Impact Of Precedence Enabled Per Hop Behaviors on TCP 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—**In the Department of Defense's (DoD) Global Information Grid (GIG) transport network, packet handling must provide preferential transport to high Precedence traffic under all networking conditions, specifically conditions of resource scarcity, e.g., network overload conditions, while simultaneously satisfying packet scheduling required to meet application Quality of Service (QoS) needs. Our approach to this duality is to enhance Active Queue Management (AQM) techniques to provide Precedence and Preemption (P&P) capabilities and rely upon standard, well studied QoS Per Hop Behavior (PHB), 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 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 and analytical modeling, 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&P level. Our previous studies concentrated on general non-flow controlled traffic and showed that our algorithms performed extremely well. In this paper we extend our analysis to flow-controlled traffic by incorporating TCP traffic models into our simulation studies. 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 for TCP flows as well.

I. INTRODUCTION

The U.S. DoD's Future Combat System (FCS) is reliant upon the development of a reliable, resilient communications capability under harsh, battlefield environments. During periods of crisis, the communications infrastructure must be capable of providing preferential delivery of information based upon the Future Force Warrior's indication of the importance of the information, as indicated by the message Precedence-Level (P-L). It is imperative for the GIG, and hence FCS communications, to support P&P capabilities for all Command and Control (C2) messages and applications [4]. In order to support P&P, the GIG all-Internet Protocol (IP) packet-based transport network must develop new packet handling and forwarding algorithms to simultaneously support application Quality of Service (QoS) and content P&P. 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] [12]. However, to date, little work exists to design PHB algorithms which simultaneously deliver QoS to applications and P&P transport to information. In [5], we proposed and analyzed PHB algorithms which accomplished simultaneous QoS and P&P handling. In this paper, we extend our analysis to TCP-based traffic flows and mixed traffic systems.

Experience in providing P&P capabilities in communications services fall into two camps, i.e., traditional telephony services, e.g., the Defense Switched Network (DSN), and message handling services, e.g., the Automated Message Handling System (AMHS). Naively mapping these onto an all-IP, packet-based transport network like the GIG is problematic. The DSN handled 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 the GIG. Message handling systems 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 [1] [7], not P-L handling.

In the GIG 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 PHB, 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 applications. Under network congestion situations, the enhanced AQM layer provides the necessary P&P preferential packet handling to high P-L information. Our approach concentrates on local processing only. 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 wired and 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 *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 handling. Efficiency 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 an efficiency measure in our analysis.

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 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 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. In that study the traffic was simulated via independent, non-flow controlled packet sources. The sources emulated data transactions of various characteristics and voice over packet constant bit rate traffic. In this paper we extend our analysis to focus on the impact of the CQ-AQM scheme in the presence of additional, flow-controlled traffic sources such as TCP-based traffic flows.

Through extensive simulation studies and analytical modeling, 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 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 *Efficiency*. The Gain measures the benefit of Preemption to the flow in question. The 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 TCP traffic sources.

II. APPROACH

Most works addressing Precedence handling in packet networks attempt to strictly emulate Precedence handling in circuit-based voice networks, like the Defense Switched Network (DSN). These architectures attempt to rely solely upon signaling and reservation protocols to manage the scarce networking resources during overload situations. However, these proposals are naive in that they only address constant bit rate flow oriented applications like Voice over IP (VoIP). A large proportion of the evolving C2 data applications in the DoD's vision of NetCentric Warfare are not flow based. Even for flow-based, constant bit rate applications, most if not all network architectures discussed within the DoD GIG planning meetings rely upon a common data transport integrating application transport and 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 methods. Finally, it is well understood that a critical component of the DoD NetCentric Warfare plans rely upon wireless mobile ad-hoc networks (MANETs). In these environments, due to their high mobility and dynamic communications 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 enabled GIG. This does not say that signaling and reservation protocols are not important in the architecture. We only believe that the architecture cannot solely rely upon reservation techniques. These arguments are more fully discussed in a companion paper [6].

Hence, we have embarked upon a program to investigate the ability of local packet handling methods to provide aspects of Precedence and Preemption. Local methods for Precedence handling include new definitions of Per Hop Behaviors (PHBs) which would simultaneously support application QoS and content 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 [6]. 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 local thresholds which trigger discarding of lower P-L traffic within the specific queues where they are triggered. The threshold indicated by (c) is a non-local 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

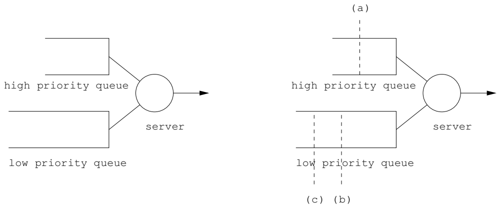

Fig. 1. The baseline scheduler and active queue management scheme.

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 in a straightforward manner and is not limited to the simple two queue system we use here for 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 treatment. Because this represents an initial study of new PHBs, we were interested in assessing its 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 in our performance studies.

A. Methodology

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. To simplify the presentation of results and the discussion of mechanisms, we model only two queues in this work. “Priority treatment” can mean strict priority, or preferential scheduling based upon a Weighted Round Robin or Deficit Round Robin scheme. 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 loss 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 and throughput performance 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 loss and delay.

For our initial studies, we investigated 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 local threshold, i.e., $T^{(local)}_{high}$ and $T^{(local)}_{low}$ respectively, where $T^{(local)}_{high} \leq B_{high}$ and $T^{(local)}_{low} \leq B_{low}$. 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 PLs, where $p = 1$ gets preferential treatment over $p = 0$. In the event that the buffer occupancy of the low priority queue equals or exceeds $T^{(local)}_{low}$, then the active queue management denies access to all packets with $p = 0$. In the event that the buffer occupancy of the high priority queue equals or exceeds $T^{(local)}_{high}$, then the active queue management denies access to all packets with $p = 0$. In the event that the buffer occupancies drop below their local thresholds, then their respective active queue management allows all PL traffic access to the queues [1].

These local thresholds help address the problem of Preemption within each local queue, but there are cases where low QoS class data is tagged high PL and high QoS class voice is tagged low PL and we need to be able to communicate a low priority buffer overflow to the high priority buffer management in order to prevent PL inversion. To prevent this situation from happening, we implement one additional threshold in the low priority queue, i.e., $T^{(non-local)}_{low}$, where $T^{(local)}_{low} \leq T^{(non-local)}_{low} \leq B_{low}$. When the buffer occupancy of the low priority queue equals or exceeds $T^{(non-local)}_{low}$, then the high priority queue management scheme causes all low PL 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 $T^{(local)}_{low}$. When we set $T^{(local)}_{low} = T^{(non-local)}_{low} = B_{low}$ and $T^{(local)}_{high} = B_{high}$, 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 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. The program is a work-in-progress and as we continue with this analysis, it will be expanded to include additional AQM and scheduling features.

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

---

[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.

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 200 milliseconds, and the off-times are 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 bursty state, where the inter-arrival time is 460μseconds and a lower-intensity state where the inter-arrival time is 46 milliseconds. 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 of 4.0.

We modeled the TCP traffic source implementing a selective acknowledgment scheme as analyzed in [2]. We choose 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, tactical MANETs where local processing intelligence is necessary due to the network dynamics. In wireless, tactical MANETs, reservation-based architectures for P&P handling are bound to fail. 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 eight in this study. Multiple instances of the TCP source are configurable in our simulation tool. 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 and background load.

IV. RESULTS

In this section we present our modeling and simulation results. We used our flexible simulation modeling facility to develop two traffic scenarios for investigation within this paper. The first traffic scenario consists of only TCP flow controlled data traffic. These results are investigated first. We then develop a mixed traffic scenario, basically adding TCP flow-controlled traffic onto our previously studied [5] mixed VoIP and bursty, non-flow controlled UDP-based data traffic. This traffic scenario is discussed second.

For each of these traffic scenarios, we concentrate on the ability of our CQ-AQM scheme to support the TCP-flows under overload conditions. Our previous study [5] reported metrics for the non-flow controlled applications. The metrics for the 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.

A. TCP-Only Results

We first investigated the impact on TCP flows alone. 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 local low threshold $T_{low}^{(local)}$ is set to 35 or 40

---

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

packets; two cases were investigated. The local high threshold setting was irrelevant to this set of simulation runs. The non-local low threshold $T_{low}^{(non-local)}$ 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 settings. In future work we hope to derive such expressions using Brownian Motion models of heavy traffic limits to define optimal threshold settings.

Figure 2 shows the results for the TCP throughputs with and without the CQ-AQM scheme. The bottom plot of the set shows the TCP throughputs in the absence of any Preemption. 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 $\delta = 30\ ms$. We chose these values to roughly emulate a channel bandwidth of 1.5 Mbps. This relatively low bandwidth was chosen as typical of future tactical radio systems. Further, detailed QoS handling as modeled here is relatively more important for lower speed interfaces versus high speed interfaces typical of wired backbone networks.

As mentioned earlier, we have set the maximum TCP window size to $W = 8$ packets. 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 the set is that the multiple TCP flows are sharing the line bandwidth equally and as we add more flows the individual throughputs diminish accordingly. The upper two plots of the set show the impact of our CQ-AQM scheme on the high and low Precedence Level TCP flows for the two different settings of the $T_{low}^{(local)}$ 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.

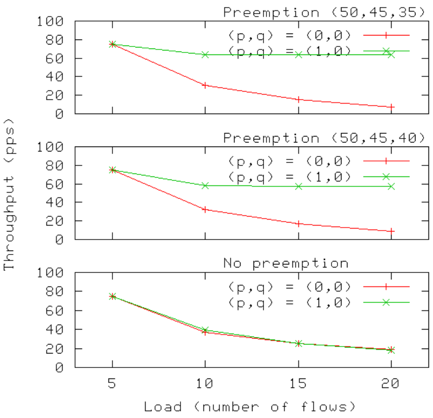

Fig. 2. Simulation results showing impact of CQ-ACM on TCP throughput.

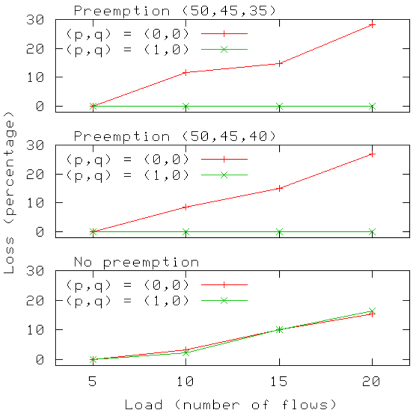

Fig. 3. Simulation results showing impact of CQ-ACM on TCP loss.

Figure 3 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 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 throughputs correspondingly increase. Now all the packet losses are incurred by the low P-L TCP flows, as desired.

Figure 4 shows the impact of the CQ-AQM scheme on the packet delays. As before the bottom plot of the set represents the no Preemption case while the upper two plots of the set represents the results for the two Preemption cases. 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 throughputs 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 throughputs are determined by the well known windowing expression, i.e., $T = W/RT$ where T is the flow throughput, W is the maximum window

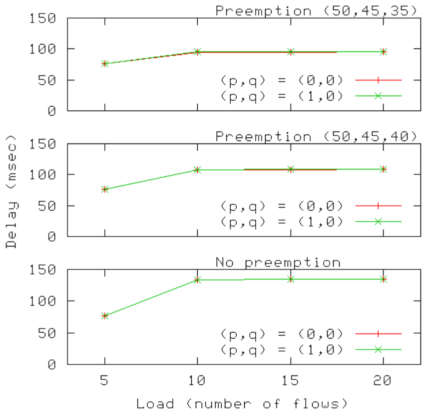

Fig. 4. Simulation results showing impact of CQ-ACM on TCP delay.

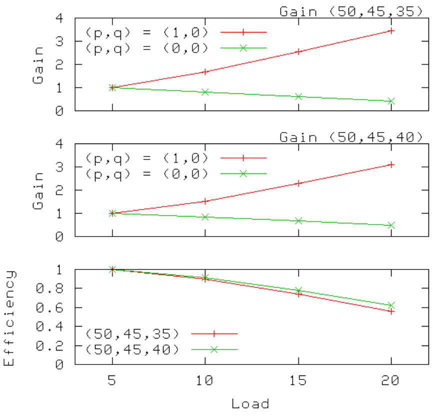

Fig. 5. Simulation results showing impact of CQ-ACM on TCP Gain and System Efficiency.

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. This phenomenon is observed in the results in Figures 2, 3 and 4. 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. These results are shown in Figure 5. The upper two plots of the set show the Gain for the low and high P-L flows for the two Preemption cases considered. 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 $T_{low}^{(local)}$ 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.

B. Mixed Traffic Results

In this section we investigate the TCP performance within the context of a mixed traffic model. We add background traffic to our system and track the behavior of the TCP flows. The background traffic model we use is identical to a traffic model used in our previous study [5]. The background traffic consists of two type of traffic, one set emulating UDB-based voice over packet and the other type emulating a highly bursty, UDB-based non-flow controlled data. 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 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 of our previous study [5]. Further, we choose to investigate the CQ-AQM scheme with the same threshold settings as investigated in [5]. Specifically we set the $T_{low}^{(local)} = 500$ packets, the $T_{low}^{(non-local)} = 650$ packets, and the $T_{high}^{(local)} = 7$ packets.

Figure 6 shows the impact of background load on the TCP flow throughput. 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 throughputs for the case of no Preemption. When Preemption

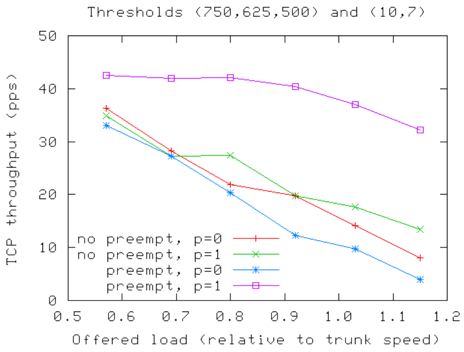

Fig. 6. Simulation results showing impact of CQ-ACM on TCP throughput in mixed traffic.

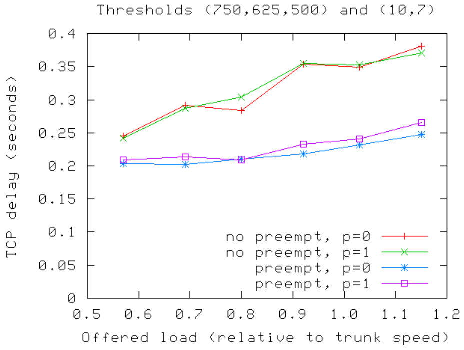

Fig. 7. Simulation results showing impact of CQ-ACM on delay in mixed traffic.

is turned on, i.e., the CQ-AQM scheme is active, the $p = 1$ TCP throughputs 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%.

Figure 7 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 throughputs according to the relationship $T = W/RT$, while the low P-L TCP flows have their throughputs diminished due to the packet discards.

Figure 8 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, in Figure 9 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 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.

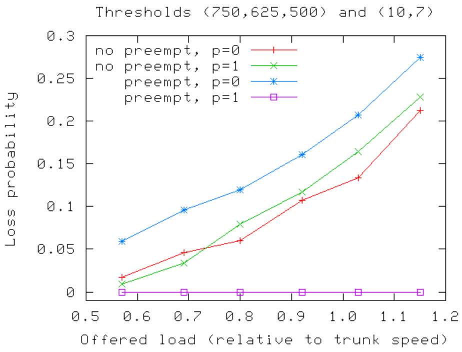

Fig. 8. Simulation results showing impact of CQ-ACM on loss in mixed traffic.

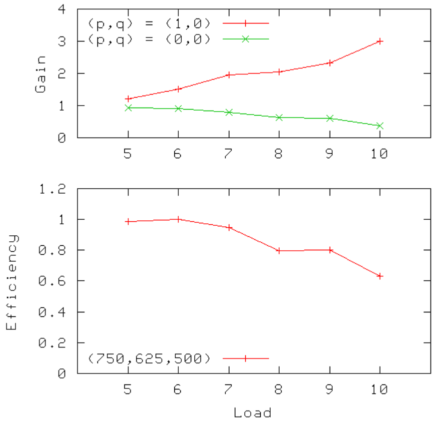

Fig. 9. Simulation results showing impact of CQ-ACM on TCP Gain and System Efficiency in mixed traffic.

V. Previous Work

The majority of works in the open literature on Precedence and Preemption handling fall into the category of measurements and admission control. An example work of this type is [11]. Although, we believe this represents a reasonable starting point for implementation of P&P handling in wire-based DoD 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, 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 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 [8], Keshav [9] [10] and others. The work on PHBs is reflected in various IETF RFCs, e.g., [7] and [1]. The Expedited Forwarding (EF) PHB defined in [7] 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 required for precedence handling. Instead they should be thought of as offering a set of throughput classes to different flow controlled applications.

We presented in [5] our approach to enhancing PHBs for Precedence handling. There we rely upon the extensive work in the literature with respect to QoS schedulers and choose to define a method to extend these well known schedulers to be Precedence-aware. The results in [5] were very encouraging. This is the approach we take in this paper.

VI. CONCLUSIONS

We have extended our analysis of our proposed CQ-AQM scheme for provide aspects of Preemption based upon solely local computations. Our initial study in [5] proposed the CQ-AQM scheme as a new PHB and investigated through simulation studies the performance of the PHB in the presence of mixed, non-flow controlled UDP-based traffic. In this paper, we extend our studies to investigate the impact of the CQ-AQM scheme on flow controlled TCP-based traffic. Specifically, we implemented in our simulation tool, a selective acknowledgment TCP traffic source similar to the SEL_ACK_TCP analyzed in [2]. This TCP variant showed exceptional throughput performance in the context of wireless networking. We chose this version of TCP for our modeling due to our interest in developing robust tactical, wireless MANET systems for military deployments.

Our simulation studies showed that our local CQ-AQM PHB performed well. This PHB protected the high P-L TCP throughputs 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 other 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 this simulation study and Barry Sweeney of the SAIC Corporation for his careful and insightful reading of an earlier draft of this paper. 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. Finally, I would like to thank the reviewers for their insightful and helpful comments on an earlier draft of this paper.

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. and B. Farroha, *Implications of Precedence and Preemption Requirements on Packet Based Transport Architectures.* submitted to IEEE MILCOM 2007, Orlando, FL, USA, October 2007.

[7] 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.

[8] 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.

[9] 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.

[10] Keshav, S., *An Engineering Approach to Computer Networking.* Addison-Wesley, 1997.

[11] Olaussen, E. and A. Karlsen, *A Policy-Based priority and Precedence Framework for Military IP Networks.* MILCOM 2007, Monterey, CA, USA, October 2004.

[12] 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.

[13] Trivedi, K., *Probability & Statistics with Reliability, Queuing, and Computer Science Applications* Prentice Hall, Inc. 1982