基于SD-Torus网络的分布式IP地址查找

王亚刚
(西安电子科技大学计算机学院 西安 710071)

摘 要 针对IP路由器的FIB(Forwarding Information Base)极限问题和分布式IP地址查找中的通信延迟问题,提出了SD-Torus(Semi-Diagonal Torus)直连网络。按照"临近存储"的原则,将路由表划分后存储在每个节点及其邻居节点上,以减少分布式IP地址查找中的通信延迟,提高整体的查找性能。在分析SD-Torus网络拓扑性质的基础上,提出了一种负载均衡的路由算法。基于SystemC的仿真结果表明,使用该结构可以大大降低分布式IP地址查找的通信延迟,提高系统的扩展性。该研究结果可以应用于高性能的分布式IP地址查找。

关键词 SD-Torus网络,IP地址查找,路由算法,直连网络
中图法分类号 TP393.03 **文献标识码** A

Distributed IP Address Lookup Architecture Based on SD-Torus Networks

WANG Ya-gang
(School of Computers, Xidian University, Xi'an 710071, China)

Abstract Aimed at the scalability of the IP router, especially the fact of the FIB(Forwarding Information Base) limit for IP routers, a scalable direct network, the SD-Torus(Semi-Diagonal Torus) network was proposed. The topology properties of the HT-Torus network was discussed and a load-balanced routing algorithm was presented. By adopting a novel mapping algorithm for the distributed FIB, the entire FIB was split into sub-tables and mapped onto each node and its neighbors, hence to guarantee the communication latency between nodes for distributed IP address lookup, and decrease the memory footprint of the FIB. The proposed schemes can be applied to high performance IP address lookup.

Keywords Semi-diagonal torus networks, IP address lookup, Routing algorithm, Direct networks

1 引言

IP路由器(Internet Protocol Router)[1]是Internet网络中的核心设备,主要完成IP分组的转发和交换。Internet规模的持续增长导致网络数目剧增、网络拓扑变化加剧,从而引起核心路由器中路由信息表的规模快速持续增长、路由表项更新日趋频繁、FIB极限问题(http://www.nanog.org/meetings/nanog39/presentations/bof-report.pdf)日益凸显。网络流量和各种新型应用的快速发展,对IP路由器的地址查找性能提出了更高的要求。同时,由于FIB极限问题的存在,难以在每个查找节点上存储完整的路由表,路由表的分布式存储和分布式IP地址查找成为研究的热点问题[2,3]。分布式IP地址查找就是将IP地址查找任务和路由表划分并映射到多个查找引擎LE(Lookup Engine)节点上,通过互连网络进行互连通信,完成IP地址的查找任务。

然而,现有基于互连网络的分布式IP地址查找结构中,划分后的路由子表在整个互连网络节点上的分布处于一种"杂乱"的状态,大部分IP地址查找需要通过多个中间节点通信。随着查找节点的增多,分布式IP地址查找的通信延迟成为提高地址查找性能的主要瓶颈。本文正是从该问题入手,提出一种适应分布式路由表存储的可扩展互连结构,并根据互连网络的拓扑性质,将路由子表"规则"地映射到该互连网络中的一个有限区域内(例如将一个完整的路由表映射到每个查找节点及其邻居节点上),从而降低分布式查找的存储代价和通信代价,提高地址查找的性能和扩展性能。

本文第2节给出了相关研究;第3节提出了在直连网络SD-Torus结构上进行分布式IP地址查找的总体框架;第4节给出了SD-Torus结构的主要拓扑性质及其负载均衡的路由算法;第5节描述了分布式IP地址查找中路由表的划分与映射;第6节对路由算法及分布式查找的性能进行了仿真和评估;最后总结全文。

2 相关研究

Venkatesh等提出了基于分布式存储的高性能并行IP地址查找技术[4],该结构根据路由表的分割算法将完整的FIB分成多个在查找任务上"无关的"子表,并将其相应地存储在多个查找引擎中。对每个IP分组的路由查找,只需要在其中某个子表中进行。IP地址查找任务的分配可以用一个简单的硬件逻辑仲裁实现,将不同的查找任务分配到各个子表中并行地进行路由查找,从而提高整体的查找性能。分布式并行路由查找框架DPRLF(Distributed Parallel Route Lookup Framework)[3]也采用了与Venkatesh结构类似的体

到稿日期:2011-05-26 返修日期:2011-08-03 本文受国家自然科学基金(60976020),陕西省教育厅科研计划(2010JK833)资助。
王亚刚(1972-),男,博士生,副教授,主要研究方向为计算机网络、计算机系统结构等,E-mail:wangyg@xupt.edu.cn。

系结构。在上述研究中，结合路由表划分技术，每个查找引擎使用独立的存储空间存储一个规模较小的路由子表，整个系统只需要存储一份完整的路由表，节省了存储空间。各个查找单元并行工作，各个引擎没有直接的信息通信，彼此独立。路由表更新也转化成多个子表的更新，算法简单，易于实现。其主要的缺点在于该类结构中需要单一的任务分配单元，容易形成单点故障，扩展性差。

在基于直连网络的可扩展 IP 地址查找结构 SPALA (Scalable Parallel IP Address Lookup Architecture)[2] 研究中，按照 MPP(Massively Parallel Processing)的思想，把大量的 IP 地址查找分配到大量的查找引擎上。每个查找引擎都具有本地的存储器，用来存储本地路由表。整个系统只存储一份完整的路由表，节省了空间，并利用可扩展的直连网络 2D-Torus 进行互连通信，从而提高分布式地址查找的性能和扩展性。然而，随着互连网络规模的增加，基于 2D-Torus 网络的分布式地址查找延迟快速增加，其中分布式查找的通信延迟成为整个系统的性能瓶颈。

另外，在互连结构方面，目前使用直连网络[5]构建可扩展路由器的研究日益得到重视。利用直连网络实现可扩展路由器的研究[6]中，将直连网络 H-Torus(Hexagonal Torus)结构[7]引入路由器设计，极大地提升了路由器的扩展能力。该研究提出了用直连网络构建可扩展路由器的研究实例，为使用直连网络结构进行分布式 IP 地址查找提供了借鉴。另外，文献[7]详细论述了 H-Torus 结构的拓扑性质，文献[8]对该结构上的负载均衡路由算法进行了详细的论述。然而，这些研究主要从直连网络本身的性质来研究路由器的扩展性，并没有深入研究 IP 路由器转发功能在互连网络上的映射，尤其是分布式的路由表存储和分布式的 IP 地址查找等问题。

3 分布式 IP 地址查找总体结构

针对 IP 路由器的 FIB 极限问题和分布式 IP 地址查找中的通信延迟问题，本文提出了一种基于 SD-Torus 网络的分布式 IP 地址查找结构，如图 1 所示。本结构主要由多个 IP 地址查找引擎和 SD-Torus 直连网络组成，包括两类节点，即查找引擎 LE 节点和 SD-Torus 网络的路由节点。其中 LE 节点负责 IP 地址查找任务的接收、分配和分布式查找，每个 LE 节点中都有本地存储器，用来存储部分路由表。每个 LE 节点均可接收 IP 地址查找任务，并根据预先给定的路由表存储策略在本地独立地完成该地址查找，或者通过 SD-Torus 网络将该 IP 地址查找任务转发到其它 LE 节点进行查找。SD-Torus 网络及其路由节点主要负责 LE 节点之间查找信息的路由转发。

在这种分布式 IP 地址查找结构中，分布式路由表的划分和存储至关重要。本文根据 SD-Torus 网络的拓扑结构特点，将一份完整路由表分割成 7 个子表，分别存储在每个 LE 节点及其 6 个邻居节点上，使得分布式 IP 地址查找可以在每个 LE 节点及其邻居节点中完成，而不需要经过其它的网络节点，大大降低了分布式地址查找的通信代价，提高了分布式 IP 地址查找的性能。同时，随着 SD-Torus 网络规模的增大，可以互连更多的 LE 节点，但分布式 IP 地址查找的通信延迟保持不变，从而极大地提高了分布式 IP 地址查找的扩展性。

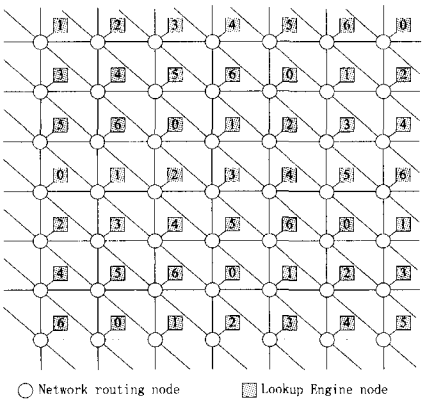


图 1 基于 SD-Torus 网络的分布式 IP 地址查找总体结构

4 SD-Torus 互连网络

4.1 定义

定义 1 直连网络可以定义为一个图 $G(V,C)$，V 表示节点集合，C 表示节点间的通信通道集合。$|V|$ 代表节点数目，$|C|$ 代表通信通道数目。$v_i(i=0,1,\cdots,|V|-1)$ 表示第 i 个节点，$c_j(j=0,1,\cdots,|C|-1)$ 表示第 j 条通信通道，记为 $c_j=(v_a,v_b)\in C$，其中 $v_a,v_b\in V$，称 v_a 为通信通道 c_j 的起点，v_b 为通信通道 c_j 的终点，并称节点 v_a,v_b 互为邻居。节点 v_i 的邻居节点的数目称为节点 v_i 的度(Degree)，记为 $deg(v_i)$。

定义 2 如果直连网络 $G(V,C)$ 所有节点具有相同的度，即 $deg(v_i)=deg(v_j),\forall v_i,v_j\in V$，则称该直连网络具有规则性(Regularity)。对于直连网络 $G(V,C)$，如果从任何节点看，整个网络都具有相同的拓扑结构，则称该直连网络具有对称性(Symmetry)。具有规则性和对称性的直连网络可以简化路由算法的设计，提高网络设计的模块化、复用性和扩展性。

4.2 SD-Torus 结构

SD-Torus 网络定义为一个具有 $t\times t$(其中 $t=7i,i=1,2,3,\cdots$)个节点的网络，网络中每个节点的度均为 6，分别连接该节点的东、西、南、北及东南、西北等 6 个邻居。SD-Torus 网络可以看作是在 2D-Torus 网络的每个节点上增加了东南及西北方向的链路，图 1 给出了一个典型的 7×7 的 SD-Torus 结构。

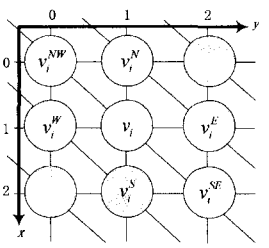


图 2 SD-Torus 网络中的节点编址及邻居表示

为了论述方便，在 SD-Torus 结构中引入 x，y 两个坐标轴。任意节点 v_i 满足 $deg(v_i)=6$，分别连接到 6 个邻居节点 (E,S,W,N,NE 及 SW 6 个方向)，即东、南、西、北、西北及东南邻居，分别记为 $v_i^E,v_i^S,v_i^W,v_i^N,v_i^{NW}$ 和 v_i^{SE}。每个节点的编址采用类似于 2D-Torus 结构的编址方法。由于 SD-Torus 结构的对称性，可以取任意节点的坐标为〈0,0〉。为了便于分析，不失一般性，在节点数目为 $t\times t$ 的 SD-Torus 结构中，假设节点 v_i 的坐标为〈x,y〉，则 v_i^E 的坐标为〈$x,(y+1)\bmod t$〉，v_i^S 的坐标为〈$(x+1)\bmod t,y$〉，v_i^{SE} 的坐标为〈$(x+1)\bmod$

$t,(y+1) \bmod t\rangle$。图 2 给出了 SD-Torus 结构中坐标轴及邻居节点的表示。

本文提出了一种分布式路由表的存储方式，即任何一个节点及其邻居节点必须存储不同的路由子表，且任何一个节点及其邻居节点上的路由子表必须构成一个完整的路由表（该条件记为临近存储条件）。因此，SD-Torus 网络的节点数目也必须满足一定的条件。分析证明，在节点数目为 $t\times t$ 的 SD-Torus 网络中，当 t 满足 $t=7i, i=1,2,3,\cdots$ 时，可以满足上述的临近存储条件。规模为 $t\times t$ 的 SD-Torus 网络记为 SD-T_t 结构。例如，图 1 给出了 SD-T_7 结构中一种可行的路由子表映射实例（每个完整的路由表划分成 7 个子表，查找引擎 LE 节点中的数字代表了路由子表的编号）。可以看出，每个节点及其邻居节点分别映射了不同的路由子表，且每个节点及其邻居节点上的路由子表构成了一份完整的路由表，满足了临近存储条件。

如果需要进行更大规模的扩展，可以使用 SD-T_7 作为基本的扩展单元进行扩展互连。使用 SD-T_7 进行扩展的互连结构同样具有良好的对称性、规则性及适合分布式 IP 地址查找的临近存储条件。

4.3 SD-Torus 结构的性质

由 $i\times i$ 个 SD-T_7 结构扩展而成的 SD-Torus 结构具有如下性质。

性质 1 SD-Torus 结构中每个节点的度均为 6，因此是规则的。从 SD-Torus 结构的任何节点出发，看到整个网络的结构是完全一致的，因此 SD-Torus 结构也是对称的。

性质 2 SD-Torus 结构的节点数目为 t^2，$t=7i, i=1,2,3,\cdots$。

性质 3 SD-Torus 结构的网络直径为 $2t/3$，其中 $t=7i, i=1,2,3,\cdots$。从图 3 的距离分布图中可以看到（详见 4.4.1 节），SD-Torus 网络中最大的节点间距离为 $t/2+(t/2)/3=2t/3$。

性质 4 SD-Torus 结构的对分带宽为 $3t$，其中 $t=7i, i=1,2,3,\cdots$。

表 1 给出了 SD-Torus 网络及其它相关互连网络的拓扑性质的比较。在节点数目相同的情况下，SD-Torus 结构同 H-Torus 结构具有相同的链路数目。与 2D-Torus 结构相比，SD-Torus 结构的网络直径较小，与 H-Torus 结构的网络直径大小相当。在对分带宽特性上，SD-Torus 结构的对分带宽特性明显优于 2D-Torus 等网络，稍低于 H-Torus 结构，网络对分带宽较大，具有良好的扩展特性。同时，与 H-Torus 结构相比，SD-Torus 结构具有规则的矩形形状，便于集成电路的设计与制造；而且 SD-Torus 满足“临近存储条件”，更加适合分布式 IP 地址查找和分布式路由表存储；另外，同 H-Torus 结构相比，SD-Torus 结构可以使用简单的二维节点编址和路由算法，易于实现。

表 1 网络拓扑性质比较

拓扑结构	节点数目 n	节点度	链路数目	网络直径	对分带宽
2D Mesh	t^2	4*	$2t^2-2t$	$2t-2$	t
2D Torus	t^2	4	$2t^2$	t	$2t$
H-Torus	$3t^2-3t+1$	6	$9t^2-9t+3$	$\frac{\sqrt{12n-3}-3}{6}$	$\frac{5\sqrt{12n-3}}{3}-5$
SD-Torus	t^2	6	$3t^2$	$2t/3$	$3t$

注：* 部分为 2 或者 3。

4.4 SD-Torus 结构的路由算法

4.4.1 距离分布规律研究

为了更清晰地描述本文提出的最短路径路由算法，先考虑在 SD-Torus 网络中相对某个源节点 v_1 的距离分布情况。在具有 $t\times t$ 个节点的 SD-Torus 结构中，假设路由源节点 v_1 的坐标为 $\langle x_1,y_1\rangle$，目标节点 v_2 的坐标为 $\langle x_2,y_2\rangle$，其中 $0\leqslant x_1,x_2\leqslant t-1, 0\leqslant y_1,y_2\leqslant t-1$。由于 SD-Torus 结构的对称性，不失一般性，假设 $x_1=t/2, y_1=t/2$。根据目标节点 v_2 相对于节点 v_1 的位置，可以将目标节点 v_2 划分到图 3 所示的 4 个区域之一。令 $d_x=(x_2-x_1) \bmod t, d_y=(y_2-y_1) \bmod t$，则 4 个区域 S_0,S_1,S_2,S_3 分别满足如下条件：

$S_0: d_x=0 \vee d_y=0$

$S_1: (d_x\leqslant\frac{t}{2}\wedge d_y\leqslant\frac{t}{2})\vee(d_x\geqslant\frac{t}{2}\wedge d_y\geqslant\frac{t}{2})$

$S_2: (d_x>\frac{t}{2}\wedge d_y<\frac{t}{2}\wedge(d_x-\frac{t}{2}\leqslant\frac{d_y}{2}\vee\frac{d_x}{2}<d_y))\vee$

$(d_y>\frac{t}{2}\wedge d_x<\frac{t}{2}\wedge(d_y-\frac{t}{2}\leqslant\frac{d_x}{2}\vee\frac{d_y}{2}<d_x))$

S_3：其它

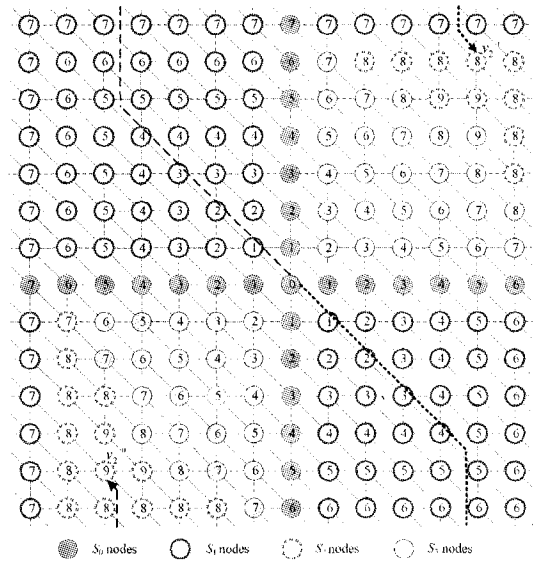


图 3 SD-Torus 网络的节点距离分布

图 3 节点中所标注的数字为该节点与节点 v_1 之间的最短距离。为了清晰起见，图中省略了边界的环回连线。

4.4.2 路由算法

本文提出了一个分布式确定性的最短路径路由算法。假设路由源节点 v_1 的坐标为 $\langle x_1,y_1\rangle$，目标节点 v_2 的坐标为 $\langle x_2,y_2\rangle$，分组的输出方向为 nexthop。由于 SD-Torus 结构的对称性，可以假设 v_1 的坐标为 $\langle t/2,t/2\rangle$。对于任意的目标节点 v_2，根据其相对于当前路由节点的相对位置，可以将其划分到 S_0,S_1,S_2,S_3 4 个区域中的某一个区域。根据图 3 中的距离分布规律，可以得出如下结论：(1)如果目标节点 v_2 位于区域 S_0，则表示 v_2 和当前节点位于同一行或者同一列，直接根据最短距离选择当前节点的邻居 v_1^E, v_1^S, v_1^W 或者 v_1^N 之一作为输出节点。(2)如果目标节点 v_2 位于区域 S_1，则直接根据最短距离选择当前节点的邻居 v_1^{SE} 或者 v_1^{NW} 作为输出节点。(3)如果目标节点 v_2 位于区域 S_2，则需要通过绕道区域 S_1 才能得到最短路径。例如，对于目标节点 v_2' 和 v_2''，图 3 中分别给出了两条最短路径。因此，根据目标节点 v_2 的具体

位置，分别选取当前节点的邻居 v_1^{SE} 或者 v_1^{NW} 作为输出节点。(4)如果目标节点 v_2 位于区域 S_3，则类似于 DO(Dimension Order)路由算法[9]，选择下一个输出节点。具体的算法如下：

```
//T=t,HR-Torus 结构中的节点数目为 t^2,下一跳的路由方向为 nexthop
int route(int x1,int y1,int x2,int y2){
    dx= x2-x1;dy= y2-y1;
    if(dx<0) dx=dx+T;if(dy<0) dy=dy+T;
    if(dx==0 || dy==0) S=0;//x-dimension or y-dimension
    else if(dx<=T/2 && dy<=T/2 || dx>=T/2 && dy>=T/2 ) S=1;
    else if((dx>T/2 && dy<T/2 &&(dx-T/2<=dy/2 || dx/2<dy)) ||(dy>T/2 && dx<T/2 &&(dy-T/2<=dx/2 || dy/2<dx)) ) S=2;
    else S=3;
     if (dx==0 && dy==0) { nexthop = IPcore; return(nexthop);} //send to local IPcore
    switch(S){
        case 0:
            if(dx==0) {
                if(dy>T/2) nexthop=v1^W;//West neighbor
                else nexthop=v1^E;//East neighbor
                break;
            }
            if(dy==0) {
                if(dx>T/2) nexthop =v1^N;//North neighbor
                  else nexthop=v1^S;//South neighbor
                break;
            }
        case 1:
            if (dx>T/2 || dy>T/2) nexthop=v1^NW;//NorthWest neighbor
            else nexthop=v1^SE;//SouthEast neighbor
            break;
        case 2:
            if ((dx>T/2 && dy<T/2 && dx+dy<T) ||(dx<T/2 && dy>T/2 && dx+dy<T) ) nexthop=v1^SE;//SouthEast neighbor
            else nexthop=v1^NW;//NorthWest neighbor
            break;
        case 3:
            if(dx<T/2) nexthop=v1^S;//South neighbor
            else nexthop=v1^N;//North neighbor
            break;
    }
    return(nexthop);
}
```

5 分布式路由表映射及分布式 IP 地址查找

采用了以上节点编址方案和节点之间的连接关系后，可以使用如下方法将路由表划分并映射到各个 SD-Torus 节点上。首先将一个完整的路由表划分为 $deg(v_i)+1=7$ 个路由子表，分别记为 $ST_0, ST_1, ST_2, \cdots, ST_6$。不失一般性，假设 v_i 节点上映射的子表为 ST_i，$i=0,1,\cdots,6$，则 v_i^S 节点映射的路由子表为 ST_j，其中 $j=(i+2) \bmod 7$，v_i^E 映射的路由子表为 ST_k，其中 $k=(i+1) \bmod 7$。

分布式 IP 地址查找可以按照如下方法进行。假设某个查找节点 v_i 收到一个 IP 地址查找任务，则可以按照路由表的划分策确定进行该 IP 地址查找的节点，即本节点或者某个邻居节点。如果可以在本地路由子表中完成 IP 地址查找，则在本地独立进行查找；如果该 IP 地址查找需要在邻居节点进行，则通过互连网络把查找任务转发到该邻居节点上，由该邻居节点完成 IP 地址查找并将查找结果返回给节点 v_i。可以看出，使用 SD-Torus 中的分布式路由表存储方式，可以将分布式 IP 地址查找的通信限制在某个节点及其邻居节点上，从而大大降低分布式 IP 地址查找的通信代价。

6 系统仿真

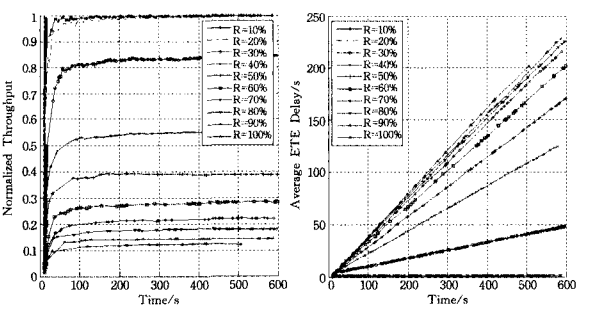
6.1 SD-Torus 路由算法的性能仿真

通过调整网络的注入速率 $\aleph$，对上述路由算法中 SD-Torus 网络的通信延迟、吞吐量以及链路负载分配进行了评估。各个节点的通信业务到达符合泊松分布，通信目标地址符合均匀分布。网络的通信延迟采用平均的端到端延迟 ETE Delay(End-to-End Delay)描述，即信息由源节点进入网络到目标网络中接收完该信息所经历的时间。吞吐量可以使用归一化吞吐量描述，即网络中成功传输的数据量与网络注入数据量的比例。链路上的负载分布情况可以通过统计每条物理链路上的数据量得出。

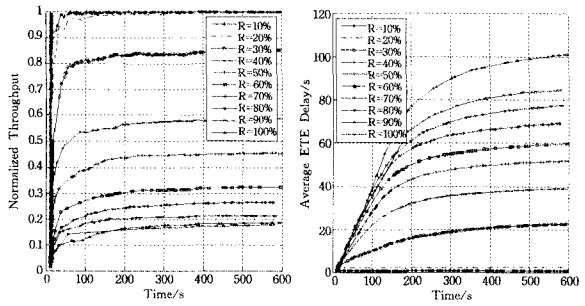
在一个 7×7 的 SD-Torus 网络中，假设链路带宽、网络节点的处理能力均为 1Gbps，网络节点中的缓冲长度为无穷大，网络流量注入速度 $\aleph$ 分别为 10%，20%，…，100%(图 4 中从高到低的顺序)时，SD-Torus 网络的归一化吞吐量和平均端到端延迟分别如图 4(a)所示。从图 4(a)中可以看出，当网络的注入速度 $\aleph$ 小于 30%时，网络的平均端到端延迟较小，归一化吞吐量接近 1；当注入速度大于 30%时，平均端到端延迟快速增加，网络归一化吞吐量急剧下降。

通过调整网络节点的缓冲长度，可以明显改善网络的平均端到端延迟特性，图 4(b)给出了网络节点缓冲长度为 32 个分组时网络的归一化吞吐量和平均端到端延迟特性。可以看出，有限长度的节点缓冲可以明显降低网络的平均端到端延迟，但是，由于缓冲长度的限制，网络的丢包率快速上升，导致网络的归一化吞吐量并没有明显改善。

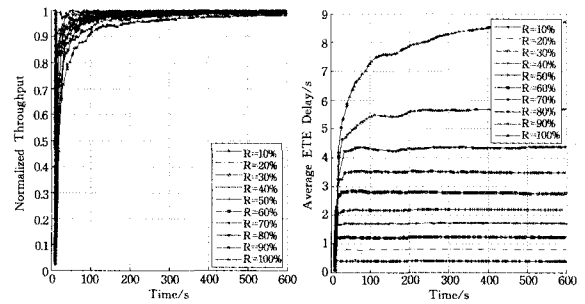
通过调整网络节点的处理能力，可以提高网络节点转发数据的效率，减少系统的丢包数量，明显降低系统的平均端到端延迟。同时，由于较快的数据处理，使得链路负载有效增加，从而显著提高网络的归一化吞吐量。图 4(c)给出了一个网络节点其处理能力为 4Gbps、节点缓冲长度为 32 个分组时的平均端到端延迟和归一化吞吐量。可以看出，网络的平均端到端延迟和归一化吞吐量性能都比较理想。



(a) 缓冲长度为无穷大



(b) 缓冲长度为 32 个分组



(c) 节点的处理能力为 4Gbps

图 4　SD-Torus 结构的平均网络延迟和归一化吞吐量

为了评估本路由算法的负载分布情况，实验采用 4 个 HR-T_7 组成 SD-Torus 结构，共有 14×14 个网络节点。图 5 示出了使用上述路由算法时，在均匀流量模式下每个节点对应东向链路、南向链路和东南向链路的负载分布情况。可以看出，整个网络上各条通信链路处于较好的负载均衡状态。

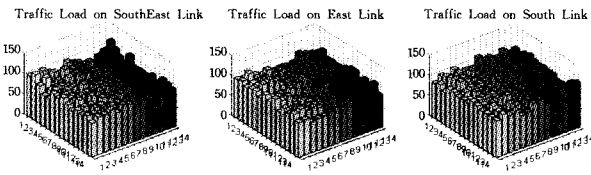


图 5　链路负载分布

6.2　分布式地址查找的性能分析比较

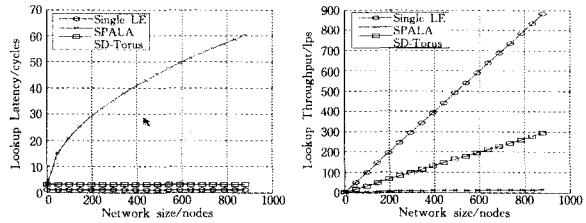
为了评估本文提出的基于 SD-Torus 网络的分布式 IP 地址查找结构的性能，对如下 3 种典型情况进行了仿真。

1)单引擎单路由表：使用单一的查找引擎，并在该引擎中存储了完整的路由表。

2)多引擎单路由表：使用 N 个查找引擎，利用 2D-Torus 结构进行互连(与 SPALA 结构类似)，按照路由表划分策略，将一份完整的路由表划分到 N 个查找中，每个查找引擎存储部分路由表。

3)多引擎多路由表：使用 N 个查找引擎，利用 SD-Torus 进行互连，并按照本文提出的路由表划分策略进行路由表映射。可以看出，该结构中总共存储了 $N/7$ 份完整的路由表。

在仿真实验中，假设基本的查找算法相同，且其查找的速度为 $T=1$ 个时钟周期，每个中间节点的通信延迟为 $T_C=2$ 个时钟周期。图 6 给出了在不同的网络规模下系统的查找速度和查找吞吐量。可以看出，在单引擎单路由表的方式下，由于各个节点之间不需要进行通信，因此查找延迟相对保持不变，查找吞吐量随着节点数目的增加而增加。然而该结构下路由表的存储代价太高，需要存储 N 份完整的路由表，因此存储的代价极大，难以实现。在多引擎单路由表的方式(SPALA 结构)下，存储代价最小，整个系统只需存储一份完整的路由表。然而，随着网络规模的增加，节点之间的通信延迟快速增加，导致地址查找的吞吐量大大下降，系统的扩展性较差。在本文提出的基于 SD-Torus 网络的分布式查找结构中，由于路由表划分后被映射到每个节点及其邻居节点上，随着网络规模的增加，分布式 IP 地址查找的通信代价不会增加，分布式 IP 地址查找的吞吐量随着网络节点数目的增加而稳定增加。同时，由于使用了分布式路由表的划分和存储，总的路由表存储代价可以下降到 $N/7$ 份完整的路由表，大大降低了路由表的存储代价。



(a) 地址查找延迟　　(b) 地址查找吞吐量

图 6　IP 地址查找的延迟和吞吐量比较

结束语　本文针对 FIB 极限问题，提出一种适合可扩展路由器的直连网络——SD-Torus 结构，并给出了相应的负载均衡的路由算法。使用本结构进行分布式 IP 地址查找时，可以将路由表分布式地存储在各个节点上，以满足在任何节点及其邻居节点上的路由子表可以构成一份完整的路由表数据，从而降低分布式地址查找的通信延迟，减少存储代价，提高系统扩展性。

参 考 文 献

[1] Aweya J. IP router architectures: an overview[J]. International Journal of Communication Systems, 2001, 14(5): 447-475

[2] Wang Y G, Du H M, Wang M M, et al. Scalable Parallel IP Address Lookup Architecture Based on 2D-Torus Network[C]// The 2nd International Conference on Information Science and Engineering(ICISE 2010). 2010: 1483-1486

[3] 郑凯. 高性能 IP 路由查找和分组分类技术的研究[D]. 北京：清华大学，2006

[4] Venkatesh K, Aravind S, Ganapath R R, et al. A high performance parallel IP lookup technique using distributed memory organization[C]// International Conference on Information Technology: Coding and Computing(ITCC'04). Las Vegas, Nevada: IEEE Computer Society, 2004

[5] Duato J, Yalamanchili S, Ni L M. Interconnection networks: An engineering approach[M]. San Francisco: Morgan Kaufmann, 2003

[6] 乐祖晖，赵有健，吴建平. 利用直连网络实现可扩展路由器[J]. 软件学报，2007，18(10)：2538-2550

[7] Chen M S, Shin K G, Kandlur D D. Addressing, routing, and broadcasting in hexagonal mesh multiprocessors [J]. IEEE Transactions on Computers, 1990: 10-18

[8] Zhao Y J, Yue Z H, Wu J P, et al. Topological properties and routing algorithms in cellular router[C]// International Conference on Networking and Services(ICNS '06). Silcon Valley, CA, 2006

[9] Dally W J, Towles B. Principles and Practices of Interconnection Networks[M]. San Francisco: Morgan Kaufmann, 2004