

Yujie PEI, Hongbo WANG, Shiduan CHENG

A self-adaptive key flow routing adjustment algorithm

© Higher Education Press and Springer-Verlag 2009

Abstract Congestion may decrease throughput and transfer efficiency of network, and lead to serious degradation of quality of service (QoS) acquired by end users. Present routing adjustment methods against congestion mostly consider single-point congestion. They cannot deal with multi-point congestion. This paper presents a probability-based routing adjustment algorithm, which solves the interference problem when several flows are adjusted simultaneously. While multiple points in the network are being or about to be congested, several key flows are rerouted by this algorithm at the same time to make traffic distribution more reasonable so as to avoid congestion. Meanwhile, the algorithm in this paper confines the path length of key flows to avoid QoS reduction due to overlength of key flows. Simulation shows that this method, compared with present algorithms, maintains better load balance and path length. Therefore it effectively increases the throughput of the network.

Keywords network management, load balancing, quality of service (QoS), routing adjustment

1 Introduction

Congestion means the traffic through links or nodes is too much to be processed. The service mode adopted by Internet is best-effort. When congestion happens, Internet can do nothing but drop packets, which seriously impairs reliability of transmission and quality of service (QoS). Not only does congestion impact on end users, network providers are also unwilling to encounter it. The reason is that it is harmful to throughput and transmission efficiency of networks. Therefore, how to avoid congestion, meet the QoS demand of end users, and enhance the reliability of networks have become important tasks demanding a

prompt solution by network managers and a hotspot of network research recently as well.

There are three causes of formation of core network congestion, which are uneven traffic distribution [1,2], link failure [3,4], and routing change [5,6]. If a great deal of traffic suddenly emerges and is not distributed rationally in the network, some links may be congested due to insufficient processing capacity. When some nodes lose all or some of transmission capacity because of failure, lots of packets that cannot be forwarded will gather at fault links and related links in a short time, behaving as congestion. In addition, if the routing of network varies, congestion will emerge at some links at the moment of change due to this sudden variation. In these cases, congestion has different causes, but the principle of manipulation is the same. That is, to make traffic distribute rationally, bypass potential congestion points, and then avoid congestion.

To cope with congestion, some researchers [7,8] devise corresponding routing schemes. With these schemes, the routes of traffic passing through congestion points are recalculated according to traffic distribution stratagem and network topology when congestion is to emerge. These methods can manage single-point congestion, but do not take multi-point congestion into consideration. Multi-point congestion means that several links congest simultaneously. Although the frequency of this is lower than that of single-point congestion, it is harder to deal with. According to statistics [3], 30% of congestion in backbone networks belongs to multi-point congestion. There are many causes for it. Firstly, new multimedia applications generate a large volume of traffic, and P2P communication aggravates the shortage of bandwidth further. This increases the probability of multi-point congestion. Secondly, some links in network topology are correlated. Some of them are vulnerable when others are congested. Thirdly, the instability of routing may lead to multi-point congestion due to the inconsistency between routing and traffic distribution.

The most difficult job in dealing with multi-point congestion is to solve interference problem of multiple flows. If congestion emerges at multiple links, we have to

Received February 11, 2009; accepted April 1, 2009

Yujie PEI (✉), Hongbo WANG, Shiduan CHENG
State Key Laboratory of Networking and Switching Technology, Beijing
University of Posts and Telecommunications, Beijing 100876, China
E-mail: yjpei51@gmail.com

calculate routes for flows passing through each congestion point. However, the routing adjustment algorithm cannot calculate current distribution of traffic alone. It has to consider the interference of routing adjustments from other congestion points. The interference problem is NP-hard, and there is no accurate algorithm with polynomial complexity for it.

To deal with multi-point congestion, this paper presents a global key flow routing adjustment algorithm, which is based on probability to select next hop. The so-called key flows refer to those flows that have great impact on network behavior. Usually they are large flows. When several links are congested, the next hop is chosen for each key flow with a probability method according to current distribution of traffic until, at last, a route that reaches the destination is acquired. We select an appropriate route for each key flow that needs to be rerouted when several links are congested, so that the traffic distribution becomes more reasonable and the throughput and reliability of network is enhanced. Moreover, the algorithm confines the path length of key flows; therefore, it guarantees QoS for end users.

This paper is organized as follows. Section 2 introduces some related work. Section 3 describes the problem and builds the model. Then the routing adjustment algorithm is proposed in Sect. 4. In Sect. 5 the results of our simulation are presented. Finally, Sect. 6 contains concluding remarks.

2 Related work

The previous research about congestion focuses on the optimization of network itself. By building well-structured networks the probability of congestion is decreased. The work of Refs. [9,10] is built on open shortest path first (OSPF) protocol. They achieve the purpose of load balancing by altering link weights of OSPF. This kind of method calculates the optimal link weights of given topology and traffic matrix to evenly distribute traffic and avoid congestion. This is a kind of proactive method and is able to decrease the probability of congestion. Nevertheless, this method has two disadvantages. First, the calculation of optimal link weights is over dependent on the traffic matrix that is presumed. If traffic varies a lot, the effect of this method is discounted. In real networks, this variation exists indeed. Second, proactive methods are short in strategy when congestion has already emerged. The calculation of optimal weights is highly complex so that it is impossible to do in real time according to current congestion of networks.

Congestion may occur at any moment in networks. While proactive methods decrease the probability of it, reactive methods are required to relieve it. Yang et al. [7] proposed a method to specify route for traffic by source nodes when congestion happens to bypass congestion

points. The method in Ref. [4] is to prepare several sets of configurations, each of which corresponds to a routing scheme when certain links fail. Once congestion is perceived, system will switch to the corresponding routing scheme immediately. References [11,12] differ from above algorithms which aim at congestion point. Instead, the target of their research is flows. They try to find two disjoint paths for each source and destination pair, one of which is the main route and the other a backup. When network is in order, all flows are forwarded along the main route. In case of an exception, the backup route starts up. Because the main route and the backup are disjoint, this kind of method guarantees the whole system to work well when a single-link failure arises. Although the reactive methods mentioned above to relieve congestion achieve certain effects, all of them are realized to manage single-point congestion only and consider less the instance that several flows need to be rerouted simultaneously due to concurrent congestion of several links.

It is called multi-commodity problem when several flows are rerouted at the same time to balance the distribution of traffic and then enhance the throughput and reliability of networks. Multi-commodity problem is divided into two kinds, which are splittable and unsplittable, respectively. Baier et al. [13] systematically analyzed splittable multi-commodity problem, that is, the traffic between each pair of nodes is allowed to be transferred concurrently along with several paths. Because this leads to disorder of packets, which is harmful to Transmission Control Protocol (TCP) [14,15] and User Datagram Protocol (UDP) [16], we do not take this scenario into consideration but take only the other problem, i.e., unsplittable multi-commodity problem. This problem is NP-hard [17]. Although there is near-optimal solution [18], it only applies to uniform networks. That is to say, residual bandwidth of all links in the network must be equal. This requirement restricts the practical application of this algorithm.

To deal with multi-link congestion in real networks, researchers apply randomized load balancing (RLB) to the design of backbone networks. This concept was originally advanced by Valiant [19], and the process is made up of two steps. First, the packets having entered the network are randomly sent to an intermediate node. Second, packets are forwarded along with the shortest path between intermediate node and destination node. The random transferring of step one makes the spatial unevenness of traffic distribution smooth to some extent. And the shortest path algorithm of step two guarantees that the path length of packets will not be too long. Recent research [20] shows that if the intermediate nodes are not composed of the whole nodes of the network but a group of them instead determined by the shortest path tree, the effect of RLB is better. But RLB is still a proactive method and not very agile when congestion has already emerged.

To deal with the disadvantages of the above mentioned

research, this paper proposes a global key flow routing adjustment algorithm, which is based on probability to select next hop. This algorithm is executed by a global routing server (RS), which is also responsible for acquisition of global information on traffic distribution. Combining the bandwidth utilization information collected, RS selects next hops for key flows with certain probability. The algorithm is self-adaptive, dynamically changing routes according to current traffic distribution so that congestion is avoided effectively. Because probability is introduced to select routes, this algorithm solves the interference problem, which is described in detail in Sect. 4.1, when multiple flows are rerouted simultaneously.

3 Modeling

3.1 Network model

We model the inner-autonomous system (inner-AS) topology as an undirected graph $G(V,E)$. V is the set of nodes, and E represents the links. An s - t path P is made up of unrepeatd links, in which s and t are source and destination of P , respectively.

The length of $(u,v) \in E$ is denoted by $l_{u,v}$. The length of path P , say l_P , is sum of those of all links composing it. That is,

$$l_P = \sum_{(u,v) \in P} l_{u,v}. \quad (1)$$

Besides, the capacity of link (u,v) is $c_{u,v}$, and the traffic volume passing through (u,v) is $b_{u,v}$. Also, we denote the key flows to be adjusted as F_i , with size f_i . The final routes calculated for key flows are represented by P_{F_i} , in which $1 \leq i \leq W$.

Table 1 lists all of the symbols in the paper.

3.2 Problem description

The distribution of network traffic is uneven. While some links are being congested, others may be idle. So we can relieve congestion by changing routes for some traffic when congestion emerges. In this way, the performance of congestion links is enhanced on one hand, and on the other, the utilization rate and throughput of networks are improved. The flow size of Internet satisfies heavy-tail distribution [21–23], that is, the majority of traffic is dominated by a few large flows. Data from Ref. [21] show that 90% of the whole traffic is made up of those large flows, which account for 9% of the number of flows. Therefore, when congestion arises, it is enough to pick up some large aggregate flows passing through congestion links and recalculate their routes to detour. With this method, the congestion of network could be effectively

Table 1 Symbolic table

symbol	description
V	set of nodes
E	set of links
W	number of key flows
F_i	key flows to be adjusted
T_{F_i}	upper bound of path length of F_i
f_i	size of F_i
s_{F_i}	source of F_i
t_{F_i}	destination of F_i
$w_{F_i}^{u,v}$	link weight of (u,v) when calculating route for F_i
$l_{u,v}$	length of (u,v)
l_P	length of P
$c_{u,v}$	capacity of (u,v)
$b_{u,v}$	existing traffic volume of (u,v)
P_{F_i}	final route for F_i
d_u^v	shortest distance between u and v

relieved. We call these large flows to be routed key flows. Large flows in backbone networks are capable of being identified by present research work [24–26], so this paper assumes key flows are known and takes only the process of calculating routes into account.

Since congestion drastically degrades the quality of transmission, networks are required to adjust promptly when it happens. On one hand, it is necessary to make traffic distribution even to relieve current congestion and decrease the probability of a future one. On the other hand, QoS to end users has to be taken into account as well. The metrics of QoS include delay, jitter, and packet loss rate. If congestion is well controlled, the impact of jitter and packet loss can be ignored. Therefore, we may think QoS is mainly determined by delay. The larger the delay is, the worse the QoS is. End-to-end delay includes three parts, which are queuing delay, sending delay, and transmission delay. The speed of interfaces of routers in backbone networks are extremely fast, so sending delay can be ignored. When congestion is cleared up, end-to-end delay is mainly composed of transmission delay, which is proportional to path length. Hence, it is necessary to confine the path length of key flows to some extent when adjusting routes.

The goal of rerouting key flows is to improve the efficiency of networks. But QoS to end users has to be taken into account as well. In order to decrease the probability of congestion and enhance the efficiency of networks, the traffic ought to be distributed evenly. Based on experience obtained from real networks, Ref. [9] proposes a cost function $\theta(b_{u,v}, c_{u,v})$ about bandwidth utilization rate. The overall cost function of the network is the sum of those of links, that is,

$$\text{overall cost} = \sum_{(u,v) \in E} \theta(b_{u,v}, c_{u,v}). \quad (2)$$

In this paper, we regard the above cost function as evaluation criterion on whether the distribution of traffic is balanced. The definition of θ in Ref. [9] is as follows:

$$\theta(b_{u,v}, c_{u,v}) = \begin{cases} b_{u,v}, & \frac{b_{u,v}}{c_{u,v}} \leq \frac{1}{3}, \\ 3b_{u,v} - \frac{2}{3}c_{u,v}, & \frac{1}{3} \leq \frac{b_{u,v}}{c_{u,v}} \leq \frac{2}{3}, \\ 10b_{u,v} - \frac{16}{3}c_{u,v}, & \frac{2}{3} \leq \frac{b_{u,v}}{c_{u,v}} \leq \frac{9}{10}, \\ 70b_{u,v} - \frac{178}{3}c_{u,v}, & \frac{9}{10} \leq \frac{b_{u,v}}{c_{u,v}} \leq 1, \\ 500b_{u,v} - \frac{1468}{3}c_{u,v}, & 1 \leq \frac{b_{u,v}}{c_{u,v}} \leq \frac{11}{10}, \\ 5000b_{u,v} - \frac{16318}{3}c_{u,v}, & \frac{11}{10} \leq \frac{b_{u,v}}{c_{u,v}}. \end{cases} \quad (3)$$

The function curve is shown as Fig. 1.

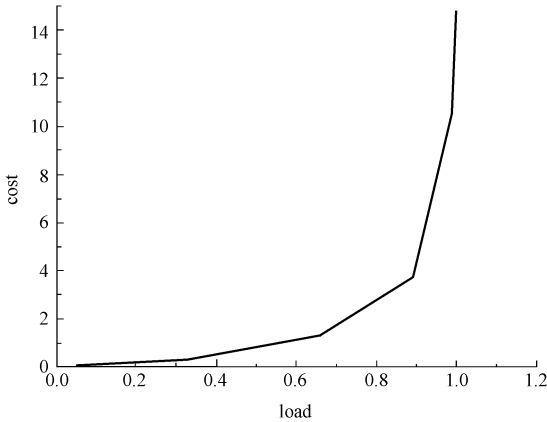


Fig. 1 Curve of cost function

From Fig. 1, we can obtain that θ is a piecewise linear convex function. The higher the bandwidth utilization rate is, the faster the function value increases. When it approaches or exceeds 100%, the impact of the corresponding link on overall cost function becomes extremely large. The large value of θ indicates that seriously congested links exist in the network. Therefore, links with high utilization rate should be avoided to make the value of overall cost function small. In addition, to

guarantee the QoS of key flows, we set an upper bound T_{F_i} for path length of each key flow F_i . The length of final path of F_i cannot exceed this bound, that is,

$$l_{P_{F_i}} \leq T_{F_i}, \quad 1 \leq i \leq W. \quad (4)$$

In summary, the problem solved in this paper is represented by the following problem:

$$\text{Min} \sum_{(u,v) \in E} \theta(b_{u,v}, c_{u,v}), \quad (5)$$

subject to

$$l_{P_{F_i}} \leq T_{F_i}, \quad 1 \leq i \leq W. \quad (6)$$

The unsplittable multi-commodity problem is NP-hard [17]. And the above problem is even more complicated than that. So it is also NP-hard, and there is no accurate algorithm with polynomial complexity for it. In Sect. 4, a self-adaptive routing selection algorithm is proposed, which makes the value of overall cost function small while strictly limiting the path length of key flows.

4 Algorithm

In this section, a self-adaptive routing selection algorithm is proposed for the sake of routing adjustment of key flows. Distributed algorithms are not easy to implement and need cooperation of many nodes. So a centralized algorithm is adopted in this paper, which calculates new routes for key flows by a centralized RS. As mentioned above, the number of key flows is small, so there is no need to worry about the processing capacity of RS.

4.1 Interference problem of key flows

With common methods, the order of calculation makes final results different.

As Fig. 2 shows, we assume that there are three key flows in the system to be rerouted, F_1 , F_2 , and F_3 . Their sources and destinations are (1,5), (6,9), and (10,11), respectively. If RS decides to calculate route for F_1 first and takes the path 1-7-8-5 as the optimal, F_2 and F_3 will be left no choice but to take 6-7-8-9 and 10-7-8-11. In this case, the traffic through link (7,8) has a large size, which leads to a high bandwidth utilization rate. So the link (7,8) becomes a congestion point, or bottleneck. Reference [27] proposes a minimum interference routing algorithm (MIRA) to solve this problem. But this algorithm requires that residual bandwidth of all links in the network must be equal. In this paper, it is key flows that are to be rerouted. MIRA cannot be applied to this scenario, except when the available bandwidth of all links is the same with key flows subtracted from the network. But this is unlikely to happen.

4.2 Routing algorithm

In this subsection, a self-adaptive probability-based routing algorithm is proposed with which RS calculates routes for key flows in turn. Every next-hop link is chosen from adjacent links of current node with certain probability. And this probability is relevant to the usage of the corresponding link. Take Fig. 2, for example, RS calculates route for F_1 first. The route starts from node 1. Then it selects (1,2) and (1,7) as next-hop links of F_1 with probability P_1 and P_2 , respectively, according to the usage of (1,2) and (1,7). Whether or not the route of F_1 is calculated first, the probability that three flows pass through link (7,8) is P_2 . There is no need for this method to deliberately consider the order of calculation. The simulation in Sect. 5 proves that this method has a great effect on load balancing.

Each calculation of RS for a key flow starts from the source node. And then the cost function θ of adjacent valid links is counted. Afterwards the next-hop link is selected with a probability inversely proportional to θ . Then the next-hop node becomes the current one, and the above process repeats until, finally, the destination node is reached. What we call valid links refer to those next-hop links that can possibly satisfy the length constraint, say, the constraint of Eq. (2). To avoid loops in the routing process, we adopt the method of Ref. [7], which selects next hop by the principle that the distance to destination should be descending.

Figure 3 is the simulation topology in Ref. [27] and is also an important experimental topology in the research on traffic and routing. The bold lines represent links with large capacity, and the light ones represent those with small capacity. With this topology, we will illustrate the process of our algorithm. For the convenience of narration, the length of each link is set to one unit. We assume that one of the key flows to be rerouted is F , with source 3 and destination 12, and the length of final path must not exceed 1.5 times that of the shortest path. The distance between 3 and 12 is 3, so the length of final path of flow F cannot be larger than 4.5. The computing process is as follows.

The route starts from node 3. Node 3 has five adjacent links, which are (3,1), (3,2), (3,4), (3,6), and (3,7), respectively. The distance between node 3 and 12 is

$$d_3^{12} = 3,$$

and

$$d_1^{12} = d_7^{12} = 3 = d_3^{12},$$

$$d_4^{12} = 4 > d_3^{12}.$$

So none of the distances of nodes 1, 7, and 4 to the destination node 12 is smaller than that between nodes 3 and 12. According to the descending distance principle, none of the links (3,1), (3,4), and (3,7) can be picked up as

next hop of node 3. But

$$d_2^{12} = d_6^{12} = 2 < d_3^{12},$$

and none of them will bring routing loops. Meanwhile,

$$d_3^2 + d_2^{12} = d_3^6 + d_6^{12} = 3 < 4.5.$$

So (3,2) and (3,6) are all valid links. If any of them is selected as next hop of node 3, there is at least one path that satisfies the total length demand. Therefore, we select (3,2) or (3,6) as next hop with the probability

$$\frac{1/\theta(b_{3,2},c_{3,2})}{1/\theta(b_{3,2},c_{3,2}) + 1/\theta(b_{3,6},c_{3,6})}$$

and

$$\frac{1/\theta(b_{3,6},c_{3,6})}{1/\theta(b_{3,2},c_{3,2}) + 1/\theta(b_{3,6},c_{3,6})},$$

respectively. And then the selected one is taken as current node, and the above process repeats until, finally, a fit route for F is acquired.

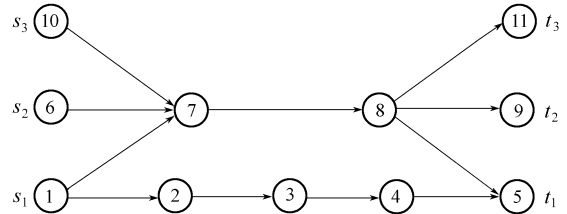


Fig. 2 Key flows routing

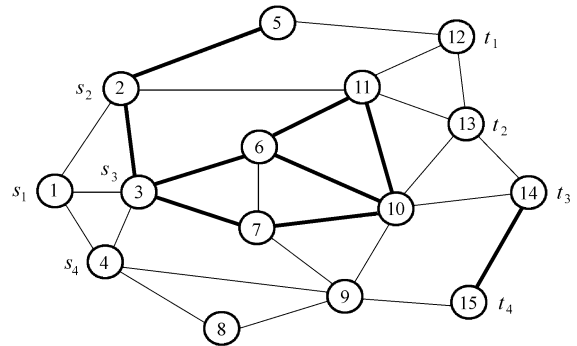


Fig. 3 Topology of MIRA net

Disadvantages still exist in the above process. If RS picks up node 2 as next hop of node 3, it will go on selecting node 5 as next hop of node 2, because the bandwidth utilization rate of link (2,5) is low. Then even if (5,12) has been seriously congested, RS has no choice but to take node 5 as next hop, because for node 5, there is only one valid link, i.e., (5,12). No matter what the usage of (2,11) and (11,12) is, RS cannot take 3-2-11-12 as the final route of flow F , but 3-2-5-12 instead. In fact, there is only

one valid link for node 5, that is (5,12). Link (5,12) is also selected at the same time when (2,5) is selected. Therefore, when calculating next hop of node 2, we cannot merely take the usage of adjacent link (2,5) of node 2 into account. Instead, (2,5) and (5,12) ought to be regarded as a whole, which is named as a valid segment. The valid segment mentioned here refers to a path composed of some links end to end. Each of the links is the unique valid link when its start node is the current one. When selecting the next-hop link, the probability is not determined by valid links but by valid segments instead.

Now we will describe the whole algorithm.

Firstly, initialize the variables and set u as the current node of flow F . Secondly, calculate the weights for all valid segments of u . The weight of a valid segment equals the largest cost function value θ of all the links belonging to the segment. Thirdly, choose a neighbor link as the next hop link of u , with a probability inversely proportional to the corresponding weight in step two. Then repeat this step until, at last, the destination node arrives. At this time, the procedure for flow F is finished, and the same one for other flows will continue.

The algorithm proposed in this paper is a self-adaptive global routing adjustment algorithm. Monitors at different places in networks send real-time usage reports of all the links to RS periodically. Once congestion arises, RS makes a decision about how to change routes for key flows to relieve congestion according to the usage reports collected. Finally, the designated routes of key flows can be achieved in several ways, such as MPLS tunnel, explicit routing, etc. In next section, we validate this algorithm by simulation method.

5 Simulation

In the simulation of this section, we generate two topologies by the tool GT-ITM (<http://www.cc.gatech.edu/projects/gtitm/>). The two topologies include 50 and 80 nodes, respectively. The former includes 298 links, and the latter includes 658 links. Each topology is partitioned by two layers. Nodes of the upper layer represent autonomous systems (ASs). Every upper node is connected with some lower ones, which represent routers within one AS. The capacity of inner AS links is 1200, and that of inter AS links is 4800. The length of links is randomly generated by GT-ITM.

The Valiant algorithm in Ref. [19] and its improvement in Ref. [20], which is called Valiant+ in the following, have a good effect on load balancing. And the similarity with our algorithm is that they adopt random method in the process of routing as well. So the simulation in this paper chooses these two algorithms as contrasts. During the simulation, we continuously add new flows to the network until 10000 units of traffic are contained. We compare the cost function values of the three algorithms at different

times. To intuitively investigate the distance between these algorithms and the optimal one, optimality gap is adopted as a metric. Optimality gap is a common concept in algorithm analysis to describe the distance between algorithms to be observed and the optimal algorithm in theory. In this paper it is the ratio of the cost function values of algorithms to that of the optimal solution, which is obtained by Lingo (<http://www.lindo.com/>). In our new algorithm which is denoted by proposed, the path length of key flows is limited to within 1.5 times the length of the shortest path for QoS requirement.

Figures 4 and 5 are optimality gap curves obtained by the three algorithms with two topologies. For the convenience of observation, Figures 6 and 7 enlarge the results of our new algorithm. Valiant and Valiant+ have close result in the 50 node topology. When traffic volume increases to 4500 units, the optimality gap begins to increase rapidly. This indicates the emergence of congestion. In the 80 node topology, Valiant+ surpasses Valiant. With the former, congestion appears at 6500. But with the latter, congestion emerges at 3500. On the contrary, the key flow routing adjustment algorithm proposed in this paper maintains a low optimality gap during the simulation, which means congestion never happens.

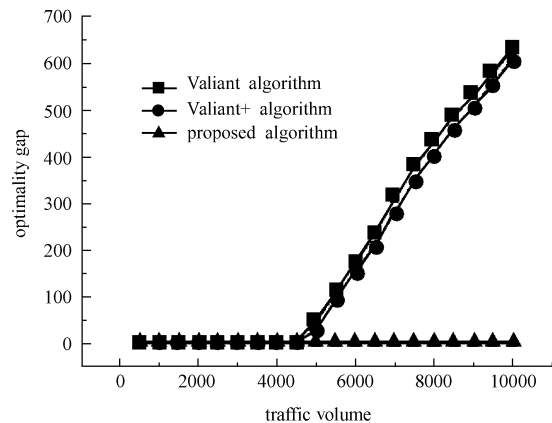


Fig. 4 Optimality gap with 50 node topology

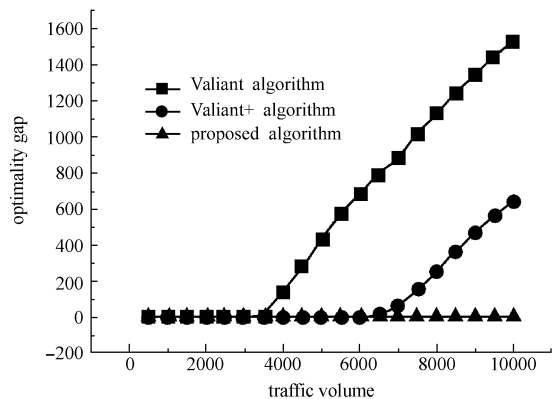


Fig. 5 Optimality gap with 80 node topology

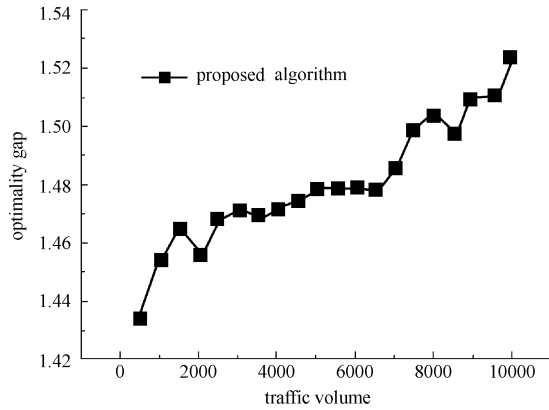


Fig. 6 Optimality gap with 50 node topology (zoom in for proposed algorithm)

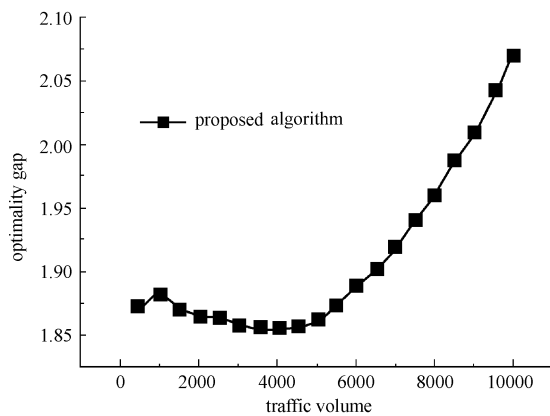


Fig. 7 Optimality gap with 80 node topology (zoom in for proposed algorithm)

Next we investigate the relationship of path length among these three algorithms. Along with the simulation, we calculate the arithmetic mean path length of flows having entered the network. Figures 8 and 9 are the results. From the figures we obtain that the mean path length of our algorithm is a little larger than that of the other two algorithms. We think this result is acceptable, because control of path length and load balancing are radically contradictory. With respect to the great disadvantages in load balancing and throughput enhancing, the larger part of path length can be ignored. And the most important is that the new algorithm strictly confines the path length of all flows to a certain range, which cannot be achieved by the other two algorithms.

6 Conclusion

Traffic distribution, link failure, and routing change may lead to congestion. When congestion arises, some links

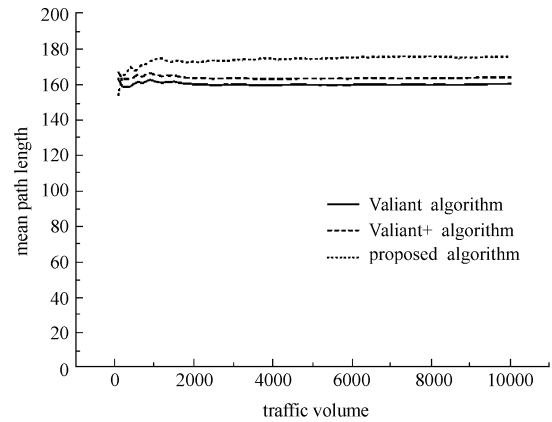


Fig. 8 Mean path length with 50 node topology

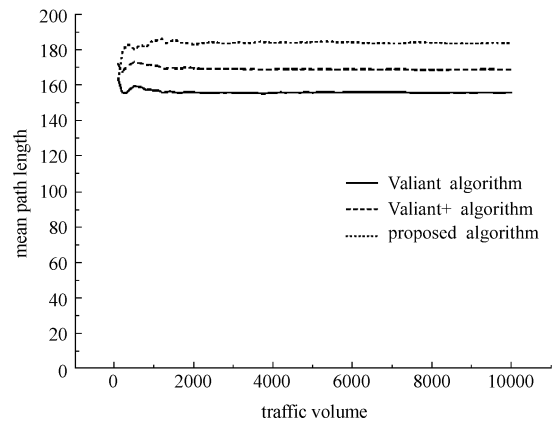


Fig. 9 Mean path length with 80 node topology

lose all or part of their transmission capacity while, at that time, other links may be idle. Therefore, it is necessary to select new routes for traffic passing through congestion points in order to make best use of idle links. This makes the traffic distribute rationally. Most present research considers only single-point congestion and pays little attention to multi-point congestion.

This paper proposes a global key flow routing adjustment algorithm, which is based on probability to select next hop. The algorithm is executed by a global routing server in the network. And the results can be implemented in practice. Compared with traditional routing adjustment algorithms, the new one not only solves the interference problem when several flows are being adjusted but also guarantees QoS by confining the path length of key flows. By the validation of simulation, the new algorithm effectively enhances the throughput of networks and decreases the probability of congestion, with QoS guaranteed.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 90604019 and 60502037), and the

National High Technology Research and Development Program of China (Nos. 2006AA01Z235 and 2007AA01Z206), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 200800131019), and Program for New Century Excellent Talents in University.

References

- Broido A, Hyun Y, Gao R, Claffy K C. Their share: diversity and disparity in IP traffic. In: Proceedings of Passive and Active Network Measurement. Lecture Notes in Computer Science. Berlin: Springer, 2004, 3015: 113–125
- Willinger W, Alderson D, Li L. A pragmatic approach to dealing with high-variability in network measurements. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. New York: ACM, 2004, 88–100
- Markopoulou A, Iannaccone G, Bhattacharyya S, Chuah C N, Diot C. Characterization of failures in an IP backbone. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies. Hong Kong: IEEE, 2004, 4: 2307–2317
- Kvalbein A, Hansen A F, Cicic T, Gjessing S, Lysne O. Fast IP network recovery using multiple routing configurations. In: Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies. Barcelona: IEEE, 2006, 1–11
- Hu N, Li L, Mao Z M, Steenkiste P, Wang J. A measurement study of Internet bottlenecks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Barcelona: IEEE, 2005, 3: 1689–1700
- Teixeira R, Duffield N, Rexford J, Roughan M. Traffic matrix reloaded: impact of routing changes. In: Proceedings of Passive and Active Network Measurement. Lecture Notes in Computer Science. Berlin: Springer, 2005, 3431: 251–264
- Yang X, Wetherall D. Source selectable path diversity via routing deflections. In: Proceedings of the 2006 SIGCOMM Conference. New York: ACM, 2006, 159–170
- Iyer S, Bhattacharyya S, Taft N, Diot C. An approach to alleviate link overload as observed on an IP backbone. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies. San Francisco: IEEE, 2003, 1: 406–416
- Fortz B, Thorup M. Internet traffic engineering by optimizing OSPF weights. In: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Tel-Aviv: IEEE, 2000, 2: 519–528
- Ericsson M, Resende M G C, Pardalos P M. A genetic algorithm for the weight setting problem in OSPF routing. *Journal of Combinatorial Optimization*, 2002, 6(3): 299–333
- Xu D, Chen Y, Xiong Y, Qiao C, He X. On finding disjoint paths in single and dual link cost networks. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies. Hong Kong: IEEE, 2004, 1: 705–715
- Orda A, Spronston A. Efficient algorithms for computing disjoint QoS paths. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies. Hong Kong: IEEE, 2004, 1: 727–738
- Baier G, Köhler E, Skutella M. The k-splittable flow problem. *Algorithmica*, 2005, 42(3–4): 231–248
- Benko P, Veres A. A passive method for estimating end-to-end TCP packet loss. In: Proceedings of IEEE Global Telecommunications Conference 2002. Taipei: IEEE, 2002, 3: 2609–2613
- Jaiswal S, Iannaccone G, Diot C, Kurose J, Towsley D. Measurement and classification of out-of-sequence packets in a Tier-1 IP backbone. *IEEE/ACM Transactions on Networking*, 2007, 15(1): 54–66
- Zhou X, Van Mieghem P. Reordering of IP packets in Internet. In: Proceedings of Passive and Active Network Measurement. Lecture Notes in Computer Science. Berlin: Springer, 2004, 3015: 237–246
- Karp R M. Reducibility among combinatorial problems. In: Miller R E, Thatcher J W, eds. *Complexity of Computer Computations*. New York: Plenum Press, 1972
- Azar Y, Regev O. Strongly polynomial algorithms for the unsplittable flow problem. In: Proceedings of Integer Programming and Combinatorial Optimization. Lecture Notes in Computer Science. Berlin: Springer, 2001, 2081: 15–29
- Valiant L G. A scheme for fast parallel communication. *SIAM Journal on Computing*, 1982, 11(2): 350–361
- Shepherd F B, Winzer P J. Selective randomized load balancing and mesh networks with changing demands. *Journal of Optical Networking*, 2006, 5(5): 320–339
- Fang W, Peterson L. Inter-AS traffic patterns and their implications. In: Proceedings of Global Telecommunications Conference 1999. Rio de Janeiro: IEEE, 1999, 3: 1859–1868
- Feldmann A, Greenberg A, Lund C, Reingold N, Rexford J, True F. Deriving traffic demands for operational IP networks: methodology and experience. *IEEE/ACM Transactions on Networking*, 2001, 9(3): 265–279
- Zhang Y, Breslau L, Paxson V, Shenker S. On the characteristics and origins of Internet flow rates. In: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM, 2002, 309–322
- Estan C, Varghese G. New directions in traffic measurement and accounting. In: Proceedings of the 2002 SIGCOMM Conference. New York: ACM, 2002, 323–336
- Smitha, Kim I, Narasimha Reddy A L. Identifying long-term high-bandwidth flows at a router. In: Proceedings of High Performance Computing (HiPC 2001). Lecture Notes in Computer Science. Berlin: Springer, 2001, 2228: 361–371
- Mori T, Uchida M, Kawahara R, Pan J, Goto S. Identifying elephant flows through periodically sampled packets. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. New York: ACM, 2004, 115–120
- Kar K, Kodialam M, Lakshman T V. Minimum Interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications. *IEEE Journal on Selected Areas in Communications*, 2000, 18(12): 2566–2579