Applying the DiffServ Model on Cut-through Networks

Sven-Arne Reinemo*, Tor Skeie, and Olav Lysne
Simula Research Laboratory
P.O.Box 134, N-0325 Lysaker, Norway
Phone: +47 67 82 83 93 Fax: +47 67 82 83 01
{svenar | tskeie | olav.lysne}@simula.no

Keywords: QoS, DiffServ, SAN, Flow-control, Irregular networks, Minimal routing

Abstract— Understanding the nature of traffic in high-speed communication systems is essential for achieving QoS in these networks. A first step towards this goal is understanding how basic QoS mechanisms work and affects the network predictability before we introduce more complex mechanisms such as admission control. In this paper we analyse the effect of a DiffServ inspired QoS concept applied to virtual cut-through networks. The main findings from our study are that (i) throughput differentiation can be achieved by weighting of virtual lanes (VL) and by classifying VLs as either low or high priority, (ii) the balance between VL weighting and VL load is not crucial when the network is operating below the saturation point, (iii) jitter, however, is large and good jitter characteristics seems unachievable with such a relative scheme.

I. Introduction

THE Internet has today evolved into a global infrastructure supporting applications such as streaming media, business to business solutions, e-commerce and network storage. All of these applications are part of what we call Internet computing and they must handle an increasing volume of data demanding predictable transfer. In this context the provision of Quality of Service (QoS) is becoming an important issue. In order to keep pace with computer evolution and the increased burden imposed on data servers, application processing, etc., created by the popularity of the Internet, we have through the years seen several new technologies proposed for System and Local Area Networking (SAN/LAN) [3, 4, 7, 14, 24]. Common for this body of technologies is that they rely on point-to-point links interconnected by off-the-shelf switches that support some kind of back-pressure mechanism. Besides, most of the referred technologies also adhere to the cut-through or wormhole switching principles - only Gigabit Ethernet is using the store-and-forward technique. For a survey of some relevant networking principles we refer to [6].

The Internet Engineering Task Force (IETF) has for several years provided the Internet community with QoS concepts and mechanisms. The best known are IntServ [8], RSVP [13], and DiffServ [5]. IntServ together with RSVP define a concept based on per flow reservations (signalling) and admission control to be present end-to-end. DiffServ, however, takes another approach assuming no explicit reservation mechanism in the interior network elements. QoS is here realized by giving data packets differentiated treatment relative to QoS header code information. With respect to the underlying network technologies QoS has to a less extent been emphasised - the key metrics here have solely been mean throughput and latency. To make end-to-end QoS possible over heterogeneous technologies this means that the lower layers should also have support for predictable transfer including the ability to interoperate with a higher level IETF concept. This issue is being challenged by emerging SAN/LAN standards, such as InfiniBand^{TM} [4] and Gigabit Ethernet [24] providing various QoS mechanisms.

Recently we have also seen several research contributions to this field. Jaspernite et. al. [9, 10] and Skeie et. al [15] discuss different aspects around taking control of the latency through switched Ethernet relative to the IEEE 802.1p standard aiming at traffic priorities. Another body of work is tailored to the InfiniBand^{TM} architecture (IBA) [1, 2, 12]. In [12] Pelissier gives an introduction to the set of QoS mechanisms offered by IBA and the support for DiffServ over IBA. In this approach the presence of admission control is assumed, hereunder a description of which bandwidth related matrices should be considered when processing QoS requests. Alfaro et. al builds on this scheme and present a strategy for computing the arbitration tables of IBA networks, moreover a methodology for weighting of virtual lanes (traffic classes) referring to the dual arbitrator (scheduler) defined by IBA [2]. The concept is evaluated through simulations assuming that only priority traffic requests QoS. In [1] Alfaro et. al also include time sensitive traffic, besides calculating the worst case latency through various types of switching architectures.

*Primary contact author.

DiffServ is foreseen to be the most prominent concept for providing QoS in the future Internet [11, 17]. DiffServ makes a distinction between boundary nodes and core (interior) nodes with respect to QoS features. Following the DiffServ philosophy no core switch should hold status information about passing-through traffic neither should there be any explicit signalling on per flow basis to these components. This means that within the DiffServ framework any admission control or policing functionality would have to be implemented by boundary (host) nodes or handled by a dedicated bandwidth broker (BB). The core switches are assumed to perform traffic discrimination only based on a QoS tag included in the packet header - all packets carrying the same QoS code will get equal treatment. From that viewpoint DiffServ is apparently a relative service model having difficulties giving absolute guarantees. None of the previous debated contributions comply with the DiffServ model. In [12] Pelissier, however, discusses interoperation between DiffServ and IBA on a traffic class and service level basis, but refer to RSVP with respect to admission control. The strategy proposed by Alfaro et. al has to recompute the IBA dual arbitrator (the packet scheduler) every time that a new connection is honoured [1, 2]. Such a scheme is not associable with DiffServ. Other QoS efforts include The Multimedia Router [20,21] and some earlier work in [19] and [22], but none of these fits well with the DiffServ philosophy.

In this paper we endeavour to provide QoS in cut-through networks by adhering to the DiffServ philosophy. We approach the problem by studying the provision of QoS without any explicit admission control mechanism. Empirically we carefully examine the sensitivity of different QoS properties under various load and traffic mixture conditions, hereunder assess the effect of back-pressure (flow-control). These experiments give valuable information regarding the QoS behaviour of cut-through networks when used as a pure relative service model. Specifically we study (i) the effect of using virtual lanes with a weighted arbitration scheme to do throughput differentiation, (ii) the robustness of a weighted arbitration scheme when virtual lane load and weight is unbalanced and (iii) the latency and jitter characteristics of virtual lanes with a weighted arbitration scheme.

The paper is organised as follows. In section II we give a basic description of our QoS architecture and routing algorithm, before we in section III present our simulation scenario. In section IV we present and discuss our results with regard to throughput, latency and robustness. And finally in section V we give some concluding remarks.

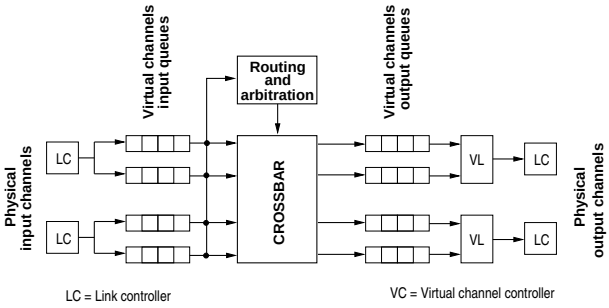


Fig. 1. Switch architecture

II. QoS architecture

THE architecture used in our simulations is inspired by InfiniBand^{TM}link layer technology [4] and is a flit based virtual cut-through switch. The overall design is based on the canonical router architecture described in [6].

A. Switch architecture

Our architecture is flit based and uses virtual cut-through (VCT) switching [23]. In VCT the routing decision is made as soon as the header of the packet is received and if the necessary resources are available the rest of the packet is forwarded directly to the destination link. If the necessary resources are busy the packet are buffered in the switch. In addition we use flow control on all links so all data is organized as flow control digits (flits) at the lowest level.

The switching core consists of a crossbar where each link has dedicated access (figure 1). Each link supports one or more virtual lanes (VL), where all of the VLs belonging to a link has multiplexed access to the crossbar. Each VL has its own buffer resources which consist of an input buffer large enough to hold a packet and an output buffer large enough to hold two flits to increase performance. VL arbitration is done at the input side to select which VL is allowed to send next. The VL arbitration is close to the InfiniBand^{TM}link layer arbitration and is described in more detail in section II-C.2. Link arbitration at the output link is done to select which input is the next to send to this output. Output link arbitration is done in a round robin fashion.

B. Layered shortest path routing

In our simulations we use a newly introduced routing algorithm called *Layered shortest path routing* (LASH) [16]. LASH is a minimal deterministic routing algorithm for irregular networks which only relies on the support of virtual layers to function. There is no need for any other functionality in the switches so LASH fits well with our simple approach to QoS.

The idea is that each virtual layer in the network has a set of source/destination pairs assigned to it,

in such a way that all source/destination pairs are assigned to exactly one virtual layer. In addition it makes sure that each virtual layer is deadlock free by ensuring that the channel dependencies stemming from the source/destination pairs of one layer do not generate cycles. An in depth descriptions of LASH is found in [16].

C. QoS mechanisms

Our switch architecture support QoS mechanisms like the ones found in the InfiniBand™architecture. InfiniBand™supports three mechanisms for QoS which are mapping of service level (SL) to VL, weighting of VLs and prioritising VLs as either low priority (LP) or high priority (HP). These mechanisms are described briefly in the following sections.

C.1 SL to VL mapping

A service level denotes what type of service a packet shall receive as it travels toward its destination. This corresponds to the packet marking approach described in DiffServ and is used to implement per hop forwarding.

In our architecture there is a one-to-many mapping between SLs and VLs since each SL belongs to a private virtual domain which might use one or more VLs to avoid routing issues such as deadlock. We use a model where there is no sharing of VLs, but where each subset of VLs are assigned a service level at startup without any possibility for change during execution. This excludes run-time reconfiguration, but makes the scheme simpler and is in line with the DiffServ philosophy.

C.2 VL priorities and weighting

After mapping SLs to VLs the individual VLs must be configured according to the demands for each SL. Each VL is classified as either LP or HP, and each VL receives a weight. Adding control VLs we get a three level scheme as follows

Level 1 On this level preemptive scheduling is used for control packets. Control packets have the highest priority and will preempt anything else.

Level 2 On this level preemptive scheduling is used for HP and LP traffic. VLs are split into two groups of high and low priority. HP packets will always preempt LP packets, but to ensure forward progress of the LP VLs a parameter called *Limit of High-Priority* (LHP) is used. The LHP is the maximum number of packets that can be scheduled on HP VLs before a packet *must* be scheduled on a LP VL if there is one waiting.

Level 3 Arbitration between individual VLs is done by a weighted fair arbitration scheme. Each VL is assigned a weight indicating the number of flits it is allowed to send when its turn occurs. VLs are scheduled in a round robin fashion.

When a VL is scheduled it is marked active and allowed to send one flit. Then the weight counter for this VL is decreased and the VL is rescheduled. The VL is rescheduled as long as its weight is larger that zero and no VL of higher priority wants to send. When the weight reaches zero it is reset and the next VL that has anything to send is set to active. Thus there is a flit level scheduling of VL across the crossbar, but there is no flit interleaving on the VL. Only one packet is allowed to hold a VL at a time.

SERVICE LEVELS					
#	DS Eq.	Load	BW[1]	UW[2]	Pri
1	EF	10	4	6	high
2	EF	15	6	1	high
3	AF	20	8	6	low
4	AF	25	10	1	low
5	BE	30	1	1	low

TABLE I
THE FIVE SERVICES LEVELS USED IN SIMULATION.

III. Simulation scenario

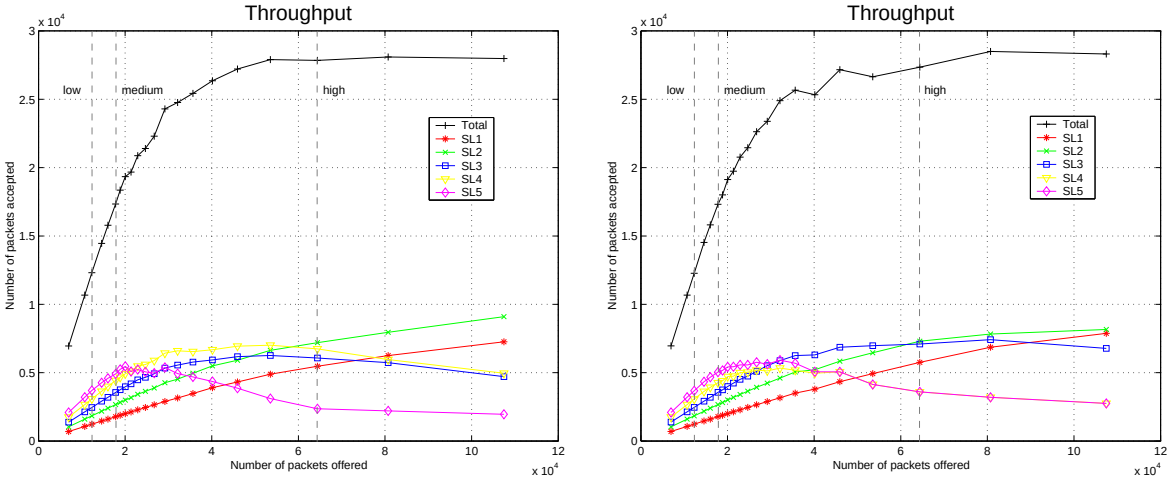
FOR evaluation of our approach to QoS we use a flit level simulator developed in house at Simula Research Laboratory.

A. Simulation methodology

In the simulations that follow, all traffic is modelled by a normal approximation of the Poisson distribution. We have performed simulations on networks of sizes 8, 16 and 32 switches, where each switch is connected to 5 end nodes and the maximum number of links per switch is 10. For each network size we have randomly generated 16 irregular topologies and we have run measurements on these topologies at increasing load.

The five different end nodes send traffic on five different service classes (table I) where SL 1 and 2 are considered to be of the expedited forwarding (EF) class in DiffServ terminology. SL 3 and 4 are considered to be of the assured forwarding (AF) class and SL5 is considered best effort (BE) traffic. We use LASH (see section II-B) as routing algorithm and random pairs as traffic pattern. In random pairs each source sends only to one destination and no destination receives packets from more that one source. The link speed is one flit per cycle, the flit size is one byte and the packet size is 32 bytes for all packets. Due to lack of space only results from networks with 32 switches are presented here.

[1]Balanced weight.
[2]Unbalanced weight.



(a) Balanced weights.

(b) Unbalanced weights.

Fig. 2. Throughput for a network with 32 switches.

IV. Performance results

THE performance results presented below are for topologies with 32 switches and 160 nodes, all other parameters are as described in section III-A unless otherwise noted.

A. Throughput

Our first results are from a balanced configuration under increasing load conditions. In the balanced configuration the weighting of each SL is according to the load on that SL, i.e. the SL with the highest load also has the highest weight[3]. This makes the configuration vulnerable to changing load conditions where a mismatch between applied load and VL weight might cause trouble. We take a closer look at this problem in section IV-B.

Figure 2(a) shows the throughput of each SL as well as the total throughput. As can be seen from the figure throughput differentiation works well. Throughput for all SLs is consistent with the weighting when the network is below saturation since there is enough bandwidth for everyone. When the network reaches saturation LP SL throughput is reduced as HP SLs preempts LP bandwidth which is consistent with the two-level priority scheme.

B. Robustness

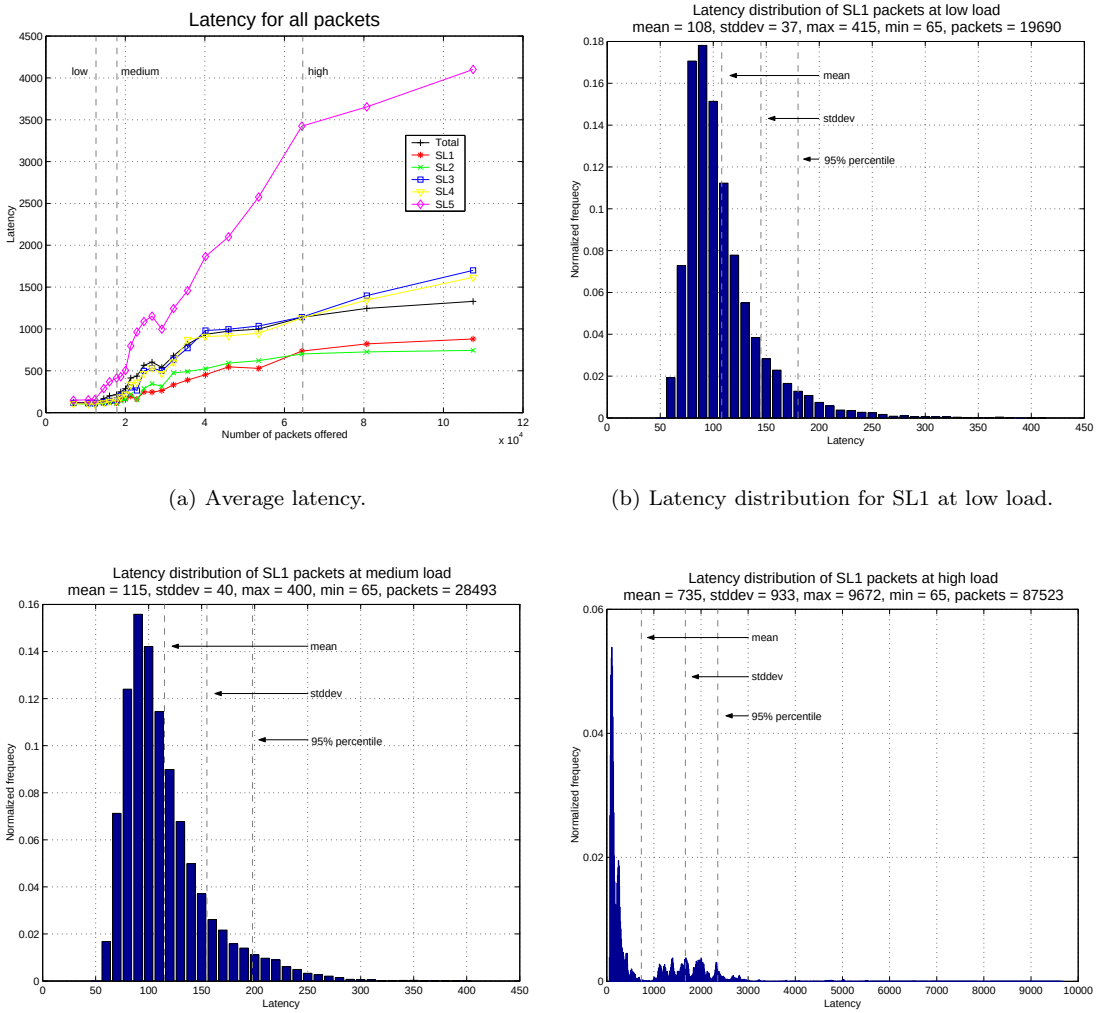
So far we have considered a balanced configuration where we require VL weights and SL load to be matched. This might not always be the case so we have looked at how well an unbalanced configuration performs. In this configuration the weights for SLs with the highest load have been swapped with the weights for SLs with the lowest load. Figure 2(b) shows the throughput of all SLs as well as the total throughput. From the figure we see that the performance is almost identical to the previous configuration as long as the network is below saturation. Only when the network reaches saturation has the mismatch between weighting and load become visible. As long as we are below the saturation level there is enough bandwidth for everyone and there is no problem fulfilling demands, but when the network gets saturated the weighting starts to work as we saw in the previous configuration. Only now the wrong SLs preempts the bandwidth because of the mismatched weights. In figure 2(b) this is visible for SL4 which should just above SL3, but is actually at the level of SL5.

From this we can conclude that the weighting configuration is *not* crucial with respect to QoS in a network working below saturation. It is crucial when the network is in saturation, but in a QoS setting we want to avoid a saturated network and only work below saturation. Thus the weighting configuration can be considered robust in a non-saturated network and the need for on the fly reconfiguration of the SL weights is not necessary. However, what we may need is some form of traffic control which can be achieved by a suitable admission control concept.

C. Latency

So far our relative QoS scheme is working well, but we will now discuss the performance with regard to network latency that we shall see introduces some

[3]Except from SL5 which is the best effort class and always has the lowest weight possible.



(a) Average latency.

(b) Latency distribution for SL1 at low load.

(c) Latency distribution for SL1 at medium load.

(d) Latency distribution for SL1 at high load.

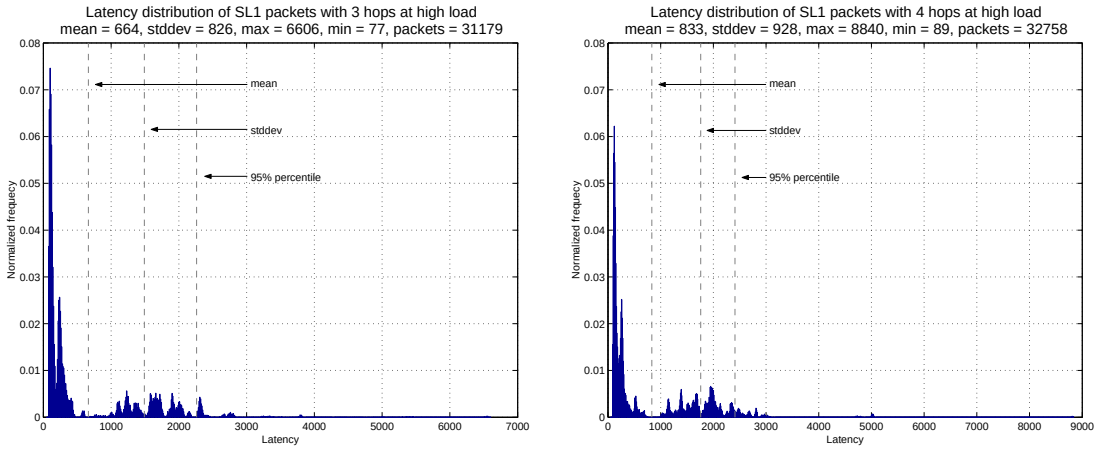
Fig. 3. Latency for a network with 32 switches.

problems.

Figure 3(a) shows the average network latency for each SL and for all traffic. As can be seen from figure 3(a) the latency is moderate as longs the load is well below saturation. Comparing it with figure 2(a) we see that LP SLs has a jump in latency at the same point as the throughput graph starts to fall off. The same effect can be seen for HP SLs at a higher load. The LP SLs are affected when we no longer can linearly increase the throughput. At this point the HP SLs starts to take away bandwidth from the lightly weighted LP SLs since these SLs have used most of the free bandwidth up to this point. Because more and more LP packets are preempted by HP packets we see a rise in latency for these SLs. As the network approaches saturation the same thing happens to HP SLs. Because the network is highly loaded the backpressure mechanism struggles to keep packets back and this introduces delays for all packets, thus increasing overall latency. Again we see a need to keep the network well below saturation, preferably in the linear area of the throughput graph, and the need for a suitable admission control mechanism. For HP traffic which is latency sensitive admission control can be used to keep the load so low that the latency properties are acceptable. LP traffic could be left out of an admission control scheme since this type of traffic is supposed to be bandwidth sensitive only, and not latency sensitive.

D. Jitter

Let us now turn our attention to the jitter characteristics. Figure 3(b) shows the latency distribution for SL1 traffic at the load level marked as *low* in figure



(a) 3 hops

(b) 4 hops

Fig. 4. Latency histograms per hop for SL1 traffic at high load

2(a). At this point the latency distribution is good. The mean, standard deviation and the 95 % percentile are marked with a dashed line in the graph. The distance between the mean mark and the standard deviation mark reflects the standard deviation. The histogram has a sharp peak and a short tail and the standard deviation is low. The 95% percentile is 180 so 95% of the packets has a latency of 180 or lower. If we move on to the *medium* load level things are only slightly worse. At this level we still have a low average latency and a 95% percentile at 198. Moving on to *high* load level things are getting much worse since we now have moved out of the linear area of the throughput graph. The jitter potential is substantial with a interquartile range of 1262 and the 95% percentile at 2356.

To investigate the jitter issue further figure 4 presents per hop latency histograms for SL1 traffic at high load with 3 and 4 hops which were the most frequent path lengths. The interesting part here is how much the latency properties increase when the path length is increased with one hop. The 95% percentile increases from 2262 to 2410 and the interquartile range increases from 1125 to 1478 as well as substantial increases for mean, standard deviation and maximum observed latency. The same behaviour can be seen for other path lengths as long as there are a substantial number of packets. This effect can be explained by the backpressure mechanism in use. When a packet somewhere in the network is delayed by backpressure the packet latency for this packet is increased, but the latency for the following packets is also affected. Thus the effect of backpressure in a node propagates throughout the whole network and introduces jitter.

Moreover, this will result in an exponential increase in the theoretical maximum latency. A simple expression to calculate the maximum network latency in a network of switches with only one VL per port and where the input buffer is one packet wide is as follows:

$$P = \sum_{sw=1}^{n} (nLinks - 2)^{sw}$$

Here P is the maximum number of packets we could end up waiting on, $nLinks$ is the number of links per switch and n is the number of switches in our path. We ignore two of the links because one is the link we are entering on and one is the link we are leaving on and they will not delay our packet. The expression for maximum network latency now becomes:

$$L = T \times P$$

Here L is the maximum network latency and T is the transmission speed in cycles. As the expression for P grows exponentially this will have significant impact on the jitter characteristics of the network. If we consider our network and a packet with path length 3 (as in figure 4(a)) we get the following:

$$L = T \times P = 32 \times 584 = 18688$$

In our simulations the maximum observed latency for a path length of 3 is 6606 so we have still some way to go before we reach the maximum. For a path length of 4 the maximum observed latency is 8840 while L is 149760.

However, in order to avoid this increase one should keep the network load at a very low level. This can

be achieve with a suitable *admission control* scheme as mention in the previous section.

Achieving low jitter in backpressure networks is a real challenge, and we plan to scrutinise this issue further to gain a better understanding and find better solutions to the problem.

V. Conclusion

UNDERSTANDING the nature of traffic in high-speed communication systems is essential for achieving QoS in these networks. In a first step toward this goal it is important to know how simple mechanisms works and affects the network before we introduce more complex mechanisms. In this paper we analyse the effect of a DiffServ inspired QoS mechanism applied to virtual cut-through networks. The main findings from our study are that (i) throughput differentiation can be achieved by weighting of VLs and by classifying VLs as either LP or HP, (ii) the balance between VL weighting and VL load is not crucial when the network is operating below the saturation point, (iii) jitter, however, is large and good jitter characteristics seems unachievable with such a relative scheme since the theoretical maximum latency grow exponentially. Even if the measured maximum latency does not show exponential behaviour its growth rate is still substansial.

We are currently examining some possible improvements to this approach. First, we are looking to add admission control to keep the network below saturation at all times. Second, we would like to address the backpressure problem so latency constrained traffic can be handled in a more appropriate way. Finally, we would like to analyse other configurations. Such as regular topologies, different traffic patterns and adaptive routing algorithms.

References

[1] F. J. Alfaro, J. L. Sanchez, J. Duato, and C. R. Das. A strategy to compute the InfiniBand arbitration tables. In *Proceedings of International Parallel and Distributed Processing Symposium*, April 2002.

[2] F. J. Alfaro, J. L. Sanchez, and J. Duato. A strategy to manage time sensitive traffic in InfiniBand. In *Proceedings of Workshop on Communication Architecture for Clusters (CAC)*, April 2002.

[3] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and Wen-King Su. Myrinet – a gigabit-per-second local-area network. IEEE MICRO, 1995.

[4] InfiniBand Trade Association. Infiniband architecture specification.

[5] Differentiated Services. RFC 2475.

[6] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks an engineering approach*, IEEE Computer Society, 1997.

[7] R. W. Horst. Tnet: A reliable system area network. *IEEE Micro*, 15(1):37–45, 1995.

[8] Integrated Services. RFC 1633.

[9] J. Jaspernite, and P. Neumann. Switched Ethernet for Factory Communication. In *Proceedings of* 8^{th} *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'01)*, 205–212, October 2001.

[10] J. Jaspernite, P. Neumann, M. Theiss, and K. Watson. Deterministic real-time communication with switched Ethernet. In *Proceedings of* 4^{th} *IEEE International Workshop on Factory Communication Systems (WFCS'02)*, August 2002.

[11] K. Kilkki. Differentiated services for the Internet. *Macmillian Technical Publishing*, 1999.

[12] J. Pelissier. Providing quality of service over InfiniBand^{TM} architecture fabrics. In *Proceedings of Hot Interconnects X*, 2000.

[13] ReSource ReserVation Protocol. RFC 2205.

[14] M. D. Schroder et.al., "Autonet: a high-speed, self-configuring local area network using point-to-point links," SRC Research Report 59, Digital Equipment Corporation, 1990.

[15] T. Skeie, J. Johannessen, and Ø. Holmeide. The road to an end-to-end deterministic Ethernet. In *Proceedings of* 4^{th} *IEEE International Workshop on Factory Communication Systems (WFCS'02)*, August 2002.

[16] T. Skeie, O. Lysne, and I. Theiss. Layered shortest path (LASH) routing in irregular system area networks. In *In proceedings of Communication Architecture for Clusters (to appear)*, 2002.

[17] X. Xiao and L. M. Ni. Internet QoS: A Big Picture In *IEEE Network Magazine*, 8–19, March/April 1999.

[18] J. S. Yang and C. T. King, "Turn-restricted adaptive routing in irregular wormhole-routed networks," in *Proceedings of the 11'th International Symposium on High Performance Computing (HPCS97)*, July 1997.

[19] Andrew A. Chien and Jae H. Kim, "Approaches to Quality of Service in High-Performance Networks," in *Lecture Notes in Computer Science*, vol. 1417, 1998.

[20] Jose Duato and Sudhakar Yalamanchili and Blanca Caminero and Damon S. Love and Francisco J. Quiles, "MMR: A High-Performance Multimedia Router - Architecture and Design Trade-Offs," in *HPCA*, pages 300-309, 1999.

[21] B. Caminero, C. Carrion, F. J. Quiles, J. Duato and Sudhakar Yalamanchili, " A Solution for Handling Hybrid Traffic in Clustered Environments: The MultiMedia Router MMR," in To appear in *Proceedings of IPDPS-03*, April 2003.

[22] M. Gerla and B. Kannan and B. Kwan and E. Leonardi and F. Neri and P. Palnati and S. Walton, "Quality of Service Support in High-Speed, Wormhole Routing Networks," in *International Conference on Network Protocols (ICNP'96)*, 1996.

[23] P. Kermani and L. Kleinrock, "Virtual Cut-through: A New Computer Communication Switching Technique," in *Computer Networks*, pages 267-286, no. 4, vol. 3, 1979.

[24] Rich Seifert, *Gigabit Ethernet,* Addison Wesley Pub Co., 1998.