Buffer Space Tradeoffs in Multi-hop Networks

Yanxia Jia, Ioanis Nikolaidis, Pawel Gburzynski
Department of Computing Science
University of Alberta
Edmonton, Alberta, CANADA T6G 2E8
{yanxia,yannis,pawel}@cs.ualberta.ca

ABSTRACT

We consider the problem of buffer space allocation in a multi-hop store-and-forward network and study the tradeoffs between the amount of buffer space alloted to the endpoints of a virtual circuit and that assigned to the core nodes, i.e., routers. Simulation results hint at the possibility that the single-path paradigm of implementing network-layer virtual circuits is fundamentally flawed (rather than merely inconvenient from the viewpoint of bandwidth allocation and routing flexibility). Despite its intuitive appeal (packet ordering, predictability of end-to-end delays, etc.), single-path forwarding neither results in efficient utilization of available buffer space from the point of view of the entire network, nor does it provide the best overall performance in terms of global bandwidth utilization and end-to-end quality of service. We demonstrate that the best quality of service for virtual circuits (in terms of drop rate) is achieved when 1. most of the buffer space in the network is at the destinations, and 2. a single virtual circuit explores multiple paths, essentially following the general principles of deflection routing.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Packet-switching networks

General Terms

Performance

Keywords

routing, multiple path routing, deflection

1. INTRODUCTION

The problem of resource allocation in contemporary networks, catering to a variety of interactive and bandwidth-hungry applications, is defined within the context of quality of service (QoS) requirements of those applications. These QoS requirements are usually specified in terms of bandwidth, packet loss, end-to-end delay, and jitter.

According to common wisdom, the most QoS friendly implementation of an end-to-end session involves a network-layer virtual circuit, whereby all packets of the session follow exactly the same path from source to destination. Intuitively, the deterministic character of such a connection makes it easier to set aside the right amount of resources at every intermediate node and predict what is going to happen when several virtual circuits pass at the same router. Consequently, most work on QoS-driven resource allocation focuses on path selection algorithms [1, 8, 9, 15], assuming that once selected the (single) path will be followed by all packets of the session. This approach essentially equates a transport-layer session with a network-layer virtual circuit, even if (as in the Internet) the network-layer virtual circuit is not explicit.

While moderate attempts at multi–path routing schemes found in the literature [5, 14] do demonstrate that a better load balancing can be achieved this way, they still restrict the selection of alternative routes based on a more or less explicit notion of a network layer session. This often tacit and automatic assumption about the inherent superiority of virtual circuits over datagrams has resulted in a complete oblivion of forwarding ideas based on deflection [3, 10] (which can be viewed as an extreme implementation of unkempt spontaneous routing) and has brought us the paradigm of ATM networks, which, until not so long ago, was viewed by many authors as the ultimate solution to all problems of networking.

Even if one was to consider a pure datagram network-layer protocol, such as IP, where virtual circuits are absent, one can still identify a strong insistence on deterministic-path forwarding. As keeping track of TCP sessions in the network (IP) layer is neither easy nor natural, this insistence practically removes all flexibility from the transport layer. This is because, with the splitting of a single session over multiple paths considered harmful, there seems to be no alternative to forwarding all IP traffic between a given host pair via the same route. This in turn effectively kills all opportunities for load balancing.

Moreover, a conceptually similar approach to ATM virtual circuits survives to this day in the guise of Multi-Protocol Label Switching (MPLS) [6] whereby bundles of transport layer flows are grouped (for the sake of efficiency) together in logical virtual circuits and are routed together. Unsurprisingly, MPLS further limits the potential for load balanc-

CNSR 2003 Conference, May 15-16, 2003, Moncton, New Brunswick, Canada.

ing that could exist between transport layer flows. While the potential for load balancing at the level of MPLS "bundles" still exists, the fact that it can only be accomplished by rather costly routing decisions involving route (re-)calculations, results in MPLS being less than attractive for responsive load balancing.

In this paper, we consider a flexible routing model based on deflection, whose degree is affected by the amount of buffer space available at the router. Our objective is to investigate how the QoS perceived by a transport-layer session depends on the way the buffer space available globally in the network is partitioned among the routers and the destinations. We conclude that with a global view of network resources, multiple alternative paths explored by different packets of the same transport-layer session need not be harmful. Just the opposite, they may in fact improve the utilization of those resources and, *at the same time,* improve the critical QoS characteristics of isochronous sessions. This is in contrast to what most people seem to believe. While one can easily agree that the increased routing flexibility naturally translates into a more balanced utilization of network resources, realizing that this approach may also imply better (more predictable) end-to-end delivery is a less obvious (and somewhat counterintuitive) step to take. This seems to confirm the speculations expressed in [2] and suggests that single-path forwarding is an inherently flawed routing strategy.

The rest of the paper is organized as follows: Section 2 introduces the basic design tradeoffs related to the placement choice for storage (buffer space) in the interior of the network versus the periphery. Section 3 provides a simplified network model for the sake of exploring the tradeoffs using a quantitative framework. Section 4 reviews the relevant simulation results and observations. Finally, conclusions and avenues for further research are summarized in Section 5.

2. THE TRADEOFF

Consider a router within the network core that is about to forward a packet belonging to a transport-layer session. Regardless of the assumed routing paradigm, the complete list of options regarding the fate of this packet consists of the following possible outcomes:

1. The packet is queued for transmission on the "best" output port (offering the "most attractive" route to the destination).
2. The packet is dropped, e.g., because of the lack of buffer space at the router.
3. The packet is queued for transmission on an output port that is considered a secondary choice (by the assumed route preference scheme).

With the single-path forwarding paradigm, the third possibility is excluded. The optimization effort regarding the utilization of network resources is thus directed toward a precise description of what is meant by the "best" route to the destination, as well as determining the right packet scheduling policy at the router. The latter can be interpreted as part of the buffer management scheme, as it also prescribes the packet dropping rules.

If the third option is admissible, an alternative to dropping a packet (or sending it over the congested preferred path) is to forward it via a suboptimal route. The most serious consequence of this decision is that the packet may (legitimately) arrive at the destination out of order. Thus, to consistently play back the received packets as fragments of the transport-layer session, the recipient must use a reassembly buffer [4, 13, 12] to re-order and possibly delay packets arriving out of schedule.

Consider a session with some specific QoS expectations carried out between a source S and destination D. Suppose that the global forwarding scheme of the network excludes option 3, i.e., the session path is fixed. A router R along the session's path may drop a packet if it runs out of buffer space, but all packets eventually delivered to D are going to arrive in order. Consequently, D needs no reassembly buffer to play the session back, although, depending on the session type, it may still need some buffer space to smooth out the jitter caused by variable buffering delays at the routers. Larger buffers at the routers will result in a lower packet dropping rate perceived by D, although they may increase the jitter, which, at least for some session types, may render the packets useless upon arrival. Depending on the scheduling policy (or policies) adopted by the routers, late packets may also be identified (and dropped) before they reach their destinations.

If option 3 is admissible, packets may arrive at D out of order, and D may have to reassemble them, for which task it may need some extra buffer space. But with this option, R is able to carry out its duties with less buffer space than in the previous scenario. This is because now R has an alternative to dropping a packet. Consequently, it is possible that the reduced amount of buffer space at the router will be compensated by the increased amount of buffer space at the destination.

In [2], it is argued that deflecting instead of dropping may be a fundamentally better approach from the global viewpoint of network resource management. First, managing the private per-session playback buffer at D is considerably simpler (and better defined as a problem) than managing the shared buffer space at R in the face of multiple and diverse sessions passing through the router and (possibly) its multiple scheduling policies. In contrast to R, D applies the buffer to a single session at the exact point where its delivered QoS parameters can be monitored with ultimate fidelity and authority. Thus, it can easily, consistently, and meaningfully adjust the buffer size to compensate for occasional fluctuations of the perceived QoS measures. Second, if the session can put up with packets arriving out of order (e.g., the packets can be processed as independent datagrams), D does not have to bother with reassembly buffers, while R would still try to "fix what ain't broke" and buffer the packets in its effort to provide for (unneeded) in-order delivery. This is because R doesn't know any better: even if it differentiates some elements of the service (the scheduling policy), this one element (i.e., the individual route of a datagram) offers no degree of freedom.

3. THE NETWORK MODEL

Our experiments reported in this paper can be viewed as the first step aimed at putting the above speculations on a formal ground. As the routing approach in our network model, we use asynchronous deflection, somewhat similar to that described and analyzed in [7], but admitting limited buffers at the routers. In our model, no packet is ever

dropped at a router. When a packet arrives for forwarding and there is no buffer space available to queue it for transmission on the preferred output port, the packet is directed to an alternative output port instead of being dropped. This way, some packets that would have to be dropped by a conventional router are now likely to reach their destinations via alternative paths.

Each node ranks its repertoire of alternative paths using the approach described in [7], and limiting the choice to 4 paths. In essence, it calculates the four shortest first-hop-disjoint paths to each destination, with the shortest path considered most attractive. Obviously, to offer alternatives from the viewpoint of routing, the different paths cannot share their first hop.

The buffer space available at a router is partitioned among all the output ports, such that each port is assigned the same fixed amount. Every time a packet arrives at the router, it will be directed to the best output port with available buffer space. The degree of deflection in this model can be adjusted by modifying the amount of buffer space at the router. In particular, single path routing can be viewed as the limiting case of deflection routing with infinite buffers. Although no packets will ever be dropped in this model, the late arrivals of excessively delayed packets will render them useless at the destinations.

One may be somewhat concerned about the inflexibility incurred by the rigid partitioning of buffer space at the router. This approach simplifies the model and seems to be acceptable in a scenario involving regular network configurations and balanced traffic patterns. Besides, one can easily see that the conclusions from our experiments cannot be reversed (and are likely to be amplified) when a more flexible buffer allocation scheme is used.

Each node in our network behaves both as a router and a host, i.e., a source and/or a destination of a traffic session. The total amount of buffer space in the network is equal to $B + b$, where B denotes the amount of space assigned to the destinations (to be used for reassembly buffers) and b stands for the amount of storage available at the routers. Intentionally, $B + b$ remains fixed in a given experimental setup, while the ratio B/b determines the adjustable balance between the two categories of storage.

The network caters to an isochronous application described by a Pulse Code Modulation (PCM) voice traffic model, whereby 53-byte packets (corresponding to ATM cells) are sent at the average rate of 64Kbps. Their actual arrival process is Poisson. We look at the behavior of a selected source/destination (S/D) pair involved in an isochronous session, with the remaining nodes uniformly contributing Poisson-distributed background traffic.

We consider perfectly regular 4-connected torus networks with sizes varying from 4×4 to 14×14 nodes. All links have the same bandwidth of 1Mbps and the length of 1000km. We also investigated the case where link lengths are different and obtained similar results. While such networks may seem large, one should notice that geographically smaller networks can only be more advantageous for deflection (owing to a smaller variance in multi-hop propagation delays), which will result in more pronounced conclusions regarding the observed tradeoffs. Also, the perfect regularity of the topology and the uniformity of the background traffic allow us to focus on the essence of the observed phenomena without having to worry about the multitude of parameters describing more realistic configurations.

Our simulator keeps track of six performance measures related to the voice session. The *loss rate* is equal to the ratio of the number of packets discarded at the destination to the total number of packets transmitted by the source. Since a packet cannot be dropped at a router, loss can only occur at the destination—in two possible scenarios. First, it may happen that when the packet arrives, the reassembly buffer is full and there is no way to store the packet until its scheduled playback time. Second, the packet may arrive too late, i.e., after its playback time, in which case the reassembly buffer cannot help, even if storage is available.

The *network delay* of a packet is measured as the interval separating the packet's generation at the source and its arrival at the destination. It is composed of the queuing delay and the propagation delay, the latter of which also includes the (re)transmission delay experienced by the packet. The *playback lag* represents the time elapsing after a packet arrives at the destination and before it is played back. It reflects the pure impact of the reassembly buffer. The *end-to-end delay* captures the overall processing time of a packet within the network counting from the moment the packet is generated at the source, until its playback at the destination. It is the sum of the network delay and the playback lag.

4. THE RESULTS

The necessarily limited size of this paper allows us to present only a small fragment of the large collection of results from our simulation experiments. As long as the network size is nontrivial (bigger than 4×4), all experiments produce results that are highly consistent in qualitative terms. Thus, we illustrate our observations with the 5×5 network, which is the smallest configuration in which the described tradeoffs are clearly visible.

A single source/destination node pair is selected for a voice session, and the remaining nodes are uniformly selected to generate Poisson background traffic with the average rate of 40Mbps, unless otherwise specified. The simulated time period is $400,000$ms, corresponding to over $60,000$ packets. Six independent experiments have been run for each data point.

Figures 1– 6 show how the observed performance measures in the network depend on the partitioning of the global buffer space between the routers and destinations. Every single curve corresponds to a specific fixed total amount of buffer storage ($B + b$) and is a function of its allocation (B/b).

Starting from Figure 1, we can see that the observed drop rate tends to decrease as the mass center of the buffer space is shifted from the routers to the destination, then flattens for a while, and finally increases sharply. What we see is two counteracting phenomena in action. The reduced amount of buffer storage at the routers results in more packets being deflected (and misordered), while the increased size of the reassembly buffers compensates for the misordering and makes it possible to reconstruct the session without dropping packets. It appears that below a certain B/b threshold this compensation is more beneficial than the increased level of deflection is harmful. According to Figure 1, a workable low-loss regime falls roughly within the range $-1 < \log(B/b) < 4$, which translates into $0.4 < B/b < 55$. This means that we have a large selection of B/b resulting in

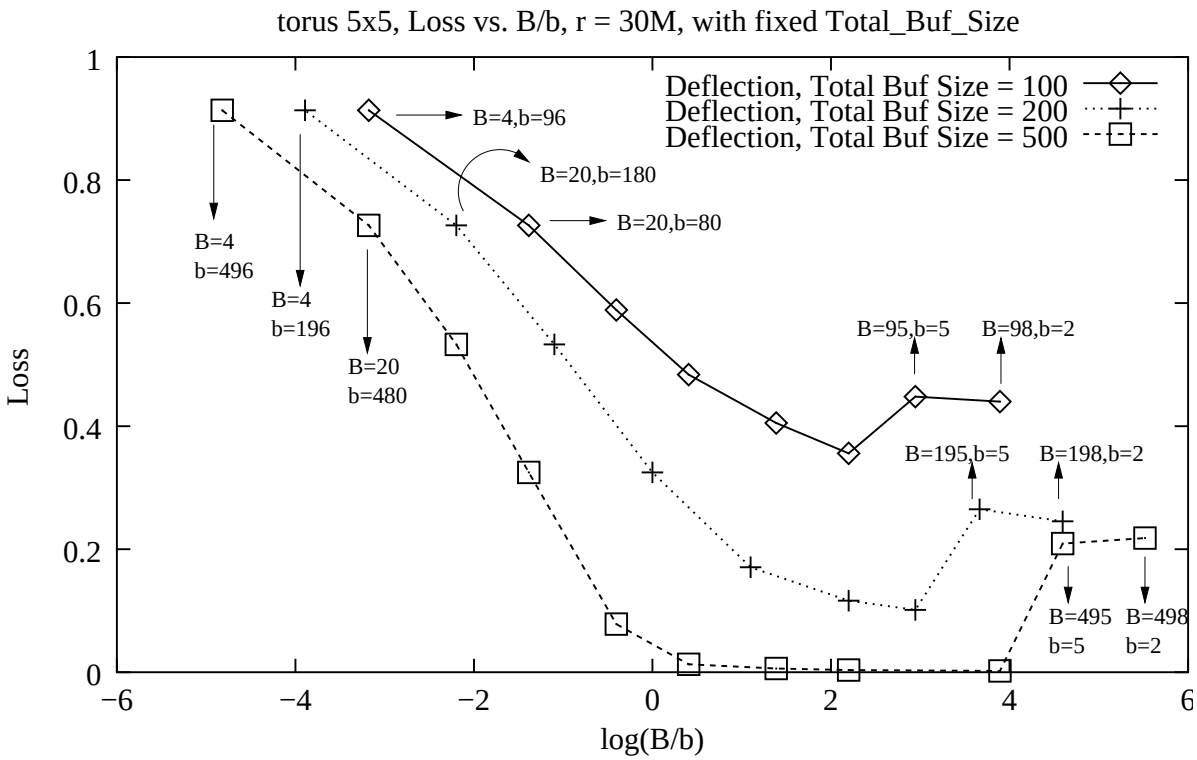

Figure 1: Loss rate.

an acceptably low loss rate, which widens as the total buffer space ($B + b$) becomes larger.

The negative impact of the reassembly buffer on the QoS measures perceived by the voice session consists in increasing the end-to-end delay, as shown in Figure 2. The end-to-end delay is composed of the network delay (Figure 3) and the playback lag (Figure 4). According to these two graphs, the playback lag is the dominating factor, and we observe an increasing end-to-end delay. The playback lag increases along with B because the destination buffer has to be filled up to a certain fraction of its total capacity before playback can commence. (This fraction was set at 80% based on the previous work [12].) The decreasing trend of the network delay is not surprising because, between its two components, the queuing delay (Figure 5) and the propagation delay (Figure 6), the former is the dominating factor.

In the following, we only focus on the loss rate and the end-to-end delay because these two QoS metrics are directly perceived by the end users. Figures 7 and 8 show that these two performance measures can be traded to some extent. Each curve represents a single B/b ratio, with the total amount of buffer space ranging from 100 to 1000 packets. We find that, for a wide range of the total amount of buffer space, some B/b ratios (e.g., 0, 1, 2, and 3 in Figure 7), offer consistently acceptable loss and delay measures. It's also interesting to note that a large ratio of B/b results in a better combined loss and delay performance. In other words, a large destination buffer combined with a small buffer at the router is more likely to provide both a satisfactory loss rate and an acceptable end-to-end delay.

The relatively small amount of buffer space at the routers at which they appear to perform satisfactorily, and the sharp

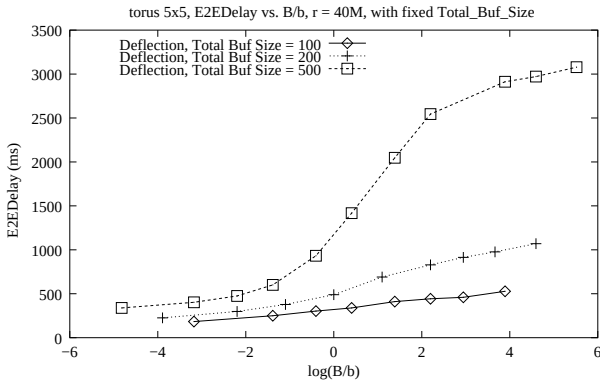

Figure 2: End-to-end Delay.

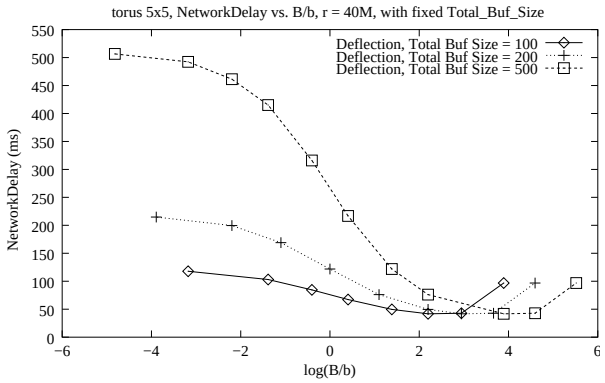

Figure 3: Network Delay.

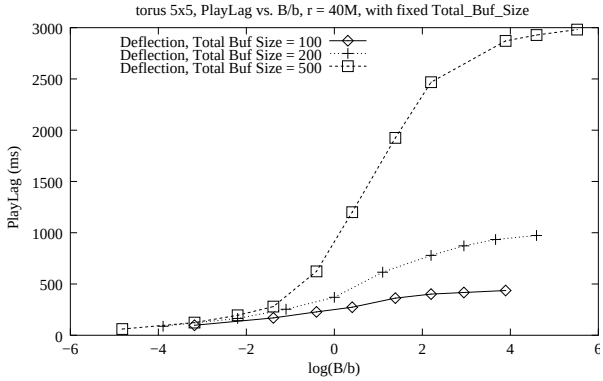

Figure 4: Playback Lag.

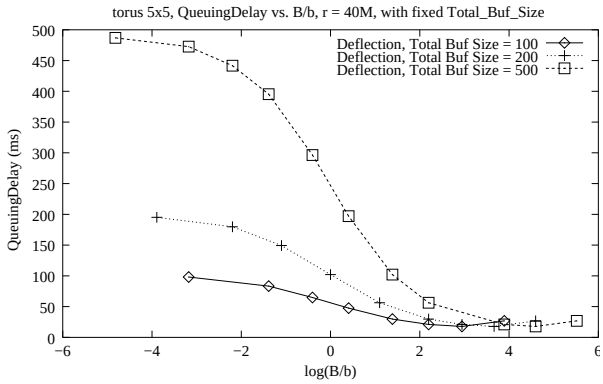

Figure 5: Queuing Delay.

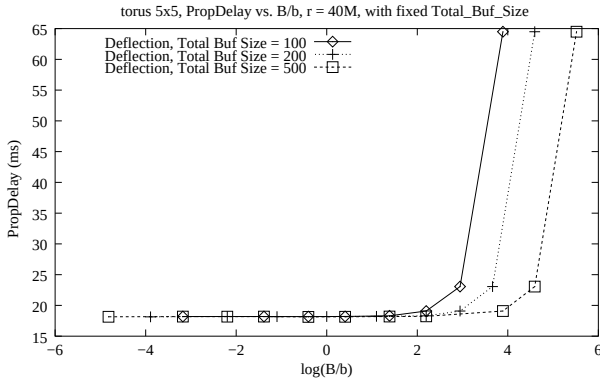

Figure 6: Propagation Delay.

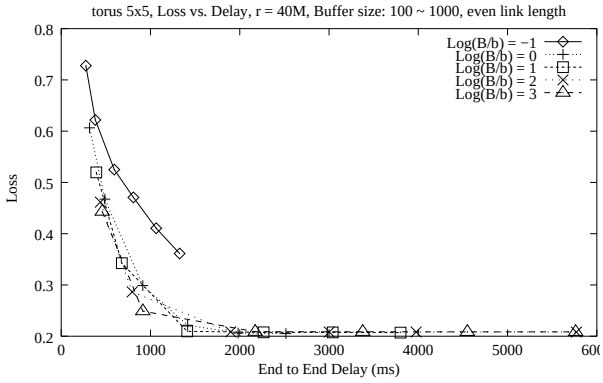

Figure 7: Loss rate vs. end-to-end delay, even link lengths

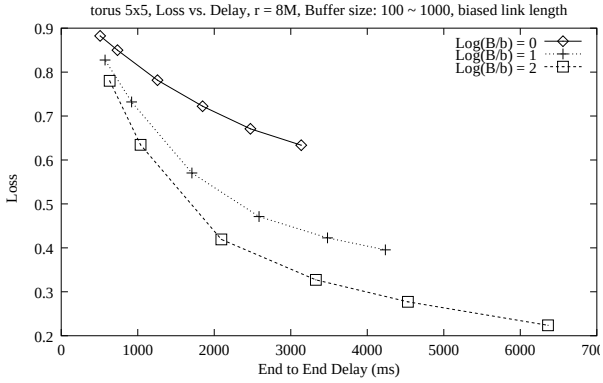

Figure 8: Loss rate vs. end-to-end delay, biased link lengths

increase in the drop rate when that small amount is reduced below a certain minimum, are consistent with the observations made in [11]. In that study, it is shown experimentally that a moderate amount of buffer space available to the routers tends to drastically improve the maximum throughput of a deflection network, bringing it quickly to a level comparable to that of a network with infinite buffers. In confrontation with our results, this seems to suggest that a single-path router with a large amount of buffer space is doubly misconfigured: it should be using little buffer storage while following alternative paths.

5. CONCLUSIONS

Our results suggest that deflection as a routing concept is less harmful than it would seem at first sight. From the global point of view of the entire network, the reassembly buffer is not a serious problem (and does not represent a new resource requirement) because its introduction reduces the resource requirements at the routers. Besides, the destination, being well aware of the specifics of its session, should be able to make better use of the reassembly buffer than a router having to cope with multiple and essentially unknown streams of traffic.

One standard argument against deflection networks is that the alternative routes incur excessive jitter, which has a detrimental impact on the performance of isochronous sessions. Note, however, that by buffering packets that can-

not be forwarded immediately, store-and-forward networks hardly solve this problem. While a deflection network can lose packets that fall outside the window provided by the reassembly buffer, a store-and-forward network can drop packets because of the lack of storage at the routers, or because those packets have been delayed too much to be useful. There is no fundamental difference at this level.

The issue of packet reassembly is often misguided (and brought forward as an erroneous argument against multiple-path routing) because of the insistence of some legacy applications on viewing their sessions as ordered sequences of packets. If we look carefully at those communication scenarios that truly require the preservation of packet ordering, we see that they fit into three categories:

- Sessions that could be carried out with packets arriving in any order (e.g., file transfers); they enforce packet ordering because the applications have been (unnecessarily) designed that way. Symptomatic of the ralization that such ordering is unnecessary and over–restrictive, is the recent availability of applications the split single ftp file transfers into multiple concurrent transfers of parts of the file, with the hope of reducing total download times.

- Sessions involving relatively short transfers (e.g., a piece of text to appear on the screen), which can be reassembled in a trivially small buffer space. There is no compeling reason for considering either single–path or deflection as a more preferred approach for this subset of applications.

- Long sustained isochronous streams (e.g., voice, video), which typically accept a non-zero packet loss and thus can be reasonably reconstructed within limited-size reassembly buffers. Note that in this category, the store-and-forward single-path approach does not guarantee zero packet loss. We thus argue that, by being inherently loss–free, deflection routing possesses a definite advantage.

Let us consider the Internet, and TCP in particular. Due to the increasing transmission rates of links, most hosts today are capable of handling TCP sessions with relatively large bandwidth × delay products, which translate into large *advertised* TCP receiver windows. The receiver window plays the role of a large reassembly buffer capable of (a) re-ordering packets potentially arriving out of order, and (b) holding packets beyond "gaps" caused by losses, while waiting for retransmissions to fill those gaps. However, the effectiveness of a large reassembly buffer to facilitate (b) is at least debatable, mostly because the dynamics of TCP constrict the window after a loss (and TCP's approach to window adjustments is quite conservative in general). Thus, case (b) provides reduced performance dividends but is typical in IP-based networks, where traditional single path routing is used. We argue that TCP would gain the full potential of receiver reassembly buffers if the balance was tilted in favor of case (a), which can be accomplished by deflecting packets when congestion occurs, instead of dropping them.

More experiments are required to verify the above speculations and put them on a more formal ground. Specifically, we need a better insight into the observed phenomena that would let us extrapolate them onto realistic irregular topologies and non-uniform traffic patterns. This is a natural direction for our further study.

6. REFERENCES

[1] G. Apostolopoulos, R. Guerin, and S. K. Tripathi. Quality of Service Routing: A Performance Perspective. In *Proceedings of SIGCOMM'98*. ACM, September 1998.

[2] C. Baransel, W. Dobosiewicz, and P. Gburzynski. Routing in Multi-hop Switching Networks: Gbps Challenges. *IEEE Network Magazine*, 3:38–61, 1995.

[3] F. Borgonovo and L. Fratta. Deflection Networks: Architectures for Metropolitan and Wide Area Networks. *Computer Networks and ISDN Systems*, 24:171–183, 1992.

[4] A. Choudhury and N. Maxemchuk. Effect of a finite reassembly buffer on the performance of deflection routing. In *Proceedings of the International Conference on Communication(ICC)*, volume 3, pages 1637–1646, 1991.

[5] I. Cidon, R. Rom, and Y. Shavitt. Multi-Path Routing Combined with Resource Reservation. In *Proceedings of IEEE INFOCOM'97*, pages 92–100. IEEE, April 1997.

[6] B. Davie and Y. Rekhter. *MPLS Technology and Applications*. Morgan Kaufmann Publishers, 2002.

[7] P. Gburzynski and J. Maitan. Deflection Routing in Regular MNA Topologies. *Journal of High Speed Networks*, 2(2):99–131, 1993.

[8] Y. Jia, I. Nikolaidis, and P. Gburzynski. Multiple Path Routing in Networks with Inaccurate Link State Information. In *Proceedings of IEEE International Conference on Communications (ICC'2001)*, June 2001.

[9] Q. Ma. *Quality-of Service Routing in Integrated Services Networks*. PhD thesis, Carnegie Mellon University, January 1998.

[10] N. Maxemchuk. The manhattan street network. In *Proceedings of GLOBECOM'85*, pages 255–261, 1985.

[11] N. Maxemchuk. Comparison of deflection and store-and-forward techniques in manhattan-street network and shuffle-exchange networks. In *Proceedings of IEEE INFOCOM'89*, pages 800–809, 1989.

[12] W. Olesinski and P. Gburzynski. Real-time traffic in deflection networks. In *Proceedings of WMC'98: Communication Networks and Distributed Systems, Modeling and Simulations*, pages 23–28, January 1998.

[13] W. Olesinski and P. Gburzynski. Service guarantees in deflection networks. In *Proceedings of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems(MASCOTS'01)*, pages 267–274, August 2001.

[14] K. N. S. Chen. Distributed QoS Routing with Imprecise State Information. In *Proceedings of the 7th International Conference on Computer Communications and Networks (ICCCN '98)*, October 1998.

[15] X. Yuan and A. Saifee. Path selection methods for localized quality of service routing. In *Proceedings of The 10th International Conference on Computer Communications and Networks (ICCCN 2001)*, October 2001.