On-the-Fly TCP Path Selection Algorithm in Access Link Load Balancing

Ying-Dar Lin, Shih-Chiang Tsao, and Un-Pio Leong
Department of Computer and Information Science
National ChiaoTung University
HsinChu, Taiwan
Email: {ydlin, weafon}@cis.nctu.edu.tw

Abstract—**Many enterprises install multiple access links for fault tolerance or bandwidth enlargement. Dispatching connections through good links is the ultimate goal in utilizing multiple access links. The traditional dispatching method is only based on the condition of the access links to ISPs. It may achieve fair utilization on the access links but poor performance on connection throughput. In this work, we propose a new approach to maximize the per-connection end-to-end throughput by the on-the-fly round trip time (RTT) probing mechanism. The RTTs through all possible links are probed by duplicating the SYN packet during the three way handshaking stage of a TCP connection. Combined with the statistical packet loss ratio and the passively collected link metrics, our algorithm can real-time select a link which provides the maximum throughput for the TCP connection. The experiment results show that the accuracy to choose the best outgoing access link is over 71%. If the second best link is chosen, it is usually very close to the best, thus achieving over 89% of the maximum possible throughput. The average per-connection throughput for our algorithm and the traditional round-robin algorithm is 94% vs. 69%.**

Keywords-- Load balancing; RTT; TCP three-way handshaking

I. INTRODUCTION

Today, broadband network access, such as xDSL service, is widely deployed for home users as well as small and medium enterprises to connect the Internet. However, these users still thirst for more scalable and reliable access bandwidth. One of the practicable ways to satisfy the thirst is renting multiple WAN access links. The way is more affordable and fault-tolerant than renting a single high-capacity link. It guarantees uninterrupted Internet access, provides high availability across multiple ISP links, and maintains compatibility and scalability. Such a practicable way has been applied in real life several years. The customers prefer renting multiple cheap ADSL links than an expensive leased line. To enjoy the advantage of renting multiple links, a device, usually called load balancer, is necessary to intelligently select a link from multiple ISP links for an establishing connection. The load balancer discussed in this work focuses on the outbound traffic handling [1], which is different with the schemes in multi-homing [2] problem for handling the inbound traffic. Generally speaking, the link with the lowest utilization is preferred. It is believed that this link can provide the new connection the most available bandwidth. In the long term such a bias leads the device to balance the loading between links. That is why such a device is called load balancer.

However, the remote target site of a connection is usually far from the local host. The access link is only the one of the many links composed the end-to-end path. Thus, selecting the link with lowest utilization does not present providing the connection the path with the most available bandwidth when the access link is not the bottleneck of the connection. For example, as a user resided in Taiwan means to surf a website located in Germany, the device should select the access link which is the head of the links composed the *fastest* path to Germany. Unfortunately, traditional load balancer only spends effort on determining the link with the lowest loading, but not the path with the most available bandwidth to the destination. Obviously, the traditional balancer strays from the aim of their duty. A few load balancers aware the stray and mean to collect more path information to improve the link selection. They retrieve the information beyond the next hop by relying on SNMP or ICMP. However, the retrieve by SNMP requires authorization, while the ICMP packet is often filtered out for security concern.

This work aims to select an access link providing the optimal path for a TCP connection, where the optimal path indicates the path providing the most available bandwidth for a TCP connection since the objective of most TCP connections is still transmitting files or web pages as soon as possible. According to the TCP throughput model [3], the mean rate of a TCP connection is determined on the packet loss ratio and RTT of a path. That is say the optimal path for a TCP connection is the path with the lowest loss ratio and the shortest RTT. The loss ratio in the Internet is variant and hard to estimate upon establishing a connection. In fact, it is difficult even when the connection is running. On the contrary, by the observation in Section II, the RTT during a connection is stable. Thus, a novel on-the-fly RTT probe strategy is proposed in Section III and implemented into the NetBSD [4] kernel. By the probe strategy, the load balancer can select a link providing the shortest-RTT path to the destination. The strategy probes the RTT upon establishing a connection. The TCP SYN segment of this establishing connection is employed as the probing packet. It is duplicated and sent concurrently along all access links, and then the link where the SYN/ACK segment the most early returns presents the path following it owns the shortest RTT. Such an *on-the-fly RTT probing* strategy, therefore, provides a TCP connection the optimal path under the assumption that the access WAN links are equal in the loss ratio although the assumption is not true in the real world. Such an assumption sometimes leads to the inaccuracy of the best selection, carefully examined in the numerical evaluation in Section IV.C.

In the rest of this work, the organization is as follows. Section II reveals the problem of the traditional load balancer. Section III describes our on-the-fly TCP path selection algorithm and the concern of the implementation into the NetBSD [4] kernel. The performance evaluation in Section IV shows how well this link load balancing approach behaves. Finally, Section V wraps up with the conclusion and future works.

II. Problems in Traditional Link Selection

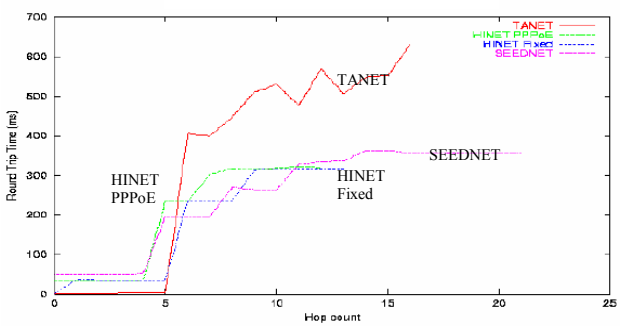
Traditional link selection algorithms [5,6] prefer selecting the link with lowest loading for establishing a new connection. Weighted round-robin selecting from links is another simple way, balancing the loading between links; thus the next selection just is the lowest one if the throughput of all connections are assumed equal. Generally speaking, the policy to select a link with the lowest loading is easy to implement. However, the link selected by such a policy does not ensure providing a path with the most available bandwidth to the remote site when the bottleneck is not the access link.

New policies further collect or probe the information beyond the next hop to select a link which is the head of the links composed the optimal path. The following exposes the problems in these policies in terms of *used protocol*, *collection distance*, and *measured moment*.

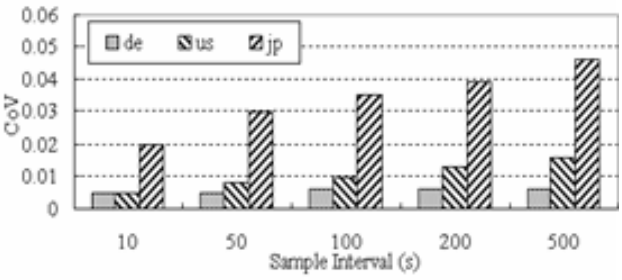
Used Protocol: SNMP and ICMP are popular protocols to collect the condition on path [7]. Unfortunately, because ICMP is used for hacking and probing security holes, today most firewalls filter out ICMP packets for security concern. Further, even without filtering them, ICMP packets may gain a low priority to forwarded or replied, affecting the accuracy of the measurement. SNMP is another protocol for collection. By it, the load balancer can directly retrieve the statistic information of the hops on the path. However, accessing a hop by SNMP requires authorization for security concern. It is impossible for a local load balancer to own the authorization to access all hops between the local host and the destination.

Collection Distance: The traditional load balancing algorithms fail for only collecting the condition of the access link. The ideal distance of the collection should cover the overall path. Fig. 1(a) shows the RTT probing results of each hop in four connection paths provided by four WAN links rented from TANET [8], HINET [9] (PPPoE), HINET (Fixed IP), and SEEDNET [10], respectively. Fig. 1(a) exposes the inaccuracy selection if only probing the near end hops. The whole path is broken down into a sequence of hops and the time spent between hops is displayed. Obviously, if only considering the RTT of the near hops and given that the packet loss ratio is equivalent between all links, the access link of TANET is the best link for a TCP connection, which makes the TCP connection send packets with a highest rate. However, once considering the RTT of the end-to-end path, the path through the TANET link has the largest RTT and provides the worst transmitting throughput for a TCP connection.

Measured Moment: If considering collecting the condition of the whole path, the only moment to measure is upon establishing a connection, because the destination of a connection is unexpected and given at real time. It is



(a). The pitfall between link selection and path selection



(b) The averaged CoV of RTT in different sampling intervals

Figure 1. The observation on RTT of each link and multiple paths.

impossible to collect *in advance* the conditions of the paths to each destination in the Internet, which is another reason, besides authorization, limits tradition off-line collections to retrieve the condition from near hops.

III. On-the-Fly TCP Path Selection (OFTPS) Algorithm

The OFTPS proposed in this work selects the link heading the path that may provide the most available bandwidth for a TCP connection. When the access link is not the bottleneck, the link heading the path with the shortest RTT is determined by the on-the-fly RTT probing mechanism. On the other hand, as detecting the access link is the bottleneck, the OFTPS degrades to the link selection algorithm and simply selects the link with the most available bandwidth.

A. RTT is short-time stable

The following reveals the RTT is short-time stable so as to probe on-the-fly. The RTT in the Internet is dynamic but not violent on a path as shown in Fig. 1(b). The RTTs between the 3 websites and local are measured through the three links in 4000s. The result reveals the largest CoV is only 0.045 even in a time scale of 500s. That is, the value of RTT shakes fall in a 4.5% range of the mean. Obviously, the RTT of the whole session is stable. Based on such a stable character, it is believed that average RTT can be represented by only the value averaged from a few samples. Some works [11,12] further stand for its practicability.

B. On-the-Fly RTT Probing mechanism

This subsection proposes how to on-the-fly probing the RTT of multiple paths. Employing the three way handshaking in TCP, the necessary procedure of establishing a TCP connection, is the most efficient and real-time solution to retrieve the RTT on one path. The RTT equals to the

difference on timestamp between SYN and SYN/ACK packets. To probe multiple paths through from multiple WAN links simultaneously, the right side of the flowchart in Fig. 2(a) illustrates the steps. First, the WAN load balancer (WLB) duplicates the SYN packet and then sends out these copies via different WAN links, respectively, as shown in Fig. 2(b). Each returned SYN/ACK packet presents the probing results of each path status via each WAN links, respectively. The SYN/ACK packet returning the earliest represents the corresponding path has the shortest RTT. Then, the link heading the path is selected. An ACK packet is replied through the link to establish the TCP connection while RST packets are sent via other links to close out the probing connection. Finally, all the duplicated handshaking are closed, and the selected connection behaves like a normal connection. The overhead of such an on-the-fly RTT probing is that the duplication of the SYN packet. It increases the load of the destination server's accepting state, but no overhead for the rest of data transmission. Besides, the useless connections are terminated as soon as deciding the shortest one by sending the RST packets.

C. Degrade to the Link Selection

Although the path with short RTT may provide larger bandwidth for a TCP connection, the connection cannot obtain the bandwidth if such a bandwidth is unavailable on the access link. In other words, it is necessary to distinguish the bottleneck between link bounded bandwidth and TCP bounded bandwidth. If the throughput is bounded by the access link, it is better to choose the link selection over the path selection algorithm. By the computation of the possible consumption of bandwidth, it can be determined where the bottleneck resides. As shown in Fig. 2(a), if the selected link by the path selection algorithm has insufficient bandwidth, then link selection mechanism is used. By using this threshold-based decision, it nearly guarantees a better WAN link load balancing algorithm than the conventional usage. Notable there is an extreme case that all the access links are provided by a single ISP, where the path conditions through these links could be similar. It causes the path selection algorithm to degrade to a link selection algorithm. However, such extreme case is not a suggested architecture for renting multiple WAN links. It gives up the fault tolerance property originally embedded in the using of multiple WAN links. Regarding the connection carried over other protocols, such as UDP, the WLB also degrade to the link selection algorithm as shown in the top-left of Fig. 2(a).

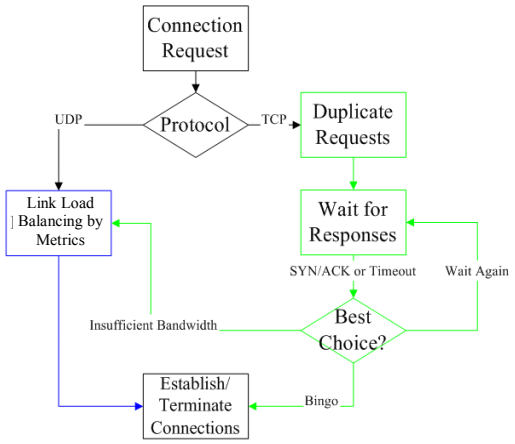


Figure 2(a). On-the-Fly Path Selection Algorithm

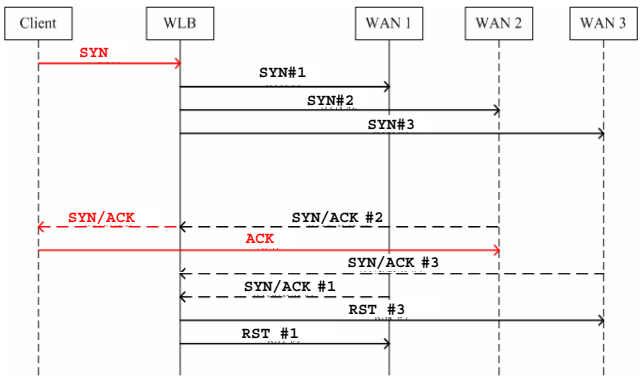


Figure 2(b). Mechanism of SYN/ACK RTT probing selection

D. Packet Loss Ratio

In fact, both RTT and packet loss ratio are necessary to estimate the bandwidth occupied by a TCP connection on a path [3]. Unfortunately, the packet loss ratio in the Internet is vary and hard to gather a stable estimation even by a long-term measurement. In other words, for one establishing connection, it is impossible in the real time to obtain the loss ratio of the link. Thus, here we assume the access WAN links are equal in the loss ratio although it is not true in the real world. Such assumption implies the RTT is the only determinant in the best TCP path selection of the OFTPS. In fact, it could lead to the inaccuracy of the selection, which was examined in our numerical evaluation.

E. Implementation Issues

The following discusses three concerns of the implementation of the WLB with OFTPS in the NetBSD [4] kernel.

***WHERE*:** This work implements the WLB into the Network Address Translation (NAT) module because two operations, *IP modification* and *stateful redirection*, done in the NAT are required by WLB. The WLB redirects the traffic to the selected outgoing access link; thus, the associated source IP address of outgoing data packets must be modified to the IP of the selected link. Otherwise, their ACKs cannot return from the selected link. Moreover, after selecting the particular link, the WLB should redirect the following packets in one connection to the same link. Thus, the stateful redirection is necessary for WLB.

HOW: For a received SYN packet, NAT handles three major operations, including (a) modifying the IP of the SYN (b) keeping state (c) sending out the SYN. 3 modifications are required to integrate OFTPS-based WLB into NAT. First, the SYN duplication should perform before operation (a). Second, the operation (b) is delayed till receiving the first SYN/ACK, the time WLB can know which link is selected. Third, on receiving the SYN/ACK, as stated in III.B, the WLB needs to send ACK packet through the selected link and RST packets through other links.

***Virtual Link*:** By the requirement of users, several WAN links may be integrated to a big *Virtual* WAN link. Since it is

impossible for a single connection to send packets over multiple links without the support of ISP, for such a virtual link, the maximum bandwidth provide for a single connection is bound by the bandwidth of the maximum one among all the physical links. Thus, as verifying whether the link bandwidth is sufficient for a TCP connection, the OFTPS should use the maximum bandwidth available in one of the physical links, but not the aggregated available bandwidth in the virtual link.

IV. Evaluation

A. Testbed Configuration

The section demonstrates the OFTPS algorithm can select the link heading the path with the maximum available bandwidth for a TCP connection by the experiments run in Internet. The testbed was built as shown in Fig. 3 and there are three WAN links providing the load balancer accessing Internet. When a connection is establishing, one of the three links is chosen according to the OFTPS algorithm implemented in the device. In the following tests, there are three destinations including ftp.de.freebsd.org (de), ftp.freebsd.org (us), and ftp.jp.freebsd.org (jp) which located in Germany, US, and Japan, respectively. For each destination, in one testing round we forecast the best link first by the OFTPS algorithm and then retrieve a file with 8 mega bytes through the three links, respectively. By comparing their mean throughput, we can verify whether the link selected by OFTPS is the best one or not. Each round of this test is performed repeatedly every ten minutes. At last, the result was compared to the traditional round-robin (RR) link selection algorithm.

B. The Link Selected by OFTPS algorithm

Fig. 4(a) displays the throughput results of 140 rounds transferred files from ftp.de.freebsd.org through the three WAN links, TANET, HINET PPPoE, and SEEDNET PPPoE. In the figure, for each round, the link selected by OFTPS is highlighted with the curve *WLB Selected*. Obviously, the curve always overlaps the result with the highest throughput, indicating the OFTPS can select the link providing the maximum throughput for TCP connections. The similar results are displayed in Figure 4(b) and 4(c), where the destinations are ftp.freebsd.org and ftp.jp.freebsd.org, respectively. Notable that Figure 4(a)(b)(c) also reveal the Internet status does change from time to time. According to our observation, the unusual large RTT seriously degrades the throughput provided by one links.

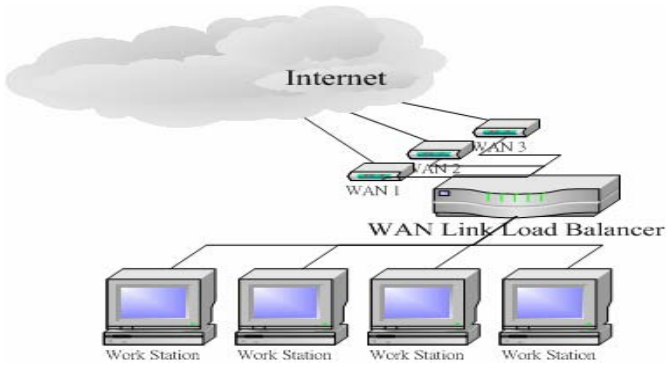


Figure 3. WAN link load balancing testbed configuration

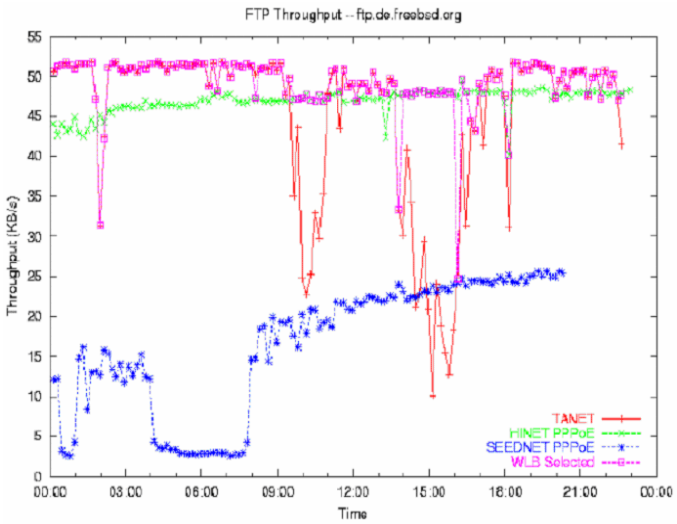


Figure 4(a). FTP Throughput – ftp.de.freebsd.org

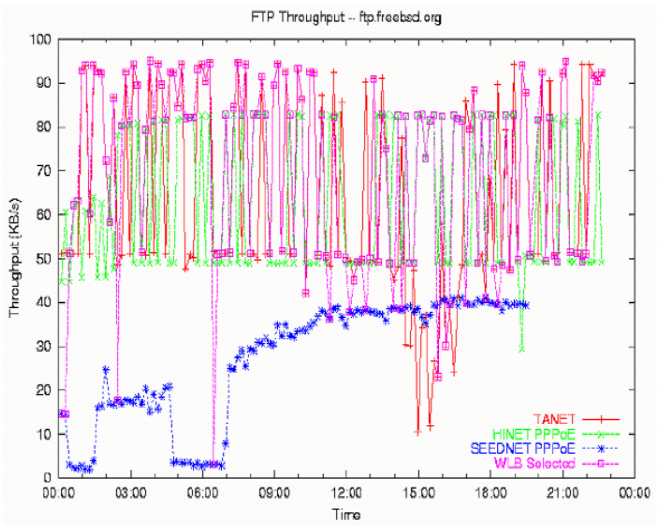


Figure 4(b). FTP Throughput – ftp.freebsd.org

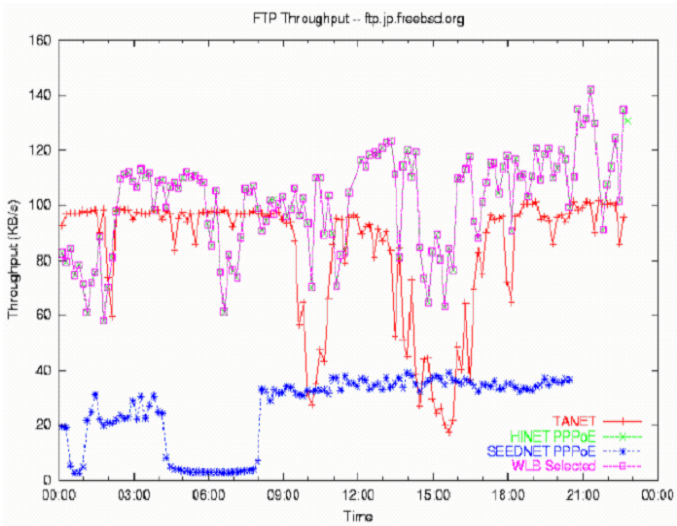


Figure 4(c). FTP Throughput – ftp.jp.freebsd.org

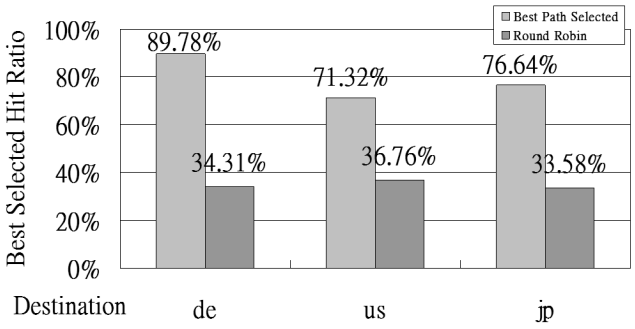


Figure 5. The hit ratio of TCP path selection algorithm

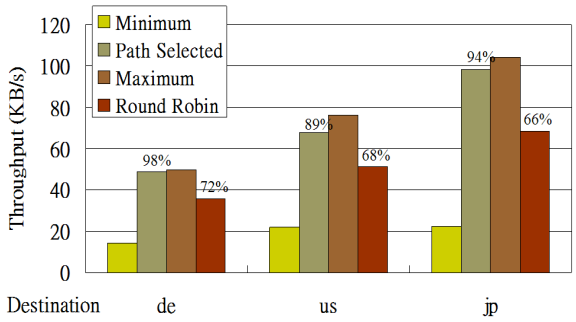


Figure 6. The selected throughput compared to the minimum and maximum possible throughput utilization

C. The Accuracy of the OFTPS algorithm

Next, for each destination, the best selection ratio of the OFTPS algorithm is averaged from 100 rounds of selecting results, where the best selection means the selected link providing the highest bandwidth for a TCP connection between the three links. Fig. 5 shows the OFTPS determines the best links for connecting to de, us, and jp with a ratio of 89.78%, 71.32%, and 76.64%, respectively. Compared to the result of the RR link selection, the OFTPS provides a higher hit ratio in the determination of the best access link.

D. The comparison between selected and maximum links

Instead of the selected hit ratio, the following reveals the link selected by OFTPS provided the bandwidth similar to the maximum bandwidth provided by one of the three links. In Fig. 6, there are four bars for each destination. The bars *Maximum* and *Minimum* indicate the mean throughput averaged from the maximum and minimum throughput in each round, respectively. The bar *Selected* and *Round-Robin* present the means averaged from the throughput provided by the links selected by the OFTPS and the RR link selection algorithm. The chart reveals the selected link to the destination *de* provide 49KBps bandwidth, which equals to 98% of the maximum bandwidth, 50KBps, the three links could provide. It is far better than the link selected by the RR, which only provides 36/50, or 72%, of the maximum bandwidth.

The result in Fig. 6 outperforms than that in Fig. 5, revealing an interesting phenomenon. That is, even on the case that the OFTPS does not correctly forecast the best link, the connection through over the link selected by the OFTPS still obtains the similar large bandwidth as one that the best path can provide. It implies for the incorrect forecasting cases, the conditions between the best link and our selected link is similar.

V. Conclusion and Future Work

The goal of link load balancing is to select a best link for establishing a connection. Besides sufficient link bandwidth, a best link should provide the best quality of network conditions over an end-to-end path. This work focuses on the best link selection for TCP connection and proposes an on-the-fly RTT probing mechanism. The mechanism uses the SYN/ACK, the three-way handshaking mechanism of TCP, to real-time retrieve RTT when a TCP connection is being established. With the RTT and the statistical packet loss ratio, we can forecast how much throughput a TCP connection could achieve when transmitting through the link. As the local link bandwidth is enough, we select the link heading the path with the highest available bandwidth for a TCP connection.

The result of our experiments on the NetBSD implementation shows that the mean throughput of connections through the link selected by our algorithm indeed can pick the best quality path to transmit packets to the destination with an average probability close to 93.7%. While the Internet status varies from time to time, we still score an average hit ratio at 79%. According to our implementation experience, the TCP best-path algorithm proposed in this work can be implemented at the gateway without modifying the client protocol. It makes the deployment easier and more compatible to existing applications. In the future, to improve the accuracy of the best TCP path selection, we will continue to study how to obtain a suitable estimation on the packet loss ratio. Besides, since the WAN link load balancing gateway requires to access information at the transport layer, we will further discuss the possible conflict if IP SEC is implemented.

References

[1] Jingsha He, “An Architecture for Wide Area Network Load Balancing” in Proc. International Conference on Communications 2000, pp.1169-73 vol.2.

[2] A. Mihailovic, G. Leijonhufvud, T. Suihko, “Providing multi-homing support in IP access networks”, the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2002, pp.540-544 vol.2.

[3] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: A simple model and its Empirical Validation,” in Proc. of ACM SIGCOMM’98, Sep 1998.

[4] The NetBSD Project, http://www.netbsd.org/

[5] Radware LinkProof, Internet Link Application Switching. http://www.radware.com/content/products /link.asp.

[6] F5 Networks BIG-IP Link Controller 2000. http://www.radware.com/ content/products/link.asp.

[7] Load Balancing, Radware Ltd., United State Patent, US006249801B1, Jun. 19, 2001

[8] TANET, http://www.edu.tw/EDU_WEB/Web/MOECC/home.htm

[9] HINET, http://www.hinet.net/english/index.htm

[10] SEEDNET, http://www.digitalunited.com

[11] F. Chatte, B. Ducourthial, S.-I. Niculescu, “Robustness issues of fluid approximations for congestion detection in best effort networks”, ISCC 2002. Seventh International Symposium on, 1-4 July 2002, pp.861-866.

[12] Tsunyi Tuan, Kihong Park, “Multiple time scale redundancy control for QoS-sensitive transport of real-time traffic”, INFOCOMM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, IEEE, 26-30 March 2000, pp. 1683-1692, vol.3.