A Precedence-Enabled Per Hop Behavior: Impact on TCP and UDP Flows

Robert G. Cole
Johns Hopkins University
Applied Physics Laboratory and Department of Computer Science
Laurel, MD, USA 20723–6099
Email: robert.cole@jhuapl.edu

Abstract

Currently, preferential packet handling in transport networks is based solely upon application-level Quality of Service (QoS) requirements. No preferential packet handling is based upon the importance of the information being carried by the network. In future transport networks, packet handling should provide preferential transport to important, high precedence traffic, specifically during conditions of resource scarcity, e.g., network overload conditions, while simultaneously satisfying packet scheduling required to meet application QoS needs. We propose and analyze an approach to supporting both Precedence and Preemption (P&P) and QoS handling in common transport infrastructures. Precedence has to do with the relative importance of the information content while preemption has to do with mechanisms to deny lower precedence traffic access to network resources in favor of higher precedence traffic, when necessary. Our approach to this duality is to enhance Active Queue Management (AQM) techniques to provide P&P capabilities and rely upon standard, well studied QoS schedulers, e.g., Weighted Round Robin, Class-Based Fair Queuing, etc., for handling QoS requirements. We refer to this combination as a Precedence-Enabled Per Hop Behavior (PHB). In this way, when operating under engineered loads, the well known scheduling algorithms support high quality QoS for applications. Under network congestion situations, the enhanced AQM layer provides the necessary P&P preferential packet handling favoring high Precedence-Level (P-L) information. Our scheme allows low order queues (within the context of QoS handling) to plead up to the next higher order queue for help in alleviating queue congestion under periods of communication link overload. We refer to our scheme as the Cross Queue-AQM (CQ-ACM) Scheme. Our scheme can be extended to higher numbers of queues and any type of scheduler in a straightforward manner.

Through extensive simulation studies, we investigate the performance of our CQ-AQM scheme under heavy traffic limits, where preemption is required. The performance metrics of interest to our analysis are packet delay, packet loss and throughput as a function of the packet QoS class and P-L. We also define here two new metrics of merit, which are specific to P&P considerations. These are the *Gain*, which measures the benefit of the P-L handling to the high P-L traffic, and the *System Efficiency*, which measures the lost total system capacity due to implementation of P-L handling.

Our studies concentrated on both flow controlled and non-flow controlled traffic, as well as mixed traffic conditions. Our simulation results show that our algorithms performed extremely well. We find that the application of our CQ-AQM scheme on top of standard QoS scheduling is effective in simultaneously supporting QoS and P&P transport and assures the delivery of high precedence traffic under all ranges of traffic loads and traffic types studied. So our local CQ-AQM scheme, which is extremely simple to implement and extends to all types of existing QoS schedulers, is an extremely useful tool in the overall architecture for packet-based, precedence-enabled, future transport services.

I. Introduction

The world's economy is becoming more and more reliant upon the development of a reliable, resilient communications capability. In particular, future networks are expected to deliver services during emergency situations resulting in periods of extreme network stress. During periods of emergency, the communications infrastructure must be capable of providing preferential delivery of information based upon the end user's indication of the importance of the information, as indicated by the message Precedence-Level (P-L). Network capabilities should exist to allow for preemption of less important traffic during these periods. Else, packet transport networks will likely suffer general collapse of capabilities during network stress, resulting in no useful data delivery.

Most work to improve the reliability of packet-transport services, e.g., the Internet, concentrate on a) capacity planning methods to attempt to maintain adequate network capacity during all times, and b) redundant network design to improve fault tolerance. Little to no work focuses upon the development of a Precedence and Preemption (P&P) capability and supporting protocols to ensure graceful degradation of network goodput during periods of network overload. Precedence has to do with the relative importance of the information content while preemption has to do with mechanisms to deny lower precedence traffic access to network resources in favor of higher precedence traffic, when necessary.

In order to support P&P, the all-Internet Protocol (IP) packet-based transport network must develop new packet handling and forwarding algorithms to simultaneously support application QoS and content importance P&P requirements. Work exists in the literature on the design of forwarding algorithms, commonly referred to as Per Hop Behaviors (PHB), to meet the QoS requirements of applications, e.g., [3] [16]. However, to date, little work exists to design PHB algorithms which simultaneously deliver QoS to applications and P&P transport to information.

Experience in providing P&P capabilities in communications services falls primarily into two camps, i.e., traditional telephony services, e.g., the ITU-T's Multi-Level Precedence and Preemption (MLPP) Service [14], and message handling services, e.g., the US Department of Defense's (DOD's) Automated Message Handling System (AMHS) and others. A straightforward mapping of these onto an all-IP, packet-based transport network like the Internet is problematic. The MLPP Service handles precedence through signaling to indicate the P-L and notification and resource reservation for assured delivery. However, this approach fails for non-session oriented applications and does not support in-band signaling architectures like those discussed for emergency and military network deployments. Message handling systems maintain the entire message as an atomic unit (i.e., they do not segment the messages and multiplex with data from other messages) and provide preferential queuing and scheduling to high P-L messages. However, packet queuing and scheduling in IP networks is designed to maintain QoS for applications, where message segmentation and multiplexing are required for efficient networking and timely packet delivery. Current PHBs defined within the Internet Engineering Task Force (IETF), see, e.g., [1] [9], address QoS forwarding but not P-L handling.

In a general service IP transport network, the packet handling must provide preferential transport to high P-L traffic under all networking conditions, specifically conditions of resource scarcity, e.g., network overload conditions, while simultaneously satisfying packet scheduling required to meet application QoS needs. Our approach to this duality is to enhance Active Queue Management (AQM) techniques to provide P&P capabilities and rely upon standard, well studied QoS schedulers, e.g., Weighted Round Robin, Class-Based Fair Queuing, etc., for handling QoS requirements. In this way, when operating under engineered loads the well known scheduling algorithms support high quality QoS for all applications. Under network congestion situations, the enhanced AQM layer provides the necessary P&P preferential packet handling and assured delivery to high P-L information. Also, our approach here concentrates on local processing only, which we feel is a necessary starting condition for a robust QoS and P&P handling in general and diverse IP-network deployment. As such it is highly relevant to wireless Mobile Ad-Hoc Networks (MANETs) where non-local, reservation-based schemes for P&P are bound to fail due to the network dynamics. As well, it naturally handles non-flow based applications in both broadband wired and low bandwidth wireless networks.

The tricky part to developing an enhanced AQM scheme for P&P handling is to prevent the possibility of *Precedence Inversion* while simultaneously achieving a high system efficiency. Precedence Inversion occurs when low precedence, delay and jitter sensitive application traffic overloads the communications resource causing high precedence, non-delay sensitive traffic to be discarded. To avoid this situation, the enhanced AQM capability must act across the entire interface buffer and not solely within individual queues partitioning the buffer due to QoS schedulers. In this paper, we define a new P&P metric, the *System Efficiency*, which is defined as a measure of the system's ability to limit losses incurred at the benefit of higher precedence traffic. We define and report our efficiency measure in our analysis, as well as another new P&P metric called *Gain* described below.

In [5], we proposed a simple and relatively straightforward scheme for coordinating packet queue admissions across all queues comprising the communications interface buffer. Our scheme allows low order queues (within the context of QoS scheduling) to plead up to the next higher order queue for help in alleviating queue congestion under periods of communication link overload. We refer to our enhanced PHB as the Cross Queue - Active Queue Management (CQ-AQM) scheme. We provided an initial modeling and simulation study of our CQ-AQM scheme under a range of traffic loads, traffic models and schedulers, see [5] [6]. In this paper, we present our full set of simulation results which more fully analyze the impact of our CQ-AQM scheme in the presence of various traffic models including both flow controlled, i.e., TCP sources, and non-flow controlled, i.e., UDP sources, and mixed flow controlled types.

The performance metrics of interest to our analysis are packet delay, packet loss and packet throughput as a function of the packet QoS class and P&P level. We also introduce two new metrics specific to P&P studies. These are referred to as the system *Gain* and the system *System Efficiency*. The gain measures the benefit of preemption to the flow in question. The system efficiency measures the system's overall ability to support precedence handling without incurring excessive losses at the expense of lower precedence levels. We report these metrics for a strict priority queuing QoS scheduling algorithm. This scheduler represents a standard scheduler and is a reasonable starting point for investigations of impact on flow-controlled traffic sources. We concentrate on a somewhat simplified buffering and packet handling system, i.e. a two queue and two precedence-level packet handling system. It is conceptually easy to see how to extend our packet handling schemes to more complex, more realistic scenarios. Our studies indicate that the application of our CQ-AQM scheme on top of standard QoS scheduling is effective in simultaneously supporting QoS and P&P transport for UDP, TCP and mixed traffic sources.

The remainder of our paper is organized as follows. In Section II we describe in detail our proposed CQ-AQM system for joint QoS and P&P handling and provide context for our interest in Precedence-Enabled PHBs. In Section III we describe our traffic models and simulation tools for the analysis presented in this paper. In Section IV we present our results for three different traffic mixes, i.e., UDP-only, TCP-only and mixed UDP and TCP traffic. In Section V we discuss previous, related work in the field of P&P services in communications networks. In Section VI we end with our conclusions and projections for future work.

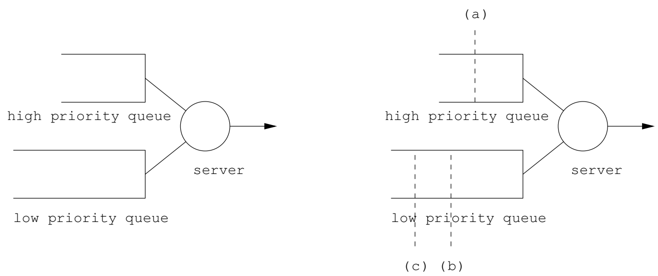

Fig. 1. The baseline scheduler and cross-queue active queue management scheme.

II. Approach and Context

Most work addressing precedence handling in packet networks attempt to strictly emulate precedence handling in circuit-based voice networks, like the MLPP Service. This body of work relies solely upon signaling and reservation protocols to manage the scarce networking resources during overload situations. However, these service approaches are applicable to constant bit-rate, flow-oriented applications like Voice over IP (VoIP). A large proportion of the evolving data applications in the commercial, financial, emergency and military networks are not flow based. Even for flow-based, constant bit-rate applications, network architectures discussed within the military and emergency services context rely upon common data transport integrated with signaling, i.e., in-band signaling. For end-to-end precedence handling the P&P architectures must support precedence handling of the signaling messages as well as the application data. Also, other infrastructure services, e.g., Domain Name Service (DNS), Mobile IP, etc., require appropriate precedence handling. These cannot be supported through reservation mechanisms. Finally, it is well understood that a critical component of emergency and military networks will rely upon wireless mobile ad-hoc networks (MANETs). In these environments, due to their high mobility and dynamic communication channels, signaling and reservation protocols are not viable. Hence, we come to the conclusion that the underlying packet transport network must be precedence aware and that local packet handling methods are required to support the new P&P services. This does not say that signaling and reservation protocols are not important in the architecture. Instead, we believe that architectures cannot solely rely upon reservation techniques. These arguments are more fully discussed in [4] [8] and [7].

Hence, we have embarked upon a program to investigate the ability of local packet handling methods to provide aspects of P&P. Local methods for precedence handling include new definitions of Per Hop Behaviors (PHBs), which would simultaneously support application QoS and content importance P&P. Local methods also include mechanisms for precedence handling feedback to applications so that they react appropriately in overload situations. Other local methods are required for an end-to-end P&P transport architecture [7]. In this work we focus solely on the development of new PHBs.

We referred to our scheme in [5] as the Cross Queue-AQM (CQ-ACM) Scheme. This is illustrated in Figure 1 in the context of a two queue priority scheduler. The thresholds indicated by (a) and (b) are thresholds which trigger discarding of lower P-L traffic within the specific queues where they are triggered. The threshold indicated by (c) is a cross queue threshold which causes the higher priority queue to discard low P-L traffic regardless of the current state of the high priority queue. This trigger, or threshold, allows the system to triage the packet handling somewhat independent of the QoS, or scheduling, of the packets. Instead, the overall system attempts to handle the packets according to their precedence indications when having to make decisions with respect to packet discarding in local overload situations. Our scheme can be extended to higher numbers of queues and alternative QoS schedulers in a straightforward manner; it is not limited to the simple two queue system we use here for Modeling and Simulation (M&S) purposes.

III. Models and Simulation

We developed a simulation model in order to assess the performance of our CQ-AQM scheme for handling precedence tagged traffic. Because this represents an initial study of new PHBs, we were interested in assessing performance under a broad range of traffic models, queuing arrangements, scheduling algorithms and drop policies. We felt the best way to accomplish this is to begin with a small simulation model of a single PHB. In this section we describe our methodology, simulation model, arrival and service processes, and scheduling and drop policies investigated.

A. Methodology

For simplicity of discussion and presentation, we assume only two levels of QoS, i.e., $q = 1$ indicating high priority class, and $q = 0$ indicating low priority treatment. The mechanism to extend our method to more QoS levels is relatively straightforward. This requires each lower QoS-level queue the ability to indicate/request preemption services from the higher-level queues. “Priority treatment” can mean strict priority, or preferential scheduling based upon a Weighted Round Robin or Deficit Round Robin scheme. Note that priority treatment is fundamentally different than precedence treatment. We assume only two levels of precedence, i.e., $p = 1$ indicating high importance, and $p = 0$ indicating low importance. Associated with each QoS class is a queue, which is serviced according to the specific scheduler under consideration. We then compare the performance of the various traffic types, i.e., $(p,q) = (0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$, for the case of no AQM versus the case of having the CQ-AQM scheme. We also track the system delay, loss, throughput, gain and system efficiency for the various traffic types. For simplicity, we compute our metrics of interest as seen by the arriving packets, hence we present packet averages for these metrics.

For our initial studies, we investigate the following active queue management scheme implemented on top of the strict priority scheduler. Both queues maintain a counter reflecting the number of packets within their queues. Each queue is configured with a threshold, i.e., T_{high} and T_{low} respectively, where $T_{high} \leq B_{high}$ and $T_{low} \leq B_{low}$. Here, B_{high} is the size (in terms of packets) of the high priority queue and B_{low} is the size (in terms of packets) of the low priority queue. Class $q = 0$ feeds the low priority queue and class $q = 1$ feeds the high priority queue. Further we model only two P-Ls, where $p = 1$ gets preferential treatment over $p = 0$. In the event that the buffer occupancy of the low priority queue equals or exceeds T_{low}, then the active queue management denies access to all packets with $q = 0$ and $p = 0$. In the event that the buffer occupancy of the high priority queue equals or exceeds T_{high}, then the active queue management denies access to all packets with $q = 1$ and $p = 0$. In the event that the buffer occupancies drop below their local thresholds, then their respective active queue management allows all P-L traffic access to the queues [1].

These per-queue thresholds help address the problem of preemption within each queue, but there are cases where low QoS class data is tagged high P-L and high QoS class voice is tagged low P-L and we need to be able to communicate a low priority threshold event to the high priority buffer management in order to prevent P-L inversion. To prevent this situation from happening, we implement one additional threshold in the low priority queue, i.e., $T_{low}^{(cross-queue)}$, where $T_{low} \leq T_{low}^{(cross-queue)} \leq B_{low}$. When the buffer occupancy of the low priority queue equals or exceeds $T_{low}^{(cross-queue)}$, then the high priority queue management scheme causes all low P-L traffic to the high priority queue to be dropped. The activity remains in effect until the buffer occupancy in the low priority queue drops below T_{low}. When we set $T_{low} = T_{low}^{(cross-queue)} = B_{low}$ and $T_{high} = B_{high}$, then we effectively disable the active queue management and recover the standard two finite queue priority model. The strict priority scheduler always checks the high priority queue and services all packet in queue prior to servicing a packet in the low priority queue.

For this work, we wrote a small custom simulation program in C++. The structure is that of a simple discrete event simulator with event heap and objects implementing the distributions that drive a given simulation run. We have the capability of instantiating as many arrival objects as necessary to achieve a given utilization level at the queue. The fact that the simulator is a custom program allows us to implement non-standard queue management mechanisms and to have exact control over what information is collected in the course of the simulation. It also allowed us to incorporate objects implementing the empirical distributions very easily.

To simplify Verification and Validation (V&V) of the simulation model, we built the scheduler upon the Heap Structure provided by C++'s Standard Template Library. Further, as the simulation was developed, we began by first building simple queueing models, e.g., M/M/1, M/G/1, M/M/1/K, with known analytic solutions to compare the simulation results against for simulation validation. As we built the various arrival processes, these were tested against known results for specific process configurations, e.g., multiple Poisson Arrival processes were compared against results of a single Poisson Arrival process with equivalent arrival loads. Finally, the simulation code was independently reviewed in order to provide verification of the final code version.

Three different sets of arrival processes were used to drive the simulation experiments: one process type simulates Constant Bit Rate (CBR) Voice over IP (VoIP) UDP-based traffic, one process type simulates non-flow controlled data UDP-based traffic and one process type simulates flow controlled, TCP-based data. The CBR model was designed to emulate a G.729a codec running over RTP/UDP with silence suppression enabled[2]. As such, it uses a constant packet size of 68 bytes and is an on-off process. The on-times are exponentially distributed with a mean of 200 milliseconds, and the off-times are exponentially distributed with a mean of 133 milliseconds; approximating talk-spurts and silence periods. The packet generation rate when the model is in an on-state is one packet every 20 ms, which is a typical packetization interval for codecs/gateways used for digitized voice. This essentially generates a UDP-based traffic stream of 2.0 Kilo-Bytes per second (KBps).

The UDP-based data stream is modeled by an exponentially distributed packet size with a mean of 100 bytes, and an arrival process that alternates between two states, according to a Markov Chain. The states are a higher-intensity state, where the average inter-arrival time is 460μseconds and a lower-intensity state where the average inter-arrival time is 46 milliseconds. In both states the inter-arrival times are also exponentially distributed. In the high-intensity state, the probability of transitioning to the low-intensity state is 0.011 and the probability of going from the low-intensity state to the high-intensity one is 0.091. This essentially generates a UDP-based traffic stream of 20 Kilo-Bytes per second (KBps) and a coefficient of variation, defined as the ratio of the standard deviation of the inter-arrival times divided by its mean, of 2.0. This represents a very bursty traffic

[1] Clearly it is desirable to implement different upper and lower thresholds for this queue management scheme to prevent thrashing. However, for our initial studies and to simplify the initial analysis we implement this single threshold strategy. Later on we will implement the two threshold scheme to eliminate thrashing.

[2] Certainly various VoIP codecs and protocol stacks will be deployed within the military and emergency services networks. Our choice of a G.729a is merely for illustrative purposes. Future studies will investigate other codec types

source.

We also modeled a TCP traffic source implementing a selective acknowledgment scheme as analyzed in [2]. We chose this specific variant of TCP because of its superior performance in wireless networks. We are extremely interested in developing a viable P&P architecture for wireless, MANETs for emergency and military applications, where local processing intelligence is necessary due to the network dynamics. However, we expect our results and conclusions to be generally applicable to other TCP variants. Our TCP traffic process implements TCP congestion management, i.e., Slow Start, selective acknowledgment, and is configured with a maximum window size, which we set to 4096 bytes and a maximum segment size of 512 bytes, for this study. Multiple instances of the TCP source are configurable in our simulation tool, as well as for the other traffic sources. The load offered by each TCP source is determined by the maximum window size, the network buffer size, network loss and delay statistics, P-L handling and background load.

IV. Results

In this section, we present our modeling and simulation results. We used our flexible simulation modeling facility to develop three traffic scenarios. The first traffic scenario consists of only UDP traffic, the next consists of only TCP flow-controlled data traffic. We then develop a mixed traffic scenario, basically adding TCP flow-controlled traffic onto our mixed VoIP and bursty, non-flow controlled UDP-based data traffic.

For each of these traffic scenarios, we concentrate on the ability of our CQ-AQM scheme to support the UDP and TCP-flows under overload conditions. The metrics for the UDP and TCP traffic which we investigate include throughput, delay, loss, gain and system efficiency. These are defined as follows:

- *Throughput* – for each TCP flow, the throughput is defined as the number of packets acknowledged divided by the total simulation time. All TCP flows continuously transmit over the duration of the simulation runs.
- *Delay* – for each traffic class, the packet delay is defined as the time the packet completed service within the packet scheduler, including any necessary queuing delays, minus the time the packet was received by the scheduler. The simulation model includes a propagation delay modeling down stream networking delays, but this time is not included into the definition of the delay. The traffic class is defined by the P-L and QoS type indications.
- *Loss* – for each traffic class, the packet loss is defined as the number of packets discarded by the queue management divided by the total number of packets offered to the queue management.
- *Gain* – for TCP flows, we define the gain as the ratio of the realized throughput to the throughput achieved in the case of no preemption. The gain should be greater than unity for applications with higher P-L and less than unity for applications with lower P-L.
- *Efficiency* – for TCP flows, we define the system efficiency as the average throughput of all P-L flows relative to the throughput achieved in the absence of preemption. An ideal preemption algorithm should achieve an efficiency of unity.

In the remainder of this section, we separately present simulation results for three cases, i.e., UDP-only flows, TCP-only flows and mixed UDP/TCP flows.

A. UDP-Only Results

We first investigate performance of our PHB model under smooth traffic conditions, where we tune the data sources to be Poisson, for the strict priority scheduler. We define a voice and data stream pair according to the traffic models discussed earlier. Our lowest offered load results are for a single pair of independent voice and data streams. For this single pair, we set their packets to carry $p = 1$. Then we increase the offered load by adding additional, identical arrival process pairs, except that the packets for the additional arrival processes are labeled with $p = 0$. For example, one set of runs consists of 6 arrival process pairs. There are six G.729a like voice streams and six 20 KBps data streams. One each of the voice and data streams are labeled $p = 1$ while the rest are labeled $p = 0$. The voice streams carry a QoS class marking of unity and the data streams carry a QoS class marking of zero. The overall offered load for this case is roughly 68.8% utilization of the DS-1 rate (i.e., 1.544 Mega bits per second (Mbps)) server.

We sized the buffers by running simulations until we found a buffer size that yielded a loss rate of about 1/1000 in the low priority queue. This resulted in a buffer size of 12 packets for low priority. We then (rather arbitrarily) set the size of the high priority buffer to 4. We ran a series of simulation studies with these buffer sizes with and without the active queue management scheme enabled for increasing levels of offered load. Each simulation ran for 100 seconds of simulation time and handled over 10^5 packet departures. Data in the plots represent the average of ten independent simulation runs.

In Figure 2, we show plots of the mean system delay and loss per QoS Class, i.e., q = 0 or 1, and per precedence-level, p = 0 or 1. The Left Hand Side (LHS) plots show the results when no active queue management is implemented. Here there is no discrimination between p = 0 or 1 traffic. We ran several scenarios ranging from an offered load of 10% to a high of roughly 115%. Because the data traffic is relatively smooth, the buffer sizes are small and the resulting delays are small as well. The high priority delays are significantly smaller than the low priority delays. The high precedence traffic is generated by one CBR flow and one data flow. The Right Hand Side (RHS) plots show the corresponding system delay and loss when the active queue management is configured. Here we set $T_{high} = 3$ packets, $T_{low} = 8$ packets, and $T_{low}^{(cross-queue)} = 10$ packets,

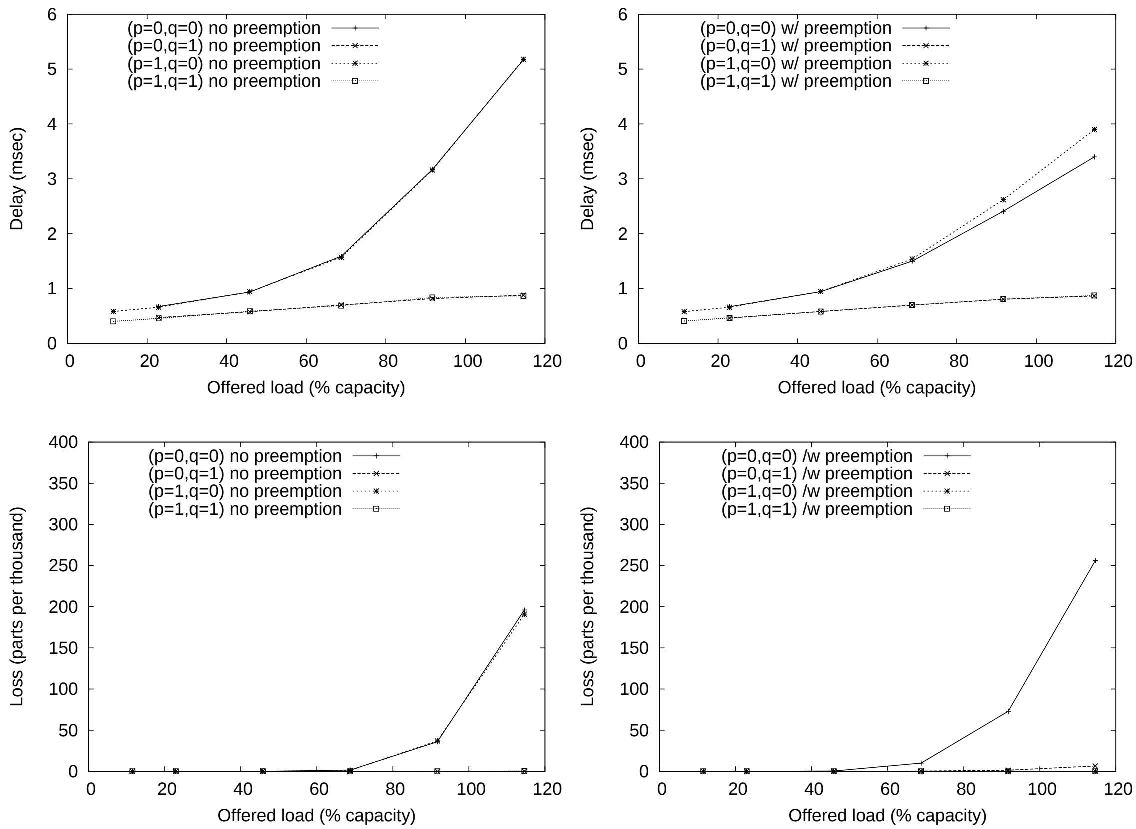

Fig. 2. Example delay (upper) and loss (lower) behavior without (left) and with preemption (right) for the Poisson data model.

We see little change in the delays for the voice streams. For the data streams we see a slight decrease in the delays. As well, we see that the system is now discriminating between the different P-L traffic.

Figure 2 shows the loss results on the bottom set of plots for the same set of simulation runs. As for the delay plots, the LHS plot show the results with no active queue management. Here we see packet losses in the low priority queue equally for both the P-L 0 and 1 traffic. Clearly, this is not a desirable result. The RHS plot show the results when the CQ-AQM scheme is configured. Here we see a slight increase in the loss probability for the voice and data streams tagged P-L equal 0, while the loss probability for the voice and data streams tagged P-L equal 1 is zero. This is the desired effect.

We now increase the burstiness in the traffic profiles. Here we set the Coefficient of Variation (Cv) for the data streams to 2.0, and repeat the process we followed above. We again made a set of simulation runs to size the buffers. This resulted in the low priority buffer size being set to 750 packets. We set $T_{high} = 3$ packets, $T_{low} = 500$ packets, and $T_{low}^{(cross-queue)} = 625$ packets. This large increase in the buffer size over the previous case is due to the increase in the burstiness of the arrival process.

Figure 3 shows the results for the system delay and loss in the simulation runs. The LHS plots show the case where no active queue management exists. The RHS plots show the comparable results with CQ-AQM configured. Of course, the actual delays have increased significantly over the previous simulation runs. However, the results are qualitatively similar. Turning on CQ-AQM slightly reduces the delays seen by the data packets. When active queue management is configured the loss probability for all P-L tagged traffic is zero. The P-L zero tagged traffic shows a slight increase in their losses for both QoS class zero and one.

We also ran a set of studies where we modeled a Deficit Round Robin scheduler as described in [13]. A summary of all of our results for UDP-only traffic, is found in Table IV-A. Here we present only the loss metrics for the various configurations for the case where the offered load was the maximum studied, i.e., an offered load of approximately 115%. For the case of the Deficit Round Robin scheduler, we ran two different sets of quantum sizes, i.e., $Quantum_{High} = 70$ and $Quantum_{Low} = 7$ or 70. We left the arrival processes and the buffer sizes and thresholds to be the same as for the previous strict priority scheduler and bursty traffic arrival processes. For each configuration listed in the table, the first row shows results for the case of no AQM and the second row shows the results for the CQ-AQM PHB. The results for the packet losses for the Deficit Round Robin cases are very similar to our previous simulation results; the CQ-AQM scheme is very effective in avoiding precedence inversion and protects the high precedence traffic regardless of QoS level.

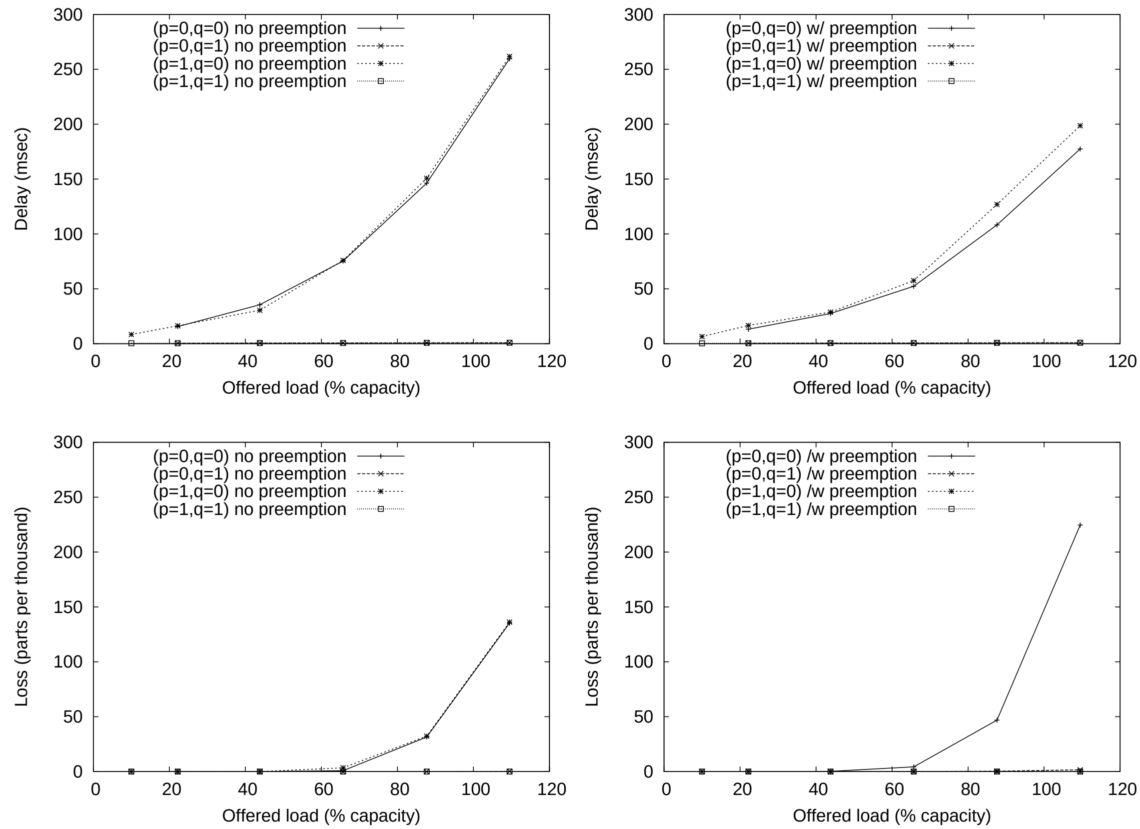

Fig. 3. Packet delay (upper) and loss (lower) behavior without (left) and with CQ-AQM (right) for the data model with Cv = 2.

Traffic Model	Scheduler	Losses (ppt)			
		(1,1)	(1,0)	(0,1)	(0,0)
CBR+Poisson	Strict	0.4	191.0	0.1	196.0
		0	0	0	256.0
CBR+CV=2	Strict	0	136.2	0.2	135.3
		0	0	1.6	224.6
	Deficit	0	96.9	0.2	101.3
	(70&7)	0	0	1.4	144.6
	Deficit	0	106.4	0.1	106.6
	(70&70)	0	0	1.3	215.0

TABLE I
SUMMARY OF LOSS RESULTS FOR THE CASES SIMULATED.

B. TCP-Only Results

We varied the number of TCP flows from a low of 5 to a high of 10. All the flows carried a QOS level of $q = 0$, and so were mapped by the scheduler into the low priority queue. A single TCP flow was tagged $p = 1$, while all other TCP flows were labeled $p = 0$. For the preemption cases, the low priority queue size was set to 50 packets. This buffer size was somewhat arbitrarily chosen in order to keep the packet loss rate low under most of the traffic load scenarios studied in this section. The low threshold T_{low} is set to 35 or 40 packets; two cases were investigated. The high threshold setting was irrelevant to this set of simulation runs. The cross-queue low threshold $T_{low}^{(cross-queue)}$ was set to 45 packets. For the non-preemption cases, all thresholds were set to the total buffer size, effectively reducing the system to a Tail Drop system.

These thresholds were somewhat arbitrarily chosen to illustrate the impact of service disciplines on packet level metrics. Ideally we should be able to derive optimal threshold settings based upon derived expressions/methods. We would then evaluate the performance of our system near these optimal settings. However, to date we do not have analytic expressions for the optimal

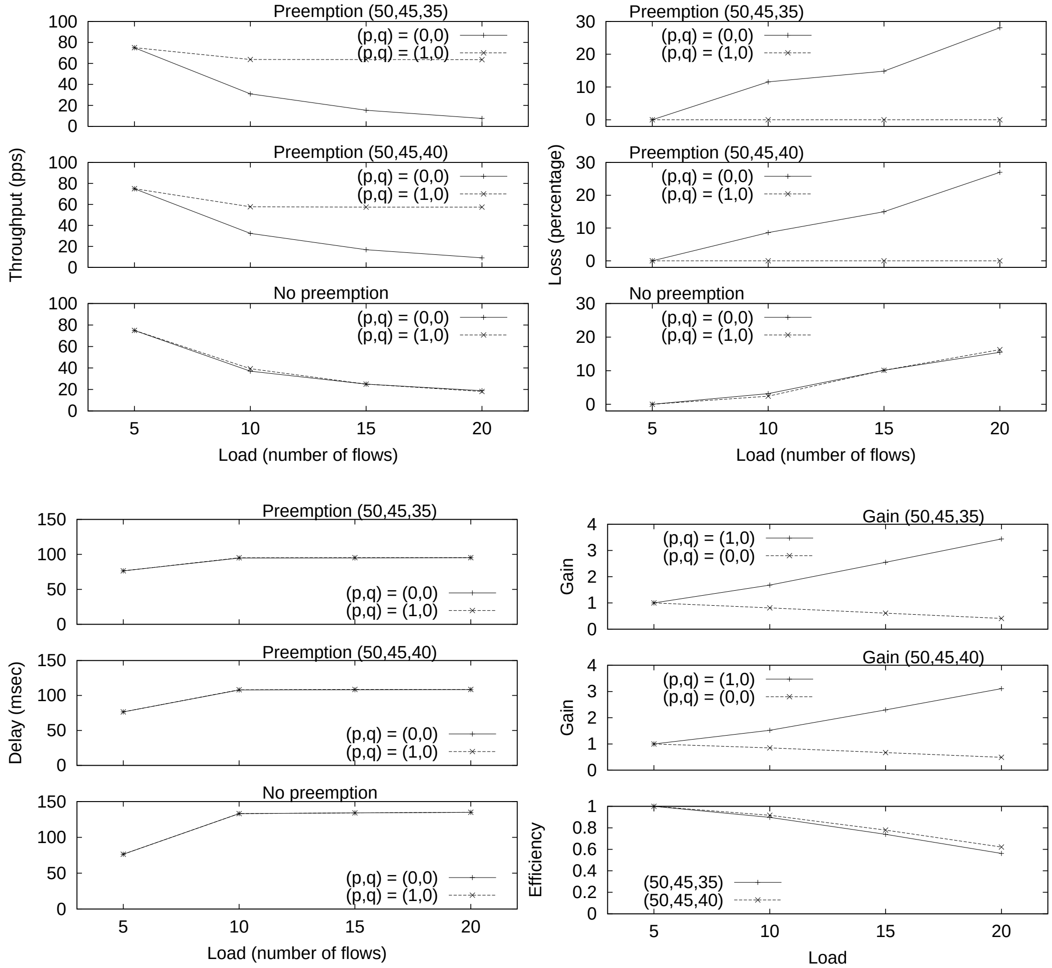

Fig. 4. Simulation results showing impact of CQ-ACM on TCP Throughput (top left), Loss (top right), Delay (lower left) and Gain and System Efficiency (lower right).

settings. In future work we hope to derive such expressions using Brownian Motion models of heavy traffic limits [11] to define optimal threshold settings.

Figure 4 shows the results for the TCP metrics with and without the CQ-AQM scheme. The upper two plots of the throughput set shows the TCP throughput in the presence of preemption, i.e., CQ-AQM. While the lower plot of each set shows the results for the non-preemption cases. Obviously, as more and more flows are added to the system, the individual throughput will decrease. For these simulation runs, the line rate or server speed is $375\ pps$, hence the packet service time is $\mu = 2.6667\ ms$. The propagation delay is set to $\delta = 30\ ms$. These values are consistent with a channel bandwidth of 1.544 Mbps as before.

As mentioned earlier, we have set the maximum TCP window size to $W = 4096$ bytes. Hence, due to the propagation delay, our system is window limited in the event that there is only a single TCP transmitter. However, for two or more TCP transmitters our system becomes server rate limited. So what is observed in the bottom plot of each set is that the multiple TCP flows are sharing the line bandwidth equally and as we add more flows the individual throughput diminish accordingly. The upper two plots of the throughput set show the impact of our CQ-AQM scheme on the high and low P-L TCP flows for the two different settings of the T_{low} value. We see that the TCP throughput of the high P-L flows reaches a fixed value of roughly 60 pps independent of the number of flows. While the throughput of the low P-L flows continuously decrease as the number of flows increases. Note that at 10 or more flows, we are overloading the system in that the low priority buffer is having to discard packets. While in all cases the line utilization is close to 100% due to the aggressive nature of the TCP flows.

The upper right plot in Figure 4 shows the impact of the CQ-AQM scheme on the TCP losses. Given the maximum window size and the fact that the low priority buffer is 50 packets, there should be no packet loss for 7 or less transmitters, while we should observe packet loss for 8 or more transmitters. This is indeed what we observe in the figure. As before, the bottom plot of the loss set show the results for no preemption, while the two upper plots show the results for the two preemption cases. We see that without preemption, both the low and high P-L TCP flows suffer packet losses which increase as the number of flows increase. However, when we invoke the CQ-AQM scheme, the high P-L TCP flows no longer suffer packet losses; hence their throughput correspondingly increase. Now all the packet losses are incurred by the low P-L TCP flows, as desired.

From Figure 4, we see that the delays initially increase as we increase the number of flows and as the buffer occupancy increases. However, the delays reach a plateau at a number of flows in excess of 8 due to the maximum window sizes of the flows and the low priority queue's buffer size. Comparing the results for the delays versus the throughput you will notice a strict relationship between the results for the high P-L flows. In this case, when preemption is implemented, the high P-L flows no longer are sharing the line rate with the low P-L flows. Instead, the high precedence flows become window limited with the invocation of preemption and their throughput is determined by the well known windowing expression, i.e., $T = W/RT$ where T is the flow throughput, W is the maximum window size and RT is the round trip delay for the flow's packets (including the propagation delay). Once the high P-L flows achieve their window limited rates, the low P-L TCP flows must share the remaining bandwidth. Hence, due to the finite delays due to the finite buffer sizes, the throughput of the high P-L TCP flows are independent of adding more low P-L TCP flows. While the throughput of the low P-L TCP flows is inversely related to the number of additional low P-L flows added to the system. We should expect no better behavior than this for a TCP transmitter/PHB combination having only packet loss as the feedback mechanism to the transmitter.

Finally, we consider the gain and the system efficiency for our results shown in the lower right plot in Figure 4. These show that the CQ-AQM scheme provides a high gain for the high P-L flows while diminishing somewhat the gain for the low P-L TCP flows, as expected. What is more interesting is the system efficiency shown in the bottom plot of the set. This represents the penalty paid by implementing a particular preemption scheme. Ideally, we would like the system efficiency to be unity. However, without *a priori* knowledge of the packet arrivals to the system, we suspect it is impossible to achieve an efficiency of unity for situations under extreme traffic overload. We do see a slight improvement in the efficiency when we increase the T_{low} threshold from 35 to 40 packets. We have not addressed the issue of optimal design of the parameters for specific implementations of the CQ-AQM scheme. However, perhaps it would be desirable to maximize the system efficiency for expected traffic patterns. This area of research is deferred to future investigations.

C. Mixed Traffic Results

In this section we investigate the TCP performance within the context of a mixed traffic model. The background traffic model we use is the model used in our previous UDP-only results above. As before, the traffic emulating VoIP flows are routed to the high priority queue and traffic emulating bursty data applications are routed to the lower priority queue. Both traffic types contain high and low P-L packet flows. We couple a single VoIP flow generating approximately 2.0 KBytes per second of traffic with a single bursty data source with $C_v = 2$ generating roughly 20 KBytes per second of traffic. Our smallest traffic load case is comprised of 5 TCP flows, 5 VoIP flows and 5 bursty data sources. Each flow type has one flow tagged $p = 1$ and four flows tagged $p = 0$. The TCP flows and the bursty data sources' traffic are marked $q = 0$ while the VoIP flows are marked $q = 1$. We then increase the background load by adding pairs of VoIP and bursty data traffic, one set at a time up to a maximum of 10 VoIP flows and 10 bursty data sources, while keeping the number of TCP flows fixed at 5. Hence, the background traffic represents a low of approximately 60% of the server rate up to a high of roughly 120% of the server rate while the TCP flows generate additional traffic.

Due to the high coefficient of variation in the traffic arrival patterns for the mixed traffic model, the buffer size, particularly for the low priority buffer had to be increased to maintain a reasonably small packet loss rates (our target loss rate was one packet in a thousand) at engineered loads of around 60%. This is based upon the results in the UDP-only case discussed above. Further, we investigate the CQ-AQM scheme with the same threshold settings as before. Specifically we set the $T_{low} = 500$ packets, the $T_{low}^{(cross-queue)} = 650$ packets, and the $T_{high} = 7$ packets.

Figure 5 shows the impact of background load on the TCP flow metrics. As the background load increases, the TCP throughput decreases. The most dramatic decreases are for the TCP flows with $p = 0$ for the case where the CQ-AQM scheme is active. However, this case roughly tracks that of the TCP throughput for the case of no preemption. When preemption is turned on, i.e., the CQ-AQM scheme is active, the $p = 1$ TCP throughput are relatively flat up to a utilization of around 90% of the server, and then the throughput begins to decrease slightly. This $p = 1$ TCP throughput decrease roughly tracks the slow increase in the $p = 1$ packet delays as the load increases beyond 90%.

The plot on the top right of Figure 5 shows the slight increase in the $p = 1$ packet delays for the case where preemption is enabled and as the load increases beyond 90% of the service rate. The packet delays for the case of no preemption are worse due to the fact that the CQ-AQM scheme begins packet discards when the buffer reaches 500 packets versus the total buffer size which is 750 packets. As before, with the CQ-AQM scheme active, the high P-L TCP flows achieve throughput according to the relationship $T = W/RT$, while the low P-L TCP flows have their throughput diminished due to the packet discards.

The plot on the lower left of Figure 5 shows the packet loss probability for the various traffic classifications both with and without preemption. The middle two curves track closely and represent the packet loss rate in the absence of preemption. When preemption is enabled, the packet loss probability for the $p = 0$ traffic becomes worse and the packet loss probability for the $p = 1$ traffic becomes zero. This is observed in the bottom line on the plot being zero for all load cases simulated.

Finally, the plot on the lower right in Figure 5 we present the results for the gain and system efficiency for our mixed traffic modeling. The upper plot of the set shows the gain for the low and high P-L traffic with preemption enabled. As expected

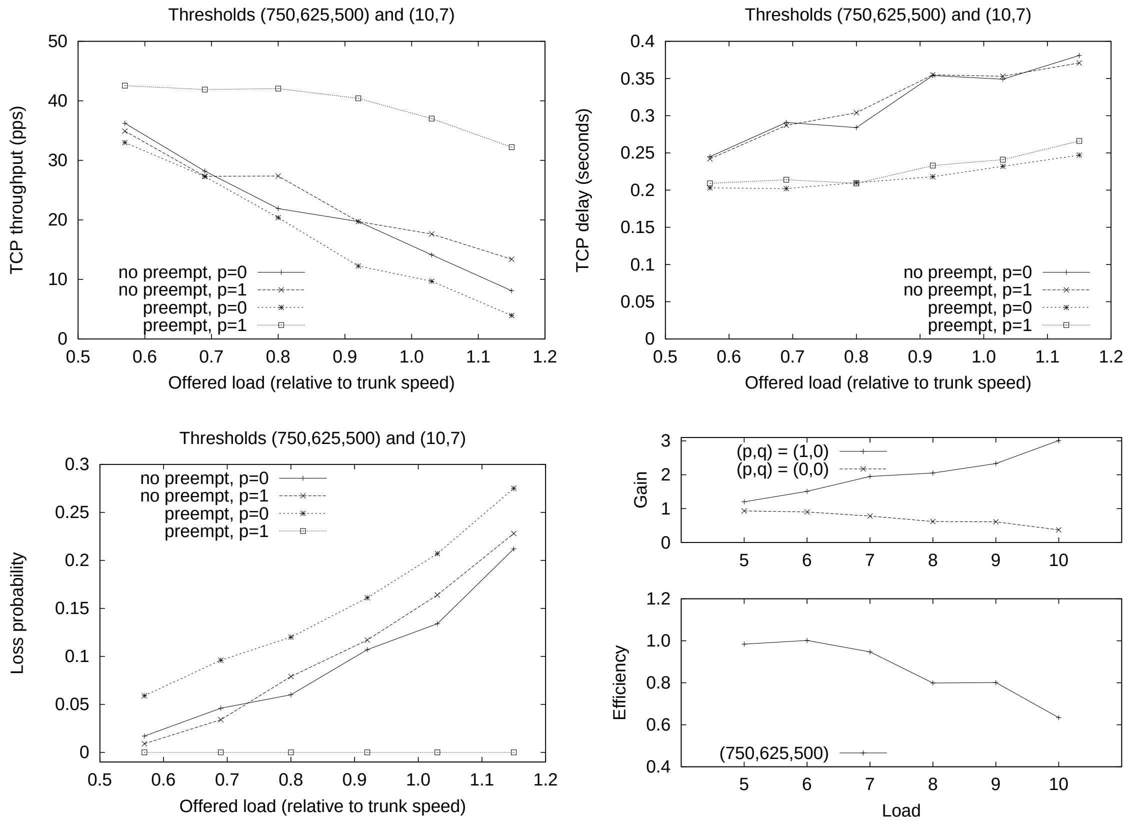

Fig. 5. Simulation results showing impact of CQ-ACM on TCP Throughput (top left), Delay (top left), Loss (lower left) and Gain and System Efficiency (lower right) in mixed traffic.

the gain for $p = 1$ traffic is greater than unity while the gain for the $p = 0$ traffic is less than unity. The lower plot of the set shows the system efficiency. As before, as the load increases the overall system efficiency decreases. Further studies are necessary to investigate optimal threshold settings for this traffic model based upon maximizing the system efficiency with the constraint that the packet loss for the $p = 1$ traffic is near zero. Further, it would be interesting to investigate a dynamic threshold setting based upon monitoring the incoming traffic characteristics. We leave these investigations for future research.

V. Previous Work

The majority of work in the open literature on Precedence and Preemption handling fall into the category of measurements and admission control for flow-based applications. An example work of this type is [15]. Although, we believe this represents a reasonable starting point for implementation of P&P handling in networks, we do not believe it is the final solution. These approaches do not address handling non-flow based applications, nor how to handle flow-based associated signaling messages for in-band signaling architectures. Further, we believe the reliance on reservation methods in highly dynamic, emergency and military, tactical wireless MANETs to be problematic.

There has been little or no work in the literature on the development of PHBs which are designed to simultaneously support application QoS and content importance P&P requirements. There has been an extensive body of literature on the development of schedulers, and hence PHBs, which support a range of application QoS requirements. This body of work includes, for example, the work of Floyd [10], Keshav [12] [13] and others. The work on PHBs is reflected in various IETF RFCs, e.g., [9] and [1]. The Expedited Forwarding (EF) PHB defined in [9] provides a low-jitter transport for real-time applications while also supporting variable bit rate data applications. The Assured Forwarding (AF) PHB defined in [1] provides a set of assured service types distinguished by levels of packet loss under engineered load conditions. It is unfortunate, that often the AF PHB is discussed in the context of precedence handling. While it does offer distinction amongst flows through variable levels of packet loss, these levels are not strictly ordered as often required for precedence handling and apply within the context of a single queue. Hence, they instead should be thought of as offering a set of throughput classes to different flow controlled applications.

We presented in [8] and [7] a more general discussion of methods, mechanisms and security issues with respect to general precedence-enabled, packet-level, transport networks. These works help put further context to our investigations of new PHBs and the analysis of the CQ-AQM scheme in this paper.

VI. Conclusions

Future commercial, financial, emergency and military, packet-based transport networks should support Precedence-Enabled Services. An important aspect of these P&P services in packet networks is the development of robust Per Hop Behaviors (PHBs) which simultaneously address application QoS requirements and information content importance requirements [4] [8] [7]. We have proposed and analyzed a natural extension to current QoS PHBs which we term the Cross-Queue Active Queue Management (CQ-AQM) scheme. Our CQ-AQM scheme relies solely upon local information, indicated in the packet protocol header, and hence can be used in broadband wired and low bandwidth wireless environments.

Our simulation studies showed that our local CQ-AQM PHB performed well under all scenarios. The CQ-AQM provided assured delivery of the high precedence traffic under UDP, TCP and mixed UDP and TCP cases. The protection to high precedence traffic was maintained even in situations where the overall system capacity was significantly exceeded. This PHB protected the high P-L TCP throughput during overload conditions, by maintaining zero packet dropping for the $p = 1$ traffic. Further, we defined two new metrics for the investigation and comparison of preemption schemes in this paper. These are the gain and the system efficiency. The gain represents the relative increase or decrease of the TCP throughput for the preemption mechanisms versus the comparable non-preemption results. The system efficiency represent a measure of the waste incurred by enabling preemption over the non-preemption case. The objective is to maximize system efficiency while maintaining $p = 1$ loss probabilities near zero.

In future studies we plan to investigate the application of our CQ-AQM mechanism to additional schedulers previously defined for wired networks. We further wish to investigate the application of our CQ-AQM scheme to new schedulers developed specifically for wireless MANET network environments. Finally, we are very much interested in developing methods to define optimal threshold settings based upon specific traffic patterns and to extend these capabilities to create adaptive threshold settings.

Acknowledgments

I would like to thank B. Farroha and P. Chimento of the JHU Applied Physics Laboratory for numerous enlightening discussion on this topic, B. Liebowitz of MITRE Corporation for encouragement in the development of the TCP results of this simulation study and Barry Sweeney of the SAIC Corporation for his careful and insightful reading of an earlier papers. I would like to thank all of the folks who participated in the 2005 P&P sub-team of the GIG QOS working group for all of their interesting and insightful discussions and presentations on the topic of precedence handling in packet data networks.

References

[1] Heinanen, J., Baker, F., Weiss, W. and J. Wroclawski, *The Assured Forwarding phb group.* Technical Report RFC2597, IETF, 1999.
[2] Balakrishnan, H., Padmanabhan, V.N., Seshan, S. and R.H. Katz, *A Comparison of mechanisms for Improving TCP Performance over Wireless Links.* IEEE/ACM Transactions on Networking, December 1997.
[3] Borst, S., Mandjes, M. and M. van Uitert, *Generalized processor sharing with light-tailed and heavy-tailed input.* IEEE/ACM Transactions on Networking, 11(5):821–834, October 2003.
[4] Chairman Joint Chiefs of Staff. *Policy, responsibility, processes and administration for the DoD GIG networks DRAFT.* Technical Report CJCSI 6215.02a, Department of Defense, July 2004.
[5] Cole, R.G. and P. Chimento, *Modeling and Preliminary Simulation Studies for Packet-Based Precedence and Preemption for FCS Communications.* Army Science Conference 2006, Orlando, FL, USA, December 2006.
[6] Cole, R.G., *Impact of Precedence Enabled PHBs on TCP Flows.* IEEE MILCOM'07 Conference, Orlando, FL, USA, December 2007.
[7] Cole, R.G. and B. Farroha, *Implications of Precedence and Preemption Requirements on Packet Based Transport Architectures.* submitted to IEEE MILCOM 2007, Orlando, FL, USA, October 2007.
[8] Farroha, B.S., Cole, R.G., Farroha, D. and A. DeSimone, *An Investigative Analysis of Precedence and Preemption Issues Associated with the GIG's P&P Architecture.* SPIE 2007, Orlando, FL, USA, April 2007.
[9] Davie, B., Charny, A., Bennett, J.C.R., Benson, K., Le Boudec, J.Y., Courtney, W., Davari, S., Firoiu, V. and D. Stiliadis, *An Expedited Forwarding PHB.* Technical Report RFC 3246, IETF, 2002.
[10] Floyd, S. and V. Jacobson, *Link-sharing and Resource management Models for Packet Networks.* IEEE/ACM Transactions on Networking, Vol.3, No.4, August 1995.
[11] Heyman, D., *A Diffusion Model Approximation for the GI/G/1 Queue in Heavy Traffic*, Bell Labs Technical Journal, Vol. 54, No. 9, November 1975.
[12] Zhang, H. and S. Keshav, *Comparison of rate-based service disciplines*. In Proceedings of the ACM SIGCOMM'91, pp.113-122, Zurich, Switzerland, September 1991.
[13] Keshav, S., *An Engineering Approach to Computer Networking*. Addison-Wesley, 1997.
[14] The International Telecommincations Union (ITU), *Multi-Level Precedence and Preemption (MLPP) Service*. ITU-T Recommendation I.255.3, 1990.
[15] Olaussen, E. and A. Karlsen, *A Policy-Based priority and Precedence Framework for Military IP Networks*. MILCOM 2007, Monterey, CA, USA, October 2004.
[16] Sahu, S., Towsley, D. and J. Kurose, *A quantitative study of differentiated services for the internet.* Technical Report UMass CMPSCI Technical Report 99-09, University of Massachusetts, 1999.
[17] Trivedi, K., *Probability & Statistics with Reliability, Queuing, and Computer Science Applications* Prentice Hall, Inc. 1982