并行网络模拟中远程路由策略的研究

崔宇，张兆心，张宏莉，田志宏
（哈尔滨工业大学 计算机网络与信息安全技术研究中心，黑龙江 哈尔滨 150001）

摘 要：首先对并行网络模拟所采用的远程路由策略进行了分析研究，之后提出了基于优化边界的远程路由策略。该策略用边界路由器 ID 取代目的 IP 地址作为路由转发方式，有效地提高了路由查询速度。同时，通过树区域收缩、后连节点去重和边界路由器去重 3 种方法降低了内存的占用量。基于 PDNS 的实验结果表明，相对于基于边界路由器的远程路由策略，该方法降低了 85%的内存使用量，并减少了 75%的模拟时间。

关键词：并行网络模拟；远程路由；边界路由器；PDNS
中图分类号：TP393.01 **文献标识码**：B **文章编号**：1000-436X(2012)05-0115-09

Study of remote routing strategy in parallel network simulation

CUI Yu, ZHANG Zhao-xin, ZHANG Hong-li, TIAN Zhi-hong
(Computer Network and Information Security Technology Research Center , Harbin Institute of Technology, Harbin 150001, China)

Abstract: The advantages and disadvantages of several remote routing strategies for parallel network simulation were studied，and a new method based on optimize-edge routers was presented. Through converting the original forwarding style of routing IP to routing ID of edge router, the mechanism increased the speed of routing lookup. Also the exploitations of the tree contraction, edge router reduction and assnode reduction in the proposed mechanism effectively decreased the memory cost. Experimental results show 85% reduction in memory use and 75% decrease in time cost for the new routing strategy, comparing with the border-based remote routing strategy in PDNS.

Key words: parallel network simulation; remote routing; edge router; PDNS

1 引言

通过网络模拟器对网络协议和网络行为进行模拟是目前研究网络的重要方法之一。较著名的单机网络模拟器有 NS-2[1]、OPNET[2]等。受硬件资源的限制，单机模拟器已经不能满足研究者对模拟规模和模拟性能的要求。因此，并行模拟和流模拟技术作为解决这一问题的有效途径被提出并广泛使用。流模拟技术如 JSim，其通过对链路和节点的抽象来计算网络的实时状态[3]。该方法具有速度快但对网络情况描述粒度较粗的特点，适合对网络流量的整体分析。并行模拟技术如 PDNS[4,5]，其采用了数据分组级别的模拟，能有效刻画网络中任意时刻节点和链路的情况，可以对网络进行细致的分析。

收稿日期：2010-10-08；**修回日期**：2011-02-23
基金项目：国家高技术研究发展计划(“863”计划)基金资助项目（2007AA010503，2010AA012504，2011AA010705，2012AA012506）；国家重点基础研究发菜计划（“973”计划）基金资助项目（2011CB302605）；国家自然科学基金资助项目（60903166）；威海市科技攻关基金资助项目（2010-3-96）
Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2007AA010503, 2010AA012504, 2011AA010705, 2012AA012506); The National Basic Research Program of China (973 Program) (2011CB302605); The National Natural Science Foundation of China (6090 166); Weihai Municipal Science and Technology Research (2010-3-96)

在并行网络模拟中，路由表的存储与查询一直是影响模拟速度和规模的主要因素之一。文献[6]通过对各阶段模拟时间的测量，得出了调度和路由计算部分占用了整个模拟过程大部分开销的结论。文献[7]指出 PDNS 原始的远程路由策略在拓扑规模不大时即占用了很大的内存空间，路由查找效率低下。

目前，主要的远程路由策略包括：全路径路由策略、GHOST[8]路由策略、基于边界路由器的路由策略[7]和层次路由策略[6]。全路径路由策略是最早使用的远程策略，其存储了本模拟器中每个节点到整个拓扑中全部节点的路由信息。假设本模拟器中节点数为 K，整个拓扑节点数为 N，则对于本模拟器来说其远程路由表的记录数目为 KN。路由查询时，该方法需遍历本模拟器内的所有记录，时间复杂度为 $O(KN)$。为了缩短路由表长度，全路径路由策略采用了网络掩码的技术，通过只存储某一区域内 IP 地址的相同前缀来减少路由表长度。但该种方法只能处理 IP 地址比较规则且可以使用掩码的情况，对于拓扑形式比较复杂的情况则不适用。GHOST 路由策略将整个拓扑的信息存储到每一个模拟器中，其中当前模拟器包含的拓扑区域以实模式表示，即建立节点、链路、应用等，而其他区域则使用虚模式，即只存储了节点与链路的邻接信息。查找时，类似 Nix-Vector[9]，使用 BFS 算法求出源到目的最短路径 P，查找的时间复杂度为 $O(N)$，N 为节点总数。P 中实模式部分采用 Nix-Vector 进行路由，当数据分组到达源节点所在子网的边界路由器时，远程路由模块根据 P 选择相应的远程链路转发数据分组，从而完成数据分组在本模拟器内的路由。在下一个模拟器中，数据分组的路由方式有 2 种：其一是从前一个模拟器接收最短路径 P 的剩余部分进行直接转发，这种方式的特点是不用再次计算，速度快，但由于进行了路径 P 的传递，因此会产生额外的同步和通信数据，也占用了大量的内存；其二是在每个模拟器上重新计算新的路径 Pi，这种方式的特点是减少了内存占用量和模拟器之间的通信量，但增加了计算时间，并且，由于 GHOST 方式需要对每个数据分组维护一个源到目的的路径，因此该方法占用的内存随数据分组数量的增大而增大，不利于大规模模拟。基于边界路由器的远程路由策略在每个模拟器中只存储了本模拟器内的边界路由器到其他所有节点的最短距离和下一跳边界路由器。在计算节点 A 到节点 B 的最短距离时，该策略取 A 到 A 所在子网某边界路由器 Ei 与 Ei 到 B 的距离之和中最短的作为源节点所在子网的出口边界路由器，并通过其进行远程路由。与全路径方式相比，该方法有效地降低了内存占用量，并减少一定的模拟运行时间，文献[7]指出其路由表的规模只为前者的 10%，模拟时间节省 25%。层次路由策略可以理解为分层的全拓扑 Flat 路由策略，其按照每层中节点的数目分配编码。比如核心层有 K 个节点，则核心层需要使用编码地址高 $\lceil \mathrm{lb}K \rceil$bit 作编号。外围层次的节点和与其相连的内层节点有共同的前缀编码，并在其后继续编码。路由时，该方法在每个节点处只计算前缀是否一致即可，速度最快，每个节点上查找的时间复杂度为 $O(1)$。内存占用量上，该方法受限于核心层的大小，假设核心层节点数为 N，则核心层的空间复杂度为 $O(N^2)$。

上述 4 种远程路由策略中，层次路由策略速度最快但内存占用量偏高，GHOST 路由策略速度偏慢且内存占用量随数据分组量动态变化，基于边界路由器的路由策略取中，在内存占用和查找速度上取得了一定的平衡。这 4 种方式均处理不了目的 IP 地址不存在的情况，其中层次方式可以对编码地址进行最长前缀匹配，但原始 IP 仍然无法与编码 IP 对应。

本文以基于边界路由器的远程路由策略为基础，提出了基于优化边界的远程路由策略，解决了前者出现的无法进行最长前缀路由、冗余查询等问题，在降低内存占用量的同时提高了模拟的运行速度。

2 基于优化边界的远程路由策略

并行网络模拟需要将整个拓扑划分为若干个模拟区域，部署在不同的模拟器上，通过远程链路进行连接。每个模拟区域包含至少一个连通区域，每个连通区域称为一个子网。在数据分组的转发过程中，远程路由只关心源到目的节点最短路径上所经过的子网以及路径上重要的节点如边界路由器，而不关心某子网中 2 个节点如何进行本地数据分组的转发。因此，远程路由只需计算出最短路径上能正确引导数据分组跨越不同子网的重要节点信息即可。

基于优化边界的远程路由策略的核心思想是将以目的 IP 地址进行路由计算与转发的方式转换

成以边界路由器编号进行转发的方式。以图 1 所示拓扑为例，实心节点为边界路由器，空心节点为终端节点，$R1$、$R2$、$R3$、$R4$ 在 $L0$ 到 $L1$ 最短路径上。当子网 0 中 $L0$ 节点向子网 2 中的 $L1$ 节点发送数据分组时，路由模块将以 $L1$ 为基础的路由方式转化为以二元组($R1$,$R4$)为基础的路由计算方式，并以该二元组为基础进行中间子网如子网 1 上的路由转发。

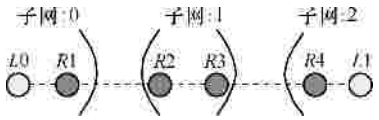

图 1　拓扑举例

为了便于描述，首先给出一些定义。

定义 1　源子网：源节点所在子网。

定义 2　目的子网：目的节点所在子网。

定义 3　源子网出口边界路由器：在源节点到目的节点的最短路径上，第一个连接远程链路的节点，如图 1 中的 $R1$ 节点。

定义 4　目的子网入口边界路由器：在源节点到目的节点的最短路径上，最后一个连接远程链路的节点，如图 1 中的 $R4$ 节点。

定义 5　中间子网：指在源节点到目的节点的最短路径上的非源子网和目的子网的子网。

由此，基于优化边界的远程路由策略的核心思想可以总结成：将以目的 IP 地址为路由计算和转发的方式转换为以源子网出口边界路由器和目的子网入口边界路由器组成的二元组进行计算和转发的方式。为了实现这种路由方式，需要完成 2 个部分的工作：第一是将目的 IP 地址转换为二元组的形式；第二是通过该二元组进行路由选择。

2.1　将目的 IP 地址转换为二元组

为了将目的 IP 转换为二元组，本策略为每个子网设计了一个存储查询框架。如图 2 所示。

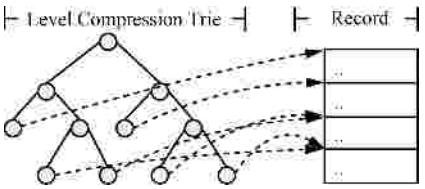

图 2　远程路由存储与查询的框架

如图 2 所示，Level Compression Trie 表示一颗层压缩树[10]，存储了拓扑中所有节点，可对目的 IP 进行最长前缀匹配，从而解决不存在节点无法路由的问题。假设拓扑节点总数为 N，则该树最多包含 $2N$ 个节点，最多查找 32 次。树中每个叶子指向 Record 中的一条记录，表示该 IP 可以使用此记录中提供的路径到达。Record 记录的格式为“(Sed1.Ded1.Len1); (Sed2.Ded2.*Len*2); …; (Sed$k$. Ded$k$. Len$k$);”。其中，“(Sed$k$.Ded$k$.Len$k$);”称作一个片段，表示从源子网边界路由器 Sed$k$ 到目的节点的长度为 Len$k$，目的子网入口边界路由器为 Ded$k$。每条记录中片段的个数与源子网中边界路由器个数一致，表示源子网中所有边界路由器到目的节点的最短路径长度和路由方式。

下面以图 3 所示拓扑为例说明 Record 中存储的内容。图中所示拓扑共 3 个子网，子网 $N2$ 中全部节点(除边界路由器)在子网 $N1$ 中存储的记录如表 1 所示。可以看到，$N1$ 子网包含 2 个边界路由器，因此每条记录均包含 2 个片段。计算路由时，需要进行比较，选择距离最短的。假设 A 节点向 B 节点发数据，A 节点通过层压缩树查找到 B 节点的 Record 记录，其内容为“(R11.R12.L1); (R21.R22. L2); …; (Ri1.Ri2.Li);”。此时有多个片段，需比较出 A 到 B 最短路径经过的片段，采用的方法是循环计算 $L_{A,\,Rk1}+L_k$ 的距离之和 $L(1 \le k \le i)$，其中，$L_{A,\,Rk1}$ 表示 A 节点到 R_{k1} 边界路由器的本地最短距离，取 L

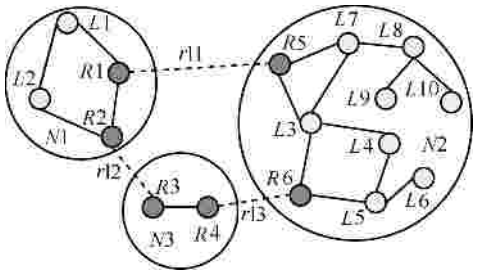

图 3　举例用拓扑

表 1　$N1$ 子网中存储的 $N2$ 子网 Record

Node	Record
$L3$	($R1$.$R5$.2);($R2$.$R5$.3)
$L4$	($R1$.$R5$.3);($R2$.$R5$.4)
$L5$	($R1$.$R5$.4);($R2$.$R6$.4)
$L6$	($R1$.$R5$.5);($R2$.$R6$.5)
$L7$	($R1$.$R5$.2);($R2$.$R5$.3)
$L8$	($R1$.$R5$.3);($R2$.$R5$.4)
$L9$	($R1$.$R5$.4);($R2$.$R5$.5)
$L10$	($R1$.$R5$.4);($R2$.$R5$.5)

最短者为 A 到 B 使用的边界路由器二元组。在处理同一子网内 2 个节点的通信时，大部分情况下不用通过远程链路，此时首先计算两节点间的本地最短距离，如果存在 Record 对应的记录则进行比较，选择距离较小的。

2.2 Record 的优化

每条 Record 记录存储了源子网中所有边界路由器到目的节点的片段，假设子网 S 中有 1K 个边界路由器，整个拓扑 1M 个节点，片段长 6byte，那么 S 对应的 Record 总量为 1K×1M×6byte=6Gbyte，这占用了大量的内存空间。因此，需要采用逐个子网计算的方式，每次只计算一对子网的 Record，减少内存占用量。为此，本文引入了采用基于 CQ(calendar queue)[11]方式的 BFS 算法对每个目的子网进行计算。该算法以距离为纵线，以与源边界路由器最短距离相同的节点组成横线。举例来说，当计算 $R2$ 对子网 $N2$ 中所有点的片段时，形成的 CQ 初始情况如图 4 所示。

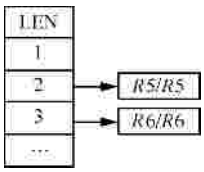

图 4 $R2$ 节点对 $N2$ 子网的 CQ 初始情况

图中 $R5$ 与 $R2$ 最短距离为 2，因此插入第二行，前后 2 个值分别表示当前扩展到的节点和到达本节点的入口边界路由器。同理 $R6$ 插入到第三行。计算时从距离为 1 开始，逐次取出每行中的节点，将没有被插入过的邻居节点插入到下一行，表示距离加 1，第二个分量不变。通过从边界路由器向子网内部的扩散，$N2$ 子网中所有的点均可计算出对应于 $R2$ 的片段。当 $N1$ 中所有边界路由器对 $N2$ 子网均进行一次扩散后，$N1$ 即存储了 $N2$ 子网中所有节点的记录，且每条记录中片段的数目与源子网中边界路由器的数目一致。

虽然可以通过逐个子网计算的方式来减少计算时内存的使用量，但记录的总量并没有改变。为此，本文使用了 3 种方法来缩减 Record 记录数目和每条记录的片段数目。

1) 树区域收缩(tree contraction)

所谓树区域收缩就是将拓扑图中树区域包含的节点用根节点表示，省去非根节点的记录，以减少记录数目。树区域收缩不影响路由选择结果，下面给出证明。

结论 经树区域收缩的路由表在路由选择上与原始情况一致。

证明 假设 Ri、Rj 为源子网边界路由器，Tr 为目的子网中一树根节点，Ri、Rj 与 Tr 形成的记录为“($Ri.Rr1.Pr1$);($Rj.Rr2.Pr2$)”。Tx 为树中任一节点，Ri、Rj 与 Tx 形成的记录为“($Ri.Rx1.Px1$); ($Rj.Rx2.Px2$)”。首先，由贪心算法可知，Tr 一定在 Tx 到 Ri 的最短路径上，因此 2 段路径重合并且由目的子网入口边界路由器的定义可知，$Rr1 = Rx1$，同理 $Rr2 = Rx2$。可见，树中节点与树根节点记录中对应片段的边界路由器一致，只是距离不同。在计算源子网中某一节点 A 到 Tx 的最短路径时，要比较 $L_{A,Ri}+Px1$ 和 $L_{A,Rj}+Px2$ 的大小，两端同时减掉 $L_{Tr,Tx}$ 的大小，比较结果不变，形式变为 $L_{A,Ri}+Pr1$ 和 $L_{A,Rj}+Pr2$。而这与比较 A 到 Tr 的方式一样，因此树区域中的节点可由树根节点代替，收缩后路由情况不变。证毕。

经树区域收缩后，由于树区域中的节点均被树根节点代替，因此 Record 记录的条数得到了减少。对图 3 所示拓扑而言，经树区域收缩后，其拓扑如图 5 所示。

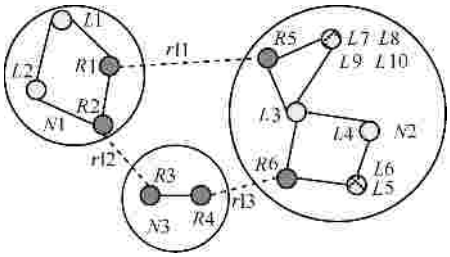

图 5 树结构收缩后的拓扑

图 5 中，网格线标识的节点为树区域的根节点，可以看到，$N2$ 包含了 2 个树形区域，共计 6 个节点。经树区域收缩后，对于源子网 $N1$ 而言，目的子网 $N2$ 对应的 Record 如表 2 所示。可以看到，树区域中的节点均指向了树根的记录。

表 2 树区域收缩后 $N1$ 中存储的 $N2$ Record

Node	Record
$L3$	($R1.R5.2$);($R2.R5.3$)
$L4$	($R1.R5.3$);($R2.R5.4$)
$L5$,$L6$	($R1.R5.4$);($R2.R6.4$)
$L7$,$L8$,$L9$,$L10$	($R1.R5.2$);($R2.R5.3$)

2) 后连节点去重(assnode reduction)

后连节点去重是减少 Record 记录数目的另

一种方法，用于树区域收缩之后，对收缩后的拓扑进行进一步处理。下面首先给出后连节点的定义。

定义 6　后连节点(assnode): 设某子网 S，SET(S)表示 S 上所有边界路由器的集合，P、Q 为另一个子网中邻接的 2 个点。若 SET(S)中的边界路由器到 Q 点的最短路径均经过 P，则称 Q 为 P 的后连节点，P 为 Q 的前导节点。

若 Q 为 P 的后连节点，则 Q 可用 P 的记录代替以进一步减少记录数目，其证明思想与树区域收缩的证明类似。后连节点去重依然采用源子网 S 对目的子网 D 进行 BFS 扩散。D 中每个节点存储了其邻接节点的链表，当从 S 中 Ri 边界路由器对 D 进行扩散时，如果 D 中的节点 B 是从 A 节点扩散到的，则将 A 的邻居节点链表中 B 对应元素的引用计数加 1。若 B 节点对应元素的引用计数和 S 中边界路由器个数相等，则说明 B 为 A 的后连节点。后连节点具有递归性，如果 B 是 A 的后连节点，C 是 B 的后连节点，则 B、C 节点均可以用 A 代替。下面给出后连节点的标识算法。

算法 1　后连节点标识算法

```
1) procedure Compute_AssNode(S, D)
2) For all Ri∈S do
3)    init Calendar queue of D;
4)    while in Calendar BFS do
5)       if B spreads from A then
6)          A.adjlist[B].count++;
7)          end if
8)    end while
9) end For
10) edcount ?  S.EDRT.count;
11) For all node∈D do
12)      For all adjnode∈node.adjlist do
13)         if adjnode.count = edcount then
14)       adjnode is AssNode of node;
15)      end if
16)    end for
17) end for
18) end procedure
```

后连节点去重后，目的子网对应的记录数目进一步减少。以图 5 所示拓扑为例，经后连节点去重，$N1$ 中 Record 如表 3 所示。

表 3　后连节点去重后 $N1$ 中存储的 $N2$ Record

Node	Record
$L5$，$L6$	($R1.R5.4$)；($R2.R6.4$)
$L3$，$L4$，$L7$，$L8$，$L9$，$L10$	($R1.R5.2$)；($R2.R5.3$)

3) **边界路由器去重**(edge router reduction)

经树区域收缩和后连节点去重后，子网中存储的 RECORD 数目得以有效的减少，形成了多个节点对应一条记录的形式。边界路由器去重对每条记录进行分析，删除其中的冗余片段，从而进一步减少内存占用。下面给出冗余片段的定义。

定义 7　冗余片段：设源子网 $S1$，目的子网 $S2$，Ri、Rj 为 $S1$ 中的 2 个边界路由器，P 为 $S2$ 中的一个节点，P 的记录为“($Ri.Rx.Pi$);($Rj.Ry.Pj$)”。若 $Pi+L_{Ri,Rj}\leq Pj$，其中，$L_{Ri,Rj}$ 表示两者之间的距离，则称 Rj 的片段对于 Ri 而言是冗余的。

冗余片段是可以去掉的，如图 6 所示，假设 A 节点向 P 节点发送数据分组，如果存在关系 $Pi+L_{Ri,Rj}\leq Pj$，则 A 一定不会选择 A-Rj-P 的路线，因为 A-Rj-Ri-P 的距离可能更短，所以 Rj 对应片段是冗余的，可以去掉。

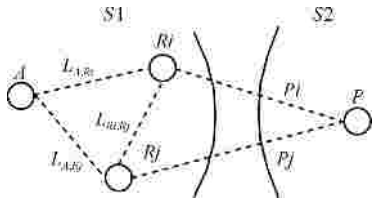

图 6　冗余片段示意

经过边界路由器去重，图 5 所示的拓扑中 $N1$ 子网存储的 $N2$ 子网的记录如表 4 所示。

表 4　边界路由器去重后 $N1$ 中存储的 $N2$ Record

Node	Record
$L5$,$L6$	($R1.R5.4$);($R2.R6.4$)
$L3$,$L4$,$L7$,$L8$,$L9$,$L10$	($R1.R5.2$)

树区域收缩，后连节点去重和边界路由器去重后，Record 的条数和每条记录的片段数得以有效的减少，总的内存占用量也随之减少。经过远程路由的查询和比较后，源子网出口边界路由器和目的子网入口边界路由器组成的二元组得以确定，进而开始数据分组的路由。如果源到目的最短路径本地可达，则不需经过边界路由器。

2.3 边界路由器二元组的路由

路由转发时，源节点通过本地路由将数据分组转发给源子网出口边界路由器，之后，数据分组在中间子网的转发由边界路由器完成，到达目的子网后，目的子网的入口边界路由器通过本地路由将数据分组送达目的节点。

为了对边界路由器二元组进行路由，首先将拓扑中所有的边界路由器编号，同一个子网中的编号连续。如图 5 中，*R*1 编号为 1，*R*2 编号为 2，以此类推。其次，每个边界路由器保存了 2 个结构：NHP、RLP，其中 NHP 称为下一跳存储表，采用数组存储了本边界路由器到所有边界路由器的下一跳序列，存储空间复杂度为 $O(N)$，N 为整个拓扑边界路由器总数，查找的时间复杂度为 $O(1)$。RLP 称为远程链路存储表，存储了与该边界路由器连接的所有远程链路和另一端的边界路由器编号，空间复杂度为 $O(k)$，查找的时间复杂度为 $O(k)$，k 为连接的远程链路个数。图 7 显示了 *R*1 节点上存储的 NHP 和 RLP 的内容。

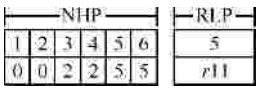

图 7 *R*1 节点上的 NHP 与 RLP

图 7 中，*R*1 边界路由器中保存的 NHP 序列为：0.0.2.2.5.5。第一个 0 表示到编号 1 即自身的下一跳；第二个 0 表示到编号 2 边界路由器的下一跳，由于不用跨子网，因此也为 0。*R*3 时，需经过 *R*2，因此记录为 2。

边界路由器可以从 2 种链路接收数据分组：本地链路、远程链路。从本地链路接收到数据分组时，如果该数据分组目的地址不是自身，则该数据分组一定要通过该边界路由器出本子网。因为当该边界路由器作为本地路径中的一个普通节点时，数据分组并不会进入边界路由器的远程计算模块，而是通过节点的本地路由模块转发了。所以，这种情况下只需通过数据分组中目的子网入口边界路由器编号在 NHP 中查找下一跳的边界路由器编号，然后进入 RLP 查找对应的远程链路进行转发，此时 RLP 中一定存在相应记录。

当边界路由器从远程链路接收到数据分组时，除目的为自身外有 3 种情况：第一，目的节点为本子网中的某个节点，此时直接调用本地路由发送到目的节点即可；第二，数据分组从本子网的另一个边界路由器出本子网，此时需要将数据分组通过本地路由转发到该边界路由器；第三，通过本边界路由器的其他远程链路发送出去。下面给出查询算法。

算法 2 远程链路接收处理算法

```
1)  procedure remoterecv(Packet)
2)    Dst_ed ? Packet->ded;
3)    if Dst_ed = this_id then
4)      call localroute to Dst node;
5)      return
6)    end if
7)    Nxt_ed ?  NHP[Dst_ed];
8)    For all (ed, rlink) in RLP do
9)      if ed = Nxt_ed then
10)       send to rlink;
11)       return
12)     end if
13)   end for
14)   call localroute to ED of Nxt_ed;
15) end procedure
```

3 实验结果与分析

本节对基于优化边界远程路由策略的性能进行了测试，主要指标是内存占用量和模拟时间开销。实验从启明星辰提供的全国拓扑数据库中选取了 8 个拓扑，每个被划分为 4 个区域，每个区域用一台服务器进行模拟（3.0GHz 的 CPU、4GB 内存）。表 5 显示了测试拓扑的详细信息。其中，*Router* 为路由器总数，*TR* 为节点度为 1 的路由器数目，*Ave_De* 表示路由器的平均节点度，*Node* 表示每个 TR 上绑定 10 个主机后拓扑节点总数，*ED* 表示每个划分区域的边界路由器个数，*Sub* 表示每个划分区域的子网数。

表 5 测试用拓扑环境

Case	*Router*	*TR*	*Ave_De*	*Node*	*ED*	*Sub*
1	3 432	1 091	5.15	14 342	80/69/19/34	2/1/2/1
2	5 098	1 886	5.02	23 958	103/52/21/30	1/2/1/2
3	6 334	2 448	5.31	30 814	148/150/47/39	1/1/2/2
4	8 171	3 251	4.53	40 681	124/89/5/60	1/1/1/1
5	10 653	4 327	3.80	53 923	62/60/19/22	1/2/1/1
6	12 237	5 025	4.04	62 487	88/105/17/48	1/1/3/1
7	14 106	6 145	3.68	75 556	77/60/25/21	1/2/1/1
8	15 690	6 483	3.87	84 120	126/66/78/29	1/1/1/2

基于优化边界的路由策略主要有 3 个存储结构：层压缩树（CT）、RECORD（RE）和边界路由器信息（ED）。实验首先测试了 2.2 节中 3 种优化方法在降低 RE 数量上的性能，之后得出了优化后 RE 片段总量缩减的比例，然后给出了 CT、RE、ED 各自使用的内存量，最后通过与基于边界路由器的远程路由策略进行内存总量对比得出内存优化的整体性能。

优化前，RE 部分占用的内存最大。因此，实验使用了树区域收缩(TC)、后连节点去重(AS)和边界路由器去重(EC)3 种方法对 RE 进行了优化，各方法的优化结果如图 8 所示。

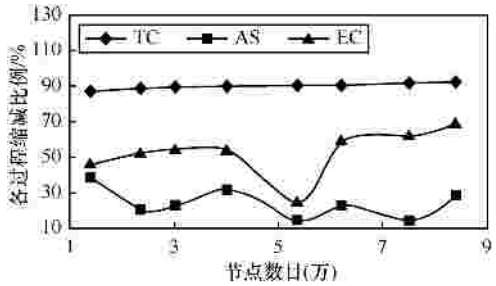

图 8　各方法缩减比例

图 8 中，树区域收缩对减少 RE 条数的贡献度最大，缩减了 90%左右的记录条数。这与每个度为 1 的路由器上绑定 10 个主机相关，可以预计，随着主机数目的增加，树区域收缩贡献度也会越大。后连节点去重总体上徘徊在 50%左右，性能次于树区域收缩。其在第 5 个用例时出现了严重的波动，是受到了拓扑形式变化的影响，存在一定的不稳定性。边界路由器去重的效用较低，在 20%左右，也随着拓扑的变化出现了一定的波动。经过优化，RE 片段的总数目得到了有效缩减，缩减比例如图 9 所示，其中，MAX、AVE 和 MIN 分别表示 4 个区域中比例最高、平均和最低的值。从中可以看出，经过简化，各模拟器在不同拓扑条件下的 Record 片段数目有了很大减少，平均减少 97%，最少减少 94%。

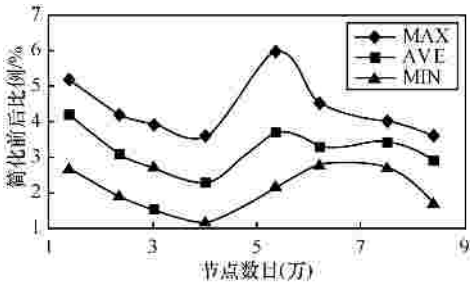

图 9　各模拟器 RECORD 缩减前后比例

优化后，CT、RE、ED 各自内存的占用量如图 10 所示。可以看到，RE 部分占用的内存量得以有效减少，最大值不超过 1Mbyte。ED 占用内存极少，基本可以忽略不计。CT 占用量最大，主要有 2 个原因。其一是 CT 的个数与一个划分区域内的子网个数一致，如 CT 曲线的最高点和次高点，虽然次高点的拓扑规模较大，但由于最高点子网数为 3 而次高点为 2，因此导致了拓扑较大但不是最高点的情况，可见本策略受划分区域子网个数影响较大。其二是每棵 CT 中，树节点数目最大为拓扑节点总数 2 倍，因此随着拓扑规模的增大，该结构占用的内存量将会线性增长。

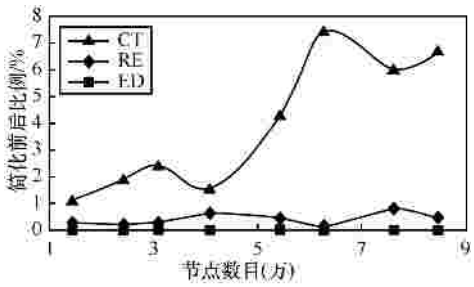

图 10　实验中各部分内存占用量

图 11 则显示了每个实例中，内存占用量最大的区域上，2 种策略占用的内存总量。

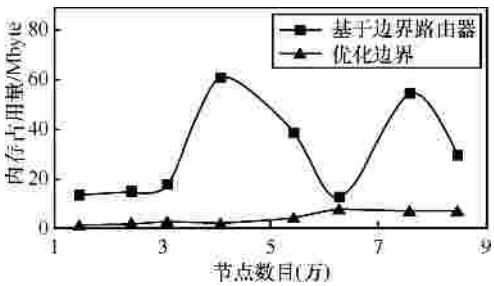

图 11　2 种路由策略的内存占用

可以看出，基于优化边界的远程路由策略内存占用量最大不过 10Mbyte，而基于边界路由器的远程路由策略最大超过 60Mbyte。两者相比，前者占用内存量最大为后者的 45%，平均为 14%。可见前者平均减少了超过 85%的内存占用量，因此在内存使用上优于后者。值得注意的是，后者的结果曲线在图 11 中横坐标为 6 时出现了较大波动，这主要是受边界路由器个数的影响，图 11 中横坐标为 4 时边界路由器为 124，而图 11 中横坐标为 6 时数目仅为 17，可见基于边界路由器的路由策略受边界路由器数目影响较大。

远程路由的计算量主要体现在以下 2 种节点上。其一是源节点，源节点向外发送的每个数据分组均需进行远程查询以确定数据分组发送的方向：直接通过本地路由发往目的节点、或者发到边界路由器上出本子网。其二是边界路由器，当需要往外子网发送的数据分组到达边界路由器，或者从外子网到本子网时，均会在此进行远程路由查找。为了平衡 2 种节点处的计算量，本实验设计了一个应用程序并将其绑定在 100 个分散的主机上，该应用循环对拓扑中所有主机 IP 发送数据分组，发分组间隔 0.001s，每秒数据分组总量为 100k，模拟过程中数据分组总量为 100n，n 为叶子节点总数。这样做的目的是，尽可能地让数据分组通过拓扑中所有的路径，避免出现数据分组过度集中在某几条链路上和过多或过少经过边界路由器的情况，从而更好地体现远程路由策略的整体性能。

图 12 显示了基于优化边界和基于边界路由器的远程路由策略在实验中的运行时间。可以看出，前者消耗的模拟时间明显短于后者，时间比平均在 25%以下。同时随着拓扑的增大，前者的增长速度也慢于后者，基本以线性的速度增长。

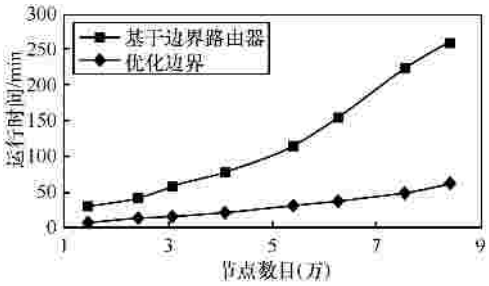

图 12 2 种模拟策略的模拟时间

图 13 所示为模拟中使用 2 种远程路由策略时模拟器平均每分钟处理新生数据分组的能力。可以看出，基于优化边界的远程路由策略每分钟可处理 10 000 以上的新生数据分组，而后者处理能力最高不到 5 000，可见前者对模拟的整体性能提高很大。

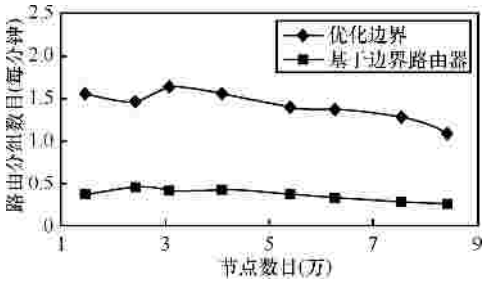

图 13 2 种策略下模拟器的处理能力

综上实验表明：在内存占用上，树区域收缩、后连节点去重和边界路由器去重共减少了 95%左右的路由表，有效地解决了拓扑规模较大时路由表过长的问题。相对于基于边界路由器的远程路由策略，该方法降低了 85%的整体内存使用量，达到了良好的效果。在模拟速度上，图 12 和图 13 说明该策略有效地提高了模拟的性能，减少了 75%以上的整体模拟时间。实验也表明，该策略的内存占用量与整个拓扑的子网数目关系很大。当子网数目增加时，层压缩树数目也会增加，从而增加内存的占用量。另外，后连节点和边界路由器去重算法有一定的局限性和不稳定性，对一些特定的拓扑形式无法进行去重。

4 结束语

路由的存储和查找是影响网络模拟性能的重要因素，相关的研究也提出了诸多降低内存和提高模拟速度的解决办法。本文针对大规模的网络模拟环境，以降低内存占用为主，提出了基于优化边界的远程路由策略，使用树区域收缩、后连节点去重和边界路由器去重 3 种方法，大幅地降低了内存占用量，同时用边界路由器 ID 取代目的 IP 作为路由转发方式，有效地降低了路由查找时间。实验中出现的问题，如层压缩树与子网个数相关和后连节点、边界路由器去重不稳定等可作为进一步研究的方向。

参考文献：

[1] Network simulator-ns-2[EB/OL]. http://www.isi.edu/nsnam/ns, 2004.

[2] DORLEU S, HOLWECK J, REN R. Modeling and simulation of fading and pathloss in OPNET for range communications[A]. Radio and Wireless Symposium IEEE[C]. 2007. 407-410.

[3] LIU Y, PRESTI F L, MISRA V. Scalable fluid models and simulations for large-scale IP networks[J]. ACM Transactions on Modeling and Computer Simulation(TOMACS), 2004,14(3):305-324.

[4] RILEY G, FUJIMOTO R, AMMAR M. A generic framework for parallelization of network simulations[A]. Proceedings of Seventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication System[C]. College Park, 1999. 128-135.

[5] SZYMANSKI B K, SAIFEE A, SASTRY A. Genesis: a system for large-scale parallel network simulation[A]. The 17th Workshop on

Parallel and Distributed Simulation(PADS'03), IEEE[C]. San Diego California, 2003. 61-68.

[6] 李博. PDNS 性能提高策略研究与实现[D]. 哈尔滨:哈尔滨工业大学, 2009.
LI B. The Research and Implementation of Strategy for g the Performance of PDNS[D]. Harbin: Harbin Institute of Technology, 2009.

[7] 郝志宇. 并行网络模拟中的远程路由计算和查找方法[J]. 通信学报, 2007,28(6): 66-73.
HAO Z Y. Approach to remote routing computation and lookup in parallel network simulation[J]. Journal on Communications, 2007,28(6): 66-73.

[8] RILEY G, JAAFAR T, FUJIMOTO R. Using ghosts for global topology knowledge in space-parallel distributed network simulations[J]. Simulation, 2005,81(4): 267-277.

[9] RILEY G, AMMAR M, FUJIMOTO R. Stateless routing in network simulations[A]. Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems[C]. San Francisco, 2000. 524-531.

[10] 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, 2006.30-31.
ZHENG K. Research on High Performance IP Route Lookup Packet Classification[D]. Beijing: Tsinghua University, 2006.30-31.

[11] AHN J S, OH S H. Dynamic calendar queue[A]. Proceedings of the Thirty-Second Annual Simulation Symposium[C]. San Diego, CA, 1999. 20-25.

作者简介：

崔宇（1985-），男，黑龙江哈尔滨人，哈尔滨工业大学博士生，主要研究方向为网络安全。

张兆心（1979-），男，黑龙江哈尔滨人，博士，哈尔滨工业大学副教授，主要研究方向为网络安全。

张宏莉（1973-），女，吉林榆树人，哈尔滨工业大学教授、博士生导师，主要研究方向为计算机网络信息安全和并行处理。

田志宏（1978-），男，黑龙江哈尔滨人，博士，哈尔滨工业大学副研究员，主要研究方向为网络安全主动防御和入侵取证。