

A study on IP-based hierarchical routing strategy in network simulation^①

Yan Jianen (闫健恩), Zhang Zhaoxin^②, Shen Yinghong

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P. R. China)

Abstract

The intrinsic routing policy of NS2—a network simulation tool—based on flat or hierarchical address has the disadvantage such as small topology scale, high resource consumption, low efficiency and so on. This paper proposes an IP-based hierarchical routing strategy in network simulation to extend the topology scale and improve performance in simulating. Firstly, a stratification mechanism to identify IP address of nodes is described. Then, an algorithm of packet transmitting method (PTM) is presented for packet forwarding in the same subnet. Finally, algorithms of region partition (RP), routing computation based on region division (RCBRD) and routing distribution (RD) is proposed to implement regional segmentation, route calculation and route distribution for forwarding packets in different subnets respectively. The experiment results show that under the same condition, compared with plane address, the topology scale of the network simulation is extended two times and the consumption of time and memory during simulation is reduced by approximately 73% and 45.8% respectively. Compared with hierarchical address, the topology scale of the network simulation is expanded by more than 50%, and the consumption of time and memory is reduced by about 59.2% and 25% respectively. In addition, along with the growth of the node size, the proportion of reducing total simulation time and memory consumption would gradually increase.

Key words: network simulation, routing policy, plane routing, hierarchical routing

0 Introduction

With the advancement of global informatization, Internet has become the major driving force of economic and social development, giving rise to security problems. In order to promote Internet and enhance Internet security, it is imperative to conduct studies on Internet and its behavior. Currently, there are six major research methods to study Internet^[1-3] including mathematical model analysis, network measurement, network simulation, network emulation, prototype experiment platform, as well as real network test and experiment. In many research methods, network simulation can not only examine the results of other research methods, but is capable of performing flexible and large-scale processing and describing network behavior in detail. So it is gradually becoming a main method for studying the network and its behavior. A number of institutions both home and abroad have engaged in the research of network simulation and many have developed well-recognized simulation tools. The typical stand-alone simula-

tors include NS2^[4], OPNET^[5], SSFNet^[6] and DaSSF^[7]. Completely free with open source and enormous protocol library, NS2 is much favored and well recognized as a simulation tool among scholars. The fundamental goal of network simulation research is to scale up and improve the efficiency of the simulation with limited hardware resources. Among the many factors affecting scale and efficiency of the simulation, routing policy is one of the most critical factors. First of all, the storage of routing table is one of the major bottlenecks of the simulator storage resources, and memory is the key to the scale of simulation. Secondly, in the process of simulation, routing lookup, which consumes a large amount of computation resource, is the main influential factor of the simulation efficiency. Therefore, efficient routing computation, storage and searching strategy can significantly improve the simulation scale and efficiency. There are many routing strategies in the current network simulators, including Flat^[5], STree_Flat^[6], MTree_Flat^[7], Nix-Vector^[8,9], MTree_Nix^[10] and dynamic routing strategy^[11-14], etc. These routing strategies are based on

① Supported by the National Science and Technology Support Project (No. 2012BAH45B01), the National Natural Science Foundation of China (No. 61100189, 61370215, 61370211, 61402137) and the National Information Security 242 Project of China (No. 2016A104).

② To whom correspondence should be addressed. E-mail: heart@hit.edu.cn

Received on June 8, 2016

flat or hierarchical address in NS2. The scale of the routing policy based on the flat address restricts the scale of simulation. When the number of nodes exceeds 1000, it cannot be simulated effectively on average machines with the 1GB memory. While the address of the routing policy based on hierarchical address is solidified into three layers and does not allow to skip address, i. e. the serial number of each layer must be restricted within $0 \sim n - 1$, with each domain or cluster having only one export. The above two kinds of routing policy cannot meet the needs of the scale and the structure of network simulation. Ref. [15] presented a method of topology-aware time series by analyzing the static state neighbor index vector mechanism and looked up routing states according to the aware time of different changes, and thus reduced the time consumption during route calculation. Ref. [16] proposed a new global dynamic routing strategy on scale-free networks with heterogeneous node capacity. Ref. [17] showed an improved TZ compact routing algorithm. And Ref. [18] proposed a cross-layer approach for minimizing routing disruption caused by IP link failures, which developed a probabilistically correlated failure (PCF) model to quantify the impact of IP link failure on the reliability of backup paths. Ref. [19] studied the minimizing interactions expense of routing mechanism in structured wireless sensor network.

However, all the studies failed to cover the current simulation mechanism of underlying network routing. Therefore, this study presents a hierarchical routing strategy based on the IP address in network simulation. Firstly, according to the characteristics of the actual IP address, a stratification mechanism of IP address is designed to solve the problem of simulation node identification and provide the basis for packet forwarding and routing calculation and distribution at the same time. Then the algorithm of PTM is proposed to solve the problem of packet forwarding in the same subnet. Lastly, the algorithms of RP, RCBGD and RD are proposed to implement regional segmentation, routing calculation and routing distribution for packets forwarding in different subnets respectively.

1 The stratification mechanism of IP address

The scale of network simulation is the key factor that limits the research and development of simulation. The 32-bit address offers 4294967296 possible variations, which does not allow direct search on the plane space at the current level of memory. However, in the hierarchical address of NS2, there is an inherent weak-

ness. The routing policy of three layer address cured for three layer assigns address for the given network in advance, which hampers flexibility and proves unsuitable for large scale simulation. Therefore, a hierarchical mechanism which is proper and conforms to the actual IP address is vital and the hierarchical strategy is required to realize the routing based on IP address. At the same time, a proper kind of mask is created on the basis of hierarchical strategy to realize routing aggregation, thus greatly reducing the scale of the routing table. For network simulation, the structure of the topology is known at the beginning of the simulation. In other words, according to the number of routers or terminals each router has, each node's IP address can be directly calculated before corresponding mask and displacement are defined. The nodes of network fall into three main categories: non-terminal router, terminal router and terminal hosts (leaf nodes). To facilitate understanding, some definitions are given as follows:

Definition 1 Terminal router refers to the route directly connected with leaf nodes in the topology.

Definition 2 Non-terminal router refers to the route not directly connected with leaf nodes in the topology.

According to the above definitions, the IP address and mask are designed for the route respectively.

1.1 Hierarchical design of the route's IP address

Suppose the non-terminal router as Rut and the terminal router as Rt , which IP address satisfies the following formula:

$$IP = \begin{cases} Unid + Hnid + Cnid + Cr & Rut \\ Unid + Hnid + Hcid & Rt \end{cases} \quad (1)$$

The meaning of each field is as follows:

$Unid$ (the number of the upper-level network) refers to the number of the upper-level network some routers belonged to. Because the 32-bit addresses are not to be used up in the simulation, the network number of top routers can be specified as any upper-level network number.

$Hnid$ (the number of the local network) refers to the number of network which some routers belongs to. $Hnid$ is used to distinguish between different routers at the same level (including the superior).

$Cnid$ (the number of the subnet) is used to distinguish different routers in the jurisdiction area.

Cr (the number of the child router) is the child router's number.

$Hcid$ (the number of the host) is used to distinguish between different hosts in the jurisdiction area of the router.

1.2 The design of mask

In order to find the IP address while forwarding packets with hierarchical routing based on IP address,

the mask and displacement are designed to gain each layer information of IP address. The details of the mask are shown in Table 1.

Table 1 The mask and displacement of routers

Field name	Implication	Function
mask	Mask of the jurisdiction area	To gain the network number of the current router's jurisdiction subnet
tmask	Mask of the child router in the subnet (or terminal host)	To gain the nodes' number in the subnet
tshift	Displacement of the child router in the subnet (or terminal host)	To gain the nodes' number in the subnet
umask	Mask of the superior router's jurisdiction area	To gain the network number of the superior router's jurisdiction subnet
nmask	Mask of child router's number in the superior router's jurisdiction area	To gain the network number of the nodes in the same superior router's subnet
nshift	Displacement of child router's number in the superior router's jurisdiction area	To gain the network number of the nodes in the same superior router's subnet

Although the IP address stratification of the non-terminal router and terminal router is slightly different, the meaning of the applied mask and displacement are exactly the same, which ensures that the operation of forwarding packets of routers can be unified.

2 Strategy of forwarding packets

There are three kinds of destination addresses for routers forwarding packets: its jurisdiction subnet, neighboring networks at the same level and other networks. Therefore, every router needs to maintain subnet routing table, peer routing table and a default upper-level route. For illustrative purposes, some definitions are given first as follows:

Definition 3 Subnet routing table (srt) refers to storing routes from the current router to other routers (or terminal hosts) in jurisdiction subnet, which is one dimensional arrays indexed by the number of subnet of the current router.

Definition 4 Peer routing table (prt) refers to storing routes from the current router to other routers (including upper-level router) at the same level, which is one dimensional array indexed by the number of subnet that upper-level router has the jurisdiction and then includes the routers having the same level with the current router.

Definition 5 Default upper-level route refers to the route from the current route to its upper-level router.

When a router receives a packet, it first judges whether the packet belongs to its jurisdiction subnet ac-

cording to the destination address of the packet. If so, the router will look up the destination address in the srt. If founded, the router forwards the packet; otherwise it drops the packet. Second, it decides whether the packet belongs to the peer subnet. If so, the router will look up the destination address in the prt. If founded, the router forwards the packet; otherwise it drops it. Finally, the router judges whether there is a default route to the upper-level router. If so, it forwards the packet; otherwise it drops the packet. The detailed algorithm of packet transmitting method (PTM) is as follows:

```
procedure PacketTransmit
  if Dstip ∈ current network then
    if Dstip ∈ srt then
      transmit the packe to the current network
    else
      discard the packet
  else if Dstip ∈ upper network then
    if Dstip ∈ prt then
      transmit the packet to the upper network
    else
      discard the packet
  else if Dstip have ur then
    transmit the packet to the upper router
  else
    discard the packet
end if
end procedure
```

For the IP address which does not exist in the simulation network, the packet is not discarded immediately, but forwarded to the corresponding subnet and discarded when the hierarchical routing strategy based on IP address is used, which is consistent with the behaviors of forwarding packets in the routers of real network.

During the simulation, judgment order of routers can be adjusted according to the need so that the number of judgments can be reduced further to improve the efficiency of forwarding.

3 Strategy of route calculating and releasing

For the hierarchical routing strategy based on IP address, a part of routes could be directly implemented by setting the IP address, but the routes between different networks still need to be implemented through calculation. On one hand, because of the routing calculation work in the process of the simulation, Flat routing policy and Nix-Vector routing policy will be affected in terms of processing efficiency. On the other hand, the general use of static routing in the large-scale network simulation and basic fixed network topology structure leads to a largely fixed routing information. Based on the above considerations, routing calculation is separated from simulation in the realization of hierarchical routing strategy based on IP address, i. e. the routes of the whole simulation network are calculated uniformly and then they are released to the routers, rather than calculate the local routing and remote routing between physical nodes separately as do Flat routing policy and Nix-Vector routing policy.

3.1 Routing calculation

For a network to be simulated, if the shortest path between all routers is to be directly calculated, an enormous amount of computing and storage expenditure is indispensable, so routing computation based on region division (RCBRD) is applied. As Fig.1 shows, all router nodes in the simulation network are divided by federating the top-level routers (level 1 in the Fig.1) to form a connected graph, federating the non-top routers with their upper-level routers to form a connected graph and federating the terminal router and its directly connected host nodes in the jurisdiction area, thus forming some calculating domains (the routing computation domains circumscribed by the circle with the dotted line in Fig.1). Through division, it can effectively reduce routing storage space and computational complexity.

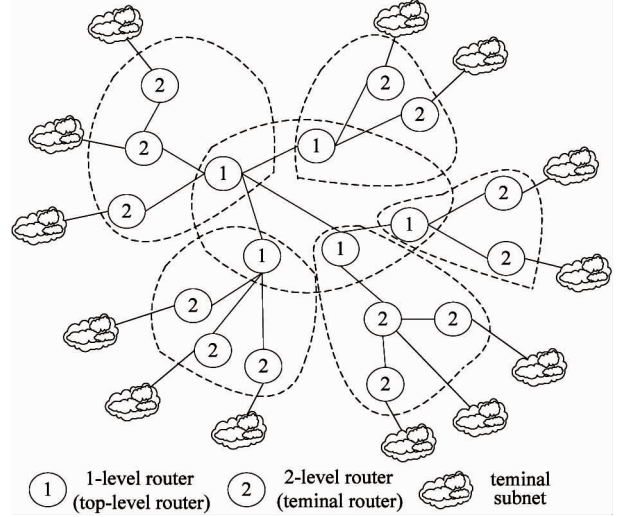


Fig.1 The example of the division of routing calculating domain

3.1.1 Region partition

In order to reduce routing storage space and improve the efficiency of routing lookup, the pre-simulation network topology is partitioned. The nodes in the network topology are first classified according to their connectivity, to find the nodes (key nodes) with higher connectivity. Second, among some minimum spanning trees that overlay the network topology, one with high coverage and more covering key nodes is selected. Intuitively, the minimum spanning tree with the minimum height and high degree node as root may cover more of the shortest paths^[20]. And to ensure that the height of the tree is minimum, the breadth first search algorithm is used to generate the minimum spanning tree. Then the threshold of the core area is set to reduce the node of the minimum spanning tree with degree one recursively, whereby information that the node associated with is recorded. Finally, the core region is obtained and the whole computational domain is acquired according to the core region and the correlation information between nodes.

According to the above ideas, it is supposed L_n for the connectivity of node n , and K_n for the key node whose number is n . Then the specific algorithm of region partition (RP) is described as follows:

procedure RP(*topo _file*)

$L_n = \text{search}(\text{topo_file})$;

sorting and storing node by L_n ;

obtain K_n ;

while $N_{\max(L_n)} - 1 > 0$ **do**

$MTree = \text{BFS}(n)$;

end while

compare and obtain the best $MTree$;

while $N_{res} > \text{some level}$ **do**

if $L_n = 1$ **then**

```

        del and store the node from the best
        MTree;
    end if
end while
obtain the core computing region by level;
obtain all the computing region by core region
and topo _file;
end procedure

```

3.1.2 The algorithm of routing calculation

According to the division design, first the shortest path information is obtained by applying the Dijkstra algorithm to the core region and store it. Then the router number is stored to the appropriate array by the router's connectivity. Finally, Dijkstra is used to calculate the information of the shortest path for each of the computational domain to be output according to the father-Array[i].

3.2 Routing distribution

When all the routing information of the computational domain has been completely calculated, the routing information is distributed to the corresponding node according to the relationship in the routing entries between the source address and the destination address. The detailed algorithm of routing distribution (RD) is as follows:

```

procedure RD
if Dst ∈ current network then
    add Dst to srl
else
    add Dst to prt
    if Dst ∈ upper network then
        set Dst upper router
    end if
end if
end if
end procedure

```

4 Experiment and result analysis

There are three key factors influencing the performance of network simulation: the scale of simulation, the time of simulation and the authenticity of simulation results. The first two are the usage of memory and the consumption of time in the simulation. Many of the routing protocols are running in the application layer, but the proposed routing strategy designed is to run in the network layer. So the aims of some new routing strategies based on content or energy are different from the strategy putting forward in this article. The aim of the hierarchical routing strategy based on IP address is

to improve the scale of the simulation, and at the same time, reduce the time consumption and memory usage of the simulation. Experiment was done on the Downing-server (CPU: AMD 2220 × 2, Memory: 4G, HDD: 146G) with fixed CBR application, and it makes comparison of the time of nodes creating, links creating, application running and total simulation and the percentage of the memory usage.

4.1 Experiment and result analysis

First 200, 500, 1000, 500, 4000, 5000, 4000, 5000 and 15000 nodes are set up respectively for the topology, and topologies with 200, 500, 1000, 2000, 4000, 5000, 4000, 5000 and 15000 links. Then 40, 100, 200, 400, 800, 1000, 800, 1000 and 3000 CBR applications are established respectively under the above conditions to run for 2s, and the results are as Figs2 ~ 16 show: flat is for routing strategy based on flat address, hier3 is for routing strategy based on hierarchical address, and hier4 is for hierarchical routing strategy based on IP address.

As Fig.2 shows, it can only set up 5000 nodes using the method of flat, 10000 nodes using the method of hier3, but more than 15000 nodes using the method of hier4, which indicates that hier4 is more suitable for large-scale network than flat and hier3. But as Fig.2 shows, the time of creating nodes for hier4 is longer than flat's and hier3's, because route computing and route distribution must have done for hier4, which do not happen in flat and hier3 in the process of building the node. As Fig.3 shows, with the number of nodes increasing, the time growth ratio of creating nodes for hier4 is longer than flat's and hier3's. As Fig.4 shows, the creating time of hier 3 is about 2.23 times

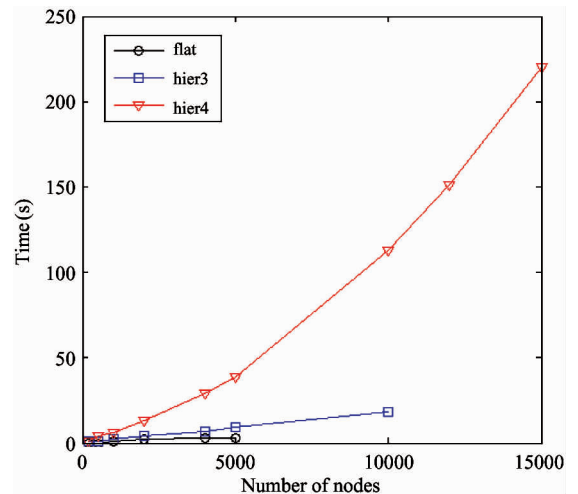


Fig. 2 Time of creating nodes

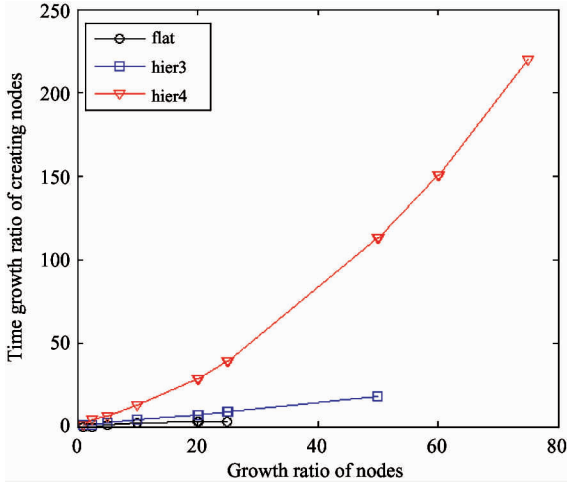


Fig. 3 The time growth ratio of creating nodes

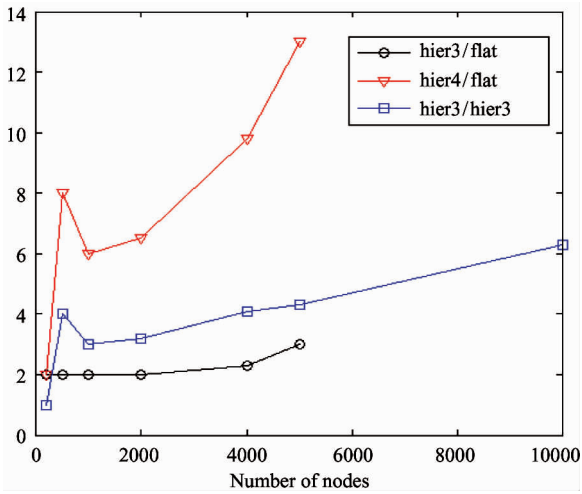


Fig. 4 Time comparison of creating nodes

as much as flat's. The creating time of hier4 is about 7.87 times as much as flat's, 3.37 times as much as hier3.

As Fig. 5 shows, the consumption time of creating links for flat and hier3 is basically the same within 5000 nodes. The time of hier4 is much less. As Fig. 6 shows, the time for creating link of hier4 increases slowly. When the number of nodes scales up from 200 to 15000, the time only increases from 0.5s to 5s, about 10 times. For flat strategy, when the number of nodes scales up from 200 to 5000, the time increases from 2 s to 38s, about 19 times. And it cannot establish links when the number of nodes is more than 5000. For hier3 strategy, when the number of nodes scales up from 200 to 10000, the time increases from 1s to 115 s, about 115 times. And it cannot establish links when the number of nodes is more than 5000. As Fig. 7 shows, for 5000 nodes, the establishing time of nodes for hier4 is 0.08 times as much as hier3 and flat. For 10000 nodes, the establishing time of nodes

for hier4 is 0.03 times as much as hier3. All this indicates that hier4 has greatly reduced the time of link establishment.

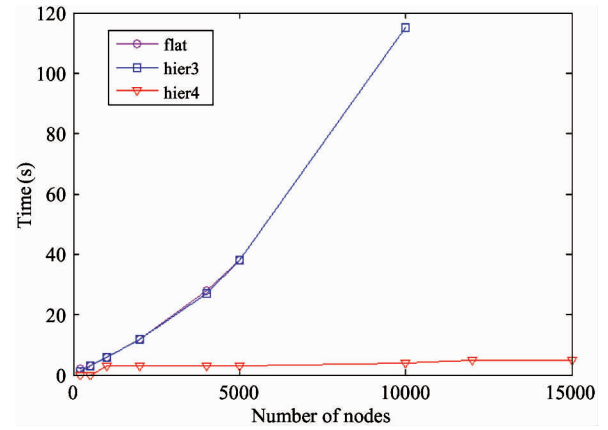


Fig. 5 Time of creating links

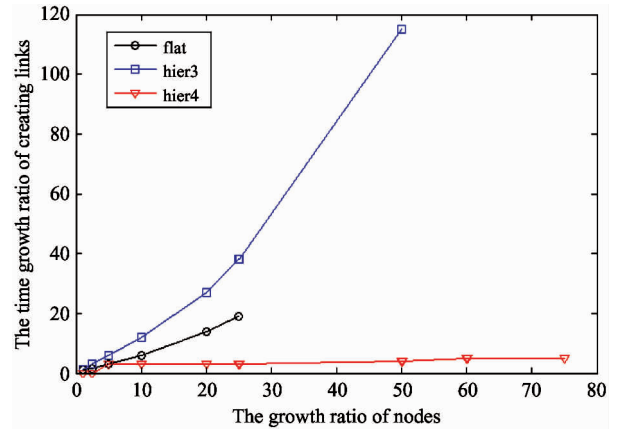


Fig. 6 The time growth ratio of creating links

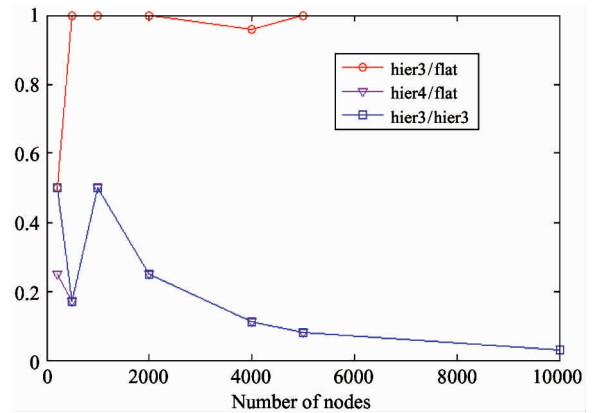


Fig. 7 Time comparison of creating links

As Fig. 8 shows, it has been unable to run the simulation for flat when the scale is more than 4000 nodes although nodes and links have been established, because memory consumption is too large. It is unable to run simulation because of the large memory consumption when the number of nodes is more than

10000 for hier3. When the number of the nodes is 4000, the time of simulation is respectively 21s, 118s and 1375s for hier4, hier3 and flat, reduced by 82.2% and 98.5% respectively. As Fig. 9 shows, the growth ratio of running time of hier4 is lower than

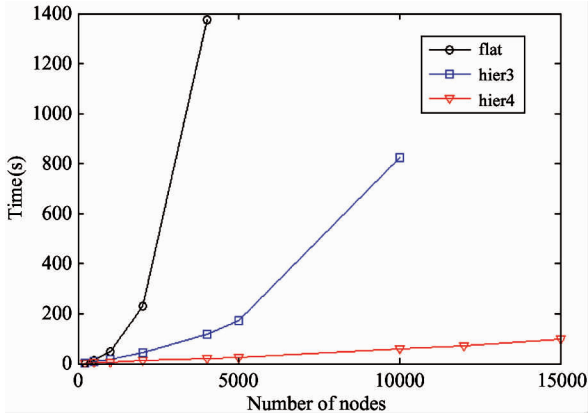


Fig. 8 Running time

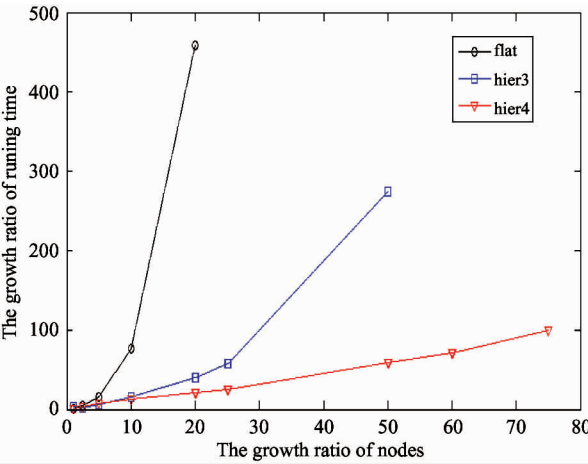


Fig. 9 The growth ratio of running time

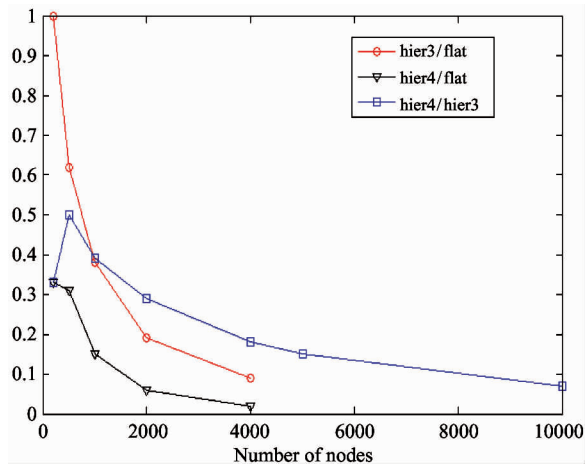


Fig. 10 Comparison of running time

flat's and hier3's with the growth of the scale of topology. And as Fig. 10 shows, for 4000 nodes, the run time for hier4 is 0.02 times as much as flat, 0.18 times as much as hier3. For 10000 nodes, the running time for hier4 is 0.07 times as much as hier3. In addition, with the growth of the scale, magnification also decreases, which greatly reduces the running time.

As Fig. 11 shows, the simulation time is respectively 1406s, 152s and 53s for flat, hier3 and hier4 routing strategies when the number of scale is 4000, which is reduced by 96.2% and 65.1%. The simulation time being 956s and 176s respectively for hier3 and hier4 routing strategy when the number of scale is 10000, is reduced by 81.6%. As Fig. 12 shows, the growth ratio of simulation time for hier4 routing strategy is lower than hier3 and flat. As Fig. 13 shows, the simulation time of hier4 policy is 0.04 times as much as flat, 0.35 times as much as hier3 for the number of nodes is 4000. For 10000, the simulation time of hier4 is 0.18 times as much as hier3. Compared with flat and

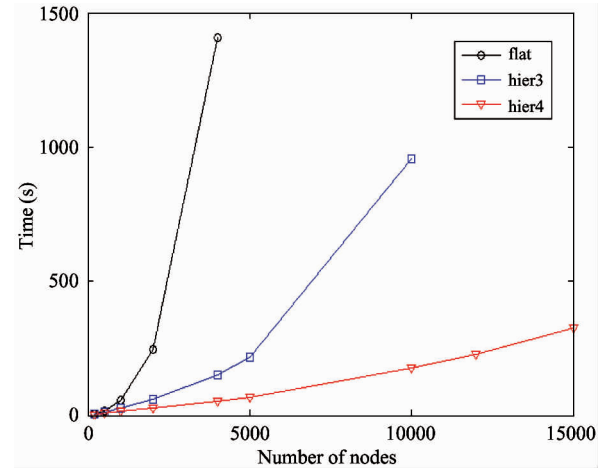


Fig. 11 Simulation time

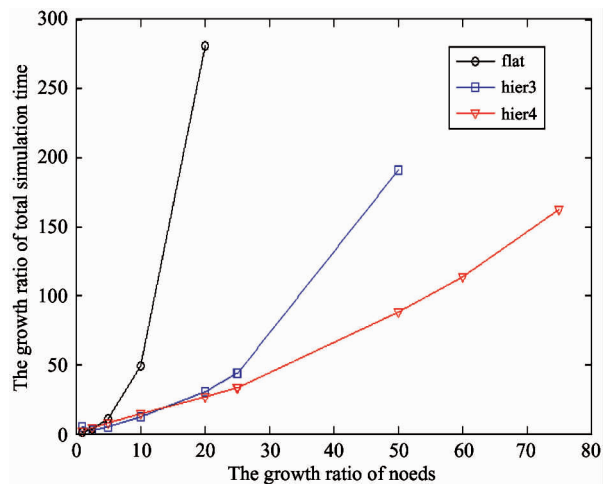


Fig. 12 The growth ratio of simulation time

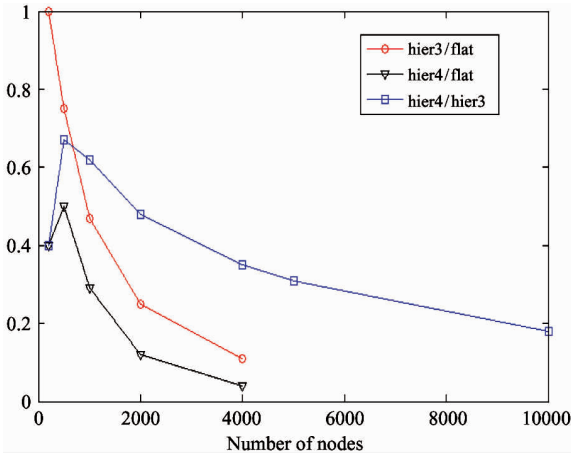


Fig. 13 Comparison of simulation time

hier3 policy, the average simulation time is reduced by about 73% and 59.2%. It shows that although the establishing time of hier4 is more than flat's and hier3's, the simulation time is significantly reduced.

As Fig. 14 shows, the memory usage of hier4 is smaller than flat's and hier3's. The memory usage is 1011MB, 288M and 198MB for flat, hier3 and hier4 routing strategies respectively when the number of nodes is 4000, which is reduced by 80.4% and 31.3%. The memory usage is respectively 1178MB and 463MB for hier3 and hier4 routing strategy when the number of node is 10000, reduced by 60.7%. As Fig. 15 shows, the growth ratio of memory usage for hier4 is lower than hier3 and Flat. As Fig. 16 shows, the memory usage of hier4 policy is 0.2 times as much as flat and 0.69 times as much as hier3 for the number of nodes is 4000. For 10000, the memory usage of hier4 is 0.39 times as much as hier3. Compared with flat and hier3 policies, the average simulation time is reduced by about 45.8% and 25%, indicating a remarkable decrease in memory consumption.

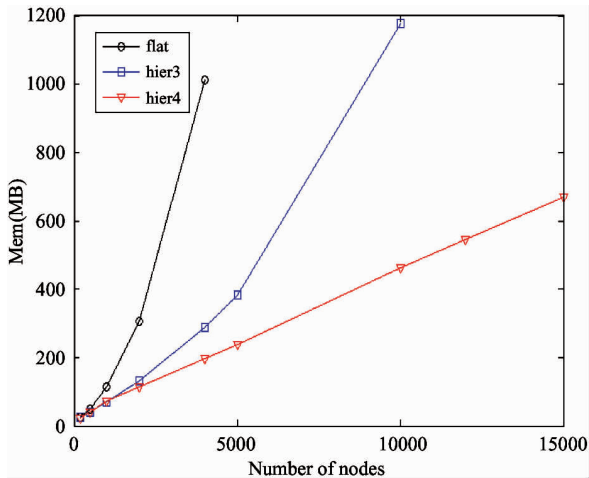


Fig. 14 The memory usage

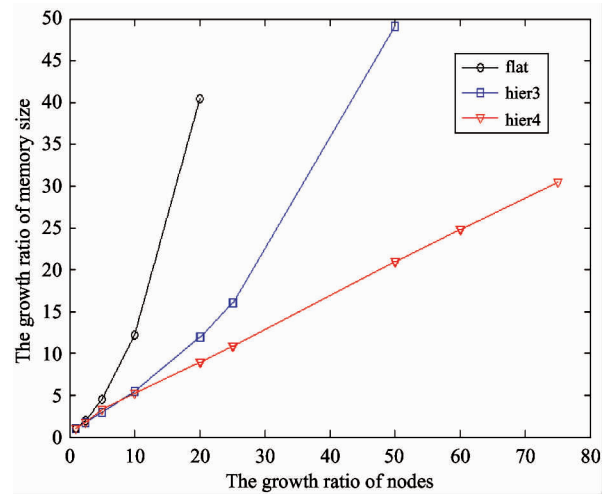


Fig. 15 The growth ratio of memory usage

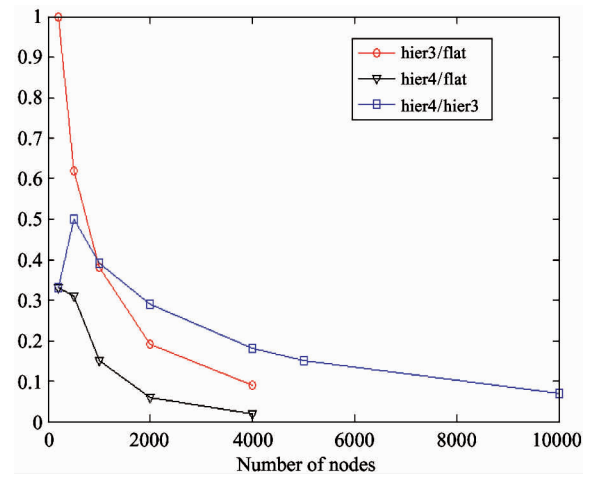


Fig. 16 The comparison of memory usage

As is shown in the results and analysis of the experiments, compared with the routing policies based on flat address or hierarchical address, the hierarchical routing strategy based on IP address has significantly enhanced the scale of the simulation and greatly reduced the simulation time and the resource consumption.

4.2 Correctness of the hierarchical routing strategy based on the IP address

The purpose of the network simulation is to output trace files for users to analyze. The trace file of hierarchical routing strategy based on the IP address is identical to the Trace file of flat IP address and the hierarchical IP address, which indicates that the strategy is correct.

5 Conclusions

Routing strategy is one of the critical factors that

restricts the scale of network simulation. This paper points out and analyzes the weaknesses of routing strategy based on flat address or hierarchical address in NS2, such as small scale, high resource consumption, low efficiency, fixed addresses and poor extensibility. Then a hierarchical routing policy based on the IP address is proposed. With authenticity ensured, the strategy scales up topology, reduces simulation time and memory consumption, so as to improve simulation performance and efficiency. The experiment results show that compared with the flat address, the scale of network simulation is expanded by more than 200% and the consumption of time and memory is reduced by about 73% and 45.8% respectively, while compared with the hierarchical address, the scale of the network simulation is expanded by more than 50% and the consumption of time and memory is reduced by about 59.2% and 25% respectively. The proportion of reducing total simulation time and memory consumption would gradually increase.

References

- [1] Ammar M H. Why we still don't know how to simulate Networks. In: Proceedings of 13th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Atlanta, USA, 2005. 179-182
- [2] Lei Q, Wang X G. Overview of the network simulation methodologies and tools. *Journal on Communications*, 2001, 22(9): 84-90
- [3] Yan G. Improving Large-Scale Network Traffic Simulation with Multi-Resolution Models. Technical Report TR2005-558, Department of Computer Science, Dartmouth College, 2005
- [4] Fall K, Varadhan K. The NS manual. <http://www.isi.edu/nsnam/ns/doc/>; isi, 2011
- [5] Zhang Z X, Du Y J, Ke W, et al. Local routing strategy in network simulation based on zone _ cut. *Journal on Communications*, 2013, 34(7):14-23
- [6] Cowie J, Liu H, Liu J, et al. Towards realistic million-node internet simulations. In: Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 99), Las Vegas, USA, 1999. 2129-2135
- [7] Nicol D, Liu J, Liljenstam M, et al. Simulation of large-scale networks using SSF. In: Proceedings of the 2003 Winter Simulation Conference. New Orleans, USA, 2003. 650-657
- [8] Huang P, Heidemann J. Minimizing routing state for light-weight network simulation. In: Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Cincinnati, USA, 2001. 108-116
- [9] Hiromori A, Yamaguchi H, Yasumoto K, et al. Reducing the size of routing tables for large-scale network simula-

- tion. In: Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, San Diego, USA, 2003. 115-122
- [10] Chen J, Gupta D, Vishwanath K V, et al. Routing in an internet-scale network emulator. In: Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Volendam, Netherlands, 2004. 275-283
- [11] Riley G F, Ammar M H, Zegura E W. Efficient routing with Nix-Vectors. In: Proceedings of IEEE Workshop on High Performance Switching and Routing, Dallas, USA, 2001. 390-395
- [12] Riley G F, Fujimoto R, Ammar M H. Stateless routing in network simulations. In: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, San Francisco, USA, 2000. 524-531
- [13] Hao Z Y, Yun X C, Zhang H L. MTree-Nix routing computation and lookup mechanism in network simulation. *ACTA Electronica Sinica*, 2008, 36(3):477-481
- [14] Hao Z Y, Zhai J H, Yun X C, et al. Research on dynamic routing mechanism in network simulation. *Journal on Communications*, 2007, 28(12):19-24
- [15] Sun L G, Zhang M Q, Dong S Q, et al. Dynamic routing mechanism in network simulation Based on topology-aware time series. *Journal of Information Engineering University*, 2016, 17(1):71-75
- [16] Zou Y, Tang B. Global dynamic routing for scale-free networks with heterogeneous node capacity. In: Proceedings of IEEE International Conference on Network Infrastructure and Digital Content, Beijing, China, 2014. 36-40
- [17] Qin X W, Zhou M Y, Zhou Z, et al. Study on TZ compact routing schemes based on the coberage of landmark. *Journal of University of Science and Technology of China*, 2015, 45(7):594-600
- [18] Zheng Q, Cao G, Porta T F L, et al. Cross-layer approach for minimizing routing disruption in IP networks. *IEEE Transactions on Parallel & Distributed Systems*, 2014, 25(25):1659-1669
- [19] Xiong S G, Yu L, Shen H Y, et al. Efficient algorithms for sensor deployment and routing in sensor networks for network-structured environment monitoring. In: Proceedings of INFOCOM, Orlando, USA, 2012. 1008-1016
- [20] Chen J, Gupta D, Vishwanath K, et al. Routing in an internet-scale network emulator. In: Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Volendam, Netherlands, 2004. 275-283

Yan Jianen, born in 1977. He is a Ph. D candidate in the School of Computer Science and Technology of Harbin Institute of Technology. His main research fields are network simulation, botnet and network attack and defense technology.