Control Transmission Pace at IP Layer to Avoid Packet Drop

Guojun Jin

Lawrence Berkeley National Laboratory
1 Cyclotron Road, Berkeley, CA 94720
g_jin@lbl.gov

Haina Tang

Chinese Science and Technology Network Center
Beijing, China
tanghn@cstnet.cn

Abstract — **Avoiding packet loss is critical for time sensitive network applications, such as multimedia streams for video/voice. Delaying and dropping low priority packets to ensure high priority and time sensitive data stream deliver during network congestion is a basic QoS (quality of service) mechanism over current network infrastructure. This mechanism works if time sensitive data stream is the minority of the network traffic and if the network is not very congested. The methodology of dropping low priority data will not scale when time sensitive data stream uses high percentage of network bandwidth. This is because bandwidth required by video/audio applications can vary in very wide range when real-time data becomes majority network traffic, that is, television (TV), telephone, visual telephone, videoconferencing, gaming, and other video/audio based applications are all deployed on Internet. Then, what is the proper percentage of bandwidth to reserve? and which packets should be dropped if available bandwidth is less than demanding? A major issue is that letting bottleneck routers drop packets is not a proper methodology to guarantee quality of service. If packets cannot be delivered due to exhausted network bandwidth, these packets should be tossed as earlier as possible to reduce bandwidth waste or should be delayed at transmission hosts for later transmission. Also, applications should have right to selectively toss data for enhancing service quality, rather than let routers randomly drop packets. Therefore, mechanisms to avoid packet drop need to be deployed in Internet infrastructure. This paper studies how well priority (class) based traffic shaping can help time sensitive data delivery, addresses technology of packet drop avoidance (PDA), and shows how packet drop avoidance mechanism improves real-time applications' performance by reducing bandwidth waste, packet delay and loss. This paper then addresses why PDA should be deployed in Internet protocol (IP) layer.**

Keywords — **Packet Drop Avoidance, PDA, Loss, Delay, Network, Bandwidth, Measure, Quality of Service, QoS, Performance**

I. Introduction

Packet-switching technology provides flexible and easy management in current Internet routing system comparing with circuit-switch technology. However, packet loss is still a major issue that hurdles high-speed network utilization and performance, and affects quality of real-time network services. At transport layer, transmission protocols use packet loss as congestion signal to prevent further packet loss. Where in network layer, routers delay (queue) and drop packets to overcome congestion or to ensure high priority packets passing through the router as fast as possible. Dropping packets seems a necessary entity in current network infrastructure. A better solution to control transmission rate and to ensure high-priority services on networks is to determine what is the available network bandwidth, and sends packets for applications at or below the available bandwidth. The other reason that packet loss occurs is because all transport protocols pace packets out on their own judgment. For example, when different applications use different protocols such as VOIP (voice over IP), TCP and UDP, to communicate the same destination host, VOIP and TCP use different rate pacing algorithms, where UDP may not pace its transmission rate at all. Under this circumstance, neither VOIP nor TCP can properly control and ensure their transmission pace since non rate controlled UDP traffic can ruin other protocol pacing as long as QoS is not deployed in full mesh on entire networks. Network engineers have been aware this issue, and people have made effort on revising protocols. For example, DCCP (Datagram Congestion Control Protocol)[9] is studied for restricting UDP pace. However, individually pacing any single protocol, like TCP, without consulting with other protocols will not properly balance network load and smooth packet transmission. Therefore, to avoid packet loss, all traffics flowing toward a bottleneck router or the same destination should have the same pacing control mechanism to balance the aggregated stream rate. That is, the pacing control needs to be done in a common network layer.

In order to control transmission rate for all transmission protocols (at transport layer), the pacing mechanism has to be implemented in network layer (IP layer) because not only will all traffics from transport layer pass through the network layer, but also network layer can tell which traffic goes to where. This is one of design characteristics in packet drop avoidance (PDA). Since PDA only needs to be implemented at end host (no router software/hardware need to be changed), the implementation will not affect standard IP specification on routers. That is, embedding PDA in IP layer in end host kernel will be fully compatible with standard IP specification. [8] describes packet drop avoidance (PDA) and shows results from PDA deployed on a sender host for reliable data transfer. The result demonstrates how packet drop avoidance (PDA) effectively avoid packet loss in general data transmission. In this paper, we will show how PDA improves quality and performance for time sensitive network applications, and compare PDA with QoS on real-time application's demanding. This paper then addresses why PDA should be deployed in layer 3 (network layer), what other reasons are necessarily for controlling pace in network layer, and how the design lays out.

II. Real-time Application Demand and Quality of Service

Delay, jitter and packet loss are extremely important network characteristics for real-time network applications. Current network service model does not cater these real-time application requirements. A number of quality of service (QoS) models, such as Differentiated Services (DiffServ)[3], Low Latency Queuing (LLQ), Weighted Fair Queuing (WFQ), Multiprotocol Label Switching (MPLS)[5][6], have been proposed to improve the quality of real-time application services. In this section, we study how QoS can improve real-time application service quality, and analyze the QoS limitation.

Our QoS study aimed at two areas: 1) dedicated end-to-end QoS path, and 2) partially enabled QoS path. Fig. 1 is our network testbed, and we use SCNM (self-configuring network monitor — a super set of tcpdump)[7] to capture packets and to analyze the delay, jitter, loss, and other network characteristics. Followings are our QoS analysis steps:

1.) Real-time (RT) traffic encounters overcapacity cross traffic (1~2 Gb/s) at a GigE (1Gb/s) router without QoS set up
2.) RT traffic encounters overcapacity cross traffic (same as in step 1) at a GigE router with QoS (LLQ) enabled
   2a.) RT traffic is 10% of the QoS router capacity
   2b.) RT traffic is 55% of the QoS router capacity
3.) 100 Mb/s RT traffic encounters 600~700Mb/s cross traffic at a non-QoS switch, than passes through the QoS router with overcapacity cross traffic described in step 1

Generic testbed set up: a continuous RT stream is sent from a video server to a video client via main router (the

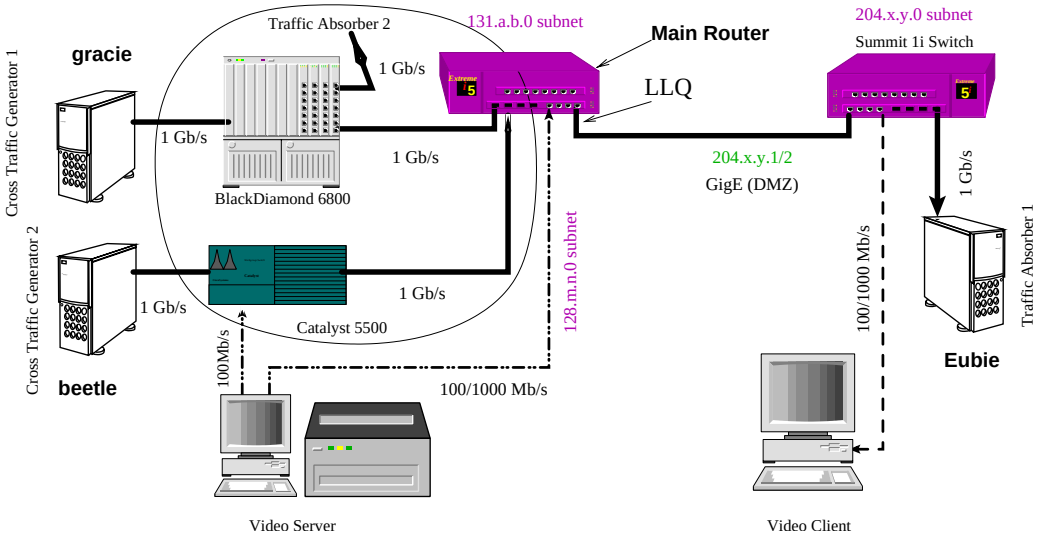

Fig. 1 Quality of Service Network Testbed

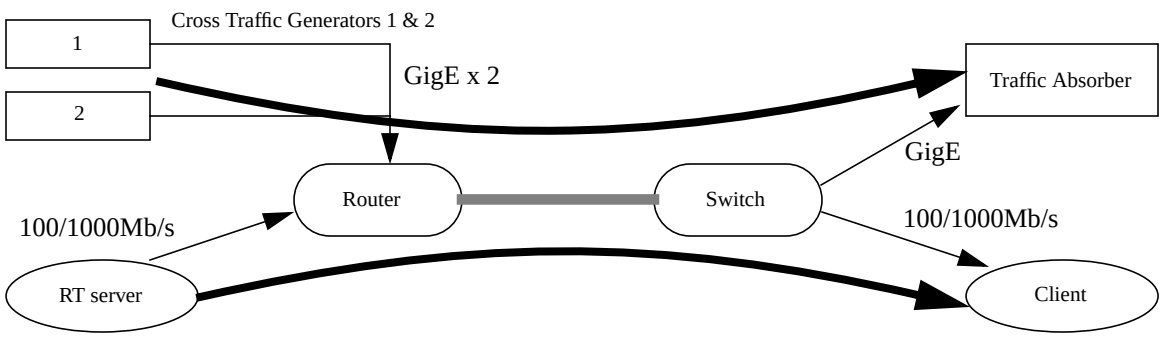

Fig. 2 Traffic paths for analysis step 1

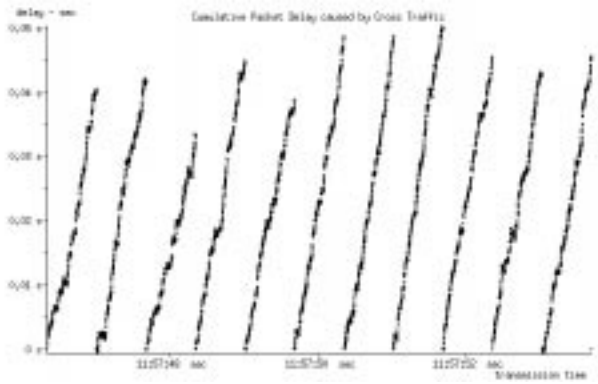

Fig. 3 Cumulative Packet Delay without QoS (analysis step 1)

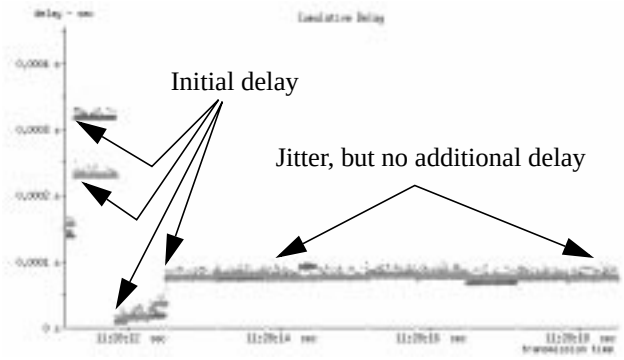

Fig. 4 Cumulative delay with LLQ

middle router controls QoS) in end-to-end QoS tests (from 128.m.n subnet to 204.x.y subnet). Cross Traffic Generators 1 and 2 (131.a.b subnet) sent timed bursts to Traffic Absorber 1 (204.x.y subnet).

Fig. 2 is the data path for analysis step 1, and Fig. 3 is cumulative packet delay chart for the test.

How to read a cumulative delay chart:

- Packet arrival time is X value (transmitting time) plus Y value (cumulative delay time).
- A straightly (smooth) ascending line means that every packet has constant delay from previous packet.
- A straight level line means no delay between packets.
- Uneven plot (level or not) means jittery.
- Declined area represents packet bunch (compacted packets).
- Empty area indicates packet loss
- An empty space followed by downward plot represents a long delay or loss followed by bunch.

Detailed packet loss is computed by the difference between the number of packets received and the number of packets transmitted, and is described in content of this paper.

In Fig. 3, real-time (RT) stream rate is 96.5 Mb/s and transmitted via a 100Mb/s connection (128.m.n subnet) to the main router, and video client is connected to switch via a 100Mb/s link. Aggregated cross traffic rate is 1.35Gb/s in average, and 2Gb/s in peak flow. Fig. 3 shows that all packets experienced delay, jitter and bunch, and many packets are lost due to congestion caused by extreme large amount of cross traffic. Packet loss rate varies between 26-64.5% (in 11 aggregated cross traffic bursts).

Fig. 4 is cumulative delay chart for analysis step 2a, a continuous real-time (RT) stream passed through a congested router where low latency queuing (LLQ) policy was deployed. The LLQ policy reserved 100 Mb/s bandwidth for RT stream with highest priority. The configuration used in this test is same as in Fig. 3 except LLQ is enabled at outgoing interface.

Two plots are in the Fig. 4. One is RT stream (darker) only, and the other one is RT stream plus the same type of cross traffic used in previous experiment. Fig. 4 shows that LLQ has fairly minimized packet delay and prevented packet loss. The maximum packet loss rate under this condition is between 0.1~0.3%, which is hard to see in the graph. From this graph, we can see that low latency queue

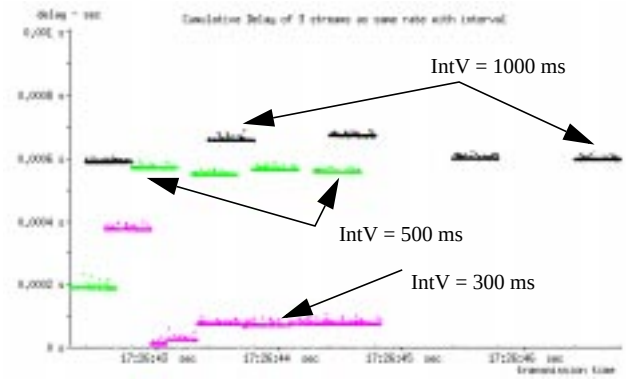

Fig. 5 Higher initial delay due to large burst interval (IntV)

Fig. 6 Cumulative delay with LLQ and bunched traffic

(LLQ) policy did not improve jitter because packet dispersion (the delay time) varies up and down all the time although the entire graph keeps level (no cumulative delay).

An interesting issue seen in Fig. 4 is that a large initial burst delay presents when a burst starts. To understand what caused this network behavior, further tests were performed. Three streams with exact same transmission rate, burst length and total bytes were sent to the same QoS network path with different burst intervals — 300ms, 500ms and 1000 ms. Fig. 5 shows that large burst interval produces higher burst initial delay. The first burst always has higher initial delay; rest burst initial delays in a stream depend on how large the burst interval is. This behavior suggests that applications can adjust their burst interval to control the burst initial delay.

Please notice that all plots for each measurement in Fig. 4, Fig. 5, and Fig. 9 have been nudged along the transmission time axis because all measurements were done on the same path, but in different time. Transmission time for each plot needs to be aligned (nudged, shifted) so all plots can be shown in the same time range for comparison.

Analysis step 2b used 550Mb/s RT stream (both video server and client are on GigE network interfaces) to replace 96.5 Mb/s stream in step 2a) had similar graph as step 2a, with slightly higher jitter (graph omitted due to no significant difference). The packet loss rate is in the similar range (reserved bandwidth for LLQ is 580Mb/s). Notice that reserved bandwidths in step 2 (a and b) are at least 3% more than the bandwidth required by applications. Otherwise, the quality of service, especially the loss, cannot be guaranteed. Preferably, 5~15% more bandwidth should be reserved for QoS. That spare bandwidth needs to be reserved for QoS depends on routers and bandwidth requirements.

Fig. 8 is the flow chart for analysis step 3. In this experiment, real-time stream met cross traffic (from XT generator 2) at Cisco Catalyst 5500 switch before going to the main router, where LLQ was configured.

Fig. 6 shows experience for analysis step 3, where QoS was partially deployed on our experimenting network. In this test, the video server was connected to a Cisco Catalyst 5500 switch and merged with another cross traffic before going to the main router. One cross traffic (from Traffic Generator 2) flowed through Catalyst 5500 switch, then passed through main router, and went to BlackDiamond router to Traffic Absorber 2. Average transmission rate of this traffic was about 600~700 Mb/s, which did not cause any packet loss when encountered with RT stream at the Catalyst 5500 switch. The burst rate of this cross traffic was still at line speed (GigE), and its purpose is to bunch RT packets at the Catalyst 5500 switch. Graph at lower right corner in Fig. 7 explains what is packet bunch. Similar initial delay effect presented in this experiment. First two bursts had higher cumulative delays due to packet bunch, and bunch impact reduced in rest bursts.

Fig. 3 has shown when traffic over saturates a router capacity, packets will experience loss. Experiment of analysis step 3 (in Fig. 6) shows even though the RT stream passes through a non-QoS router without experience loss, the cross traffic can cause packet bunch. Bunched packet bursts will cause packet loss on further non-QoS router, and increase packet delay and lost rate on

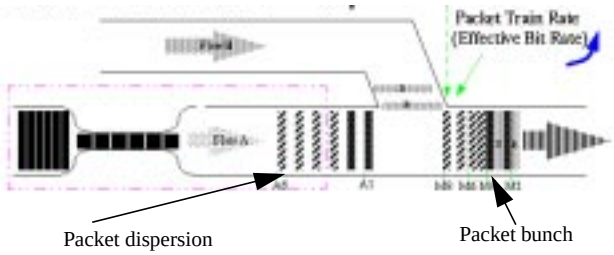

Fig. 7 Packet bunch caused by cross traffic

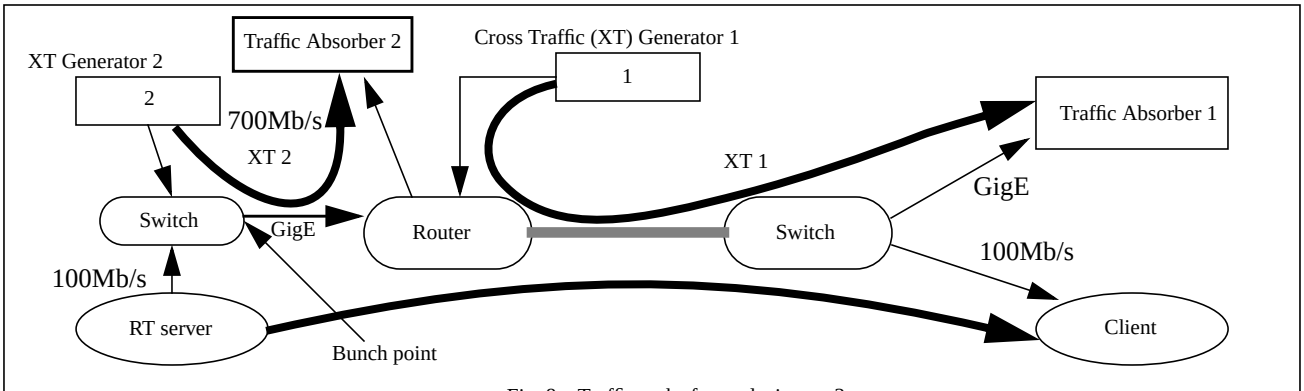

Fig. 8 Traffic paths for analysis step 3

QoS router. In this experiment, rest configuration is the same as in step 1 (Fig. 3) except video stream is not directly sent to the main router, but via Catalyst 5500, see Fig. 8. There is no packet loss reported by Catalyst 5500. However, the packet loss rate is increased from 0.1~0.3% to 5~10%. This is due to a number of RT-stream packets bunched at Catalyst 5500 GigE interface to form some short Gigabit flows, which are much higher than reserved bandwidth. Also, these short high-speed bursts encountered with ultra high-speed cross traffic to overwhelm the main router capability. Both conditions caused packet loss. In Fig. 6, we also see that the cumulative delay is also increased.

III. Packet Drop Avoidance

Deploying quality of service (QoS) with the same parameter and characteristic setting on entire Internet and local networks is difficult. Without full mesh QoS deployment, Fig. 3 (analysis step 1) shows that real-time stream packets can be delayed and lost on non-QoS network segments; and Fig. 6 (analysis step 3) shows even though real-time stream packets are not lost on non-QoS network segments, these packets can be bunched together to form a short high-speed burst, which can be dropped on further routers with or without QoS configuration. Therefore, to avoid packet loss is still a serious task for all types of network traffics in high-speed network architecture design.

Congestion avoidance [2] can avoid "further" packet loss after congestion occurs for reliable transmission protocols. This mechanism, however, will not avoid congest to happen, and cannot eliminate packet loss. Congestion avoidance is not suitable for real-time network applications due to its elastic feature. In other words, congestion avoidance delays packet transmission to avoid further congestion or packet loss, and the delay is not acceptable characteristic for trading packet loss in real-time applications. Real-time network applications need mechanisms to avoid both delay and packet loss.

Our experiments demonstrate that to avoid packet delay and loss, the basic concept is not to fill network router queues during packet transmission. A number of factors need to be considered in packet drop avoidance (PDA) based transmission protocol design.

- Do not fully (100%) utilize networks, especially on ultra high-speed networks. This will avoid filling up router queues.
- Develop and deploy mechanisms to avoid router queue overflow when queue is filling up
- Selectively drop obsolete real-time data as soon as possible, and preferably drop these packets at transmission hosts.

All these factors are directly or indirectly related to router queues, which are used for surge protection. Transmission protocol design that tries to use router queues as flow cushion is improper use of router queues. For example, sending a large packet burst at or above the bottleneck router capacity to cause the router queue partial burst (bunch) and letting the router pace this queued burst out is extremely poor transmission methodology as bunched packets can cause queue overflow and overwhelm further slow links. Paper [8] addressed basic mechanisms to avoid packet loss, and two main components are in PDA:

- Rate pacing is based on network available bandwidth — transmitting data at or below the available bandwidth to avoid router queuing
- Transmitting burst size is preferably smaller than one quarter of bottleneck router queue size — avoiding router queue overflow in case that network surge happens

Knowing available bandwidth is the mechanism to avoid filling up the queue at a bottleneck router, and transmitting bursts in proper size is the key to avoid queue overflow in case the bottleneck router queue is filling up due traffic

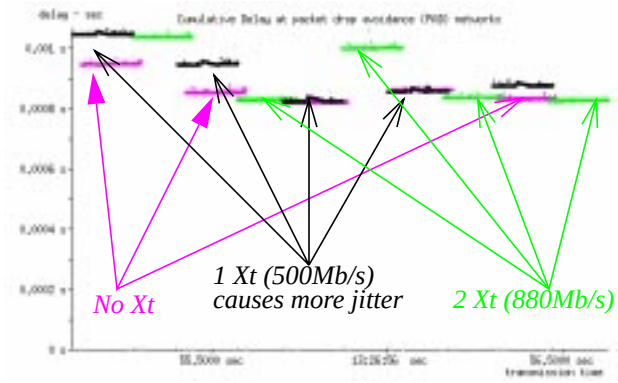

Fig. 9 Cumulative delay with packet drop avoidance

surge. In video-based application transmission, another key mechanism is desired: selectively dropping packets (SDP). Video-based applications need to have ability to selectively drop packets when available bandwidth is less than real-time application needs. Selectively tossing time sensitive data, which is obsolete now or will be obsolete when data reaches the destination, can avoid bandwidth waste, and reduce delay and loss for all applications that share the same network, thus, improving application's servicing quality.

Fig. 9 is cumulative delay chart on the same testbed used in Fig. 6. The difference is that there is no low latency queue (LLQ) set on main router to guarantee 100 Mb/s bandwidth for a 96.5Mb/s real-time (RT) stream. Instead, packet drop avoidance (PDA) is enabled on Traffic Generators 1 and 2. Under this configuration, the maximum aggregated cross traffic is confined within 880 Mb/s because bandwidth estimation embedded in PDA detected path bandwidth is 1Gb/s and about 100 Mb/s bandwidth is used by other traffic(s). In this experiment, packet loss rate is 0. From Fig. 9, we can see that packet delay plots were kept level, which means no cumulative delay occurred during this test, for both real-time (RT) bursts stand-alone and with cross traffics.

Readers may notice that a single 500Mb/s cross traffic stream produced additional jitter (cumulative delay followed by packet bunch) in middle of RT bursts. This is the burst initial delay caused by the burst interval, which has been described in Fig. 5. When two or more cross traffic streams flow through the same path, burst intervals are overlapped by other bursts, so the effect of burst initial delay is reduced.

The result of packet drop avoidance (PDA) deployed on real-time application networks shows that PDA can effectively reduce packet delay and loss for all network traffics when PDA is deployed on all transmission hosts. For reliable data transmission, PDA can adjust packet transmission pace (delay) to avoid packet loss. For real-time data transmission, PDA can selectively discard obsolete packets to ensure rest data can be delivered in time and to avoid wasting available network bandwidth.

IV. Which Layer to Deploy Packet Drop Avoidance Mechanism

Experiments in § II. and § III. show that packet drop avoidance (PDA) is a necessary step to improve quality of network services in high-speed networks because PDA can reduce packet bunch and avoid packet loss on high-speed networks. In this section, we discuss where is suitable place to realize and deploy PDA.

Intuitively, packet drop avoidance (PDA) should be realized at network layer (Layer 3) because network layer can control pacing for all upper layer protocols and simplify upper layer's design. Different transport layer (Layer 4) protocols can be flexibly added on top of Layer 3 without implementing their own transmission pacing control. All upper layer protocols can obtain available bandwidth information from the network layer for their transmission decision. This benefits real-time (RT) protocols, which can selectively discard data (packets) that will be obsolete as reaching destination hosts due to insufficient bandwidth. Other advantages of design PDA in network layer are: 1) group traffics to the same destination or to the same bottleneck together, and prioritize time sensitive packets. 2) move system on chip (SoC) and make zero-copy network I/O easy.

Since network layer knows where traffic is routed to, network layer can group all traffics going to the same route together and promote time sensitive packets in front of queue. Where current network layer design, packets are served as first come first out (FIFO). Thus, implementing PDA at network layer will smoothly pace out all traffics to the same destination or pass through the same bottleneck, and send out high priority packets without waiting in the outgoing queue in network layer.

Moving network system onto network interface card (NIC) becomes necessary because NIC speed has closed to computer memory sub-system (not memory chip) bandwidth and exceeded I/O sub-system bandwidth. This means that to drive NIC in full speed with continuous network data stream, a computer system needs to devote all its bus and CPU bandwidth for network process. Then, no resource is left for computation.

To resolve this issue, network process needs to be moved onto NIC to balance loads on different sub-systems. Some companies designed both Layer 4 (transport) and Layer 3 (network) onto to their NICs to try to save system resource, but users and applications have not accepted this idea. The problem is that many transport protocols exist and putting one or some Layer 4 protocols

onto the chip and leaving others in main system make socket level design difficult. Also, moving a reliable transport protocol, like TCP, onto a chip without considering future ultra high-speed network demanding is not practicable. A reliable transmission protocol needs to keep all data in transmission buffer till an acknowledgment comes back to confirm a packet has been received, then this packet can be removed from the transmission buffer. To maintain a large transmission on a chip is impossible for ultra high-speed network. For example, 1Tb/s network path with 200 ms delay requires transmission hosts to keep 25 Gigabytes data before acknowledgment coming back. The 25 GB static RAM (dynamic RAM is not fast enough) will require about 100 CM^2 silicon die estimated on future chip technology. This die size is too large to manufacture on a chip, as well as the light (electronics) can only travel through this big silicon in one clock cycle if network interface card (NIC) speed is below 2Gb/s. Therefore, putting Layer 4 (transport layer) onto a chip is not a suitable design for building network system on chip.

The other advantage to move network layer onto a chip is easy to achieve zero-copy network I/O. To realize zero-copy network I/O, the system memory page size (page size) must be the same size as NIC I/O buffer, which is determined by maximum transfer unit (MTU). Most commonly used MTU is 1500 bytes, where typical system page size is 4 kilobytes or 8 kilobytes. So, make zero-copy network I/O under this condition is very difficult or impossible. When network layer is moved onto a chip, then MTU size is no longer affecting the I/O buffer size. The on-chip network layer will re-map I/O buffer to receive/transmit different MTU. Because zero-copy network I/O can greatly reduce system memory bandwidth and CPU usage, computer systems will then have more power for computation.

V. Conclusion

Packet loss affects not only quality of service for applications that causes packet loss, but also wastes network bandwidth, thus affecting other application's performance and quality on entire networks. Therefore, avoiding packet loss is fatal and imperative in ultra high-speed network infrastructure operation and management. This paper studied and analyzed one of the best bandwidth guarantee mechanism — low latency queueing (LLQ) — on paths where end-to-end QoS is enabled, and on a path where QoS is partially enabled. On the dedicated QoS path, LLQ can highly minimize the delay and packet loss. However, this method does not scale when real-time traffic uses higher percentage of network bandwidth because "what is the proper percentage of the network bandwidth to reserve?" Also, deploying QoS on entire network is a difficult task. Dynamically setting up QoS, such as using MPLS, is under development, and requires all router to support the same type of QoS mechanism, so MPLS can cross all ISP (internet service provider) boundaries to configure the entire path.

This paper addressed what is packet drop avoidance (PDA) and how important PDA is in improving network performance. This paper has also discussed how to design PDA in network layer that makes moving network system on chip easier, in order to improve network I/O performance as well system performance and computation power. Thus, adding PDA in network layer and moving network layer on chip is imperative step for ultra high-speed network improvement.

VI. Acknowledgments

This work was supported by the Director, Office of Science. Office of Advanced Scientific Computing Research. Mathematical, Information, and Computational Sciences Division under U.S. Department of Energy Contract No. DE-AC03-76SF00098. This is report no. LBNL-53921. See disclaimer at

http://www-library.lbl.gov/disclaimer.

References

[1] BRADEN, R., Clark D. and Shenker S. RFC 1633: Integrated Services in the Internet Architecture: an Overview. IETF, June 1994.

[2] Jacobson, V. Congestion avoidance and control. In Proceedings of SIGCOMM '88, Stanford, CA, Aug. 1988

[3] BLAKE, S. et al. An Architecture for Differentiated Services. IETF, RFC 2475, December 1998

[4] CROLL, A. and PACKMAN, E. Managing Bandwidth: Deploying QoS in Enterprise Networks. Prentice Hall, 1999.

[5] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol Label Switching Architecture, RFC 3031, January 2001

[6] http://www.mplsrc.com/mplsfaq.shtml

[7] D. Agarwal, J. M. González, G. Jin, B. Tierney An Infrastructure for Passive Network Monitoring of Application Data Streams, 2003 Passive and Active Measurement Workshop, San Diego, CA, April 2003, LBNL-51846

[8] G. Jin, Packet Drop Avoidance for High-speed Network Transmission Protocol, In Proceeding of 2004, International Conference on Information Technology: Research and Education, June 28 - July 1, 2004, London, UK, LBNL-53920

[9] Datagram Congestion Control Protocol (DCCP), available at http://www.icir.org/kohler/dcp/