

Zone_Cut 网络模拟本地路由策略

张兆心^{1,2}, 杜跃进^{1,3}, 王克¹, 王轩春¹, 郝志宇²

(1. 哈尔滨工业大学 网络与信息安全研究中心, 黑龙江 哈尔滨 150001;

2. 中国科学院 计算技术研究所, 北京 100190; 3. 国家计算机网络应急技术处理协调中心, 北京 100029)

摘要:提出并实现了 Zone_Cut 网络模拟本地路由策略, 根据节点属性的不同将节点分为 T 区、LD 区和 HD 区。对不同的区域采用不同的存储和查找策略, 降低并更好地平衡了路由存储空间和查找时间。基于 PDNS 的实验结果表明, Zone_Cut 路由策略比 MTree_Nix 路由策略综合性能有大幅提高。低频分组情况下, 模拟时间平均减少 18.08%, 模拟空间平均减少 51.23%; 高频分组情况下, 模拟时间平均减少 55.29%, 模拟空间平均减少 74.4%。

关键词:网络模拟; 路由策略; Zone_Cut; MTree_Nix

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2013)07-0014-10

Local routing strategy in network simulation based on Zone_Cut

ZHANG Zhao-xin^{1,2}, DU Yue-jin^{1,3}, WANG Ke¹, WANG Xuan-chun¹, HAO Zhi-yu²

(1. Network and Information Security Research Center, Harbin Institute of Technology, Harbin 150001, China;

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;

3. National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

Abstract: Local routing strategy based on Zone_Cut was put forward for network simulation. All nodes in topology were divided into three zones, including T Zone, LD Zone and HD Zone according to the property of the node. To the storage space and search time of routing strategy, different storage and search mechanisms were adopted for different zones, which balanced the space and time of routing mechanism. Experimental results on PDNS show that Zone_Cut improves the comprehensive properties of routing strategy, compared with the MTree_Nix. Compared with MTree_Nix, it can reduce the simulation time and simulation space by about 18.08% and 51.23% respectively in low frequency, while in high frequency, it can reduce the simulation time and simulation space by about 55.29% and 74.4% respectively.

Key words: network simulation; routing strategy; Zone_Cut; MTree_Nix

1 引言

随着网络技术和互联网的高速发展, Internet 呈爆炸性增长, 并且已深入到社会和人民生活的方方面面, 为了更好地建设和发展互联网, 需要对互联网及其行为进行更加深入的研究。而互联网庞大的规模, 以及异构性、复杂性等为其研究带来巨大的

困难^[1]。网络模拟因其可以详细刻画互联网的各种行为, 并具有成本低、灵活、多变、支持多种协议等特点成为互联网研究的主要方法和研究的热点^[2-4]。

模拟的规模和效率是评价网络模拟器性能的重要指标。如何在有效的硬件资源下, 提高模拟的规模和效率成为网络模拟研究工作的重点。影响模拟规模和效率的因素有很多, 其中路由策略为最关

收稿日期: 2012-08-01; 修回日期: 2013-03-20

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); 国家自然科学基金资助项目(61100189, 61003261); 国家科技支撑计划基金资助项目(2012BAH45B01); 山东省中青年科学家奖励基金资助项目(BS2011DX001); 威海市科技攻关基金资助项目(2010-3-96); 哈尔滨工业大学科研创新基金资助项目(HIT.NSRIF.2011119)

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2007AA010503); The National Natural Science Foundation of China (61100189, 61003261); The National Science and Technology Support Program(2012 BAH45B01); Young and Middle-Aged Scientists Research Awards Fund of Shangdong Province(BS2011DX001); Weihai Municipal Science and Technology Research(2010-3-96); Harbin Institute of Technology Scientific Research Innovation Foundation (HIT.NSRIF.2011119)

键因素之一。从空间复杂性上来说,大规模网络模拟中路由的内存需求是一个主要的瓶颈^[5];从时间复杂性上来说,路由表的建立、计算和查找效率是影响路由性能的重要因素。网络模拟路由策略包括本地路由策略^[6,7]、远程路由策略^[8,9]和动态路由策略 3 种。其中本地路由策略是网络模拟路由策略的核心和基础。当前网络模拟中的路由策略主要包括如下几种。Flat 策略:该方法对节点规模为 n 的网络拓扑来说,空间复杂度为 $O(n^2)$,查找源、目的节点间的路由时,仅需在 $O(1)$ 时间内即可得到所需路由信息,路由查找时间最小。Hierarchical 路由策略与 Flat 路由策略相似,唯一的区别在于对于拓扑中的骨干网、本地区域和 AS 域分别进行单独的路由计算,空间复杂性是 $O(kn \log_k n)$,其中 k 是每一个自治区域的子网数。建立路由表需要 $O(n^{2.3} \sqrt{n})$,查找路由表也是 $O(1)$ 。AR(algorithmic routing)策略^[10]:该路由机制在全局网络拓扑和 k -ary 树之间进行地址映射;该路由表占用的存储空间为 $O(n)$,而每一次查找 k -ary 树的时间复杂度为 $O(\log_k n)$ 。STree_Flat 策略^[11]:该方法采用一棵近似优化的最小生成树来保存尽可能多的路由信息,而对不能被该树包含的最短路径路由信息保存在一张静态路由表中,该方法在保证查询路由信息准确的前提下节省了大量空间,但综合效果较差。MTree_Flat 策略^[12]:该方法由于采用多棵最小生成树来保存路由信息,与 STree_Flat 方法相比节省了更多存储空间,查找静态路由表所需时间也更小,综合效果更优。Nix-Vector 策略^[13]:该方法在模拟初始化过程中不进行任何的路由计算和存储工作,而是在数据分组生成时根据源、目的地址信息实时计算路由,并将计算出的路由信息保存在数据分组头中,当数据分组经过当前路由器节点时,根据数据分组头携带的信息解析出该数据分组的路由,若节点数为 n ,平均路径长度为 k ,平均节点出度为 d ,则该方法所需要的存储空间为 $O(knlgd)$ 。每次模拟一个新的数据分组传输过程都需要遍历全局拓扑以计算路由,其时间复杂度很大,若使用邻接表方式描述的拓扑结构,则查找时间复杂度为 $O(n+e)$,其中 e 为边的数目。MTree_Nix 策略^[14,15]:该方法引进了多棵最小生成树作为路由表,将未被任何最小生成树覆盖的最短路径的源、目的地址信息保存到 Cache 中。路由查找时,首先查找 Cache 来确定源、目的节点的最短路径是否被最小生成树覆盖,若被最小生成

树覆盖,则查找多棵最小生成树,选择路径最短的路由作为下一跳使用的路由信息;若未被任何最小生成树覆盖,则生成源、目的节点的 Nix 向量并返回下一跳路由信息。其中,Flat、AR、STree_Flat、MTree_Flat 和 Nix-Vector 都是利用增加路由查找时间的办法减少路由存储空间,但当面对大规模复杂网络模拟时,往往会导致难以忍受的模拟运行时间。虽然 MTree_Nix 路由策略通过增加少许的存储空间,在空间和时间上获得了较好的平衡。但是当节点规模数达到 3 000 及以上时,内存空间仍然成为大规模模拟的瓶颈。

因此,在大规模网络模拟中,如何有效地降低路由存储空间和查找时间,提高模拟的规模和效率,成为本地静态路由策略急需解决的问题。为此,本文提出 Zone_Cut 本地静态路由策略。根据网络拓扑结构特点和节点度对网络拓扑进行区域分割,从而针对各区域采用高效的数据存储结构和查找策略,降低路由的存储空间和查找时间,提高模拟的规模和效率。

2 Zone_Cut 路由策略

Zone_Cut 路由策略的核心思想是区域分割,进而根据区域属性建立相应的存储和查找策略,故首先根据网络拓扑结构特点和节点度对路由由节点组成的拓扑进行区域划分,形成树区(T区)、低度区(LD区)和高度区(HD区)3个子域。进而根据各区域性质设计不同数据存储结构和查找策略。针对T区,采用树型结构存储,查找中使用最早公共祖先的方法;针对LD区,采用比特图的形式进行存储和查找;针对HD区,采用邻接表存储,查找中采用BFS算法。然后为提高效率,对经区域分割后的节点重新编码。接着为降低拓扑规模,对T区进行抽象,同时定位T区代表节点。最后为快速定位各节点所属区域,从而选取不同的路由存储与查找策略,采用节点着色算法,根据节点属性的不同着以不同的颜色。

2.1 拓扑形式化描述

在大规模并行网络模拟中,需要将模拟任务划分给不同的模拟节点执行,任务划分普遍采用基于拓扑的划分方式,将整个待模拟网络拓扑根据需求划分为 n 份,即每一份拓扑及其上的网络行为由一个模拟节点进行模拟,共有 n 个模拟节点并行执行模拟任务。将整个网络拓扑采用无向图 $G(V, E)$ 表示,其中, V 表示节点, E 表示边。为描述方便省

略终端主机并做如下定义与假设。

定义 1 模拟节点：设一个模拟任务需要 n 个并行网络模拟节点共同执行，则 $SIM_i (1 \leq i \leq n)$ 为一个模拟节点。

定义 2 节点度：节点 v_i 的度定义为 $d(v_i) = \sum_{e_{ij} \in E} e_{ij}$ 。

定义 3 普通链路：如果 e_{ij} 为普通链路，则 $e_{ij} | v_i, v_j \in SIM_k, i \neq j$ 。

定义 4 远程链路：如果 e_{ij}^r 为远程链路，则 $e_{ij}^r | v_i \in SIM_p, v_j \in SIM_q, i \neq j, p \neq q$ 。

定义 5 终端路由器：如果节点 v_i 连接终端主机，且 $d(v_i) = 1$ ，则称 v_i 为终端路由器。

定义 6 边界路由器：远程链路 e_{ij}^r 两端的节点 v_i 和 v_j 。

定义 7 普通路由器：除边界路由器和终端路由器外其他的路由器。

定义 8 子网：因拓扑划分导致整个网络拓扑由多个连通分支组成，每个连通分支称为一个子网。

将启明星辰 2009 年探测的部分全国拓扑划分成 4 个部分，每个部分由一个模拟节点进行模拟，拓扑划分后数据信息如图 1 所示。

从图 1(a)中可以看出，各模拟节点中普通路由器约占 59.32%，终端路由器约占 38.51%，边界路由器约占 2.16%。从图 1(b)中可以看出，各模拟节点中分别有 4、6、1、16 个子网。从图 1(c)中可以看出，度为 1 的节点占 42.65%，度在 [2,4] 占 33.67%，[4,8] 占 12.03%，其中度小于 4 的节点占据 76.32%。如果根据拓扑结构特点对某些低度节点进行抽象，如树型抽象，必可降低拓扑的规模，减少路由的存储空间和查找时间。同时根据抽象后路由器度的分布情况，将拓扑分成多种区域，针对不同区域，采用不同的存储和查找方式，可进一步降低路由的存储空间和查找时间。

2.2 拓扑结构分析

设有 n 个 SIM，利用拓扑划分算法将整个拓扑图 G 分成 n 个拓扑子图 G_i ，拓扑子图之间使用远程链路进行连接。设 V_i 、 E_i 分别表示子图 G_i 的节点和普通链路集合， R_i 表示子图 G_i 与其他子图远程链路集合， R 表示所有子图间远程链路集合， R_i 表示子图 G_i 和其他子图之间的远程链路集合， R_{ij} 表示子图 G_i 和 G_j 之间远程链路集合，即有

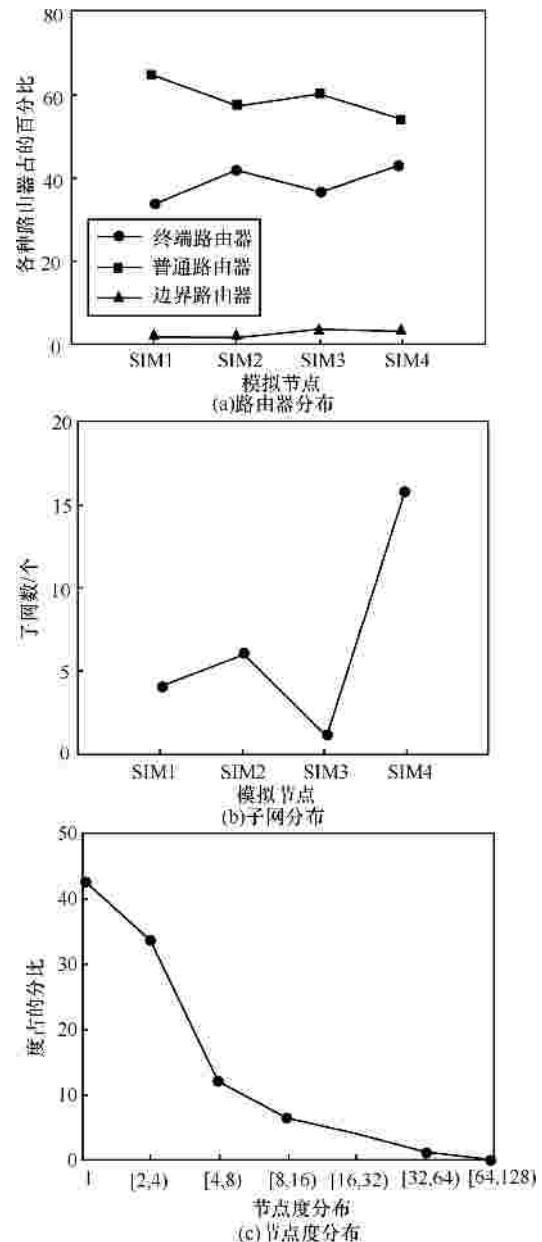


图 1 拓扑划分后信息

$$G = \bigcup_{i=1}^n G_i \tag{1}$$

$$G_i = (V_i, E_i, R_i) \tag{2}$$

$$V = \bigcup_{i=1}^n V_i \tag{3}$$

$$\bigcap_{i=1}^n V_i = \emptyset \tag{4}$$

$$E = \left(\bigcup_{i=1}^n E_i \right) \cup R \tag{5}$$

$$R = \bigcup_{i < j, i, j \in [1, n]} R_{ij} (n > 1) \tag{6}$$

将各子图之间的远程链路删除后，对于每个子图 G_i ，其可能包含 n 个子网，设 I_i 表示 SIM_{*i*} 中被错误划分的信息，主要指边界路由器中被错分的基本信息，如因为删除远程链路而导致某些边界路由器的度数发生变化等，对于这种错误划分的节点进行特殊存储，保证数据的正确性。 S_i 表示 SIM_{*i*} 中的子网信息。故每个子图在删除远程链路后，可用式 (7) 代替式 (2) 表示。

$$G_i = (V_i, E_i, I_i, S_i) \tag{7}$$

根据 Zone_Cut 路由策略的核心思想做如下定义。

定义 9 树型分支：从终端主机群开始遍历，通过不断迭代和链路剪枝，最终到某一个节点的度大于 1 停止迭代，而获取到的最大的树形区分支。

定义 10 代表节点：每一个树型分支都有一个根节点，称之为此树形分支的代表节点。

定义 11 T 区：所有 T_{ij} 构成的集合， $T_{ij} | v_i \in T_{ij}, v_i \in G_j | T_{ij} \in G_j$ ，表示 T_{ij} 为一个树型分支，其节点属于子图 G_j 并且根节点为该树型分支的代表节点。

定义 12 LD 区：所有 LD_{ij} 构成的集合， $LD_{ij} | v_i \in LD_{ij}, v_i \in G_j | LD_{ij} \in G_j | d(v_i) = 2$ 表示经树型抽象后，由 T_{ij} 的代表节点和剩余节点中度为 2 的节点组成的集合。

定义 13 HD 区：所有 HD_{ij} 构成的集合， $HD_{ij} | v_i \in HD_{ij}, v_i \in G_j | HD_{ij} \in G_j | d(v_i) \geq 3$ ，表示经树型抽象后，由 T_{ij} 的代表节点和剩余节点中度大于 2 的节点组成的集合。

以图 2 所示拓扑为例进行说明。如图 2(a) 所示网络拓扑，共有 38 个路由器节点，划分成 2 个拓扑区域，分别对 SIM1 和 SIM2 这 2 个模拟节点进行模拟。根据树型抽象，可在拓扑中建立 8 个 T 区，抽象后的拓扑如图 2(b) 所示。抽象后根据节点度对拓扑进行进一步划分，包括 3 个 LD 区和 1 个 HD 区。

针对 T 区，根据树形区节点在访问时的特点，即孩子的父亲是唯一的，孩子寻找父亲时间复杂度是 $O(1)$ ，而父亲到孩子则需要遍历孩子集合，此时时间复杂度取决于孩子个数。因此，可以使用寻找最早公共祖先的方式将父亲寻找孩子转化为孩子寻找父亲，然后一个倒序即可获得父亲要到达孩子需要经历的下一跳。存储时只存储孩子和父亲之间的关系，因此，只需要一个映射表可以实现存储。

针对 LD 区，根据图形稀疏区的特点和计算机中对于二进制的特殊处理方式，可以选择比特图用于进行稀疏区节点信息存储和查找。

针对 HD 区，借鉴平面路由中的策略，存储图形稠密区内任意节点对之间的距离和节点对的下一跳 ID。对于图形稠密区的详细邻接信息，选择使用邻接表进行存储。查找时使用 BFS 算法代替 Dijkstra 算法，在 $O(n+e)$ 内就可以获得当前节点到其他任意节点之间的最短路径。

根据 Zone_Cut 路由策略，对 T 区节点进行抽象，以其根节点作为代表节点。为便于区分，对拓扑中所有节点进行着色处理。设 Z 为区域划分集

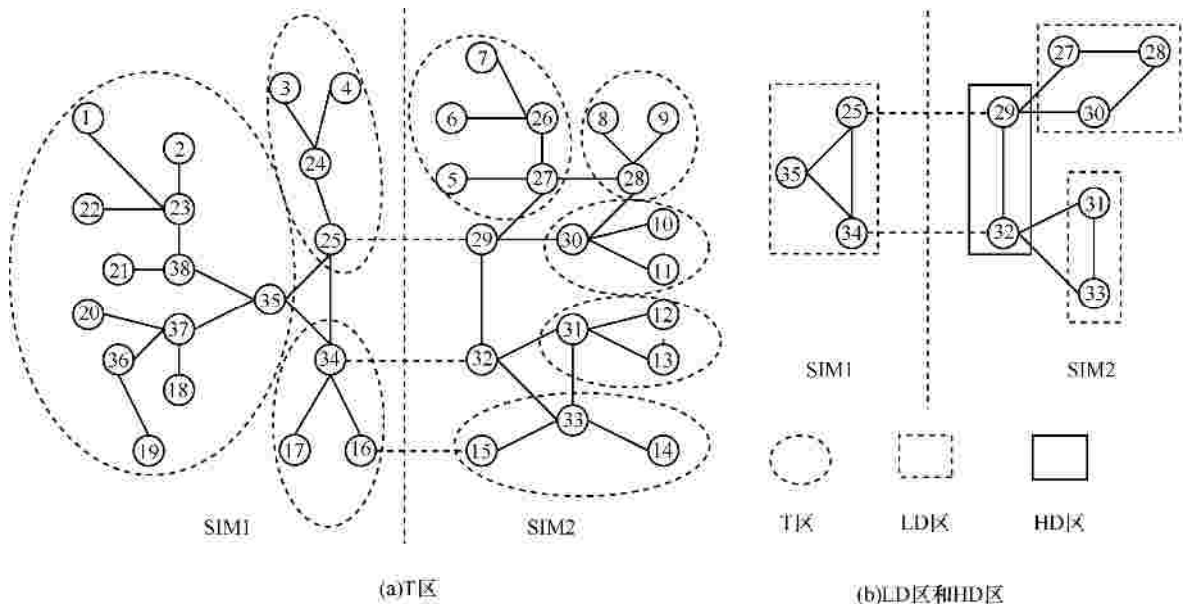


图 2 子域分割示意

合, U 为代表节点集合, Z_T, Z_{LD}, Z_{HD} 分别代表 T 区、LD 区和 HD 区, 则

$$G = (V, E, Z, U) \quad (8)$$

$$Z = (Z_T, Z_{LD}, Z_{HD}) \quad (9)$$

设 $|Z_{LD}| = n_1, |Z_{HD}| = n_2, |Z_T| = n_3, |U| = n_4$, 则 n_1 为 LD 区节点数, n_2 为 HD 区节点数, n_3 为 T 区节点数, n_4 为代表节点数。则节点总个数为

$$n = n_1 + n_2 + n_3 - n_4 \quad (10)$$

2.3 节点编码

根据 Zone_Cut 路由思想, 将大规模拓扑进行区域划分后, 其中大部分节点属于 T 区, 而 LD 区和 HD 区节点相对较少, 故对各区采用相应的编码方式。首先对 LD 区节点进行顺序编码, 然后对 HD 区节点进行顺序编码, 最后对树型区进行编码。

对 LD 区节点和 HD 区节点采用类似于平面地址的方式进行顺序编码, 然后扩充至 32 bit。

对于 T 区的节点, 编码分为 4 段, 首先采用 3 bit 表示节点类型, 如普通路由器节点、代表节点等; 其次采用 12 bit 表示一个树型区分支, 一个 SIM 内可有 4 096 个树型分支; 再次采用 4 bit 表示节点所在层数, 节点层数最多为 16; 最后采用 13 bit 表示节点的偏移量, 一个树型分支最多有 8 192 个节点。其中编码格式具体示例如图 3 所示。

代表节点偏移量	所在层数	树形区分支	节点类型判定码
32	19	15	3

图 3 树型区节点编码格式示例

根据上述编码表, 可以发现除了 LCA 算法的查询之外, 其他 T 区的查询全部都是在 $O(1)$ 的时间内。

2.4 树型抽象算法

树型抽象(TA, Tree_Abstract)的目的是降低网络拓扑规模, 同时定位 T 区代表节点。设 r_t, r_r, V_T, V_R 分别代表终端节点、T 代表节点, 终端节点集合和 T 区代表节点集合。树型抽象算法如下。

- 1) procedure Tree_Abstract(*graph*)
- 2) $n = |V|$;
- 3) for each $v_i \in V$ do
- 4) $v_i.number = 1$;
- 5) $v_i.property = null$;
- 6) $v_i.content = null$;
- 7) end for
- 8) for each $v_i \in V$ do

- 9) if $d(v_i) = 1$ then
- 10) $v_i.property = r_t$;
- 11) add v_i to V_T ;
- 12) end if
- 13) end for
- 14) $V'_T = V_T$
- 15) while $\exists v_i \in V'_T$ do
- 16) if $\exists e_{ij} \in E$ then
- 17) del v_i from V'_T, V ;
- 18) $v_j.number = v_j.number + 1$;
- 19) $v_j.property = r_r$;
- 20) $v_j.content = v_j.content + (v_i)$;
- 21) if $v_j \notin V_R$ then
- 22) add v_j to V_R
- 23) end if
- 24) if $d(v_j) = 1$ then
- 25) add v_j to V'_T ;
- 26) end if
- 27) end if
- 28) end while
- 29) end procedure

通过 TA 算法, 可以抽象拓扑中所有度为 1 的节点, 同时通过节点 v_i 的属性 number、property 和 content 分别确定节点 v_i 代表节点的个数、节点属性(终端节点或代表节点)和代表节点结构信息。该算法的最大时间和空间复杂度都为 $O(n) = O(|V|)$ 。

2.5 节点着色算法

经树型抽象后, 采用拓扑划分算法对拓扑进行划分, 划分后可能存在度为 1 的节点, 此节点为边界路由器节点, 重新调用树型抽象算法对各划分区域的各子网再次抽象。

为快速定位各节点所属区域, 从而选取不同的路由存储与查找策略, 在此使用节点着色算法, 根据节点属性的不同着以不同的颜色。终端路由器颜色标记为 BLACK1, 边界路由器节点标记为 BLACK2, 其他路由器标记为 BLACK3。根据节点所属区域的不同对所有节点重新着色, T 区的代表节点和其被抽象节点分别标记为 RED0 和 RED1, LD 区节点标记为 BLUE, HD 区节点标记为 GREEN。在着色过程中会存在如下情况。

- 1) 通常情况下, 边界路由器的度一定大于 1, 但由于区域划分可能导致某些边界路由器的度为

1, 从而归属 T 区, 并且被抽象, 此类节点着色为 BRED。如图 2 (a) 中的 15 和 16 号节点。

2) 某些边界路由器节点的度可能大于等于 3, 但由于区域划分, 导致其度在子网内变为大于等于 2。如果为 2, 此类节点应归属于 LD 区, 着色为 BBLUE; 如果为 3, 此类节点应归属于 HD 区, 着色为 BGREEN, 如图 2 (b) 中 34 和 32 号节点。

3) 拓扑划分可能导致某些子网的节点数非常少, 如小于等于 5, 甚至等于 1, 此时可通过优化拓扑划分算法, 根据负载均衡的原则, 通过节点移动, 避免此类情况的发生。故在着色过程中不用考虑孤立节点、孤立环等小规模孤立子网。

根据以上分析, 设 r_{ld} 、 r_{hd} 、 r_b 、 V_B 分别代表 LD 区路由器、HD 区路由器、边界路由器和边界路由器集合, 建立节点着色(NC, Node_Color)算法。

```

1) procedure Node_Color(graph)
2) for each  $v_i \in V$  do
3) if  $v_i \in V_T$  then
4)  $v_i.color = BLACK1$ ;
5) else if  $v_i \in V_B$  then
6)  $v_i.color = BLACK2$ ;
7) else
8)  $v_i.color = BLACK3$ ;
9) end if
10) end for
11) while  $i \leq TPN$  do //TPN is the number of
topology partition
12) while  $j \leq NN$  do //NN is the number of
subnet
13) Tree_Abstract(graph);
14) for each  $v_k \in V_{NN(j)}$  do
15) if  $v_k.property = r_r$  then
16)  $v_k.color = RED0$ ;
17) for each  $v_p \in v_k.content$  do
18) if  $v_p.color = BLACK2$  then
19)  $v_p.color = BRED$ ;
20) else
21)  $v_p.color = RED1$ ;
22) end if
23) end for
24) else if  $d(v_k) == 2$  then

```

```

25) if  $v_p.color = BLACK2$  then
26)  $v_p.color = BBLUE$ ;
27) else
28)  $v_p.color = BLUE$ ;
29) end if
30)  $n_{ld}++$ ;
31)  $v_k.property = r_{ld}$ ;
32) else
33) if  $v_p.color = BLACK2$  then
34)  $v_p.color = BGREEN$ ;
35) else
36)  $v_p.color = GREEN$ ;
37) end if
38)  $n_{hd}++$ ;
39)  $v_k.property = r_{hd}$ ;
40) end if
41) end for
42)  $j++$ ;
43) end while
44)  $i++$ ;
45) end while
46) end procedure

```

通过 NC 算法可对经树型收缩和拓扑划分后的各区域的各子网的所有节点进行着色, 从而通过节点颜色区分各节点所属区域。设共划分为 n 个区域, 每个区域所包含的子网数为 n_i , i 为区域编号, 设各区域所包含的最大子网数为 n_i ; 各子网的节点数为 n_{jk} , 其中 j 为区域编号, k 为子网编号, 设各子网所包含的最大节点数为 n_{jk} ; 各代表节点所包含的节点数为 n_{mnp} , 其中 m 为区域编号, n 为子网编号, p 为包含的节点数, 设各代表节点所包含的最大节点数为 n_{mnp} , 则算法的最大时间和空间复杂度为 $O(n \cdot n_i \cdot n_{jk} \cdot n_{mnp})$ 。

3 实验及结果分析

路由策略的性能主要是从模拟的时间和空间上进行综合分析。实验使用启明星辰 2009 年探测的拓扑, 拓扑数据如表 1 所示。

实验采用单机模拟, 与效率较高的 MTree_Nix 路由策略比较。实验环境相同, 且在相同拓扑下分

表 1 拓扑结构

分组编号	拓扑来源	总路由器数	终端路由器数	节点总数	链路总数	主机总数	路由器平均度
1	国家骨干	130	32	1 154	1 253	1 024	3.52
2	云南	99	46	1 571	1 582	1 472	2.22
3	重庆	180	71	2 452	2 563	2 272	3.23
4	黑龙江	162	82	2 786	2 824	2 624	2.47
5	江苏	250	121	4 122	4 293	3 872	3.37
6	浙江	318	139	4 766	4 976	4 448	3.32
7	广东	334	149	5 103	5 268	4 768	2.99

别对低频分组(1个/秒)和分组(1000个/秒)情况进行实验对比。

3.1 多点低频分组对比

为验证路由的正确性,需要测试整个拓扑中任意节点对之间的路由,所有节点之间以1个/秒的速度互相发送数据分组,实验结果如图4所示。

从图4(a)和图4(b)中可以看出相同情况下,模拟时间 Zone_Cut 路由策略低于 MTree_Nix 路由策略,模拟时间平均减少 18.08%。从图4(c)中可以看出,Zone_Cut 的模拟时间增长率低于 MTree_Nix 模拟时间增长率,这是因为 Zone_Cut 路由策略根据节点属性不同,采用不同的查找策略,但2种路由策略的增长速度都远高于规模增长的速度。

从图5(a)和图5(b)中可以看出,Zone_Cut 路由策略相对于 MTree_Nix 路由策略模拟空间有了很大的降低,其减少比例基本呈线性增长,平均减少 51.23%,随规模的增长减少的空间更多。从图5(c)中可以看出,MTree_Nix 模拟空间增长速度高于模拟规模增长速度,而 Zone_Cut 路由策略模拟空间增长速度低于模拟规模的增长速度,并且随规模的增长之间的差距越来越大。这是因为 Zone_Cut 路由策略根据节点属性不同采用了不同存储策略。

3.2 单点高频分组对比

为测试路由策略的性能,随机选取拓扑中的源节点,向其他节点以1000个/秒的速度发送数据分组,实验结果如下。

从图6(a)和图6(b)中可以看出,采用高频发送分组 Zone_Cut 路由策略相对于 MTree_Nix 路由策略时间减少明显,平均减少 55.29%。从图6(c)中可以看出,MTree_Nix 路由策略模拟时间增长速度与规模增长速度有交点,但整体趋势是高于规模增长速度,并且增长速度较快。Zone_Cut 路由策略模拟时间增长速度低于模拟规模增长速度,并

且增长率也较低。原因同图4分析。

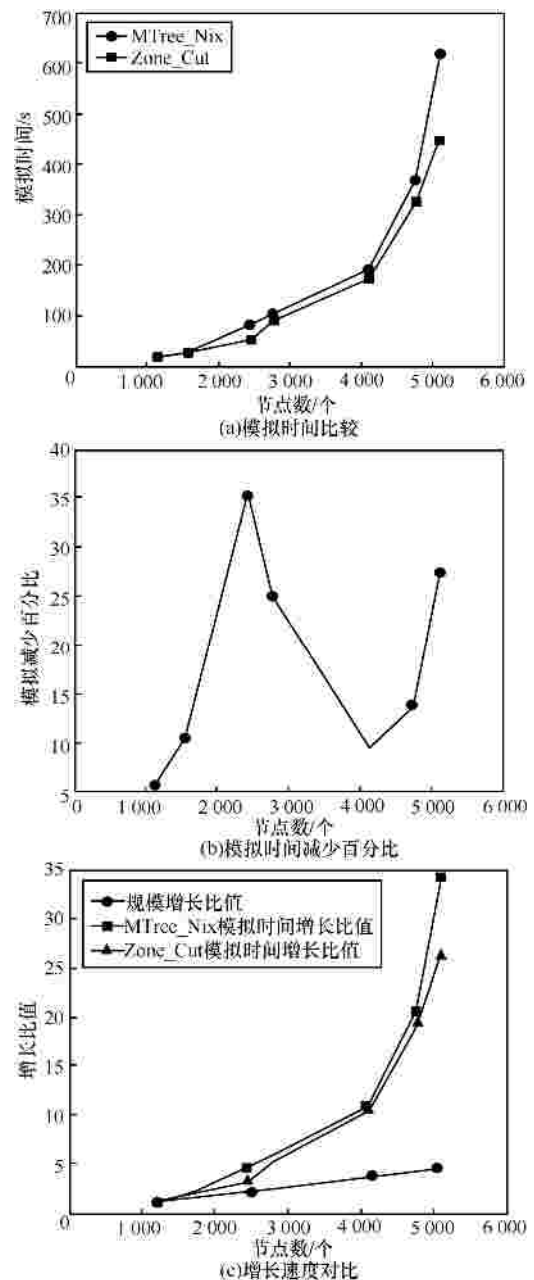
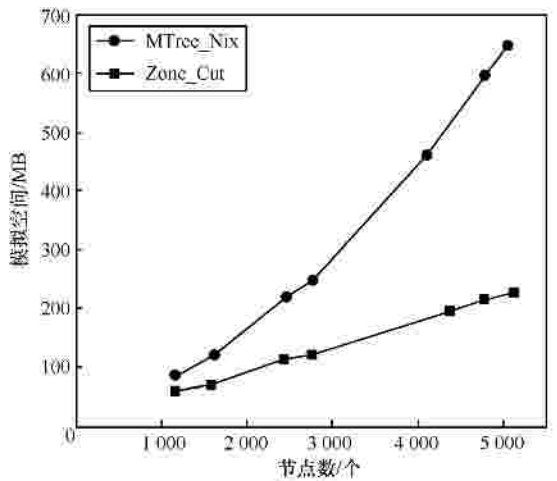
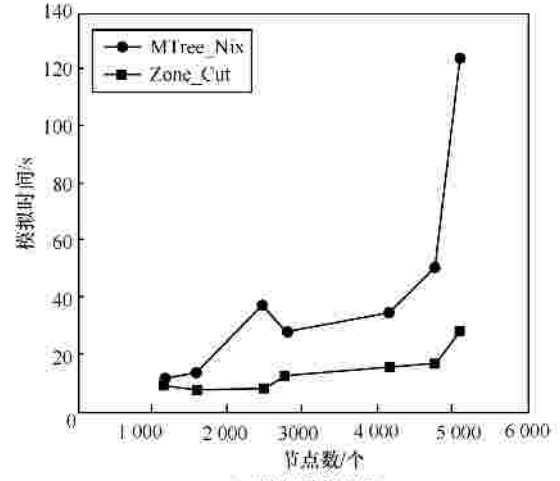


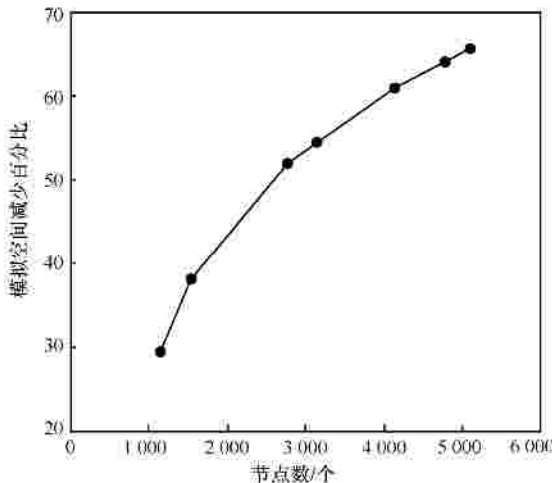
图 4 多点低频发送分组模拟时间实验对比



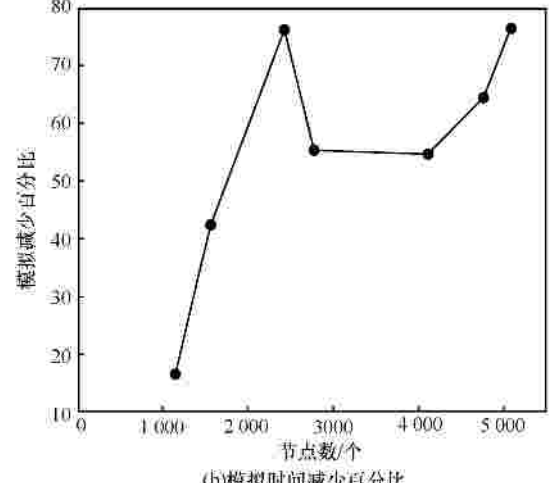
(a)模拟空间比较



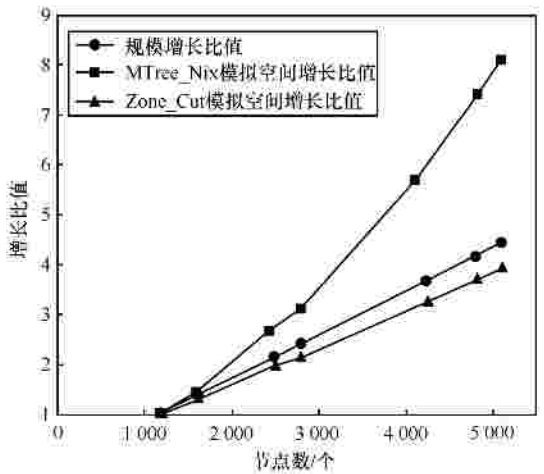
(a)模拟时间比较



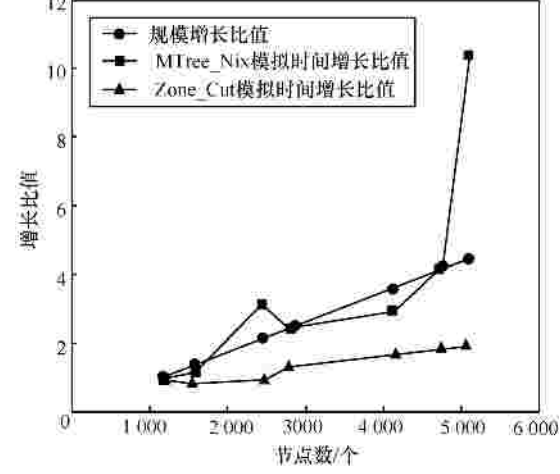
(b)模拟空间减少百分比



(b)模拟时间减少百分比



(c)增长速度对比



(c)增长速度对比

图 5 多点低频发送分组模拟空间实验对比

图 6 单点高频发送分组模拟时间实验对比

从图 7 (a) 和图 7 (b) 中可以看出, Zone_Cut 路由策略相对于 MTree_Nix 路由策略模拟空间有了很大的降低, 其减少比例基本呈线性增长, 平均减少 74.4%, 这说明随规模的增长, Zone_Cut 路由策略比 MTree_Nix 路由策略减少空间更多。

从图 7 (c) 中可以看出, MTree_Nix 路由策略和 Zone_Cut 路由策略模拟空间增长速度都低于模拟规模增长速度, 并且 Zone_Cut 路由策略模拟空间增长速度更低, 并且随规模的增长, 它们之间的差距越来越大。原因同图 5 分析。

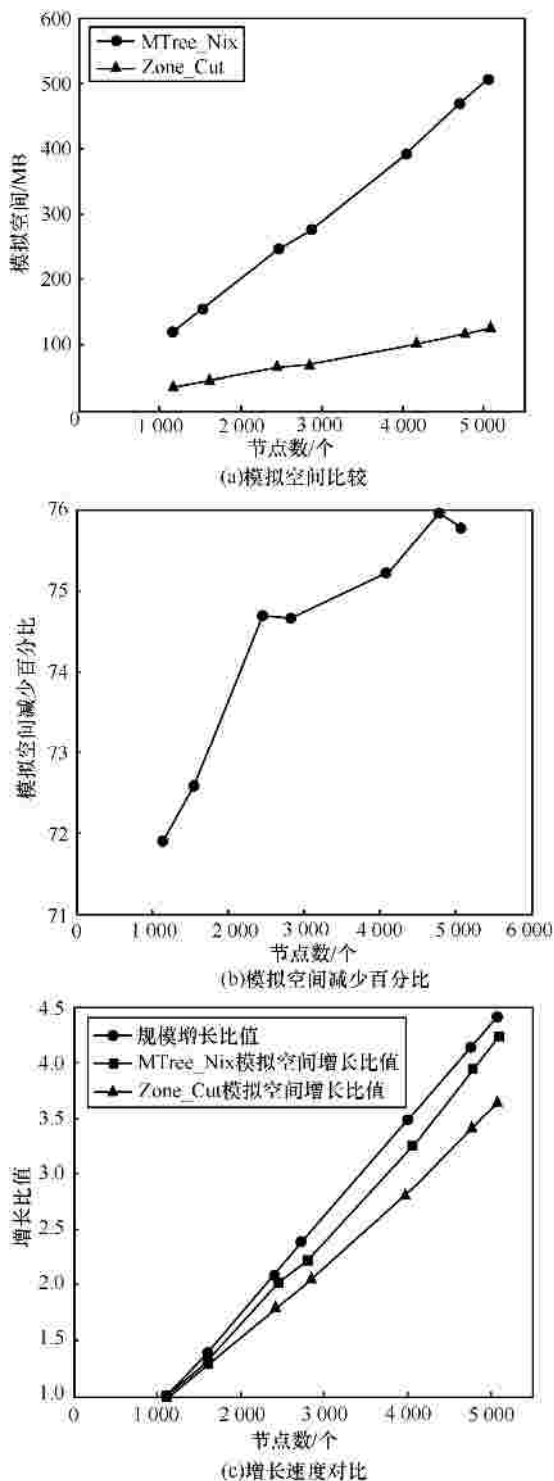


图 7 单点高频发送分组模拟空间实验对比

以上实验说明 Zone_Cut 路由策略无论在模拟时间和模拟空间上都要优于 MTree_Nix 路由策略，从文献[15]的实验结果中可以看出，MTree_Nix 路由策略的模拟时间和模拟空间无论从规模角度还是平衡性角度都要优于 Flat、STree-Flat、MTree-Flat 和 Nix-Vector。

4 结束语

本文分析了目前网络模拟主要使用的本地路由策略存在的缺陷及形成的原因，通过分析提出 Zone_Cut 本地静态路由策略。Zone_Cut 路由策略根据节点属性的不同将节点分为 T 区、LD 区和 HD 区，对不同的区域采用不同的存储和查找策略，降低了路由存储空间和查找时间，更好地平衡了路由策略的空间和时间，有效地提高了模拟运行的效率。实验结果表明，Zone_Cut 路由策略比 MTree_Nix 路由策略综合性能有了大幅提高。低频分组情况下，模拟时间平均减少 18.08%，模拟空间平均减少 51.23%；高频分组情况下，模拟时间平均减少 55.29%，模拟空间平均减少 74.4%，提高了模拟的效率，更适合大规模网络环境下复杂行为的模拟。该研究成果已经运用在“863”课题“高性能大规模网络行为模拟系统”中。

参考文献：

- [1] 李越, 钱德沛. 基于 NS 的分布式并行网络模拟器[J]. 电子学报, 2004, 32 (2) :246 - 249.
LI Y, QIAN D P, A distributed parallel network simulator based on NS[J]. Acta Electronica Sinica, 2004, 32 (2) :246 - 249.
- [2] XIONG Z G, ZHANG X M, XIA X W, et al. Analysis of discrete event generation model on result of network simulation[A]. Proc of the 2nd International Conference on Software Technology and Engineering[C]. San Juan, PR, 2010, (2):2119-2122.
- [3] SCHMIDT S, BYE R, CHINNOW J, et al. Application-level simulation for network security[J]. Simulation, 2010,86(5-6):311-330.
- [4] SHEN J J , GAN Z C, YU X H. The research on tactical Internet simulative training and its dynamic real-time network simulation[A]. Proc of the 2nd IEEE International Conference on Information Management and Engineering[C]. Chengdu, China, 2010.354-358.
- [5] HUANG P, HEIDEMANN J. Minimizing routing state for lightweight network simulation[A]. Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems[C]. Cincinnati, OH, United states, 2001. 108 -116
- [6] 宋丽静. 网络模拟中本地路由策略的研究与优化[D]. 哈尔滨: 哈尔滨工业大学,2010.
SONG L J. Research and Optimization of Local Routing Mechanisms in Network Simulation[D]. Harbin: Harbin Institute of Technology, 2010.
- [7] 王轩春. 基于PDNS的本地静态路由算法的优化[D]. 哈尔滨: 哈尔滨工业的学,2010.
WANG X C. Optimization of the Local Static Routing Algorithm on PDNS[D]. Harbin: Harbin Institute of Technology, 2010.
- [8] 崔宇, 张兆心, 张宏莉等. 并行网络模拟中远程路由策略的研究[J]. 通信学报, 2012, 33(5):115-123.
CUI Y, ZHANG Z X, ZHANG H L, et al. Study of remote routing strategy in parallel network simulation[J]. Journal on Communications, 2012, 33(5):115-123.
- [9] 崔宇. 大规模并行网络模拟中高性能远程路由策略的研究[D]. 哈尔滨: 哈尔滨工业大学, 2010.
CUI Y. Research on High Performance Remote Routing Strategy i

Large-scale and Parallel Network Simulation[D]. Harbin: Harbin Institute of Technology, 2010.

- [10] HUANG P. Enabling Large-scale Network Simulations: a Selective Abstraction Approach[D]. Los Angeles, University of Southern California, 1999.
- [11] HIROMORI A, YAMAGUCHI H, YASUMOTO K, *et al.* Reducing the size of routing tables for large-scale network simulation[A]. Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems[C]. IEEE Computer Society, Washington, DC, USA, 2003. 115-122.
- [12] CHEN J, GUPTA D, VISHWANATH K, *et al.* Routing in an Internet-scale network emulator[A]. Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems[C]. Volendam, Netherlands, 2004.275-283.
- [13] RILEY G, FUJIMOTO R, AMMAR M. 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, CA, USA, 2000.524-531.
- [14] HAO Z Y, YUN X C, ZHANG H L. An efficient routing mechanism in network simulation[A]. Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation[C]. Singapore, 2008, 84(10-11):511-520.
- [15] 郝志宇, 云晓春, 张宏莉. MTree_Nix 网络模拟路由计算与查找策略[J]. 电子学报, 2008, 36(3):477-481.
HAO Z Y, YUN X C, ZHANG H L. Mtree_Nix routing computation and lookup mechanism in network simulation[J]. Acta Electronica Sinica, 2008, 36(3):477-481.



杜跃进 (1972-), 男, 山东招远人, 博士, 国家计算机网络应急技术处理协调中心副总工程师, 哈尔滨工业大学兼职教授, 主要研究方向为网络与信息安全、网络模拟等。



王克 (1948-), 男, 江苏阜宁人, 博士, 哈尔滨工业大学教授、博士生导师, 主要研究方向为应用数学、网络与信息安全、信息网络模拟等。



王轩春 (1986-), 男, 山东梁山人, 哈尔滨工业大学硕士生, 主要研究方向为网络模拟。

作者简介:



张兆心 (1979-), 男, 满族, 黑龙江哈尔滨人, 博士, 哈尔滨工业大学副教授, 中国科学院计算技术研究所博士后, 主要研究方向为网络与信息安全、网络模拟、域名体系安全等。



郝志宇 (1980-), 男, 黑龙江哈尔滨人, 博士, 中国科学院计算技术研究所副研究员, 主要研究方向为网络与信息安全、网络模拟等。