

并行计算

Parallel Computing

主讲人 孙广中

Spring, 2018

并行计算——结构·算法·编程

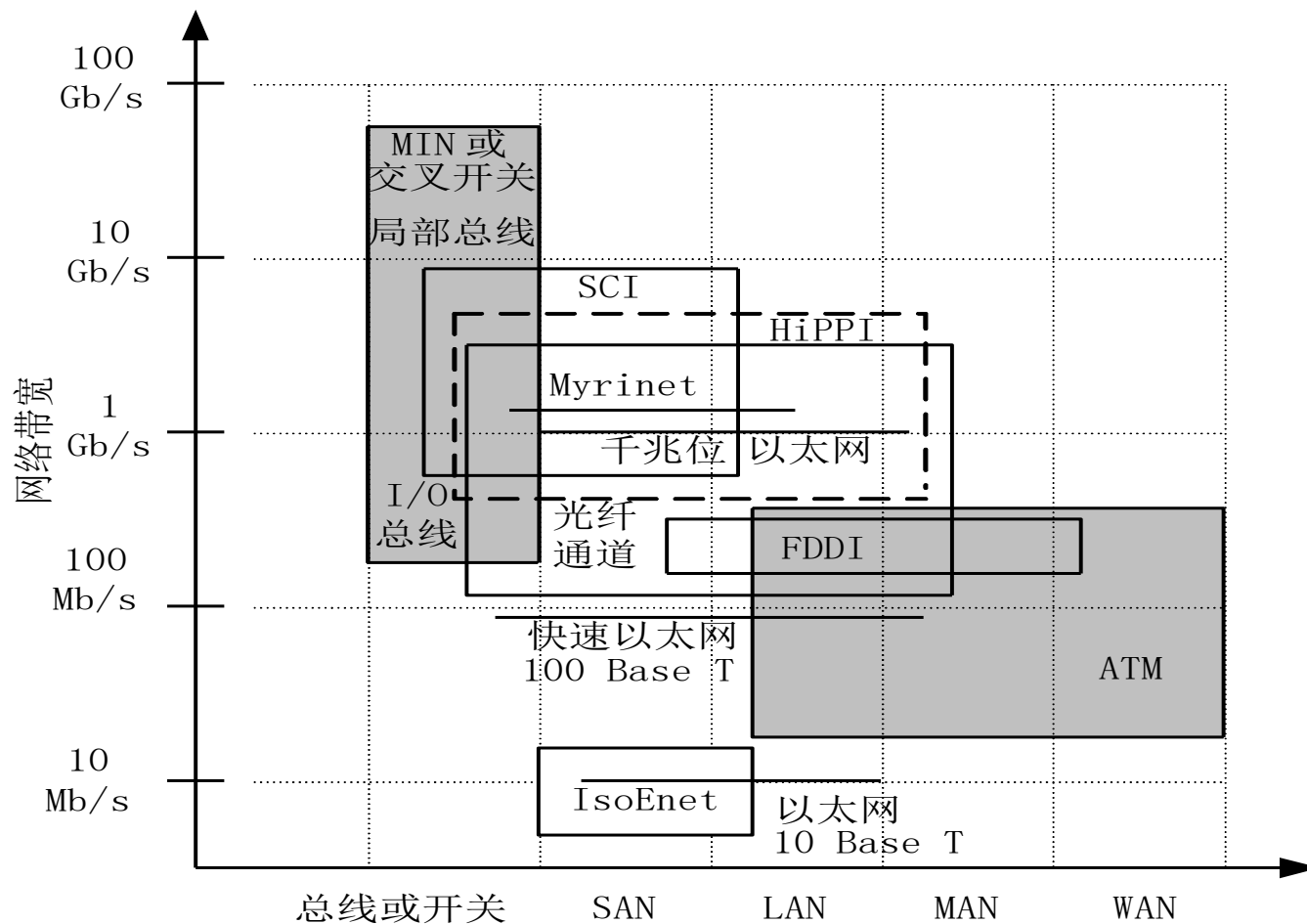
- 第一篇 并行计算的基础
 - 第一章 并行计算与并行计算机结构模型
 - 第二章 并行计算机系统互连与基本通信操作
 - 第三章 典型并行计算机系统介绍
 - 第四章 并行计算性能评测

第二章 并行机系统互连与基本通信操作

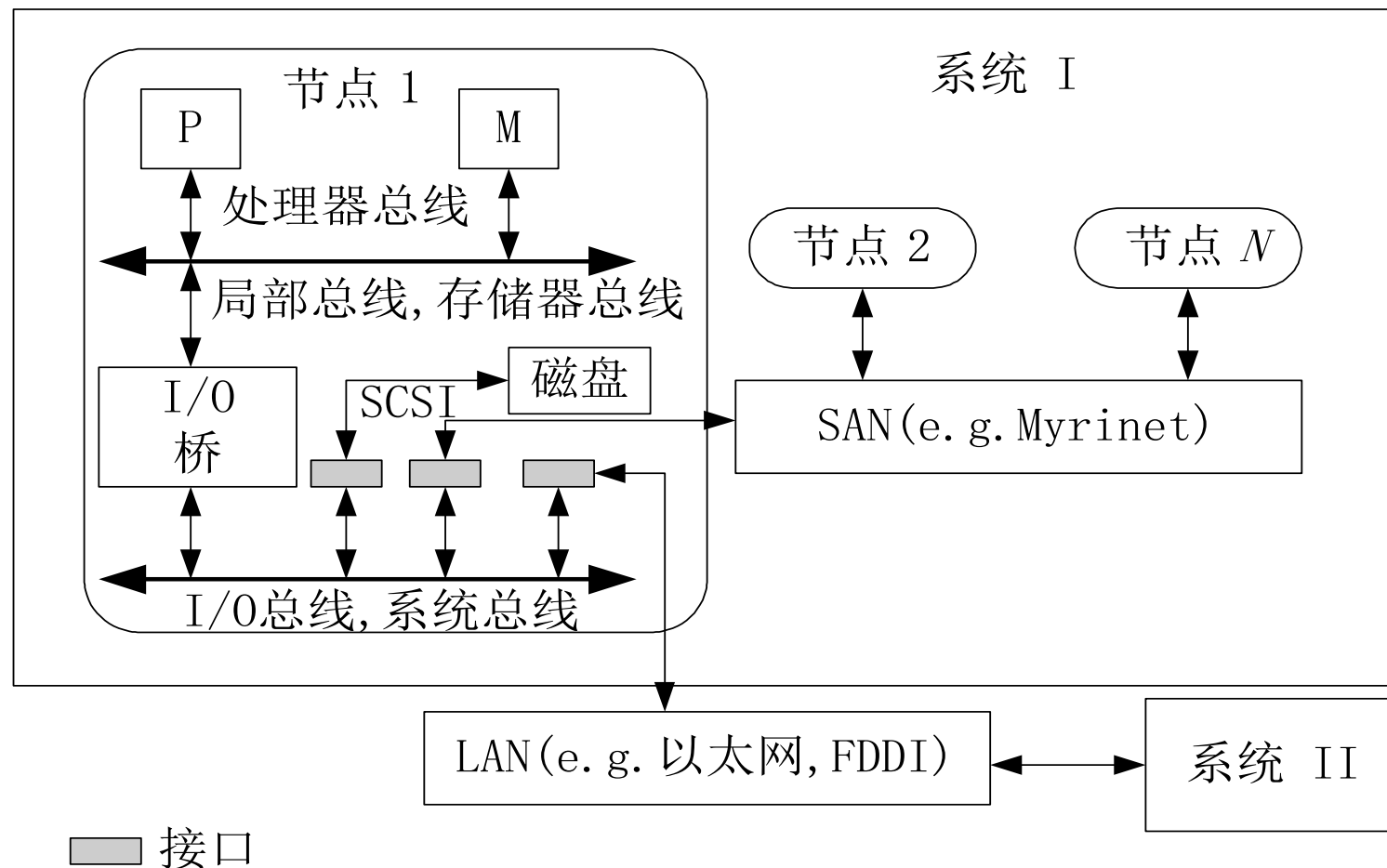
- 2.1 并行计算机互连网络
 - 2.1.1 系统互连
 - 2.1.2 静态互连网络
 - 2.1.3 动态互连网络
 - 2.1.4 标准互连网络
- 2.2 选路方法与开关技术
- 2.3 单一信包一到一传输
- 2.4 一到多播送
- 2.5 多到多播送

系统互连

- 不同带宽与距离的互连技术：
总线、SAN、LAN、MAN、WAN



局部总线、I/O总线、SAN和LAN



网络性能指标

- **节点度** (Node Degree) : 射入或射出一个节点的边数。在单向网络中, 入射和出射边之和称为节点度。
- **网络直径** (Network Diameter) : 网络中任何两个节点之间的最长距离, 即最大路径数。
- **对剖宽度** (Bisection Width) : 对分网络各半所必须移去的最少边数
- **对剖带宽** (Bisection Bandwidth) : 每秒钟内, 在最小的对剖平面上通过所有连线的最大信息位 (或字节) 数
- 如果从任一节点观看网络都一样, 则称网络为**对称的** (Symmetry)

静态互连网络 与动态互连网络

- **静态互连网络**：处理单元间有着固定连接的一类网络，在程序执行期间，这种点到点的链接保持不变；典型的静态网络有一维线性阵列、二维网孔、树连接、超立方网络、立方环、洗牌交换网、蝶形网络等
- **动态网络**：用交换开关构成的，可按应用程序的要求动态地改变连接组态；典型的动态网络包括总线、交叉开关和多级互连网络等。

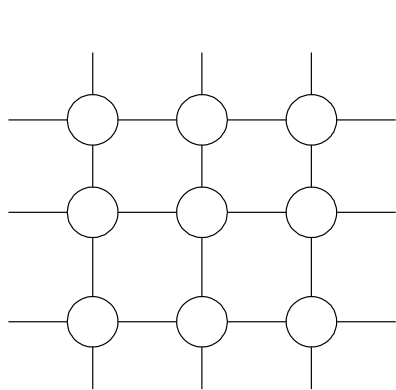
静态互连网络（1）

- 一维线性阵列（1-D Linear Array）：
 - 并行机中最简单、最基本的互连方式，
 - 每个节点只与其左、右近邻相连，也叫二近邻连接，
 - N 个节点用 $N-1$ 条边串接之，内节点度为2，直径为 $N-1$ ，对剖宽度为1
 - 当首、尾节点相连时可构成循环移位器，在拓扑结构上等同于环，环可以是单向的或双向的，其节点度恒为2，直径或为 $\lfloor N/2 \rfloor$ （双向环）或为 $N-1$ （单向环），对剖宽度为2

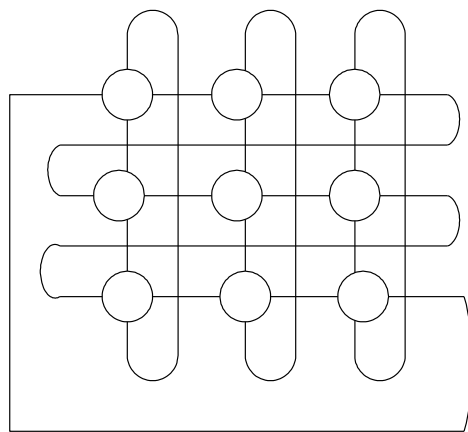
$$\lfloor N/2 \rfloor$$

静态互连网络（2）

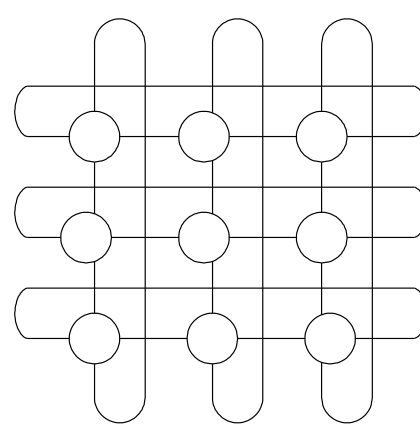
- $\sqrt{N} \times \sqrt{N}$ 二维网孔（2-D Mesh）：
 - 每个节点只与其上、下、左、右的近邻相连（边界节点除外），节点度为4，网络直径为 $2(\sqrt{N}-1)$ ，对剖宽度为 \sqrt{N}
 - 在垂直方向上带环绕，水平方向呈蛇状，就变成Illiac网孔了，节点度恒为4，网络直径为 $\sqrt{N}-1$ ，而对剖宽度为 $2\sqrt{N}$
 - 垂直和水平方向均带环绕，则变成了2-D环绕（2-D Torus），节点度恒为4，网络直径为 $2\sqrt{N}/2$ ，对剖宽度为 $2\sqrt{N}$



(a) 2-D网孔



(b) Illiac网孔

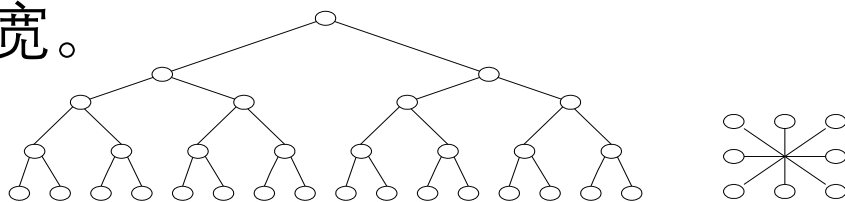


(c) 2-D环绕

静态互连网络 (3)

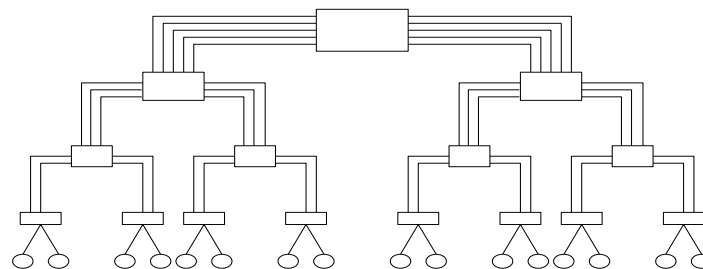
- 二叉树：

- 除了根、叶节点，每个内节点只与其父节点和两个子节点相连。
- 节点度为3，对剖宽度为1，而树的直径为 $2(\lceil \log N \rceil - 1)$
- 如果尽量增大节点度数 $\lfloor \sqrt{N} \rfloor$ ，则直径缩小为2，此时就变成了星形网络，其对剖宽度为
- 传统二叉树的主要问题是根易成为通信瓶颈。胖树节点间的通路自叶向根逐渐变宽。



(a) 二叉树

(b) 星形连接

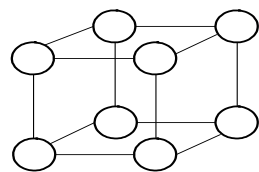


(c) 二叉胖树

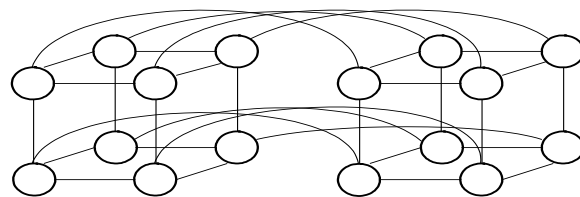
静态互连网络（4）

- 超立方：

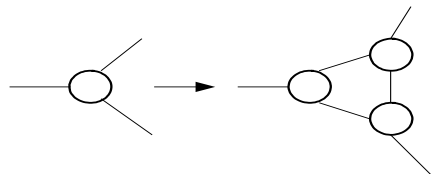
- 一个 n -立方由 $N = 2^n$ 个顶点组成，3-立方如图(a)所示；4-立方如图(b)所示，由两个3-立方的对应顶点连接而成。
- n -立方的节点度为 n ，网络直径也是 n ，而对剖宽度为 $N/2$ 。
- 如果将3-立方的每个顶点代之以一个环就构成了如图(d)所示的3-立方环，此时每个顶点的度为3，而不像超立方那样节点度为 n 。



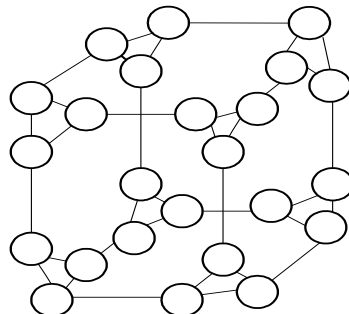
(a) 3-立方



(b) 4-立方



(c) 顶点代之以环

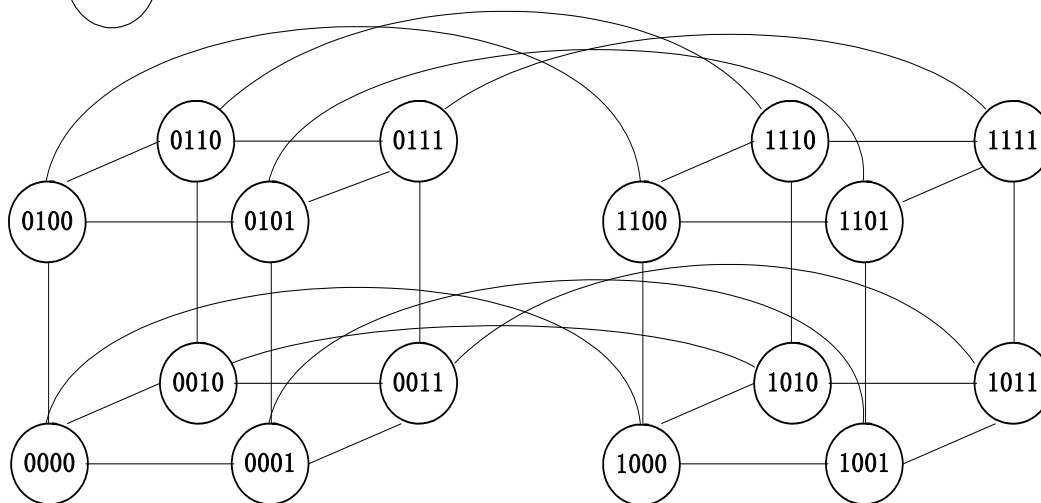
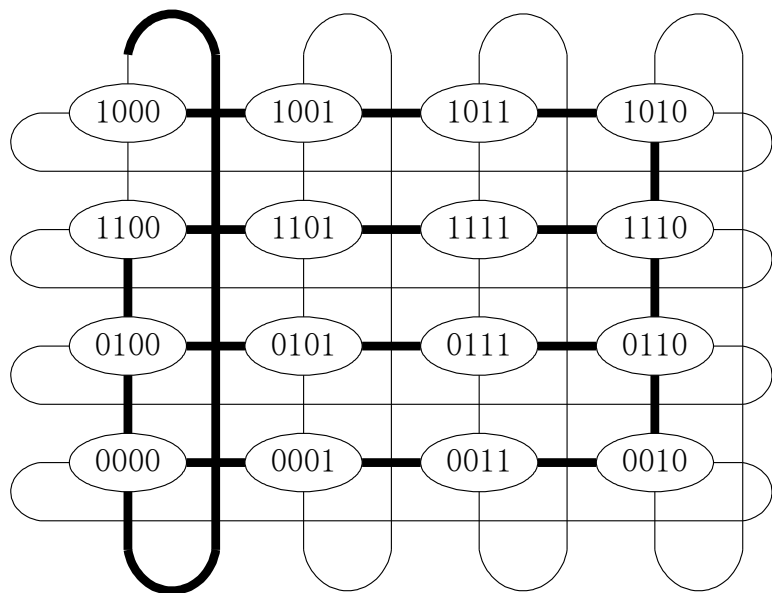


(d) 3-立方环

嵌入 (1)

- 将网络中的各节点映射到另一个网络中去
- 用**膨胀 (Dilation) 系数**来描述嵌入的质量，它是指被嵌入网络中的一条链路在所要嵌入的网络中对应所需的最大链路数
- 如果该系数为1，则称为完美嵌入。
- 环网可完美嵌入到2-D环绕网中
- 超立方网可完美嵌入到2-D环绕网中

嵌入 (2)

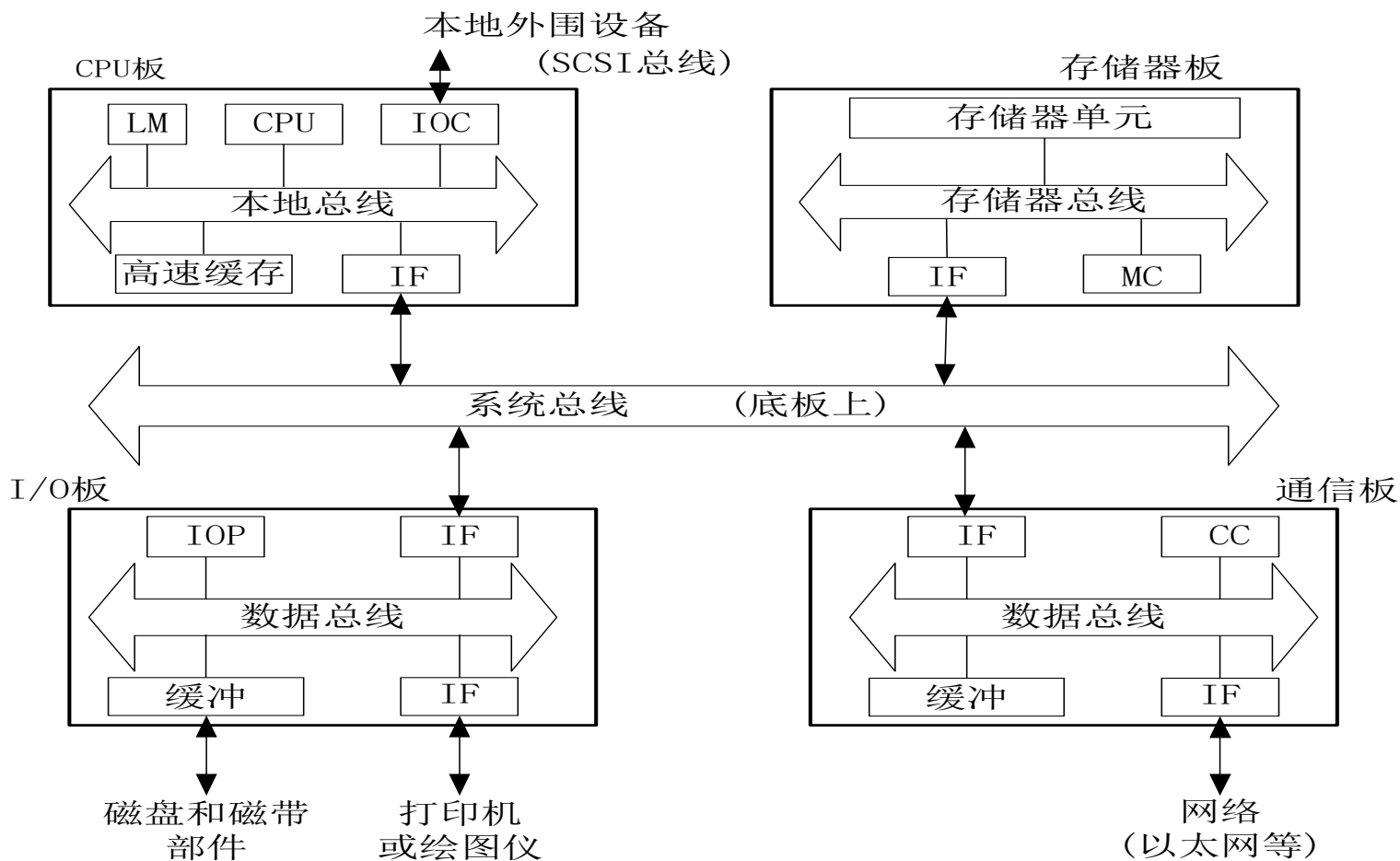


静态互连网络特性比较

网络名称	网络规模	节点度	网络直径	对剖宽度	对称	链路数
线性阵列	N	2	$N - 1$	1	非	$N - 1$
环形	N	2	(双 $\lfloor N/2 \rfloor$ 向)	2	是	N
2-D网孔	$(\sqrt{N} \times \sqrt{N})$	4	$2(\sqrt{N} - 1)$	\sqrt{N}	非	$2(N - \sqrt{N})$
Illiac网孔	$(\sqrt{N} \times \sqrt{N})$	4	$\sqrt{N} - 1$	$2\sqrt{N}$	非	$2N$
2-D环绕	$(\sqrt{N} \times \sqrt{N})$	4	$2\lfloor \sqrt{N}/2 \rfloor$	$2\sqrt{N}$	是	$2N$
二叉树	N	3	$2(\lceil \log N \rceil - 1)$	1	非	$N - 1$
星形	N	$N - 1$	2	$\lfloor N/2 \rfloor$	非	$N - 1$
超立方	$N = 2^n$	n	n	$N/2$	是	$nN/2$
立方环	$N = k \cdot 2^k$	3	$2k - 1 + \lfloor k/2 \rfloor$	$N/(2k)$	是	$3N/2$

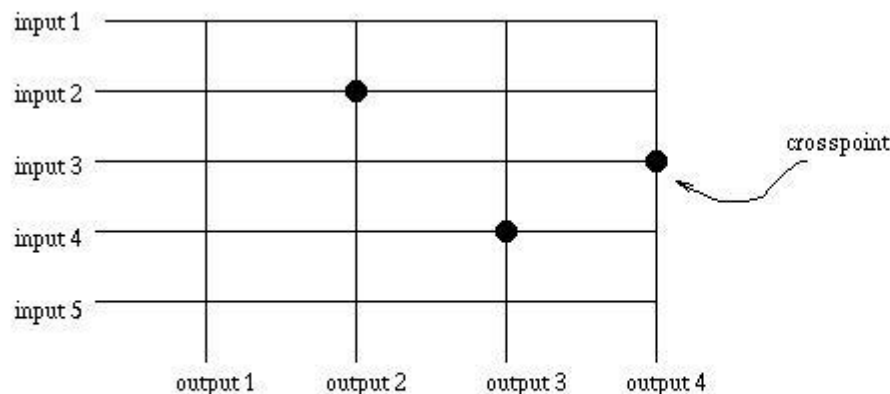
动态互连网络 (1)

- 总线：PCI、VME、Multibus、Sbus、MicroChannel
 - 多处理机总线系统的主要问题包括总线仲裁、中断处理、协议转换、快速同步、高速缓存一致性协议、分事务、总线桥和层次总线扩展等



动态互连网络（2）

- 交叉开关（Crossbar）：
 - 单级交换网络，可为每个端口提供更高的带宽。象电话交换机一样，交叉点开关可由程序控制动态设置其处于“开”或“关”状态，而能提供所有（源、目的）对之间的动态连接。
 - 交叉开关一般有两种使用方式：一种是用于对称的多处理机或多计算机机群中的处理器间的通信；另一种是用于SMP服务器或向量超级计算机中处理器和存储器之间的存取。



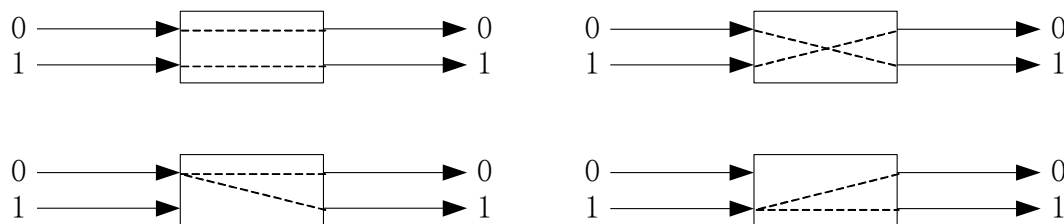
Input 2 is connected to output 2,
input 3 is connected to output 4,
input 4 is connected to output 3.

Note: Connections between free
inputs and outputs are always possible.

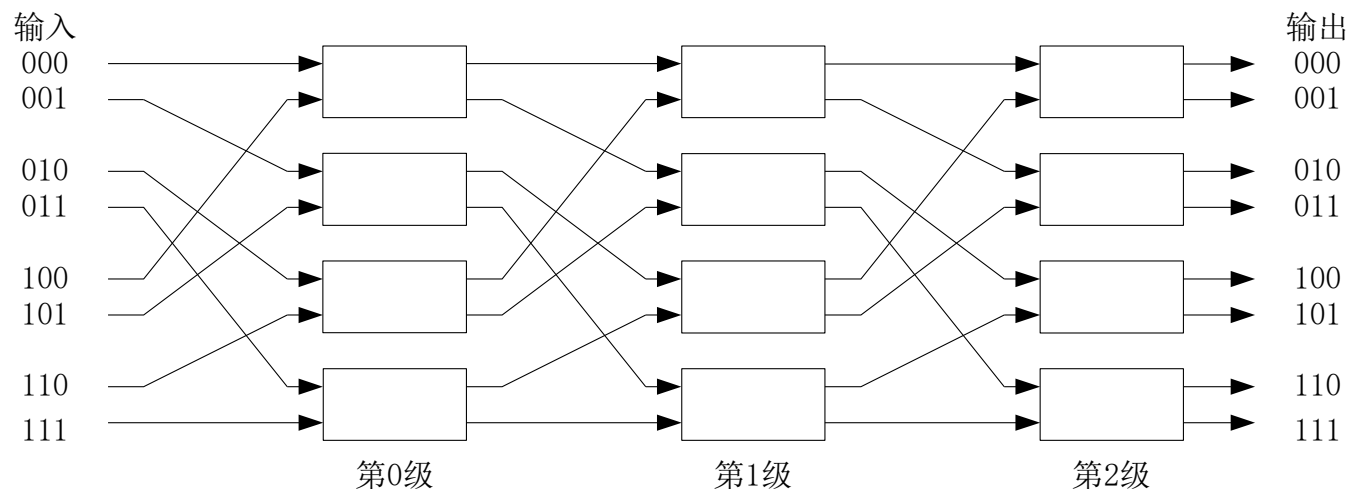
A Small Crossbar Switch

动态互连网络 (3)

- 单级交叉开关级联起来形成多级互连网络MIN (Multistage Interconnection Network)



(a) 4种可能的开关连接



(b) 一种8输入的Omega网络

动态互连网络（4）

- 交换开关模块：
 - 一个交换开关模块有 n 个输入和 n 个输出，每个输入可连接到任意输出端口，但只允许一对一或一对多的映射，不允许多对一的映射，因为这将发生输出冲突
- 级间互连（Interstage Connection）：
 - 均匀洗牌、蝶网、多路均匀洗牌、交叉开关、立方连接
 - n 输入的 Ω 网络需要 $\log_2 n$ 级 2×2 开关，在Illinois大学的Cedar[2]多处理机系统中采用了 Ω 网络
 - Cray Y/MP多级网络，该网络用来支持8个向量处理器和256个存储器模块之间的数据传输。网络能够避免8个处理器同时进行存储器存取时的冲突。

动态互连网络比较

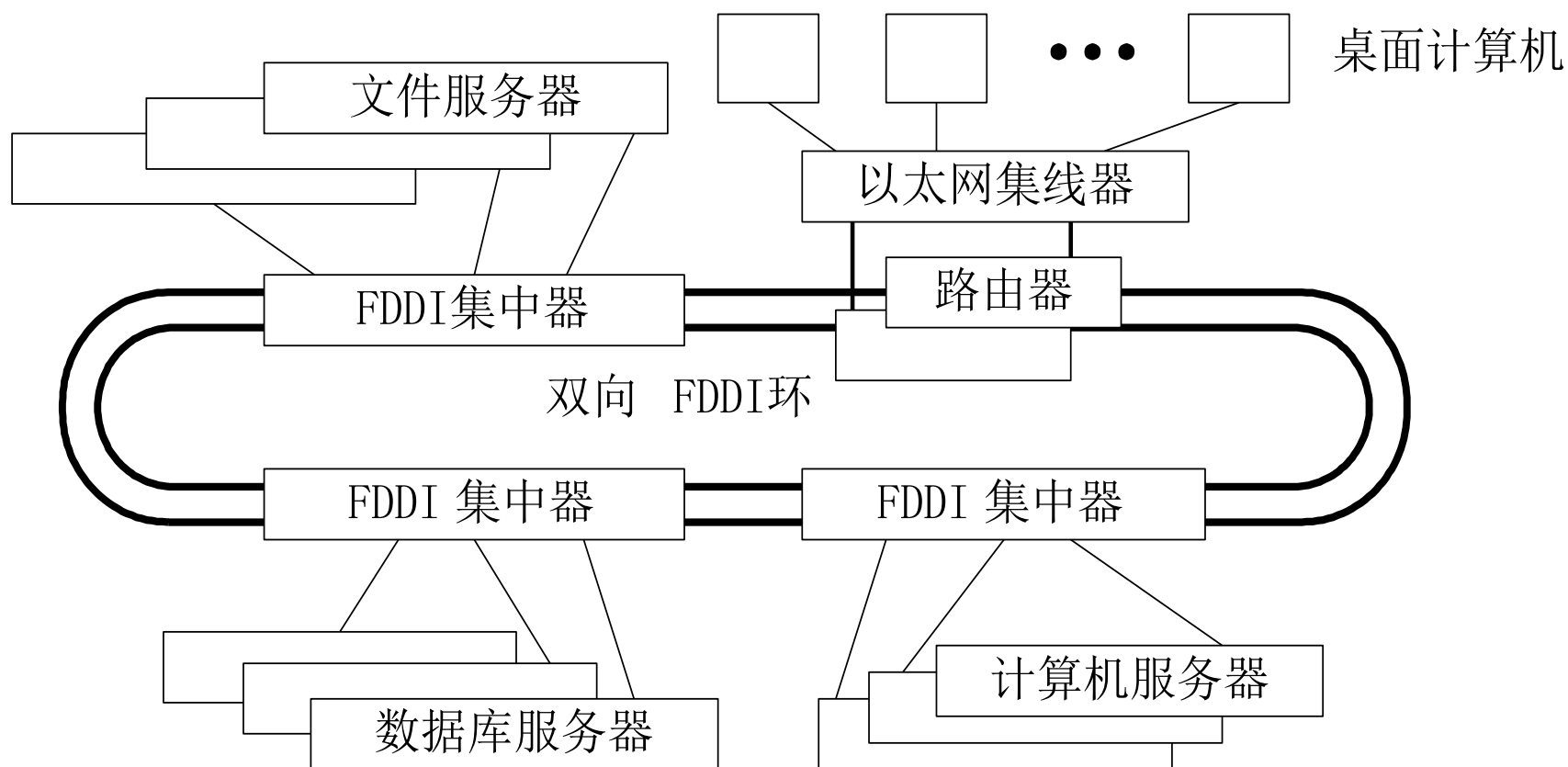
- n , 节点规模 w , 数据宽度

动态互连网络的复杂度和带宽性能一览表			
网络特性	总线系统	多级互连网络	交叉开关
硬件复杂度	$O(n + w)$	$O((n \log_k n)w)$	$O(n^2 w)$
每个处理器带宽	$O(wf/n) \sim O(wf)$	$O(wf)$	$O(wf)$
最小时延	恒定(轻负荷)	$O((n \log_k n))$	恒定

标准互连网络（1）

- 光纤通道FC（Fiber Channel）：
 - 通道和网络标准的集成
 - 光纤通道既可以是共享介质，也可以是一种交换技术
 - 光纤通道操作速度范围可从100到133、200、400和800Mbps。FCSI厂商也正在推出未来具有更高速度（1、2或4Gbps）的光纤通道
 - 光纤通道的价值已被现在的某些千兆位局域网所证实，这些局域网就是基于光纤通道技术的
 - 连网拓扑结构的灵活性是光纤通道的主要财富，它支持点到点、仲裁环及交换光纤连接
- FDDI：
 - 光纤分布式数据接口FDDI（Fiber Distributed Data Interface）
 - FDDI采用双向光纤令牌环可提供100-200Mbps数据传输速率
 - FDDI具有互连大量设备的能力
 - 传统的FDDI仅以异步方式操作

双向FDDI环作为主干网



标准互连网络（2）

- 快速以太网

已经历了4代：

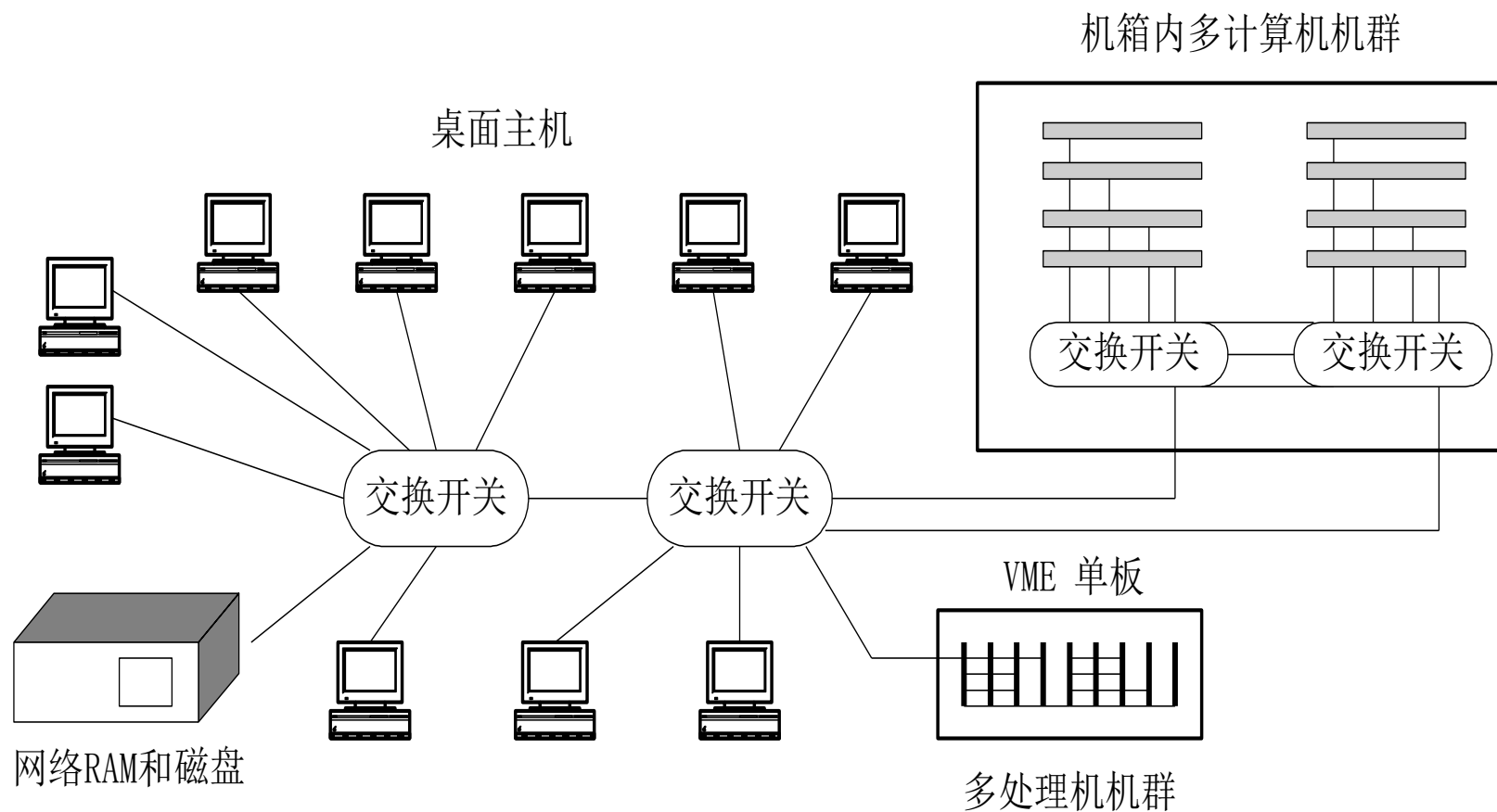
- 第一代，1982年引入的10Mbps
- 第二代，1994年宣布的100Mbps
- 第三代，1997年IEEE 802.3工作组宣布的1Gbps
- 第四代，2002年IEEE 802.3ae通过的10Gbps，并且2010年6月IEEE802.3ba公布了40-100Gbps

标准互连网络 (3)

- Myrinet:

- Myrinet是由Myricom公司设计的千兆位包交换网络，其目的是为了构筑计算机机群，使系统互连成为一种商业产品。
- Myrinet是基于加州理工学院开发的多计算机和VLSI技术以及在南加州大学开发的ATOMIC/LAN技术。Myrinet能假设任意拓扑结构，不必限定为开关网孔或任何规则的结构。
- Myrinet在数据链路层具有可变长的包格式，对每条链路施行流控制和错误控制，并使用切通选路法以及定制的可编程的主机接口。在物理层上，Myrinet网使用全双工SAN链路，最长可达3米，峰值速率为 $(1.28 + 1.28)$ Gbps (目前有 $2.56+2.56$)
- Myrinet交换开关 :8,12,16端口
- Myrinet主机接口 : 32位的称作LANai芯片的用户定制的VLSI处理器，它带有Myrinet接口、包接口、DMA引擎和快速静态随机存取存储器SRAM。
- 140 of the November 2002 TOP500 use Myrinet, including 15 of the top 100

Myrinet连接的LAN/Cluster



- InfiniBand

- Intel及IBM、Compaq、HP等分别提出了“下一代I/O”和“未来I/O”技术，这就是放弃了总线结构的新I/O规范：InfiniBand。
- 集合了整个业界的努力开发的，能够替代PCI总线的新标准。
- InfiniBand体系结构的第一个版本，于2000年10月发布
- 是一种交换式通信结构，包括分层结构、基于信息包的通信机制、3种连接速度等技术。
- 具体结构参见图2.16和图2.17.

标准互连网络（5）

代别 类型		以太网 10BaseT	快速以太网 100BaseT	千兆位以太网 1GB
引入年代		1982	1994	1997
速度（带宽）		10Mb/s	100Mb/s	1Gb/s
最大 距离	UTR（非屏蔽双扭对）	100m	100m	25 – 100m
	STP（屏蔽双扭对） 同轴电缆	500m	100m	25 – 100m
	多模光纤	2Km	412m（半双工） 2Km（全双工）	500m
	单模光纤	25Km	20Km	3Km
主要应用领域		文件共享， 打印机共享	COW计算， C/S结构， 大型数据库存取等	大型图像文件， 多媒体， 因特网， 内部网， 数据仓库等

第二章 并行机系统互连与基本通信操作

- 2.1 并行计算机互连网络
- 2.2 选路方法与开关技术
 - 预备知识
 - 2.2.1 选路方法
 - 2.2.2 开关技术
- 2.3 单一信包一到一传输
- 2.4 一到多播送
- 2.5 多到多播送

预备知识 (1)

- 选路(Routing)

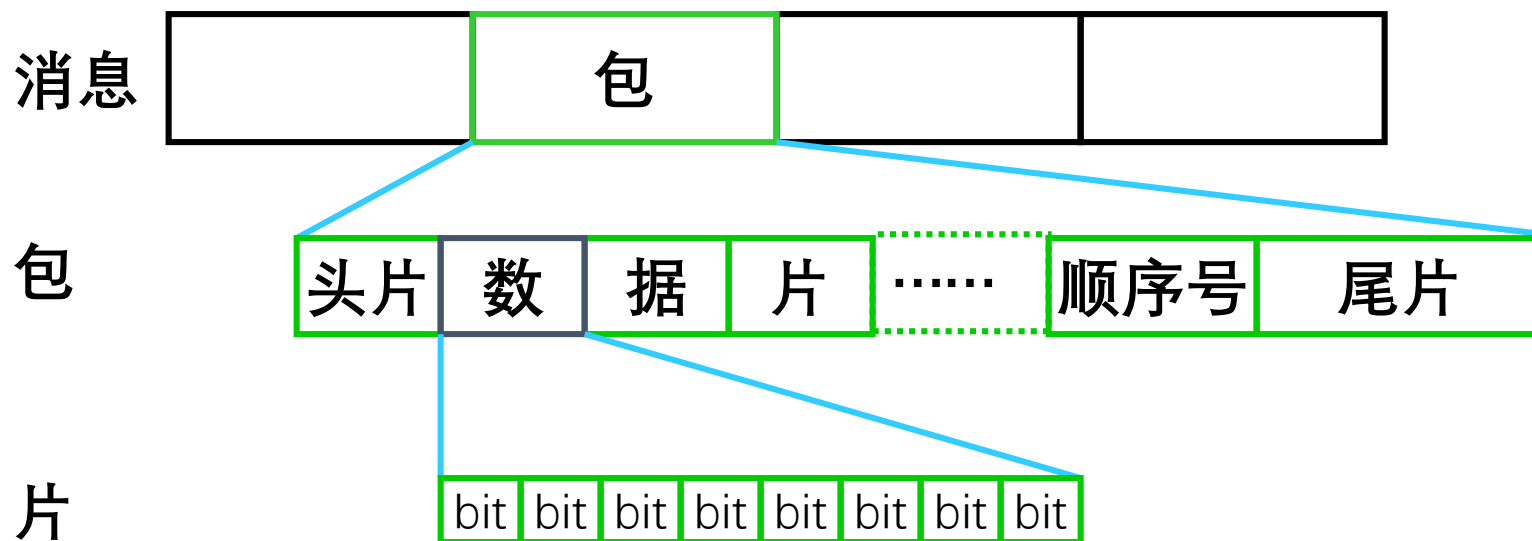
- 又称为选径或路由。产生消息从发源地到目的地所取的路径, 要求具有较低通讯延迟、无死锁和容错能力。应用于网络或并行机上的信息交换。

- 消息、信包、片

- 消息(Message) : 是在多计算机系统的处理接点之间传递包含数据和同步消息的信息包。它是一种逻辑单位, 可由任意数量的包构成。
- 包(Packet) : 包的长度随协议不同而不同, 它是信息传送的最小单位, 64-512位。
- 片(Flit) : 片的长度固定, 一般为8位。

预备知识 (2)

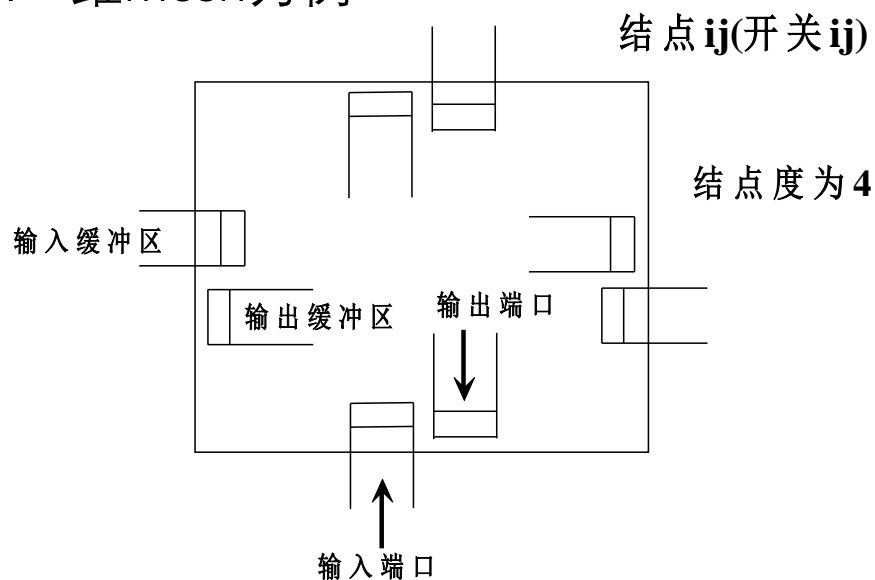
- 消息、信包、片的相互关系



预备知识 (3)

- 互连网络、传输节点结构

- (1)互连网络可以表示为一个图 $G(V,E)$, $V=\{\text{switches or nodes}\}$, $E \in V \times V$
- (2)描述: 拓扑(Topology)、选路算法(Routing)、流控制(Flow Control)
- (3)两个重要指标:传输时延(Transmission Latency)、吞吐量(Throughput)
- (4)节点(开关)结构:二维mesh为例



预备知识 (4)

• 一些术语

- 信道带宽 b ：每个信道有 w 位宽和信号传输率 $f = 1/t$ (t 是时钟周期), $b = wf \text{ bits/sec}$
- 节点和开关的度：与节点和开关相连的信道数目
- 路径：信包在网络中走过的开关和链路(link)序列
- 路由长度或距离：路由路径中包括的链路(link)数目

• 信包传输性能参数

- 启动时间 t_s (startup time)：准备信包头信息等
- 节点延迟时间 t_h (per-hop time)：信包头穿越相邻节点的时间
- 字传输时间 t_w (transfer time)：传输每个字的时间
- 链路数 l 、信包大小 m

预备知识（5）

- 选路算法的三种机制

- 基于算术的:

- 开关中具有简单的算术运算功能，如维序选路；

- 基于源地址的:

- 在源点时就将沿路径的各个开关的输出端口地址 p_0, p_1, \dots, p_n 包在信包的头部，每个开关只是对信包头的输出端口地址进行剥离；

- 基于查表的:

- 开关中含有一个选路表，对信包头中的选路域查出输出端口地址。

预备知识 (6)

• 选路方式

- 信包
 - 存储—转发 (*Store and Forward*)
 - 虫孔 (*Wormhole*)
 - 线路: 用交换机
- 静态: 在选路开始时所有的信息都已到达网络
 - 动态: 信包可在任意时刻到达网络
- 联机(*online*): 没有事先计算好的路径
 - 脱机(*offline*): 事先算好传输路径
- 一到一 (单播)
 - 一到一 (置换): 每个处理器开始时最多发送一条信包,
每条信包有且仅有一个目的地;
 - 多到一 (集中)
 - 一到多 (多播)
 - 一到所有 (广播、组播)
 - 多到多 (会议)

第二章 并行机系统互连与基本通信操作

- 2.1 并行计算机互连网络
- 2.2 选路方法与开关技术
 - 预备知识
 - 2.2.1 选路方法
 - 2.2.2 开关技术
- 2.3 单一信包一到一传输
- 2.4 一到多播送
- 2.5 多到多播送

选路方法（1）

- 分类
 - 最短路径/非最短路径(贪心选路/随机选路),
如维序选路是贪心的, 二阶段维序选路是随机的
 - 确定选路/自适应选路(寻径确定/寻径视网络状况)
- 维序选路(Dimension-Ordered Routing)
 - 一种确定的最短路径选路
 - 二维网孔中的维序选路: X-Y选路
 - 超立方中的维序选路: E-立方选路

选路方法（2）

- X-Y选路算法

- 算法2.1：二维网孔上的X-Y选路算法

begin

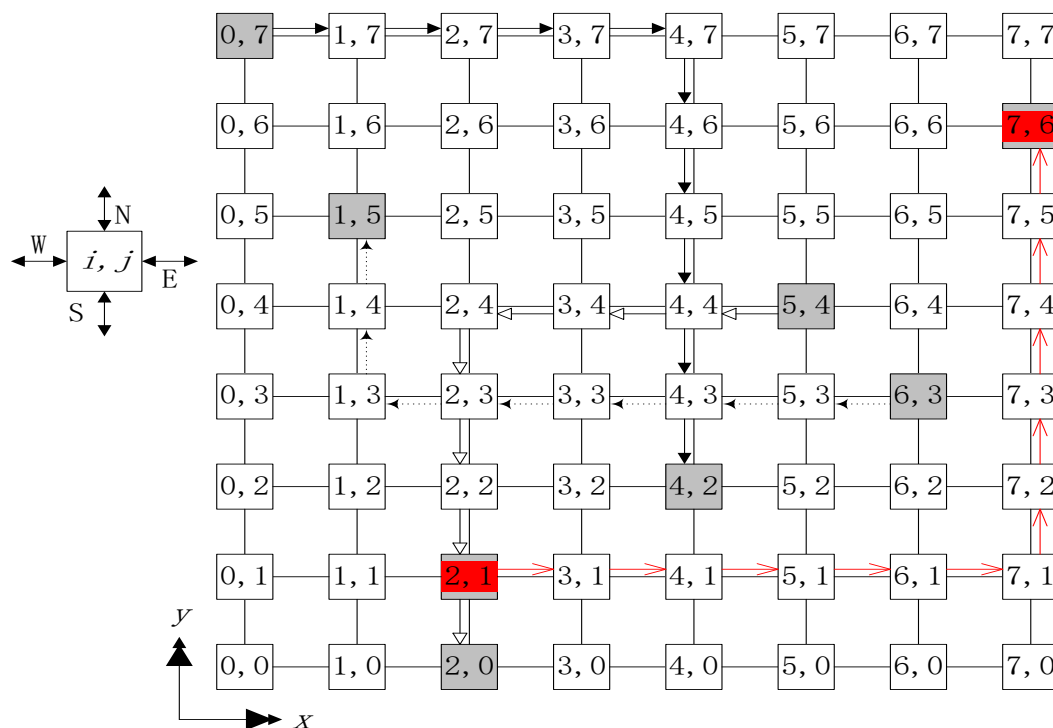
step1: 沿X方向将信包送至目的地处理器所在的列

step2: 沿Y方向将信包送至目的地处理器所在的行

end

选路方法 (3)

• 例2.1



注：本例无链
路与节点竞争
和死锁现象

4 (源;目的)对: $(2, 1; 7, 6) \longrightarrow (5, 4; 2, 0) \longrightarrow$
 $(0, 7; 4, 2) \longrightarrow (6, 3; 1, 5) \cdots \longrightarrow$

选路方法（4）

- E-立方选路算法

- 路由计算：
$$\begin{array}{r} s_{n-1}s_{n-2}\cdots s_1s_0(\text{源地址}) \\ \text{异或} \quad d_{n-1}d_{n-2}\cdots d_1d_0(\text{目的地址}) \\ \hline r_{n-1}r_{n-2}\cdots r_1r_0(\text{路由值}) \end{array}$$

- 路由过程：

$$s_{n-1}s_{n-2}\cdots s_1s_0 \rightarrow s_{n-1}s_{n-2}\cdots s_1s_0 \oplus r_0 \rightarrow$$

$$s_{n-1}s_{n-2}\cdots s_1s_0 \oplus r_1 \rightarrow \cdots$$

- 算法2.2：超立方网络上的E-立方选路算法

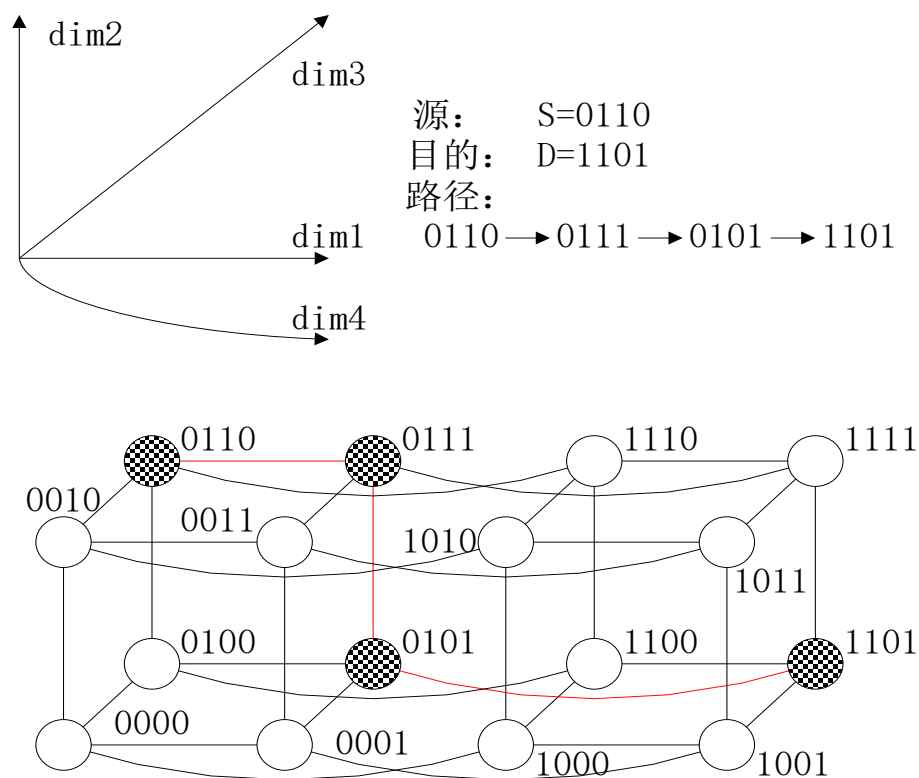
选路方法 (5)

- 例2.2

0110(S)

1101(D)

1011(R)



开关技术 (1)

- 存储转发(Store-and-Forward)选路

- 消息被分成基本的传输单位---信包(Packet), 每个信包都含有寻径信息 ;
- 当一个信包到达中间节点A时, A把整个信包放入其通信缓冲器中, 然后在选路算法的控制下选择下一个相邻节点B, 当从A到B的通道空闲并且B的通信缓冲器可用时, 把信包从A发向B ;
- 信包的传输时间: $t_{comm}(SF) = t_s + (mt_w + t_h) = O(m)$

缺点 :

- 每个结点必须对整个消息和信包进行缓冲, 缓冲器较大 ;
- 网络时延与发送消息所经历的节点数成正比。

开关技术 (2)

- 切通(Cut Through)选路

- 在传递一个消息之前，就为它建立一条从源节点到目的节点的物理通道。在传递的全部过程中，线路的每一段都被占用，当消息的尾部经过网络后，整条物理链路才被废弃。
- 传输时间: $t_{comm}(CT) = t_s + mt_w + lt_h = O(m+l)$

缺点：

- 物理通道非共享；
- 传输过程中物理通道一直被占用。

- 虫孔(Wormhole)选路

- Dally于1986年提出，利用了前二种方法的优点，减少了缓冲区，提高了物理通道的利用。
- 首先把一个消息分成许多很小的片，消息的**头片**包含了这个消息的所有寻径信息。**尾片**是一个其最后包含了消息结束符的片。中间的片均为**数据片**；
- 片是最小信息单位。每个节点上只需要缓冲一个片就能满足要求；
- 用一个头片直接牵引一条从输入链路到输出链路的路径的方法来进行操作。每个消息中的片以流水的方式在网络中向前“蠕动”。每个片相当于Worm的一个节，“蠕动”以节为单位顺序地向前爬行。当消息的尾片向前“蠕动”一步后，它刚才所占用的节点就被放弃了。

开关技术（4）

- 虫孔(Wormhole)选路

优点：

- (1)每个节点的缓冲器的需求量小。易于用VLSI实现；
- (2)较低的网络传输延迟。存储转发传输延迟基本上正比于消息在网络中传输的距离；Wormhole与线路开关的网络传输延迟正比于消息包的长度，传输距离对它的影响很小（消息包较长时的情况）；
- (3)通道共享性好、利用率高；
- (4)易于实现Multicast和Broadcast。

• 几种开关技术的时空图

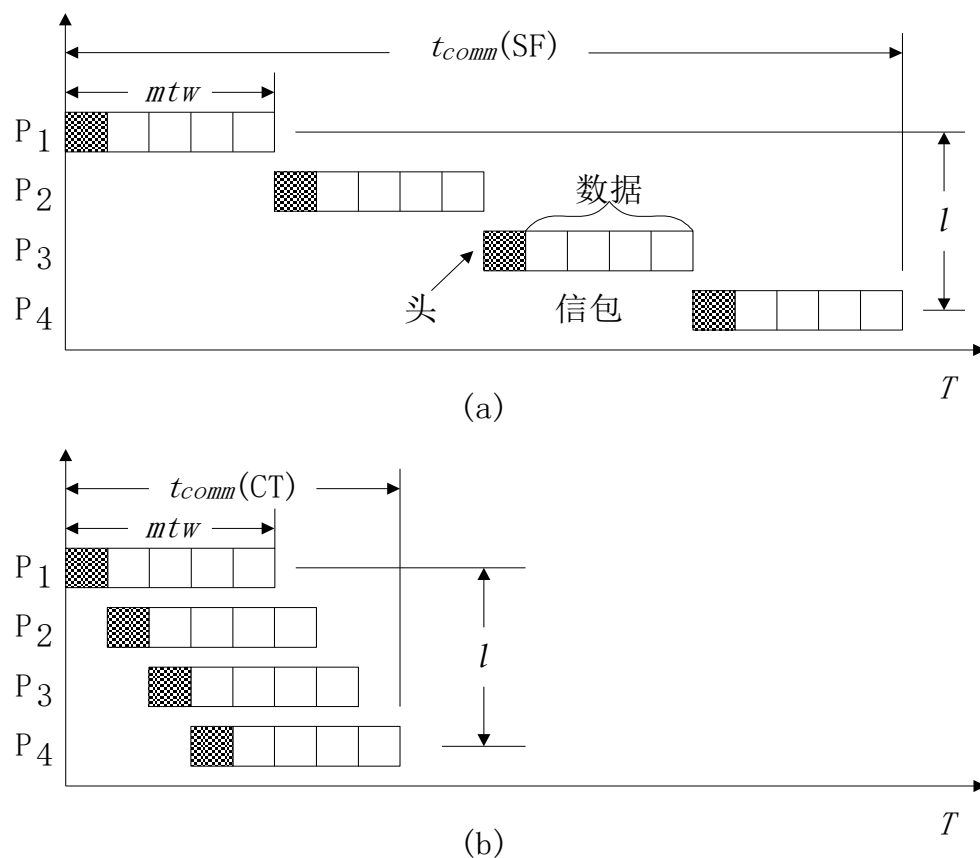
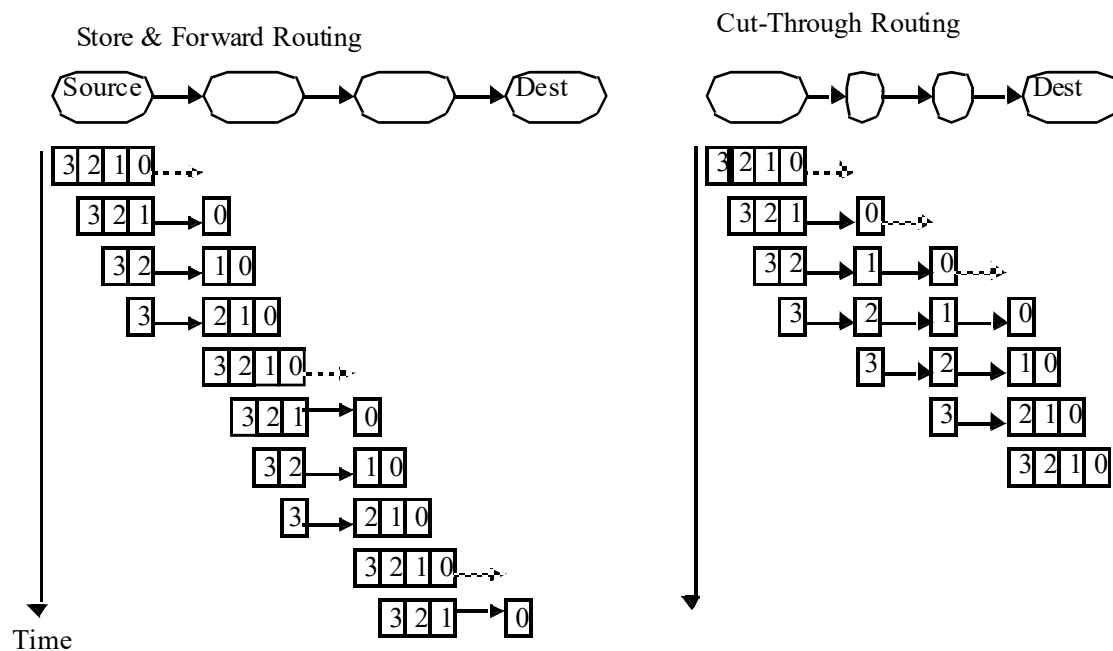


图8.4

开关技术 (6)

• 比较和演示



第二章 并行机系统互连与基本通信操作

- 2.1 并行计算机互连网络
- 2.2 选路方法与开关技术
- 2.3 单一信包一到一传输
- 2.4 一到多播送
- 2.5 多到多播送

单一信包一到一传输

- 距离 l 的计算：对于 p 个处理器
 - 一维环形： $l \leq \lfloor p/2 \rfloor$
 - 带环绕Mesh($\sqrt{p} \times \sqrt{p}$)： $l \leq 2\lfloor \sqrt{p}/2 \rfloor$
 - 超立方： $l \leq \log p$
- $t_{\text{comm}}(\text{SF})$ 的计算(可由(8.1b)式得到)
 - 一维环形： $t_{\text{comm}}(\text{SF}) = t_s + t_w \cdot m \cdot \lfloor p/2 \rfloor$
 - 带环绕Mesh： $t_{\text{comm}}(\text{SF}) = t_s + 2t_w \cdot m \cdot \lfloor \sqrt{p}/2 \rfloor$
 - 超立方： $t_{\text{comm}}(\text{SF}) = t_s + t_w \cdot m \cdot \log p$
- $t_{\text{comm}}(\text{CT})$ 的计算(可由(8.2b)式得到)
$$t_{\text{comm}}(\text{CT}) = t_s + mt_w$$
- 如果 $m \gg p$ ： $t_{\text{comm}}(\text{SF}) \approx t_{\text{comm}}(\text{CT}) = t_s + mt_w$

第二章 并行机系统互连与基本通信操作

- 2.1 并行计算机互连网络
- 2.2 选路方法与开关技术
- 2.3 单一信包一到一传输
- 2.4 一到多播送
 - 2.4.1 使用SF进行一到多播送
 - 2.4.2 使用CT进行一到多播送
- 2.5 多到多播送

一到多播送—SF模式 (1)

- 环

- 步骤：①先左右邻近传送;②再左右二个方向同时播送
- 示例：

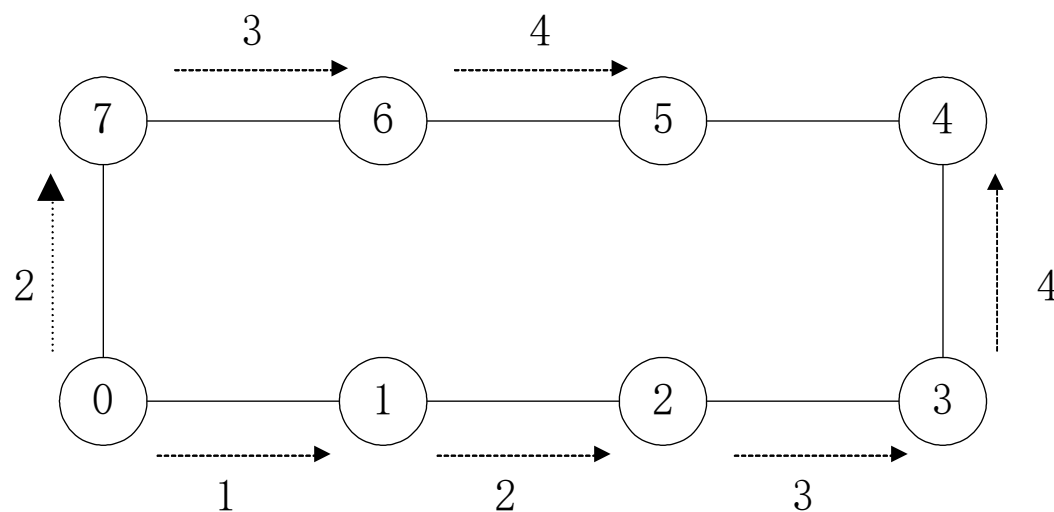


图8.5

- 通讯时间：
$$t_{one-to-all}(SF) = (t_s + mt_w) \lceil p/2 \rceil$$

一到多播送—SF模式 (2)

- 环绕网孔

- 步骤：①先完成一行中的播送;②再同时进行各列的播送
- 示例：
共4步(2步行、2步列)

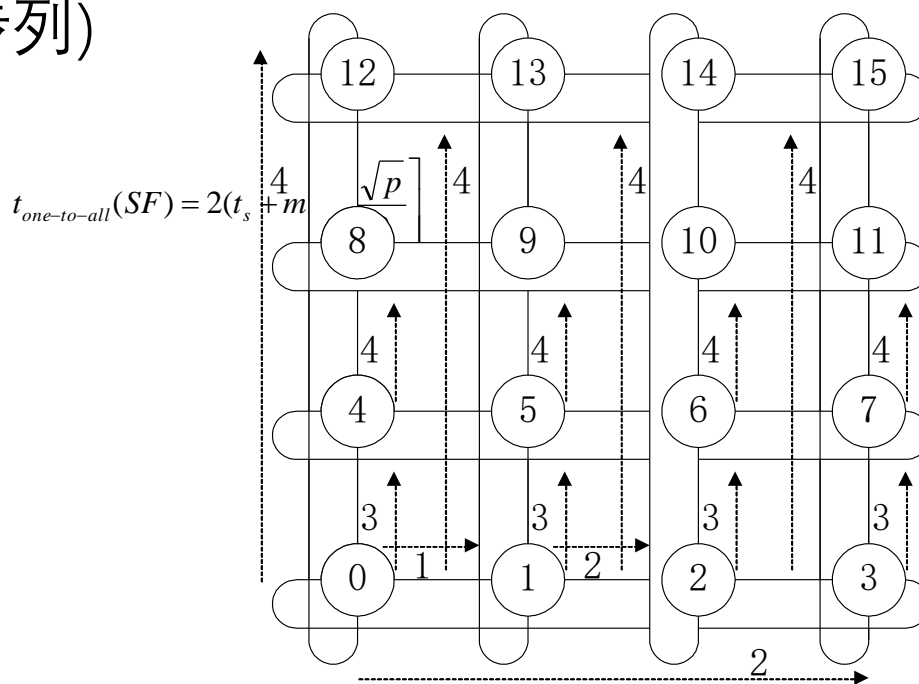


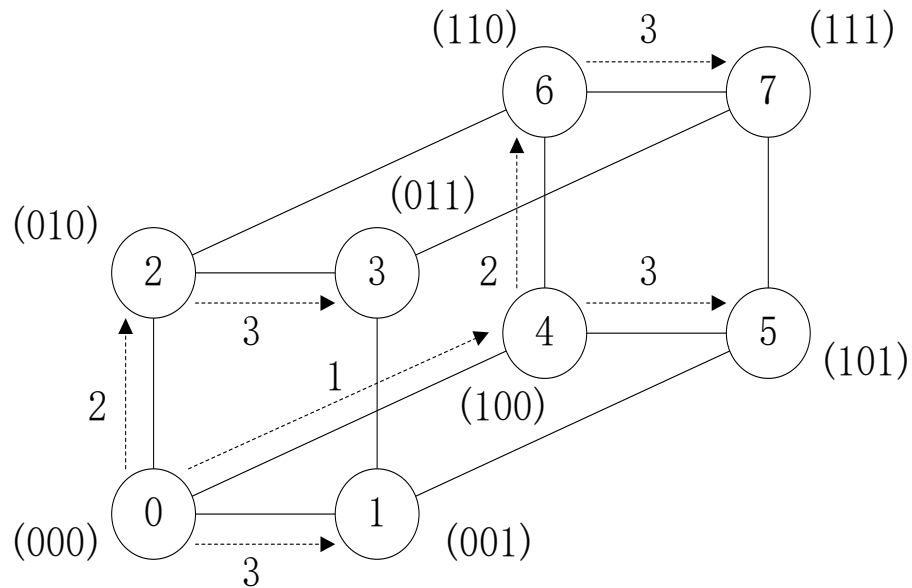
图 8.6

- 通讯时间：

一到多播送—SF模式 (3)

● 超立方

- 步骤：从低维到高维，依次进行播送;
- 示例： $(110)_3$



- 通讯时间： $t_{one-to-all}(SF) = (t_s + mt_w) \log p$

一到多播送—CT模式 (1)

- 环

- 步骤：

- (1)先发送至 $p/2$ 远的处理器;

- (2)再同时发送至 $p/2^2$ 远的处理器;

.....

- (i)再同时发送至 $p/2^i$ 远的处理器;

- 示例：如左图

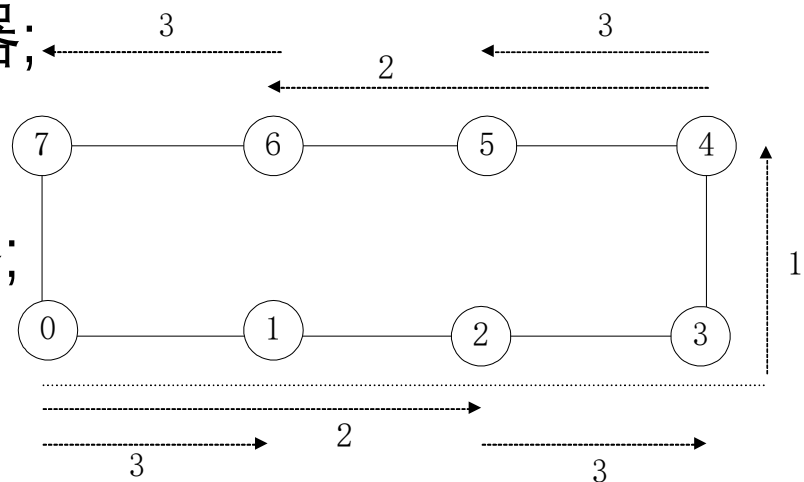


图 8.8

- 通讯时间：

$$t_{one-to-all}(CT) = \sum_{i=1}^{(8.2) \log p} (t_s + mt_w + t_h p / 2^i)$$

$$= t_s \log p + mt_w \log p + t_h (p - 1)$$

$$\approx (t_s + mt_w) \log p \quad (t_h \text{可忽略时})$$

一到多播送—CT模式 (2)

- 网孔

- 步骤:

- (1) 先进行行播送;

$$// t_s \log \sqrt{p} + mt_w \log \sqrt{p} + t_h(\sqrt{p}-1)$$

- (2) 再同时进行列播送;

$$// t_s \log \sqrt{p} + mt_w \log \sqrt{p} + t_h(\sqrt{p}-1)$$

- 示例: 图8.9

- 通讯时间:

$$\begin{aligned} t_{one-to-all}(CT) &= 2(t_s \log \sqrt{p} + mt_w \log \sqrt{p} + t_h(\sqrt{p}-1)) \\ &= (t_s + mt_w) \log p + 2t_h(\sqrt{p}-1) \end{aligned}$$

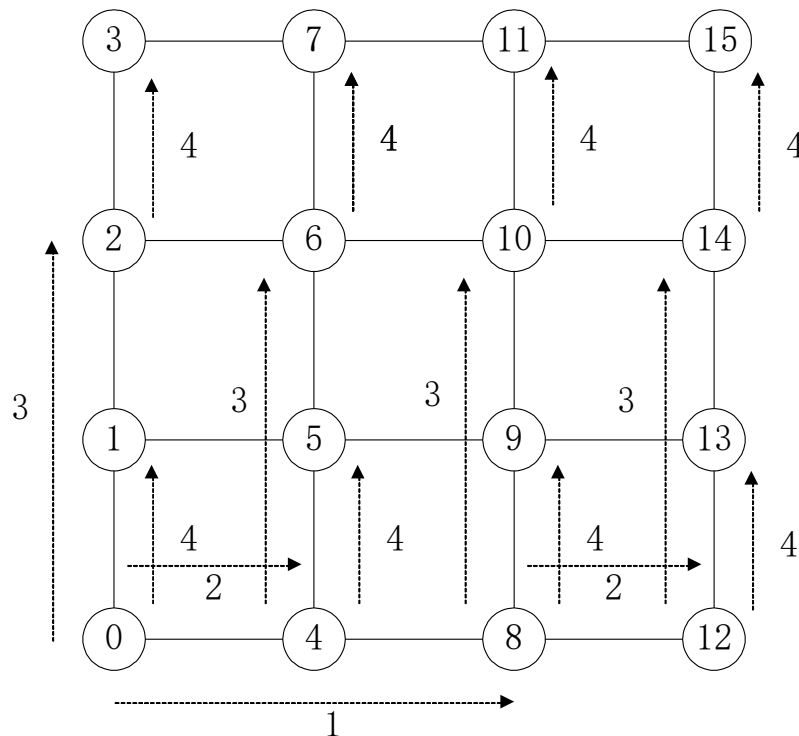


图 8.9

一到多播送—CT模式 (3)

- 超立方

- 步骤：

依次从低维到高维播送, d-立方,
 $d=0,1,2,3,4\cdots$;

- 通讯时间： $t_{one-to-all}(CT) = (t_s + mt_w) \log p$

- 小结

	环	网孔	超立方
$t_{one-to-all}(SF)$	$(t_s + mt_w) \lceil p/2 \rceil$	$2(t_s + mt_w) \lceil \sqrt{p}/2 \rceil$	$(t_s + mt_w) \log p$
$t_{one-to-all}(CT)$		$(t_s + mt_w) \log p$	

第二章 并行机系统互连与基本通信操作

- 2.1 并行计算机互连网络
- 2.2 选路方法与开关技术
- 2.3 单一信包一到一传输
- 2.4 一到多播送
- 2.5 多到多播送
 - 2.5.1 使用SF进行多到多播送
 - 2.5.2 使用CT进行多到多播送

多到多播送—SF模式 (1)

• 环

- 步骤：
同时向右(或左)播送
刚接收到的信包

- 示例：如左图

- 通讯时间：

$$t_{all-to-all}(SF) = (t_s + mt_w)(p - 1)$$

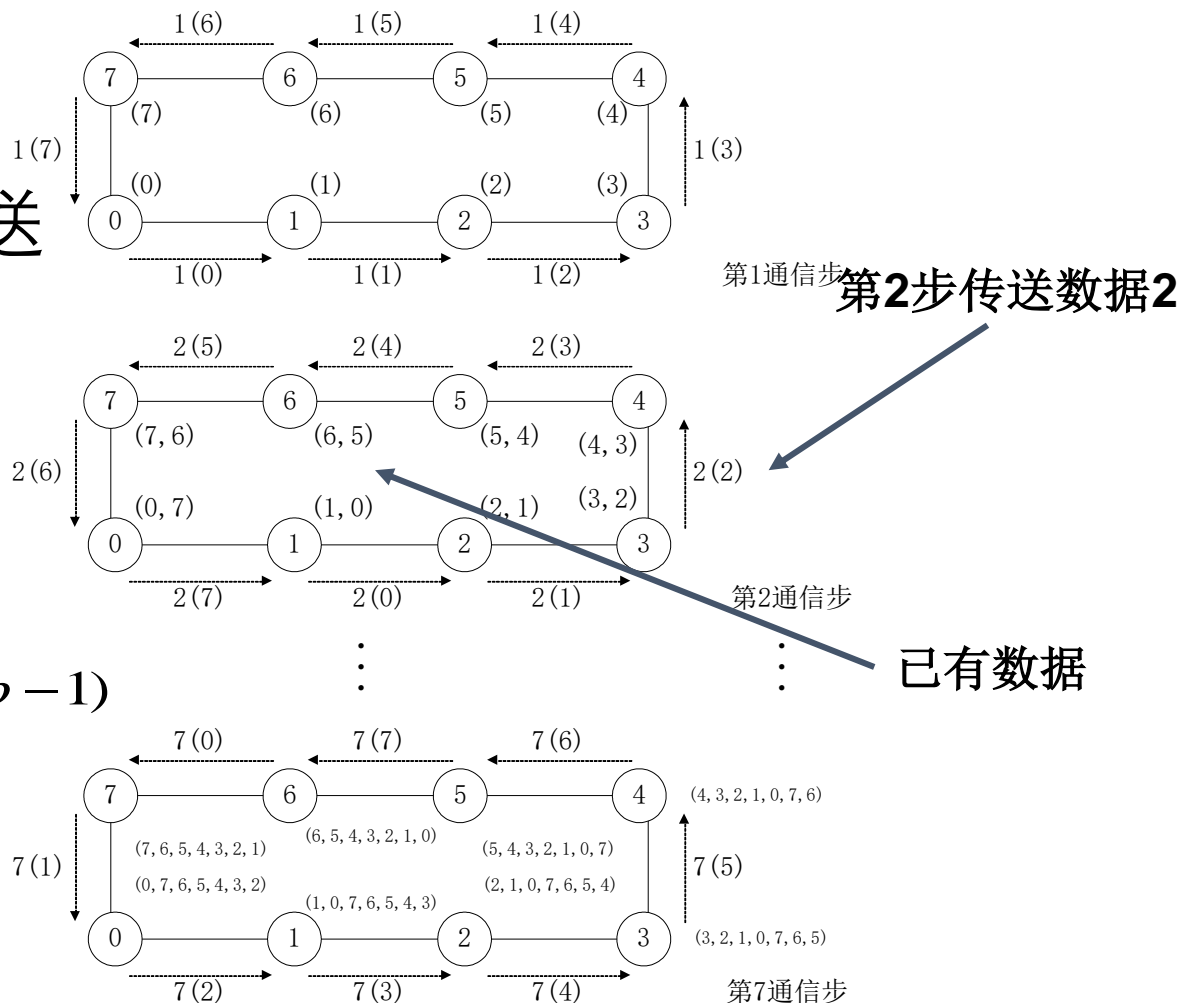


图8.10

多到多播送—SF模式 (2)

- 环绕网孔

- 步骤：

- (1)先进行行的播送

- (2)再进行列的播送

- 示例：如左图

- 通讯时间：

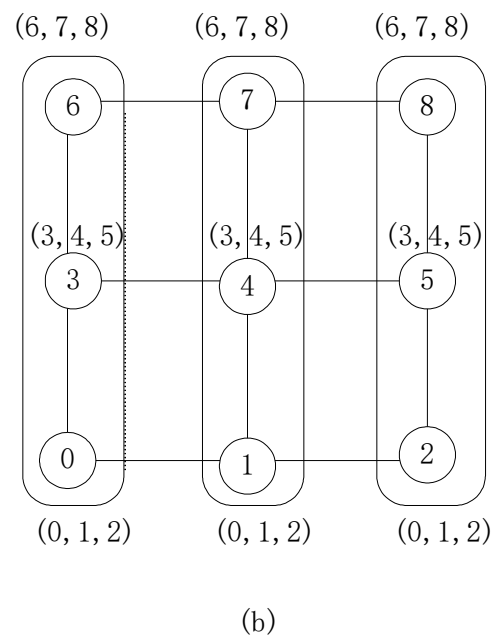
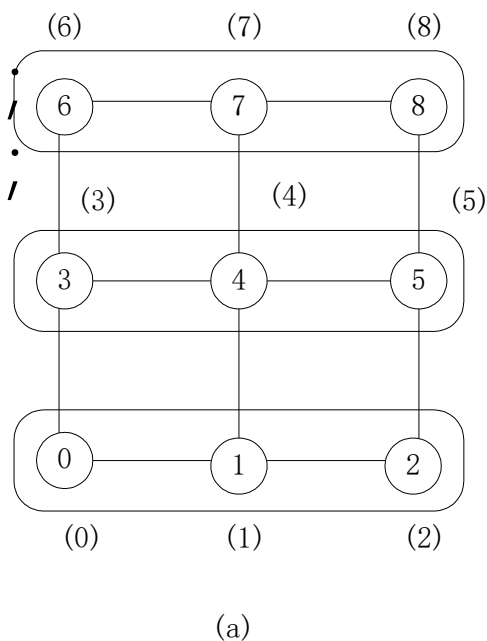


图8.11

$$\begin{aligned} t_{all-to-all}(SF) &= (t_s + mt_w)(\sqrt{p} - 1) + (t_s + m\sqrt{p} \cdot t_w)(\sqrt{p} - 1) \\ &= 2t_s(\sqrt{p} - 1) + mt_w(p - 1) \end{aligned}$$

多到多播送—SF模式 (3)

• 超立方

- 步骤：
依次按维进行
多到多的播送；

- 示例：如左图

- 通讯时间：

$$t_{all-to-all}(SF) = \sum_{i=1}^{\log p} (t_s + 2^{i-1} mt_w)$$

$$= t_s \log p + mt_w (p-1)$$

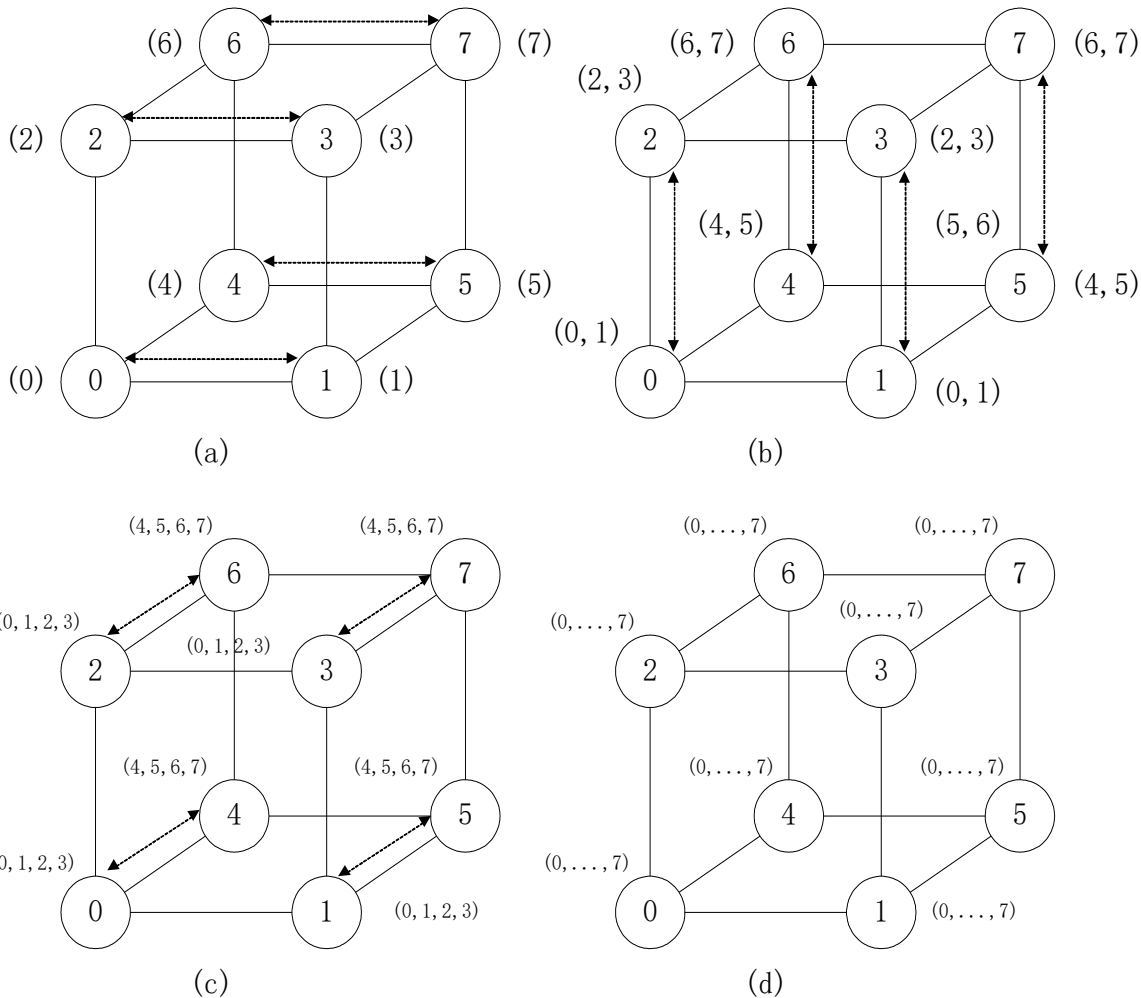


图8.12

多到多播送—CT模式

- 使用一到多的策略会造成链路竞争

$$t_{all-to-all}(CT) = t_{all-to-all}(SF)$$

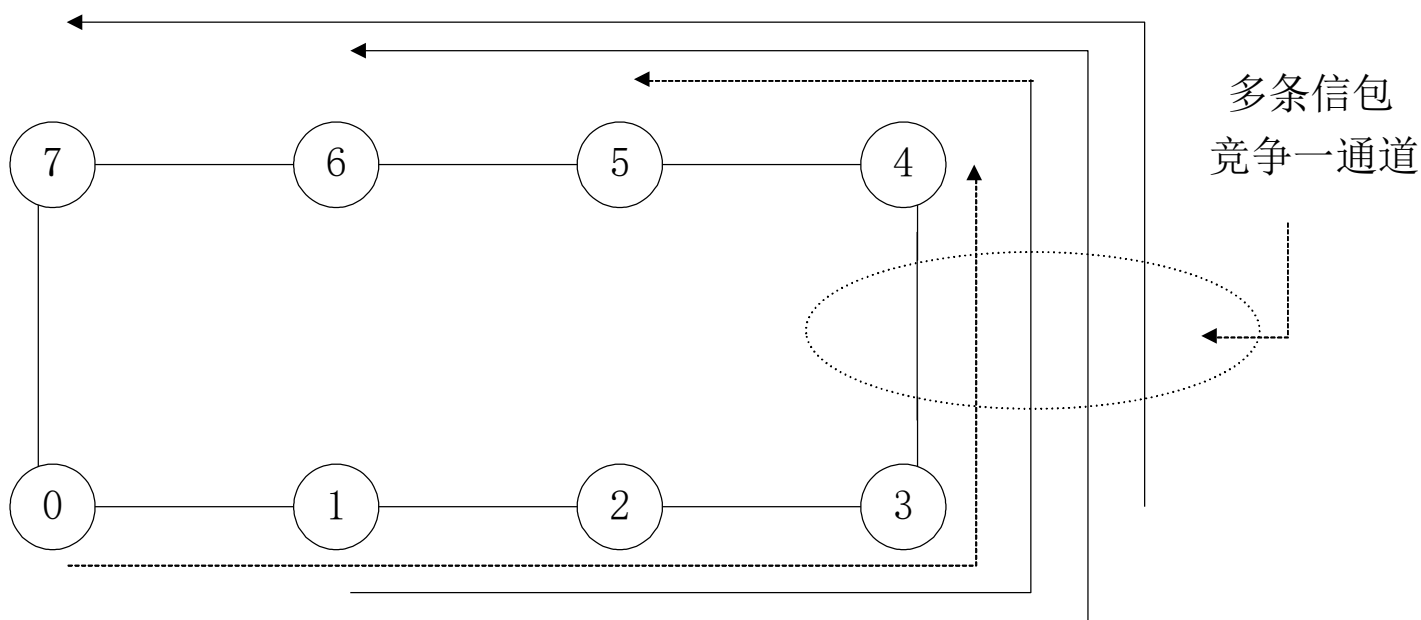


图8. 13

通信时间一览表

基本公式

$$T_{comm}(SF) = t_s + (mt_w + t_h) \cdot l \approx t_s + mt_w \cdot l = O(m \cdot l)$$

$$T_{comm}(CT) = t_s + mt_w + lt_h \approx t_s + mt_w = O(m + l)$$

p -环	$\sqrt{p} \times \sqrt{p}$ -环绕网孔	p -超立方
$T_{one-to-one}(SF): t_s + mt_w \lfloor p/2 \rfloor$	$t_s + 2mt_w \lfloor \sqrt{p}/2 \rfloor$	$t_s + mt_w \log p$
$T_{one-to-one}(CT): t_s + mt_w$	$t_s + mt_w$	$t_s + mt_w$
$T_{one-to-all}(SF): (t_s + mt_w) \lceil p/2 \rceil$	$2(t_s + mt_w) \lceil \sqrt{p}/2 \rceil$	$(t_s + mt_w) \log p$
$T_{one-to-all}(CT): (t_s + mt_w) \log p$	$(t_s + mt_w) \log p$	$(t_s + mt_w) \log p$
$T_{all-to-all}(SF): (t_s + mt_w)(p-1)$	$2t_s(\sqrt{p}-1) + mt_w(p-1)$	$t_s \log p + mt_w(p-1)$
$T_{all-to-all}(CT):$ 同上	同上	同上