一种实用的互联网络拓扑结构RPC(k)及路由算法

邢长明[1] 刘方爱[1] 杨 林[2]
(山东师范大学信息科学与工程学院 济南 250014)[1]
(山东商业职业技术学院国际交流学院 济南 250103)[2]

摘 要 Pertersen图由于具有短直径和正则性等特性,在并行计算与分布式计算中具有良好的性能。基于环结构,提出了一种Pertersen图的新扩展方法,构造了互联网络RPC(k)。分析了该互联网络的性质,它具有连接度小、网络直径短、拓扑结构简单以及易于扩展等特点。同时给出了RPC(k)优于二维Torus以及RP(k)互联网络的直径和节点可分组性的条件。最后,分别设计了RPC(k)上的单播路由、置换路由、广播路由和多对多路由,它们的通信效率分别为$\lceil k/2 \rfloor+5$,$k+9$,$\lceil k/2 \rfloor+5$和$k+9$。特别是随着k的增大,RPC(k)网络路由算法的通信效率近似于RP(k)网络上的对应算法通信效率的1/3倍。

关键词 互联网络,RPC(k),Petersen图,环,路由算法
中图法分类号 TP393 **文献标识码** A

Practical Interconnection Network RPC(k) and its Routing Algorithms

XING Chang-ming[1] LIU Fang-ai[1] YANG Lin[2]
(Information Science and Technology College, Shandong Normal University, Jinan 250014, China)[1]
(International School of SICT, Shadong Institute of Commerce and Technology, Jinan 250103, China)[2]

Abstract Petersen graph has good performance in parallel and distributed computation because of its characteristics such as short diameter and regularity. Based on ring structure, a new extension of Petersen graph was proposed, and a new interconnection network, the RPC(k) network was developed. Additionally, the properties of the RPC(k) network were investigated. It was proved that RPC(k) has lower-degree connectivity, smaller network diameter, simple topology and good extensibility. On the basis of these analysis, the conditions satisfying that the network diameter and grouping ability of RPC(k) are better than the diameter and grouping ability of 2-D Torus and RP(k) interconnection networks were presented. Finally, based on the RPC(k) network we designed a set of routing algorithms which are point-to-point routing, permutation routing, one-to-all routing and all-to-all routing. Their communication efficiencies are $\lceil k/2 \rfloor+5$, $k+9$, $\lceil k/2 \rfloor+5$ and $k+9$ respectively. Especially as the k increasing, the efficiencies of these routing algorithms based on RPC(k) approximate to 1/3 of which based on RP(k) network.

Keywords Interconnection network, RPC(k), Petersen graph, Ring, Routing algorithm

在并行计算机体系结构中,处理机之间的通信效率是影响系统性能的主要因素之一。为了提高通信效率,人们一直在研究高带宽、低延迟和可扩展的互联网络。到目前为止,研究者从不同的方面对互联网络进行了大量的研究,提出了各种各样的互联网络拓扑结构[1-9]。文献[3,4]提出了WK-recursive网络并讨论了其路由算法。文献[5,6]讨论了Cayley图、mesh和Hypercube上的路由性质。文献[7]将Petersen图嵌入Hypercube网络,提出了一种HP网络。这些研究都集中在寻找具有低节点连接度、较小网络直径、简单路由策略和易于扩展的互联网络上。

在文献[1]中,将Petersen图与环相结合,构造了RP(k)网络,该网络具有简单的拓扑性质和路由算法,当节点数小于300时,其通信性能明显优于2-D mesh。当节点数大于300时,RP(k)网络不再具有明显的优势。在文献[2]中,提出了一类层次环互联网络HRN,并重点讨论了HRN的一个子类,即RP(k)互联网络的扩展$RP(P,k_1,k_2,\cdots,k_t)$。另外,文献[8,9]也对RP(k)互联网络进行了扩展,其中文献[8]利用双环网络与Petersen图结合,构造了一个双环Petersen互联网络DLCPG(k)。文献[9]基于二维环/双环,构造了二维双环互连Petersen图网络DCP(k)和二维环互连Petersen图网络TCP(k)。这些针对RP(k)网络的研究,虽然提出了更小直径的互联网络,但是这些互联网络均增大了RP(k)的网络连接度,它们将RP(k)网络的连接度由5变为7或11。而节点的度是网络复杂性和成本的一种度量,网络中节点的度越高,其网络成本越高,网络的复杂性越大。

本文在保持RP(k)互联网络节点度的前提下,利用环结

到稿日期:2009-07-08 返修日期:2009-09-27 本文受国家自然科学基金项目(60373063,90612003),山东省自然科学基金项目(Y2007G11)资助。

邢长明(1983-),男,博士生,主要研究方向为互联网络、网格计算,E-mail:xingchm@tom.com;刘方爱(1962-),男,博士,教授,博士生导师,主要研究方向为并行处理、互联网络、网格计算;杨 林(1983-),女,硕士,主要研究方向为计算机网络及应用研究。

构提出了 Petersen 图的一种新的扩展方法，并在该扩展方法的基础上构造了一种新型的互联网络 RPC(k)。研究了 RPC(k)网络的性质，比较了 RPC(k)，RP(k)，环和 Tours 互联网络。分析表明，RPC(k)互联网络具有良好的可扩展性、短的网络直径和简单的拓扑结构。另外，对于由 $30\times k$ 个节点组成的互联网络，RPC(k)还具有如下性质：(1)当互联网络连接的节点个数大于 90 时，RPC(k)网络的直径小于 RP(k)的直径，且 k 越大，RPC(k)网络的直径越接近 RP(k)直径的 1/3。(2)当互联网络连接的节点个数小于 2940 时，RPC(k)网络的直径小于二维 Torus 的直径。(3)当最优分组的节点数 $m=3$ 或 $m>46$ 时，RPC(k)互联网络的可分组性优于 RP(k)互联网络的可分组性，特别地，m 越大，RPC(k)最优分组的距离越接近 RP(k)最优分组距离的 1/3。(4)当最优分组的节点数 $m\leqslant 3136$ 时，RPC(k)互联网络的可分组性优于二维 Torus 互联网络的可分组性。特别地，当 $81\leqslant m\leqslant 441$ 时，RPC(k)最优分组的距离小于 Torus 最优分组距离的 1/2。

最后，本文还基于 RPC(k)互联网络，设计了单播路由、置换路由、广播路由(One-to-all)和多对多路由(All-to-all)路由算法，它们的通信效率分别$\lceil k/2\rfloor+5$，$k+9$，$\lceil k/2\rfloor+5$ 和 $k+9$。通过分析比较，证明了这些算法的通信效率与 RP(k)网络上对应算法的通信效率相比均有明显的提高。因此，RPC(k)是一种具有良好拓扑性质的实用互联网络拓扑结构，可为并行计算、分布式共享等应用领域的通信网络设计提供理论基础。

1 预备知识

为了分析方便，下面先给出有关的基本概念和基础知识。

1.1 基本概念

定义 1(度) 若 p 为图 $G=(V,E)$中的任意节点，即 $p\in V$，称 p 在图 G 中的相邻节点的个数为节点 p 的度。

定义 2(直径) 图 $G=(V,E)$中任意两个节点之间的距离最大值，称为图 G 的直径。

定义 3(节点组的距离) 对于一个互联网络 N 中的一组节点 G，节点组 G 的距离定义为该组中任意两个节点距离的最大值。

定义 4(最优节点分组)[1,2] 对于给定的正整数 m，在互联网络 N 中存在多个包含 m 个节点的组，称距离最短的组为含有 m 个节点的最优节点分组，记为 $Gm(N)$。

定义 5(可分组性)[1,2] 已给两个互联网络 N_1，N_2，若对于任意正整数 m 有 $G_m(N_1)$的距离$\leqslant G_m(N_2)$的距离，则称互联网络 N_1 的可分组性优于互联网络 N_2 的可分组性。

1.2 Petersen 图及其性质

如图 1 所示，Petersen 图有许多良好的性质。

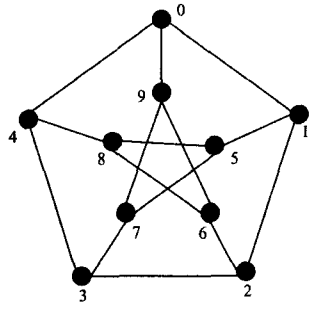

图 1 Petersen 图

(1)Petersen 图的度是 3；

(2)Petersen 图的直径是 2，即两个节点或者直接相连，或者通过第三个节点相连；

(3)Petersen 图是关于节点对称的；

(4)Petersen 图的任意两个节点之间存在 3 条无交的路；若两点直接相连，则这 3 条路的长度分别为 1,4,4，否则这 3 条路的长度分别为 2,3,3。

以上性质比较直观，通过观察就可以得到。另外，研究还表明 Petersen 图互联网络具有良好的可嵌入性[10]，其缺点是网络的可扩展性差。怎样才能使其既具有较好的连接性、较短的直径，又具有可扩展性呢？

2 互联网络 RPC(k)

2.1 PC 图

环作为简单的互联网络已被广泛应用，我们将环的简单性和 Petersen 图的可连接性结合在一起，借鉴 CCC 图的思想，构造 PC 图(petersen-connected cycles)，如图 2 所示(为了便于描述，下文将 PC 图中的环称为圈)。

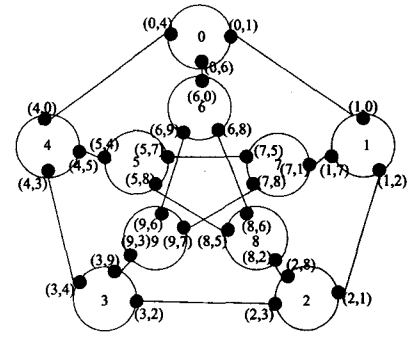

图 2 PC 图及编码

(1) PC 图的构造。PC 图是在 Petersen 图上将各个节点替换成含 3 个节点的圈而构成的。

(2)PC 图的性质。PC 图中包含 30 个节点，每个节点的度为 3，PC 图的直径为 5。类似 Petersen 图，PC 图中两个圈要么直接相连，要么通过第三个圈相连。同时，PC 图仍然保持着 Petersen 图的正则性，对称性等优良特性。

(3) PC 图的编码。PC 图采用如下的编码方式：首先对 PC 图中的圈按照 0,1,…,9 编号，如图 2 所示。每一个节点由两部分 (m,n) 决定，其中 m 是节点所在圈的编号，$0\leqslant m\leqslant 9$，n 是节点连接圈的编号。

2.2 RPC(k)的结构

利用环的易扩展性，将环与 PC 图结合在一起，构造易于扩展的互联网络 RPC(k)。

(1)RPC(k)网络的构造。互联网络 RPC(k)由 $30\times k$ 个节点组成。每 30 个节点组成一片，一片内节点按照 PC 图互联在一起，片内节点按照 PC 图编码方式进行编号。k 个 PC 图按照以下办法连接：不同片内编号相同的节点按照环的结构互联在一起，这样共得到 30 个环。

(2)RPC(k)的编址。RPC(k)采用如下编址方式，任意节点的编址由 3 部分(s,m,n)决定，其中 s 是片的编号，$0\leqslant s\leqslant k-1$，$(m,n)$是片内节点的编号。

2.3 RPC(k)的性质

RPC(k)网络不仅继承了许多 Petersen 图和环的性质，而且还有自身独特的性质，它们对于互联网络的设计具有重要

的价值。

性质 1 假定互联网络的节点个数为 $30\times k$，则 RPC 是正规互联网络，且当 $k=1$ 时，各节点的度为 3，当 $k>1$ 时，各节点的度为 5。

性质 2 假定互联网络的节点个数为 $30\times k$，RPC(k)网络的直径是$\lceil k/2\rfloor+5$。其中$\lceil\rfloor$为下取整运算符。

任意两个节点 x 和 y 之间的距离可以分为片内距离和环上距离之和。片内的最大距离是 5，环上的最大距离是$\lceil k/2\rfloor$，因此 RPC(k)的直径为$\lceil k/2\rfloor+5$。

假定互联网络节点的个数为 $30\times k$，则 RPC(k)与 RP(k)，二维 Tours 及环比较，可得到表 1 的结果。

表 1 RPC(k)与 RP(k)，二维 Torus 及环的比较

	RPC(k)	*RP*(k)	二维 *Torus*	环
正规性	是	是	是	是
连接度	5	5	4	2
直径	$\lceil k/2\rfloor+5$	$\lceil 3k/2\rfloor+2$	$2\times\lceil\sqrt{30\times k}/2\rfloor$	$15\times k$

由表 1 中 *RPC*(k)的特性可以验证以下性质 3、性质 4 成立。

性质 3 当互联网络连接的节点个数大于 90 时，RPC(k)网络的直径小于 RP(k)的直径，且 k 越大，RPC(k)网络的直径越接近 RP(k)直径的 1/3。

证明：若使$\lceil k/2\rfloor+5\leqslant\lceil 3k/2\rfloor+2$ 成立，则需要 $k\geqslant 3$。因此，性质 3 成立。

性质 4 当互联网络连接的节点个数小于 2940 时，RPC(k)网络的直径小于二维 Torus 的直径。

证明：当 $k\leqslant 98$ 时，$\lceil k/2\rfloor+5<2\times\lceil\sqrt{30\times k}/2\rfloor$成立。因此，性质 4 成立。

假设在机群系统中，系统采用 2940 个节点，一个节点含有 2～16 个处理机，共享内存，节点之间由互联网络连接，那么，该互联网络可以连接 5880～47040 个处理机组成的并行计算机系统。在这个范围内，RPC(k)的直径小于 Torus 的直径。

我们知道，并行程序要在一组节点内进行频繁的通信，因此组内的通信性能对程序的效率影响非常大。为此，考虑互联网的可分组性具有重要意义。下面分别计算 RPC(k)、RP(k)及二维 Torus 网络最优分组的距离。

依据各互联网络的性质易知(以下 div 为取商运算符)：

RPC(k)最优分组的距离为

$$d(Gm(RPC(k)))=\begin{cases}1, & m=2,3\\ 2, & m=4,5,6\\ 3, & m=7,8,\cdots,12\\ 4, & m=13,14,\cdots,20\\ 5+(m-1)\ \mathrm{div}\ 30, & m\geqslant 21\end{cases}\tag{1}$$

RP(k)最优分组的距离为

$$d(Gm(RP(k)))=\begin{cases}1, & m=2\\ 2+(m-1)\mathrm{div}\ 10, & m\geqslant 3\end{cases}\tag{2}$$

二维 Torus 最优分组的距离为

$$d(Gm(Torus))=(\sqrt{m}-1)\times 2\tag{3}$$

由式(1)－式(3)易知下述性质 5、性质 6 成立。

性质 5 当最优分组的节点数 $m=3$ 或 $m>46$ 时，RPC(k)互联网络的可分组性优于 RP(k)互联网络的可分组性，特别地，m 越大，RPC(k)最优分组的距离越接近 RP(k)最优分组距离的 1/3。

性质 6 当最优分组的节点数 $m\leqslant 3136$ 时，RPC(k)互联网络的可分组性优于二维 Torus 互联网络的可分组性。特别地，当 $81\leqslant m\leqslant 441$ 时，RPC(k)最优分组的距离小于 Torus 最优分组距离的 1/2。

由于各种通信模式往往建立在点点通信模式的基础上，因此两点间的距离对通信性能影响很大。从上面分析可以看出，在一定的范围内，RPC(k)网络最优分组的距离远远小于 RP(k)和 Torus 网络的最优分组距离，因此 RPC(k)作为互联网络具有较大的优势。

3 RPC(k)上的路由算法及其性能分析

前面分析了 RPC(k)网络的拓扑性质，讨论了其直径和节点可分组性。下面讨论另一个性质，即网络路由的方便性。路由是互联网络的基本操作，路由效率的高低直接影响并行系统的效率，决定互联网络的可用性。本节讨论一组基于 RPC(k)网络的路由算法，它们是单播路由、置换路由、广播路由和 All-to-all 路由。为了讨论这些算法，引入以下定义。

定义 6(邻接圈) 对于同一片内的任意两个圈 m_1，m_2，如果圈 m_1 中存在节点 $A(s,m_1,m_2)$，圈 m_2 中存在节点 $B(s,m_2,m_1)$，则称圈 m_1 与圈 m_2 互为邻接圈。当数据包需要由圈 m_1 传送到圈 m_2 时，节点 A 称为圈 m_1 的出圈邻接点，节点 B 称为圈 m_2 的入圈邻接点；同理，当数据包需要由圈 m_2 传送到圈 m_1 时，节点 B 称为圈 m_2 的出圈邻接点，节点 A 称为圈 m_1 的入圈邻接点。

定义 7(片内三等分) 如图 3(a)所示，相同片内的所有节点，可根据任意节点 A 的最小生成树，划分成图中虚线所示的三组节点，称这种划分为节点 A 的片内三等分。

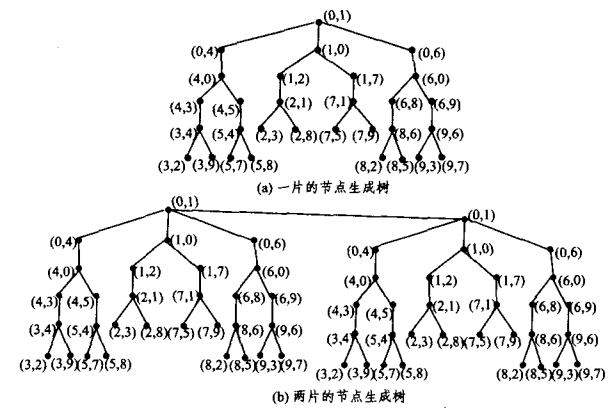

图 3 广播路由传播过程

另外，为了分析算法的性能，以算法中不可能重叠发送操作的次数作为评价算法的尺度。例如，一个算法需要 6 次通信操作，是指该算法需要 6 次不能重叠的通信操作时间。

为了下面讨论的方便，在此先给出一个结论，即考察实现广播路由需要多少次通信操作。广播是互联网络中的一个重要操作，其语义是实现从一个节点向其它所有节点广播数据。图 3 分别是具有 30 和 60 个节点的 RPC(k)网络的最小生成树，可以看出，在同一个片内，广播操作需要发送 5 次，在两片内需要 3 次，很容易得知在三片内也需要 3 次。实际上，可以得到如下定理。

定理 1 在 RPC(k)互联网络中，实现广播路由需要的通信次数 C_k 由以下关系决定：

$C_k=\lceil k/2\rfloor+5$

证明：从任意一个节点开始，可以构造一个片、相邻两片、相邻三片等的最小生成树。由构造过程可知，当只有一个片时，树的高度为 5，因而 $C_1=5$；当相邻两片或三片时，生成树的高度为 6，因此 $C_2=6$，$C_3=6$。以此类推，当 $k=2n$ 片时，$C_{2n}=C_{2n-1}+1$；当 $k=2n+1$ 片时，$C_{2n+1}=C_{2n}$，因此，上面结论成立。

3.1 单播路由算法及性能分析

单播路由是互联网络最基本的操作，其它操作都建立在该操作的基础上。假设节点 $A(s_1,m_1,n_1)$ 向节点 $B(s_2,m_2,n_2)$ 发送数据，RPC(k)网络上的单播路由算法如下：

1. 当节点 A，B 在同一个片内时，沿着以 A 为根节点的最小生成树，将数据包送到节点 B。

2. 当节点 A，B 不在同一个片内时，按照最短距离，沿着环(m_1,n_1)，将数据包送到片 s_2 上的(m_1,n_1)节点，然后在同一片内将数据包从(m_1,n_1)送到(m_2,n_2)。

性能分析：从 s_1 片开始，沿着环将数据发送到 s_2 片时至多需要$\lceil k/2\rfloor$次通信操作，而在同一片内完成两点间的通信至多需要 5 次通信操作，因此在最坏情况下，该算法需要$\lceil k/2\rfloor+5$ 次通信操作。该算法沿最短路径传送数据，因而是最优算法。

由性质 3 的证明过程易知下述性质 7 成立。

性质 7 在最坏情况下，RPC(k)互联网络的单播路由算法需要$\lceil k/2\rfloor+5$ 个时间步。当节点个数大于 90 时，RPC(k)的单播路由算法的通信效率高于 RP(k)的单播路由算法的通信效率，特别地，k 越大，RPC(k)网络的单播路由算法的通信效率越接近 RP(k)的单播路由算法的通信效率的 1/3 倍。

3.2 置换路由算法及性能分析

置换路由是互联网络中常用的操作，其要求每一个节点需要向另一个节点发送数据包，而且每一个节点只能接收到一个节点的数据包。因此，若发送节点不同，那么它们的目标节点也不相同，这样就构成了 n 个节点间的一个置换。假设通信线路是半双工的，节点是 all-port，即可以同时向多个端口发送数据包。

3.2.1 圈内的置换路由

实现圈内置换路由，圈内的 3 个节点按顺时针方向(或逆时针)，并行将数据发送到它的目标节点即可。由于该操作过程中每条边至多使用一次，由此可得定理 2。

定理 2 任意圈内的置换路由至多需要 1 次通信操作完成。

3.2.2 片内的置换路由

定理 3 任意片内的置换路由至多需要 13 次通信操作完成。

我们先设计一个片内置换路由算法，然后说明它至多需要 13 次通信操作完成。

算法 Slice-Permutation

片内各个节点并行的执行以下操作：

Step1 如果节点上存在发向非本地圈的数据，首先将数据发送到出圈邻接点；

Step2 当节点接收到数据后，将非本地圈的数据打包，接着将打包后的数据发送到入圈邻接点；

Step3 当节点接收到数据包后，解开数据包，执行 Step1 一次；

Step4 执行 Step2 一次；

Step5 当节点接收到数据包后，解开数据包，相同圈内的节点将数据直接发送到它的目标节点。

性能分析：由于采用半双工通信，算法的 Step1 和 Step2，每一步至多使用一条边两次，从而这两步至多需要 4 个通信时间步。完成这两步后，需要置换的数据要么在目标节点的本地圈内，要么在目标节点的邻接圈内。算法执行 Step3，同一个圈内最多有 9 个发向邻接圈的数据，即一个节点上最多有 3 个发向邻接圈的数据，将这些数据发送到它的出圈邻接点，最坏的情况下，需要 5 个通信时间步。算法的 Step4 同 Step2 需要 2 个通信时间步。执行完 Step4 后，所有数据都已到达其本地圈内。在算法的 Step5，同一个圈内的一条边至多使用两次，从而这一步至多需要 2 个时间步。因此，整个片内置换至多需要 13 次通信操作即可完成。

3.2.3 简单置换路由

现在讨论 RPC(k)网络上的一种简单置换路由算法，即置换路由满足下面的条件：如果两个源节点 $A(s_1,m_1,n_1)$ 和 $B(s_2,m_1,n_1)$ 在同一个环上，那么它们的目标节点一定不在同一个片上。也就是说，置换将同一环上不同片内两个节点的数据变换到不同的片上去。我们称这种置换路由为简单置换路由。实现该路由的思想如下：每一个待发送的数据沿着所在环的同一个方向(如顺时针)发送，使其到达目的节点所在的片，每个节点在同一个环上至多需要 $k-1$ 次通信操作，注意，这个过程不会发生通信冲突。这时，由简单置换路由的定义可知，在每一个节点上，有且仅有一个数据，在每一片内执行一次片内置换路由，最终实现整个 RPC(k)网络上的简单置换路由。

性能分析：由于采用流水方式，当每个节点沿着它所在的环传递 $k-1$ 次后，所有数据包都在其目标节点所在的片内。又由定理 3 知，实现片内置换路由至多需要 13 个通信时间步，因此实现简单置换路由至多需要 $k+12$ 个通信时间步。

定理 4 在 RPC(k)互联网络上，实现简单置换路由至多需要 $k+12$ 次通信时间步。

3.2.4 一般置换路由

现在讨论 RPC(k)网络上的置换路由算法。在简单置换路由中，已经看到，一个包经过至多 $k-1$ 次通信，就可以沿着同一个环到达目标节点所在的片。当采用一般置换路由时，经过以上 $k-1$ 次通信后，在一个节点上可能包含多个数据包，最多可以有 30 个。也就是说，在一个片内的所有节点上共有 30 个数据包。要将这 30 个数据包分配到片内的各个节点上，每个节点只能接收 1 个数据包。我们称这种分配为片内不完全路由。

定理 5 实现片内不完全置换路由至多需要 10 个时间步。

下面构造一个算法，完成片内不完全置换路由，然后证明该算法需要 10 次通信。

算法 Incomplete-Permutation

Step1 每个节点根据其片内三等分，将要发送的数据按照目标节点所在的分组分成 3 组，每组数据按照从当前节点到目标节点最短距离递减的顺序排序，然后按照该顺序各组并行发送一个数据；

Step2 当节点接收到数据后，执行 Step1 一次。重复执行该步

骤 9 次。

性能分析：将每个节点的数据进行片内三等分后，每个节点与另外 3 个节点直接相连，因此 3 组数据可以并行向外传送。在相同片内，源节点与其目标节点的最远距离为 5。由于采用单向传输，每组一次只能向外发送 1 个数据，每组最多有 10 个数据，所以经过 10 次通信后每组的数据都能发送到其目标节点。

从定理 5 和简单置换路由算法可以得到：

性质 8 RPC(k)网络上的置换路由至多需要 $k+9$ 次通信时间实现。假定互联网络的节点个数为 $30\times k$，RPC(k)网络的置换路由算法的通信效率优于 RP(k)网络的置换路由算法，特别地，随着 k 的增大，RPC(k)网络的置换路由算法的通信效率越接近 RP(k)的置换路由算法的通信效率的 1/3 倍。

3.3 广播路由算法及性能分析

广播操作是互联网络的基本操作，其语义是将某一个节点上的数据广播到其他所有节点上。假设数据开始保存在节点 $A(s,m,n)$ 上，现在讨论如何在 RPC(k)网络上实现该操作。

可以采用这样一种策略，将数据沿着以 A 为根节点的最小生成树进行广播。根据定理 1，该策略的效率为$\lceil k/2\rfloor+5$。

还可以采用另一种策略，首先将数据在一个片内完成广播，然后沿着 30 个环向两个方向进行扩散。该策略的效率同样为$\lceil k/2\rfloor+5$。

性质 9 RPC(k)网络上的广播路由至多需要$\lceil k/2\rfloor+5$次通信时间实现。假定互联网络的节点个数为 $30\times k$，当 $k>3$ 时，RPC(k)网络广播路由算法的通信效率优于 RP(k)网络的广播路由算法，特别地，随着 k 的增大，RPC(k)网络的广播路由算法的通信效率越接近 RP(k)广播路由算法通信效率的 1/3 倍。

3.4 多对多路由算法及性能分析

多对多(All-to-all)路由要求每个节点都向其它所有节点发送一个数据。在这种操作中，每一个节点都含有一个数据，要求将该数据发送到其它所有节点中去。也就是说，每个节点都要完成一个广播操作。

显然，可以采用一种简单的策略，即每一个节点都做一次广播操作，按照上面的算法，所需要通信时间为：$30\times k\times(\lceil k/2\rfloor+5)$次。但是，这样的算法效率太低。现在给出另外一种策略。

定理 6 在 RPC(k)网络中，实现圈内的多对多操作需要 2 次通信时间。

由于圈内共有 3 个节点，每个节点并行将自身的数据沿同一个方向(如顺时针)发送，每个节点至多需要 2 次通信，即可实现圈内多对多路由操作。

定理 7 在 RPC(k)网络中，实现片内的多对多操作需要 10 次通信时间。

下面构造一个算法，实现片内的多对多操作，其算法如下。

算法 Slice-All-to-All

Step1 各个节点并行地执行圈内多对多操作，完成该操作后，将节点上的 3 个数据打包；

Step2 每个节点将打包后的数据发送到与其直接相连的邻接圈；

Step3 各个节点并行地对收到的数据包进行圈内多对多操作；

Step4 每个节点将 Step3 收到的两个数据包打包后，发送到与其直接相连的邻接圈；

Step5 各个节点并行地对收到的数据包进行圈内多对多操作。解开数据包，完成该操作。

性能分析：算法的 Step1，Step3，Step5 都是做圈内多对多操作，各需要 2 次通信操作；Step2，Step4 都使用圈的连接边两次，各需要 2 次通信操作。因此实现片内多对多路由需要 10 通信操作完成。

在前面讨论的基础上，下面给出 RPC(k)网络上的多对多路由算法。

算法 All-to-All

Step1 每片并行实现片内的多对多操作，将每个节点上的 30 个数据打包；

Step2 每个节点沿着它所在的环，按顺时针方向，向相邻节点发送数据包，并接收到达的数据包，重复 $k-1$ 次；在每一个节点上，解开所接受到的 $k-1$ 个数据包，这样就完成了该操作。

性能分析：根据定理 6，我们知道，算法的 Step1 需要 10 次通信操作，算法的 Step2 需要 $k-1$ 次，因此整个算法需要 $(k+9)$次通信操作完成。

性质 10 RPC(k)网络上的 All-to-all 路由需要 $k+9$ 次通信时间实现。假定互联网络的节点个数为 $30\times k$，当 $k>2$ 时，RPC(k)网络的多对多路由算法的通信效率优于 RP(k)网络的多对多路由算法，特别地，随着 k 的增大，RPC(k)网络的多对多路由算法的通信效率越近似于 RP(k)多对多路由算法通信效率的 1/3 倍。

结束语 基于环结构，提出了一种 Petersen 图的新型扩展方法，并在该扩展方法的基础上构造了一种新型的互联网络 RPC(k)。该互联网络具有良好的性质，其拓扑结构简单，连接度为常数，易于扩展。同时给出了 RPC(k)优于二维 Torus 以及 RP(k)互联网络的直径和节点可分组性的条件。最后，基于 RPC(k)互联网络，构造了一组路由算法，即单播路由、广播路由、置换路由以及多对多路由，其通信效率均优于 RP(k)网络上的对应算法的通信效率，特别是随着 k 的增大，RPC(k)网络路由算法的通信效率近似于 RP(k)网络上的对应算法通信效率的 1/3 倍。分析表明，在相当大的范围内，RPC(k)网络在通信开销上具有显著的优越性，因此 RPC(k)互联网络及其路由算法能够为并行计算和分布式计算提供高效和实用的通信支持。今后的工作是进一步研究 RPC(k)网络在具体应用上的优越性，研究其它常用网络嵌入 RPC(k)拓扑结构的可行性及 RPC(k)网络对常用算法的适应性。

参 考 文 献

[1] Liu Fang-ai, Liu Zhi-yong, Qiao Xiang-zhen. A practical in-terconnection network RP(k) and its routing algorithms[J]. Science in China(Serial F), 2001, 44(6): 461-473

[2] 刘方爱，刘志勇，乔香珍. 一类层次环网络的构造及路由算法[J]. 计算机学报，2002，25(12)：1397-1404

[3] Chen G H, Duh D R. Topological properties, communication and computation on WK-recursive networks[J]. Networks, 1994, 24(6): 303-317

[4] Su M Y, Chen G H, Duh D R. Broadcasting on incomplete WK-recursive networks[J]. Journal of Parallel and Distributed Computing, 1999, 57: 217-224

(下转第 175 页)

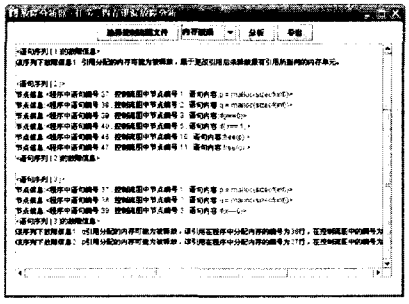

图 10　检测程序界面

选取 *Sequence*1 作为示例，根据 3.3 节中的基于可达路径的内存泄漏检测过程，有表 3 所列的结论。检查 *pointerSet*1，由于 *pointerSet*1 不为空，根据定义 6，该段程序存在内存泄漏故障。

表 3　路径 Sequence1 及其指针映射集合状态

节点(Node)	Sequence1 指针映射集合的状态
Node1	{⟨p,1211886733562⟩}
Node2	{⟨p,1211886733562⟩,⟨q,1211886733622⟩}
Node3	{⟨p,1211886733562⟩,⟨q,1211886733622⟩}
Node12	{⟨p,1211886733562⟩,⟨q,1211886733622⟩}

实验表明，*Sequence*1 存在两处内存泄漏故障。遍历路径 *Sequence*2 结束后，指针映射集合 *pointerSet*2 中任存在一个指针变量⟨Null,1211886733562⟩，根据定义 6，*Sequence*2 存在一处内存泄漏故障。遍历路径 *Sequence*3 结束后指针映射集合 *pointerSet*3 为空，根据定义 6，*Sequence*3 不存在内存泄漏故障。

5　相关工作对比

常见的检测内存泄漏故障的方法主要分为动态检测和静态检测，其中文献[13]采取了动态检测的方式，主要考虑针对函数调用时的内存堆栈信息来判定是否存在内存泄漏故障，但是该方法严重依赖测试用例来确保对待测程序的代码覆盖，从而在测试效率上存在缺陷。而静态检测方式大部分基于待测程序的控制流图，文献[1]通过待测程序的控制流图进行静态分析，较好地解决了测试的代码覆盖问题，然而该方法没有解决控制流图的可达路径生成问题。另一些研究人员通过别名集合[4-9]来分析待测程序的内存分配与释放，以此作为检测内存泄漏故障的依据，其优点是检测效率较高，但该方案存在误报的可能性，并且在别名分析[6]阶段，需要对待测程序的 PCG[10]与 SCFG[10]不断地进行迭代扫描，在算法效率上存在缺陷。本文采用重新定义的控制流图模型来检测内存泄漏故障，并设计控制流图的可达路径生成算法来实现对控制流图的遍历，定义了指针映射集合上模拟待测程序所执行的内存操作，以指针映射集合的状态来作为内存泄漏故障的判断依据，从而降低了误报率。

结束语　本文提出了一种新的内存泄漏故障测试方法，该方法基于对待测程序的控制流图的分析，生成控制流图的所有可达路径并建立其对应的指针映射集合，通过对指针映射集合的操作来反映待测程序相应路径的内存操作，从而给判断该路径是否存在内存泄漏故障提供了依据。根据本文中的指针映射集合，建立了内存泄漏故障模型，并按照该故障模型给出了检测算法以及步骤。下一步研究的主要目标是继续完善指针映射集合上的操作规则，从而增强检测程序的适应性，进而减少检测程序的误报率。另外，该检测程序目前用于检测内存泄漏故障，如何将控制流图的可达路径用于检测其他故障还有待进一步研究。

参 考 文 献

[1] 张威，卢庆龄，李梅，等. 基于指针分析的内存泄露故障测试方法研究[J]. 计算机应用研究，2006

[2] Hastings R, Joyce B. Purify: Fast Detection of Memory Leaks and Access Error[C]// Proceedings of the Winter USENIX Conference. 1999:125-136

[3] Sagiv M, Reps T, Wilhelm R. Solving Shape-analysis Problems in Language with Destructive Updating[C]// Symposium on Principles of Programming Languages. Florida, 1996:110-118

[4] Landi W, Ryder B G. Safe Approximate Algorithm for Interprocedural Pointer Aliasing[C]// ACM SIGPLAN Notices. 1992: 235-248

[5] Wilson R P, Lam M S. Efficient Context - sensitive Pointer Analysis for C Program[C]// Proceedings of the ACM SIGPLANE'95 Conference on Programming Language Design and Implementation(PLDI). California, 1995:18-21

[6] 张广梅，李晓维. 动态内存错误的静态分析[J]. 计算机辅助设计与图形学学报，2005(3)

[7] Heine D L, Lam M S. A practical flow sensitive and context sensitive C and C++ memory leak detector[J]. ACM SIGPLAN Notices, 2003:168-181

[8] Austin T M, Breach S E, Sohi Gurindar S. Efficient detection of all pointer and array access errors[J]. ACM SIGPLAN Notices, 1994, 29(6):290-301

[9] Hastings R, Joyce B. Purify: Fast detection of memory leaks and access errors[C]// Winter Usenix Conference. SanFrancisco, California, 1992:125-136

[10] Hallem S, Chelf B, Xie Y, et al. A system and language for building system specific, static analyses[J]. ACM SIGPLAN Notices, 2002:69-82

[11] 肖庆，张威，宫云战，等. 内存泄漏的一种静态分析方法[J]. 装甲兵工程学院学报，2004(6)

[12] 宫云战. 一种面向故障的软件测试新方法[J]. 装甲兵工程学院学报，2004(3)

[13] 吴民，涂奉生. 内存泄漏的动态跟踪分析[J]. 计算机工程与应用，2005(14)

(上接第 135 页)

[5] Wu F L, Lakshmivarahan S, Dhall S K. Routing in a class of cayley graphs of semidirect products of finite groups[J]. Journal of Parallel and Distributed Computing, 2000, 60:539-565

[6] Lakshmivarahan S, Dhall S K. Ring, torus and hypercube architectures/algorithms for parallel computing[J]. Parallel Computing, 1999, 25:1877-1906

[7] Das S K, Ohring S, Baneriee A K. Embedding into hyper Petersen network: Yet another Hypercube-like topology[J]. Journal of VLSI Design, 1995, 2(4):335-351

[8] 王雷，林亚平，夏巍. 双环 Petersen 图互联网络及其路由算法研究[J]. 软件学报，2006，17(5)：1115-1123

[9] 王雷，林亚平，陈治平. 二维环/双环互连 Petersen 图网络及其路由算法[J]. 计算机学报，2004，27(9)：1290-1296

[10] 刘方爱，刘志勇，乔香珍. 光 RP(k)网络上 Hypercube 通信模式的波长指派算法[J]. 软件学报，2003，14(3)：575-581