于*j*

从 i 接收到的最后一个序号。如果大于,则转发该消息,否则丢弃该消息。每条消息中序号域应足够大,使得在正常情况下,序号不会从高增加到最大值然后又重新置零。如果采用 48 bit的序号域,即使每 ms 更新一次,也需要 500 多年才能再次置零。

序号的使用可以保证每个节点仅发送一次同样的拓扑更新信息。这同时也解决了如何区分新旧消息的问题。但是,一旦在传输的过程中序号出错,将会导致很多困难。

序号出错来自两个方面:一是由于节点故障或 CRC 漏检将序号改变了。这样可能会出现两种意外情况:第一种情况是某节点内部意外地将序号置 0,则该节点的更新消息将被网络忽略。第二种情况是节点 *j* 意外地将节点 *i* 的序号从一个较小的值变成了较大的值,并广播到全网。这个错误的序号将被存储在各节点中,这将会导致其他节点不可能会听到 *i* 的信息。二是网络部分节点被复位或网络被分离后又重新连通,这会导致网络中仅靠序号无法识别正确的拓扑信息。

ARPANET 采用两种机制解决这一问题。

(1) 每个更新消息包括一个年龄域 (Age Field),它表明该消息已在网络中传播了多长时间。当一条消息到达某个节点后,该节点记录它的到达时间,并根据传播时间和传播时延,增加它的年龄域。一个节点可在任何时间计算其内存中所有消息的年龄。当该消息的年龄超过门限后,该消息将被丢弃,不再转发。

年龄域使用规则是:不论序号如何,过时消息将被未过时的消息取代,而未过时的消息只有在有一个更大序号的新消息时才被替代。这个规则可以保证被破坏或不正确的具有较高序号的消息不会被相信得太久。

(2) 每个节点除了在检测到链路状态变化时发送拓扑更新消息外,每个节点要周期性地发送更新消息 (至少每 60 s 发送一次)。周期性广播保证在网络两个分离的部分再连通以后,使最新的拓扑更新消息能在某个固定的时间内得到。周期性广播的缺点是网络开销较大。

小　结

本章主要解决在网络中任意两个节点之间如何选取最好的传输路径问题。一个网络的设计者面临的问题包括:采用什么策略来选择合适的路由?依据什么信息来进行这种选择?应该如何执行这种选择的策略?用什么标准来评判选择路径的好坏等问题。

本章首先讨论了路由算法的作用和各种常用的路由算法,包括广域网中的路由算法、互连网中的路由算法以及 Ad Hoc 网络的路由算法。然后,讨论了最

短路由的定义,并且详细介绍了三种集中式的最短路由算法(B－F算法、Dijkstra算法和F－W算法)。在此基础上讨论了分布式最短路由算法(距离矢量算法和链路状态法)。接着就分布式路由算法的稳定性问题进行了详细的讨论。最后介绍了如何实现路由信息的广播问题。

习 题

5.1 一个理想的路由算法应具有哪些特点?为什么实际的路由算法总是不如理想的?

5.2 路由算法有哪些类型?所谓"确定型"和"自适应型"的分类,是在什么意义上而言的。

5.3 试述广域网的路由与互连网的路由的区别和联系?

5.4 分别使用Bellman－Ford和Dijkstra算法求解图5－21中从每一个节点到达节点1的最短路由。

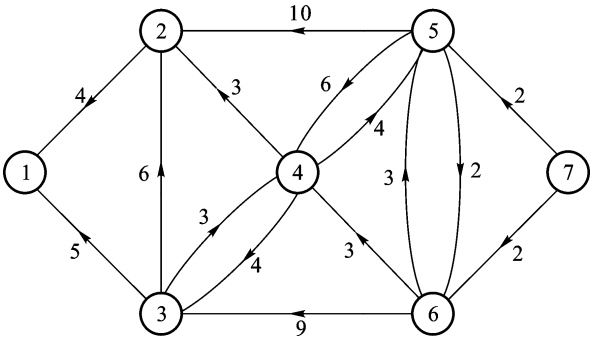


图5－21 习题5.4图

5.5 在距离矢量法中为什么会出现"计数至无穷"的现象?如何解决?

5.6 链路状态法的基本步骤是什么?它与距离矢量法相比有何优点?

5.7 一个广域网有50个节点,每个节点和其他3个节点相连。若采用距离矢量算法,每秒钟交换路由信息2次,而节点间的时延用8 bit编码。试问:为了实现分布式路由算法,每条链路(全双工)需要多少带宽?

5.8 假定在图5－17中节点数改为6个,即节点1,2,4,5发送一个单位到节点6,而节点3发送ε给节点6,并且有$0<\varepsilon\ll 1$。⑴ 试画出路由更新的结果。⑵ 若$d_{ij}=\alpha+F_{ij}$,$\alpha=1$,试画出路由更新的结果,并考虑各种可能的初始路由选择。⑶ 无论初始条件如何选择,除节点3以外,所有节点的最短路径最终都保持不变的α的最小值是多少?⑷ 假定在第一次迭代以后一条链路的长度是当前链路流量和以前路由流量的平均,试重复⑴。

第6章 流量和拥塞控制

6.1 流量和拥塞控制概论

在计算机网络中,链路的容量、交换节点中的缓冲区和处理机等都是网络的资源。在某段时间内,若对网络某一资源的需求超过了该资源所能提供的可用部分,网络的性能就会恶化。同时,如果信息流无限制的进入网络也会导致网络的性能恶化。当报文在网络中经历了比所期望的时延更长的时间时,就认为网络产生了拥塞。当网络发生拥塞时,只能有很少的信息流动,而且拥塞会很快的延伸,甚至导致"死锁"。发生死锁时,网络中几乎没有分组能够传送。因此,如果对网络的拥塞不加控制,网络的性能将会迅速下降。

造成网络出现拥塞有若干种原因。例如:当从多个输入端到达同一节点的分组要求同一条输出链路时,就形成分组的排队。如果该节点的缓冲器容量不够,就会造成分组丢失。在某种程度上增加缓冲器的容量可以减少这种由于缺少缓冲容量而造成的分组丢失率;同时,如果处理器的处理速度太低,或者链路带宽不够也会导致拥塞;另外,拥塞节点本身也会使拥塞变得更严重。因为如果节点没有空闲的缓冲区,那么新到达的分组必定被丢弃,而发送端会因为等待应答超时而重发,甚至重发若干次,这必然会增加网络的负荷,使整个网络中节点的缓冲器逐渐地都将处于饱和。随着网络负荷的激增,吞吐率骤降,最后达到零值,导致网络进入死锁状态。

网络不仅在拥挤严重时会发生死锁,而且在一定条件下,在轻负荷时也会发生死锁。图6-1给出了三种死锁产生的原因。图6-1(a)是一种直接存储转发死锁现象,图中节点A中的缓冲池被欲发往节点B的分组占满,彼此都期待对方能接收本端的分组而腾出缓冲空间来,但双方都无法做到,因而处于对峙和僵持状态。图6-1(b)是一个间接存储转发死锁现象,图中一个闭环上的各个节点的相关链路缓冲池都被占满,任意一个节点都无法腾出空闲存储空间来接收邻节点的报文分组,因而处于无法解脱的僵持局面。图6-1(c)是重装死锁(所谓重装死锁是指节点无法重传报文而引起的死锁)的现象,图中目的节点无法重装报文而出现死锁现象。图中A、B、C三个报文的部分分组已占满了目的节点的缓冲池,但都没有完成重装工作。因为报文A的A_6和A_7两个分组仍分别在节点N_2和N_3中排队等待发送;报文B的B_4还在N_2中排队,报文C

的 C_4 还在 N_1 中排队。由于三个报文都短缺一些分组而无法重装，也就无法腾出空间来接收剩余的分组数据，形成僵局，出现死锁现象——“报文重装死锁”。死锁的直接恶果就是使整个网络（或网络的局部地区）瘫痪，吞吐率降至0。

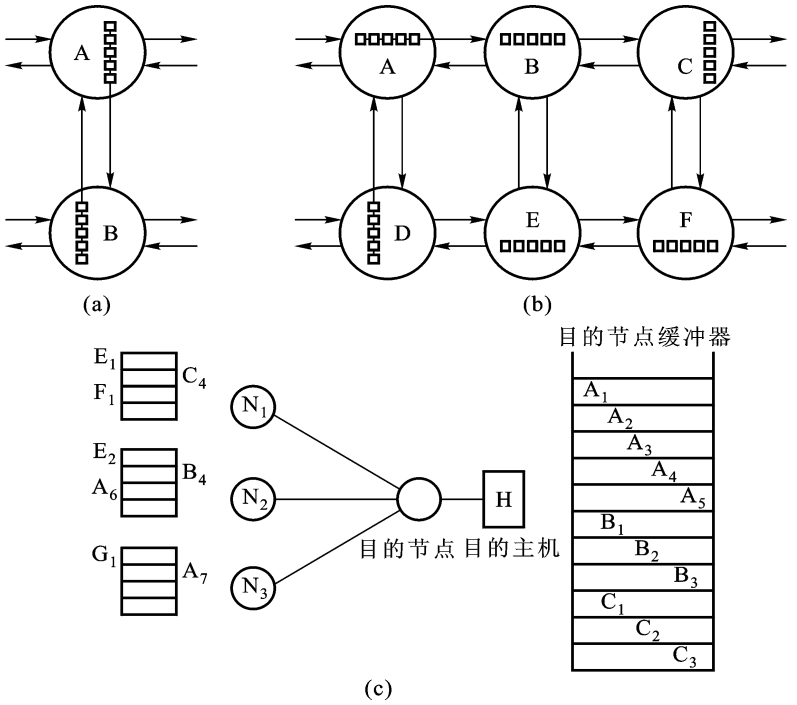


图6－1 死锁现象

(a) 直接存储转发死锁 (b) 间接存储转发死锁 (c) 重装死锁

6.1.1 网络数据流的控制技术分类

网络数据流的控制技术可以分为三类，即流量控制、拥塞控制和死锁防止。它们有不同的目的和实施对象，而且各自在不同的范围和层次上实现。

(1) 流量控制

流量控制是对网络上的两个节点之间的数据流量施加限制。它的主要目的是让接收端来控制链路上的平均数据速率，以满足接收端本身的承受能力，以免过载。

(2) 拥塞控制

拥塞控制的目的是将网络内（或网络的部分区域内）的报文分组数目保持在某一量值之下，超过这一量值，分组的平均排队时延将急剧增加。因为一个分组交换网络实质上是一个排队的网络，每个节点的输出链路端口都配置一个排队队列，如果分组到达的速度超过或等于分组发送的速度，队列就会无限制地增

长,致使分组平均传输时延趋于无穷大。如果进入网络的分组数目继续增加,节点缓冲器将会占满溢出,丢失一些分组。丢失分组的后果是导致发端重发。重发实际上又增加了网络内流通的业务量,最终可能使所有节点缓冲器都被占满,所有的通路完全被阻塞,系统的吞吐率趋于0。

避免这种灾难性事件发生就是拥塞控制的任务。所有拥塞控制技术的目的,都是为了限制节点中的队列长度以避免网络过载。这类控制技术不可避免地要引进一些控制信息开销,因而实际效果不如理论上那么理想。

(3) 死锁防止

网络拥塞到一定程度时,就要发生死锁现象。死锁发生的条件就是在构成一个封闭环路上的所有节点的相关链路缓冲器都被积压的报文分组所占满而失去了节点所担负的存储转发能力。即使在网络轻负荷情况下,也可能出现死锁的现象。死锁防止技术旨在合理地设计网络使之免于发生死锁现象。

拥塞控制和流量控制的概念经常被混淆,实际上两者是有差异的。拥塞控制必须使得通信子网能够传送所有待传送的数据,它是一个全局性的问题,涉及到所有的主机、路由器、路由器中存储转发处理的行为,以及所有将导致削弱通信子网能力的其他因素。而流量控制只与某发送者和某接收者之间的点到点的业务量有关。它的任务是确保一个快速发送者不能以比接收者能承受的速率更高的速度传输数据。流量控制几乎总是涉及到接收者,接收者要向发送者送回一些直接反馈。简单地说,流量控制是防止网络拥塞的一种机制。

当一个节点向某特定的接收节点发送的报文超过了接收节点处理或转发报文的能力时,会发生拥塞。因此,问题就简化为对给定节点提供一种控制从其他节点接收报文多少的机制。这样,流控就是使发送节点产生报文的速率受到控制,从而使接收节点能及时处理发端报文的一个过程。从用户的观点看,流控就是要防止进入网络的报文不能在预先规定的时间内传递的问题。例如:有一个网络的传输容量为1000 *Gb/s,假定一个超级计算机利用该网络以1 Gb/s的速*率向一个PC发送一个文件。尽管网络不会有拥塞问题,但需要流控,以不断地暂停超级计算机的传输,从而使PC可以完成相应处理。又例如:一个分组网络的各链路的速率为1 *Mb/s,有*1000个大型计算机连入该网络,其中一半的计算机要向另一半的计算机发送以100 *kb/s的速率传输文件,这里没有高速发送节*点使接收端溢出的问题(即没有流控问题),但存在着总的呈现给网络的业务量大于网络能处理的业务量问题,因而需要采用拥塞控制。

流量和拥塞控制在网络中所起的作用可用图6-2描述。从图中可以看到,网络传送的分组速率是输入负载或递交给网络的分组速率的函数。理想的情况下:只要输入负载低于网络的容量,网络应传送全部已递交的分组。当输入负载高于网络容量时,网络(仍是理想情况)应继续以最大容量传送分组。图中标有

“理想情况”的曲线表示这一理想情况。然而,在实际的网络中,如果网络没有流控,只有当网络输入负载低于某一定值时,(与理想情况相比)网络才能传送全部输入负载。当输入负载的增长超过这一定值时,网络的实际吞吐量与理想曲线开始分离(尽管实际吞吐量的变化仍是输入负载的函数)。随着输入负载的进一步提高,无流控网络的吞吐量开始下降。输入网络的业务量越高,实际传递的业务量越低。在某种情况下,足够高的输入负载会导致死锁,即网络中没有或几乎没有成功的传送分组。

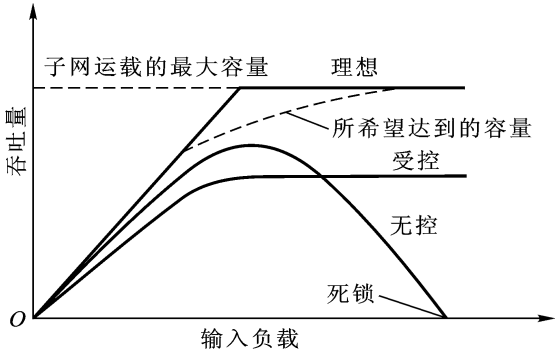


图 6-2 流量和拥塞控制的作用

为了对流量和拥塞控制的作用以及在无流量及拥塞控制时对网络存在的问题(主要体现在吞吐量和公平性方面)有一个初步理解,以图 6-3 所示的网络为例加以说明。在该例中,分组的传输规则是:如果分组到达节点时没有可用的缓冲器,分组将被丢弃。为了恢复丢弃的分组,节点或主机在规定的时间内没有收到应答信号则重发该分组。(节点或主机将保留分组副本,直至给分组被接收者所确认。)

例 6.1 在图6-3所示的网络中,链路上的数字分别代表其通信容量,单位为 kb/s。设网络的业务需求如下:主机 B 至主机 A 的业务需求量为 λ_{BA} kb/s,主机 C 至主机 D 的业务需求量 λ_{CD} kb/s。*B 到 A 的路径是 B→ Y→ X→ A,C 到 D* 的路径是 C→ *Z*→ *X*→ *D*。*试分别讨论以*下几种传输方案下网络的状态:(1) λ_{BA} = 7 kb/s 且 λ_{CD} = 0;(2) λ_{BA} = (8+δ) kb/s (δ>0),λ_{CD} = 0;(3) λ_{BA} = 7 kb/s 且 λ_{CD} = 7 kb/s;(4) λ_{BA} = (8+δ) kb/s (δ>0),λ_{CD} = 7 kb/s。

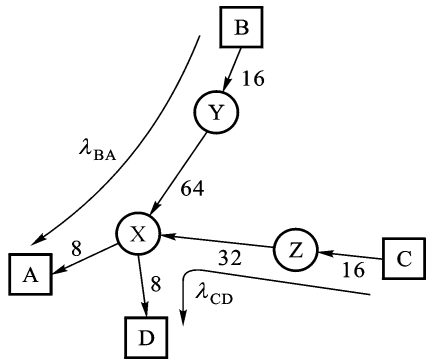


图 6-3 拥塞网络举例

方案 (1):此时 B 到 A 的业务请求能够在现有网络容量下得到解决,不会出现

拥塞情况。这里分组发往主机 A 的速率与主机 B 发送的速率相同。链路 B→Y, Y→X, X→A 每一段的速率均为 7 kb/s。

方案 (2):这时,提交网络的分组速率高于 X→A 链路能够处理的速率。因此,在某一时刻,X 节点的缓冲区满。这将导致从节点 Y 发出的分组将被丢弃而不会得到确认。由于 Y 节点保留未确认分组以便重发,最后 Y 节点缓冲区满。这样会造成另一个很有意思的现象:由于节点 X 能够传送 8 kb/s,而最初要求提供 (8+δ) kb/s,因此,开始会拒绝发送 δ kb/s。此时,为重发丢失的 δ kb/s, Y→X 链路将传送 (8+2δ) kb/s,但 X 节点只能发送 8 kb/s,所以被丢弃 2δ kb/s,丢弃的 2δ kb/s 仍需重发,因此 Y→X 链路将传送 (8+3δ) kb/s。因为重复发送, Y→X 链路上的业务量不断增加直至总量达到 64 kb/s。同样的原因,B→Y 链路的业务将达到 16 kb/s,其中包括新分组和重发分组。由此可见,若要求网络以高于其容量的速率传送分组,这种过高的要求会大量消耗网络资源。

解决方案 (2) 拥塞问题可以有下列两种方法:

① 网络有足够大的容量, X→A 的链路能适应 B 节点最大可能的业务量。

② 限制 B 节点的最大业务量,使其不超过 8 kb/s。

若已知最大的业务需求量,可采用方法①。但是,方法①只有在 B 频繁要求最大业务量且持续较长时间时才有经济意义。如果在大部分时间,B 至 A 的业务需求量很低 (如 2 kb/s),只有偶然的峰值超过 8 kb/s,则应该限制 B 的瞬时最高流速为 8 kb/s,任何高于 8 kb/s 的业务将延迟直至脱离过载状态。这两种方法的本质区别在于:第一种方法是一种设计思路,不能实时实现;第二种方法是用于网络控制的策略,网络可以实时的根据业务需求,实施该策略。

方案 (3):与方案 (2) 相同,这时不会出现拥塞状态。发往 A 和 D 的数据总的速率为 14 kb/s,每个方向的数据速率为 7 kb/s,每条网络链路承担 7 kb/s。

方案 (4):在本方案中,C 至 D 的路径有足够的容量,可以满足业务需求。存在的问题是:在无控的网络中,B 至 A 与 C 至 D 的分组需共享 X 节点的缓冲区容量。从方案 (2) 可知,B 至 A 的业务请求会导致 X 节点缓冲区满。反过来,缓冲区使主机 C 和主机 B 发出的分组到达 X 节点后被频繁丢弃。虽然,事实情况是主机 B 的业务引发这一问题,但是,所有发往 X 节点的分组都会被丢弃。根据方案 (2),X 节点的缓冲区满,最后,使 Y 节点和 Z 节点也发生缓冲区满,各链路以各自的容量传送业务。

无论何时,只有 X 节点发送分组至 A 或 D,节点 A 或 D 都要接收并确认此输入分组。因为 X 从 Y 接收分组的速率是从 Z 接收速率的两倍 (Y→X 的容量是 Z→X 容量的两倍),节点 X 发往 A 的分组速率似乎是发往 D 分组速率的两倍。因此,X 节点缓冲区中,至 A 分组与至 D 分组的比率为 2:1。至 A 分组

以X→A链路的最大速率(8 kb/s)传送,因而,至D的传送速率是此速率的一半,即4 kb/s。

对比方案(3)和方案(4),可以发现当λ_{BA}从7 kb/s提高到(8+δ) kb/s时,会出现:

① 总吞吐量降低。网络传送的总业务量从14 kb/s降至12 kb/s。

② 对主机C的业务量待遇不公平。由C发往D的业务速率从7 kb/s降至4 kb/s。这样,虽然是由于主机B的业务引起的问题,但主机C的损失却超过了主机B。

解决方案(4)的拥塞问题类似于解决方案(2)的拥塞问题。即在方案(2)中讨论的两种方法仍然适用。另外,可以采用第三种方法:在节点X处,为节点D的业务保留一定数量的缓冲区。这样,无论节点B是否过载,都能够保证来自节点C的分组具有进入X节点缓冲区的入口,这样,使分组得到公平的待遇。当然,保留资源与分组交换的首要目的(理想的资源共享)相矛盾。看来,牺牲一部分资源共享的利益,是保证网络公平合理的代价。因此可以看出缓冲区的管理是非常重要的,缓冲区满可以引起整个网络的瘫痪。通过例6.2可以看出缓冲区管理的重要性。

下面再通过一个例子讨论一下公平性的问题。

例6.2 图6-4所示的网络有n条链路,n+1个用户,其中n个用户各自要求使用一条链路,另一个用户要求使用全部n条链路。每一用户要求的数据率为1单位/秒,每条链路的容量为1单位/秒。若要求使用n条链路的这个用户被完全禁止,则其余用户都可以被接纳,得到全网总的吞吐量为n个单位。但若同样地对待所有用户,则每一个用户可得到的最大吞吐量为$\frac{1}{2}$单位,系统总的吞吐量为$\frac{n+1}{2}$。如果n较大,则相当于只能到达前面总吞吐量的一半。即此时为了满足系统的公平性,是以降低系统的总吞吐量为代价的。

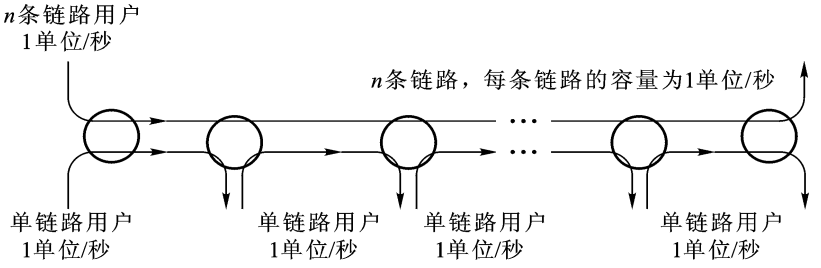


图6-4 公平性举例

通过前面的讨论,可以将流量和拥塞控制的功能概括为以下四个方面:

① 防止由于网络和用户过载而产生的吞吐量降低及响应时间增长；
② 避免死锁；
③ 在用户之间合理分配资源；
④ 网络及其用户之间的速率匹配。

6.1.2 拥塞控制的基本原理

拥塞控制的基本原理是：寻找输入业务对网络资源的要求小于网络可用资源成立的条件。这可以通过增大网络的某些可用资源（如输入业务繁忙时增加一些链路，增大链路的带宽，或使额外的通信量从另外的通路分流），或者通过减少一些用户对某些资源的需求（如拒绝接受新的建立连接的请求，或者要求用户减轻其负荷，这属于降低服务质量）。

拥塞控制是一个动态控制的问题。从控制论的角度分类，可以分为两类：一类是开环控制，另一类是闭环控制。开环控制方法就是在设计网络时事先将有关发生拥塞的因素考虑周到，力求网络在工作时不产生拥塞。一旦整个系统运行起来，就不再中途进行更改了。

闭环控制是建立在反馈控制的概念之上，其控制过程有以下几种部分：

(1) 实时监测网络，以便监测到拥塞在何时、何处发生；
(2) 将拥塞发生的信息传送到可采取行动的地方（如控制中心）；
(3) 调整系统的操作过程，以便纠正拥塞问题。

有多种度量可用来监视子网的拥塞状态。其中主要的有：因缺少缓冲区空间而丢失分组的比例，平均队列长度，超时和重发分组的数量，平均分组时延等等。这些因素数值上的增加意味着拥塞可能性的增加。

一般在监测到拥塞发生时，要将拥塞发生的信息（控制分组）传送到产生分组的信源。当然，这些额外的控制分组要在子网中传输，即恰好在子网拥塞时又增加了子网的负荷。另一种方法是在路由器转发的分组中保留一位或一个字段，用该比特或字段的值表示网络的状态（拥塞或没有拥塞）。也可以由一些主机或路由器周期性的发送控制分组，以询问网络是否发生拥塞。

此外，过于频繁的采取行动以缓和网络的拥塞也会使系统产生不稳定的振荡。但过于迟缓的采取行动又不具有任何实用的价值。因此，要采用某种折衷的方法。

6.1.3 流控和拥塞控制所经历的层次

流量及拥塞控制可以出现在所有的协议层次上，不过主要还是在数据链路层、网络层和传送层：分段（逐跳）流控是数据链路层的功能，称为节点到节点之间的流控；而端到端流控主要在传送层，称为全局流控；拥塞控制则主要集中在

网络层。影响拥塞控制的一些主要策略如表6－1所示。

表6－1 影响拥塞的主要策略

层 次	策 略
传送层	• 重传的策略 • 乱序缓存的策略 • 应答的策略 • 流控的策略 • 定时的确定
网络层	• 在子网内采用虚电路还是数据报方式 • 分组排队和服务的策略 • 分组丢弃的策略 • 路由的算法 • 分组寿命管理
数据链路层	• 重传策略 • 乱序缓存的策略 • 应答的策略 • 流控策略

6.2 流量和拥塞控制技术

流量和拥塞控制技术按执行流控和拥塞的方式可分为:集中式和分布式流量拥塞控制。在集中式流控中,网络中有一个特定的网控节点执行某种算法,为各个节点计算报文流量的分配值,然后将新的流量分配值传送给所有其他网络节点。而分布式流控则是把管理网络的流量过程分配到若干个网络节点来完成,这些节点控制通过其自身或外部的业务流量。流量及拥塞控制有许多具体的实现方式,下面将着重讨论窗口式流量拥塞控制和漏斗式流量控制算法。

6.2.1 窗口式流量和拥塞控制

窗口式流控的思想类似于数据链路层的返回n－*ARQ,在一个 session 中,* 发端 A 在未得到收端B的应答情况下,最多可以发送 W(窗口大小)个消息或分组或字节。收端 B 收到后,回送给发端 A 一个 permit(它既可以是应答,也可以是分配消息),A 收到后方可发送新的数据。下面首先介绍一下在窗口式流控中应该注意的问题,然后再集中讨论具体的窗口式流控算法。

1. 滑动窗口控制机构的建立

通信子网中的任意节点对之间都可能构成源/目的节点对。若子网的节点数为 m,则一个节点可能与其他 m-1 个节点结合,最多形成 m-1 个源/目的节点对。如果在一个节点内为每一个源/目的节点对常设一个滑动窗口控制机构,将使节点控制机构变得相当复杂,占用许多的缓冲存储容量。减少这种复杂性的途径就是采用动态方法:在每个节点中只为有当前通信业务的源/目的节点对设置窗口控制机构,并相应地分配缓冲区。每一个源/目的节点对实际上就是一条虚拟的线路。可见,窗口控制机构应当随着每一条虚拟线路的建立而建立,这与为每一条数据链路建立一对滑动窗口控制机制是一样的。

2. 窗口宽度的确定

一个源节点可能与许多不同节点构成源/目的节点对。如果源/目的节点对之间的距离比较远,则相应的两个节点之间的端到端时延就比较长,从源节点发出一个分组到它收到应答期间连续发出的分组数会比较大。因此,为了有效地利用通信子网的传输能力,这时源/目的节点对之间的窗口宽度 W 就应当比较大。相反,源/目的节点对之间跳数比较少的节点对之间的窗口宽度 W 就应该相应的小一些。可见,窗口宽度应当根据源/目的节点对之间的距离来选择,而不能简单地选成一样大小。为此,可以在每一个源节点内设置一张说明窗口宽度和节点距离关系的对应表。根据这种对应关系,动态选择合适的窗口宽度,以便建立起相适应的窗口控制机制。

理想的窗口宽度应当这样选择:源节点从发送第一个分组到收到目的节点对该分组的确认时,源节点的窗口控制应该刚好发完窗口宽度允许的最后一个分组。这种情况下,源节点就能以最佳的速率不间断的发送分组。

3. 报文的重装

当目的节点全部收完一个报文的所有分组后,才能着手重装报文,然后再提交给目的主机。这是,该源目的节点对上的一个报文才算传输成功,目的节点将给源节点返回一个确认应答。可见,端到端流控功能中还包括在目的节点进行报文重装的功能,特别是当通信子网采用自适应型路由策略和采用网内数据报传输方式时报文重装的功能尤其重要。在这两种情况下,源节点发出的报文分组可能沿不同的路径到达目的节点,因而造成分组到达的次序与发送次序可能不一致。如果某一分组在中转过程中丢失,在目的节点将无法将报文重装出来。此时,目的节点应在返回的应答中报告这一情况。源节点在确知后,应立即重发被丢弃的那个分组。为此,源节点缓冲区必须保留全部未应答的数据分组,以便在需要重传时使用。

下面讨论具体的窗口式流控技术。

1. 端到端的窗口流控

为了便于讨论,做以下一些定义:

W:窗口大小,即流控窗口为 W 个分组,如果收端希望收到的分组序号为

k,则发端可以发送的分组序号为 k～k+W－1 的分组。

d:分组传输的来回时延,它包括来回的传播时延、处理时延、分组传输时间、应答分组的传输时延。

X:单个分组的传输时间。

图 6－5 描述了 d,X,W 之间的关系。

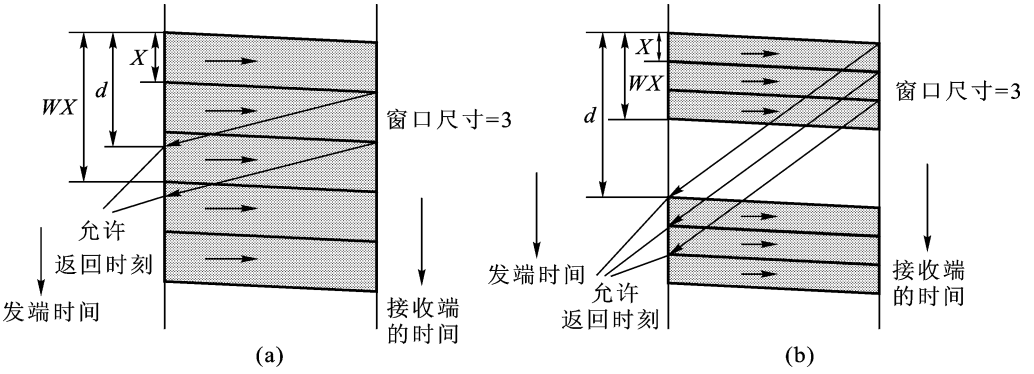


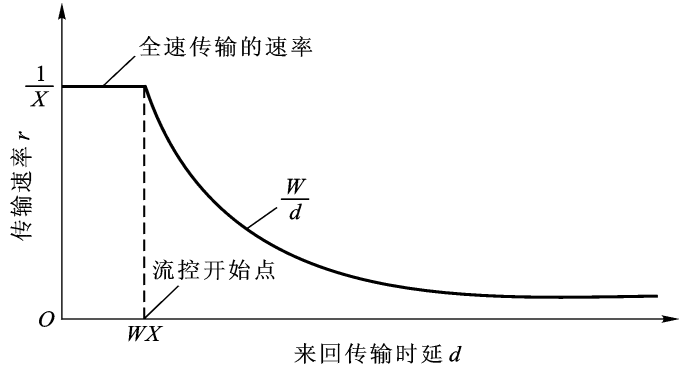
图 6－5 d,X,W 的相互关系

(a) d≤WX 的情况 (b) d>WX 的情况

在图 6－5(a)中,d≤ *WX,则发端可以以* $\frac{1}{X}$ (分组/秒)的速率全速发送,流控不会被激活。而在图 6－5(b)中,d> *WX,在 d 时间内,最多只能传送 W 个*分组,即分组传输的速度为 $\frac{W}{d}$ (分组/秒)。显然,对于给定来回时延 d,最大的分组传输速率 r 为

$$r=\min\left(\frac{1}{X},\frac{W}{d}\right) \tag{6-1}$$

其结果如图 6－6 所示。从图中可以看出,d 增加表明网络中拥塞增加,这样会导致分组传输速率 $\frac{W}{d}$ 下降。如果 W 较小,则拥塞控制反应较快,即在 W 个分组内就会作出反应。这种以很小的开销获得的快速反应是窗口流控策略的主要优势之一。



下面通过一个具体的例子讨论端到端流控的工作过程。

例 6.3 设有一系统如图 6-7 所示,采用了端到端窗口流控技术。在该系统中,传输的所有分组的长度均为 53 字节,信源 S 的最大输出速率为 80 Mb/s,信道的传输速率为 80 Mb/s。若目的节点 D 可接收信息的速率为 40 Mb/s、10 Mb/s 和 1 Mb/s 时,试讨论如何进行流控?(假定所有节点的处理时延均为 0,各节点都采用存储转发的机制。)

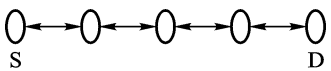


图 6-7 网络拓扑图

解 从图中可以看出,从源节点 S 到目的节点 D 之间经过三个节点,由于各个节点采用的是存储转发机制,因而从源节点发出一个分组到目的节点返回一个许可的来回时延为分组在各链路上传输时间之和,即

$$d=\frac{8\times 53\times 8\,\text{bit}}{80\ \text{Mb/s}}=42.4\ \mu\text{s}$$

由于目的节点可接收的信息速率为 40 Mb/s、10 Mb/s 和 1 Mb/s 均小于源节点的输出速率,所以应有 $WX<d$。所对应的目的节点的接收的信息速率为 $\frac{WX\cdot 80\ \text{Mb/s}}{d}$。$X=\frac{53\times 8\,\text{bit}}{80\ \text{Mb/s}}=5.3\ \mu\text{s}$。因此,当 $\frac{WX\cdot 80\ \text{Mb/s}}{d}=40\ \text{Mb/s}$,10 Mb/s 和 1 Mb/s,可分别求得 $W=4,1,0.1$。$W=0.1$ 意味着目的节点应当将允许分组延迟 $(9\times 42.4\ \mu\text{s})$ 以后发出。在具体实现时,采用 $W=1$,但要通过增加缓冲的方法使得 d 增加 10 倍,此时需要控制发端的速率,以避免造成缓冲区的溢出。

端到端窗口流控的主要问题有:

(1) 不能保证每一个 session 有一个最小的通信速率。

(2) 必须处理和确定窗口的大小。它必须在允许每个 session 的最大传输速率、传输时延、信道的最大传输能力、网络的拥塞等因素之间综合考虑。

(3) 无法提供对分组时延的合适控制。

假定网络中有 n 个活动的 session,各 session 的窗口分别为 $W_1, W_2, \cdots, W_n$,则在网络中流动的总分组数近似等于 $\sum_{i=1}^{n} W_i$,根据 Little 公式有分组的时延为:$T=\sum_{i=1}^{n}\frac{W_i}{\lambda}$。式中,$\lambda$ 是各 session 输入的总速率。随着 session 数的增加,通过量 λ 受链路容量的限制,将接近常量,这时就会有时延 T 将正比于 session 的数目或者说总窗口的大小。因此说窗口方法不能把时延维持在适当的水平上。

(4) 端到端窗口流控在公平性方面较差。

一个路径较长的 session,如果窗口较大,经过重负荷的链路时,等待的分组较多。而另一个路径较短的 session,如果窗口较小,经过重负荷的链路时,等待

的分组较少。这样就会导致长路径的 session 得到较大比例的服务。

2. 虚电路中逐跳窗口流控

虚电路中逐跳窗口流控 (node by node windows for virtual circuits) 是在虚电路经过的每个节点中保留 W 个分组的缓冲区。在该链路上的每一个节点都参与流控,每一条链路的窗口都为 W,每个接收分组的节点可以通过减缓回送允许 (应答) 分组给发送节点的方式来避免内存中积压太多的分组。在这种方式下,各个节点的窗口或缓冲区是相关的。假定虚电路是经 $(i-1) \rightarrow i \rightarrow (i+1)$ 等节点,当节点 i 缓冲区满时,只有当节点 i 向节点 i+1 发送一个分组后,i 才可能向 i−1 发送一个应答分组。这样就会导致 i 的上游节点 i−1 缓冲区满,依此类推,最后将导致源节点的缓冲区满。这样从拥塞节点缓冲区满返回到源节点缓冲区被填满的现象被称为反压现象 (Backpressure)。

3. 流控窗口的动态调整

为了能够适应网络的拥塞情况,可以动态调整窗口的大小。当发生拥塞时,自动减小窗口,以减缓拥塞的状态。实现的基本方法是:通过从拥塞点到源节点的反馈控制来实现。

方法一:当某节点感觉到拥塞时 (即发现缓冲区短缺或队长过长) 发送一个特殊分组给源节点。源节点收到后,减小其窗口。在一个适当的时间后,如果拥塞状态已缓解,源节点再逐步增大它的窗口。

方法二:通过收集正常分组从源节点到达目的节点的拥塞信息,目的节点利用这些信息,通过一些控制分组来调整窗口的大小。

6.2.2 漏斗式速率控制算法

拥塞发生的主要原因在于通信量往往是突发性的,如果主机能够以一个恒定的速率发送信息,拥塞将大大减少。除了前面讨论的窗口式流控方法之外,还有一种管理拥塞的方法,即强迫分组以某种有预见性的速率传送。这种管理拥塞的方法被称为业务整形 (traffic shaping)。

1. 业务整形

业务整形是调整数据传输的平均速率 (以及突发性)。与之相比,前面讨论的滑动窗口协议只是限制一次能传送数据的数量,而不是传送的速率。当在虚电路网络中应用业务整形算法时,在虚电路建立阶段,用户和子网之间共同协商一个关于该电路的业务流模型,只要用户按照协商的业务流模型发送分组,子网将确保按时传送这些分组。业务流模型的协商对文件传输不是很重要,但对实时数据 (如音频和视频传输) 是很重要的,这些实时业务不能容忍拥塞的出现。当子网同意某一业务流模型的用户接入后,子网要对该用户的业务流进行监视,以确保守法用户的传输,限制违约用户的传输。业务整形算法的思想同样适用

于数据报网络。

2. 漏斗算法

下面以生活中的一个例子来看流量控制。假设有一个漏斗，如图6-8(a)所示。不管注水的流量如何，只要漏斗中有水，漏斗将以恒定的速率向外流水。而且，当漏斗装满水后，如果还向其注水，将导致注入的水从漏斗中溢出。只有当漏斗为空时，输出的速率才为0。这种思想也可以应用到分组传输的过程中，如图6-8(b)。从概念上讲，每台主机都可以通过一个类似于漏斗的接口与网络相连，即漏斗是一个容量有限的内部队列。如果分组到达队列时，队列满，则分组将被丢弃。只要队列的长度不为0，分组就会以恒定的速率进入网络。也就是说，当队列的长度已经达到最大值时，如果主机还试图发送分组，这些分组将会被毫不客气地丢掉。实际上，这种策略相当于将用户产生的非平稳的分组流变成了一个平稳的分组流。从而平滑了用户数据分组的突发性，进而大大降低了拥塞的机会。该算法首先是由Turner(1986)提出来的，被称为漏斗算法(leaky bucket algorithm)。

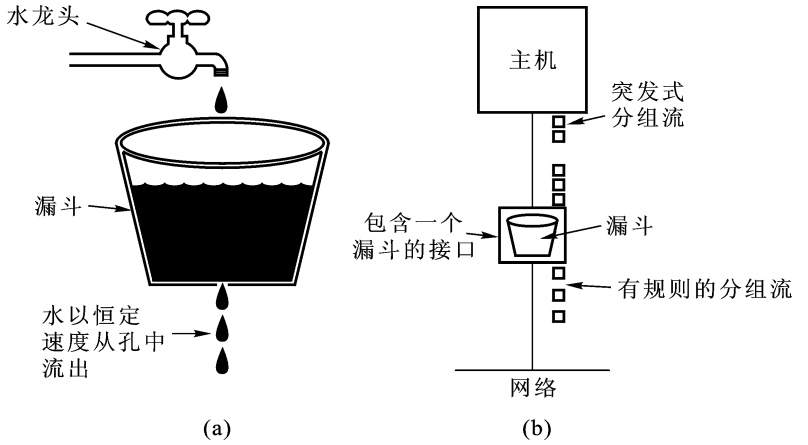


图6-8 漏斗算法

(a) 一个装水的漏斗 (b) 一个分组的漏斗模型

漏斗算法有两种实现方式：一种是针对分组长度固定的情况(如ATM信元)；另一种是针对可变长度分组的情况。如果分组的长度是固定的，漏斗算法每隔一个固定的时间间隔输出一个分组。当分组的长度可变时，则漏斗算法每隔一个固定的间隔，输出一个固定数目的字节(或比特)。如果漏斗每次可输出1024字节，则意味着每次可输出两个512字节的分组或4个256字节的分组。假设每次输出的最大字节数为 n，如果队列中的第一个分组长度 $l<n$，则该分组将被送入网络。如果队列中的第2个及后面的若干个分组长度之和小于 $n-l$，则这些分组都可以在这一次中发送。如果某一分组的长度 $l>n$ 且满足 $kn\leq$

$l \leq (k+1)n$，则该分组必须等待 k 个间隔方可传输，以保证每个间隔输出的平均字节数小于规定的数值。

例 6.4 有一台计算机能以 25 MB/s 的速率发送分组，而且网络也可能以该速率运行。但是，网络中的路由器只能在很短的时间内处理这样的速率。在长时间内，路由器只能以不超过 2 MB/s 的速率工作。假定漏斗的输出速率 $\rho=2$ MB/s，漏斗的容量为 $C=1$ MB，输入数据的突发长度为 1 MB，试求经过漏斗后，输入数据的输出速率及持续时间。

解 由于输入数据的突发长度为 1 MB，它刚好等于漏斗的容量，所以该突发数据都可以进入漏斗。输入数据产生的速率是 25 MB/s，突发长度是 1 MB，所以共持续 $\frac{1\text{ MB}}{25\text{ MB/s}}=40$ ms，如图 6－9 (a) 所示。由于漏斗的输出速率为 $\rho=2$ MB/s，所以突发数据将被整形，以 2 MB/s 的恒定速度输出。而持续时间为 $\frac{1\text{ MB}}{2\text{ MB/s}}=500$ ms，如图 6－9 (b) 所示。

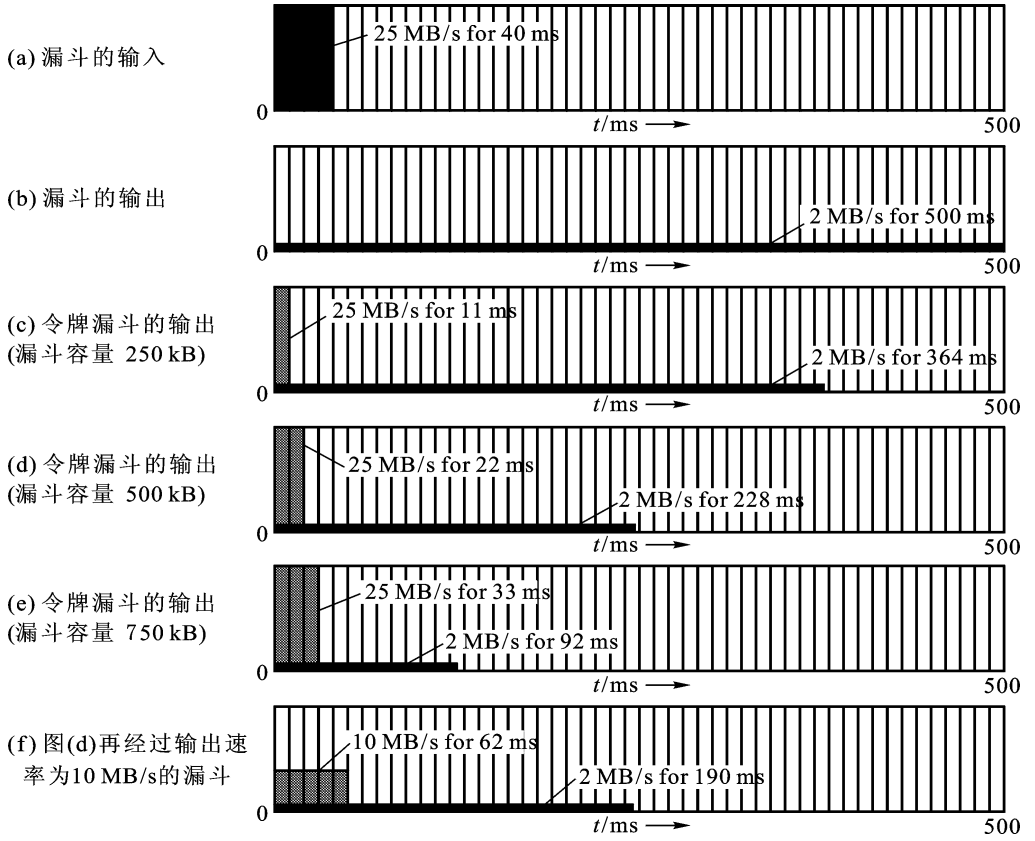


图 6－9 不同接入速率控制算法所对应的输入输出数据流示意图

3. 令牌漏斗算法

前面讨论的漏斗算法总是强迫输出模式保持一个固定的平均速率，而不管突发业务流的大小。通常，在很大应用环境中，都希望当大的突发业务到来时，输出可以相应地加快一些，即使得输出的业务流也具有一定的突发性。因此，人们提出了令牌漏斗算法（token bucket algorithm）。在令牌漏斗算法中，漏斗中保留的不再是数据分组，而是令牌。系统每隔 ΔT 个时间单位产生一个令牌，并送入漏斗。当漏斗满时，产生的新令牌将被丢弃。对于数据分组而言，只有当其获得了令牌后，才可以发送。当数据长度固定时，每获得一个令牌就可以发送一个分组。当有多个分组要发送时，可以获取漏斗中存在的多个令牌，根据可获得的令牌数来决定一次可以发送的分组数。

例 6.5　如图 6-10 (a) 所示，漏斗中共有 3 个令牌。此时，某一主机产生了 5 个新的分组。由于此时只有 3 个可用令牌，所以在这次发送的过程中，只能发送 3 个分组，而另外 2 个分组必须等待新的令牌后，才能发送。如图 6-10 (b) 所示。

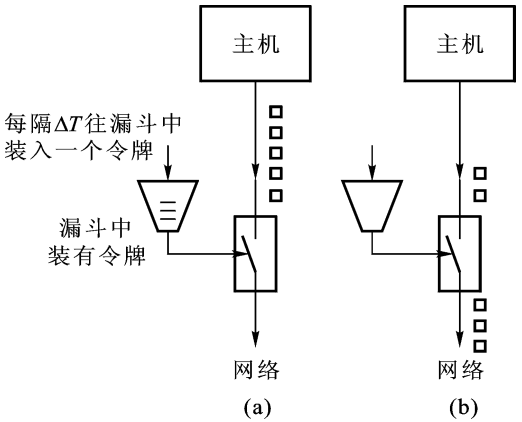
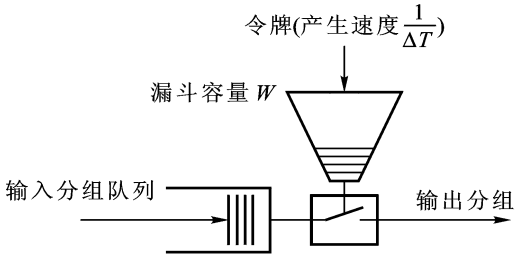


图 6-10　令牌漏斗算法举例
(a) 发送之前　(b) 一次发送之后

如果分组长度是可变的，一个令牌表示一次可发送 k 个字节，那么只有当该分组获得的多个令牌允许发送的字节数之和大于其长度时，才可以发送该分组。令牌漏斗算法工作原理图如图 6-11 所示。漏斗中令牌输出速率取决于用户的需求，如果令牌池的容量为 W，则每次可输出 0～*W 个令牌。如果用户无*需求，则漏斗无输出。如果漏斗满（即共有 W 个令牌在令牌池中），并且输入突发超过了 W 个分组，则漏斗一次可输出 W 个令牌，但后续令牌的输出速率仅为 $\frac{1}{\Delta T}$ 令牌/单位时间。要实现令牌漏斗算法，只需设置一个令牌计数器变量。这个计数器每隔 ΔT *个时间单位加*1，而每发送一个分组便减 1。当计数器为 0 时，就不能再发送分组。在以字节计数方式中，计数器每隔 ΔT *个时间单位增加*

k 个字节,每发送一个分组便减去分组的字节长度。



注:输入分组队列一次可捕获漏斗中的多个令牌而输出多个分组,或者说漏斗的输出速率取决于输入队列的需求。

图 6-11 令牌漏斗算法工作原理示意图

需要注意的是:令牌漏斗算法虽然允许业务流有一定的突发性,但对其突发长度有一个限制。通过下面的例子可以看到对突发长度的限制。

例 6.6 有一个容量为 250 KB 的令牌漏斗。在令牌漏斗中无积累的令牌时,令牌到达的速率允许用户以 2 MB/s 的速率向网络输出数据。网络可传输的速率可达 25 MB/s。若当 1 MB 突发数据到达时,令牌池满,求业务流的输出速率及持续时间。同时,讨论令牌漏斗容量为 500 KB 和750 KB 时,业务流的输出速率及持续时间。

解 由于在突发数据输出时,又有新的令牌产生。因此,设业务突发以最高速率输出的持续长度为 S 秒,令牌漏斗的容量为 C 个字节,令牌的产生速率为 *ρ 字节/秒,最大的输出速率为 M 字节/秒。显然,在 S 秒内,可输出的业务量为 (C+ρS) 字节。同时,在 S 秒内,以最高速率输出的突发字节数为 MS。因* 此可以得到

$$C+\rho S=MS \quad S=\frac{C}{(M-\rho)}$$

将参数代入,即 $C=250$ KB, $M=25$ MB/s, $\rho=2$ MB/s,可得突发以全速率输出的时间约为11 ms。剩余的时间内,突发将以 2 Mb/s 的速率输出。持续时间为 $\frac{1\ M-S\times 25\ \text{MB/s}}{2\ \text{MB/s}}\approx 364$ ms,其结果如图 6-9 (c) 所示。

当令牌漏斗容量为 500 KB 和 750 KB 时,类似于上面的方法,可以得到其结果分别如图 6-9 (d) 和 6-9 (e) 所示。

如果想使业务流更平滑,则可以在令牌漏斗之后再加一个漏斗。这个漏斗允许的分组传输速率应该比令牌漏斗在无令牌积累情况下允许的分组传输速率大,但要比网络最高可支持的速率低。图 6-9 (f) 给出了一个容量为 500 KB 的令牌漏斗后面再连一个 10 MB/s 的漏斗的输出流示意图,即将图 6-9 (d) 的输

出再经过输出速率为 10 MB/s 的漏斗。

为了进一步理解令牌漏斗算法，来讨论一下该算法的性能。

假定分组的长度固定。分组的到达过程是速率为 λ 的 Poisson 过程（注意这里没有考虑强突发性的分组到达）。令牌产生的间隔为 $\Delta T=\frac{1}{r}$，令牌漏斗的容量为 W。当漏斗满时，产生的新令牌将被丢弃。

可以用离散时间的 Markov 链来分析该系统。令系统的状态为 i，i 表示令牌的使用和需求情况。当 $i=0,1,\cdots,W$ 时，它表示漏斗中还有 $W-i$ *个令牌*可用，且没有未获得令牌的分组在等待，即 i 相当于漏斗中的剩余空间。当 $i=W+1,W+2,\cdots$ 时意味着有 $i-W$ *个未获得令牌的分组在等待且没有可用的*令牌。令系统的状态转移发生在 $0,\Delta T,2\Delta T,\cdots$ 时刻，即状态转移恰好发生在令牌产生之后。

由于分组到达服从 Poisson 分布，所以在 ΔT *内有 k 个分组到达的概率为*

$$a_k=\frac{e^{-\frac{\lambda}{r}}\left(\frac{\lambda}{r}\right)^k}{k!} \tag{6-2}$$

则状态转移概率为

$$P_{0i}=\begin{cases}a_{i+1} & i\geq 1\\ a_0+a_1 & i=0\end{cases} \tag{6-3}$$

由于当前的状态为 0，意味着令牌池中有 W 个可用令牌，且没有分组等待。则在 ΔT *内如果有分组到达，则必然会消耗令牌池中的令牌。但值得注意的是，*假定状态的转移时刻是在恰好有一个新令牌产生之后，所以由于有令牌消耗，令牌池已经不是满的了，则新产生的这个令牌可以注入令牌池。所以式（6-3）中的第一项为 ΔT *内有* $i+1$ 个分组的概率。式（6-3）中的第二项是有一个分组到达或没有分组到达的概率。因为如果没有分组到达，由于令牌池满，新令牌无法注入，所以状态保持不变；而如果有一个分组到达，则其消耗掉的一个令牌会在状态转移时刻之前由一个新的令牌补充。

对于 $j\geq 1$，有

$$P_{ji}=\begin{cases}a_{i-j+1} & j\leq i-1,\ i\geq 2\\ 0 & \text{其他}\end{cases} \tag{6-4}$$

由式（6-3）和式（6-4）可得状态转移图如图 6-12 所示。

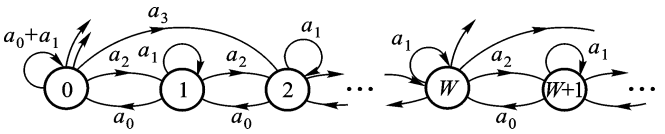


图 6-12 令牌漏斗系统的状态转移图

全局平衡方程为：

$$p_0 = a_0 p_1 + (a_0 + a_1) p_0 \tag{6-5}$$

$$p_i = \sum_{j=0}^{i+1} a_{i-j+1} p_j \qquad j \geq 0 \tag{6-6}$$

式中 p_i 为状态 i 的概率。

这些等式可以通过递归的方式来求解，即：

$$p_1 = a_2 p_0 + a_1 p_1 + a_0 p_2 \tag{6-7}$$

$$p_2 = \frac{p_0}{a_0} \frac{(1 - a_0 - a_1)}{a_0} (1 - a_1) - a_2 \tag{6-8}$$

依此类推，可写出 p_3，p_4，…的表达式。

另外，在稳态的情况下，令牌产生的平均速率（对所有的状态进行平均）应等于分组到达率 λ。令牌产生的平均速率为 $(1 - p_0 a_0)\gamma$，则有：

$$\lambda = (1 - p_0 a_0)\gamma$$

$$p_0 = \frac{\gamma - \lambda}{\gamma a_0} \tag{6-9}$$

令牌漏斗系统中一个分组获得一个令牌的平均时延为

$$T = \frac{1}{\gamma} \sum_{j=0}^{\infty} p_j \max(0, j - W) = \frac{1}{\gamma} \sum_{j=W+1}^{\infty} p_j (j - W) \tag{6-10}$$

6.3 实际系统中流量和拥塞控制算法

本节讨论几种现有系统中的流量和拥塞控制算法，并讨论其他一些流量和拥塞控制算法。

6.3.1 ARPANET 中的流量和拥塞控制

ARPANET 中的流控部分基于端到端的窗口式流控方法。它把通过子网连接的节点对之间的分组流看成是通过逻辑管道的一个“session”。对于每一个逻辑管道，窗口的长度为 8 个消息（每个消息由 1~8 个分组组成）。每一条消息有一个编号，指明它在窗口中的位置，目的节点收到一条消息后，向源节点返回一条准备接收下一条消息（RFNM，Ready For Next Message）的允许分组。发端收到 RFNM 后，将释放该消息的空间，以便接纳新的消息。如果在规定的时间内，发端没有收到 RFNM，则发端要发送一个控制分组来询问目的节点是否收到相应的信息。

由于 ARPANET 中分组的传输可能会乱序，对于有多个分组组成的消息，收端应当有足够的缓存空间来重装消息。做法是源节点发送一个预约缓存的消息（REQALL，Request for Allocation）给目的节点，以便使接收节点预留足够的

空间,如果目的节点同意预留,则发回一个分配消息 (ALL,Allocate) 。

在 ARPANET 中速率调整的方法是 :一个 session 通过的每一个节点都为该 session 计算一个流速的上限,称为 ration。该 ration 的确定是根据节点的处理能力、相关联的链路传输容量、缓存空间等因素动态调整。该 ration 信息和路由更新消息一起在全网中广播。源节点根据收到的各节点的 ration 信息,应使自己的流速低于该 session 中各节点确定的最小 ration。

6.3.2　SNA 网络中的流量和拥塞控制

SNA 网中的流控是基于对每一个虚电路的端到端流控。该算法称为 VRP (Virtual Route Pacing) 。该算法中窗口的大小根据业务的情况动态调整。最小窗口的长度等于路径上的链路数,最大窗口为其 3 倍。在具体实现时,每个分组头中包含两个拥塞比特,用来表示路径上拥塞情况,在源节点这两个拥塞比特均为 0。当中间发现虚电路出现“中等程度的拥塞”,它将第一个拥塞比特置为 1。当某一节点出现“严重拥塞”时,则将这两个拥塞比特都置为一。当分组到达目的节点时,根据拥塞比特的取值来调整窗口 :如果没有拥塞则增加窗口 ;如果有中等拥塞,则减少窗口 如果有严重拥塞,则将窗口置到最小值。

6.3.3　PARIS 网络中的流量和拥塞控制

PARIS (Packetized Automatic Routing Integrated System) 网络是一个高速分组交换网,它综合传输语音、图像和数据等信息,网络中采用自适应的最短路由,每条链路的长度是基于该链路运载的负荷大小。

速率控制基于漏斗法。用户在接入网络时,要向网络节点提供它的业务特征 (如 :平均速率、峰值速率、平均突发长度) ,网络节点把这些信息变成一个“等效容量”,即每一个 session 对其路径上各条链路的带宽需求。该网络节点在所有能接纳该 session 的路由中,计算出一条最短路由。如果没有合适的路径,该用户的 session 将被拒绝。

一个 session 接入网络后,其漏斗参数的选取将取决于该 session 的需求和路径上的负荷。每一个 session 可以发送比漏斗算法允许发送的分组 (称为“green”分组) 数更多的分组,这些超出的部分被称为“red”分组。网络将尽力保证“green”分组的传输。在网络拥塞时,这些“red”分组比“green”分组更容易被网络丢弃,并且网络还要保证“red”分组对“green”分组的丢失影响最小。

小　结

流量和拥塞控制的目的是限制网络中分组传输的平均时延和缓冲区溢出,

并公平地处理各session。本章首先介绍了几种常用的数据流控制技术:流量控制技术、拥塞控制技术和死锁防止技术,并对其在网络中所处的位置以及功能进行了详细的描述。然后,着重讨论了窗口式流量拥塞控制和漏斗式控制算法。在窗口式流量拥塞控制中,主要讨论了根据网络的拥塞情况,动态地调整拥塞窗口的大小,从而到达流量和拥塞控制的目的;在漏斗式流控算法中,主要是通过限制和平滑输入业务的突发性,使得输入业务的突发性在可控制的范围内,从而实现对网络拥塞的控制。最后,给出了几种实际网络中的流量和拥塞控制策略。

习 题

6.1 分组交换网中会出现哪几种死锁现象?它们的根源是什么?

6.2 分组交换网中可在几个层次上实现"流控"?试比较各层次上流控措施的优缺点及改善网络性能的效果大小。

6.3 试述流量控制和拥塞控制的区别和联系?

6.4 假定有一个网络如图6-13所示,该网络由5个节点组成,链路C→O,O→B,O→D的容量为1,链路A→O的容量为10。有两个session:第一个session经过C→O→D,其输入Poisson到达率为0.8;第二个是经过A→O→B,其输入Poisson到达率为f。假定中心节点O的缓冲较大,但它是有限的,它采用先到先服务的准则为两个session服务。如果节点O缓冲区满,输入分组将被丢弃,这些分组将由发送节点重发。发送节点重发的速率与其输出链路的容量成正比。试画出该网络总的通过量与输入速率f的关系曲线。

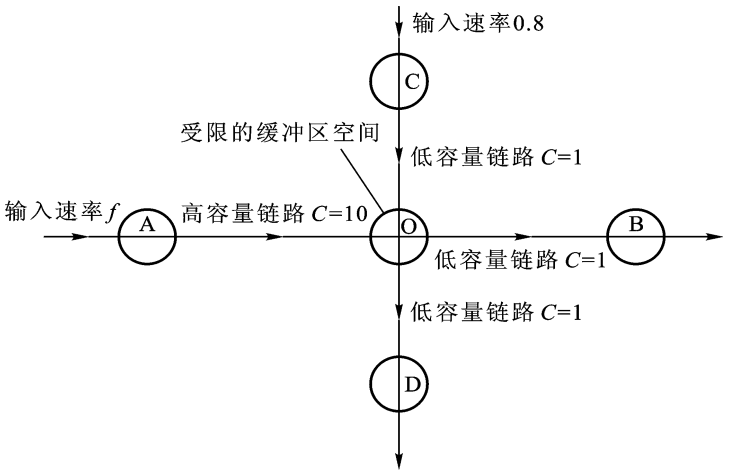


图6-13 习题6.4的图

6.5　假设有一个网络如图 6－14 所示。在该网络中仅有一条从节点 1 到节点 3 的虚电路，采用逐跳窗口流控。分组在链路 (1,2) 上的传输时间为 1，在链路 (2,3) 上的传输时间为 2 s，忽略处理和传播时延，允许分组的传输时间在各条链路上均为 1 s，节点 1 可不停地产生分组。在 $t=0$ 时，节点 1 和 2 有 W 个允许分组，节点 2 和 3 的缓存中没有分组，试求出 0～10 s 内分组在节点 1 和节点 2 开始传输的时间？

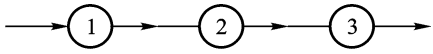


图 6－14　习题 6.5 的图

6.6　一个计算机连接到一个传输速率为 6 Mb/s 的网络上，计算机与网络节点之间采用令牌漏斗进行业务整形，令牌注入漏斗的速率为 1 Mb/s。假定漏斗在开始有 8 MB 的令牌容量，问计算机以全速 6 Mb/s 发送数据可持续多长时间？

第7章 网络结构设计

如果把网络中的主机或终端看成“节点”,通信线路看成“链路”,则网络的拓扑结构是指一个网络的通信链路和节点的几何排列或物理布局图形。网络拓扑是通过网络中的各个节点与通信线路之间的几何关系来表示的网络结构,并反映出网络中各实体之间的结构关系。因此,网络拓扑结构主要是指它的通信子网的拓扑结构。因而,拓扑设计是设计网络的第一步,也是实现各种协议的基础,而网络拓扑结构直接关系到网络性能、系统可靠性与通讯费用等因素。拓扑是一个数学概念,它把通信节点抽象成与其大小和形状无关的点,把连接实体的线路抽象成线,进而研究点、线、面之间的关系。

7.1 常用的网络拓扑结构

网络拓扑结构按照几何图形的形状可分为五种类型:总线拓扑、环形拓扑、树形结构、星形拓扑和网状拓扑。这些形状也可混合,构成混合拓扑结构。不同的网络拓扑结构适用于不同的网络规模。例如局域网应用的是总线、星形或环形拓扑结构,而广域网采用网状拓扑结构。

1. 总线拓扑

总线形拓扑结构由单根电缆组成,该电缆连接网络中所有节点。单根电缆称为总线,它仅仅只能支持一种信道,因此,所有节点共享总线的全部带宽。总线型网络拓扑结构如图7-1所示。

图7-1 总线拓扑结构

在总线网络中,当一个节点向另一个节点发送数据时,所有节点都将被动地侦听该数据,只有目标节点接收并处理发送给它的数据,其他节点将忽略该数据。基于总线拓扑结构的网络很容易实现,且组建成本很低,但其扩展性较差。当网络中的节点数量增加时,网络的性能将下降。此外,总线网络的容错能力较差,总线上的某点中断或故障将会影响整个网络的数据传输。因此,很少有网络采用一个单纯的总线拓扑结构。

2. 环形拓扑

在环形拓扑结构中,每个节点与两个最近的节点相连接以使整个网络形成

一个环形，数据沿着环向一个方向发送。环中的每个节点接收到传输的数据后，再将其转发到下一个节点。环形拓扑结构如图7－2所示。

图7－2 环形拓扑结构

图7－3 星形拓扑结构

与总线拓扑结构相同，当环中的节点不断增加时，响应时间也就变得越长。因此，单纯的环形拓扑结构非常不灵活或不易于扩展。此外，在一个简单环形拓扑结构中，单个节点或一处发生故障将会造成整个网络的瘫痪。

3. 星形拓扑

在星形拓扑结构中，网络中的每个节点都和一个中心控制节点连接在一起。网络中的每个节点将数据发送到中心控制节点，再由中心控制节点将数据转发到目标节点。星形拓扑结构如图7－3所示。

由于在星形网络中任何通信链路只连接两个设备（如一个工作站和一个集线器），因此链路故障最多影响两个节点。单个节点或链路发生故障，将不会导致整个网络的通信中断。但中心控制节点的失败将会造成一个星形网络的瘫痪。

由于使用中心控制节点作为连接点，星形拓扑结构可以很容易地移动、隔绝或与其他网络连接，这使得星形拓扑更易于扩展。因此，星形拓扑是目前局域网中最常用一种网络拓扑结构，现在的以太网都使用星形拓扑结构。

4. 树形结构

树形结构是星形结构的扩展分层结构，具有根节点和各分支节点，除了叶节点之外，所有根节点和子节点都具有转发功能，其结构比星形结构复杂，数据在传输的过程中需要经过多条链路，时延较大，适用于分级管理和控制系统。

5. 网状拓扑

在网状拓扑结构中，每两个节点之间都直接互连。如图7－4所示。

网状拓扑常用于广域网，在这种情况下，处于不同地理场所的节点都是互连的，数据能够从发送地直接传输到目的地。如果一个连接出了问题，将

图7－4 网状拓扑

能够轻易并迅速地更改数据的传输路径。由于对两点之间的数据传输提供多条链路,因此,网状拓扑是最具容错性的网络拓扑结构。

在计算机网络中还有其他类型的拓扑结构,如总线型与星型混合。总线型与环型混合连接的网络。在局域网中,使用最多的是总线型和星型结构。

7.2 网络拓扑结构的基本问题

当我们面对一个新的城市、一个新的办公场所,或者是对一个老的电信网络进行重新规划时,就需要确定干线节点的位置、节点之间传输的手段和容量大小,即进行网络的拓扑设计。

具体地讲,网络拓扑设计是在给定的条件下,针对给定的内容达到一定的设计目标。拓扑设计给定的条件有:需要相互通信的终端的地理位置;各终端之间业务需求。拓扑设计的主要内容:骨干通信网络的拓扑设计,它包括网络节点的位置、链路的选择和链路容量的确定;本地接入网络的设计,即如何将用户的终端连接到骨干网的网络节点上。拓扑设计的目标是:使得平均分组或消息的时延低于给定的值;满足可靠性的要求,即在有链路和节点故障的情况下,保证网络服务的完整性;在满足前两个目标的条件下,使投资和运行的成本最小。

一个在用户终端固定情况下的网络拓扑设计问题如图7-5所示。

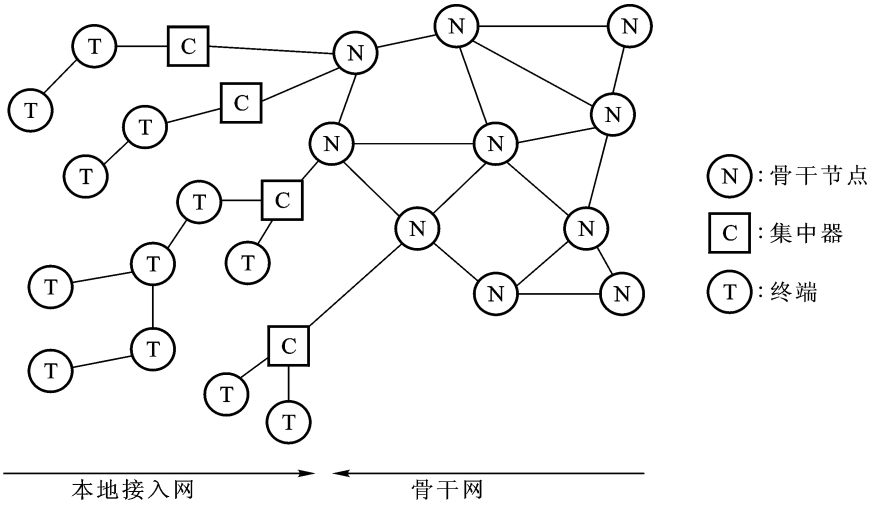


图7-5 网络拓扑设计举例

图7-5中包括两部分:骨干网部分和本地接入网络部分。骨干网由骨干网节点(N)和节点间的链路组成。骨干网节点相当于核心网路由器或交换机。本地接入网络有终端(T)和集中器(C)组成。集中器相当于局域网中的Hub或边

缘路由器或交换机等设备。

当用户终端是移动的情况下,接入网设计还要解决一个覆盖问题,即在一个网络覆盖的区域内,用户无论在何处,都要能够接入到网络中,这时网络中的集中器(或者基站)应足够多。

在一般情况,很难用公式来严格地描述上述拓扑设计问题。然而在实际当中,问题可以简化。在某些情况下,拓扑设计的部分问题已经得以解决。例如,部分接入网或骨干网已经存在。因此,可以将拓扑设计问题分为接入网拓扑设计和骨干网拓扑设计。

7.3 接入网的拓扑设计

接入网主要负责将非交换设备接入骨干网络,同时接入网设备还肩负着不同技术间的转换。而接入网拓扑设计要解决的主要问题则是如何将分散的通信终端汇聚起来,接入到高一层的通信子网中去。接入网拓扑设计是最复杂和多样化的。它采用什么样的类型接入与骨干网的类型紧密相关。

7.3.1 接入网的分类

接入网的分类方法有很多种,例如可以根据传输媒介、拓扑结构、使用技术、接口标准、业务带宽、业务种类等等进行分类。将这些因素都考虑进去,接入网的分类方法自然就很多,但常用的主要有下面几大类,它们可单独使用或混合使用。

(1) 金属用户线上的 xDSL:它又可分为 IDSL(ISDN 数字用户环路)、HDSL(利用两对线双向对称传输 2 Mb/s 的高速数字用户环路)、VDSL(甚高速数字用户环路)、ADSL(不对称数字用户环路)。上述系统的拓扑结构是点到点。

(2) 同轴电缆上的 HFC(双向混合光纤同轴电缆接入传输系统),拓扑结构是树型或总线型,下行物理上通常为广播方式。

(3) 光纤接入系统:可分为有源与无源系统,有源系统有基于 PDH 和 SDH 之分,拓扑结构可以是环型、总线型、星型或它们的混合型,也有点对点的应用。无源光网络即 PON,有窄带与宽带之分,目前宽带 PON 已经标准化的是基于 ATM 的 PON,即 APON。PON 本身下行是点到多点系统,上行为多点到点,上行时需要解决多用户争用问题,目前上行大多用 TDMA(时分多址)技术。

(4) 无线接入系统:如无绳电话(如 DECT)、集群电话、蜂窝移动通信、微波通信或卫星通信可分为很多类,对应不同的频段,容量、业务带宽和覆盖范围各异。无线接入主要的工作方式是点到多点(即星型拓扑结构),上行解决多用户争用的技术有 FDMA(频分多址)、TDMA(时分多址)和 CDMA(码分多址),从

频谱效率看 CDMA 最好，TDMA 其次。其中 CDMA 又可有扩谱（DS）、跳频（FH）和同步（S－CDMA）几种。

总的说来，接入网可以分为有线接入和无线接入网两大类。

7.3.2 有线接入网的设计

有线接入网的设计主要是考虑如何将用户通过有线（铜线或光纤等）接入骨干网络。首先考虑如何将图 7－6（a）所示网络中的终端用户通过骨干节点（集中器）接入到骨干网中。

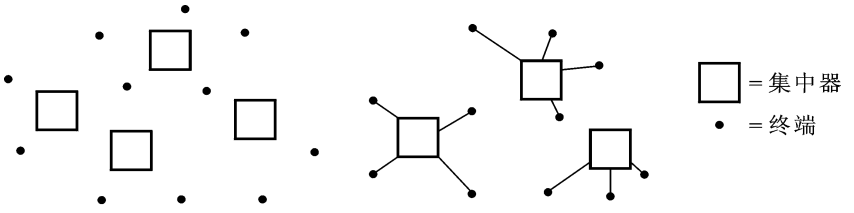


图 7－6 有线接入网络设计

（a）终端用户和集中器的位置 （b）终端用户和集中器之间的连接

设用户 *i* 连接到集中器的成本为 a_{ij}，用户 i 与集中器 j 是否相连用变量 x_{ij} 表示，$x_{ij}=1$ 表示用户 *i* 与集中器相连，否则 $x_{ij}=0$。总的成本函数可以表示为

$$\text{cost} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_{ij} \tag{7-1}$$

因此将 n 个终端连接到 m 个集中器上可以表示为：

$$\text{目标函数：} \min_{i,j}\text{cost} = \min_{i,j} \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_{ij} \tag{7-2}$$

$$\text{约束条件：} \sum_{j=1}^{m} x_{ij} = 1 \qquad \text{对所有用户 } i \tag{7-3}$$

$$\sum_{j=1}^{m} a_{ij}x_{ij} \le k_j \qquad \text{对所有的集中器} \tag{7-4}$$

式（7－3）表示每个用户仅会连接到一个集中器上，式（7－4）表示每个集中器最大能接入 k_j 个用户。如果将 x_{ij} 取值 0 或 1，用 $0 \le x_{ij} \le 1$ 来代替，则式（7－2）到式（7－4）是线性规划中的运输问题，有很多有效方法可以解决。

下面讨论集中器的位置优化问题。

如果集中器在每一个位置 j 的成本为 b_j，则式（7－1）的成本函数变为

$$\text{cost} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_{ij} + \sum_{j=1}^{m} b_j y_j \tag{7-5}$$

式中，y_j=1或0分别表示集中器是否放置在位置 j。y_j=1表示集中器放在位置 j。此时，式(7-4)对应的约束条件应变为

$$\sum_{i=1}^{n} x_{ij} \leq k_j y_j \qquad \text{对所有的集中器} \qquad (7-6)$$

在式(7-3)和式(7-6)的约束条件下，对式(7-5)最小化的问题是非线性规划中一个典型的仓库选址问题。有很多成熟的方法可以解决该问题，请参考有关文献。

7.3.3 无线接入网的设计

无线接入可分为移动接入与固定接入两种。其中移动接入又可分为高速和低速两种。高速移动接入一般可用蜂窝系统、卫星移动通信系统、集群系统等。低速接入系统可用微小区和微微小区。固定接入是从交换节点到固定用户终端采用无线接入，它实际上是PSTN/ISDN网的无线延伸。其目标是为用户提供透明的PSTN/ISDN业务，固定无线接入系统的终端不含或仅含有限的移动性。接入方式有微波一点多址、蜂窝区移动接入的固定应用、无线用户环路及卫星VSAT网等。

无线接入的基本方式如图7-7所示。在用户侧和网络侧都必须有对应的无线传输设备，网络侧的设备常称为基站。

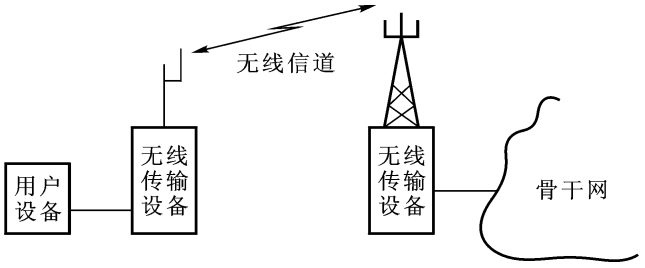


图7-7 无线接入示意图

无线接入网与有线接入网设计的主要差别是无线链路的长度受到工作频率、发射功率、天线高度、地形地物等因素的限制有一个最大长度(通信距离)。当无线用户的位置固定时，基站的位置选择类似于有线网中集中器的选择，但这时要附加一个通信距离的限制。当用户可以任意移动，要保证用户在给定的区域(如一个城市)内的任何一个地方都可以接入网络，就要求在任一地点的通信半径内至少有一个基站，或者说要求无线接入网络无缝地覆盖给定的区域。

下面介绍地面无线接入网

在无线移动通信系统中，传输损耗是随着距离的增加而增加的，并且与地形环境有着密切的关系，因此移动台和基站之间的通信距离是有限的，例如：若基

站天线高度为70 m,工作频率为450 MHz,天线增益为8.7 dB,发射机功率为25 W,移动台天线为3 m,接收灵敏度为－113 dBm,接收天线增益为1.5 dB,则通信可靠性达90%的通信距离为25 km。

为了使得服务区达到无缝覆盖,提高系统的容量,就需要采用多个基站来覆盖给定的服务区域(每个基站的覆盖区域称为一个小区)。从理论上讲可以给每个小区分配不同的频率,但这样需要大量的频率资源,且频谱利用率很低。为了减少对频率资源的要求和提高频谱利用率,需将相同的频率在相隔一定距离的小区中重复使用,只要使用相同频率的小区(同频小区)之间的干扰足够小即可。这就给无线接入网络的设计提出了一个新的附加条件。下面针对目前最常用的蜂窝网来讨论接入网设计时应该注意的问题。

在将平面区域划分成小区时,需要考虑:

(1) 小区的形状

全向天线辐射的覆盖区域是一个圆形。为了不留空隙地覆盖整个平面的服务区,一个个的圆形辐射区之间一定含有很多的交叠。在考虑了交叠之后,实际上每个辐射区的有效覆盖区是一个多边形。根据交叠情况的不同,若在每个小区相间120°设置三个邻区,则有效覆盖区为正三角形;每个小区相间90°设置四个邻区,则有效覆盖区为正方形;若每个小区相间60°设置六个邻区,则有效覆盖区为正六边形;小区形状如图7－8所示。可以证明,要用正多边形无空隙、无重叠的覆盖一个平面的区域,可取的形状只有正三角形、正方形和正六边形这三种。那么这三种形状哪一种最好呢?在辐射半径 r 相同的条件下,计算出三种形状小区的邻区距离、小区面积、交叠区宽度和交叠区面积如表7－1。

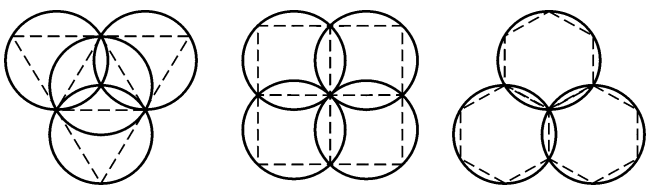


图7－8 小区的形状

表7－1 三种形状小区的比较

小区形状	正三角形	正方形	正六边形
邻区距离	r	$2r$	$3r$
小区面积	$1.3r^2$	$2r^2$	$2.6r^2$
交叠区宽度	r	$0.59r$	$0.27r$
交叠区面积	$1.2\pi r^2$	$0.73\pi r^2$	$0.35\pi r^2$

由表可见,在服务区面积一定的情况下,正六边形小区的形状最接近理想的圆形,用它覆盖整个服务区所需的基站数最少,也就最经济。正六边形构成的网络形同蜂窝,因此称小区形状为六边形的小区制移动通信网称为蜂窝移动通信网。

(2) 区群的组成

相邻小区显然不能用相同的信道。为了保证同信道小区之间有足够的距离,附近的若干小区都不能用相同的信道。这些不同信道的小区组成一个区群,只有不同区群的小区才能进行信道再用。

区群的组成应满足两个条件:一个是区群之间可以邻接,且无空隙无重叠的进行覆盖;二是邻接之后的区群应保证各个相邻同信道小区之间的距离相等。满足上述条件的区群形状和区群内的小区数不是任意的。可以证明,区群内的小区数应满足下式:

$$N = i^2 + ij + j^2 \qquad (7-7)$$

式中 i,j为正整数。由此可算出 N 的可能取值如表 7-2。相应的区群形状如图 7-9 所示。

表 7-2 区群小区数 N的取值

j \ N \ i	0	1	2	3	4
1	1	3	7	13	21
2	4	7	12	19	28
3	9	13	19	27	37
4	16	21	28	37	48

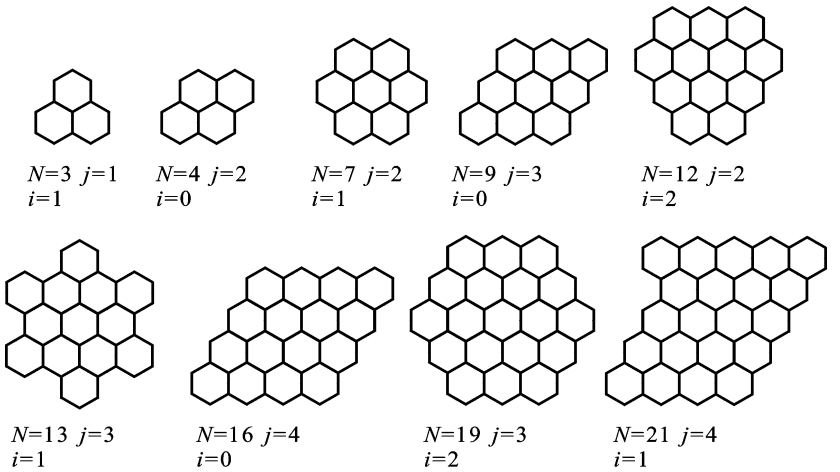


图 7-9 区群的组成

(3) 同频(信道)小区的距离

区群内小区数不同的情况下,可用下面的方法来确定同频(信道)小区的位置和距离。如图7-10所示,自某一小区A出发,先沿边的垂线方向跨 j 个小区,再向左(或向右)转60°,再跨 i 个小区,这样就到达相同小区A。在正六边形的六个方向上,可以找到六个相邻同信道小区,所有A小区之间的距离都相等。

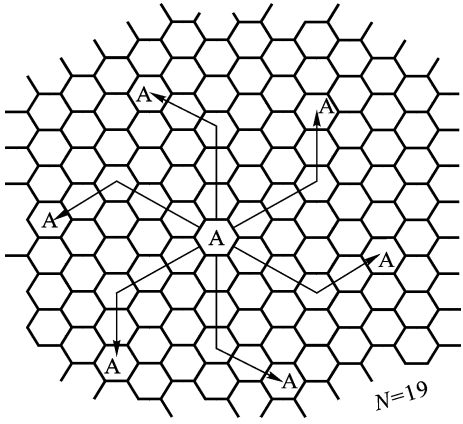


图7-10 同信道小区的确定

设小区的辐射半径(即正六边形外接圆的半径)为 r,则从图7-10可以算出同信道小区中心之间的距离为

$$D = \sqrt{3}r\sqrt{\left(j+\frac{i}{2}\right)^2+\left(\frac{\sqrt{3}i}{2}\right)^2}$$

$$= \sqrt{3(i^2+ij+j^2)}\cdot r \qquad (7-8)$$

$$= \sqrt{3N}\cdot r$$

可见,群内小区数N越大,同信道小区的距离就越远,抗同频干扰的性能就越好。例如:$N=3,\frac{D}{r}=3$;$N=7,\frac{D}{r}=4.6$;$N=19,\frac{D}{r}=7.55$

(4) 中心激励和顶点激励

在每个小区中,基站可以设在小区的中央,用全向形成圆形覆盖区,这就是所谓的"中心激励"方式,如图7-11(a)所示。也可以将基站设计在每个小区六边形的三个顶点上,每个基站采用三副120°扇形辐射的定向天线,分别覆盖三个相邻小区的各三分之一区域,每个小区由三副120°扇形天线共同覆盖,这就是所谓的"顶点激励"方式,如图7-11(b)所示。采用120°的定向天线后,所接收的同频干扰功率仅为采用全向天线系统

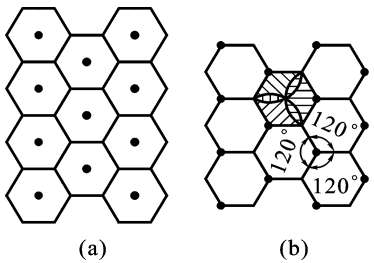


图7-11 两种激励方式

(a) 中心激励 (b) 顶点激励

的$\frac{1}{3}$,因而可以减少系统的同道干扰。另外,在不同地点采用多副定向天线可消除小区内障碍物的阴影区。

(5) 小区的分裂

在整个服务区中每个小区的大小可以是相同的,这只能适应用户密度均匀的情况。事实上服务区内的用户密度是不均匀的,例如城市中心商业区的用户密度高,居民区和市郊区的用户密度低。为了适应这种情况,在用户密度高的市中心可以使小区的面积小一些,在用户密度低的市郊区可以使小区的面积大一些。如图 7-12 所示。

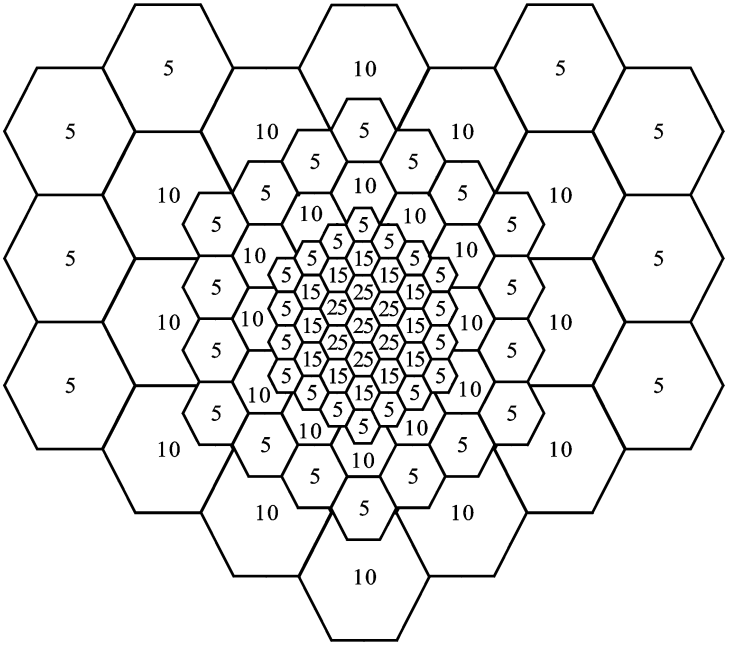


图 7-12 用户密度不等时的小区结构

图中每个小区中的数字表明的是在该小区内可使用的信道数。从图中可以看出市中心用户密度最高,所以小区的面积划得最小,但分配给这些小区的可用信道数却最多。

另外,对于已设置好的蜂窝通信网,随着城市建设的发展,原来的低用户密度区可能变成了高用户密度区,这时应在该地区设置新的基站,将小区面积划小。解决以上问题可用小区分裂的方法。

以 120° 扇形辐射的顶点激励方式为例,如图 7-13所示,在原小区内分设三个发射功率更小一些的新基站,就可以形成几个面积更小些的正六边

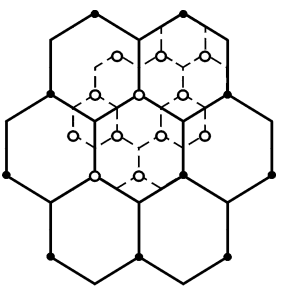


图 7-13 小区分裂

形小区，如图中虚线所示。

7.4 骨干网的拓扑设计

虽然在不同的应用场合会有不同的类型的骨干网，但在其设计过程中都会面临同样的问题。总的来说，骨干网的设计问题可以表述为：给定骨干网节点的位置和节点之间的流量，在使网络成本最小的原则下，来选择每条链路的容量和流量，以满足时延和可靠性的要求。如果不是非常简单的情况，上述问题是一个非常难解的组合问题（combinatorial problem）。而且，即使该问题有解，所得到的解也可能不切实际。在实际系统中，节点间的链路可能仅有有限种选择，通常采用试探法来解决实际问题。

假定将一条链路的容量指定为0，则实际上是消除了该链路。因此，可把拓扑设计问题看作是容量分配问题的特例。下面讨论容量分配的试探法。

试探法的基本思想是从一个给定的网络拓扑开始，一次改变其一条或多条链路的容量，检查该新的拓扑是否满足约束条件并且具有更低的成本，直到求得最佳的结果。试探法的假定条件有：

① 已知网络的节点以及节点之间的输入业务流。

② 给定所有的链路容量 C_{ij}，并已选定一种路由算法来确定所有的链路（i，j）的流量 F_{ij}。最常用的可能方法是该流量可以使系统的成本函数最小。

③ 时延必须满足要求。例如，由M/M/1队列给出的时延

$$D = \frac{1}{\gamma}\sum_{(i,j)} \frac{F_{ij}}{C_{ij} - F_{ij}} + d_{ij}F_{ij} \qquad (7-9)$$

应小于给定的时延门限T，即 $D \leq T$。式中，d_{ij}为链路（i，j）的处理和传播时延，γ是输入到网络的总的业务流。

④ 可靠性必须满足要求。例如，通常要求网络具有2连通性（即单个节点故障后，网络仍是连通的），或k连通性（即在k－1个节点故障后，剩余节点仍是连通的）。

⑤ 有一个成本函数，并利用该函数来对不同的网络拓扑排序。

我们的目标就是寻找一个拓扑满足条件(3)和(4)的时延和可靠性要求，并使条件(5)的成本尽可能地小。为了求得满足上述条件的拓扑，这里给出一个原型迭代试探法（prototype iterative heuristic method）。在每次迭代开始，有两个拓扑：一个是“当前最好的拓扑”，它是满足时延和可靠性约束条件，到目前为止发现的成本最低的拓扑；另一个是“试验拓扑”，它是在当前的迭代步骤中需要评估的拓扑，以确定它是否具有更好的性能。迭代的步骤如下：

Step1：（分配流量）利用假定(2)的路由算法来计算试验拓扑的链路流量 F_{ij}。

Step2：(检查时延) 求解试验拓扑中每个分组的平均时延 D [例如，可采用式 (7－9)]。检查 $D \leq T$ *是否成立，T 为时延门限。如果* $D \leq T$，*执行* Step3，否则执行 Step5。

Step3：(检查可靠性) 测试试验拓扑是否满足可靠性的要求。如果满足，执行 Step4 如果不满足，执行 Step5。

Step4：(检查成本的改进程度) 如果试验拓扑的成本低于当前最好的拓扑，则用试验拓扑代替当前最好的拓扑。

Step5：(产生一个新的试验拓扑) 利用某种试探的方法，改变当前没有考虑过的一个试验拓扑，返回 Step1。

当没有新的试验拓扑可以产生或不可能有实质性改善的情况下，算法结束。利用这种求解方法得到的解不一定保证是最佳的。一个可能的改进方法是采用不同的开始拓扑，重复上述迭代，直至找到一个最满意的解。

在上述迭代过程中，一个重要的问题是如何产生试验拓扑。有很多试探的规则可以用来产生新的拓扑。例如降低或删除使用率很低的链路 $\frac{F_{ij}}{C_{ij}}$和$\frac{C_{ij}}{F_{ij}}$很低；当时延不满足要求时，可以增加高负荷链路的容量。这些可能性的组合叫做支路交换试探法 (branch exchange heuristic)。该方法的基本思路是在原来的拓扑中删去一条链路，并重新增加一条链路。选择删除链路和增加链路的一个常用方法称为饱和割集法 (saturated cut method)。

所谓割集是指网络拓扑中的这样一些链路的集合，删除该集合中的所有链路，将使网络拓扑分为两个不相连 (不连通) 的部分，如图 7－14 所示。如果将虚线上的两条链路删除，网络将分为两个不相连的部分 N_1 和 N_2。所谓饱和割集是指链路利用率 (负荷) 非常高的割集。例如图 7－14 中与虚线对应的割集为饱和割集。

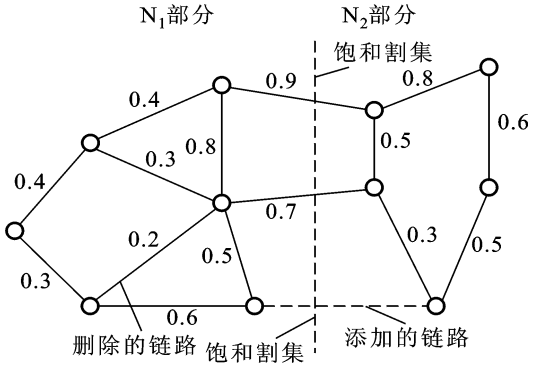


图 7－14 割集和饱和割集示意图

由于饱和割集中链路利用率太高，会影响网络中的时延性能。因此，在 N_1 和 N_2 之间增加一条链路，将会有助于降低割集链路的利用率。同时可以删除一条利

用率最低的链路，如图 7－10 中，增加一条虚线，删除利用率为 0.2 的链路。

总而言之，对于骨干网的建设需要考虑以下几个方面的问题：

① 骨干网承载的业务和这些业务的流量分配；

② 不同的拓扑结构，它的数据交换能力会有所不同；

③ 是否需要建设有冗余配置的骨干来抵御意外事件。链路冗余度、节点冗余度和数据交换能力的相互关系和影响；

④ 采用不同的组网技术对网络流量的影响；

⑤ 骨干节点的分布。

通过合理的设计来布设一个功能强大、性能稳定的网络。

小 结

本章首先讨论了常用的网络拓扑结构，然后详细论述了网络拓扑设计中应该关心的基本问题，一个可行的网络拓扑应能够很好地平衡网络的成本、网络的可靠性和网络的传输能力（通过量和时延）等方面的因素。接着讨论了接入网和骨干网的拓扑设计问题。在接入网中要解决怎样使更多的用户接入网络的问题，在有线接入网中主要讨论了 n 个终端如何连接到 m 个集中器上的数学模型；对于无线接入网主要通过典型的蜂窝网络进行了详细的讨论；在骨干网中我们设计的基本方法是原型迭代试探法，在该方法中可通过支路交换试探法来产生网络试验拓扑。

习 题

7.1 常用的网络拓扑结构有哪些？它们分别有什么特点？

7.2 拓扑设计主要要考虑哪些因素？

7.3 在无线接入网中，为什么说最佳的小区形状是正六边形？

7.4 设某蜂窝移动通信网的小区辐射半径为 8 km，根据同频干扰抑制的要求，同信道小区之间的距离应大于40 km，问该区群应如何组成？试画出区群的构成图、群内各小区的信道配置以及相邻同道小区的分布图。

7.5 假设有一个网络拓扑如图 7－15所示，试求该图的饱和割集，并提出采用支路交换法改进网络传输性能的建议。

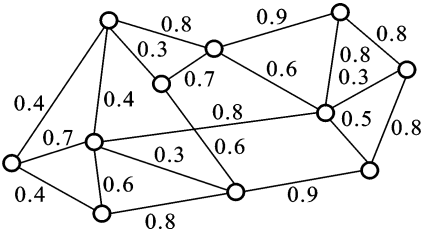


图 7－15 习题7.5的图

附　录

A

ABM	Asynchronous Balanced Mode
ADCCP	Advanced Data Communications Control Protocol
ADSL	Asymmetric Digital Subscriber Line
ANSI	American National Standards Institute
AODV	Ad hoc On Demand Distance Vector Routing
ARM	Asynchronous Response Mode
ARP	Address Resolution Protocol
ARQ	Automatic Retransmission Request
ATM	Asynchronous Transfer Mode

B

B－F	Bellman-Ford

C

CCITT	International Consultative Committee on Telegraphy and Telephone
CDMA	Code Division Multiple Access
CGSR	Clusterhead Gateway Switch Routing Protocol
CLP	Cell Loss Priority
CRC	Cyclic Redundancy Check
CRP	Collision Resolution Period
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTS	Clear To Send

D

DARPA	Defense Advanced Research Projects Agency
DCF	Distribution Coordination Function

DECT Digital Enhanced Cordless Telecommunications
DIFS Distribution (Coordination Function) Inter Frame Space
DISC DISConnect
DLC Data Link Control
DNS Domain Name Service
DREAM Distance Routing Effect Algorithm for Mobility
DSDV Destination-Sequenced Distance Vector Routing
DSL Digital Subscriber Line
DSR Dynamic Source Routing

F

FCFS First Come First Service
FDDI Fiber Distributed Data Interface
FDM Frequency Division Multiplexing
FDMA Frequency Division Multiple Access
FSR Fisheye State Routing
FSLS Fuzzy Sighted Link State
FTP File Transfer Protocol
F-W Floyd-Warshall

G

GFC Generic Flow Control
GPRS General Packet Radio Services
GPSR Greedy Perimeter Stateless Routing
GSM Global System for Mobile communication

H

HDSL High Speed Digital Subscriber Line
HDLC High-level Data Link Control
HEC Header Error Control
HSR Host Specific Routing protocol
HTTP Hyper Text Transfer Protocol

I

ICMP Internet Control Message Protocol

IFS	Inter Frame Space
IMT－2000	International Mobile Telecommunications in the year 2000
ISDN	Integrated Service Digital Network
IS－IS	Intermediate System-to-Intermediate System
ISO	International Standardization Organization
IP	Internet Protocol

L

LAN	Local Area Network
LANMAR	Landmark ad hoc Routing
LAR	Location Aided Routing
LAPB	Link Access Protocol Balanced
LLC	Logical Link Control
LMDS	Local Multipoint Distribution System

M

MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
MST	Minimum Weight Spanning Tree

N

NAV	Network Allocation Vector
NRM	Normal Response Mode
NNI	Network Node Interface
NNTP	Network News Transfer Protocol

O

OSI	Open Systems Interconnection
OSPF	Open Shortest Path First

P

PARIS	Packetized Automatic Routing Integrated System
PCF	Point Coordination Function
PIFS	Point (Coordination Function) Inter Frame Space
PRNET	Packet Radio Network

PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
PT	Payload Type

Q

QPSK	Quadrature Phase Shift Keying

R

RARP	Reverse Address Resolution Protocol
REJ	Reject
REQALL	Request for Allocation
RFNM	Ready For Next Message
RIP	Routing Information Protocol
RNR	Receive Not Ready
RR	Receive Ready
RTS	Request To Send

S

SABM	Set Asynchronous Balanced Mode
SDMA	Spatial Division Multiple Access
SDH	Synchronous Digital Hierachy
SIFS	Short Inter Frame Space
SLIP	Serial Line Internet Protocol
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SNRM	Set Asynchronous Response Mode
SONET	Synchronous Optical Network
SREJ	Selective Rejected
SSR	Signal Stability Routing protocol

T

TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TORA	Temporally-Ordered Routing Algorithm

U

UA	Unnumbered Acknowledgment
UDP	User Datagram Protocol
UNI	User-Network Interface

V

VCI	Virtual Channel Identifier
VDSL	Very high speed Digital Subscriber Line
VRP	Virtual Route Pacing
VPI	Virtual Path Identifier

W

WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WRP	Wireless Routing Protocol