并行计算
Parallel Computing

主讲 孙经纬

2024年 春季学期

概要

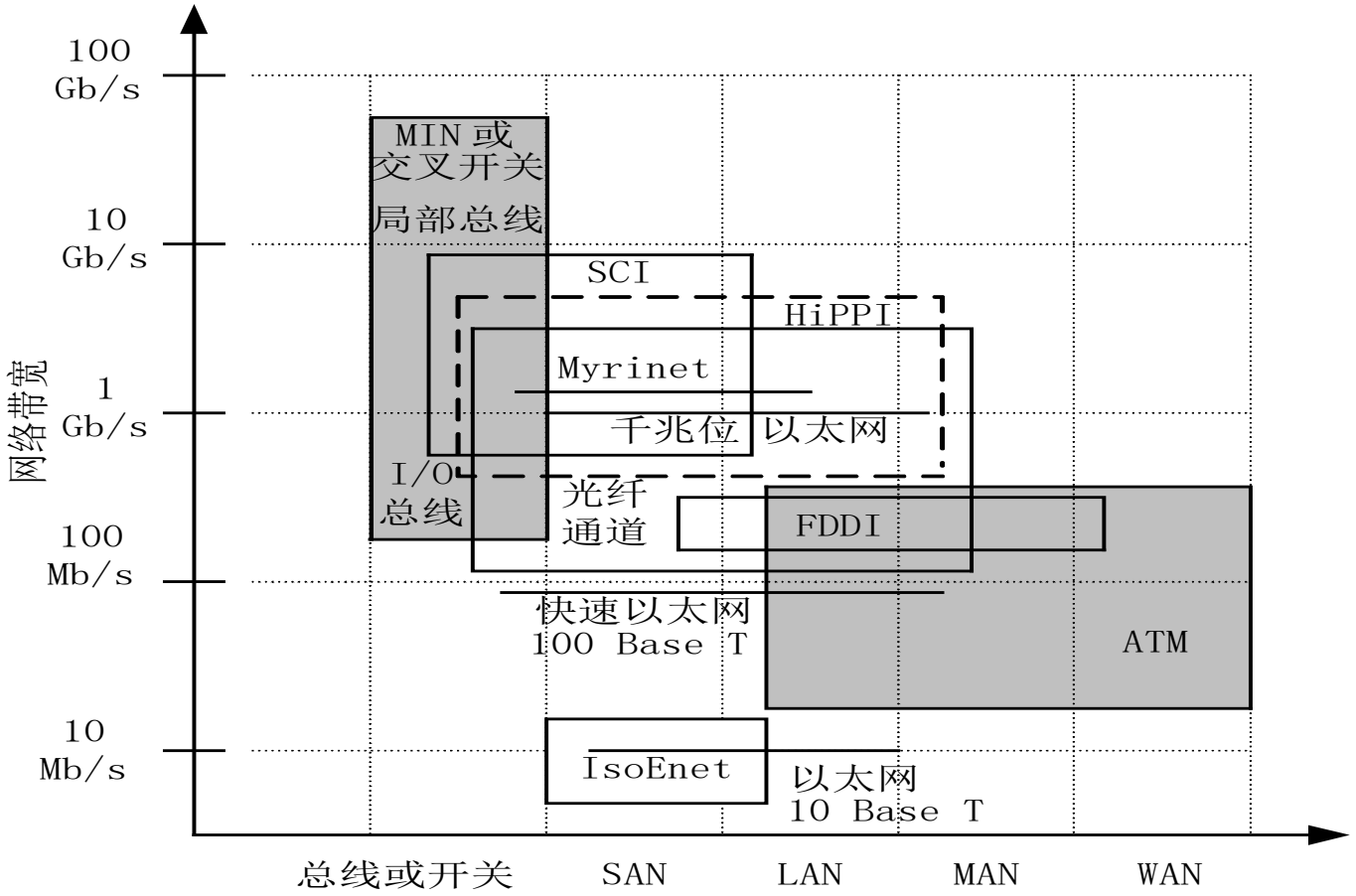
- 第一篇 并行计算硬件平台：并行计算机
  - 第一章 并行计算与并行计算机结构模型
  - **第二章 并行计算机系统互连与基本通信操作**
  - 第三章 典型并行计算机系统介绍
  - 第四章 并行计算性能评测

概要

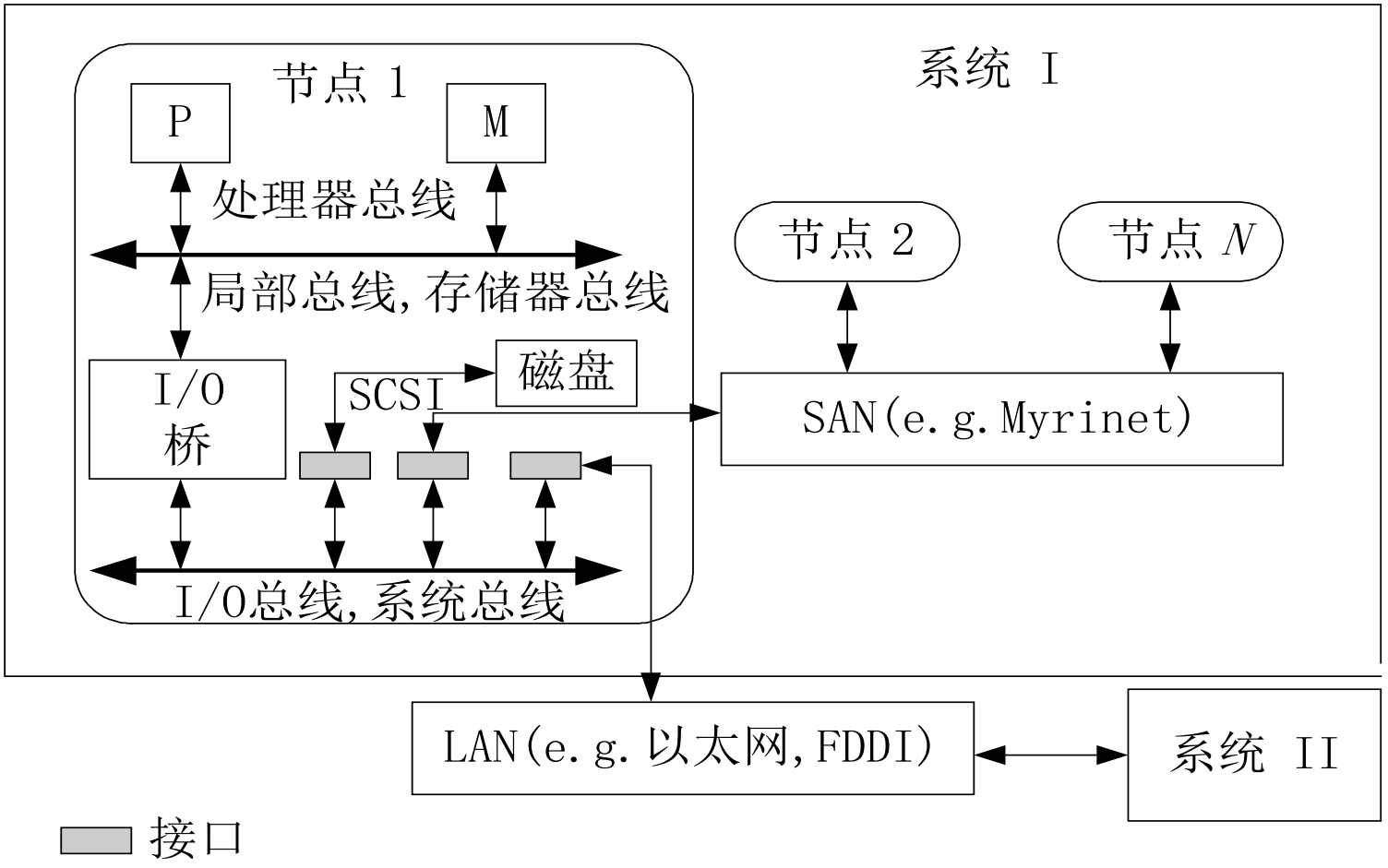
- 第二章 并行机系统互连与基本通信操作
  - **2.1 并行计算机互连网络**
    - 2.1.1 系统互连
    - 2.1.2 静态互连网络
    - 2.1.3 动态互连网络
    - 2.1.4 标准互连网络
  - 2.2 选路方法与开关技术
  - 2.3 单一信包一到一传输
  - 2.4 一到多播送
  - 2.5 多到多播送

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



网络特征指标

- 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短距离的最大值。
- 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。
- 如果从任一节点观察网络都一样，则称网络为对称的（Symmetrical）。

静态互连网络与动态互连网络

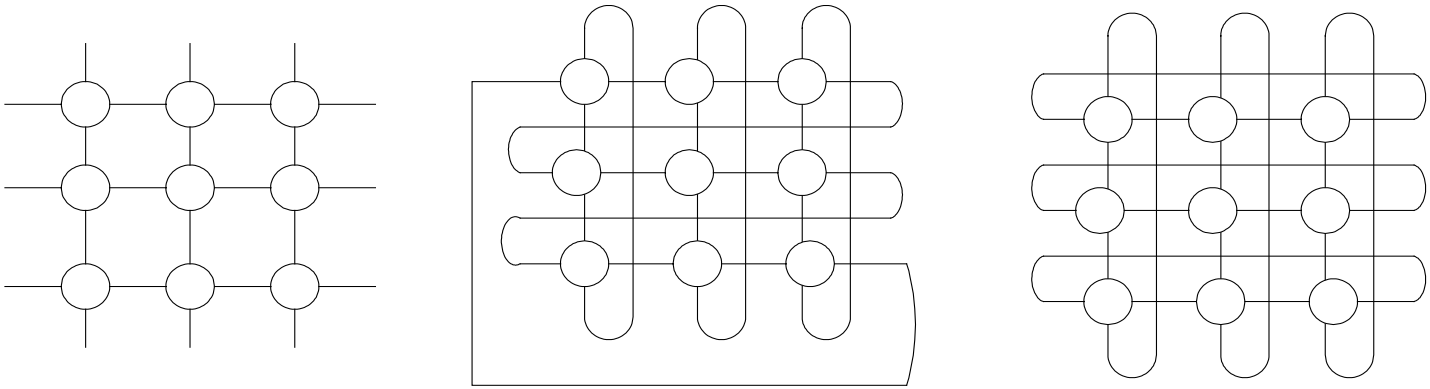
- 静态互连网络：处理单元间有着固定连接的一类网络，在程序执行期间，这种点到点的链接保持不变；典型的静态网络有一维线性阵列、二维网孔、树连接、超立方网络、立方环、洗牌交换网、蝶形网络等。
- 动态网络：用交换开关构成的，可按应用程序的要求动态地改变连接组态；典型的动态网络包括总线、交叉开关和多级互连网络等。

静态互连网络

- 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，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，直径为⌊N/2⌋（双向环）或N-1（单向环），对剖宽度为2 。

静态互连网络

- $\sqrt{N} \times \sqrt{N}$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），内节点度为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S形蛇状，就变成Illiac网孔，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2-D环绕（2-D Torus），节点度恒为4，网络直径为$2\lfloor\sqrt{N}/2\rfloor$，对剖宽度为$2\sqrt{N}$

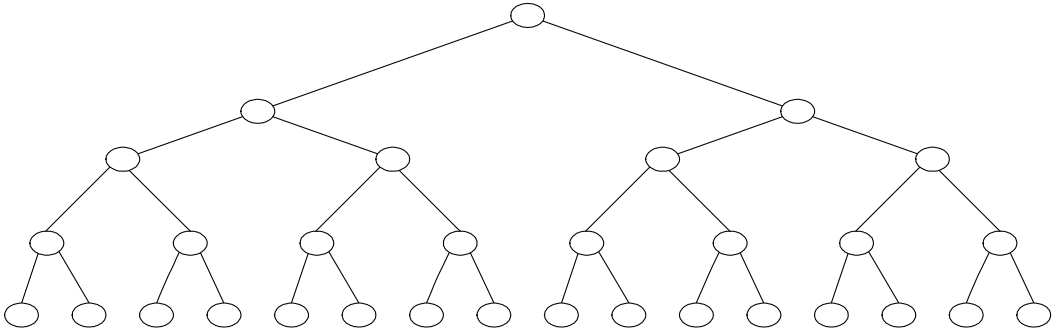


(a)2-D网孔　(b)Illiac网孔　(c)2-D环绕

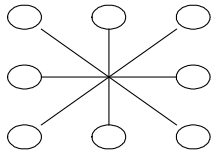
静态互连网络

- 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，则直径缩小为2，此时就变成了星形网络，其对剖宽度为$\lfloor N/2 \rfloor$
  - 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

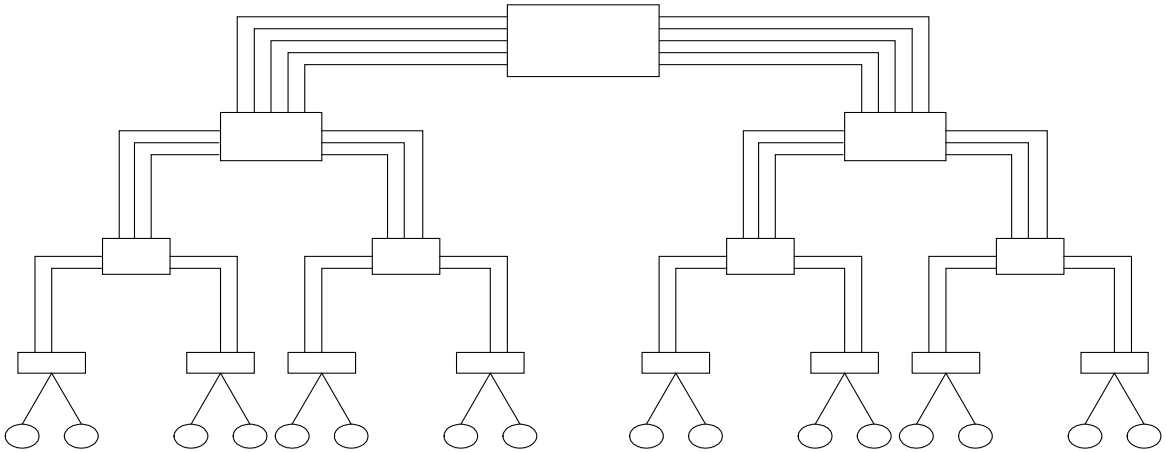
静态互连网络



(a) 二叉树



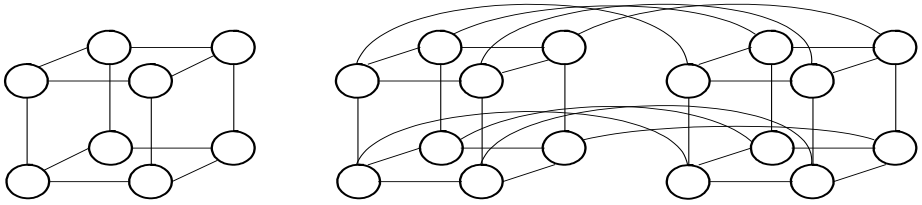
(b) 星形连接



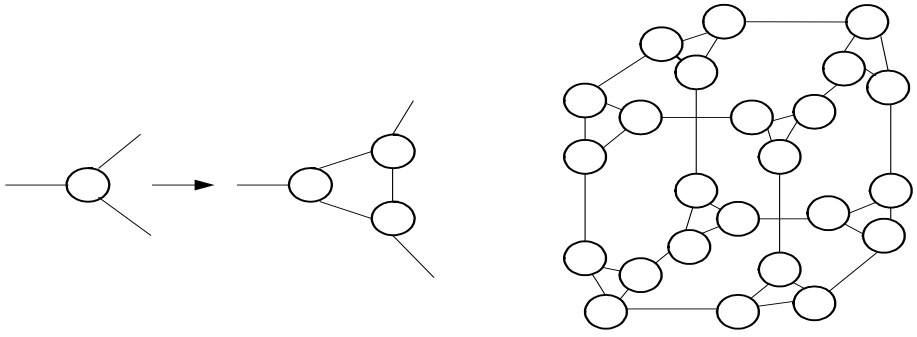
(c) 二叉胖树

静态互连网络

- 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



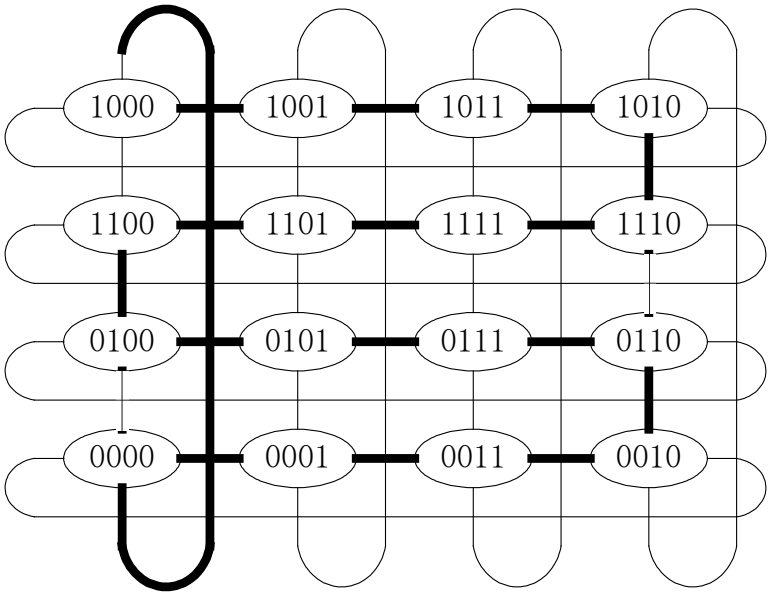
(c) 顶点代之以环　　(d) 3-立方环

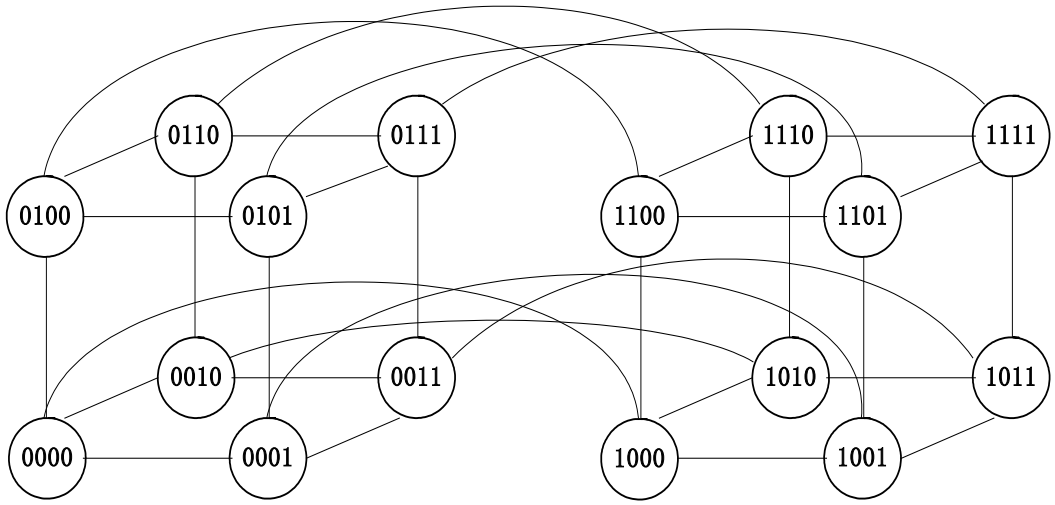
嵌入

- 将网络中的各节点映射到另一个网络中
- 用**膨胀**（Dilation）系数来描述嵌入的质量: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

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



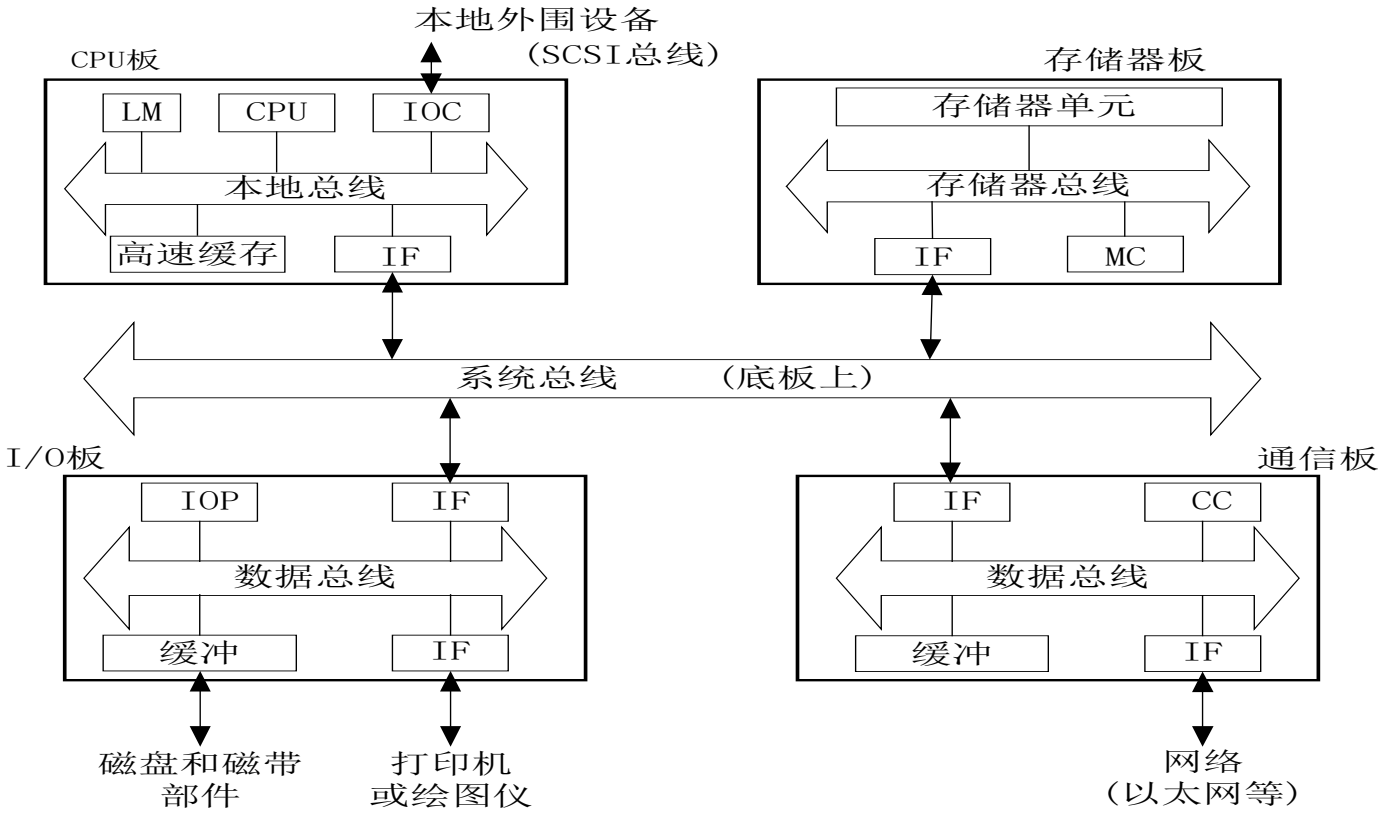


静态互连网络小结

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	否	$N-1$
环形	N	2	$\lfloor N/2 \rfloor$ (双向)	2	是	N
2-D网孔	$\left(\sqrt{N}\times\sqrt{N}\right)$	4	$2(\sqrt{N}-1)$	$\sqrt{N}$	否	$2(N-\sqrt{N})$
Illiac网孔	$\left(\sqrt{N}\times\sqrt{N}\right)$	4	$\sqrt{N}-1$	$2\sqrt{N}$	否	$2N$
2-D环绕	$\left(\sqrt{N}\times\sqrt{N}\right)$	4	$2\lfloor\sqrt{N}/2\rfloor$	$2\sqrt{N}$	是	$2N$
二叉树	N	3	$2(\lceil\log N\rceil-1)$	1	否	$N-1$
星形	N	$N-1$	2	$\lfloor N/2 \rfloor$	否	$N-1$
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	$3N/2$

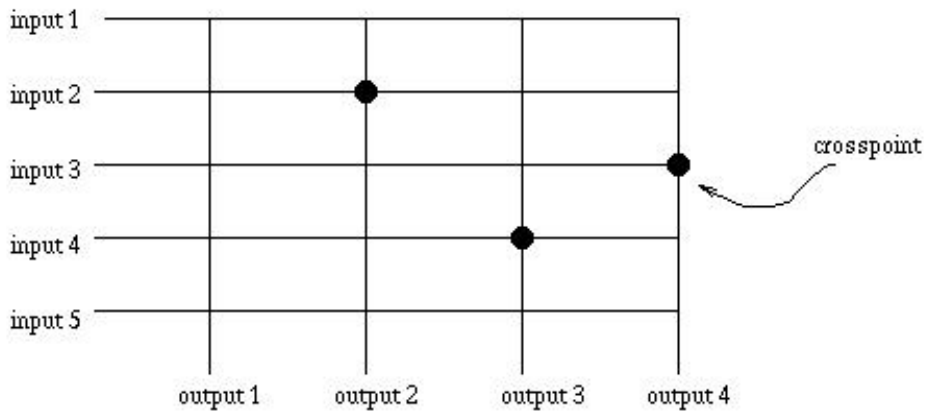
动态互连网络

- 总线（分时工作）：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

- 交叉开关（Crossbar）：
  - 单级交换网络，可为每个端口提供更高的带宽。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或PVP中处理器和存储器之间的存取。

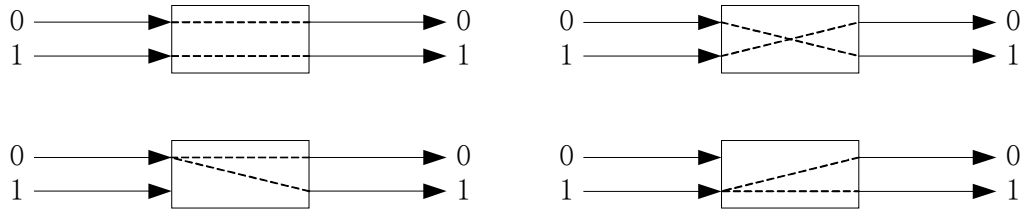


A Small Crossbar Switch

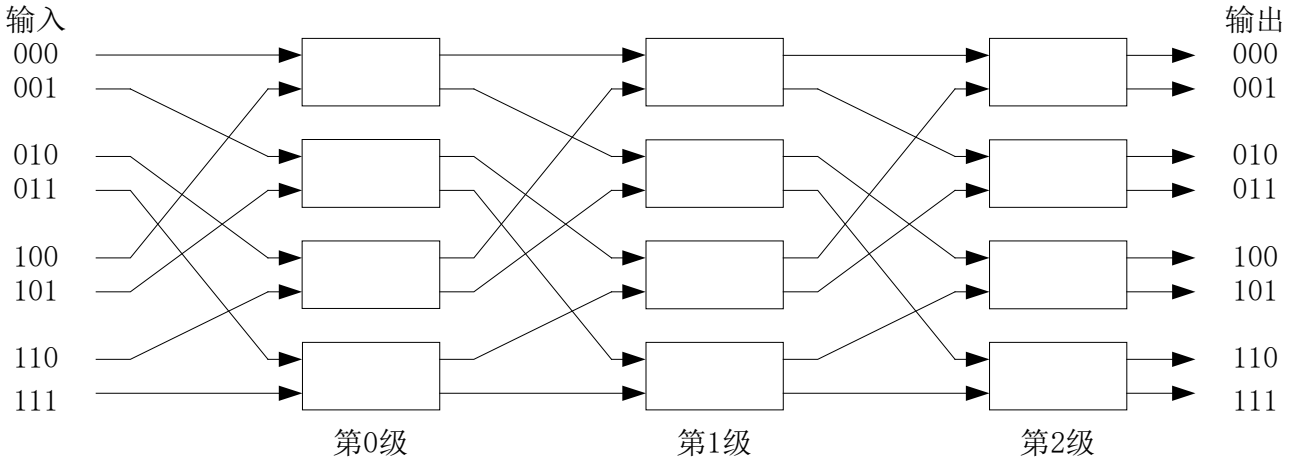
动态互连网络

- 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

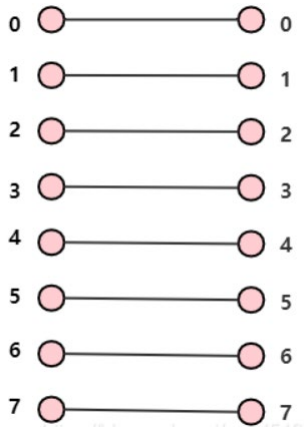
- 交换开关模块:
  - 一个交换开关模块有n个输入和n个输出，每个输入可连接到任意输出端口，但只允许一对一或一对多的映射，不允许多对一的映射，因为这将发生输出冲突
- 级间互连（Interstage Connection）：
  - 均匀洗牌、蝶网、多路均匀洗牌、交叉开关、立方连接
  - n输入的Ω网络（均匀洗牌）需要$\log_2 n$ 级 2×2开关

互连函数

- 恒等函数

  同号输入端和输出端之间的连接

$$I(x_{n-1}x_{n-2}\dots x_1x_0) = x_{n-1}x_{n-2}\dots x_1x_0$$

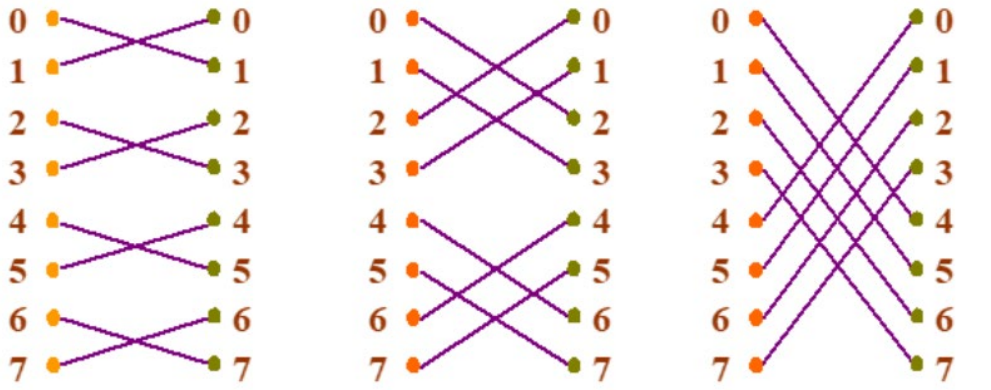


互连函数

- 交换函数

  二进制地址编码中第k位互反的输入端与输出端之间的连接

$$Cube_0(x_2x_1x_0) = x_2x_1\bar{x}_0$$
$$Cube_1(x_2x_1x_0) = x_2\bar{x}_1x_0$$
$$Cube_2(x_2x_1x_0) = \bar{x}_2x_1x_0$$



(a) $Cube_0$ 交换函数　(b) $Cube_1$ 交换函数　(c) $Cube_2$ 交换函数

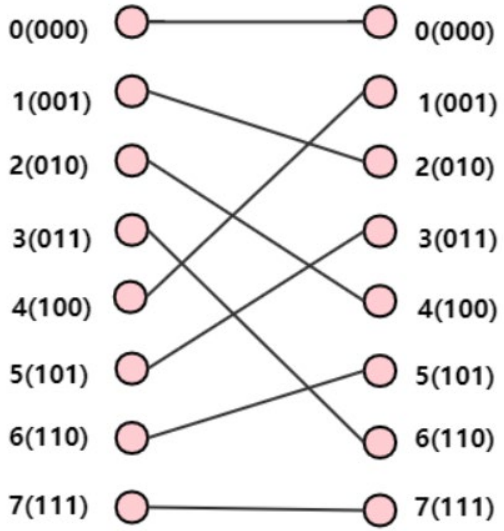
互连函数

- 均匀洗牌函数

  输入端的二进制编号循环左移一位

$$\sigma(x_{n-1}x_{n-2}\ldots x_1x_0) = x_{n-2}x_{n-3}\ldots x_1x_0x_{n-1}$$

  将输入端分成数目相等的两半，前一半和后一半按类似均匀混洗扑克牌的方式交叉地连接到输出端（输出端相当于混洗的结果）

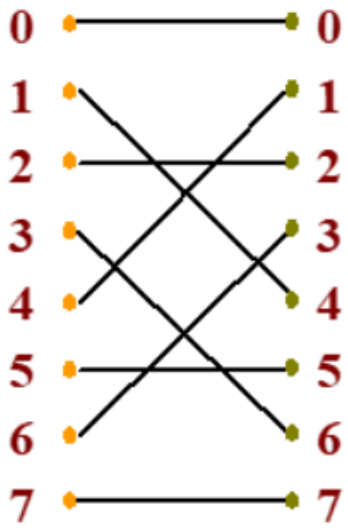


互连函数

- 蝶式函数

  输入端的二进制编号的最高位与最低位互换位置，便得到了输出端的编号

$$\beta(x_{n-1}x_{n-2}\ldots x_1x_0) = x_0x_{n-2}\ldots x_1x_{n-1}$$



互连函数

- 反位序函数

  输入端二进制编号的位序颠倒过来求得相应输出端的编号

$$\rho(x_{n-1}x_{n-2}\ldots x_1x_0) = x_0x_1\ldots x_{n-2}x_{n-1}$$

- 移数函数

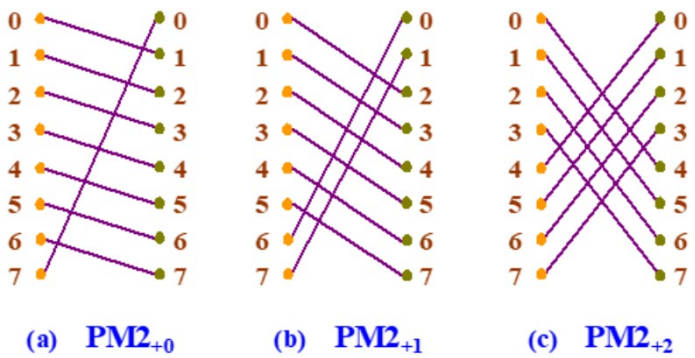
  将各输入端都错开一定的位置（模N）后连接到输出端

$$\alpha(x) = (x \pm k)\ mod\ N\ 1 \le x \le N-1, 1 \le k \le N-1$$

- PM2I函数

  将各输入端都错开一定的位置（模N）然后连接到输出端

$$PM2_{+i}(x) = x + 2^i\ mod\ N$$
$$PM2_{-i}(x) = x - 2^i\ mod\ N$$



N=8 的PM2I函数

动态互连网络比较

n：节点规模

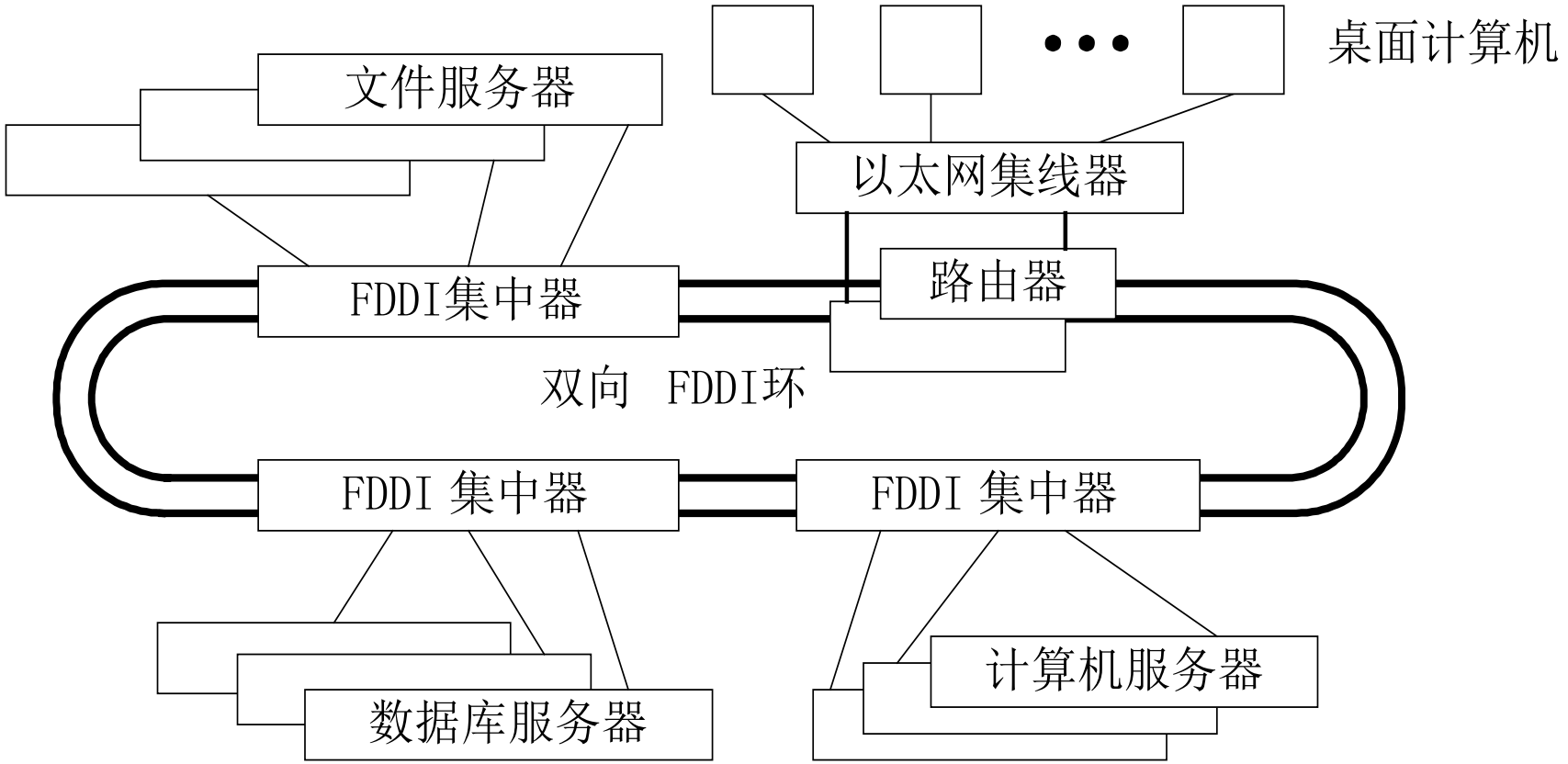
w：数据宽度

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

- 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

标准互连网络



双向FDDI环作为主干网

标准互连网络

- 以太网（Ethernet）
  - 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1Gbps以太网
  - 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

  - 2010年IEEE 802.3ba发布，支持40 Gbps和100 Gbps的传输速度。这标志着以太网进入了超高速网络的时代，主要应用于数据中心和高性能计算网络。
  - 2017年及以后：IEEE 802.3bs和802.3cd等标准化工作推动了200Gbps和400Gbps以太网的发展。这些技术继续扩大以太网在数据中心和服务提供商网络中的应用。

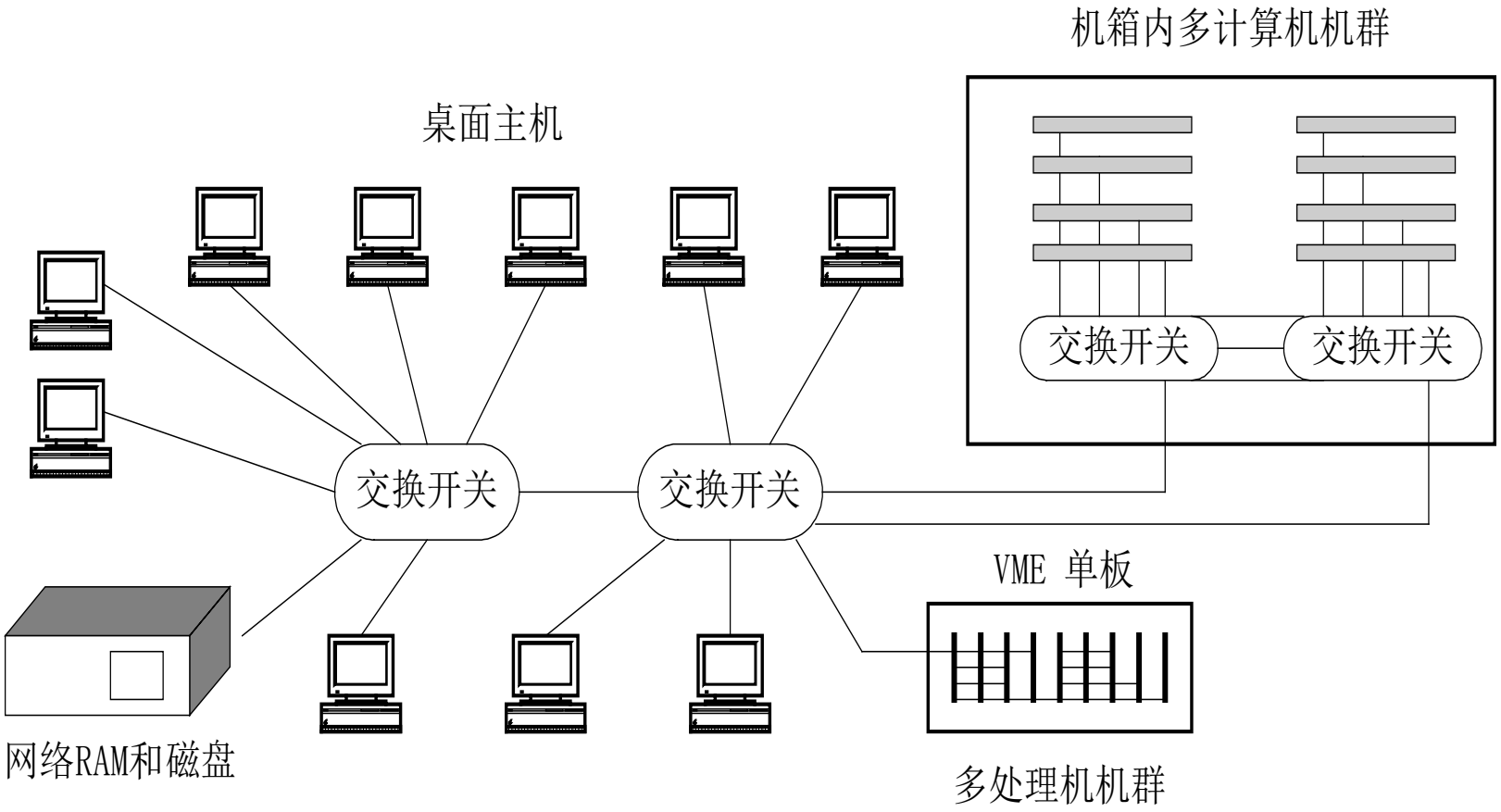
以太网特性一览表

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

标准互连网络

- 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

标准互连网络

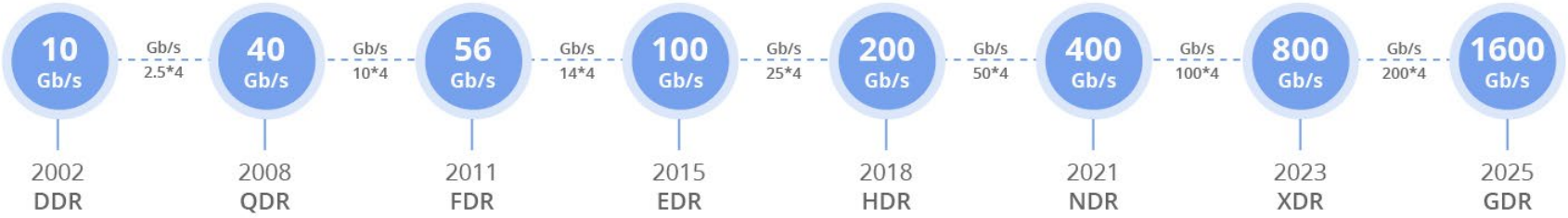


Myrinet连接的LAN/Cluster

标准互连网络

InfiniBand（直译为“无限带宽”技术，缩写为IB）是一个用于高性能计算机的网络通信标准，具有极高的吞吐量和极低的延迟，用于计算机与计算机之间的数据互连。IB由InifiBand商业联盟开发，包括Dell, HP, IBM, Intel, MSFT等200多成员。

- SDR - Single Data Rate, 8Gbps.
- DDR - Double Data Rate, 10Gbps/16Gbps.
- QDR - Quad Data Rate, 40Gbps/32Gbps.
- FDR - Fourteen Data Rate, 56Gbps.
- EDR - Enhanced Data Rate, 100Gbps.
- HDR - High Dynamic Range, 200Gbps.
- NDR - Next Data Rate, 400Gbps.
- XDR - eXtreme Data Rate, 800Gbps.



现代超级计算机的互连网络

Rank	Computer Name	Manufacture	Interconnect Family	ICN	Bidirectional Bandwidth	Switch Radix	Latency	Topology
1	Fugaku [2,10]	Fujitsu	Proprietary Network	Tofu D	108.8 Gbps	20	≤0.54 µs	6D-Torus
2	Summit [2,17]	IBM	InfiniBand	EDR InfiniBand	200 Gbps	36	0.6 µs	Fat-tree
3	Sierra [2,17]	IBM/ NVIDIA/ Mellanox	InfiniBand	EDR InfiniBand	200 Gbps	36	0.6 µs	Fat-tree
4	Sunway Taihu Light [2,11]	NRCPC	Custom Interconnect	Sunway	200 Gbps	36	1 µs	Fat-tree
5	Perlmutter [2,6]	HPE	Gigabit Ethernet	Slingshot-10	400 Gbps	64	N/A	Dragonfly
6	Selene [2,18–20]	NVIDIA	InfiniBand	HDR InfiniBand	400 Gbps	40	N/A	Fat-tree
7	Tianhe-2A [2,12–14]	NUDT	Custom Interconnect	TH Express-2	224 Gbps	24	0.6 µs	Fat-tree
8	JUWELS Booster Module [2,18–20]	Atos	InfiniBand	HDR InfiniBand	400 Gbps	40	N/A	Fat-tree
9	HPC5 [2,18–20]	Dell EMC	InfiniBand	HDR InfiniBand	400 Gbps	40	N/A	Fat-tree
10	Voyager-EUS2 [2,18–20]	Microsoft Azure	InfiniBand	HDR InfiniBand	400 Gbps	40	N/A	Fat-tree
42	Tera1000–2 [2,15]	Atos	Custom Interconnect	Bull BXI1.2	200 Gbps	48	<1 µs	N/A
N/A	N/A	NVIDIA	InfiniBand	NDR InfiniBand [18–20]	800 Gbps	64	≤1 µs	N/A

Lu, Ping-Jing, Ming-Che Lai, and Jun-Sheng Chang. 2022. "A Survey of High-Performance Interconnection Networks in High-Performance Computer Systems" *Electronics* 11, no. 9: 1369.

概要

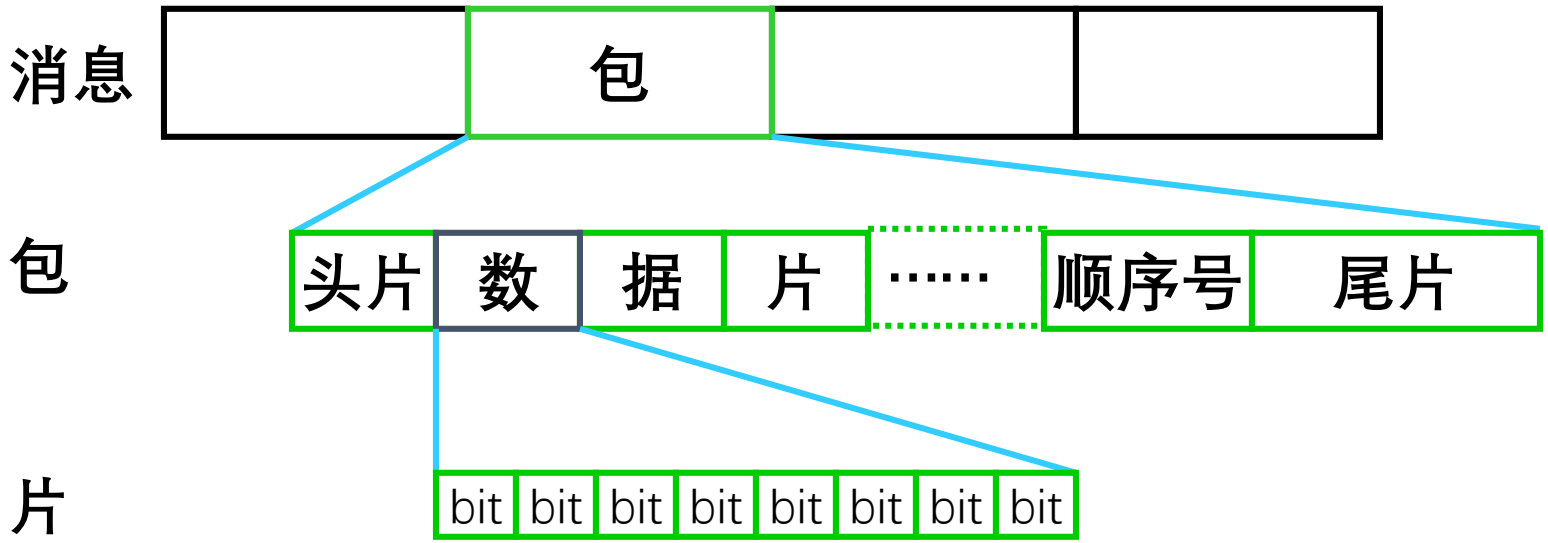
- 第二章 并行机系统互连与基本通信操作
  - 2.1 并行计算机互连网络
  - **2.2 选路方法与开关技术**
  - 2.3 单一信包一到一传输
  - 2.4 一到多播送
  - 2.5 多到多播送

预备知识

- 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

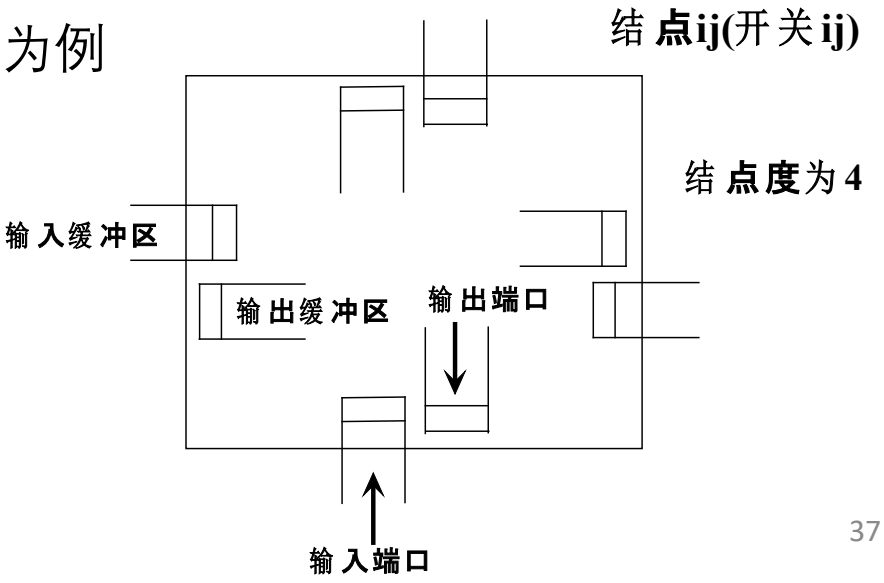
预备知识

- 消息、信包、片的相互关系



预备知识

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



预备知识

- 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* *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

- 概念分类

  - 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）

概要

- 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

- 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

- X-Y选路算法
  - 算法2.1：二维网孔上的X-Y选路算法

begin

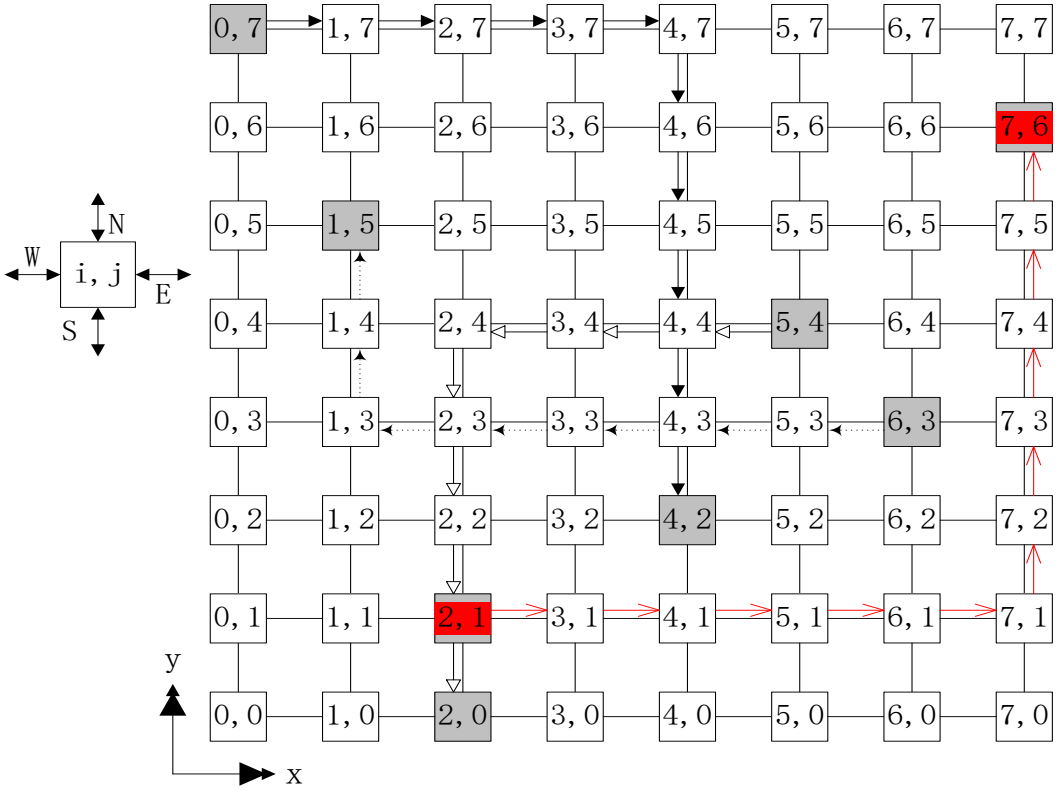
step1: 沿X方向将信包送至目的地处理器所在的列

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

- X-Y选路算法



4(源;目的)对: (2, 1; 7, 6) → (5, 4; 2, 0) ⇢

(0, 7; 4, 2) → (6, 3; 1, 5) ⇢

选路方法

- E-立方选路算法
  - 路由计算：

$s_{n-1}s_{n-2}\cdots s_1s_0$(源地址)

异或 $\oplus$

$d_{n-1}d_{n-2}\cdots d_1d_0$(目的地址)

⬇

$r_{n-1}\,r_{n-2}\cdots r_1\,r_0$(路由值)

  - 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

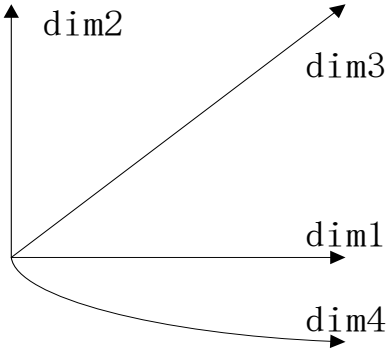
选路方法

- E-立方选路算法

0110(S)

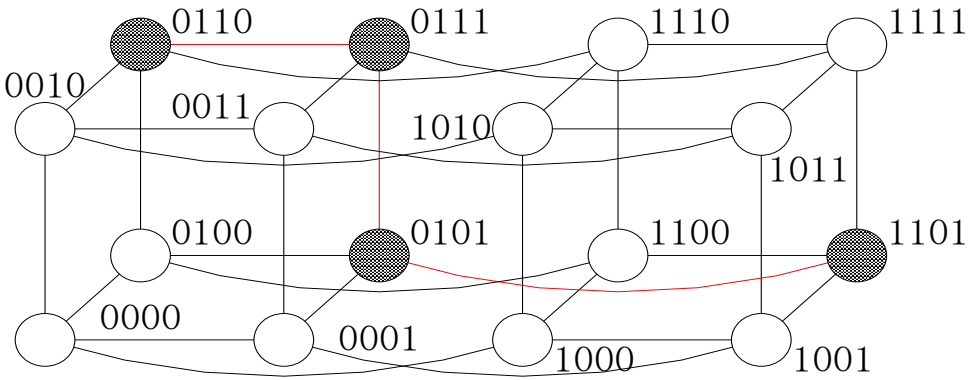
1101(D)

1011(R)



源： S=0110
目的： D=1101
路径：
0110 → 0111 → 0101 → 1101

异或结果为1的维度翻转为0则不变



开关技术

- 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: $t_{comm}(SF) = t_s + (mt_w + t_h)l = O(ml)$

启动时间t_s
节点延迟时间t_h
字传输时间t_w
链路数l
信包大小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

- 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: $t_{comm}(CT) = t_s + mt_w + lt_h = O(m+l)$

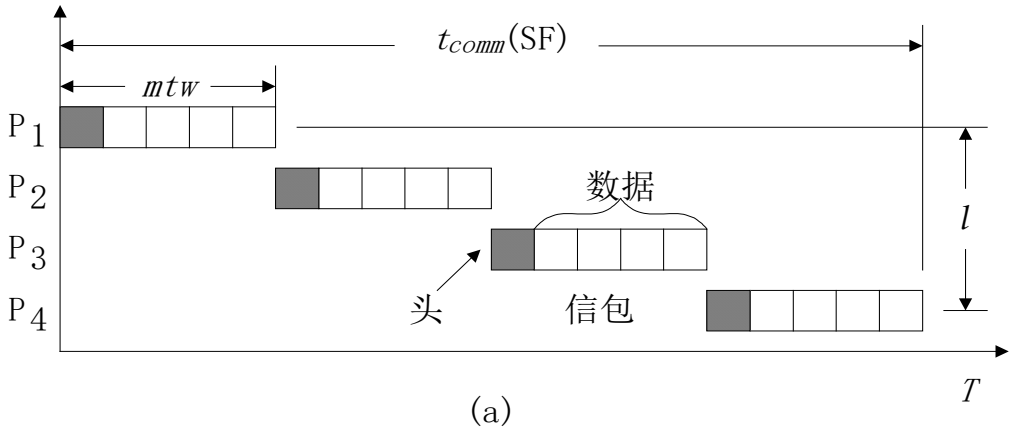
启动时间t_s
节点延迟时间t_h
字传输时间t_w
链路数l
信包大小m

缺点:

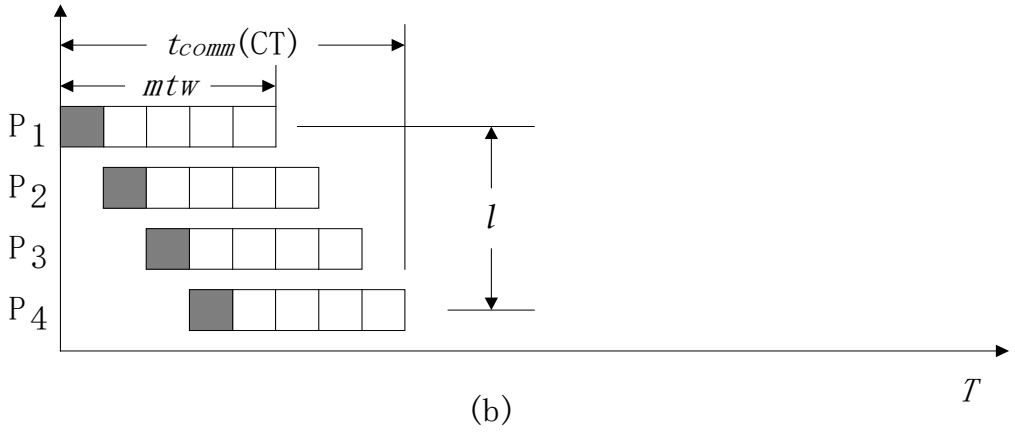
- 物理通道非共享
- 传输过程中物理通道一直被占用

开关技术

- 存储转发（SF）和切通（CT）对比



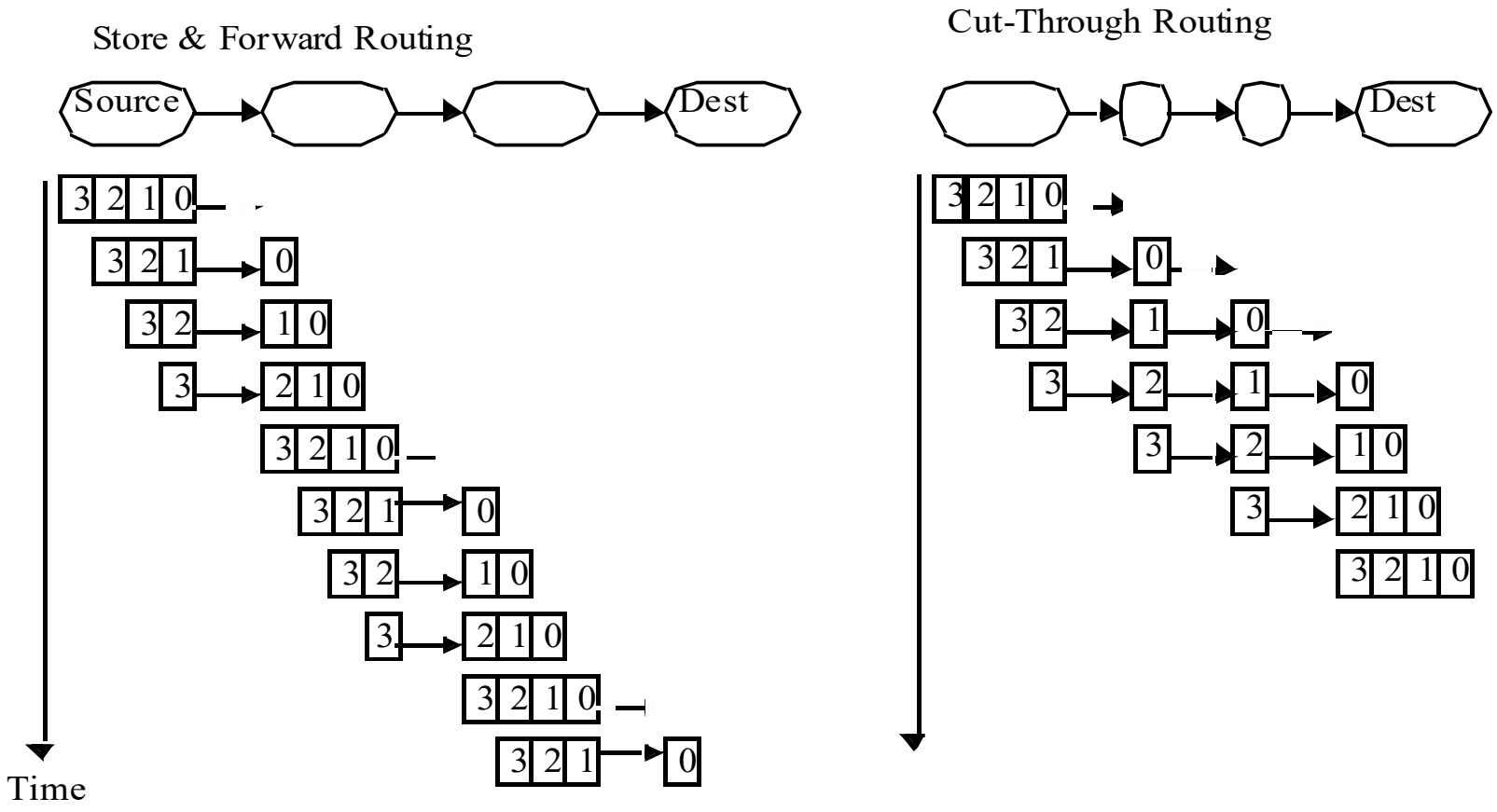
(a)



(b)

开关技术

- 存储转发（SF）和切通（CT）对比



开关技术

- 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

- 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。

概要

- 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：$l \le \lfloor p/2 \rfloor$
  - 带环绕Mesh($\sqrt{p} \times \sqrt{p}$)：$l \le 2\lfloor \sqrt{p}/2 \rfloor$
  - 超立方：$l \le \log p$
- t_{comm}(SF)的计算
  - 一维环形：$t_{comm}(SF) = t_s + t_w \cdot m \cdot \lfloor p/2 \rfloor$
  - 带环绕Mesh：$t_{comm}(SF) = t_s + 2t_w \cdot m \cdot \lfloor \sqrt{p}/2 \rfloor$
  - 超立方：$t_{comm}(SF) = t_s + t_w \cdot m \cdot \log p$
- t_{comm}(CT)的计算

$$t_{comm}(CT) = t_s + mt_w$$

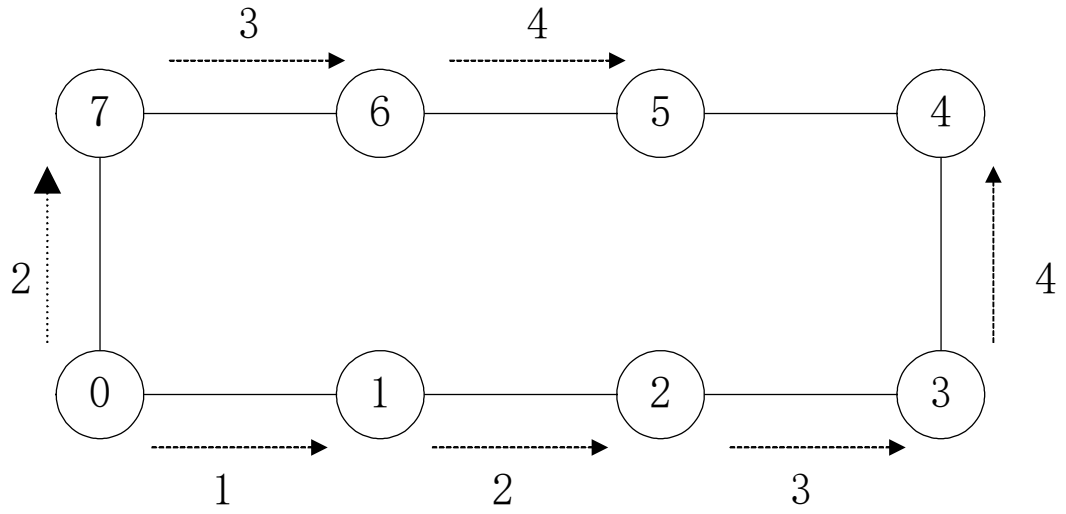
- 如果m>>p：$t_{comm}(SF) \approx t_{comm}(CT) = t_s + mt_w$

概要

- 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

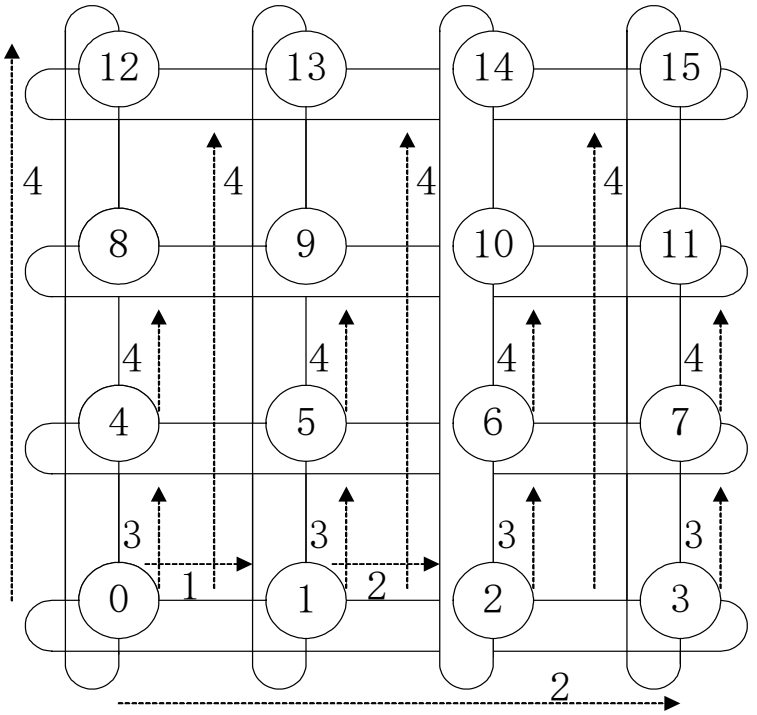
- 环
  - 步骤：　①先左右邻近传送;②再左右二个方向同时播送
  - 示例：



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

一到多播送—SF模式

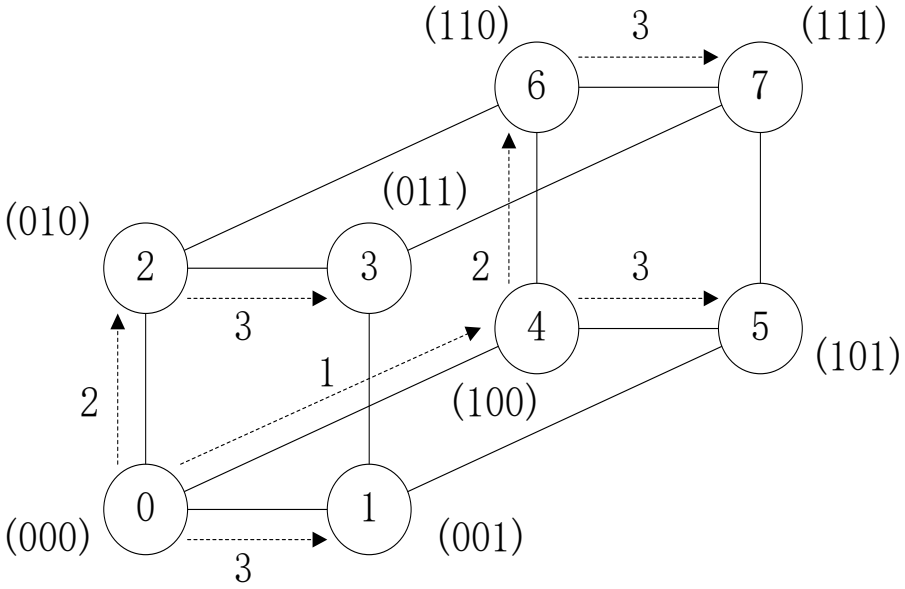
- 环绕网孔
  - 步骤：①先完成一行的播送;②同时进行各列的播送
  - 示例：
    共4步(2步行、2步列)



- 通信时间：$t_{one-to-all}(SF) = 2(t_s + mt_w)\left\lceil \frac{\sqrt{p}}{2} \right\rceil$

一到多播送—SF模式

- 超立方
  - 步骤：从低维到高维，依次进行播送;
  - 示例：



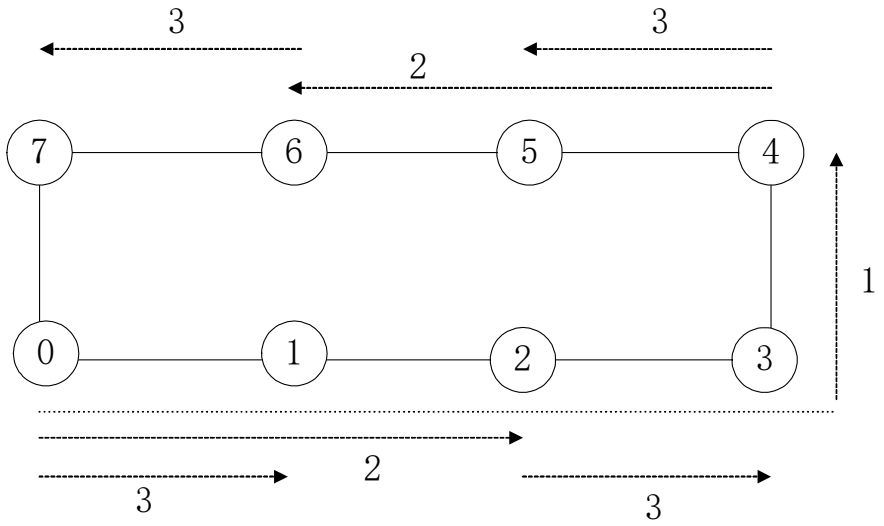
- 通信时间：

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

一到多播送—CT模式

- 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;



- 通信时间:

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

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

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

一到多播送—CT模式

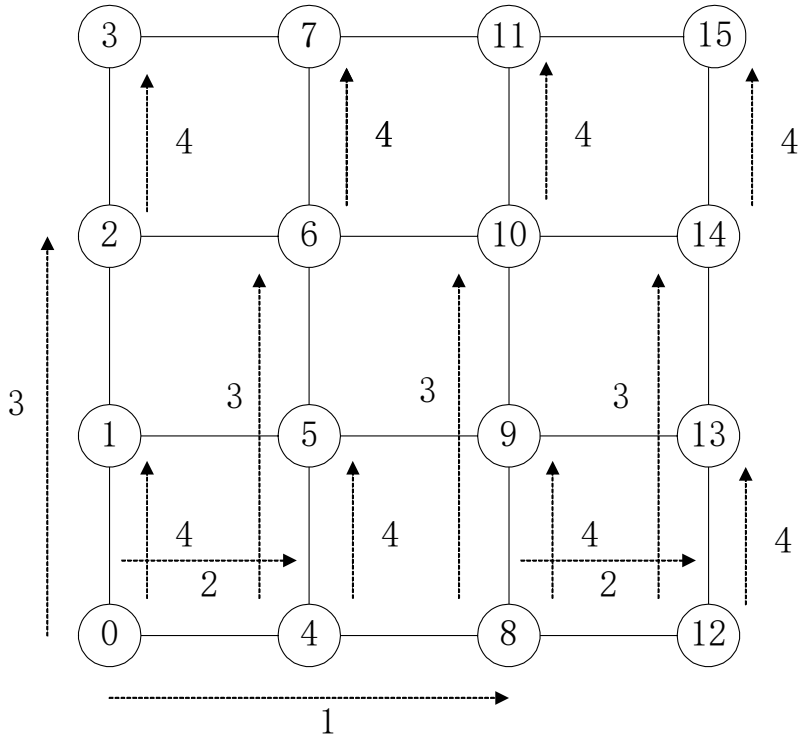
- 环绕网孔
  - 步骤:

(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;

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



- 通信时间: $t_{one-to-all}(CT) = 2(t_s \log\sqrt{p} + mt_w \log\sqrt{p} + t_h(\sqrt{p}-1))$

$$= (t_s + mt_w)\log p + 2t_h(\sqrt{p}-1)$$

一到多播送—CT模式

- 超立方
  - 步骤：从低维到高维，依次进行播送;
  - 通信时间： $t_{one-to-all}(CT) = (t_s + mt_w)\log p$
- 小结

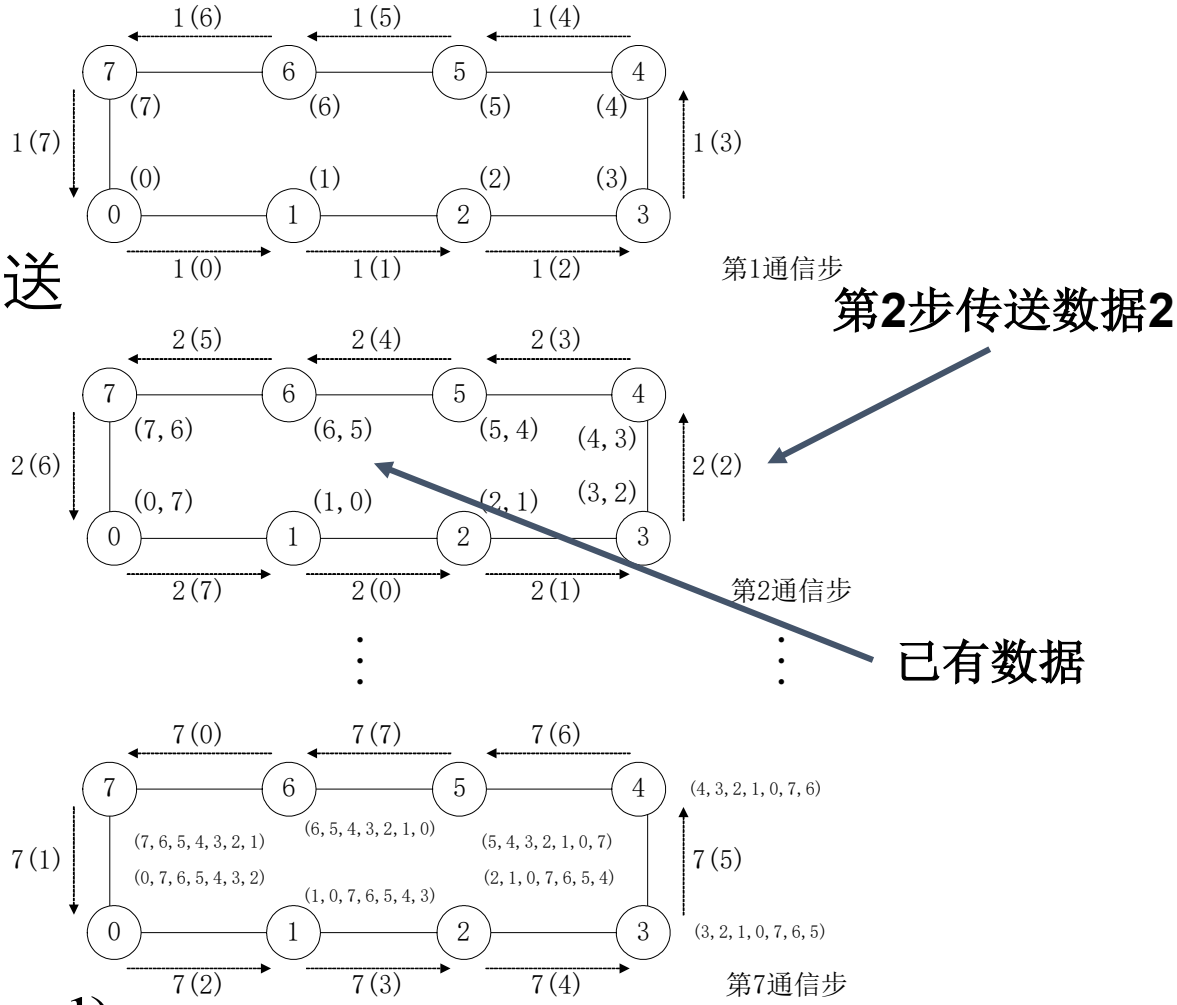
	环	网孔	超立方
$t_{one-to-all}(SF)$	$(t_s + mt_w)\lceil p/2 \rceil$	$2(t_s + mt_w)\lceil \sqrt{p}/2 \rceil$	$(t_s + mt_w)\log p$
$t_{one-to-all}(CT)$		$(t_s + mt_w)\log p$	

概要

- 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

- 环
  - 步骤:
    同时向右(或左)播送
    刚接收到的信包



  - 通信时间:

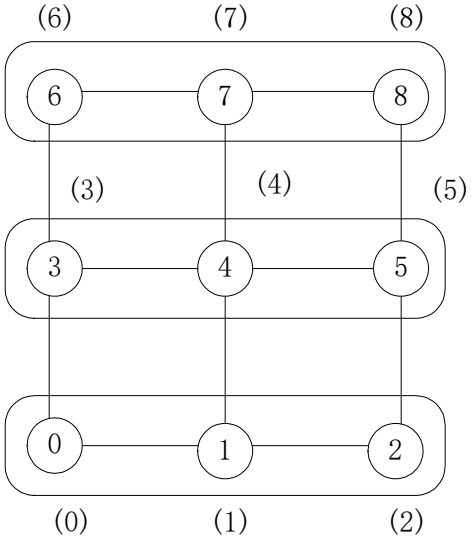
$$t_{all-to-all}(SF) = (t_s + mt_w)(p-1)$$

多到多播送—SF模式

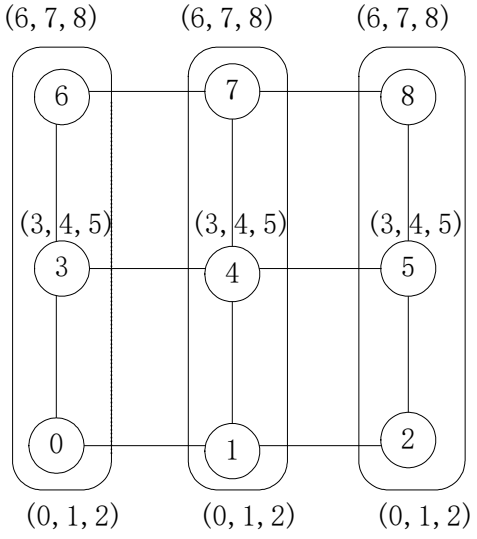
- 环绕网孔
  - 步骤:

(1)先进行行的播送;

(2)再进行列的播送;



(a)



(b)

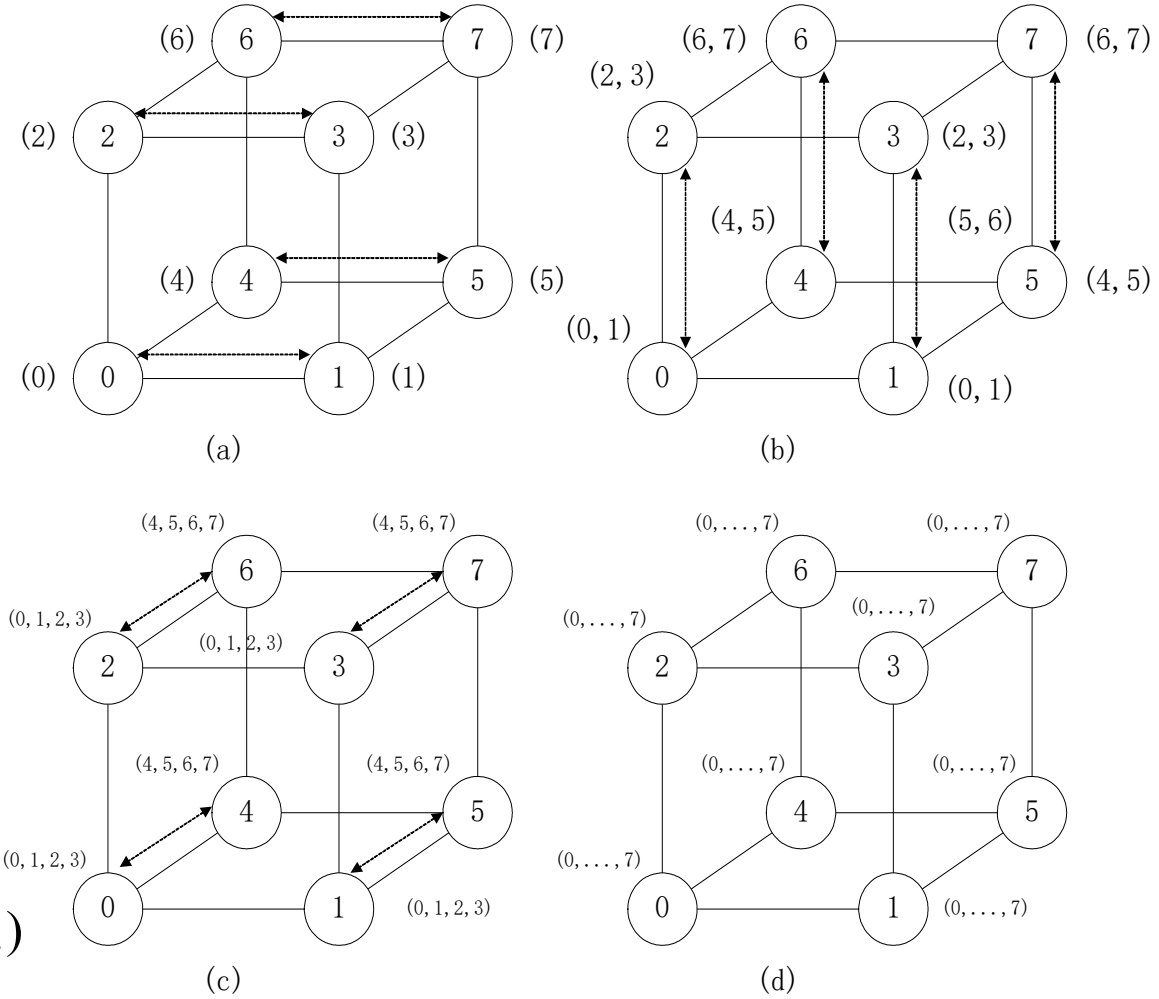
  - 通信时间:

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

$$= 2t_s(\sqrt{p} - 1) + mt_w(p - 1)$$

多到多播送—SF模式

- 超立方
  - 步骤:
    依次按维进行
    多到多的播送



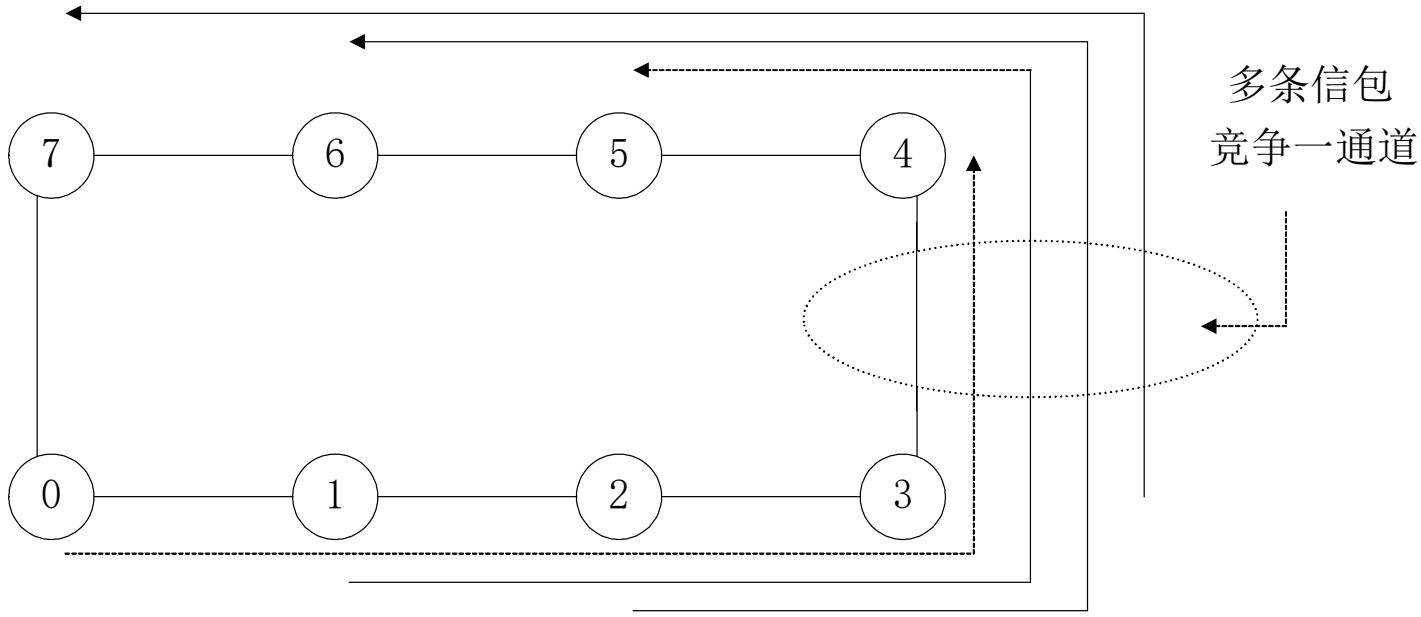
- 通信时间:

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

$$= t_s \log p + m t_w (p-1)$$

多到多播送—SF模式

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



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