# T
# W M
# M

Jason Leigh, Oliver Yu, Alan Verlo, Alain Roy, Linda Winkler, Thomas A. DeFanti
(evlnet@evl.uic.edu)

Dec 12, 2000

Differentiated Services (DiffServ) is a mechanism for supporting network Quality of Service (or QoS) whereby packets that are transmitted by a client program are marked with a priority setting that can be interpreted by the router to effect special treatment of the packet. In particular the marked packets are promoted to a higher priority queue in the router and, as a result, spend a minimum amount of time in the router. Packets that are not marked are attached to a lower priority queue, and in some cases may be dropped when congestion arises. A more detailed description of DiffServ may be found in the paper by Sander et al [Sander et al 2000].

A series of experiments were performed over a wide area DiffServ testbed as part of the EMERGE project. (www.evl.uic.edu/cavern/EMERGE) is a Department of Energy funded project for designing, deploying and testing Differentiated Services on an IP/ATM Regional GigaPoP Network interoperating with ESnet for applications in Combustion, Climate and High-Energy Physics. The main participants of the experiments in this report were EVL and Argonne National Laboratory (ANL).

DiffServ routers at ANL were connected to DiffServ routers at EVL as shown in Figure 1. The goals of the experiments include:

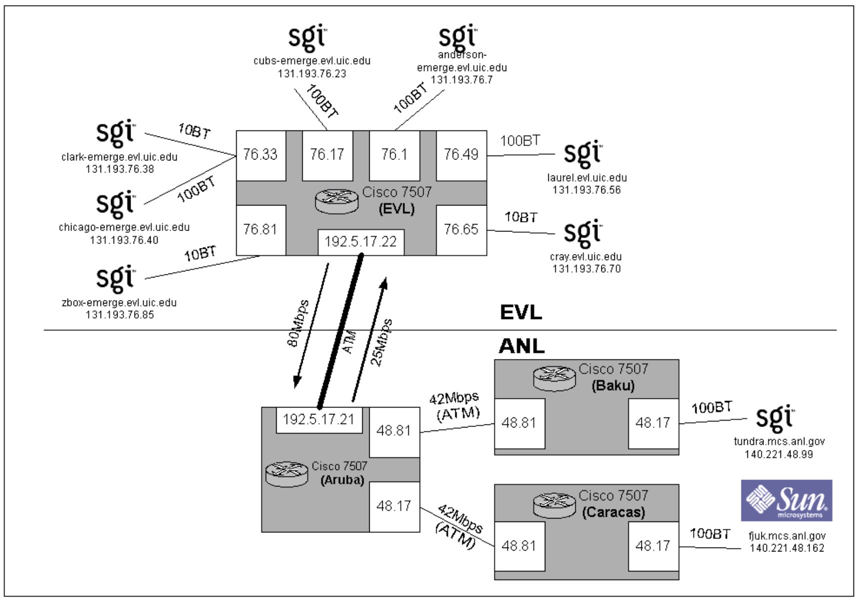
- determining how well DiffServ provides bandwidth and latency recovery over congested networks;
- determining how the performance gains achieved by using DiffServ would impact the corresponding performances of non-DiffServ (best-effort) traffic;
- and determining how DiffServ behaved for a realtime tele-immersive application.

The router at EVL had Weighted Fair Queueing enabled; and the routers at ANL had Priority Queueing enabled to produce DiffServ's Expedited Forwarding behavior. At each of the end points several Silicon Graphics workstations and a single Sun workstation were connected to the routers. The DiffServ and resource reservation control operations were provided by the DiffServ Manager, which is based on the Globus' Architecture for Reservation and Allocation (GARA) [Foster et al 2000, Globus, GARA]. The DiffServ manager was used to enable DiffServ as well as traffic policing and shaping during the experiments.

## T

In the first experiment 25Mbps of foreground data traffic was streamed over UDP from Laurel (at EVL) to Tundra (at ANL). After some time additional competitive background traffic of 25Mbps was activated between Cubs (at EVL) and Tundra. The foreground and background data traffic passed through the EVL ingress router, the Aruba intermediate router and the Baku egress routers at ANL. The ATM pipes from the EVL router to Aruba and from Aruba to Baku were policed at 80 Mbps and 42 Mbps respectively. Hence a congestion bottleneck is created at the output port from Aruba to Baku.

Finally after some additional time, DiffServ was enabled for the flow between Laurel and Tundra to determine how well bandwidth would be recovered. The graphs in Figure 2 show the bandwidth, latency and packet loss of the foreground flow during the three stages of the experiment. Notice that in the first stage when there is no competing traffic, a little less than 25Mbps was achieved (as expected). However when background congestion was introduced, bandwidth drops while latency and packet loss increases sharply. Finally when DiffServ is enabled, the original bandwidth and latency is restored and packet loss also declines. Hence it would appear that DiffServ is capable of providing bandwidth guarantees. This result confirms results from similar experiments conducted at ANL [Sander et al 2000] with data traffic transported over TCP.




T .

In the second experiment traffic is reversed. 25Mbps of foreground data traffic over UDP is first sent from Tundra to Laurel. Then 25Mbps of background traffic is sent from Fjuk to Laurel. The foreground and background data traffic passed through the Baku and Caracas ingress routers respectively, converging via the Aruba intermediate router and the EVL egress router. The ATM pipes from Baku or Caracas to Aruba and from Aruba to EVL were policed at 42 Mbps and 25 Mbps respectively. This time the congestion bottleneck is at the output port from Aruba to EVL. Finally as in experiment 1, DiffServ is enabled on the foreground flow.

Initially the results (Figure 3) appear to be the same as those in experiment 1. The introduction of background traffic has a detrimental effect on the foreground traffic's bandwidth, latency and packet loss rate. And again, when DiffServ is enabled, bandwidth recovery occurs as before. However note that latency was only partially restored and packet loss has doubled. Note also that the restored latency is at approximately 150ms which has been shown to be an intolerable level for realtime tightly-coupled interactions in Tele-Immersion. Park et al[1] [Park99, Leigh98] have found that the latency threshold where human performance begins to noticeably degrade is approximately 200ms

[1] Park's results also suggest that 200ms of roundtrip latency with 0 jitter has the same effect on users as 10ms of roundtrip latency with 7ms jitter. In essence, this implies that the minimization of jitter is also critical in realtime tightly coupled tele-immersive applications.

This would suggest that while DiffServ is suitable for making bandwidth guarantees it is unable to reliably make latency guarantees.

### T
### . T

It is not sufficient that DiffServ is able to provide bandwidth guarantees. DiffServ should be able to provide bandwidth guarantees without having an unexpectedly adverse effect on non-DiffServ (best-effort) traffic. In this experiment varying ratios of foreground traffic and background traffic were sent over the testbed. This time, throughput and latency for both the foreground traffic and the background traffic were monitored. We define the in using DiffServ as the amount of recovery that occurs when Diffserv was applied to the foreground traffic. And we define the as the performance degradation in the non-DiffServ background traffic. Ideally the ratio of gain to loss should be greater than or equal to 1. That is DiffServ should provide equal, if not more, gain than loss in performance on the network as a whole. Figure 4 shows this is indeed the case for a variety of foreground and background traffic quantities.

### 5

The previous experiments generate streams of traffic at relatively high rates to show DiffServ's ability to recover bandwidth and latency in an artificial setting. This final experiment uses a real tele-immersive application (the Tele-Immersive Data Explorer – or [Sawant et al. 2000]) and observes whether the same behavior occurs for one tele-immersive data stream- namely the avatar data stream. In this experiment a tele-immersive client is launched on Laurel and on Clark at EVL, while a tele-immersion server is launched on Tundra. Avatar data is transmitted to Tundra and then routed to Clark; and vice versa. Avatars are virtual representations of participants in a tele-immersive environment. Avatars often consist of a head, body and hand that mimic the gestures of a remote participant. The data stream to convey these gestures are often small in size (only about 1Kbps per avatar) but latency and jitter sensitive. Hence this is one of the streams that one would normally consider a suitable candidate for applying QoS.

Figure 5 shows the latency of the avatar stream before and after background competitive traffic is injected, and also when DiffServ is enabled. In this case since the avatar bandwidth is so small, 45 Mbps of background traffic had to be injected to adequately saturate the network. The results are very similar to those found in experiment 2. That is, original latency levels are not restored. There is little noticeable effect on bandwidth because the avatar bandwidth is so low as compared to the overall bandwidth of the network connection and the bandwidth of the background competitive traffic.

DiffServ appears to be able to reliably provide bandwidth guarantees without incurring any unexpected performance loss in competing non-DiffServ streams. However DiffServ should not be relied upon to maintain latency guarantees when the network becomes overloaded. Tele-immersive applications (as well as realtime audio and video conferencing applications) that require latency guarantees should consider an Integrated Services QoS scheme, such as RSVP over Multi-Protocol Label Switching (MPLS), to achieve per flow latency guarantees.

## S


EMERGE

Foster, I., Roy, A., Sander A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation., ***8th International Workshop on Quality of Service IWQOS 2000***, pp. 181-188, June 2000.

GARA

Globus

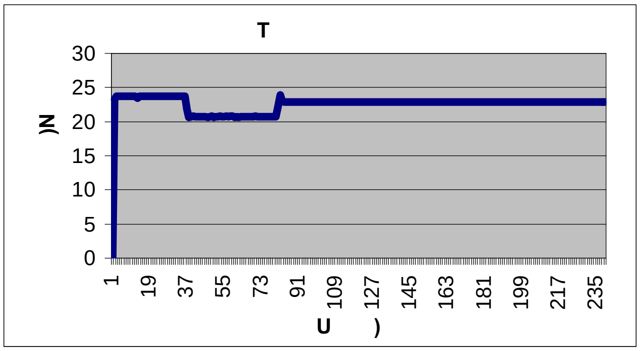
Leigh, J., Park, K., Kenyon, R., Johnson, A., DeFanti T., Wong, H , ***Proceedings of HPCAsia '98***, Singapore, 09/22/98-09/22/98, pp. 687-693.

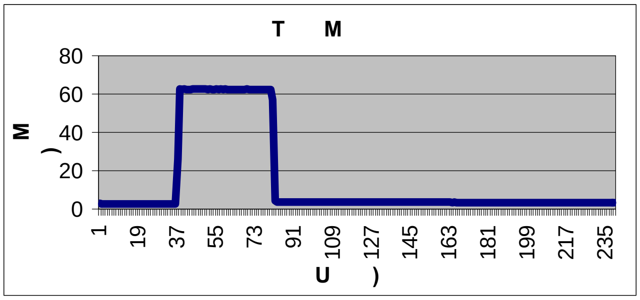
Park, K., Kenyon, R., , ***Proceedings of IEEE VR `99,*** *Houston, TX, 03/13/99-03/17/99.*

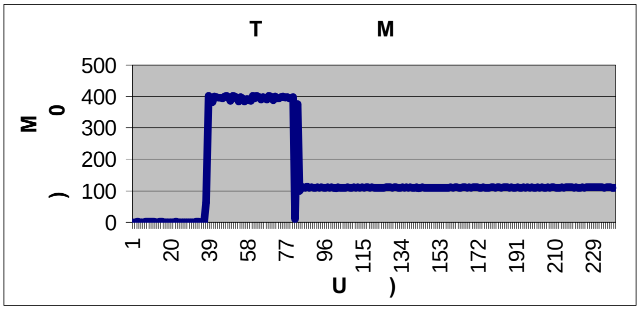
Sander, V., Foster, I., Roy, A., Winkler, L., ***TERENA Networking Conference 2000***

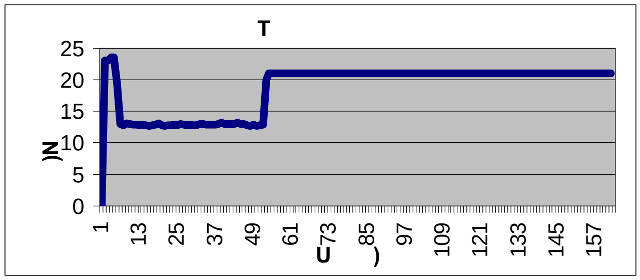
Sawant, N., Scharver, C., Leigh, J., Johnson, A., Reinhart, G., Creel, E., Batchu, S., Bailey, S., Grossman, R., , ***4th International Immersive Projection Technology Workshop***, Ames, Iowa, June 19-20, 2000.

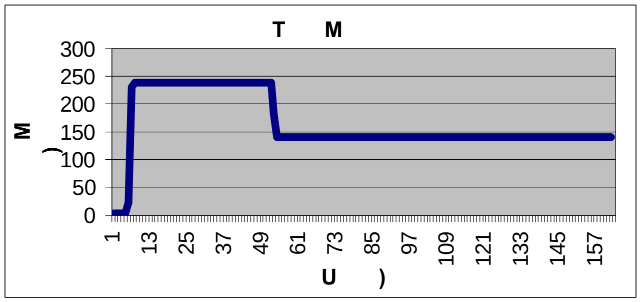

EVL internal report.

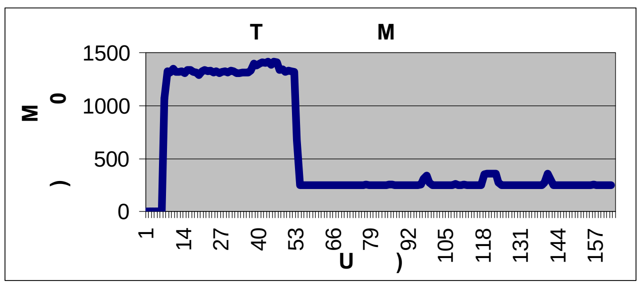

T
30
25
20
15
10
5
0
1
19
37
55
73
91
109
127
145
163
181
199
217
235

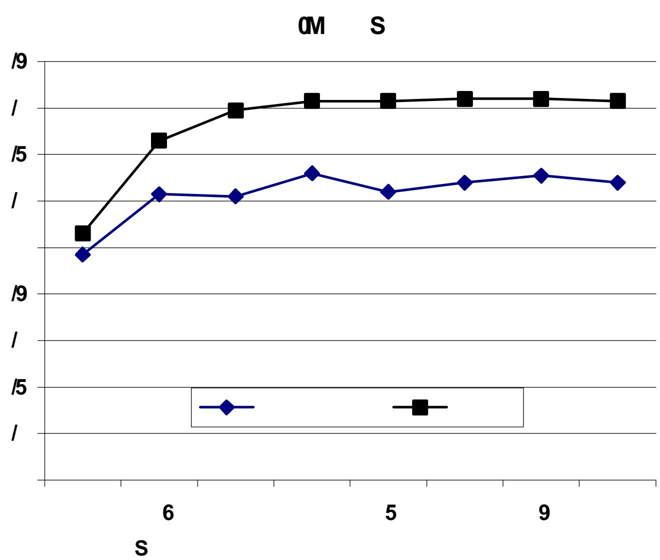

T M
80
60
40
20
0
1
19
37
55
73
91
109
127
145
163
181
199
217
235

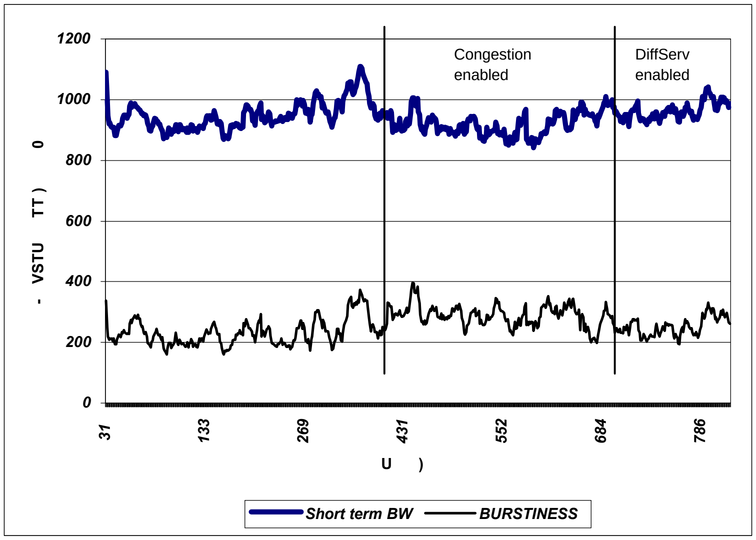

T M
500
400
300
200
100
0
1
20
39
58
77
96
115
134
153
172
191
210
229

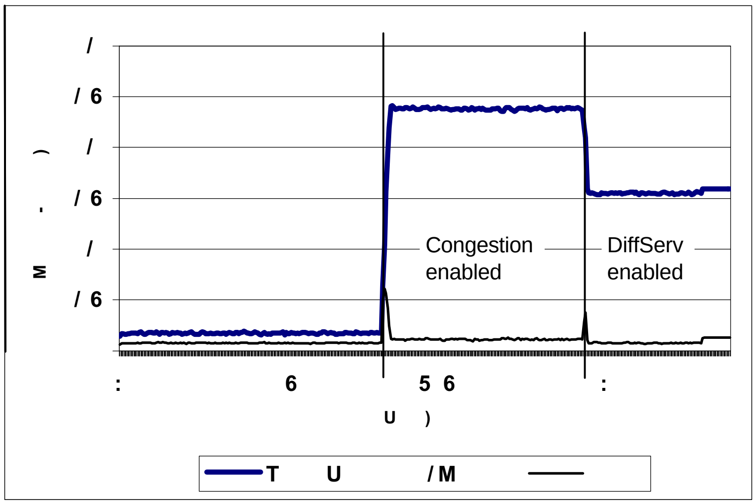

T
25
20
15
10
5
0
)N
1
13
25
37
49
61
73
85
97
109
121
133
145
157
U
)


T M
300
250
200
150
100
50
0
M
)
1
13
25
37
49
61
73
85
97
109
121
133
145
157
U )


T M
1500
1000
500
0
M
0
)
1
14
27
40
53
66
79
92
105
118
131
144
157
U
)


0M S
/9
/
/5
/
/9
/
/5
/
6
5
9
S




EVL internal report.