

第 1 章

# 简介

---

网络编码是一个新兴的研究领域。2000 年, Ahlswede、Cai、Li 和 Yeung 发表了开创性论文<sup>[3]</sup>, 标志着网络编码的诞生。网络编码像许多其他新兴领域一样, 存在一定程度的混乱性, 人们既对其可能性充满兴奋, 又对其潜能存在一定争议。本书的首要目的之一是澄清这种混乱。因此, 我们严谨地以网络编码定义开篇。

## 1.1 网络编码的定义

网络编码的定义不是简明易懂的。目前使用的定义方法有以下几种。

开创性论文[3]中, Ahlswede、Cai、Li 和 Yeung 指出他们将网络中节点进行的编码称为网络编码, 其中编码指由输入到输出的任意因果映射。这是网络编码最广义的定义。然而, 该定义并没有将网络编码的研究与历史悠久的、存在诸多难题的网络信息论或多用户信息论区别开来。由于不希望本书专注于网络信息论(已有一些很好的涵盖网络信息论的文献, 如文献[27]的第 14 章), 我们将对网络编码作进一步的定义。

Ahlswede 等的文章区别于大多数网络信息论文章的特性之一在于：他们不是研究本质上各节点对其他节点均有任一概率影响的一般性网络，而是明确研究由节点通过端到端无差错链路所连接而成的网络。因此，Ahlswede 等的网络模型为常规网络信息论研究网络的一个特例，而这种模型与当前网络非常相关，本质上所有的有线网络一旦将物理层抽象为传输比特的无差错管道便均属于该模型。

因此，网络编码的另一个可能定义为无差错链路网络中节点上的编码。该定义区分了网络编码与有噪链路下信道编码的功能；类似地，我们可以区分网络编码与信源编码的功能。这里，网络编码可视为独立、不可压缩信源编码。此定义将被频繁使用，且基于此可将网络编码的研究简化为网络信息论的一个特例。事实上，该特例在 2000 年前已经得到了一些研究（比如，参考文献[51, 131]）。这虽然减少了网络编码的部分创新性，但我们仍能够在该定义下作进一步的研究。

网络编码的大量研究工作集中在网络编码的一个特定形式上，即随机线性网络编码。随机线性网络编码由文献[58]引入，是一种简单的随机编码方法，它为每一个信源过程维持一个系数向量，该系数向量由每个编码节点进行更新。换句话说，随机线性网络编码要求网络传输消息时附加一定程度的额外信息——这里即为系数向量。当今通信网络中，存在一种广泛使用的网络，不仅可以简单地提供这样的额外信息，而且还是由无差错链路组成：分组网络。经分组处理后，这样的额外信息或边信息可以放置在包头，而将边信息放置在包头也的确是现在的通用惯例（比如，经常将序列号放置在包头以保持顺序）。

网络编码的第三种定义为分组网络（对数据进行分组，网络编码应用于分组的内容）节点编码，或者更广义地定义为物理层以上的编码。这与网络信息论不同，后者通常关注物理层编码。本书采用此种定义。局限于分组网络有时的确会不必要地限制我们的研究范围，而且超出分组网络的相关研究成果也不宜作介绍。然而，该定义还是非常有用的，因为它使得我们的论述根植于与实际相关的具体设置。

## 1.2 网络编码的效用

定义完网络编码，现在我们讨论网络编码的效用。网络编码可以改善吞吐量、鲁棒性、复杂性和安全性。下面依次讨论这些性能因素。

### 1.2.1 吞吐量

网络编码最为人熟知且最易解释的效用是提高吞吐量。吞吐量的改善由更有效的分包传输实现,即使用更少的分包传输获取更多的信息。Ahlswede 等的论文[3]给出了这种改善的最著名例子,他们考虑有线网络中的多播问题。他们的例子通常被称为蝴蝶网(如图 1.1 所示),该网具有一个信源节点到两个信宿或目的节点的多播特性。两个信宿都想获取信源节点发送的全部信息。在他们考虑的约束网络中,只有当一个中间节点(非信源/信宿节点)打破分组网络中传统的复制转发路由方式,并进行编码操作后,才能够建立所需的多播连接。该中间节点对接收到的两个包进行二进制加法或者异或处理得到一个新包,并将该新包输出。因此,如果两个接收包的内容为比特向量  $b_1$  和  $b_2$ ,则输出包为  $b_1 \oplus b_2$ ,即向量  $b_1$  和  $b_2$  的逐位异或。两信宿节点分别对其接收到的数据包作进一步的编码处理实现解码。信宿节点  $t_1$  对  $b_1$  和  $b_1 \oplus b_2$  进行异或处理解出  $b_2$ ,同样的,信宿节点  $t_2$  对  $b_2$  和  $b_1 \oplus b_2$  进行异或处理解出  $b_1$ 。传统路由模式下,我们可以传输信息  $b_1$  和  $b_2$  到节点  $t_1$ ,但是同时只能够传输信息  $b_1$  或  $b_2$  中的一个到节点  $t_2$ 。

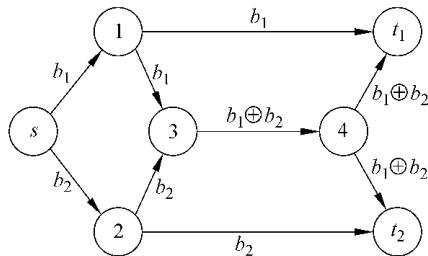


图 1.1 蝴蝶网。该网络中,每条弧代表能够可靠传输一个包的有向链路。

信源节点  $s$  有两个包  $b_1$  和  $b_2$  待传输,我们希望将这两个包的内容同时通信至信宿节点  $t_1$  和  $t_2$ 。

设计该蝴蝶网说明了重要的一点:网络编码能够增加有线网络的多播吞吐量。蝴蝶网中 9 个包完成了两个信息包的通信。不进行编码的情况下,通过 9 个包的传输无法完成同样多信息的通信,必须通过补充额外的传输(比如,节点 3 到节点 4 的额外传输)来实现。

虽然网络编码能够增加有线网络的多播吞吐量,但是其对吞吐量的改进不仅仅局限于多播或者有线网络。对蝴蝶网进行简单修改,可以得到一个涉及两个单播连接的例

子,它在编码情况下可以实现,而无编码则做不到,如图 1.2 所示。本示例包括两个单播连接。目前考虑的无失真有线网络单播传输,至少需要两个单播连接网络编码才能够带来吞吐量增益。在 2.3 节中我们将详细证明,网络编码在无失真有线网络仅存在一个单播连接下不能带来吞吐量增益。

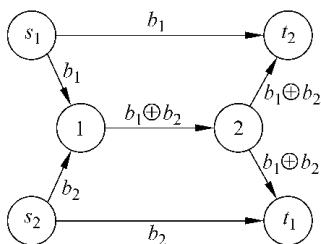


图 1.2 改变式蝴蝶网。该网络中,各条弧代表能够可靠传输一个包的有向链路。源节点  $s_1$  有一个包  $b_1$  需要传输给信宿节点  $t_1$ ,而源节点  $s_2$  有一个包  $b_2$  需要传输给信宿节点  $t_2$ 。

网络编码也可以扩展到无线网络。无线网络中,更容易找到例子说明网络编码较之传统路由会带来吞吐量增益。确实,蝴蝶网所对应的无线网络(如图 1.3 所示)和改变式无线蝴蝶网(如图 1.4 所示)分别只涉及 6 个以及 3 个节点,相比于原来的 7 个和 6 个要少一些。和前述一样,这些例子说明使用路由方式无法达到所需通信目标,而采用编码方式能够实现。和有线网络不同的地方在于,这些无线例子中不再假设从单一节点到另一单一节点的包传输,而是允许从单一节点到多个节点的包传输。因而,我们不再使用弧表示传输,而使用超弧表示之,这里超弧是更广义的弧,可以有多个信宿节点。

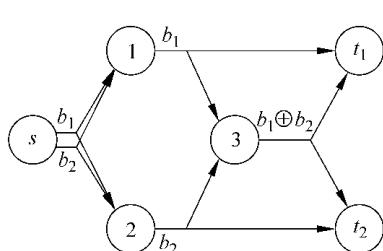


图 1.3 无线蝴蝶网。该网络中,各超弧代表能够可靠传输一个包到一个或者多个节点的有向链路。信源节点  $s$  有两个包  $b_1$  和  $b_2$  需要同时传输给信宿节点  $t_1$  和  $t_2$ 。

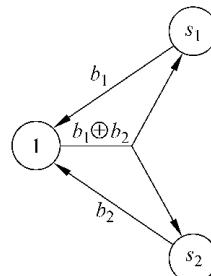


图 1.4 改变式无线蝴蝶网。该网络中,各超弧代表能够可靠传输一个包到一个或者多个节点的有向链路。信源节点  $s_1$  有一个包  $b_1$  需要传输给节点  $s_2$ ,而信源节点  $s_2$  有一个包  $b_2$  需要传输给节点  $s_1$ 。

到目前为止,我们已经讨论的例子说明,即使在没有丢失和误码的情况下,只要应用于一个或多个同时传输的多播连接或者两个或多个同时传输的单播连接,网络编码就能

够带来吞吐量增益。该结论对信息包从一个单一节点传输到另一单一节点(有线网络)或者从一个单一节点传输到一个或多个节点(无线网络)都成立。然而,这些例子看起来像是设计的微型示例,不具备一般性,所以很容易使人产生疑问,网络编码是否可以推广呢?即使可以,又会是什么情况呢?本书后续大部分内容将致力于推广前述结论到更一般的情况。

## 1.2.2 鲁棒性

### 1.2.2.1 丢包鲁棒性

在进一步论述之前,先强调分组网络尤其是无线分组网络中一个到目前为止我们一直忽略的重要问题——丢包。丢包由网络中的多种原因造成,包括缓存溢出、链路中断和冲突。处理这些丢包的方法很多。其中,传输控制协议(TCP)使用的机制可能是最直接的。TCP中设置应答(Ack)系统,信宿节点接收到数据包后传输一个Ack信息给信源节点,如果信源节点没有收到某特定包的Ack信息,则重传该数据包。另一个有时使用的可选方法是信道编码,或者更确切的说为纠删码。信源节点运用纠删码对数据包进行一定程度的冗余,这样即使信宿节点仅接收到信源节点发送的部分数据包,也能恢复出全部信息。

纠删码是信源节点采用的编码。中间节点怎样编码呢?即如何进行网络编码呢?网络编码是否可以帮助对抗丢包呢?确实可以。下面通过一个简单的例子说明其原因。考虑简单双链路序列网络,如图1.5所示。该网络中,节点1到节点2链路的丢包率为 $\epsilon_{12}$ ,节点2到节点3链路的丢包率为 $\epsilon_{23}$ 。节点1使用纠删码后,可以以每单位时间 $(1-\epsilon_{12})(1-\epsilon_{23})$ 个数据包的速率进行通信。本

质上,节点1到节点3之间为一个删除概率为 $1-(1-\epsilon_{12})(1-\epsilon_{23})$ 的删除信道,其容量为 $(1-\epsilon_{12})(1-\epsilon_{23})$ ,可由一个适当设计的编码达到。然而,系统的真正容量可以更大。如果在节点1和节点2之间的链路上采用一个纠删码,在节点2和节点3之间的链路上采用另一个纠删码,即节点2分两步进行完全译码和再编码,则节点1和节点2之间可以以每单位时间 $(1-\epsilon_{12})$ 个数据包的速率进行通信,节点2和节点3之间可以以每单位时间 $(1-\epsilon_{23})$ 个数据包的速率进行通信。因此,节点1和节点3之间可以以 $\min(1-\epsilon_{12}, 1-\epsilon_{23})$ 的速率进行通信,这通常比 $(1-\epsilon_{12})(1-\epsilon_{23})$ 要大。

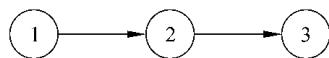


图 1.5 双链路序列网络。节点1和节点2分别可以在其输出链路上每单位时间传输1个包。

那么为何分组网络中不使用这种解决方案呢？关键的原因是延迟。无论使用分组码或循环码，每一步纠删码都将产生一定的延迟，因为每步译码器都需要接收到一定数量的包后才能开始译码。这样，如果每条链路或者连接均采用纠删码，将带来巨大的总延迟。然而，使用纠删码这样的额外步骤仅仅是一种特殊形式的网络编码——这里中间节点进行了编码。因此，网络编码可以用于提高对丢包的鲁棒性，从而可以转化为吞吐量增益。不过，我们需要的网络编码解决方案不仅仅能够提高吞吐量，而且要超越单单使用纠删码的额外步骤，即中间节点采用额外编码且不进行译码。本书第4章将讨论随机线性网络编码如何满足这种编码方案的要求。

丢包在网络编码问题中增加了一个额外的因素。丢包存在的情况下，即使仅有一个单播链路也可以获得增益。丢包与无线网络非常相关，并且考虑丢包因素使得网络编码与无线应用联系更紧密。我们讨论过无线网络的另一个特性是存在广播链路，即链路可以传输到一个以上的终端节点，因此还需要将丢包和广播链路结合起来。

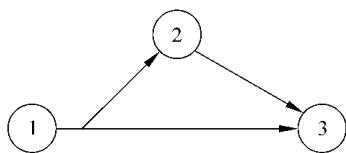


图 1.6 分组中继信道。节点 1 和节点 2 分别可以在其输出链路上单位时间内传输 1 个包。

图 1.6 中给出了一种改进的双链路序列网络，我们称之为分组中继信道。这里，节点 1 的输出链路不仅到达节点 2，而且到达节点 3。然而，由于存在丢包，节点 1 发出的包是否节点 2 和节点 3 都未接收、仅节点 2 接收、仅节点 3 接收或者节点 2 和节点 3 都接收是由概率分布决定的。现在，假定节点 1 传输的包仅被节点 2 接收的概率为  $p_{1(2)2}$ ，仅被节点 3 接收的概率为  $p_{1(2)3}$ ，被节点 2 和节点 3 同时接收的概率为  $p_{1(2)(23)}$ （全部丢失的概率为  $1 - p_{1(2)2} - p_{1(2)3} - p_{1(2)(23)}$ ）。对节点 2 的包传输，假定节点 2 传输的包被节点 3 接收的概率为  $p_{233}$ （彻底丢失的概率为  $1 - p_{233}$ ）。该设置下，网络编码，特别是随机线性网络编码能够实现最大可达吞吐量，即最小割容量，为  $\min(p_{1(2)2} + p_{1(2)3} + p_{1(2)(23)}, p_{1(2)3} + p_{1(2)(23)} + p_{233})$ 。

这是相当好的结论：首先，从网络信息论的观点，是否存在一种简单的、可达容量的网络编码甚至还不清楚；其次，对无线分组网络而言，它提供了主流方法的一个重大替代。主流的路由方法提倡将无线分组网视为有线分组网的扩展。因而，它提倡经路由传输信息；即在前述例子中，要么节点 1 传输给节点 2 后再传给节点 3，要么直接从节点 1 传输至节点 3，要么使用更复杂的方案将前面两种方式结合。采用网络编码后，不再有这样的路径概念，即节点对特定连接提供传输，但这些节点不一定必须沿着一条路径。因

此,有必要重新思考路由。这种思考带来了子图选择,我们将在第 5 章讨论。

### 1.2.2.2 链路中断鲁棒性

除了具备对随机丢包的鲁棒性,网络编码还可以用于对非遍历链路中断的保护。实时路径保护对每个连接提供主备流传输,这样不需要重新路由,从而可以快速恢复链路中断。然而,这种方法使得网络流量倍增。网络编码通过在不同流之间共享网络资源,能够提高资源利用率。任一多播会话下对可通过任意重新路由机制恢复的任何中断模式集合,总是存在一种静态网络编码方案,使得该集合中的任意中断模式均不使用重新路由即可恢复<sup>[82]</sup>。

### 1.2.3 复杂度

某些情况下,虽然最优路由可能达到近似于网络编码的性能,然而最优路由解却很难获得。比如,多播路由的最小代价子图选择涉及 Steiner 树,该问题即使在集中设置下也很复杂,而相应的网络编码问题则仅是线性最优化问题,它可生成低复杂度分布式解。这些将在 5.1.1 节中进一步讨论。

有些场景下,受实际限制必须选择次优方案,例如,基于闲谈的数据分发<sup>[32]</sup>以及 802.11 无线自组织网络<sup>[74]</sup>。经论证,网络编码在这些情况下也能够大幅提高性能。这些将在 3.5.2.2 节讨论。

### 1.2.4 安全

从安全性角度看,网络编码既有利处也有弊端,以蝴蝶网(如图 1.1 所示)为例说明。假如对手仅仅获取了包  $b_1 \oplus b_2$ ,单单只有包  $b_1 \oplus b_2$ ,对手不能获取  $b_1$  或  $b_2$ 。因此,我们具备安全通信的可能方案。这个例子说明网络编码能够提供安全增益。

另一方面,假设节点 3 为恶意节点,不传送真实的包  $b_1 \oplus b_2$ ,而是传送伪装成  $b_1 \oplus b_2$  的包。由于数据包是由编码而非路由得到,对数据包的篡改很难发现。这个例子说明网络编码带来了潜在的安全隐患。后续将在第 6 章中讨论网络编码在安全方面的应用。

目前,我们已经给出了一些微型示例以说明网络编码带来的好处。这些例子很明显与分组网络相关,而它们说明的原理在实际情况下如何使用则可能不同。下一节中我们将给出模型说明更一般的情况。

## 1.3 网络模型

分组网络,尤其是无线分组网络非常复杂,因此很难精确建模。此外,网络编码应用于如此宽泛、多种多样的背景下,使用统一模型也是不合理的。不过,我们使用的所有模型也有共同之处,现在进行讨论。至于各种模型的特殊方面,后面用到的时候再进行讨论。

作为建模的出发点,我们假设需要建立许多连接或者会话。这些连接可能是单播(一个信源节点和一个信宿节点)或者多播(一个信源节点和多个信宿节点)。在多播连接中,所有的信宿节点期望从信源节点获得相同消息。这些连接与我们希望通信的包相关,而包速率是已知或未知的。因此,我们的模型忽略拥塞控制,即模型中不考虑连接速率的控制。我们认为拥塞控制是一个独立的问题,本书中不详细讨论,但它易于融合进模型中(参见文献[25,135])。

我们使用有向超图  $H = (N, A)$  描述网络拓扑,其中  $N$  为节点集合,  $A$  为超弧集合。超图为广义图,其中不再是弧,而是超弧。一个超弧为一个节点对  $(i, J)$ , 其中出发节点  $i$  是  $N$  中的一个元素,信宿节点集合  $J$  为  $N$  的非空子集。每一条超弧  $(i, J)$  表示一个从节点  $i$  到非空集合  $J$  中节点的广播链路。特殊情况下,  $J$  仅包含一个元素  $j$ , 则退化为点到点链路。此时,超弧即为简单的弧,有时记作  $(i, j)$  替代  $(i, \{j\})$ 。如果网络仅仅包含点到点链路(比如有线网络),那么  $H$  变为图,记为  $G$  替代  $H$ 。超弧  $(i, J)$  表示的链路可以是无失真或者有失真的,即它可以不受丢包或者受丢包影响。

为建立一个或多个所需连接,分包在超弧上传输。假定  $z_{ij}$  为包注入到超弧  $(i, J)$  上的平均速率。定义向量  $z$ ,由  $z_{ij}$  组成,其中  $(i, J) \in A$ 。向量  $z$  表示网络中所有超弧上包注入速率。这种抽象模型下,我们没有明确说明任何队列。假定排队发生在抽象模型隐藏下的一层。假定向量  $z$  在某约束集合  $Z$  以内,网络中所有的队列均是稳定的。有线网络中,链路通常是独立的,约束集合  $Z$  分解为  $|A|$  个约束的笛卡儿积。无线网络中,链路通常是不独立的,集合  $Z$  的形式可能很复杂(例如参见文献[28,72,73,79,137,141])。暂时,对集合  $Z$ ,除了它是一个正象限下并包括原点的凸子集,我们没有作任何假设。

集合对  $(H, Z)$  定义了一个约束图,表示我们要处理的网络。该网络可以是一个完整的物理网络,也可以是物理网络的子网。这样向量  $z$  可以视为该约束图的子集——其为

实际使用的一部分——在所需的一个或多个连接下称为编码子图。假定编码子图不仅定义超弧上的包注入速率，而且还指定了这些注入发生的时间。因此，路由和调度这些经典网络问题成为编码子图选择问题的特定子问题。

前一节中讨论的示例已经给出了编码子图，并明确了包注入情况，剩下的任务则是尽可能高效地使用它们。无失真网络中表现编码子图最简单的方式可能为，使用不同的超弧描述特定时间周期内的每一个包传输，和我们在图 1.1~图 1.4 所设置的一样。其中，可以设置并行超弧，如图 1.3 所示（存在两条超弧 $(s, \{1, 2\})$ ），表示同样的节点在单一时间周期内同时接收和发送多个包。此外，图 1.7 给出了节点编码。我们称这种子图表示方式为静态子图。静态子图中，不明确表示时间，假定事件即时发生。实际上，包在链路上的传输总会存在一些延迟，因此链路上的包输出会比其输入晚。所以，静态子图隐藏了时间细节，这些细节虽然不难解决，但是必须注意。此外，这里只考虑无环图，因为有环图会引起包即时反馈。尽管静态子图存在一些局限，但它足够我们开展许多希望的讨论。在第 2 章处理无失真网络中将或多或少地专门使用它。

对有失真网络而言，时间的因素将变得越发重要。有失真网络下使用的网络编码通常比无失真网络中的长很多，即一个编码块中包含更多的信源包。换个角度看，有失真网络中网络编码必须考虑的时间周期比无失真网络中长很多。因此，必须正视编码节点

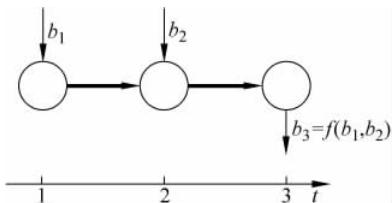


图 1.8 时间扩展子图中的节点编码。包  $b_1$  在时刻 1 被接收，包  $b_2$  在时刻 2 被接收。水平加粗弧具备无穷容量，表示一个节点存储的数据。因此，在时刻 3，包  $b_1$  和  $b_2$  可用于形成包  $b_3$ 。

处编码与时间的相互影响。为了分析该问题，我们将静态子图拓展到时间扩展子图。

时间扩展子图不仅表示了包的注入和接收点，而且还说明了这些注入和接收发生的时刻。我们仅仅描绘成功接收部分；因此，有失真网络中，时间扩展子图实际上表示了编码子图的随机总体中的一个特定元素。以图 1.7 为例，包  $b_1$  在时刻 1 被接收，包  $b_2$  在时刻 2 被接收，包  $b_3$  在时刻 3 被接收。采用时间扩展子图表述这些注入和接收，如图 1.8 所示。这个例子

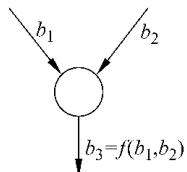


图 1.7 静态子图中的节点编码。两输入弧分别传输包  $b_1$  和  $b_2$ ，输出弧传输的包为  $b_1$  和  $b_2$  的一个函数。

中使用了整数时刻值,而实数时刻值也能被很容易地应用。现在,有了同一节点的多个举例,每个举例表示不同时刻下的节点。融合这些举例则为时间长河上的无穷容量链路,表示节点对包的存储能力。第4章中采用时间扩展子图分析有失真网络。

## 1.4 本书概要

第2章以网络编码理论起源设置开端:无失真有线网络多播传输。该设置下,说明网络编码应用的主要例子是蝴蝶网(如图1.1所示)。第2章中,我们将把从该例子得到的结论扩展到一般拓扑。这里给出了多播问题下的网络编码容量,并探讨了达到该容量的确定或随机编码结构。理论分析中,我们研究静态子图。正如前面已经说明的那样,静态子图隐藏了时间细节。因此,如何将该理论应用于实际分组网络不是很显然的,动态行为常常很重要。在2.5节中明确讨论分组网络,并论证随机网络编码如何可以自然地应用于动态设置。这里讨论的策略将在第4章中回顾。最后,我们以无失真多播基础理论的另外两个拓展结束全章:有环网络情况(回顾静态子网中的即时反馈因素)以及相关信源过程情况。

第3章扩展至无失真有线网络中的非多播问题,也就是,我们研究希望建立多个连接的情况,并对来自于不同连接的包一起进行编码。这和改变式蝴蝶网所示情况一样(如图1.2所示)。我们称这种编码为会话间编码。这种情况和每个连接均保持独立的情况(比如第2章中我们仅讨论一个单独连接)形成对比,后者称为会话内编码。我们证明线性网络编码通常不足以达到容量,可能需要非线性网络编码。考虑到构建非线性结构编码的困难度,这里对会话间编码构造的研究通常集中在可带来很大性能提升的次优方法上。第3章中给出一些这样的方法。

第4章研究有失真网络。我们论证随机线性网络编码如何应用于有失真网络,比如分组中继信道(如图1.6所示),并且对这种网络下的单播或多播连接给出一个容量可达策略。同时,我们还推导出一个错误幂指数,用以定量描述错误概率随编码延迟衰退的速率。

第2~4章中均假设编码子图已经定义。在第5章中,将讨论选择合适编码子图的问题。给出两种方法:其一为基于流的方法,这里我们假定以某给定流速率建立连接为通信目标;其二为基于队长的方法,这里流速率虽存在但未知。本章主要分析会话内编