BOIN: 一种新型无缓存高性能计算机光互连网络

齐星云, 窦 强, 陈永然, 钱 悦, 杨 威, 窦文华
(国防科学技术大学计算机学院, 湖南长沙 410073)

摘 要: 现有的高性能计算机光互连网络大都需要对数据报文进行光-电-光转换并缓存, 或者需要预先申请并建立从源节点到目的节点的光链路, 这在一定程度上限制了网络性能. 提出了一种既不需要对光数据报文进行缓存排队, 也不需要预先申请并建立光链路的新型的光互连网络结构 BOIN. 给出了 BOIN 网络上的无死锁和无活锁路由算法, 推导出 BOIN 网络中数据传输的延时上界, 并通过模拟实验对比了 BOIN 网络与其它两种典型的网络的性能. 实验结果表明, BOIN 光互连网络具有良好的延时和吞吐率特性, 能够很好地满足高性能计算机系统的要求.

关键词: 光互连网络; 无缓存; 无死锁路由; 无活锁路由; 网络性能
中图分类号: TP303 文献标识码: A 文章编号: 0372-2112 (2008) 11-2171-07

BOIN: A Novel Bufferless Optical Interconnection Network for High Performance Computer

QI Xing-yun, DOU Qiang, CHEN Yong-ran, QIAN Yue, YANG Wei, DOU Wen-hua
(*School of Computer Science, National University of Defense Technology, Changsha, Hunan 410073, China*)

Abstract: Most of the revealed optical interconnect networks within the high performance computers require either the buffering and O-E-O conversion of the data packets or the pre-assigning of the optical path from the source to destination, which to a certain extent influence the performance metrics such as latency and throughput. Aiming at the limitation mentioned above, a novel Bufferless Optical Interconnection Network (BOIN) for high performance computer is brought forward. Together with the data link protocol, deadlock freedom and livelock freedom routing algorithms, the upper bound of the data transmission latency in BOIN are also presented. The simulation is based on the comparison of BOIN and the other two similar networks. The results show that BOIN has the advantage over the rest that can deliver high throughput at low latency, which can well satisfy the need of high performance computing systems.

Key words: optical interconnect networks; bufferless; livelock freedom routing; deadlock freedom routing; network performance

1 引言

随着数值天气预报、地质石油勘探、生命科学以及大规模并行仿真等高性能计算应用的迅猛发展, 人们对并行计算机系统计算能力的要求越来越高. 当前, 限制并行计算机性能的主要因素有三个方面: 计算性能、存储性能和互连性能[1]. 由于当今高性能计算机大都采用并行体系结构, 多个处理节点通过高性能的互连网络连接起来, 并行执行计算任务, 因此各处理节点之间互连网络性能的好坏直接影响到系统整体性能的高低. 传统的电互连方式由于存在带宽低、功耗大、互连密度小、抗干扰能力差等不足, 已经限制了系统互连性能的进一步提高. 自 Goodman J. W. 提出在 VLSI 中采用光互连方案以来[2], 光互连技术已经取得巨大的进步, 并开始在计算机和通信系统中代替电互连技术. 光互连作为提高数据传输带宽的一种有效方法已经被广泛接受, 并得到迅速发展[3~5].

由于当前技术水平的限制, 计算机光互连系统面临两个最大的困难[6,7]: 缺乏有效的光缓冲机制, 以及不能对光数据信息直接进行逻辑处理. 这两个难点是制约光互连技术进一步发展的主要因素. 一种常用的解决方法是采用光电混合互连技术, 利用光信号进行相邻两个节点之间的数据传输, 而在每个中间节点上将光信号转换为电信号, 并对其进行缓冲和路由判断, 再将电信号

收稿日期: 2008-01-02; 修回日期: 2008-06-05
基金项目: 国家自然科学基金(No. 60633050, No. 60603061)

转换为光信号送至下一跳链路上[3]. 这种方法中的光电转换和缓存排队限制了互连网络的性能,不能充分发挥光互连高带宽低延时的优势. 还有一种方法是采用“请求-应答-传输”的机制, 由源端申请一条从源到目的的传输路径, 如果请求成功, 目的端返回给源端一个应答, 源端收到应答之后, 在建立好的路径上传输数据, 完成后由目的端再撤销该条路径, 以便其他节点可以继续申请. 在这种方式中, 请求、应答以及链路撤销等过程带来的额外开销从一定程度上降低了网络的性能.

本文针对上述问题,结合当前高速发展的光电子器件技术的成果[8,9], 提出了一种无缓存的光互连网络结构 BOIN(Bufferless Optical Interconnection Network) , 该结构可充分发挥光互连系统高带宽低延时的优势, 同时避免在中间节点对光数据进行光电转换和缓存, 极大地提高了网络的吞吐率, 降低了网络延时.

2 BOIN 网络结构

2.1 网络拓扑结构

BOIN 网络中的基本功能节点为交换节点, 每个交换节点由 2×2 的光开关和处理节点构成.

定义 **1** 一个 2×2 的光开关 $S=(V_S, E_S, ST)$ 为一个有向图, 其中 $V_S=\{X-, Y-, X+, Y+\}$ 为 S 的顶点集合, $E_S\subseteq\{(x, y)\mid x, y\in V_S\}$ 为 S 的边集合, ST 为开关 S 当前的状态, $ST\in\{ON, OFF\}$. 为了具体指明某个开关 S 的顶点, 可用 $X_-(S), X_+(S), Y_-(S), Y_+(S)$ 来分别表示 S 的 $X-$, $X+$, $Y-$, $Y+$ 顶点.

其中 E_S 的定义如下:

$$E_S=\begin{cases}\{(X-, Y+), (Y-, X+)\}, & \text{if } ST=OFF\\ \{(X-, X+), (Y-, Y+)\}, & \text{if } ST=ON\end{cases}$$

令 Switch 表示所有 2×2 交换开关的集合. 当一个 2×2 交换开关为 ON 状态时, 从输入端口到达的数据不改变方向直接从对应的输出端口送出; 当其为 OFF 状态时, 到达输入端口的数据需要改变方向再送出. 其结构如图 1 所示.

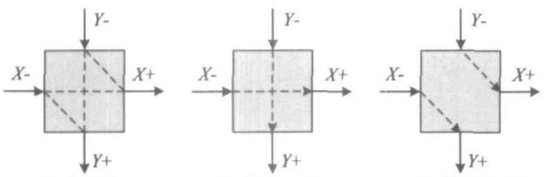

图1 2×2光开关

定义 **2** 一个交换节点 $N=(V_N, E_N)$ 是一个有向图, 其中 $V_N=\{S_{X+}, S_{Y+}, S_{X-}, S_{Y-}, S_F, S_S, P_X, P_Y\}$ 为 N 的顶点集合, $S_i\in Switch$, $i\in\{X+, Y+, X-, Y-, F, S\}$, P_j 为处理节点, $j\in\{X, Y\}$. $E_N=\{(X_+(S_{X-}), X_-(S_F)), (X_+(S_F), X_-(S_{X+})), (Y_+(S_{Y-}), Y_-(S_F)), (Y_+(S_F), Y_-(S_{Y+})), (Y_+(S_{X-}), X_-(S_S)), (X_+(S_S), P_X), (X_+(S_{Y-}), Y_-(S_S)), (Y_+(S_S), P_Y), (P_X, Y_-(S_{X+})), (P_Y, X_-(S_{Y+}))\}$ 为 N 的边集合. 令 $Node$ 表示所有交换节点的集合, 同时令 $X_-(N)=X_-(S_{X-})$, $X_+(N)=X_+(S_{X+})$, $Y_-(N)=Y_-(S_{Y-})$, $Y_+(N)=Y_+(S_{Y+})$.

一个交换节点 N 包括 6 个 2×2 的光开关和 2 个处理节点, 其相互间使用光链路连接起来. 交换节点对外具有 2 个输入端口 $X-$和 $Y-$, 以及 2 个输出端口 $X+$ 和 $Y+$, 可以将内部 2 个处理节点中的数据报文分别送至 $X+$ 方向和 $Y+$ 方向, 同时接收从 $X-$方向和 $Y-$方向到达的数据报文. 根据所处位置和功能的不同, 交换节点内部的光开关可以分为 3 种类型: 转发开关, 端口开关和接收开关. 转发开关 S_F 负责将从输入端口到达的数据报文转发到输出端口上. 4 个端口开关 S_{X-}, S_{X+}, S_{Y-}, S_{Y+} 控制本交换节点在 X_-, Y_-, X_+, Y_+ 四个方向的输入和输出, 接收开关 S_S 负责将到达目的地的报文送给最终的处理节点. 一个交换节点的内部结构如图 2 所示.

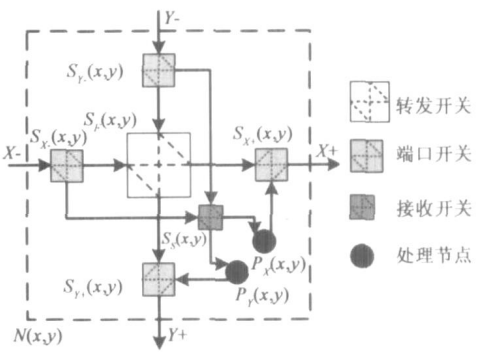

图2 交换节点结构

定义 **3** 一个 $m\times n$ 的 BOIN 网络 $B=(V_B, E_B)$ 是一个有向图, 其中 $V_B=\{N(x, y)\mid N(x, y)\in Node, 0\leqslant x\leqslant m-1, 0\leqslant y\leqslant n-1\}$ 为 B 的顶点集合, $E_B=\{(X_+(N(x, y)), X_-(N((x+1)\bmod m, y))), (Y_+(N(x, y)), Y_-(N(x, (y+1)\bmod n)))\mid 0\leqslant x\leqslant m-1, 0\leqslant y\leqslant n-1\}$ 为 B 的边集合.

一个 $m\times n$ 的 BOIN 网络由 mn 个交换节点 $N(x, y)(x=0, 1, 2, \cdots, m-1, y=0, 1, 2, \cdots, n-1)$ 及相互间的光链路组成, mn 个交换节点通过两个输入端口和两个输出端口使用沿 $X+$ 方向和 $Y+$ 方向的单向光链路相互连接起来. 图 3 给出了一个 4×4 的 BOIN 网络的结构.

2.2 网络链路控制协议

在 BOIN 网络中, 任何两个相邻交换节点之间的链路都是等长的. 设相邻两个节点之间的光链路长度为 L, 链路带宽为 B, 链路上光速为 c, 则设定网络中每个

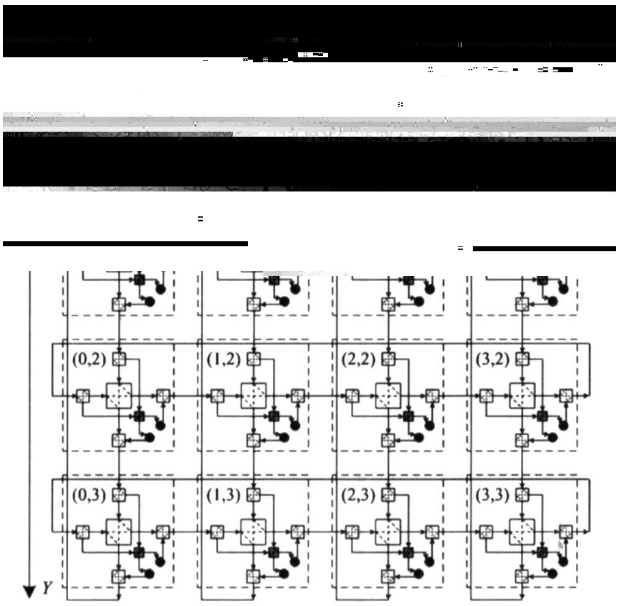

图3　4×4的BOIN网络结构

数据报文的长度为 $D= BL/c$. 报文中任何一位(不妨设第一位)在链路上传输延时为 $t_1= L/c$, 报文从第一位发出到最后一位发送完成所需要的发送延时为 $t_2= D/B= t_1$. 即是说, 当报文的第一位到达链路末端的时候, 报文的最后一位刚好被发送出来, 这样, 整个报文就完全被缓存在网络链路上, 而不需要在每个节点处对光数据信号进行光电转换然后再缓存.

令 $\Delta T= t_1= t_2$, 称 ΔT 为 BOIN 网络中的一个时隙(slot), 一个时隙的大小即为光信号在相邻两个交换节点之间的链路上的传输时间. 网络中所有数据都是按照时隙同步进行传输的. 在 BOIN 网络中, 相邻两个节点之间除了光数据链路之外, 还有电控制链路, 用来在相邻节点间传输控制报文, 设相邻节点间控制链路长度为 l. 由于在光链路上要缓存数据报文, 所以光链路的长度须满足 $L= Dc/B$. 然而电控制链路无此特殊要求, 故 l 可以非常短, 使得相邻节点间控制信号的传输延时为 $\Delta t= l/v$, 其中 v 为电信号的传输速度. 设电控制信号到达下一个节点后, 译码并设置其中光开关的状态的时间为 t_c, 则在 BOIN 网络中, 必须满足下面的条件:

$$\Delta t+ t_c< \Delta T$$

即

$$l/v+ t_c< D/B \qquad (1)$$

在 v, t_c 以及 B 为常数的时候, 选择适当的 l 和 D, 可以满足条件(1). 当该条件成立时, 电控制信号在到达下一个节点并根据报文的目的地址成功设定其中的光开关状态后, 光链路上的数据信号才会到达, 然后光数据报文和电控制报文再同时从输出端口发出. 这样, 电控制报文始终和光数据报文在网络中同步传输, 利用两者之间的传输延时差, 使得控制信号可以提前为数据报文的传输建立起下一步的传输链路.

在这种链路协议中, 需要网络中所有节点实现全局的时钟同步, 以便能够确定数据报文和控制报文的发送时间. 由于相邻节点间链路等长, 假设在节点内部数据传输的时间可以忽略不计, 这样, 从网络配置完成开始运行起, 所有的节点都有一个以 ΔT 为周期的时钟. 在工程实现中, 即使该时钟在长期运行后可能会有误差, 但每次有报文到达该节点之后就会根据到达的报文重新同步一次, 这样可以保证各节点上的时钟实现全局同步. 每个报文都在某个时钟开始时从某个节点发出第一位, 并在下一个时钟的开始完成最后一位的发送, 同时其第一位到达下一个节点. 这样的过程一直继续, 直到报文到达目的节点.

采用这种链路控制协议, 使得在不需要对网络数据报文进行缓存的情况下, 能够实现报文在光互连网络中的高速传输, 同时避免了由源端到目的端的链路请求所带来的网络延时和吞吐率的降低.

3　BOIN 网络中的无死锁/ 活锁路由

3.1　路由算法

在 BOIN 网络中, 报文的路由过程实际上就是报文在由光开关和光链路组成的路径上的传输过程. 因此, 对于任何一个交换节点 $C(x_C, y_C)$, 其上的路由算法即可用描述其中每个光开关状态的路由函数 $R(P, C)$ 和仲裁函数 $A(P_x, P_y, C)$ 来描述, 其中 P 表示从任何端口到达的一个报文, P_x, P_y 分别表示从 $X-$ 和 $Y-$ 端口到达的报文. 用 $D(P)$ 和 $S(P)$ 分别表示报文 P 的目的交换节点和源交换节点. 路由函数 $R(P, C)$ 根据当前节点 C 和目的节点 $D(P)$, 决定当前节点 C 中各个开关的状态. 如果同时到达的两个报文发生冲突, 则由仲裁函数 $A(P_x, P_y, C)$ 在这两个报文之间仲裁, 给出仲裁完成后各开关应该处于的状态. 对于 BOIN 网络中的任何报文 P. 令 $H(P)$ 表示其在网络中的跳步数, 即 $H(P)$ 的初始值为 0, 若 P 沿 $X+$ 方向或 $Y+$ 方向前进 1 步, 则 $H(P)= H(P)+ 1$. 在路由算法中, 我们以 $H(P)$ 的值来表示某个到来报文在冲突仲裁中的优先级. 同时规定, 如果 P 不存在, 则 $H(P)= 0$.

在 BOIN 网络中, 数据报文按照 XY 路由方式进行路由. 报文首先沿 $X+$ 方向前进, 当到达与目的节点具有相同 X 坐标的中间节点之后, 再转向 $Y+$ 方向路由. 如果在某个中间节点上, 从 $X-$ 端口和从 $Y-$ 端口到达的报文竞争同一个输出端口时, 则允许具有最大跳步数的报文从该端口输出, 而竞争失败的报文在网络中绕道前进, 等待下一次到达目的节点. 在 BOIN 网络中, 数据报文的路由分为两步: 第一步为将数据报文送达目的交换节点, 此过程由转发开关上的路由算法决定; 第二步

为将数据报文送达目的处理节点,此过程由接收开关和转发开关上的路由算法共同决定. 在有冲突产生时,这两个步骤可能交替出现. 下面我们分别给出三种不同的光开关上的路由算法.

3.1.1 端口开关 $\mathbf{S_X}$/ $\mathbf{S_Y}$ 上的路由算法

假设报文 P 从节点 C 的某个端口 $p(P, C)$ 到达某个中间节点 $C(x_C, y_C)$,其中 $p(P, C) \in \{X-, Y-\}$. 设报文 P 的目的节点 $D(P)$ 的坐标为 $(x_{D(P)}, y_{D(P)})$,且 $|x_{D(P)} - x_C| + |y_{D(P)} - y_C| > 0$. 由于一个交换节点内有 2 个处理节点,用 $D_P(P)$ 来表示报文 P 的最终目的处理节点. 则其端口开关 S_X 和 S_Y 上的路由函数可以表示为(其中 P_x 和 P_y 分别为从 $X-$ 端口和 $Y-$ 端口到达节点 C 的报文):

$$R_X(P_x, P_y, C) = \begin{cases} OFF, \begin{pmatrix} ((H(P_x) \geqslant H(P_y)) \text{AND} (x_{D(P_x)} = x_C) \text{AND} (y_{D(P_x)} = y_C)) \text{OR} \\ ((H(P_x) < H(P_y)) \text{AND} (x_{D(P_x)} = x_C) \text{AND} (y_{D(P_x)} = y_C) \text{AND} (D_P(P_x) \neq D_P(P_y))) \end{pmatrix} \\ ON, \text{else} \end{cases}$$

$$R_Y(P_x, P_y, C) = \begin{cases} OFF, \begin{pmatrix} ((H(P_y) > H(P_x)) \text{AND} (x_{D(P_y)} = x_C) \text{AND} (y_{D(P_x)} = y_C)) \text{OR} \\ ((H(P_y) \leqslant H(P_x)) \text{AND} (x_{D(P_y)} = x_C) \text{AND} (y_{D(P_y)} = y_C) \text{AND} (D_P(P_x) \neq D_P(P_y))) \end{pmatrix} \\ ON, \text{else} \end{cases}$$

从上述路由函数可以看出,一个已经到达目的交换节点的报文,能够被成功送到最终的处理节点的条件是其有较高的优先级,或者没有其他报文与其竞争同一个处理节点.

3.1.2 转发开关 $\mathbf{S_F}$ 上的路由算法

$$R_F(P, C) = \begin{cases} ON, (((p(P, C) = X-) \text{AND} (x_C \neq x_{D(P)})) \text{OR} \\ ((p(P, C) = Y-) \text{AND} (x_C = x_{D(P)}) \text{AND} (y_C \neq y_{D(P)}))) \\ OFF, \text{else} \end{cases}$$

路由函数 $R_F(P, C)$ 决定了当报文 P 到达节点 C 时,如果 C 不是 P 的目的节点,节点 C 上的光开关 S_F 应该处于何种状态.

若两个报文 P_x, P_y 在 BOIN 网络中发生冲突时,需要通过选择函数 $A_F(P_x, P_y, C)$ 来决定转发开关的最终状态,消除冲突.

$$A_F(P_x, P_y, C) = \begin{cases} R_F(P_x, C), \begin{pmatrix} ((R_X(P_x, P_y, C) = ON) \text{AND} \\ (R_Y(P_x, P_y, C) = OFF)) \text{OR} \\ ((H(P_x) \geqslant H(P_y)) \text{AND} \\ (R_X(P_x, P_y, C) = ON) \\ \text{AND}(R_Y(P_x, P_y, C) = ON)) \end{pmatrix} \\ R_F(P_y, C), \text{else} \end{cases}$$

从 $A_F(P_x, P_y, C)$ 的定义可以看出,在产生冲突时,网络中具有最大跳步数的报文具有最高的优先权;当两个报文的跳步数相等时,从 X-方向到达的报文具有最高的优先权.

3.1.3 接收开关 $\mathbf{S_S}$ 上的路由算法

当报文 P 到达开关 $S_S(x_C, y_C)$ 时,其路由函数为:

$$R_S(P, C) = \begin{cases} ON, \begin{pmatrix} ((p(P, C) = X-) \text{AND} (D_P(P) = P_x(x_C, y_C)) \text{OR} \\ ((p(P, C) = Y-) \text{AND} (D_P(P) = P_y(x_C, y_C))) \end{pmatrix} \\ OFF, \text{else} \end{cases}$$

若两个报文 P_x 和 P_y 同时到达节点 C 的 $X-$ 端口和 $Y-$ 端口,并竞争同一个处理节点,则由 S_X 和 S_Y 进行冲突避免处理,使得竞争失败的报文在网络绕道前进,从而最终在 S_S 上不会有冲突发生. 故其选择函数为:

$$A_S(P_x, P_y, C) = R_S(P_x, C)$$

3.2 路由算法的特性

对于网络中的任何一个中间节点 C,有 2 个输入端口 $X-$, $Y-$ 和 2 个输出端口 $X+$, $Y+$. 如果某个到来的报文 P 的目的节点不是 C,则无论节点 C 中的各个交叉开关处于何种状态,P 肯定有一个输出通道;如果节点 C 是 P 的目的节点,则或者 P 被成功接收,或者 P 与别的报文发生冲突而从节点 C 的某个输出端口送出,在网络中绕道前进. 因此,只要报文 P 未被成功接收,总可以在网络中前进,而不会被阻塞在某个节点上. 因此我们有如下结论:

结论 **1** BOIN 网络中不存在死锁.

下面我们将着重研究 BOIN 网络中是否存在活锁. 设某个报文 P 的目的节点为 $T(x_T, y_T)$. 由于当前网络中的报文是有限个,我们设当前网络中报文的最大跳步数为 H_m. 即 $H_m = \max\{H(P) \mid P \text{ 为当前网络中的报文}\}$.

结论 **2** 设报文 P 在某个时隙 t_0 到达某个中间节点 $C(x_C, y_C)$ 的 $X-$ 端口,$H(P) = H_m$,且 $x_C \neq x_T$,则 P 在 $t_1 = t_0 + 1$ 时隙到达节点 $C_1((x_C + 1) \bmod m, y_C)$ 的 X-端口.

证明:因为 $H(P) = H_m \geqslant H(P_y)$,且 $x_C \neq x_T$,故 $A_F(P, P_y, C) = R_F(P, C) = ON$,因此 P 将从 C 的 $X+$ 端口送出,并在 $t_1 = t_0 + 1$ 时隙到达节点 $C_1((x_C + 1) \bmod$

$m, y_C)$ 的 $X-$ 端口.

推论 **1**　设报文 P 在某个时隙 t_0 到达某个中间节点 $C(x_C, y_C)$ 的 $X-$ 端口, $H(P)=H_m$, 且 $x_C \neq x_T$, 则 P 在 $t=t_0+(x_T-x_C) \bmod m$ 时隙到达节点 $C_1(x_T, y_C)$ 的 $X-$ 端口.

证明: 由结论 2 知, 当 $x_C \neq x_T$ 时, 报文 P 始终沿 $X+$ 方向前进, 并且每个时隙前进一步, 由于节点 $C(x_C, y_C)$ 与节点 $C_1(x_T, y_C)$ 的 X 方向距离为 $(x_T-x_C) \bmod m$, 因此, 报文 P 在 $t=t_0+(x_T-x_C) \bmod m$ 时隙将到达节点 $C_1(x_T, y_C)$ 的 $X-$ 端口.

结论 **3**　设报文 P 在某个时隙 t_0 到达某个中间节点 $C(x_T, y_C)$ 的 $X-$ 端口, $H(P)=H_m$, 且 $y_C \neq y_T$, 则 P 在 $t=t_0+1$ 时隙到达节点 $C_1(x_T, (y_C+1) \bmod n)$ 的 $Y-$ 端口.

证明: 因为 $H(P)=H_m \geqslant H(P_y)$, 故 $A_F(P, P_y, C)=R_F(P, C)=OFF$, 因此 P 将从 C 的 $Y+$ 端口送出, 并在 $t_1=t=t_0+1$ 时隙到达节点 $C_1(x_T, (y_C+1) \bmod n)$ 的 $Y-$ 端口.

结论 **4**　设报文 P 在某个时隙 t_0 到达某个中间节点 $C(x_T, y_C)$ 的 $Y-$ 端口, $H(P)=H_m$, 且 $y_C \neq y_T$, 则 P 在 $t=t_0+1$ 或 $t=t_0+(m+1)$ 时隙到达节点 $C_1(x_T, (y_C+1) \bmod n)$ 的 $Y-$ 端口.

证明: 分两种情况.

(1) 如果在 t_0 时刻有报文 P' 到达节点 C 的 $X-$ 端口, P' 的目的节点为 $T'(x_T, y_T')$, $y_C \neq y_T'$, 且 $H(P')=H_m$, 则 $A_F(P', P, C)=R_F(P', C)=OFF$. 此时, P 将从 C 的 $X+$ 端口送出, 经过 1 个时隙到达节点 $C'((x_T+1) \bmod m, y_C)$ 的 $X-$ 端口. 由推论 1 知, 报文 P 到达节点 C' 后, 将经过 $((x_T-(x_T+1) \bmod m) \bmod m)=m-1$ 个时隙到达节点 $C(x_T, y_C)$ 的 $X-$ 端口. 由结论 3 知, P 还需要 1 个时隙可以到达节点 $C_1(x_T, (y_C+1) \bmod n)$ 的 $Y-$ 端口. 故在此情况下, 报文 P 共需要 $1+(m-1)+1=(m+1)$ 个时隙, 于 $t=t_0+(m+1)$ 时刻到达节点 C_1 的 $Y-$ 端口.

(2) 否则, $A_F(P_x, P, C)=ON$. 此时, P 将从 C 的 $Y+$ 端口送出, 经过 1 个时隙于 $t=t_0+1$ 时隙到达节点 $C_1(x_T, (y_C+1) \bmod n)$ 的 $Y-$ 端口.

推论 **2**　设报文 P 在某个时隙 t_0 到达某个中间节点 $C(x_T, y_C)$ 的 $Y-$ 端口, 且 $H(P)=H_m$, 则存在 $k \in \{0, 1, \cdots, (y_T-y_C) \bmod n\}$, 使得在时隙 $t=t_0+((y_T-y_C) \bmod n+km)$ 到达其目的节点 (x_T, y_T) 的 $Y-$ 端口.

证明: 由结论 4 知, 当报文 P 到达节点 (x_T, y_C) 的 $Y-$ 端口后, 每在 $Y+$ 方向前进 1 步, 可能需要在网络中多走 m 步, 由于节点 $C(x_T, y_C)$ 与目的节点 $T(x_T, y_T)$ 的 $Y+$ 方向距离为 $(y_T-y_C) \bmod n$, 因此需要多走 k 个 m 步才能到达目的节点, $k=0, 1, \cdots, (y_T-y_C) \bmod n$. 故最终达到目的节点 (x_T, y_T) 的 $Y-$ 端口的时隙为 $t=t_0+((y_T-y_C) \bmod n+km)$.

结论 **5**　设报文 P 在某个时隙 t_0 到达某个中间节点 $C(x_C, y_C)$ 的 $X-$ 端口, 且 $H(P)=H_m$, 则存在 $k \in \{0, 1, \cdots, (y_T-y_C) \bmod n\}$, 使得最多经过 $((x_T-x_C) \bmod m+(y_T-y_C) \bmod n+km)$ 个时隙 P 将到达目的节点 (x_T, y_T).

证明: 由推论 1 知 P 将在 $t_0+((x_T-x_C) \bmod m)$ 个时隙到达节点 $C_1(x_T, y_C)$ 的 $X-$ 端口. 若 $y_C=y_T$, 显然已到达目的节点. 若 $y_C \neq y_T$, 由结论 3 知, P 在 $t_0+((x_T-x_C) \bmod m+1)$ 时隙到达节点 $(x_T, (y_C+1) \bmod n)$ 的 $Y-$ 端口. 由推论 2 知, 存在 $k \in \{0, 1, \cdots, (y_T-y_C-1) \bmod n\}$, 使得 P 在时隙 $t_0+((x_T-x_C) \bmod m+1)+((y_T-y_C-1) \bmod n+km)=t_0+((x_T-x_C) \bmod m+(y_T-y_C) \bmod n+km)$ 时到达目的节点 (x_T, y_T). 故报文最多经过 $((x_T-x_C) \bmod m+(y_T-y_C) \bmod n+km)$ 个时隙可到达目的节点.

结论 **6**　对于任何报文 P, 如果 $H(P)=H_m$, 则最多经过 $(2mn+2n-3)$ 个时隙, P 将到达其目的交换节点 (x_T, y_T).

证明: 如果 P 在某个时隙 t_0 到达某个中间节点 $C(x_C, y_C)$ 的 $X-$ 端口, 由结论 5 知, P 将在 $\max\{((x_T-x_C) \bmod m+(y_T-y_C) \bmod n+km) \mid x_C, x_T \in \{0, 1, \cdots, m-1\}, y_C, y_T \in \{0, 1, \cdots, n-1\}, k \in \{0, 1, \cdots, (y_T-y_C) \bmod n\}\}=(mn+n-2)$ 个时隙内到达其目的节点. 否则, 如果 P 此前一直在 Y 方向上前进, 由于当前网络中在 X 方向传输且跳步数为 H_m 的报文为有限个, 同理可知, 这些报文将在 $(mn+n-2)$ 个时隙内到达其目的节点并从网络上消失. 之后, 如果 P 还在 Y 方向传输, 到达某个中间节点 $C'(x_{C'}, y_{C'})$ 的 $Y-$ 端口, 若 $x_{C'}=x_T$, 则由推论 2 知, P 最多经过 $\max\{((y_T-y_{C'}) \bmod n+km) \mid y_{C'}, y_T \in \{0, 1, \cdots, n-1\}, k \in \{0, 1, \cdots, (y_T-y_{C'}) \bmod n\}\}=(n-1)(m+1)$ 个时隙可到达其目的节点; 若 $x_{C'} \neq x_T$, 则在节点 C' 上, 由于 $H(P)=H_m>H(P_x)$, $A_F(P_x, P, C)=R_F(P, C)=OFF$, P 将从 C' 的 $X-$ 端口发出, 经过 1 个时隙到达 $((x_{C'}+1) \bmod m, y_{C'})$ 的 $X-$ 端口, 由结论 5 知, 存在 $k' \in \{0, 1, \cdots, (y_T-y_{C'}) \bmod n\}$, 使得 P 最多经过 $\max\{((x_T-x_{C'}-1) \bmod m+(y_T-y_{C'}) \bmod n+km) \mid x_{C'}, x_T \in \{0, 1, \cdots, m-1\}, y_{C'}, y_T \in \{0, 1, \cdots, n-1\}, k \in \{0, 1, \cdots, (y_T-y_{C'}) \bmod n\}\}=(mn+n-2)$ 个时隙到达目的节点. 由上, 无论当前 P 在网络的 X 方向还是 Y 方向前进, 如果 $H(P)=H_m$, 且当前 P 到达节点 $C(x_C, y_C)$, 则 P 在有限的时间 $\max\{(mn+n-2), (mn+n-2)+(n-1)(m+1), (mn+n-2)+1+(mn$

$+ n - 2)\} = 2mn + 2n - 3$ 个时隙内必定可到达其目的节点.

结论 **7** 对于任何报文 P, 如果 $H(P) = H_m$, 则最多经过 $(2mn + m + 2n - 3)$ 个时隙, P 将到达其最终目的处理节点 $(x_T, y_T, D_P(P))$.

证明: 由结论 6 可知, 最多经过 $(2mn + 2n - 3)$ 个时隙, P 肯定会到达节点 (x_T, y_T). 如果 P 从 $X-$ 端口到达节点 (x_T, y_T), 则无论是否有报文从 $Y-$ 端口到达, 由 S_X 上的路由算法可知, P 都具有最高的优先权, 故 P 将在 $(2mn + 2n - 3)$ 个时隙内到达最终的处理节点 $(x_T, y_T, D_P(P))$. 如果 P 从 $Y-$ 端口到达节点 (x_T, y_T), 由 S_Y 上的路由算法可知, P 或者直接经过 S_Y 到达最终的处理节点 $(x_T, y_T, D_P(P))$, 或者从节点 (x_T, y_T) 的 $X+$ 端口送出, 再经过 m 个时隙才能到达最终的处理节点 $(x_T, y_T, D_P(P))$. 因此报文 P 在到达目的节点 (x_T, y_T) 之后, 还最多需要经过 m 个时隙, 才能送到最终的处理节点 $(x_T, y_T, D_P(P))$. 故 P 最多需要 $(2mn + m + 2n - 3)$ 个时隙, 就可以从网络上任意一点到达最终的处理节点 $(x_T, y_T, D_P(P))$.

结论 **8** 对于任何报文 P, 最大经过 $2mn(2mn + m + 2n - 3)$ 个时隙, P 将到达其目的节点.

证明: 令 $H = H_m - H(P)$, 下面用用归纳法证明最多经过 $(2mn + m + 2n - 3)(H + 1)$ 个时隙, P 将到达其目的节点.

如果 $H = 0$, 即 $H(P) = H_m$, 由结论 7 可知, 最多经过 $(2mn + m + 2n - 3)$ 个时隙, P 将到达其目的节点, 结论成立.

如果 $H = i(i \geqslant 0)$ 时结论成立, 则当 $H = i + 1$ 时, 首先最多经过 $(2mn + m + 2n - 3)(i + 1)$ 个时隙, 对于任意报文 P', 如果 $H(P') > H(P)$, P' 将全部到达其目的处理器节点并被接收. 此时, 再最多经过 $(2mn + m + 2n - 3)$ 个时隙, P 将到达其目的处理节点. 故共经过 $((i + 1) + 1)(2mn + m + 2n - 3)$ 个时隙, P 到达其目的节点.

即对于任意节点 P, 最多经过 $(2mn + m + 2n - 3)(H + 1)$ 个时隙, P 将到达其目的处理节点并被接收. 由于网络中共有 $2mn$ 条链路, 故当前网络中包括报文 P 最多有 $2mn$ 个报文, 因此 $H = H_m - H(P) \leqslant 2mn - 1$. 所以最多经过 $2mn(2mn + m + 2n - 3)$ 个时隙 P 将会到达其目的处理节点.

由结论 8, 我们知道在 BOIN 网络中, 任何报文都可以在有限时间内到达其目的节点. 因此, 在前面所述算法下, BOIN 网络无活锁.

结论 8 给出了一个数据报文在到达目的处理节点之前在 BOIN 网络中传输所需时间的上限, 这一结论具有重要意义. 在很多高性能计算应用中, 各处理器之间的最大通信延时都有严格的时间限制. 当采用 BOIN 网络作为互连网络时, 由于其数据传输延时具有确定的上限, 这对评估系统性能、确保报文的可达性、发现并解决系统性能瓶颈具有重要意义.

4 BION 网络性能分析

本节通过模拟实验研究了 BOIN 网络在不同条件下的延时和吞吐率特性. 我们通过模拟实验对比了三种具有相同规模、相似结构但不同实现方式的网络 (BOIN、Buffered 以及 RAT) 的延时特性, 以及网络综合性能指标—吞吐率延时比 (吞吐率/延时). 以吞吐率延时比作为网络性能指标既考虑了网络的延时特性又考虑了吞吐率特性, 其大小表征了网络的综合性能指标, 吞吐率延时比越大, 表明网络的综合性能越高. 这三种网络均由各自的交换节点组成, 每个交换节点内部具有 2 个处理节点, 分别向 $X+$ 和 $Y+$ 方向发送数据, 并从 $X-$ 和 $Y-$ 方向接收数据. N^2 个交换节点组成一个 $N \times N$ 的单向 Torus 结构. 在 Buffered 网络中, 每个中间节点均对数据报文进行缓存, 并在有限长度的队列中进行排队等待; 在 RAT 网络中, 源和目的之间的数据传输首先要进行链路请求, 然后响应, 成功后再进行数据报文的传输. 模拟工具采用 OMNet + + [10], 为了便于比较, 在实验中, 这三种互连结构的网络规模为 4 × 4, 链路带宽 10Gb/s, 光链路延时 25.6ns, 电链路延时 6.4ns, 报文长度 256b, 模拟时间均为 1,000,000 个时隙. 当网络负载为 $\lambda (0 < \lambda < 1)$ 时, 相邻报文间隔时间 T (单位为时隙) 服从参数为 λ 的几何分布, 即 $Pr(T = k) = \lambda(1 - \lambda)^{k-1}, k = 1, 2, 3, \cdots$.

4.1 网络平均延时-网络负载

图 4 给出了 Buffered 网络、BOIN 网络和 RAT 网络在不同的网络负载下的平均延时. 从图中可以看出, Buffered 网络的延时明显高于 BOIN 网络和 RAT 网络, 特别是当网络负载增大时. 而 BOIN 网络的延时随着网络负载的增大非常缓慢的增加, 且始终保持很低, 这说明 BOIN 网络在低网络负载和高网络负载情况下都具

有较低的延时.

4.2 网络吞吐率延时比-网络负载

图5是网络的吞吐率延时比与网络负载的关系. 我们可以看出, 在绝大多数负载下, BOIN 网络的吞吐率延时比都比其他两种网络高. 这说明 BOIN 网络具有良好的综合性能.

从上面的实验我们可以看出, 在延时特性和吞吐率延时比特性方面, BOIN 网络相对于 Buffered 和 RAT 网络都具有较为明显优势. 这说明 BOIN 网络在具有较低网络延时的同时, 还保持着较高的网络吞吐率.

5 结论

BOIN 光互连网络有两个基本的设计思想: (1) 采用较短的数据报文, 在网络的中间节点上不进行数据缓冲, 而是将其缓存在相邻节点之间的网络链路上. 当两个数据报文在某个中间节点处竞争输出链路时, 采用 2×2 的光开关来实现报文的偏折路由, 从而消除冲突. (2) 将报文的控制信息和数据字段分开, 分别在电控制网络和光数据网络上同步传输, 并利用控制报文和数据报文之间的时间差, 在下一个节点处提前进行路由判断和路由选择, 使得在光数据网络上传输的数据报文在中间节点上不需要停顿或路由判断, 能够持续前进.

BOIN 网络是一种高速光互连网络, 在通信延时、吞吐率和带宽方面具有明显的优势, 同时其无死锁/活锁的特性以及确定的通信延时上限, 使得其更加适合于高性能并行计算机系统内部各处理节点之间的互连.

参考文献:

[1] National Research Council of the National Academies. Getting Up to Speed-The Future of Supercomputing [R]. Washington DC: The National Academies Press, 2005.

[2] Goodman J W. Optical interconnections for VLSI systems [J]. Proceedings of the IEEE, 1984, 72(7): 850- 866.

[3] Ronald Luijten, Cyriel Minkenberg, Roe Hemenway, Michael Sauer, Richard Grzybowski. Viable opto-electronic HPC interconnect fabrics [A]. Proceedings of the 2005 ACM/IEEE Conference on Supercomputing [C]. Washington DC: IEEE Computer Society, 2005.

[4] Avinash Karanth Kodi, Ahmed Louri. Design of a high-speed optical interconnect for scalable shared-memory multiprocessors [J]. IEEE Micro, 2005, 25(1): 41- 49.

[5] C Hawkins, B A Small, D S Wills, and K. Bergman. The Data Vortex, an all optical path multicomputer interconnection network [J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(3): 409- 420.

[6] Georgios I Papadimitriou, Chrisoula Papazoglou, Andreas S. Pomportsis. Optical switching: switch fabrics, techniques, and architectures [J]. Journal of Lightwave Technology, February 2003, 21(2): 384- 405.

[7] Assaf Shacham. Architectures of Optical Interconnection Networks for High Performance Computing [D]. New York: Columbia University, 2007.

[8] Q Xu, S Manipatruni, B Schmidt, J Shakya, M Lipson. 12.5 Gbit/s carrier-injection-based silicon microring silicon modulators [J]. Optics Express, 2007, 15(2): 430- 436.

[9] Q Xu, B Schmidt, S Pradhan, M Lipson. Micrometre-scale silicon electro-optic modulator [J]. Nature, 2005, 435(7040): 325 - 327.

[10] OMNet++ [CP/OL]. http://www.omnetpp.org/, 2008-5.

作者简介:

齐星云　男, 1979 年生于陕西岐山. 国防科技大学计算机学院博士研究生, 研究方向为高性能计算机体系结构, 并行计算机互连网络等.
E-mail: qi_xingyun@nudt.edu.cn

窦　强　男, 1973 年生于湖南长沙, 博士. 国防科技大学计算机学院计算机系统结构研究室主任, 副研究员. 研究方向为高性能计算机体系结构, 高性能微处理器设计, 并行互连网络等.