BI-DIRECTIONAL RESOURCE RESERVATION ANALYSIS FOR DIFFERENTIATED SERVICES NETWORK

Zhao Chen[*], *Tingzhou Yang*[*], *Dimitrios Makrakis*[*]

[*]School of Information Technology & Engineering, University of Ottawa

Colonel By Hall (A617), P.O. Box 450 Stn A, Ottawa, Ont., K1N 6N5, Canada

Email: {zhchen, tyang, dimitris}@site.uottawa.ca

ABSTRACT

IETF has proposed some service models and mechanisms to meet the demand of QoS (Quality of Service), such as Integrated Services and Differentiated Services. In those architectures, QoS and resource reservation has been shown to work when applied in one direction, while availability of bandwidth at the reverse path is taken for granted. This condition may not be fulfilled in a real network. In this paper, we will investigate the effect of both, unidirectional and bi-directional reservation. Our performance analysis demonstrates that in more a realistic network environment, only bi-directional resource reservation can provide to the customers QoS guarantees. Several scenarios of aggregation of applications traffic flows are used to evaluate the performance, such as shot-lived WWW traffic and non-adaptive UDP traffic. Eventually, a bi-directional resource allocation scheme is described and evaluated.

Key words: Differentiated Service, Bandwidth Reservation, Bandwidth Broker, QoS

1.INTRODUCTION

Today's Internet only provides one class of service, known as best effort, to all users. Traffic is processed as soon as possible, but there is no any type of guarantee or ability to provide users with some sort of differential treatment. This creates difficulties for several types of applications, especially those having real time constraints. Multiple autonomous networks called autonomous systems (AS) or domains, each controlled by a separate network operator, form the entire cyber space, as it exists today. Each domain contacts its neighbor domains for datagram delivery; the neighbor domain will in turn pass data to its next neighbor until the traffic reaches its destination.

With the rapid transformation of Internet, it is obvious that several classes will be needed to add enhanced services to different levels of demand. The Internet Engineering Task Force (IETF) has proposed several service models to meet this demand. Differentiated Service (diffserv) is one of them. Differentiated Service defines a DS field in IP packet's header, and a base set of packet forwarding processing (termed Per-Hop Behaviors or PHB) that has to be applied by core routers. By marking the packet' s DS field individually and by handling packets based on their DS fields within diffserv structure, several service classes can be given to clients. For example, 1) Premium Service or Expedite Service [1] for real time applications; 2) Assured Forwarding Service [8] for applications requiring higher reliability from what can be offered by the best effort service. The idea behind diffserv is pushing complex tasks such as classification, policing, shaping and scheduling to the leaf or edge nodes, without increasing the burden for the core routers. For more details, readers can read IETF RFC2475 [3] for the framework of Diffserv.

A number of proposals for providing differentiated services have been presented, such as the RIO (Random Early Detection with In/Out) scheme by D. Clark *et al* [6], the two-bit scheme by K.Nichols *et al* [1] and User-Share Differentiation (USD) scheme by Z. Wang [9]. The above schemes differ primarily on the queue management methods they use, which have to be implemented at, and be applied by the core routers. Results reported in several papers [6,10] have shown that the above schemes perform well when the bottleneck bandwidth is matching the aggregated expected bandwidth profiles. But in those architectures, QoS and resource reservation are conducted only in one direction, while an abundance of bandwidth at the reverse path is assumed. This means the returning ACKs are supposed to never be lost along the reverse path; however, in a real network, this condition might not be satisfied. In this paper, we investigate the performance of a diffserv structure under bi-directional heavy traffic loading conditions. We can see that use of unidirectional reservation does not work; only bi-directional reservation can really differentiate the users' services according to their subscribed profiles.

The rest of the paper is organized as follows: Section 2 introduces the resource reservation policies for diffserv and some terminology. A complete example for delivering end-to-end unidirectional service and related reservation is presented here as well. Section 3 describes our network configuration and reports the performance evaluation results for unidirectional reservation, under different traffic types and link conditions. In section 4 we propose a possible bi-directional resource reservation scheme and compare it with the unidirectional scheme under two-way heavy traffic loading. Finally section 5 concludes our work and discusses some relevant issues.

2.CURRENT RESOURCE RESERVATIONSCHEMES FOR DIFFSERV

2.1 Useful Terminology

The following terms that are used throughout the paper are defined here for clarity. Some of the terms defined in the Diffserv (DS) architecture [3] are repeated here for completeness.

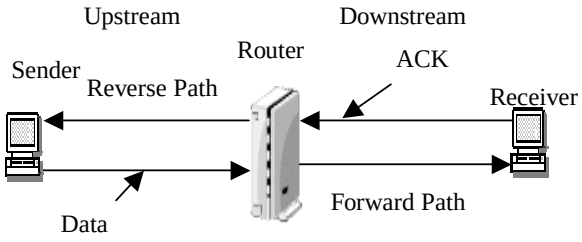

Figure 1 Direction of Data and ACK streams DS

Boundary node or Edge Router: A DS node that connects one DS domain to a node in another DS domain or in a domain that is not DS-capable.

DS domain: A DS-capable domain; it consists of a contiguous set of nodes, which operate with a common set of service provisioning policies and PHB definitions.

Bandwidth Broker (BB): A bandwidth broker (BB) manages the network resources for a single domain, in order to support the QoS requirements of the customers

Forward Path: Along this path, data originated from a sender are transported to its destination, the appropriate receiving station.

Reverse Path: Used to transport the ACK messages generated by the receiving station back to the sender.

Service Level Agreement (SLA): A service contract between a customer and a provider that specifies the forwarding service a customer should receive. A customer may be a user organization (source domain) or another DS domain (upstream domain).

2.2 A Unidirectional End-to-end Reservation Example

The work reported in [2] proposed a two-tier resource management model for diffserv networks. The domain can still be the basic resource agent to control bandwidth allocation. Moreover, we assume that a Bandwidth Broker (BB), presented by Van Jacboson [1], takes this responsibility for each domain. Adjacent domains with related aggregation of border-crossing traffic reach bilateral SLA. Meanwhile, each domain may choose its own resource reservation protocol for its internal QoS need. [1] and [2] also give us some details describing the bandwidth allocation strategy and its feasibility. Their conclusion is that end-to-end QoS support can be implemented through the concatenation of inter- and intra-domain resource allocations, which is similar to current IP two-level routing architecture.

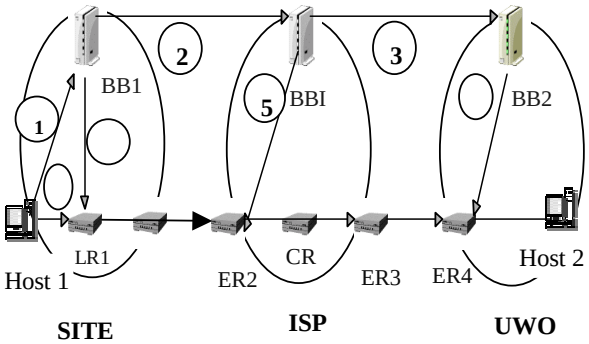

Figure 2. Example of end-to-end resource allocation, with unidirectional reservation

Here we give an end-to-end allocation example to illustrate the ideas mentioned above. In figure 2, there are dynamic SLAs between domain SITE, its ISP domain and domain UWO, and the BBs exits in each domain are using RSVP as their signaling protocol. Assuming that Host 1 (in domain SITE) needs to send 100 kbps data to host 2 (in domain UWO) using Assured Service class. The reservation procedure is described below.

1. Host 1 sends a RSVP PATH request to its Bandwidth Broker BB1.

2. BB1 makes an admission control decision to check if it can accept this request. If the request is denied, the reservation process has failed. If accepted, BB1 sends a PATH message to its neighbor BBI.

3. BBI then makes its admission control decision and in turn sends its PATH to BB2, if the request is authenticated.

4. BB2 checks this request and grants it permission. Then, BB2 will use RSVP or another local signaling protocol to configure the edge router ER4 to allocate resource for this request. Meanwhile, BB2 returns an RSVP RESV message to BBI.

5. After receiving RESV, BBI will configure its edge router ER2 to support the reservation. As next action, it sends RESV back to BB1.

6. After receiving RESV, BB1 will configure the leaf router LR1 to correctly classify and shape the admitted data from H1. BB1 then sends RESV to host 1.

7. All BBs will also set the policing and reshaping rules on the egress routers like ER1 and ER3. Now the reservation process is completed and host 1 starts to send data.

From the above example, we can observe that this reservation scheme is applied only in one direction, the data forwarding path. It assumes implicitly that the reverse path has enough bandwidth available, so that ACKs can traverse it without loss and congestion. However, with the deployment of a QoS-capable network like Diffserv and MPLS, even if ACKs are smaller than data packets, if marked as low priority or best effort class, they may experience congestion in the reverse path and be discarded or be delayed considerably, as the network takes action to protect and accommodate higher priority class traffic. This will affect the performance of the data forwarding path in a negative

manner, despite the fact that reservation has already been made along the forward path. In the next section we will see the effect of this lossy reverse link under different network conditions.

3. SIMULATION SETTINGS AND RESULTS

Figure 3 shows our simulation network configuration. Our

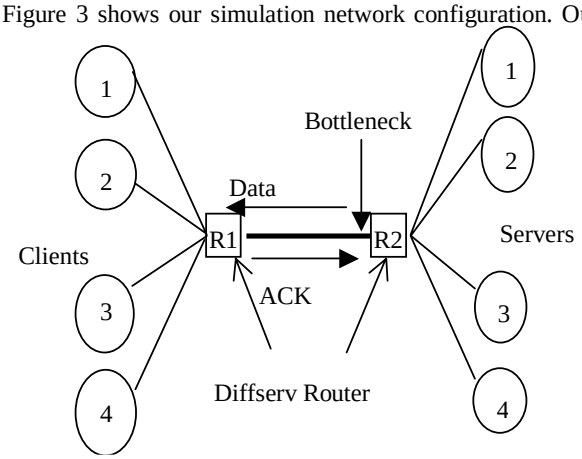

Figure 3. Network Topology with one bottleneck

simulation environment is based on the OPNET 6.0 [7] network simulation tool. We have run simulations under different network conditions, in order to be able to find out the effect of resource allocation. The thick line represents the bottleneck link, with 3 Mbps bandwidth with negligible delay, while other lines have 10 Mbps bandwidth, which is much larger than the traffic on it. There are 4 servers and 4 client groups in the network shown in Figure 3. Each client group consists of 100 workstations and each workstation in client group ***i*** will served by server ***i.*** Therefore, the traffic traversing each thin line is aggregation of 100 individual flows, and the traffic on the bottleneck link is aggregation of all 400 flows. R1 and R2 are diffserv-capable routers and implement Weighted Fair Queuing (WFQ) to support multiple service classes. We partition the subscribers into two groups. The first is named "high profile" and the second "low profile". Client groups 3 and 4 belong to the high profile group while client groups 1 and 2 belong to low profile. Low profile is assigned 0.5 Mbps or 1/6 of the bottleneck bandwidth, while the high profile is assigned double the bandwidth of the low profile. Since Web traffic constitutes the largest portion of current Internet [4], we select OPNET WWW application as our workload. It models a session in which the workstation can establish multiple connections to servers, send multiple request commands for HTML pages and inline objects, and process the request responses. Each WWW client group generates traffic at a average rate equal to 2.0 Mbps and total generated traffic rate exceeds the bottleneck link capacity. TCP NewReno [5] is employed for the adaptive sources, which can fast recover from packet losses.

3.1 One Way Reservation without ACK Disturbing on Reverse Path

First, we consider the case where bulk traffic is sent from server ***i*** (***i*** = 1,2,3,4) to client group ***i***, and ACKs are travelling at the opposite direction. In the scenario examined in the present section, we provide considerably high bandwidth to the reverse path, so that ACK will not be delayed or lost due to queuing and congestion. Under these conditions, our results confirm the findings by Z. Wang in [10].

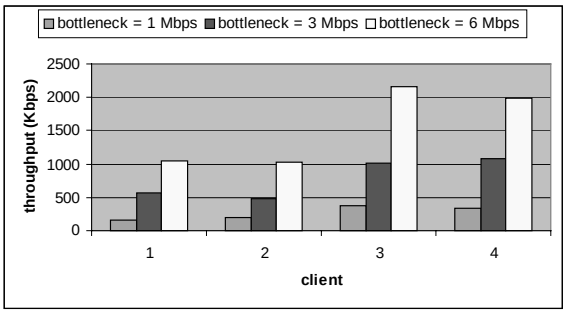

Figure 4. Case 1: Bandwidth share of each client group, for 3 different bottlenecks

When WFQ scheme is used, in Figure 4 we present the average throughput of each client group. For all three different bandwidth allocations made to the bottleneck path (1 Mbps, 3 Mbps, 6 Mbps), the high profile subscribers (group 3 and 4) always get twice the average throughput of the low profile subscribers (group 1 and 2), which is matching their expected bandwidth allocation.

Next, we consider a scenario where a mix of TCP and non-adaptive UDP traffic competing for the bottleneck capacity.

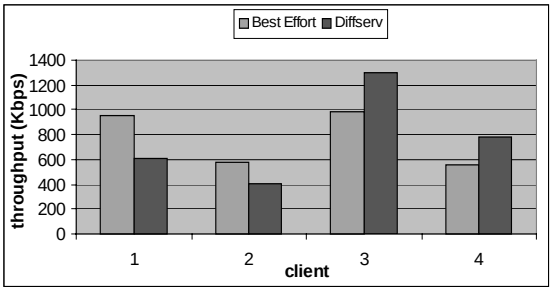

Figure 6. Case 3: Impact of RTT

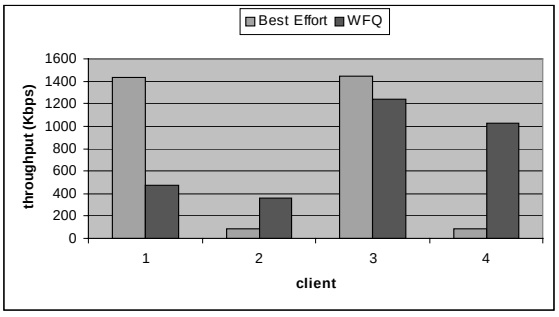

Figure 5. Case 2: Effect of non-adaptive sources

Table 1 Configuration with UDP traffic

Client group	1	2	3	4
Traffic type	UDP	WWW	UDP	WWW
Expect profile	0.5 Mbps	0.5 Mbps	1.0 Mbps	1.0 Mbps

In our simulations, each UDP client group is flooding the network at constant bit rate 2.0 Mbps and is not responsive

to congestion and ACK loss. Table 1 shows the assignment of profiles to TCP and UDP traffic. This time, the bottleneck bandwidth is 3 Mbps. We notice from the curves displayed in Figure 5 that the Constant Bit Rate (CBR) UDP traffic from clients 1 and 3 takes almost all the link capacity when competing with the TCP traffic in a best effort network. This is due to the fact that CBR UDP sources are sending at a fixed rate, without backing off in the face of congestion. When WFQ is employed, the bandwidth allocations are much closer to the expected bandwidth profiles. Low profile clients in groups 1 and 2 achieve about 400 Kbps throughput, while their profile was 500 Kbps. High profile clients in groups 3 and 4, achieve 1.2 Mbps and 1.0 Mbps throughput respectively, while their profile is 1 Mbps. In each group, CBR UDP sources still get slightly higher throughput than their TCP counterparts but the difference is acceptable. This implies that diffserv is effective when dealing with non-responsive traffic.

It is well known that long round-trip time (RTT) flow can affect TCP connections in a negative manner. Large RTT connections might not be able to get as much bandwidth as shorter RTT connections. To see its effect on a diffserv capable environment, we establish four WWW client groups and apply 1.0 seconds extra delay to the links from clients in groups 2 and 4 (to servers 2 and 4) respectively. This almost doubles the RTT of the above links. From the curves displayed in Figures 6 we see that in a best effort network shorter RTT connections have considerable advantage as compared to longer RTT links. In our case, clients from groups 1 and 3 have throughput that is almost twice the throughput of clients from groups 2 and 4. This confirms that TCP throughput is inversely proportional to the RTT. In the same figure, we provide the throughput when diffserv is used. We assign clients from groups 1 and 2 to low profile and clients from groups 3 and 4 to high profile. Obviously, the fairness of the different RTT connections improves a lot under diffserv, even though clients with shorter RTT still get more bandwidth than their profile, at the expense of the bandwidth of clients with long RTT. While the profile of the low profile group is 500 Kbps, the throughput of client group 1 with short RTT is about 600 Kbps, and client group 2 with long RTT obtain 400 Kbps. Regarding the high profile groups, the throughput of clients from group 3 with short RTT is 1.2 Mbps while the throughput of long RTT clients from group 4 is 0.8 Mbps. We conclude that the large RTT connections are quite assured of their profile rate (20% short from expectation according to the results reported in Figure 6).

In summary, our findings reported in this section, lead to the following conclusions. Assuming that the ACK path is not experiencing losses and delays, diffserv network architectures are able to render rate guarantee to subscribers under different network contexts. In the following section, we will investigate the validity of this conclusion when realistic approach regarding the effect of network on ACK packets is adopted.

3.2 Impact of ACK Loss on Reverse Path

TCP uses the ACK clock to estimate the conditions within the path. Congestion in reverse path, carrying the ACK messages, increases the RTT of the connection and causes loss of ACKs. Longer RTT reduces throughput and increases end-to-end delay. Furthermore, multiple loss of ACK slows the growth of congestion window fast, which results in poor performance for TCP connections. To see the impact of ACK loss, we intentionally introduce packet losses on the ACK path for the network configurations presented above (See section 3.1). ACK packets over the reverse path are randomly discarded according to the certain probability specified by the desired loss rate.

In Figure 7, the bottleneck bandwidth is 3 Mbps. The high profile is 1 Mbps and low profile is 0.5 Mbps. As Figure 7 indicates, Web sources using the NewReno version of TCP are fairly sensitive to ACK loss. When ACK loss rate is above 2%, all TCP connections suffer considerable degradation, to the point that we can not differentiate the performance of high profile clients in groups 3 and 4 from the performance of low profile clients in groups 1 and 2. The reason is that WWW sources will set up multiple short duration TCP connections for each Web page and TCP Reno can only recover only one ACK packet loss. As result, many lost ACK are not recoverable. Moreover, when multiple ACK loss occurs, TCP is forced to enter slow start status, with only one segment congestion window size. Although we have made reservation on the data-forwarding path, the performance of all clients is heavily impaired, regardless of the class they belong, due to the ACK losses. Their throughput is considerably below their profile, and it is almost the same for all of them Figure 8 displays the utilization at bottleneck link as a function of ACK loss rate, which is a good indicator of the network performance Note that the utilization is only 40% when the ACK loss rate becomes 5%.

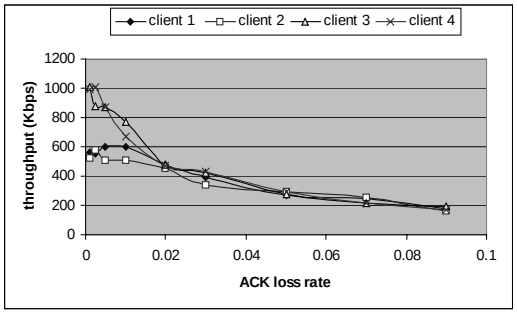

Figure 7. Case 1:Bandwidth share as a function of ACK loss rate

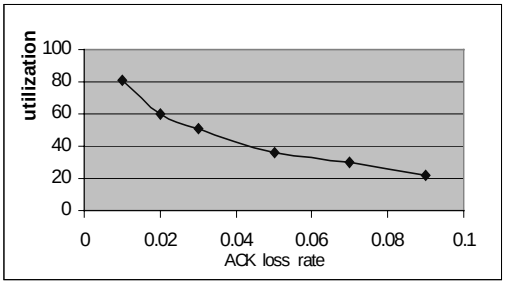

Figure 8. Utilization of bottleneck link

Next we consider the presence of non-adaptive sources. Instead of having all clients making use of TCP, clients in groups 1 and 3 are producing CBR traffic running over the

UDP protocol. We assign again clients from groups 1 and 2 to low profile and clients from groups 3 and 4 to high profile. From the results in Figure 9 we see that UDP clients from groups 1 and 3 are almost immune to the increase in ACK loss rate, but TCP clients from groups 2 and 4 suffer considerable deterioration as the ACK loss rate increases. Also, clients of both groups receive almost the save throughput for ACK loss rates 2% and above. Obviously, the TCP links' throughput is much lower than its expected profile. This behavior is reasonable, since UDP does not react to ACK loss, while TCP is backing off when experiencing loss of ACK. It is clear that one-way reservation does not make sense when there is the possibility (and it definitely exists in most cases) of facing heavy ACK loss.

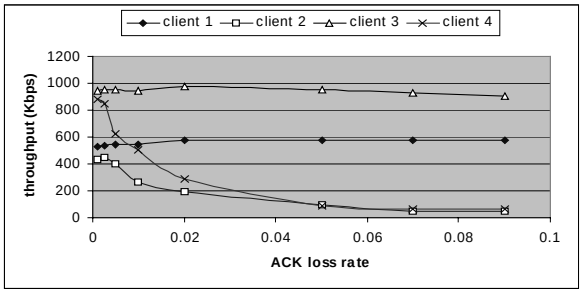

Figure 9.Case 2:Mix of UDP and TCP with ACK loss

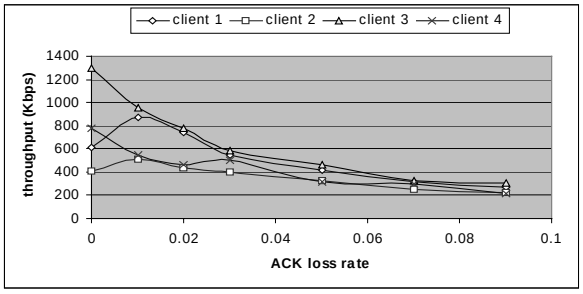

Figure 10. Case 3:Impact of different RTT with ACK loss

Finally, we examine the combined effect of round-trip time delay and ACK losses on the performance of TCP connections. We use the same configuration as in section 3.1.3 and vary the ACK loss rate on reverse direction. The curves shown in Figure 10 indicate that the reservation at the forward path does not work when experiencing ACK losses at the reverse link. While for ACK loss rates higher than 6%, all TCP connections appear to be reaching the same throughput level, TCP connections with shorter RTT (clients from groups 1 and 3) have an advantage over TCP connections with longer RTT (clients from groups 2 and 4). As result, high profile reservation does not give any advantage to clients from group 4 over the low profile but shorter RTT clients of group 1. This leads to the conclusion that RTT has a more significant impact on the performance than bandwidth reservation. According to our results shown in Figure 10, when the ACK loss rate exceeds 6%, the difference between clients that were supposed to receive different treatment diminishes.

All the above results imply that the unidirectional reservation does not provide advantage under the realistic scenario of bi-directional network traffic loading. Should we wish to provide differential treatment to Different Internet users, it is imperative that provision of some form of protection to ACK messages is provided. Next section we will provide a feasible modification to the scheme described in Figure 2 for bi-directional reservation.

4. BI-DIRECTIONAL RESERVATION AND ITS EFFECT

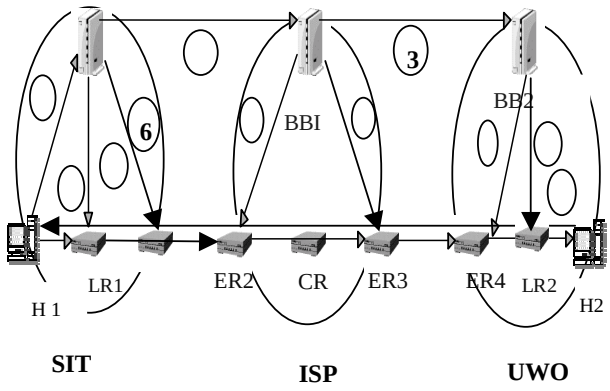

Figure 11. Network architecture with bi-directional reservation

4.1 Example of Bi-directional Reservation

The results presented in Section 3 suggest the failure of unidirectional reservation in case of ACK losses. Hence, we argue that we need to make reservation not only on the data forwarding path but also on the reverse path, in order to avoid ACK loss. In the present section we modify the end-to-end allocation architecture that was described in section 2, in order to introduce bi-directional reservation. In figure 11, Host 1 (in domain SITE) needs to send data to host 2 (in domain UWO) at an average rate of 100 kbps using Assured Service class. At the same time, we assume that host 2 returns 20 kbps of ACK traffic to host 1. The bi-directional reservation procedure is described below.

1. Host 1 sends a RSVP PATH request to its Bandwidth Broker BB1. The request now includes 100 Kbps on the forward path and 20 Kbps on the reverse path.

2. BB1 makes an admission control decision to check if it can accept this request. If the request is denied, the reservation process has failed. If accepted, BB1 sends PATH message to its neighbor BBI.

3. BBI then processes the request, makes its admission control decision and in turn sends its PATH to BB2 if the request is authenticated.

4. BB2 checks this request and grants its permission. BB2 will use RSVP or another local signaling protocol to configure the edge leaf router LR2 to mark and shape ACK traffic sent from host 2. ER4 is configured to support 100 Kbps data from host 1. Meanwhile BB2 returns a RSVP RESV message to BBI.

5. Receiving RESV, BBI will configure its edge routers

ER2 and ER3 to support the reservation on both directions. Next, it sends RESV back to BB1.

6. After receiving RESV, BB1 will configure the leaf router LR1 to correctly classify and shape the admitted data from H1, and ER1 is set to accommodate ACK traffic from host 2. BB1 then sends RESV to host 1.

7. Now the reservation is completed. Host 1 starts to send data and host 2 returns ACK with their reservation envelops.

Notice the difference from the unidirectional reservation scheme described in section 2. First, both ends can initiate the reservation process, not only the subscriber host 1. This makes sense, since sometimes subscribers want to see the incoming traffic protected by their profile, not the outgoing traffic. For instance, a user needs to download Web pages from an ISP server. It is possible that the ISP server is a better candidate to initiate reservation since it is more likely to know the requirement of traffic than the user. Secondly, the reservation is made on both directions to avoid ACK loss and facilitate interaction between both ends. Sometimes, we must set up two-way allocation. Examples are interactive applications like videoconference, which demand both incoming and outgoing traffic to be guaranteed. Third, a BB may aggregate multiple requests and pass a single request to its neighbor. This can reduce the overhead of signaling messages. Since a signaling message may contain two-way reservation and multiple request aggregation, we can say that this reservation scheme scales well with the increase of connections.

4.2 Simulation Results of Bi-directional Reservation

To evaluate the performance of our reservation schemes in a realistic network, we change our network settings in Figure 3. We are allowing both ends to send WWW traffic to the other side; i.e. bulk traffic is sent not only from server ***i*** to client ***i*** but also from client ***i*** to server ***i*** at the same time. On both directions, ACK messages are competing with bulk data for bandwidth, and the overall traffic rate exceeds the bottleneck capacity of 3 Mbps. Surely ACK packets are not immune to bottleneck congestion and can be delayed and lost. We ran two sets of simulations under two scenarios, one using unidirectional and the other bi-directional reservations. For the unidirectional scheme, only the bulk data is protected by the contracted profile, while ACK flow is treated as best effort traffic. For the bi-directional one, ACK flow is treated as the same service as its acknowledged bulk data. We use a fairness index defined in [7] to evaluate the fairness of the shared bandwidth among users: where n is the number of links sharing the network resources, and x_i is the ratio of the actual throughput of a connection to the fair share of the available bandwidth for the connection.

$$Fairness\_Index = \frac{\left[\sum_i x_i\right]^2}{n \bullet \sum_i x_i^2}$$

This index reaches its maximum value 1 when the connections receive the allocation they subscribed for and is an increasing function of fairness.

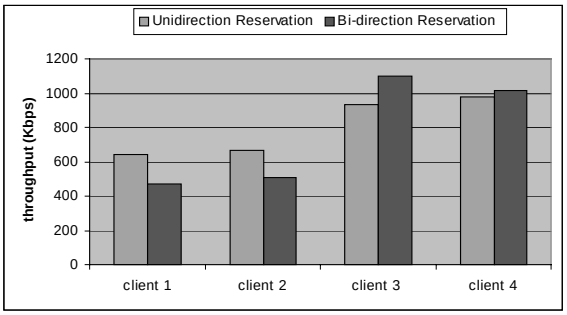

Figure 12. Case1: Two profiles reservation

Figure 13. Case2: UDP and TCP

Figure 14. Case3: Different RTT

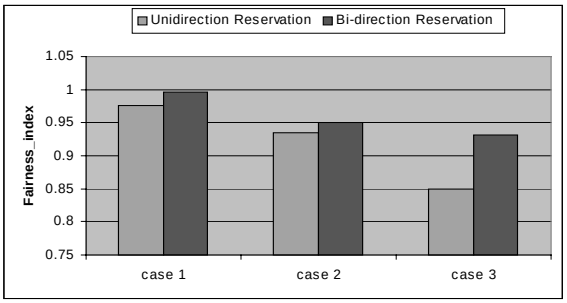

Figure 15. Fairness comparison

As in section 3, clients 1 and 2 are low profile subscribers and clients 3 and 4 are high profile subscribers. Low profile subscribes for 0.5 Mbps and high profile for 1.0 Mbps. In case 1, identical WWW settings are applied to all clients. In case 2, clients 1 and 3 are CBR UDP sources, so that we can investigate the effect of UDP on WWW TCP traffic from clients 2 and 4. Case 3 shows the impact of RTT on four WWW TCP connections. Here the connections of clients 2 and 4 are given 1 second extra delay, and

all traffic is Web traffic. Figures 12, 13 and 14 provide us with the average throughput received by clients in three different cases. As can be seen from the figures, in all cases, bi-directional reservation improves the throughput of high profile clients and lowers the throughput of low profile clients, when compared with unidirectional reservation. In case 1, the bi-directional reservation's allocation of bandwidth is very close to subscribers' profile. The impact of UDP traffic is shown in Figure 13. The allocation here is worse than in case 1 but still acceptable. In case 3, the RTT acts as predominant factor for bandwidth share, not bandwidth reservation, since short RTT client 1 gets more throughput than high profile client 4, which has a longer RTT link. However, bi-directional reservation again performs better than the unidirectional scheme.

The fairness index shown in Figure 15 is a clear proof of the advantage of bi-directional reservation against unidirectional one. In all cases, the fairness index of the bi-directional scheme is better than that of its unidirectional counterpart. The increase of the fairness is 2.1% for the first case, 1.8% for the second case and 9.5% for the last case. Judging from our results, we can claim safely that bi-directional reservation provides a definite advantage in a diffserv capable environment.

5. CONLUSION

In this paper, we have evaluated the performance of two different bandwidth allocation schemes for a differentiated services capable environment, under realistic network conditions. The first is based on unidirectional reservation, while the second on bi-directional reservation. The two different schemes have been evaluated under several different scenarios and traffic loading. A key part of our analysis, is the assessment of the effect, loss of ACK messages has on the performance of TCP connections running through the differentiated services capable network and on the bandwidth allocation. The results show that: (1) diffserv-capable networks perform well without ACK loss; (2) ACK loss may destructively impair the performance of TCP connections even when bandwidth reservation has been made on data forwarding path; (3) bi-directional reservation performs better than unidirectional reservation in all examined scenarios. We also outline a possible reservation solution that protects from ACK losses that are occurring at the return path. Our conclusion is that if we take the realities of current Internet and the traffic loading levels into consideration, we must make bi-directional reservation in order to be able to provide the QoS guarantee to users. Also, under bi-directional reservation, the diffserv network can provide class service differentiation even under complex and highly loaded network conditions, something that is not capable of achieving under unidirectional reservation.

6. REFERENCES

[1] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", RFC 2638, July 1999

[2] A. Terzis, L. Wang, J. Ogawa, L. Zhang, "A Two-Tier Resource Management Model for the Internet", Global Internet 99, Dec 1999

[3] S. Blake, D. Clark, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, Dec. 1998

[4] K. Thompson, G.J. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", IEEE Network, Nov./Dec. 1997.

[5] K. Fall and S. Floyd, "Simulation-Based Comparisons of Tahoe, Reno, and SACK TCP", Comp. Comm. REV., July 1996

[6] D. Clark and W. Fang, "Explicit Allocation Of Best Effort Packet Delivery Service", IEEE/ACM Trans. on Networking, vol. 6, no. 4, August 1998.

[7] OPNET, http://www.mil3.com/

[8] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group", IETF RFC 2597, June 1999.

[9] Z. Wang, USD: Scalable Bandwidth Allocation for the Internet, HPN'98, Vienna, Sept. 1998

[10] A. Basu, Z. Wang, "Fair Bandwidth Allocation for Differentiated Services", Proceedings of the IFIP Workshop on Protocols for High Speed Networks (PfHSN), Salem, Massachusetts, June 1999.

[11] ATM-Forum, "ATM Traffic Management Specification Version 4.0", April 1996.