

## 基于焦点折叠的网络模拟拓扑抽象模型

张兆心<sup>1,2</sup>, 杜跃进<sup>1,3</sup>, 王克<sup>1</sup>, 丁振全<sup>1</sup>, 郝志宇<sup>2</sup>

(1. 哈尔滨工业大学 计算机科学与技术学院, 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:** 针对大规模网络模拟的高资源消耗问题提出基于焦点折叠的网络模拟拓扑抽象模型, 采用终端节点抽象算法、树型收缩算法和切割边抽象算法, 根据抽象系数对拓扑进行抽象。实验结果表明, 该技术可减少路由器节点数约 30%, 减少路由器间链路数约 13.74%, 总节点和总链路数分别减少约 98.48%和 96.1%; 在 250 万节点规模的拓扑上进行 DDoS 攻击模拟仅需 400s 的时间和 2 710MB 的内存; 以山东省拓扑为例, 进行 DDoS 攻击实验, 减少内存约 75.34%, 降低模拟时间约 91.76%; 以北京市拓扑为例, 进行蠕虫传播实验, 减少内存 68.84%, 减少模拟时间 38.64%。可见, 该模型可提高模拟的规模和效率, 降低模拟的资源开销和模拟运行时间。

**关键词:** 网络模拟; 焦点折叠; 终端节点抽象; 树型收缩; 切割边抽象

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2012)07-0009-13

## Topology aggregation model based on focus folding for network simulation

ZHANG Zhao-xin<sup>1,2</sup>, DU Yue-jin<sup>1,3</sup>, WANG Ke<sup>1</sup>, DING Zhen-quan<sup>1</sup>, HAO Zhi-yu<sup>2</sup>

(1. School of Computer Science and Technology, Harbin Instituted of Technology, Harbin 150001, China;

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

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

**Abstract:** High resource consumption is the main problem of large-scale network simulation, to solve this problem, the topology aggregation model based on focus folding for network simulation was put forward. According to the abstract coefficients, the terminal node abstract, tree shrinking, and cut-edge abstract algorithm was adopt to abstract the topology. The test results show that, this technology can reduce the router node about 30 percent, the link between routers about 13.74 percent, total node about 98.48 percent, and total link about 96.1 percent. Simulating DDoS on the topology with 2.5 million nodes only need 400s time and 2 710MB memory. Take Shandong topology as an example to simulate DDoS, after abstracted, simulation memory and time have been reduced about 75.34 percent and 91.76 percent respectively. Take Beijing topology as an example to simulate worm, after abstracted, simulation memory and time have been reduced about 68.84 percent and 38.64 percent respectively, which prove that this mode can improve the simulation scale and efficiency and decrease the resource consumption and running time of simulation.

**Key words:** network simulation; focus folding; terminal node abstract; tree shrinking; cut-edge abstract

收稿日期: 2011-06-28; 修回日期: 2012-03-20

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); 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); 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 Fundation (HIT NSRIF 2011119)

## 1 引言

随着全球信息化进程的不断推进, Internet 规模越来越大, 应用越来越广泛, 异构性和动态性越来越强, 网络结构及网络行为越来越复杂。为进一步促进互联网的发展, 提高互联网的安全性, 越来越有必要对互联网及其行为进行研究。目前主要有 6 种研究 Internet 的方法<sup>[1-3]</sup>, 包括数学模型分析、网络测量、网络模拟、网络仿真、原形实验平台、实际网络测试与实验。在众多研究方法中, 网络模拟不但可对其他研究方法的结果进行检验, 而且具有灵活多变、规模大、详细刻画网络行为等优点, 逐渐成为网络及其行为研究的主要方法。国内外已经有很多机构长期从事网络模拟方面的研究, 并开发了很多著名的模拟工具, 其中典型的单机模拟器包括 DARPA 资助研发的 NS2<sup>[4]</sup>、OPNET Technologies Ins 公司的 OPNET<sup>[5]</sup>、DARPA 资助研发的 SSFNet<sup>[6]</sup>和达特茅斯学院开发的 DaSSF<sup>[7]</sup>。其中, NS2 及其并行版 PDNS<sup>[8]</sup>因其完全免费、开源、带有大量协议库支持等特点得到众多学者的青睐, 已成为目前普遍使用及公认的模拟工具, 也是本文的研究基础。

Internet 与生俱来的复杂性、异构性、动态性、及庞大的规模都给网络模拟研究工作带来巨大挑战。简化网络模拟模型, 提高模型的抽象度, 可以降低大规模网络模拟所需要的计算开销和存储开销, 从而减少网络模拟的运行时间, 增大模拟的规模, 提高模拟的效率<sup>[9]</sup>。网络模拟模型抽象技术主要包括拓扑模型抽象、流量模型抽象以及协议模型抽象。在各类模型抽象中, 拓扑模型抽象最为关键。这是因为合理的高度抽象不仅可以极大地缩减网络拓扑的规模, 并且可以极大降低模拟所产生的离散事件数, 也就意味着可观的降低了模拟的计算和存储开销。近年来多位学者对拓扑进行了研究, 文献[10]通过对 2001 年 12 月至 2006 年 12 月互联网自治系统拓扑中节点数目、边的数目的分析, 验证了互联网自治系统级拓扑具有小世界网络的特征<sup>[11]</sup>。文献[12]主要对互联网拓扑结构知识发现及其应用方法进行研究, 总结了互联网拓扑结构特征, 为路由器级拓扑抽象提供一定的参考。文献[13]揭示互联网拓扑结构节点度分布符合幂律, 表现为网络中存在大量的低度节点(拥有少量连接的节点)和少量的高度节点(拥有大量连接的节点)。这些拓扑的属性从多个角度反映了拓扑的特性, 并可用于评价拓扑

的变化。在拓扑抽象及应用与网络模拟的拓扑抽象方面, 一些学者也进行了较深入的研究。自相似网络分形维度的重正化方法<sup>[14]</sup>将网络拓扑采用不同长度的盒子覆盖整个网络, 并保证盒子数最少, 然后将每个盒子抽象为一个点。以该思想为出发点作者提出了一种基于 local-area 的 Internet 路由级拓扑抽象算法<sup>[15]</sup>, 每个 local-area 可看作一个大小不同的盒子, 这样就可以在不改变路径的情况下, 将每个盒子覆盖的节点抽象为一个节点, 从而对网络拓扑进行抽象。文献[16]在 NS2 上实现 TranSim 技术, 即只维持网络内一些固定参数, 如延迟一带宽、TCP 窗口、链路分组丢失率等, 从而将大规模网络拓扑抽象成小规模网络。文献[17]采用减少客户端数量, 将其行为叠加到其他客户端的抽象规则。以德国教育网为拓扑数据进行实验, 实验结果表明, 在保证端到端延迟、平均流量负载、分组丢失率大致不变的情况下, 抽象后拓扑规模降低为原来的 1/8, 模拟时间减少 33%。文献[18]通过删减部分所要模拟网络拓扑中的主机节点来降低模拟规模, 以降低模拟的存储及计算开销。文献[19]和文献[20]采用删除、压缩、遍历等方法, 将大规模网络拓扑的规模降低, 以便使大规模网络模拟的速度变快。文献[21]和文献[22]提出采用离散事件模拟与蠕虫传播数学模型相结合的方法来进行蠕虫传播的模拟。

由上述分析可知, 目前应用于网络模拟的拓扑抽象技术仅通过删减部分主机节点或结合简单数学模型的方法降低拓扑的规模, 拓扑抽象率低, 真实性无法保证, 无法满足百万节点的大规模网络模拟。为此, 本文提出基于焦点折叠的网络模拟拓扑抽象模型。首先将待模拟网络分为焦点区域和非焦点区域; 然后对非焦点区域, 根据抽象系数, 利用相应算法对网络拓扑进行抽象; 最后在抽象后的拓扑上进行事件模拟。

## 2 网络模拟资源消耗分析

网络模拟中, 系统的资源消耗分为静态消耗与动态消耗 2 部分。

### 1) 静态消耗

静态消耗在系统模拟的初始化阶段, 主要用于建立网络模拟中的各种结构和路由表。建立模拟结构所耗的资源取决于拓扑中各种节点的数量, 而路由表所耗的资源一般取决于拓扑的大小及复杂程度。由于节点、链路、路由表等是整个网络模拟的

基础，在模拟开始前需要申请，并直到模拟结束后才能释放，因此资源的固有消耗往往占系统资源消耗的绝大部分。

2) 动态消耗

Internet 上的网络流量由一个个数据分组组成。对于离散事件模拟器来说，为了能够准确地描述网络行为，需要对每一个数据分组的每一次行为（如数据分组在连接上的入队和出队、数据分组被路由器和终端节点的处理等）都作为一个事件来进行模拟，这就是所谓分组级别（packet-level）的模拟。目前包括 NS2 在内的很多网络模拟器都采用的是基于分组级别的模拟。虽然分组级别的模拟能够保证较高的真实性，但是由于每一个数据分组的每一次行为都需要由单独的事件来描述，这对于网络模拟器的计算开销来说是极大的挑战。系统的计算开销也与网络拓扑的规模及结构息息相关，数据分组传送路径上包含的节点越多，包括出队、入队、延迟计算等每个数据分组收发过程中所经历的环节也就越多，这种资源的动态消耗随模拟进行，当数据分组传输过程结束后，该数据分组所占的资源被释放，离散事件完成后，也会释放相应的资源。

由上述分析可知，网络拓扑的规模和结构与网络模拟的资源消耗的程度基本成正比。故有效的网络拓扑抽象，可减少模拟的资源消耗，从而提高模拟的规模，减少模拟的运行时间，提高模拟的效率。

3 网络模拟拓扑抽象模型

待模拟网络拓扑可分为 2 种：焦点区域，即对该区域的各种细节需要重点关注，如数据分组具体传送过程等；非焦点区域，即对该区域仅关注一些粗线条数据，如蠕虫感染数量随时间的变化情况，因此可对非焦点区域拓扑进行压缩。基于焦点折叠的网络模拟拓扑抽象模型如图 1 所示。

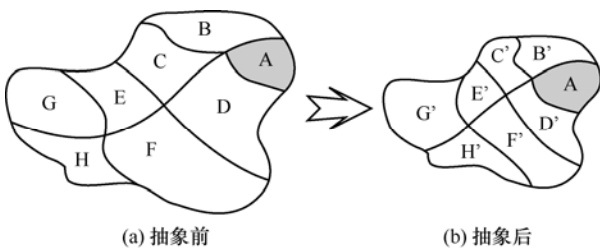


图 1 基于焦点折叠的网络模拟拓扑抽象模型

设 A 区域为焦点区域，B~H 区域为非焦点区域，通过抽象算法，将 B~H 区域抽象为 B'~H' 区

域，降低网络拓扑规模。本文不区分节点的入度与出度，因此采用无向图描述，节点度等于该节点的边数。在抽象过程中，未被抽象的节点及链路保持原有参数，如节点度、带宽、延迟等。将整个网络拓扑用无向图  $G(V, E)$  表示，每个路由器节点用顶点  $V_i$  表示，路由器之间的链路用  $E_{ij}$  表示，从而将网络拓扑的抽象转化为无向图的抽象。

目前，计算最短路径的算法主要包括 Floyd、Dijkstra、Bellman-Ford 和 SPFA 等。Floyd 算法为求多源最短路径算法，用矩阵记录图，时效性较差，时间和空间复杂度为  $O(|V|^3)$ ；Flody-Warshall 算法是解决任意 2 点间最短路径的一种算法，其时间复杂度为  $O(|V|^3)$ ，通过在原有空间上进行迭代，可将其空间复杂度降为  $O(|V|^2)$ ；Dijkstra 算法时间和空间复杂度为  $O(|V|^3)$ 。Bellman-Ford 算法为求单源最短路径，时效性较好，时间复杂度为  $O(|V| \cdot |E|)$ 。SPFA 是 Bellman-Ford 的队列优化，对于不同图所需要的时间差别很大，最好情形下，每个节点只入队一次，则算法变为广度优先搜索，其时间复杂度仅为  $O(|E|)$ ，最坏情况下，每个节点都被入队  $|V|-1$  次，则算法退化为 Bellman-Ford 算法，其时间复杂度为  $O(|V| \cdot |E|)$ 。在此选取 Dijkstra 算法计算最短路径，为便于描述，首先建立几个定义。

定义 1 节点度：节点  $v_i$  的度定义为  $d(v_i) = \sum_{e_{ij} \in E} e_{ij}$ 。

定义 2 终端节点集合：终端节点集合的定义为  $spV(v_i) | v_i \in V, d(v_i) = 1$ ，终端节点实际为叶节点，即主机节点，故其节点度为 1。

定义 3 最短路径边集： $\forall v_i, v_j \in V$  的最短路径边集定义为  $spE(v_i, v_j)$ ，表示节点  $v_i$  到节点  $v_j$  最短路径上所包含的边。

定义 4 最短路径点集： $\forall v_i, v_j \in V$  的最短路径点集定义为  $spN(v_i, v_j)$ ，表示节点  $v_i$  到节点  $v_j$  最短路径上所包含的节点。

定义 5 节点权值：节点  $v_i$  的权值定义为  $W(v_i) = \sum_{v_m, v_n \in spV} (v_i \in spN(v_m, v_n))$ ，表示节点  $v_i$  在所有终端节点间最短路径经历的次数。

定义 6 边权值：边  $e_{ij}$  的权值定义为

$W(e_{ij}) = \sum_{v_m, v_n \in spV} (e_{ij} \in spE(v_m, v_n))$ , 表示边  $e_{ij}$  在所有终端节点间最短路径经历的次数。

**定义 7** 抽象系数: 抽象系数定义为  $\lambda = \varphi(T(W(v)), T(W(e)))$ , 其中,  $T(W(v))$  表示节点的抽象权值阈值, 即权值小于该阈值的节点可被抽象,  $T(W(e))$  表示边的抽象权值阈值, 即权值小于该阈值的边可被抽象。抽象系数  $\lambda$  为  $T(W(v))$  和  $T(W(e))$  经  $\varphi$  作用得出的值,  $\varphi$  可根据实际需求定义。

### 3.1 拓扑抽象依据

Internet 拓扑具有层次性、聚集性、富人俱乐部<sup>[22]</sup>等属性, 而这些描述性属性无法直接作为拓扑抽象的依据, 基于焦点折叠的网络模拟拓扑抽象模型是以 Internet 的层次性和富人俱乐部属性为基础提出的。

Internet 可被看作是由大量互联的 AS 系统组成, 其中, 每个 AS 系统可被看作 Stub 域或 Transmit 域, Stub 域承载那些起源或终止于域内的通信量, 而 Transmit 域无此限制。Transmit 域的目的是有效互联 Stub 域。Stub 域通常由校园网或其他一些互联 LAN 组成, 而 Transmit 域则由 WAN 或 MAN 组成, 是典型的国家层次的 ISP 或骨干网。AS 层面的 Internet 层面结构如图 2 所示。

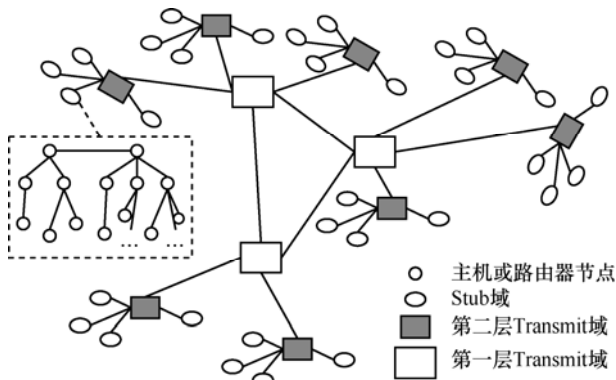


图 2 AS 层面 Internet 结构

图 2 中的虚线框中的部分是一个 Stub 域的展示示意, Internet 的层次性说明, 在 Internet 拓扑的最外层拥有大量的 Stub 域, 这些域占据了 Internet 节点的大部分, 这些域中或域间部分节点呈树状拓扑结构。

Internet 中少量的节点具有大量的边, 并且这些节点倾向于彼此之间的相互连接, 本文将这些节点称为“富节点”(rich nodes), 由这些节点构成的拓扑具有“富人俱乐部”(rich-club) 特点。可用富人俱乐部的连通性  $\varphi(r/N)$  来刻画这种现象, 它表示

网络中前  $r$  个度最大的节点之间, 实际存在的边数  $L$  与这  $r$  个节点之间可能存在的总边数  $r(r-1)/2$  的比值, 即

$$\varphi(r/N) = \frac{L}{r(r-1)/2} = \frac{2L}{r(r-1)} \quad (1)$$

如果  $\varphi(r/N)=1$ , 那么前  $r$  个最富的节点组成的富人俱乐部为一个完全连通子图。因此 Internet 的富人俱乐部原理说明, 在 Internet 中存在很多互联的核心节点, 如 AS 层面 Internet 结构示意图中的第一层 Transmit 域。但 Transmit 域和 Stub 域中节点的拓扑并非全部是树状结构, 如果要压缩这部分拓扑, 一个理想的方法为切割边, 但切割边必然会改变原有拓扑的一些属性, 如度分布、平均度等, 因此在切割边时势必要寻找那些对拓扑属性改变最小的边进行切割, 如拓扑边缘的那些边, 进行开环操作, 从而产生度为 1 的新节点。

### 3.2 拓扑抽象算法描述

拓扑抽象的目的是为了简化网络拓扑, 本文提出基于焦点折叠的拓扑抽象算法 (TABFF, topology aggregation based on focus folding)。该算法将拓扑视为无向图, 在给定的抽象系数下, 通过终端节点抽象算法、树型收缩算法和切割边抽象算法简化拓扑, 降低拓扑规模。本文是对拓扑进行抽象, 与模拟的应用无关, 更加体现出抽象算法的通用性。

原始拓扑示意如图 3 所示, 虚线左侧为焦点区域, 右侧为非焦点区域, 如无特殊说明,  $\circ$  代表主机,  $\blacksquare$  代表 Stub 域路由器,  $\boxtimes$  代表第二层 Transmit

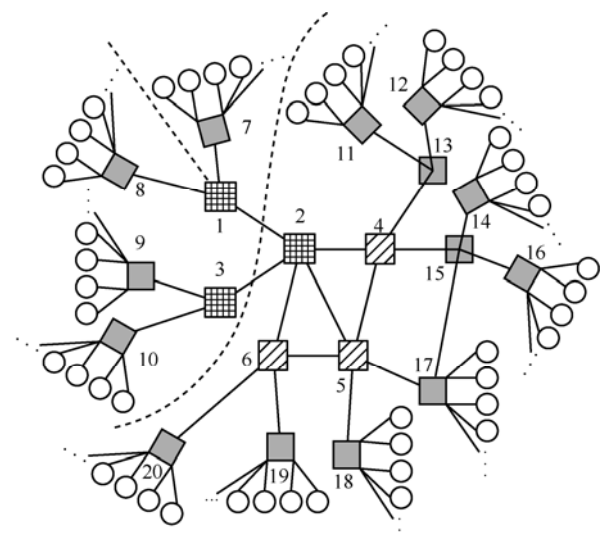



图 3 原始拓扑

域路由器，代表第一层 Transmit 域路由器，算法描述如下。

算法第 1 步：终端节点抽象，对如图 3 所示拓扑进行终端节点抽象后的拓扑如图 4 所示。

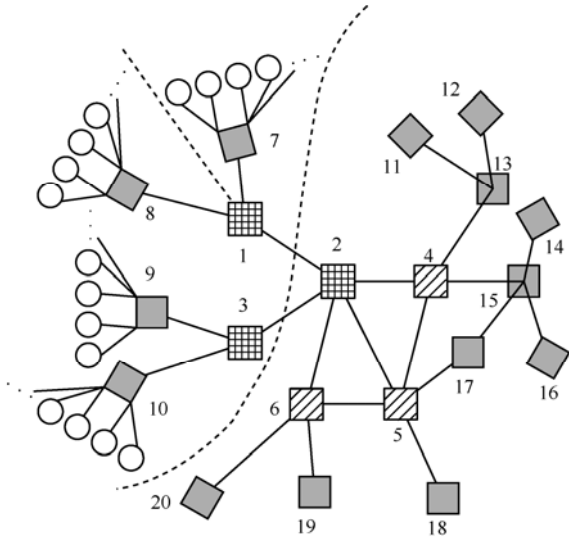


图 4 终端节点抽象后拓扑

首先将拓扑采用邻接表存储，获取所有节点度及终端节点集合；然后采用 Dijkstra 算法计算任意两终端节点间最短路径，并存储最短路径信息，进而获取所有节点权值和边权值，并存储；最后根据焦点区域条件，将所有非焦点区域的主机节点进行抽象，并将其归并到其上层路由器，输出抽象后拓扑。算法描述具体如下：

```

1) procedure Reduce_host_node
2)   LinkTable(node_num, spV, topo_data);
3)   for each  $v_i, v_j \in spV$  且  $i \neq j$  do
4)      $epE(v_i, v_j), spN(v_i, v_j) = BFS(v_i, V - v_i)$ ;
5)   end for
6)   for each  $v_i \in V$  do
7)      $W(v_i) = Weight\_Node(spN)$ ;
8)   end for
9)   for each  $e_{ij} \in E$  do
10)     $W(e_{ij}) = Weight\_Edge(spE)$ ;
11)  end for
12)  for each  $v_i \in spV$  do
13)    if  $v_i \in focus\_region$  then
14)       $del(v_i, spV, spV, V)$ ;
15)    if  $\exists e_{ij} \in E$  then
16)       $add(v_i, v_j)$ ;
17)       $del(e_{ij}, E)$ ;

```

```

18)    end if
19)  end if
20) end for
21) end procedure

```

在算法的实际应用过程中，可对网络拓扑先进行归并简化，这样在计算所有终端节点间最短路径时，可将所有主机节点归结到其相连的路由器上，计算所有终端路由器间的最短路径，从而可以极大地降低算法的空间复杂度和时间复杂度。设整个网络拓扑有  $n$  个路由器和  $n_t$  个终端路由器，每个终端路由器下有  $m$  个主机，焦点区域有  $n_f$  个路由器和  $n_{tf}$  个终端路由器，每个主机只与一个路由器相连，则经过算法第 1 步，抽象率为

$$f = \frac{(n_t - n_{tf})m}{n - n_f + (n_t - n_{tf})m} \quad (2)$$

设  $m = 50$ ，则经过算法第 1 步，如图 4 所示非焦点区域抽象率为 96.88%。

算法第 2 步：树型收缩，对如图 4 所示拓扑进行树型收缩后的拓扑如图 5 所示。

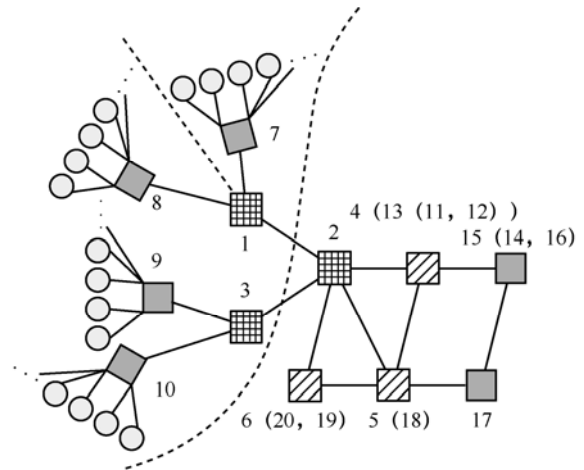


图 5 树型收缩后的拓扑

如图 5 所示，根据 Internet 的层次性，将图 4 中所有度为 1 的路由器抽象到与其相连的上层路由器，如 11 号和 12 号节点抽象到 13 号节点。首先根据抽象系数定义节点抽象权值和边抽象权值；然后寻找终端节点抽象后网络拓扑中度为 1 的路由器节点，将其抽象到与其相连的上层路由器；接着判断抽象系数，如果不满足，则进行递归抽象，直到满足抽象系数或拓扑中不再具有度为 1 的路由器节点，并将抽象节点的代表节点归入  $spV$  中；最后输出抽象后的拓扑。算法具体描述如下：

```

1) procedure Tree_abstract
2)  $T(W(v)) = p, T(W(e)) = q, n = 1$ ;
3) while  $\varphi(T(W(v)), T(W(e))) < \lambda$  do
4)     while  $\exists d(v) == n$  and  $v \in spV'$  do
5)         if  $d(v_i) \leq T(W(v))$  then
6)              $del(v, spV', V)$ ;
7)             if  $\exists e_{ij} \in E$  then
8)                 if  $v_j \notin spV''$  then
9)                      $add(v_i, v_j, spV'')$ ;
10)                     $del(e_{ij}, E)$ ;
11)                end if
12)            end if
13)        end if
14)    end while
15)    if  $spV' \neq \emptyset$  then
16)        increase( $T(W(v)), T(W(e))$ );
17)        goto 04;
18)    end if
19) end while
20) end procedure
    
```

根据算法第 1 步的各种参数假设，如图 5 所示非焦点区域抽象率为 98.6%，路由器抽象率为 57.14%。经过算法第 2 步如无法满足抽象系数的要求，则进行算法第 3 步。

算法第 3 步：切割边抽象，对如图 5 所示拓扑进行切割边抽象后拓扑如图 6 所示。

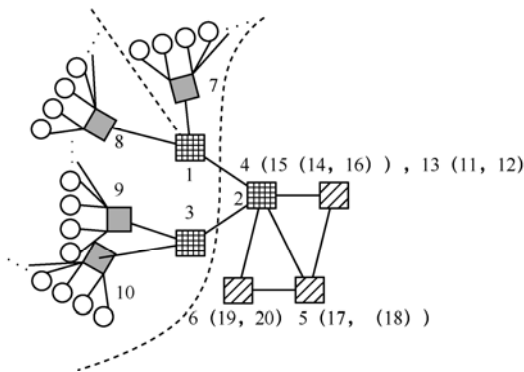


图 6 切割边抽象后的拓扑

如图 5 所示，将某些权值较小的边进行切割，从而产生新的度为 1 的节点，如节点 15 和节点 17 之间的链路，从而进一步将拓扑进行压缩。首先将  $spV$  中节点根据其权值进行排序，根据抽象系数定

义获取权值最小点的集合；然后比较最小点集合中所有节点的边权值，并排序；最后根据权值的排序依次选取边权值较小的边进行切割，并将新产生不属于  $spV$  中的节点归入  $spV$  中，递归调用第 2 步<sup>[3]</sup>，直到满足抽象系数要求。具体算法描述如下：

```

1) procedure Reduce_cut_edge
2) while  $\varphi(T(W(v)), T(W(e))) < \lambda$  do
3)     while  $\exists d(v) == n$  and  $v \in spV[n]$  do
4)         if  $d(v_i) \leq T(W(v))$  then
5)              $del(v, spV[n], V)$ ;
6)             if  $\exists e_{ij}, e_{ik} \in E$  then
7)                 if  $v_j, v_k \notin spV[n]$  then
8)                     if  $W(e_{ij}) < W(e_{ik})$  then
9)                          $add(v_i, v_j, spV[n+1])$ ;
10)                         $del(e_{ij}, E)$ ;
11)                    else
12)                         $add(v_i, v_k, spV[n+1])$ ;
13)                         $del(e_{ik}, E)$ ;
14)                    end if
15)                end if
16)            end if
17)        end while
18)    end while
19)     $Tree\_abstract()$ ;
20)    if  $spV[n] \neq \emptyset$  and  $\varphi(T(W(v)), T(W(e))) < \lambda$  then
21)        increase( $T(W(v)), T(W(e))$ );
22)         $n++$ ;
23)        goto 03;
24)    end if
25) end while
26) end procedure
    
```

根据第 1 步的各种参数假设，如图 6 所示非焦点区域抽象率为 99.03%，路由器抽象率为 71.43%。

### 3.3 算法复杂度分析

算法第 1 步采用 Dijkstra 算法计算所有终端节点间最短路径，根据算法第 1 步的假设，其最大空间复杂度为  $O(|V|^3) = O(|n|^3)$ ，最小空间复杂度为

注：当切割一条边产生新的度为 1 的节点时，调用树型收缩，进行节点抽象，同时将抽象后的代表节点归入  $spV$  中。切割过程中并非每次都只切割一条边，可同时切割权值相同的几条边。

$O(|n_r| \times |n|^2)$ 。Dijkstra 算法的时间复杂度为  $O(|V|^2) = O(|n|^2)$ ，但是需要计算所有终端节点间的最短路径，故其最大时间复杂度为  $O(|V|^3) = O(|n|^3)$ ，最小时间复杂度为  $O(|n_r| \times |n|^2)$ 。在权值计算过程中，其节点权值计算的时间复杂度为  $O(|V|)$ ，边权值计算的时间复杂度为  $O(|E|)$ ，假设所有路由器节点间都有链路，则其边权值计算的最差时间复杂度为  $O\left(\frac{n(n-1)}{2} + m \times n_r\right)$ ，故算法第 1 步的整体时间复杂度最大为  $O(|n|^3)$ ，最小为  $O(|n_r| \times |n|^2)$ 。

算法第 2 步进行树型收缩，从终端节点开始遍历，最多遍历  $n_t$  个节点，最少遍历  $n_f$  个节点。设递归遍历次数为  $k$ ，则其空间复杂度与算法第 1 步相同，其时间复杂度最大为  $O(k(n_t))$ ，最小为  $O(n_f)$ ，算法在实际应用过程中，递归遍历次数  $k$  为正整数，远小于节点的数量。

算法第 3 步进行切割边抽象，其空间复杂度和时间复杂度与算法第 2 步基本相同。此处不再赘述。

综上所述，算法整体的最大空间复杂度和时间复杂度为  $O(|n|^3)$ ，最小为  $O(|n_r| \times |n|^2)$ ，与文献[14]所提及算法的时间复杂度在同一级别。目前，大多数的拓扑抽象算法都需要计算最短路径，而计算最短路径的算法主要包括 Floyd、Dijkstra、Bellman-Ford 和 SPFA 等。当求多源最短路径时，这些算法的复杂度基本上都在同一个量级。

### 4 实验

为验证 TABFF 算法优劣，采用 4 组拓扑数据对算法进行实验分析。实验数据来源于启明星辰 2009 年探测部分拓扑，各区域拓扑信息如表 1 所示。

表 1 各模拟区域信息

序号	省市	路由器 / 个	终端路由器 / 个	终端主机 / 个	焦点区域
区域 1	北京+广东	6 548	2 748	2 748×50=137 400	北京
区域 2	北京+上海	5 884	2 148	2 148×100=214 800	上海
区域 3	北京+黑龙江+山东+广东	8 187	3 476	3 476×300=1 042 800	广东
区域 4	国家骨干+北京+上海+湖北+浙江	10 521	3 996	3 996×511=2 041 956	北京

### 4.1 抽象前后拓扑规模比较

首先比较抽象前后网络拓扑的节点和链路的变化情况。对表 1 所示的 4 组网络拓扑应用 TABFF 算法进行抽象。抽象结果如图 7 所示。

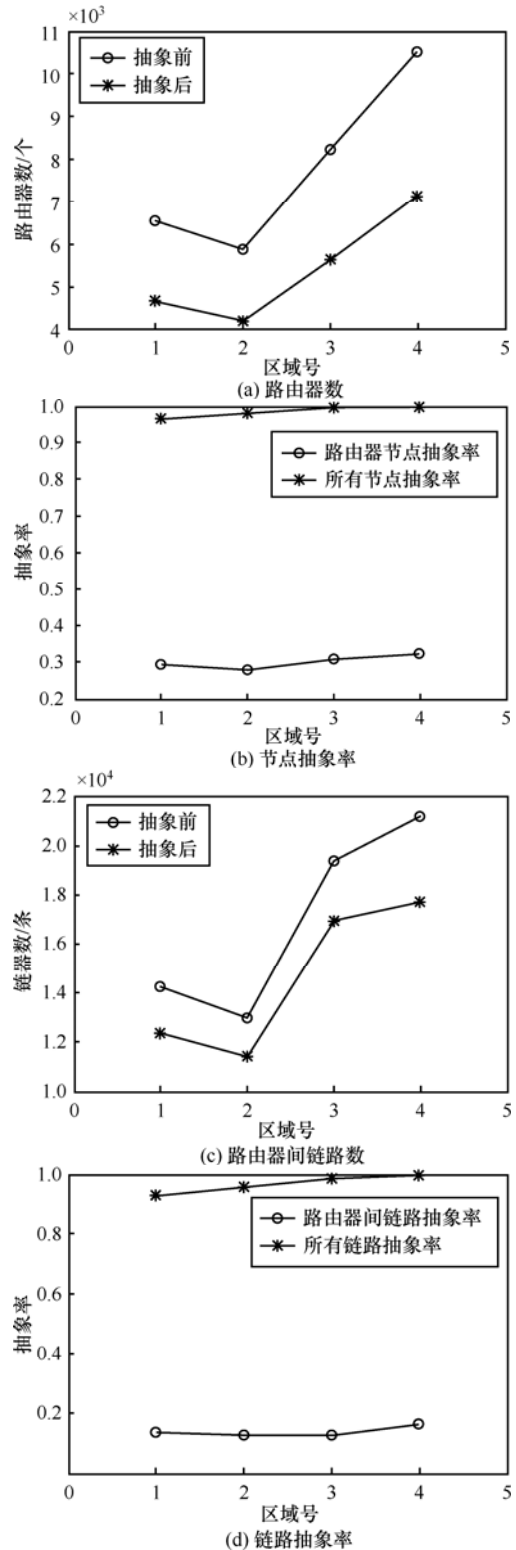


图 7 抽象前后拓扑规模比较

从图 7 中可看出，抽象后拓扑规模得到了一定压缩，路由器节点及其之间链路抽象率均值 30%和 13.74%，并且随拓扑规模的增加，抽象率基本上逐渐增加。这是因为随着拓扑规模的增加，非焦点区域在整个网络拓扑中的比例逐渐增加，进而提高了网络拓扑的抽象率。如果将主机节点的抽象算在内，所有节点和链路的抽象率均值分别为 98.48%和 96.1%，可见该算法可对拓扑进行高度的抽象。

### 4.2 抽象前后拓扑的差异性分析

从图 7 可看出，通过 TABFF 算法，拓扑被高度抽象，但研究者同样关心抽象前后网络拓扑的变化情况，因此需要对网络拓扑抽象前后的差异性进行分析。在此选取度、距离、聚集系数和度关联性进行比较分析。

1) 节点度：抽象前后拓扑的节点度分布如图 8 所示。

从图 8 中可以看出，被抽象节点主要集中在低度节点，特别是度小于 5 的节点，中度和高度节点数量有增有减，但整体趋势是减少。从图 8 (e) 可以看出，抽象前后节点平均度分别为 4.384 和 5.405，增加的均值为 1.021，约 18.83%。

2) 距离：抽象前后拓扑的距离分布如图 9 所示。

从图 9 中可以看出，抽象后拓扑的距离（距离指两终端节点间最短路径的跳数）整体减少。从图 9 (e) 可以看出，抽象前后距离均值分别为 14.461 和 12.832，距离减少均差为 1.629，约 11.5%。

3) 聚集系数：抽象前后拓扑的聚集系数分布如图 10 所示。

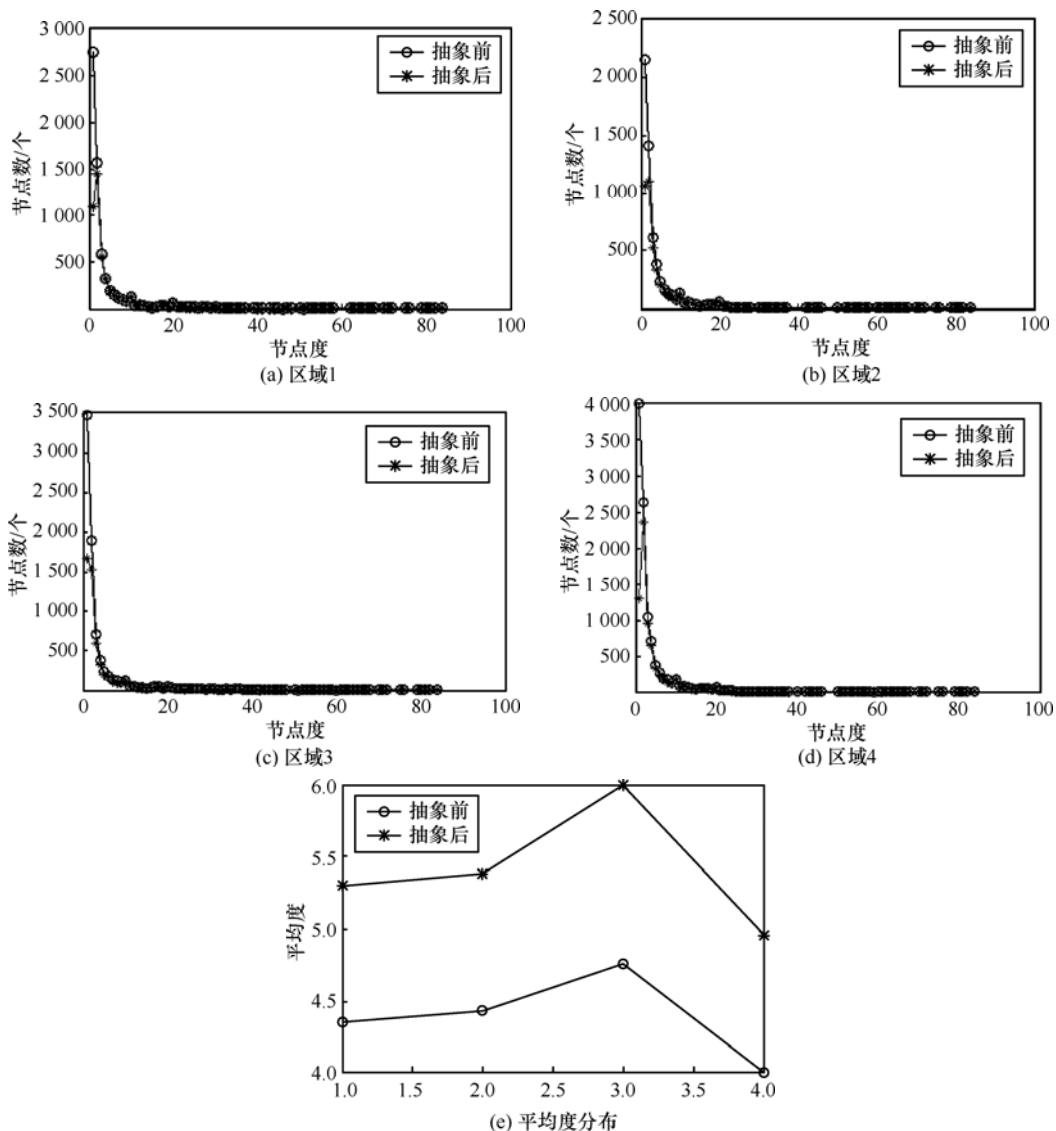


图 8 抽象前后节点度分布情况

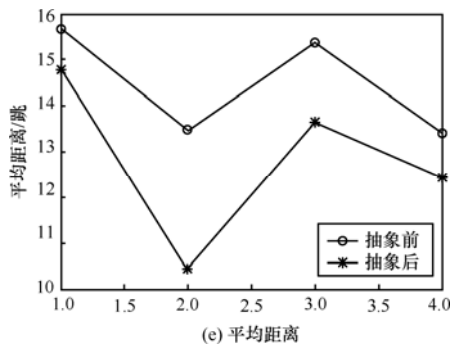
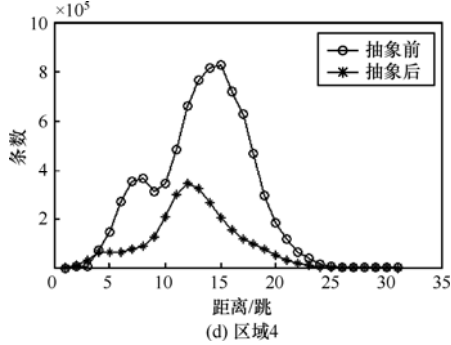
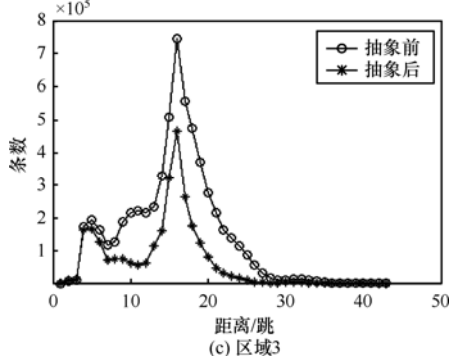
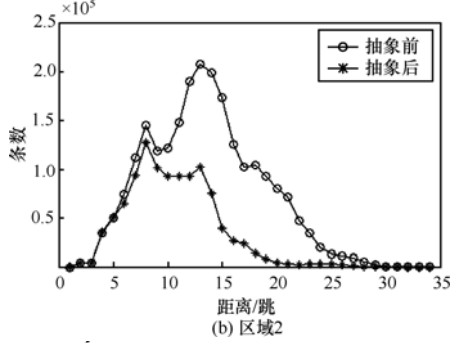
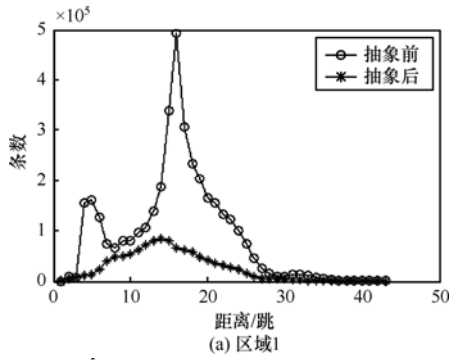


图 9 抽象前后距离分布情况

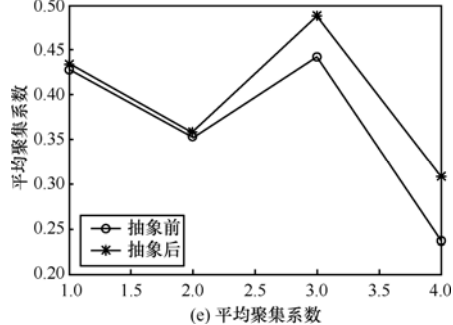
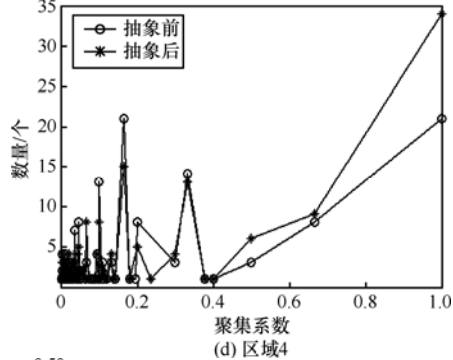
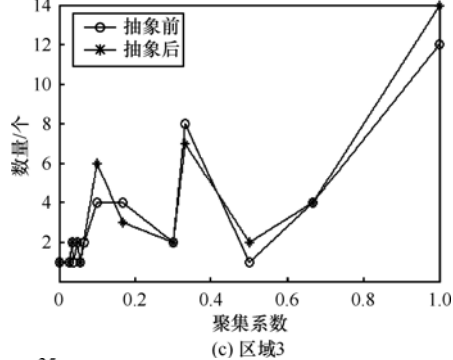
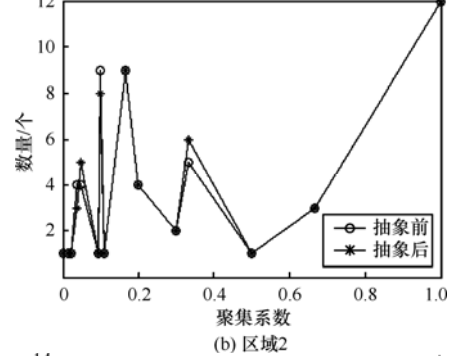
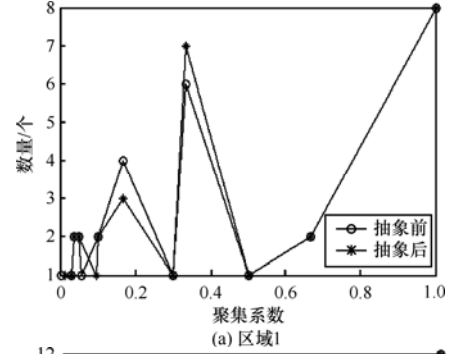


图 10 抽象前后聚集系数分布情况

从图 10 中看出, 抽象后聚集系数大于 0.5 的节点数有所增加, 在[0,0.5]范围内的节点数有增有减, 从图 10 (e) 中看出, 抽象前后聚集系数均值分别为 0.365 5 和 0.397, 聚集系数均值增加 0.031 5, 约 10.66%。

4) 度关联性: 在拓扑中, 分别从节点度在 40~50、50~60、60~70、70~80 和 80~90 范围内各选取 1 个节点, 抽象前后各节点度相关性如图 11 所示, 其中图标格式为 IP 地址—抽象前度—抽象后度。

从图 11 中可以看出, 抽象后某些节点的度相

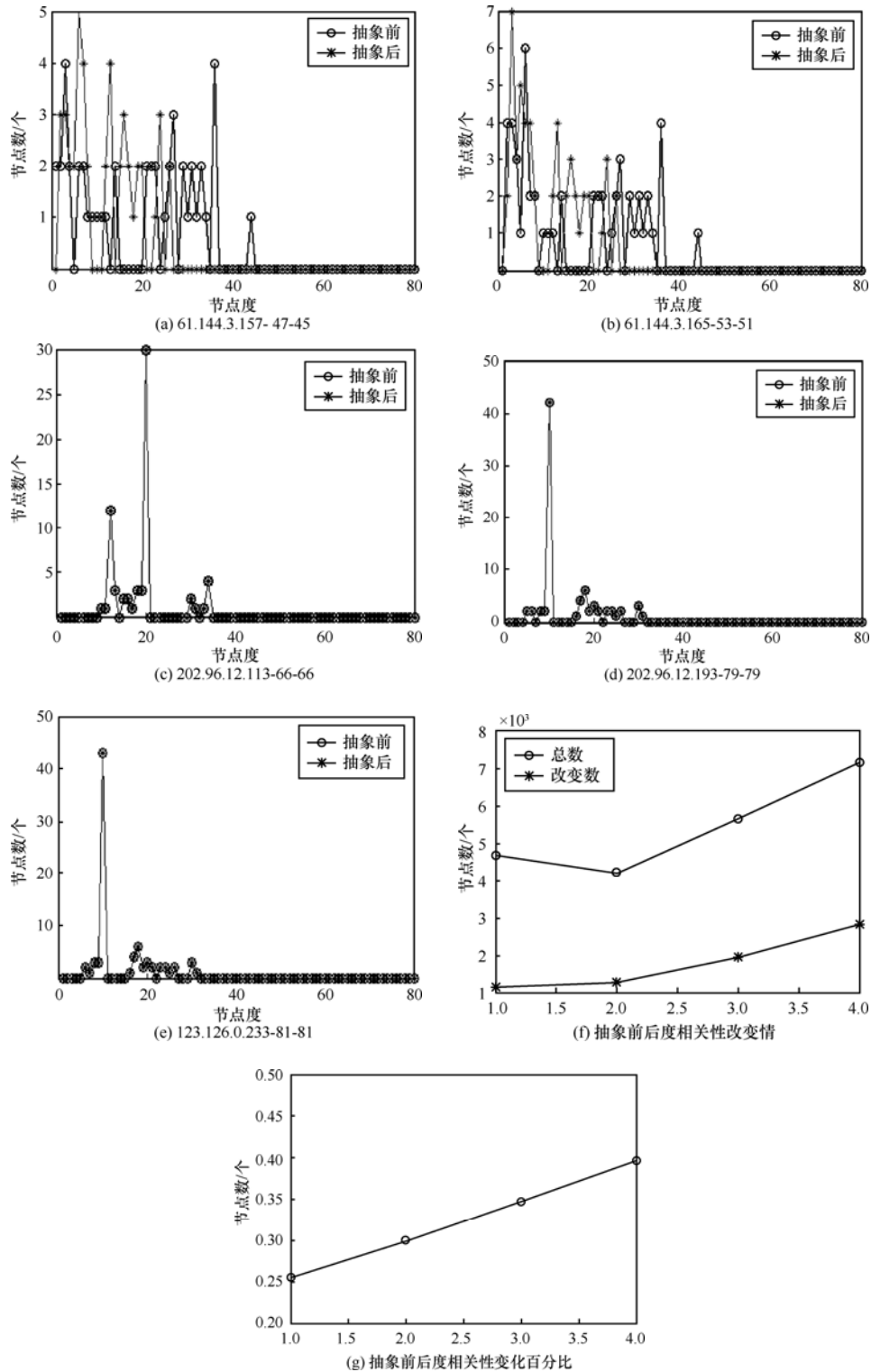


图 11 度相关性

关性发生变化, 4 个拓扑平均改变 32.46%, 这在预料之中, 因为抽象会导致节点的度发生变化。另外从图和实验结果中还发现, 高度节点与低度节点连接数更多, 这进一步说明了路由器级拓扑的度关联性呈现异配性。

抽象后节点度增加、距离减少、聚集系数增加和度相关性的变化这些都与算法的预期相符, 同时也验证了算法的合理性。

### 4.3 模拟案例测试

#### 1) 模拟性能测试

分别选取 5 万、20 万、150 万和 250 万节点规模的拓扑进行模拟, 测试抽象后系统的性能。模拟案例采用恒定流量的 DDoS, 共 10 个攻击节点, 50 个被攻击节点, 每个攻击节点以 10packet/s 的速度向受攻击节点发送数据分组, 模拟攻击时间为 8s。采用 2 台模拟机, 通过交换机互联, 模拟机配置 CPU: 双核 2.8GHz, 内存: 4GB, 硬盘: 60GB, 模拟结果如图 12 所示。

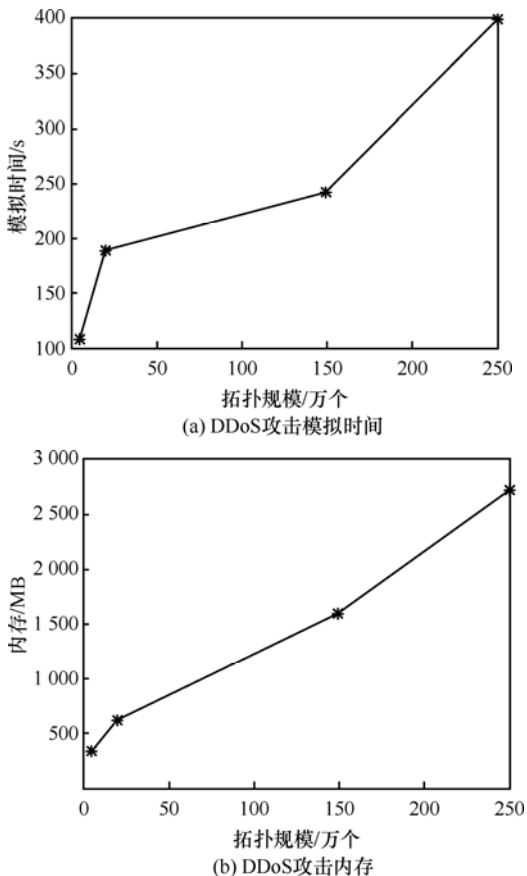


图 12 DDoS 攻击模拟结果

从图 12 中可以看出, 在 250 万节点规模环境中, 模拟上述 DDoS 案例, 需要 400s 的时间,

2 710MB 内存, 模拟时间和内存占用都在可接受的范围内。但如使用原始的 PDNS 系统, 则模拟无法进行。

#### 2) 模拟时间和内存比较

以山东省拓扑为例, 共有 1 176 个路由器, 其中终端路由器 546 个, 链路数 3 167 条, 每个终端路由器下连 32 台主机, 进行上述 DDoS 实验, 实验结果如图 13 所示。

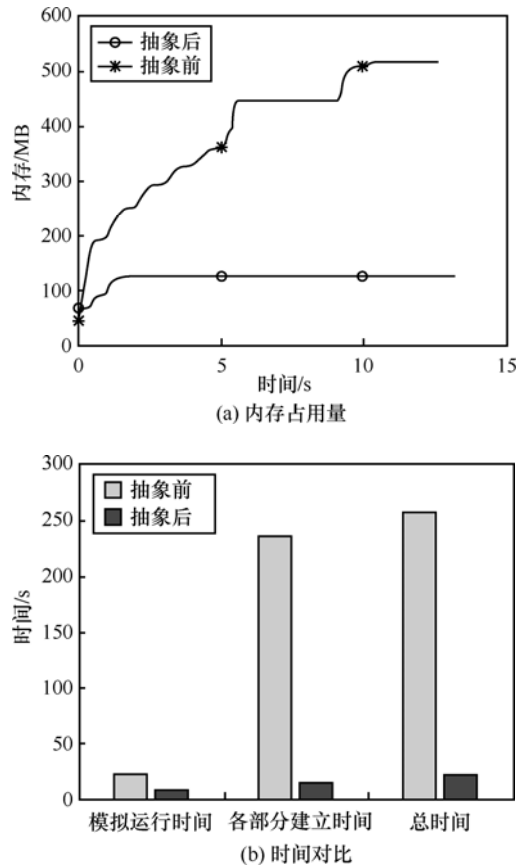


图 13 DDoS 攻击实验

通过图 13 可以看出, 系统模拟运行所需的最大内存抽象前后分别在 515MB 和 127MB, 减少了 75.34%。系统模拟初始化所需时间抽象前后分别为 233s 和 14s, 减少了 94%; 模拟运行时间抽象前后分别为 22s 和 7s, 减少了 68.18%; 模拟总时间抽象前后分别为 255s 和 21s, 减少了 91.76%。可见, 抽象极大地降低了模拟所需资源开销和模拟运行时间。

以北京市拓扑为例, 共有 3 432 个路由器, 其中终端路由器 1 091 个, 链路数为 8 997 条, 每台主机下连 32 台主机, 采用本地优先为扫描策略, 进行蠕虫扩散模拟, 模拟结果如图 14 所示。

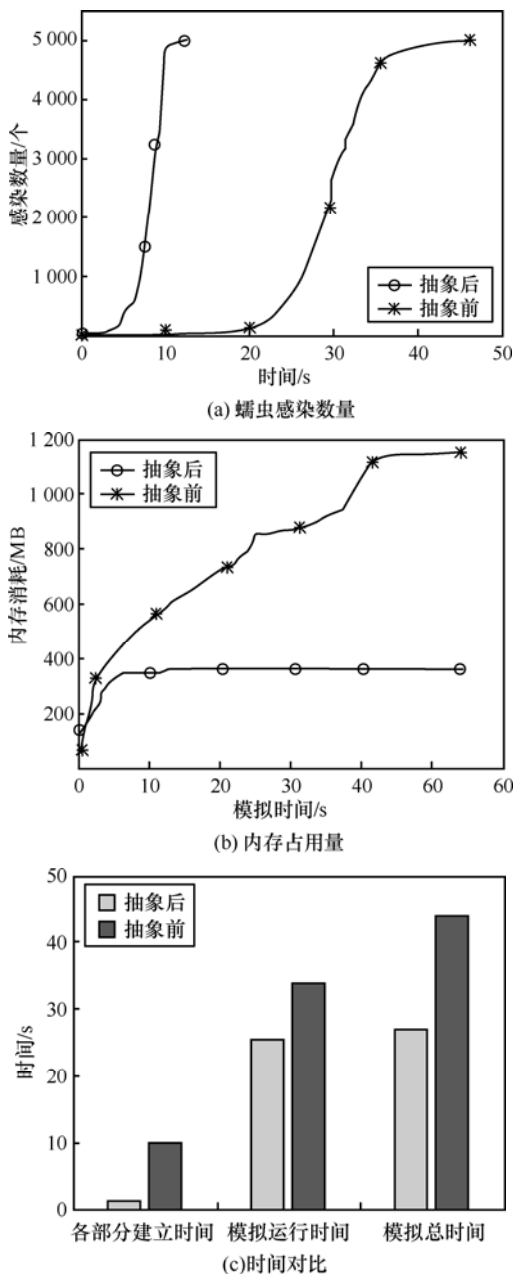


图 14 蠕虫模拟实验

从图 14 (a) 中可以看出, 抽象前后蠕虫的感染趋势一样, 并且感染的数量也一样, 而抽象后蠕虫感染速度明显加快, 这是因为抽象后链路数减少, 从而导致蠕虫传播路径缩短导致。从图 14 (b) 和图 14 (c) 可以看出, 抽象后内存减少了 68.84%, 时间减少了 38.64%。

### 5 结束语

网络模拟已成为研究 Internet 的主要方法, 如何有效地降低大规模网络模拟所需的资源消耗已成为模拟技术研究的热点。为此, 本文提出了基于焦点

折叠的网络模拟拓扑抽象模型, 对非焦点区域通过终端节点抽象算法、树型收缩算法和切割边抽象算法, 降低拓扑规模, 从而降低模拟所需的资源开销和模拟运行时间。以实验拓扑为例, 实验结果证明, 该技术可减少路由器节点约 30%, 减少路由器间链路约 13.74%, 总节点和总链路分别减少约 98.48% 和 96.1%。在 250 万节点规模的拓扑上进行 DDoS 攻击模拟仅需 400s, 占用内存 2 710MB; 以山东拓扑为例, 进行 DDoS 攻击实验, 减少内存约 75.34%, 降低模拟时间约 91.76%; 以北京市拓扑为例, 进行蠕虫传播实验, 减少内存 68.84%, 减少模拟时间 38.64%。提高模拟的规模和效率, 降低模拟的资源开销和模拟运行时间。但系统的真实性有待完善, 在后续研究工作中将继续研究完善抽象后模拟真实性的方法。

### 参考文献:

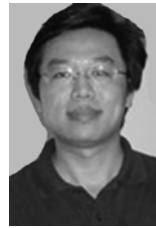
- [1] AMMAR M H. Why we STILL don't know how to simulate networks[A]. Proceedings of 13th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems[C]. Atlanta, GA, USA, 2005. 179-182.
- [2] 雷擎, 王行刚. 计算机网络模拟方法与工具[J]. 通信学报, 2001, 22(9): 84-90.  
LEI Q, WANG X G. Overview of the network simulation methodologies and tools[J]. Journal on Communications, 2001, 22(9):84-90.
- [3] YAN G. Improving Large-Scale Network Traffic Simulation with Multi-Resolution Models[R]. Department of Computer Science, Dartmouth College, 2005.
- [4] FALL K, VARADHAN K. The NS manual[EB/OL]. <http://www.isi.edu/nsnam/ns/doc/>.
- [5] OPNET[EB/OL]. <http://cn.opnet.com/solutions/>.
- [6] COWIE J, LIU H, LIU J, et al. Towards realistic million-node internet simulations[A]. Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)[C], Las Vegas, NV, 1999. 2129-2135.
- [7] NICOL D, LIU J, LILJENSTAM M, et al. Simulation of large-scale networks using SSF[A]. Proceedings of the 2003 Winter Simulation Conference[C]. New Orleans, LA, 2003. 650-657.
- [8] [http://www.cc.gatech.edu/computing/compass/pdns/\[EB/OL\]](http://www.cc.gatech.edu/computing/compass/pdns/[EB/OL]).
- [9] NICOL D M, LILJENSTAM M, LIU J. Advanced concepts in large-scale network simulation[A]. Proceedings of the 2005 Winter Simulation Conference[C]. Orlando, Florida, 2005. 153-166
- [10] ZHANG G Q, ZHANG G Q, YANG Q F, et al. Evolution of the Internet and its cores[EB/OL]. [http:// iopscience.iop.org/1367-2630/10/12/123027](http://iopscience.iop.org/1367-2630/10/12/123027).
- [11] ALBERT R, BARABÁSI A L. Statistical mechanics of complex

- networks[J]. *Reviews of Modern Physics*, 2002, 74(1): 47-97.
- [12] 张国清. 互联网拓扑结构知识发现及其应用[J]. *通信学报*, 2010, 31(10):18-25.  
ZHANG G Q. Discovery of internet topology knowledge and its applications[J]. *Journal on Communications*. 2010, 31(10):18-25.
- [13] FALOUTSOS M, FALOUTSOS P, FALOUTSOS C. On power-law relationships of the internet topology[J]. *ACM SIGCOMM Computer Communication Review*, 1999, 29(4):251-262.
- [14] SONG C M, HAVLIN S, MAKSE H A. Self-similarity of complex networks[J]. *Nature*, 2005, 433(7024): 392-395.
- [15] 李乔, 张兆心. 基于 local-area 的 Internet 路由级拓扑抽象算法[J]. *高技术通讯*. 2011, 21(9):922-927.  
LI Q, ZHANG Z X. An internet router-level topology aggregation algorithm based on local-area[J]. *Chinese High Technology Letters*, 2011, 21(9):922-927.
- [16] YANG X Q, HE H, ZHANG H L. A topology partition algorithm based on link coarsening in parallel network simulation[A]. *Proceedings of the 2nd International Symposium on Information Science and Engineering*[C]. Shanghai, China, 2009. 514-518.
- [17] HWANGNAM K, JENNIFER H, HYUK L. TranSim: Accelerating simulation of large-scale IP networks through preserving network invariants[J]. *Computer Networks*, 2008, 52(15):2924-2946.
- [18] KAI B, ULRICH K. Reducing the complexity of realistic large scale internet simulations[A]. *Proceedings of the IEEE Global Communications Conference*[C]. San Francisco, USA, 2003. 3818-3823.
- [19] KRISHNAMURTHY V, FALOUTSOS M, CHROBAK M, *et al.* Reducing large internet topologies for faster simulations[A]. *Proceedings of IFIP Networking 2005*[C]. Waterloo, Ontario, Canada, 2005. 328-341.
- [20] KRISHNAMURTHY V, SUN J, FALOUTSOS M, *et al.* Sampling internet topologies: how small can we go[A]. *Proceedings of the International Conference on Internet Computing*[C]. Las Vegas, Nevada, USA, 2003. 577-580.
- [21] MICHAEL L, DAVID M D, VINCENT H B, *et al.* Simulating realistic network worm traffic for worm warning system design and testing[A]. *Proceedings of the 2003 ACM workshop on Rapid Malcode*[C]. Washington, DC, USA, 2003. 24-33.
- [22] LILJENSTAM M, YUAN Y, PREMERE B J, *et al.* A mixed abstraction level simulation model of large-scale internet worm infestations[A]. *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*[C]. Fort Worth, Texas, USA, 2002. 109-116.
- [23] ZHOU S, MONDRAGON R J. The rich-club phenomenon in the internet topology[J]. *IEEE Communications Letters*, 2004, 8(3): 180-182.

#### 作者简介:



张兆心 (1979-), 男, 满族, 博士, 黑龙江哈尔滨人, 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。



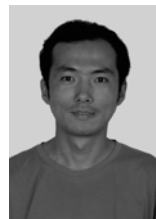
杜跃进 (1972-), 男, 山东招远人, 博士, 国家计算机网络应急处理协调中心副总工程师, 哈尔滨工业大学兼职教授, 主要研究方向为网络与信息安全、网络模拟等。



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、网络模拟等。



丁振全 (1986-), 男, 山东聊城人, 哈尔滨工业大学硕士生, 主要研究方向为网络模拟。



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