MultiServ: Congestion Alleviation Using Overlay Network

Xinyan Zhang[2*], Gang Song[3*], Qian Zhang[1], Wenwu Zhu[1], T.S. Peter Yum[2]

[1] Wireless and Networking Group, Microsoft Research Asia, P.R.China
[2] Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong
[3] Department of Computer Science, Tsinghua University, Beijing, P. R. China

Abstract— **In this paper, a novel model named MultiServ is proposed to alleviate the congestion and to provide better quality of service for end-host using overlay network. In MultiServ, a special overlay is built so that end-host and its neighbors can cooperatively transmit data efficiently. Meanwhile, a joint congest control scheme is proposed for multiple path data transmission. As a result, the traffic in the underlying network can be balanced and smoothed and the congestion can be alleviated or avoided. This provides a promising solution for application with demand of good quality of service for throughput sensitive transmissions.**

I. Introduction

The end-to-end performance is crucial in today's Internet. Recently, more and more applications require small delay or large bandwidth for data communications. However, these requirements may not be easily satisfied. For example, streaming service, which suffers from the link congestion and server's heavy load, cannot easily be deployed in large scale. Typically, the performance of communication through congested links will be degraded. A link becomes congested when lots of transmissions go through, these transmission will be all affected. Therefore, congestion in small part of links may affect the overall network performance in large scale. This in turn will result in the poor experience of huge number of Internet users. On the other hand, reports [1] show that the utilization of Internet backbones is quite small, typically no more than 38% of link capacity. The contradiction in consumption and supply lead us to think about the fundamentals of the current status of today's Internet.

We consider the congestion of Internet is caused by the following reasons.

- Hot spot phenomena often occur in current Internet. This is the reason of unbalanced traffic in different paths which may be the main cause of congestion;
- Current dominating transport protocol in Internet, TCP, does not perform stable in large delay bandwidth production environment, which decrease the performance in global scale.
- From protocol and human behavior aspects, the traffic has natural characteristic of burst, which will fill the buffer of routers and therefore cause congestion.
- For economic and administrative reasons, ISP may set up improper routing policy, which leads the network to function abnormally or result in poor performance.

Various solutions have been proposed to solve these problems. Over-provision is the method ISPs nowadays use to deal with congestion with growing demand. However, it is expensive to estimate the traffic and deploy sufficient equipment to accommodate the future demand. From research community, improved protocols [2] [3] are proposed to exploit the available bandwidth in end-to-end link, which still need time to validate. The research results on routing policy, especially inter-domain routing protocol, the Border Gateway Protocol (BGP) [4] [5], show that in current heterogeneous environment, there is long way to go to achieve stable and efficient Internet routing.

Recently, new trends on decentralized network model have attracted Internet researchers' attentions. In this model, systems build overlay networks to provide flexibility and to enable rich services. Some attempts in this area are also tried to solve the above problems. For instance, resilient Overlay Networks (RON) [6] has been proposed to allow end-hosts and applications to cooperatively gain improved reliability and performance in the Internet in comparison with traditional routing schemes. However the maintenance cost for RON is quite high and it could not be easily extended to large scale. OverQos [7] aims to provide architecture to offer Quality-of-Service (QoS) using overlay network. Without addressing the problem of routing policy, the technique of aggregating flow in that architecture will not fundamentally solve the problem.

To solve the problem, we propose a new model named *MultiServ* that uses overlay network to alleviate the congestion and to provide better QoS to end-hosts. The target of our approach is mainly at the applications with large data transmission, such as file downloading, streaming, where throughput is a critical issue. The MultiServ model focuses on building a special overlay for efficient multiple paths transmission. In this model, a cooperatively congestion control scheme is introduced to achieve traffic balancing among all links.

The remainder of the paper is organized as follows. In section 2 we provide an overview of the model. Section 3 will introduce the model in details. Section 4 verifies the model using simulation and real experiments. In section 5 we conclude the paper and propose the future work.

[*] This work is performed while they are visiting students at Microsoft Research Asia.

0-7803-7974-8/03/$17.00 © 2003 IEEE

II. OVERVIEW

In current Internet, typically every user uses a single connection to exchange data with others. This performs well when there is no congestion in the path to the destination. However, when congestion occurred, the user has no way but to endure the longer transmission time. However, possibly there are some users who have no or less congestion in the path to the same destination. In that case, if the original user transmits through this kind of users, using them as a proxy, with the help of this kind of users, the transmission may be accelerated; meanwhile, the traffic can be delivered through other un-congested paths so that the congested path is alleviated. This is the basic concept of our MultiServ model.

To use the MultiServ model, we need cooperation of multiple users. Therefore a special overlay should be constructed to cooperatively deliver traffic. Each host in the overlay can be considered as a *sender* as well as a *receiver*. A sender has several *neighbors*, which are also the hosts selected from the overlay. During the transmission, different from traditional model, the neighbors will be responsible to deliver partial of the packets from the sender when encountering congestion. To some degree, transmission through the neighbors can be treated as the complement to the traditional end-to-end delivery.

To achieve that purpose, first we use a heuristic method to construct a special overlay network by selecting the appropriate neighbors. Keep in mind that the overlay construction will try to help alleviating the congestion and balancing the traffic in the underlying network, our algorithm selects the neighbor so that the paths from the neighbors to the destination are different.

Second, a rate-based congestion control algorithm is used to disseminate the packets for further transmission. Some of the neighbors can transfer data to the destination without congestion. Notice that the faster a neighbor can transfer, the more requests it will process. So the congestion connections will not transfer as much data as it used to do. In that sense, the pressure of congestion will be partially released. This will degrade the hot-spot on the Internet. Our algorithm is also to minimize the traffic burst by sending data using a unified rate in a smoothed manner. This will further release the pressure on router buffer.

Ordinary users can benefit from our MultiServ model. By building a large scale overlay network using the MultiServ software, the user can experience better QoS. The packets sending and receiving will go through multiple paths, where alternative paths will be used as a complement to direct connection. So the user will not experience worse than using best-effort Internet. This gives the incentive for the user to use that software. Meanwhile, ISPs will also be pleased because the traffic generated by the user will be smoothed and may not be as aggressive as before.

The model can be extended to ISP level, which illustrated in Figure 1. ISPs can use MultiServ to provide a better service scenario. Typically the congestion occurs in the edge of an ISP, this is mainly caused by unbalanced traffic to each gateway and traffic burst generate from the users. Using MultiServ, special servers can be placed in each gateway of ISP to redistribute the traffic to other gateway in terms of utilization of gateways. The servers will aggregate the traffic sending outside and deliver packets to other special servers in different ISPs by agreement with other ISPs. MultiServ can dynamically combine capacity of different paths to deliver the data to the destination efficiently with minimized congestion. During the transmission, ISP can even reserve some bandwidth or use special aggressive protocols for performance enhancement. Furthermore, the joint flow control also can flexibly adjust the priority of each flow. Therefore, an implementation of differentiated services can also be provided.

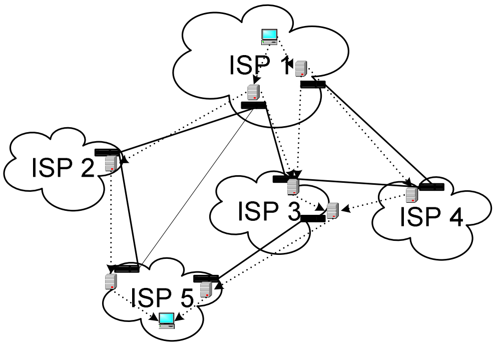

Figure 1. MultiServ illustration.

III. THE MODEL

We divided the model into two stages. First, a special overlay network is built as a platform. Second, a control algorithm is introduced to deliver the packets through the neighbors.

A. Building the overlay network

Methods have been provided to construct overlays which try to exploit the topological information, particularly locality in the underlying network. For example, application-layer multicast [8] and CDN network is of this kind, where each node keeps nearby neighbors in the underlying network in order to transfer data efficiently. In MultiServ, topological information is also a very important factor. Different from the described overlays, here the neighbors should have different links to destination host. Therefore, the nearby hosts may not be the proper neighbors to deliver the traffic. The reason is that the path from that host to the destination is equivalent or similar to the original sender. For example, the hosts which placed in the same ISP or in the same LAN should not considered as good neighbors. So, the criteria for good neighbors should be: (a) The neighbors, together with the sender itself, should connect to the Internet through different gateways; (b) The connections from the sender to neighbors should maintain good QoS. Under these assumptions, the good neighbors of a host should be in different ASes, where these ASes has good connection with the AS the sender belongs to.

The hop number between two hosts can help us estimate if the two hosts are within the same AS, generally speaking, the more hops between the hosts, the more probability that the two are in different ASes. Other techniques, such as King [9], can also estimate the distance of gateways and ASes for hosts.

Considering the large network scale, it is hard to find the most appropriate neighbors that meet the above requirements. In case that the overlay network might be a large scaled overlay with lots of hosts, such as the scale of current file sharing software, complicated algorithms are not desirable, since any bugs in the algorithm may be harmful to the whole network. So here we use a simple heuristic algorithm to select proper hosts as the neighbors, which is illustrated in Figure 2.

1. Find a random host A, put it into candidate list
2. Get a host from candidate list, named X. If candidate list is empty, goto 1.
3. Return X if X is suitable for being a neighbor or better than one current neighbor. X should have good QoS with the sender and in different AS with current neighbors and the sender.
4. Find neighbors of X, put them into candidate list.
5. Goto 2

Figure 2. Algorithm for neighbor selection.

In ideal situation, each neighbor of a host would belong to different ASes. A breadth-first order search by the above algorithm will quickly span over hosts in many different ASes. Therefore new host can find proper ASes efficiently.

During transmissions, host can rank neighbors by transmission rate and delay. Using this information, our algorithm can compare the measured distance among different neighbors for refinement. The addresses of good neighbors can also be stored for future refinement when the host needs a restart. Therefore the next selection process can be eliminated by using stored information.

The selection time for good neighbors can be different for different types of hosts. For example, some hosts may have congestion on the gateway so that almost no hosts can be a good neighbor. In this scenario, none of the methods can help the hosts to improve the QoS unless making modifications on the gateway. However, for hosts have congestion on several of its physical links to outside, large portion of hosts can be good neighbors.

We also expect to balance the neighbor number of hosts. In our scenario, one host may serve as neighbor for too many hosts. In that case, the bandwidth of that specific host would be exhausted. To solve this problem, for ordinary host, a maximum utilization of bandwidth should be defined. Meanwhile each host will maintain a host-cache storing local or nearby hosts address. When one host finds one suitable neighbor, unfortunately it is may be overloaded. The overloaded host gives the original host the nearby hosts, which has high probability to be suitable neighbors, as possible candidates. This step can speed up neighbor selection while balancing the load for hosts.

MultiServ is designed as an underlying service for hosts, so hosts should not frequently join or quit the overlay. Therefore the availability and stability of the overlay can probably be guaranteed. In delivery, one neighbor are not the only host responsible, therefore, failure of neighbors or links will not be critical. Unless all links or neighbors failed, the delivery can still use direct connection. However, our algorithm selects neighbors which are likely to be in different regions so that situation is not possible to occur.

B. *Joint congestion control*

Communication can be classified into two types, one is time sensitive, for example, VO-IP phones; another is throughput sensitive, such as streaming or file transferring. For time sensitive communications, direct connections can be used to achieve minimized delay. Our model is mainly used in throughput sensitive communications. However, since it can alleviate the congestion and the two communications share the same underlying network, it is also beneficial for time sensitive communications.

In our model, hosts may use TCP connections to communicate with neighbors. This will decrease the pressure on the congestion link. However, the following reasons prevent us from using direct TCP connections between hosts:

- The rate of different TCP connections cannot be easily controlled, so that the performance is unpredictable and uncontrollable;
- Multiple TCP connections may involve the burst of traffic so that the router buffer may be filled and deteriorate the congestion;
- Some applications, such as streaming, may not be suitable to be carried by TCP.

For these reasons, it is necessary to propose a joint congestion control model for smooth data transmission with better control.

We propose a rate-based congestion control algorithm, similar with CM [10]. The idea is to aggregate the flows sending from the host, the aggregated flow to one neighbor uses an Additive-increase multiplicative-decrease (AIMD) congestion control in order to be friendly to background TCP flows. The sending rate will increase when no packet loss. Upon a packet loss, the rate will be halved. When persistent congestion occurs, the rate drops to a small value forcing slow start to occur. An ARQ-based mechanism is used. The sender will retransmit the packet until receiving the acknowledgement.

An illustration is presented in Figure 3. A large data transmission from the sender to the receiver encountered congestion. Therefore, the sender makes connections to its neighbors to achieve better experience of transmission.

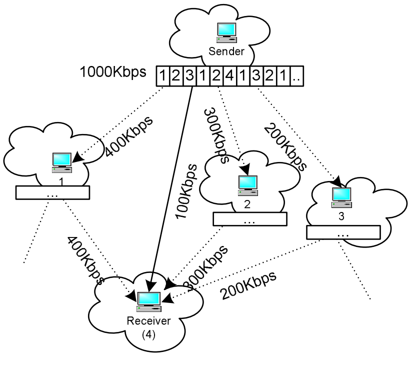

Figure 3. Example of joint congestion control.

Flows will be aggregated while sending to one neighbor. Each flow may have different transmission speed because neighbors will send out the packets to different destinations. Through aggregation, we can easily adjust the rate of flows by setting different weights in the aggregate flow, as the scheme described in [7].

To further smooth the traffic generating form host, also set up a framework for easily rate controlling, instead of making individual TCP connections with the neighbors, a unified packet sender is used. In transmission, flow to neighbor *i* will report a rate for sending packets, say r_i. The sender will use a unified sending rate of $\sum r_i$ instead of individual sending rate r_i. Flow *i* can be controlled using a weight w_i, where $w_i = r_i / \sum r$. A round robin scheduler will be used to distribute packets proportional to fit the rate. In Figure 3, suppose the rate with AIMD control for the four destinations are 400Kbps, 300Kbps, 200Kbps and 100Kbps, respectively, the rate sending out the packets for the host will be 1Mbps. In average, 4 of 10 packets will be sent to neighbor 1, 3 will send to neighbor 2, and so on and so forth.

Packets sending out from one host are controlled using unified packet sender. This enables the smoothest traffic. The interval of packets for every router in the path will be approximately equivalent if no congestion encountered.

Each host in the overlay will use that algorithm to control the packets. The intermediate host in a multiple path transfer will do some additional tasks. One is that the intermediate hosts will buffer the data from sending hosts in order to make delivery consistent. Another is that the intermediate host will feedback the delivery rate to the sender so that the sender can adjust the sending rate. Finally, the receiver should have a buffer for rearranging packets that are out of order.

Using MultiServ in ISP level, bandwidth may be reserved for the aggregate flow to achieve better throughputs. More aggressive protocols can also be used to enhance performance. For example, the edge server may set up a constant total bit rate for multiple paths, and then distribute the packets to each path by balancing the loss rate among all the paths.

The benefits of the joint congestion control are of three folds. (a) The burst of traffic will be smoothed. The unified and rate-based sender sends data in a smooth way which will decrease the burst of traffic. (b) Rate control scheme can be easily applied. To adjust the rate for each path, users just need to modify their sending rate and the weight. (c) The aggregate flow has more control in QoS. Flows are aggregated so the hosts can easily adjust the rate for each flow to enable rich flow control.

Most importantly, in MultiServ model, users do not play a passive role in congestion avoidance any more. By cooperation, they can actively alleviate congestion. We no longer need to consider the model of user behavior to control the traffic since our method is based on the phenomena already from the user's point of view. This may be the main difference between our method and other methods toward better QoS for users.

IV. EVALUATION

We show a real experiment to verify our model, currently CERNET backbone (China Education and Research Network, AS number 4538) and HARNET (Hong Kong Academic and Research Network, AS number 3662) is connected by a 2Mbps link. Using such a small bandwidth to support large number of users in the two networks, the link is extremely busy.

In fact, both of the two networks have good connections to other network. Figure 4 shows part of the topology of two networks. From Figure 4, it is clear that these two networks have good connections with Germany and Japan Network, respectively.

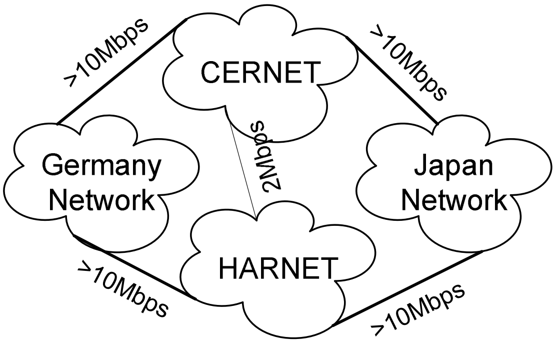

Figure 4. Simplified Topology between CERNET and HARNET.

In our experiment, we use one host in Germany and one host in Japan as our MultiServ hosts. Figure 5 shows that the transmission rate comparison with and without using MultiServ model. It can be seen that the transmission rate is much higher and smoother using MultiServ model comparing with the one obtained by direct connections. Best performance can be achieved when both nodes in German and Japan network had employed our MultiServ model.

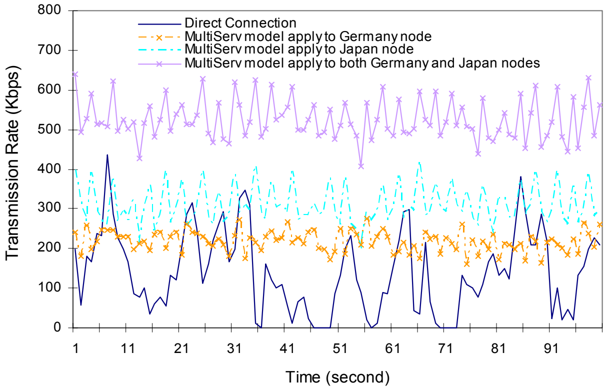

Figure 5. Transmission rate between CERNET and HARNET, with and without MultiServ.

It is believed that all ISPs experience such kind of situation, where the links and traffic are unbalanced. Typically it needs difficult traffic tuning and negotiations with other ISPs. Here MultiServ provides a promising solution.

In general, links with smaller capacity is likely to face congestion. In Figure 6, we show a simple scenario of links between ISPs. Each point in the figure corresponds to an ISP and the links between ISPs have different capacities. Therefore, the utilization of each link could be quite different. For example, the user in ISP 1 may have good experience communicating with users in ISP 3. However, the throughput for users between ISP 2 and ISP 4 may not be satisfied. Moreover, depends on routing policies, packets may not always be able to choose the best path to the destinations.

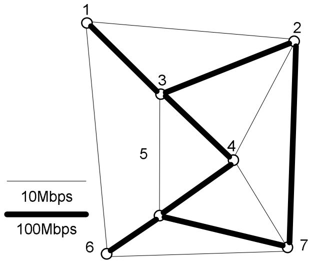

Figure 6. ISPs topology scenario.

Using MultiServ, we can use edge server to redistribute the traffic through low utilized links. Figure 7 shows the network utilization for different types of links with and without using our MultiServ model. In this simulation, we generate self similar background web traffic by placing 420 clients and 84 servers attached to the topology showed in Figure 6, according to the model described in [11]. An ftp session to transfer content of 10MB is setup between each pairs of ISP. To use MultiServ model, we place an edge server in each ISP to redistribute the ftp packets using our algorithm. For example, the transmission between ISP 1 and 2 will be mainly delivered through ISP 3, this decreases the utilization of link between ISP 1 and 2. It can be seen from Figure 7 that, the average utilization for 10Mbps links with direction connection scheme is rather high, and causes much congestion. However, the utilization for 100Mbps is quite low, which causes the unbalanced traffic in the underlying network. Much better performance can be achieved in our MultiServ model. Figure 7 shows that 100Mbps links are highly used to avoid the congestion in 10Mbps links.

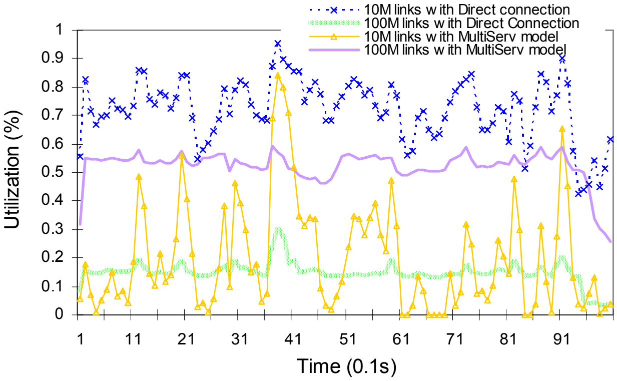

Figure 7. Average link utilization with/without MultiServ.

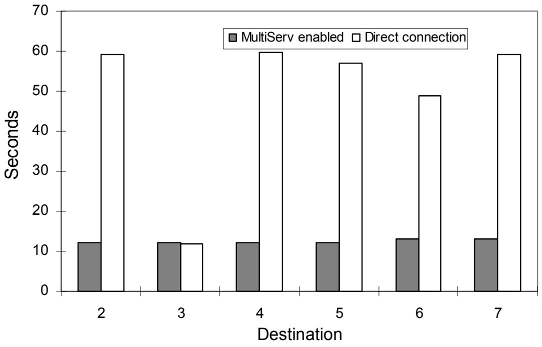

Figure 8. Transmission time from ISP 1 to other ISPs.

We illustrated the transmission time of ftp sessions for ISP 1 to other ISPs in Figure 8. It can be seen that the transmission time can be reduced greatly using MultiServ.

V. CONCLUSION AND FUTURE WORK

This paper proposes a new model named MultiServ to alleviate the congestion and to provide better quality of service for end-host using overlay network. A special overlay is proposed in this model. Meanwhile, a joint congest control scheme is introduced to reduced the unbalanced congestion. The advantages of MultiServ are summarized as follows.

- Congestion Alleviation: MultiServ is a cooperative platform for users to actively balance their traffic and alleviate the congestion;
- Scalable and Stable: The overlay uses a simple construction algorithm which can perform in stable way and can be easily extended to large scale;
- Smooth traffic: The traffic for each delivery is smoothed by the joint congestion control with minimum burst;
- Rich QoS control: The aggregate flow has enabled rich control on flows;
- Easy to deploy: For ISPs, edge servers with MultiServ can be enough to balance traffic; for users, simple software enables the service.

This is an ongoing work with lots of room to improve. We will implement the algorithm and propose to deploy it to real environment for further evaluations. Application-layer multicast based on this platform is also considered as a promising research direction.

REFERENCES

[1] C. Fraleigh, S. Moon, C. Diot, B. Lyles, and F. Tobagi, "Packet-Level Traffic Measurements from a Tier-1 IP Backbone," *Sprint ATL Technical Report TR01-ATL-110101,* November 2001, CA.

[2] Katabi, D., Handley, M., Rohrs, C., "Congestion Control for High Bandwidth-Delay Product Networks", in *Proc. ACM SIGCOMM* 2002, August 2002

[3] S. Floyd, “HighSpeed TCP for Large Congestion Windows”, Internet draft, draft-floyd-tcp-highspeed-01.txt, work in progress, 2002.

[4] D. Wetherall, R. Mahajan, and T. Anderson,. "Understanding BGP misconfigurations," in *Proc. ACM SIGCOMM*, 2002

[5] Z. Morley Mao, R. Govindan, G. Varghese, and R. Katz, "Route Flap Damping Exacerbates Internet Routing Convergence,” in *Proc. ACM SIGCOMM*, 2002

[6] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. “Resilient Overlay Networks”, in *Proc. ACM SOSP*, 2001.

[7] L. Subramanian, I. Stoica, H. Balakrishnan and R. H. Katz, “OverQoS: Offering QoS using Overlays”, in *1st Workshop on Hop Topics in Networks (HotNets-I)*, 2002.

[8] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast”, in *Proc. ACM SIGCOMM*, 2002

[9] K. P. Gummadi, S. Saroiu, and S. D. Gribble, “King: Estimating Latency between Arbitrary Internet End Hosts”, in *Proceedings of the 2nd Internet Measurement Workshop*, Marseille, France, November 2002.

[10] H. Balakrishnan, H. Rahul, and S. Seshan, “An Integrated Congestion Management Architecture for Internet Hosts”, in *Proc. ACM SIGCOMM*, September 1999.

[11] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger, “Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control”, in Proc. ACM SIGCOMM, September 1999.