A Novel Probabilistic Deadline Scheduling Mechanism for DCCP

Daniel Wilson
School of Information Technology
Murdoch University
Perth, W.A., Australia, 6150
Email: Daniel.Wilson@murdoch.edu.au

Mike Dixon
School of Information Technology
Murdoch University
Perth, W.A., Australia, 6150
Email: m.dixon@murdoch.edu.au

Terry Koziniec
School of Information Technology
Murdoch University
Perth, W.A., Australia, 6150
Email: t.koziniec@murdoch.edu.au

***Abstract*—This paper introduces a novel cross layer probability based deadline scheduling mechanism designed specifically for real time Data Congestion Control Protocol (DCCP) flows. Scheduling in this mechanism is determined based on the probability of a data packet being received within its useful lifespan. In order to predict this probability, DCCP is modified to access routing table information used by CISCO Systems Inc.'s Enhanced Interior Gateway Routing Protocol (EIGRP), to estimate the approximate forward path delay period. Once the packet's probability of arriving within its useful lifetime is determined, the scheduling algorithm then places the packet into one of three predefined queues to ensure all packets that are received are given the highest chance of being delivered within their useful lifespan period. In addition to describing the design of the mechanism, this paper will also present proof of concept modelling carried out to quantify the effectiveness of the mechanism. The results presented in this paper show the mechanism described is able to predict the time a packet will need to traverse the network using EIGRP's metrics with greater than 90 percent accuracy (on average) in the tested topologies. The results will also show the mechanism is stable and able to operate in medium sized networks with marginal overhead.**

***Index Terms*—congestion control; real time; stale packets; queuing; scheduling;**

I. Introduction

This research highlights how real-time data possesses a finite lifespan and describes a mechanism that allows this lifespan property to be incorporated into DCCP [1] to allow more efficient transportation of DCCP-Data packets carrying real-time data. To achieve this, this paper will present a mechanism that can be employed on intermediate Bandwidth Optimization Devices (BODs), that will calculate a packets probability of arriving within its useful lifespan to determine if the packet should be sent or not, and the priority needed to send it. In order to do this, information relating to the network located between the device where the forwarding decision is being made and the end point, is derived from EIGRP [2]. This paper will begin with a high level overview of the mechanism. Next, specific details describing the mechanism will be presented to show how EIGRP metrics are used to calculate a packet's probability of arriving within its usefil lifespan. This is followed by description of a scheduling scheme that prioritizes packets based on their probability of arriving in time. Following this, proof of concept experimentation that was used to quantify the effectiveness of the scheme is presented. The paper concludes with discussion of the experiments, their associated limitations and finally a conclusion of the paper and its findings.

II. Related Research

Gurtov and Ludwig [3] are credited as being the first to introduce packet lifetime discard mechanisms to the research community. In their model, each packet has a packet life variable embedded into a custom built IP header option field. Based on this information, packets that have existed beyond their useful time are purged on intermediate routers placed between the sender and the receiver. They also propose that packets not likely to be delivered within the valid time should also be purged, further increasing efficiency of bandwidth utilization.

Gurtov and Ludwigs' research showed that purging stale packets on the network was able to improve overall performance for all applications utilizing the network. Specifically, the reduction in the overall delay in delivery times experienced by fresh packets, when stale packets were purged, was significant. Performance of the application for which packets were purged appeared to be unaffected by the purging process, as the packets would likely have been dropped by the application once received if the purging action had not taken place.

While Gurtov and Ludwigs' study was, and remains, the most significant work in the area, there have been a number other researchers who have carried out similar research. Yuen and Yue in [4], present a scheme of purging stale packets and then subsequently prioritizing real time packets in a data queue. Their research was of a mathematical nature and no mention was made regarding the practical implementation of the scheme. Their findings showed that by purging stale packets from queues, fresh packets in those same queues would experience shorter delays and less contention for available bandwidth.

Gurtov and Ludwigs' foundation idea was also used by Chebrolu's and Rao in [5]. In this work, they examine bandwidth optimization specifically for MPEG video where bandwidth was severely restricted. Using a combination of the Gurtov and Ludwigs packet discard mechanism and knowledge of how MPEG video standards operate, they created Minimal Cost Drop (MC Drop) [5]. MC Drop is used firstly to determine

 ISBN: 978-1-61208-186-1

which packets already are, or are likely to become stale, due to the high levels of congestion on a network link. Packets that are already stale are dropped immediately. Packets that are likely to be become stale are applied to a policy to determine which of these packets should be dropped first to ensure the best video quality at the receiver side. The results obtained from experiments showed that MC Drop allowed for higher overall video quality in instances where there was a severely restricted bandwidth link, when compared to conventional non discard techniques.

TCP-RTM [6] was designed by Liang and Cheritan in 2002, and proposes a number of extensions that make TCP more suitable to real time data. One of these extensions involves marking stale packets to speed up the delivery of fresh packets. Radovanovic et al., in [7], took Liang and Cheritan's work one step further towards Gurtov and Ludwigs' ideology by using the SNOOP protocol to purge packets identified by the TCP-RTM extension as being stale. In addition, they implemented a mechanism for suppressing packet retransmission and congestion events resulting from the purging process. Results gathered from a NS2 simulation found that the packet discarding scheme improved overall network goodput and reduced the overall delay experienced by packets when compared to conventional non discard techniques. This technique offered improved performance for real time applications using the TCP protocol and has much promise.

All the above findings show unequivocally the potential improvement possible by implementing a packet discarding scheme for real time traffic. There does not however appear to be a packet life discard scheme proposed or implemented specifically for the DCCP protocol, a protocol which appears so well suited to such a scheme. Given that the DCCP protocol has been designed for the application of real time data, adding such a scheme could potentially be of major benefit to the protocol. Additionally, a number of problems which have hindered the wide spread adoption of packet discarding schemes, such as the loss of congestion control integrity and having to suppress retransmissions, are removed in DCCP as it is an unreliable protocol. Using Gurtov and Ludwigs' packet discard principles, this paper presents an efficient and effective packet discard policy designed specifically for the nuances of the DCCP protocol.

III. High level overview

In this section a high level overview of how the probability based scheduling (PBS) mechanism operates is presented. In the PBS mechanism, the application layer on the sending device is configured to specify the packet's maximum lifespan value concurrently with the passing down of the payload data. This value represents the maximum time the data is of use to the receiving application. Upon receiving data from the application layer, DCCP on the sending device timestamps the packet with a birthtime variable. The birthtime variable is extracted from a NTP synchronized system clock and specifies the exact time when the packet was created. The payload data will then be encapsulated into a generic DCCP-Data packet with the two above-mentioned variables appended as additional lifespan option fields. The packet will then be passed to the relevant network layer protocol for transportation across the network.

Located within the network, between the sender and the receiver, will be a number of intermediate BODs. These devices are modified/enhanced layer 3 network devices, such as routers and layer 3 switches (switched networks). In order to convert these devices into BODs, the PBS mechanism described in this paper, or other similar mechanisms, are added to the devices to allow them to utilize the lifespan variables in the DCCP data packets. In this scheme, when the BOD receives a packet, it also queries/extracts information from the routing protocol in order to determine the approximate time it will take for the packet to reach its intended destination. Once this time is known, the PBS mechanism then determines if the packet has sufficient lifetime remaining to reach the destination. Packets are then be assigned a probability of arriving on time, and categorized into four distinct categories. This allows advanced scheduling decisions to be made in order to offer the greatest good to all packets attempting to traverse the network. In addition, packets that have no possibility of arriving in time are also removed from the network making additional bandwidth available to other flows which would be impacted by the transmission of these stale packets.

IV. Definition of terms

In the following sections a number of acronyms will be used in order to keep the specification concise. In this section these acronyms will be defined with an accompanying description of where they are derived from. *BT* will be used in place of birth time. This variable is created when the sender receives data from the application layer and encapsulates the payload into a DCCP-Data packet. This variable is obtained from the system clock and is the exact time that the DCCP packet comes into existence. When the application layer passes the information down to the DCCP layer, it also passes down a *Maximum Time-to-Live (MTTL)* variable. This variable represents, in milliseconds, the amount of time the real-time data is valid for. In order to determine if a packet has expired or become stale, the MTTL value is added to the BT value to get the *Explicit NTP Expiry Time (ENET)*. At any point in the transmission, an intermediate device can query the NTP clock on the system for the *Current NTP Time (CNT)*. If the $ENET > CNT$ then the packet's contents are stale. In order to find out how much life time a packet has remaining the *Time-left-to-live (TLTL)* variable is used. To calculate the TLTL the ENET is subtracted from the CNT i.e. $(TLTL = ENET - CNT)$.

The next acronym that will be used is the *Time-Left-In-Queue (TLIQ)*. This variable represents the minimum amount of time it will take for a packet to be transmitted in the current queue if no change is made to the position of any packets in that queue. The final acronym used is the *Time-Needed-To-Cross-Network (TNTCN) variable*. This value is derived from the routing protocol and represents the estimated time that will be needed for a given packet to be delivered from

 ISBN: 978-1-61208-186-1

the current device to the destination network based on current network conditions. How the TLIQ and TNTCN variables are calculated will now be presented in greater detail.

A. Time left in Queue (TLIQ)

From the moment the PBS mechanism is initiated, it is vital that queue depth information is monitored and updated constantly for each of the queue categories. In addition to this, the current expected queue delay for newly arriving packets must also be calculated continuously. To achieve this, the *Time-Left-In-Queue (TLIQ)* variable is calculated every time a packet is received by the BOD device. The TLIQ value represents the time a packet will remain on the device for while in the outgoing interface queue and is critical in determining whether a packet should be placed in the critical queue or the normal queue. The formula for calculating the TLIQ value is shown in Equation 1.

$$TLIQ = \frac{TSQ}{IS * BA} \quad (1)$$

In order to calculate the TLIQ value for a newly arriving packet, the following steps take place. First, the interface service rate is calculated by taking the interface speed (IS) and multiplying it by the bandwidth percentage available to that queue (BA). Next, the queue depth information is accessed to determine the cumulative size of all existing packets in the queue in kilobits (TSQ). From these two variables, the time that is required for the queue to be serviced is deduced. When a new packet arrives, the TLIQ value must be readily available to the packet categorization mechanism so it can determine whether the packet belongs in the critical queue or the normal queue.

B. Time Needed to Cross Network (TNTCN)

The novel element of this scheme is provided through the addition of the *Time-Needed-to-Cross-Network (TNTCN)* variable. This variable is derived using information obtained from a routing protocol and is used to provide an estimate of the time that will be required for the packet to reach the destination network. From this formula, the probability of the packet arriving at the intended destination within its useful lifespan is then calculated. EIGRP is used in the PBS mechanism to calculate the TNTCN variable. There are a number of reasons why EIGRP was selected to provide this information. Firstly, EIGRP uses a number of variables to calculate its routing and topology tables. These variables include Delay, Reliability, Load, Bandwidth, Hop count and MTU. Each of these variables represent unique conditions that exist in the network between the queue and the destination network.

To reduce the computational overheads of the PBS mechanism, these EIGRP variables are utilized for the purposes of determining the probability of a packet arriving within its lifespan. The ethos behind this research is to promote cross layer cohesion and reduce computational overheads by alleviating task repetition wherever possible. To configure the PBS mechanism to calculate the variables EIGRP already provides was deemed unnecessary when these variables were already readily available. To add to this, the algorithms used to derive these variables in EIGRP are well matured and have been proven to work efficiently.

Another reason why the EIGRP routing protocol was selected was because it offers fast convergence times through its Diffusing Update Algorithm (DUAL)[8]. Fast convergence times equate to higher levels of accuracy in relation to actual network conditions, which is a prerequisite for the PBS mechanism. If changes to the network occur, the routing protocol must be able to detect these quickly in order to provide accurate information to this mechanism. Failure to do so in a timely manner will lead to incorrect scheduling and prioritization of packets. The drawbacks to using the EIGRP protocol are that the choice to do so limits the PBS mechanism purely to topologies that are configured with CISCO Systems Inc.'s devices. In addition, EIGRP is an interior routing protocol and therefore, this limits the scale of the PBS mechanism to single autonomous systems and not the broader Internet in its current form. Finally, it is also important to mention that EIGRP does utilize bandwidth in order to communicate routing updates to other routing devices on the network. This paper will assume that the EIGRP protocol has been installed on the network device for core routing functionality and not for the express purposes of this mechanism.

V. Mechanism Details

There are two main components that make up the PBS mechanism. Firstly, there is the categorization component that is used to determine which category and subsequently which queue is appropriate for an incoming packet. The second component is a packet schedule, which services the respective queues in such a way as to ensure strict adherence to the queue's allocated bandwidth percentage. These components will now be discussed in greater detail.

A. Categorization of Packets

1) Queue Structure: The first phase of this mechanism involves the categorization of incoming packets into one of three unique queues based on information gathered from the BT and MTTL variable in the DCCP header option fields. This section will commence by describing the function/role of each of the three queue classes.

Discard queue

The first packet queue class is for packets that have expired, or for packets that will expire before they reach their destination. This queue, known as the discard queue, will employ techniques that remove stale packets from queues. In order to qualify for this queue the following two criteria are checked. Firstly, if the packet arrives stale $(ENET < CNT)$, then the packet is placed in the discard queue. Secondly, if the packet will not reach its destination before becoming stale $(TLTL < TNTCN)$, then the packet will also be placed in

the discard queue.

Critical Queue

The second queue class, named the critical queue, is for packets that will expire unless a prioritization action takes place to prevent them from doing so. Specifically, if the process of passing through the queue on the BOD will directly lead to the packet becoming stale, then the packet is placed into this queue. The critical queue is allocated a predetermined amount of bandwidth to service packets at a prioritized rate. To qualify for this queue a packet must meet the following criteria. The packet must have sufficient lifespan remaining to allow it to travel across the network to its intended destination without becoming stale, i.e. $(TLTL > TNTCN)$. In addition, to qualify for this queue the packet must be in a position where the cumulated time needed for queuing, (TLIQ) and the time needed for the packet to cross the network, is greater than the time the packet has remaining before becoming stale (TLTL). The notion behind this queue is that packets that can reach the destination if they are not delayed excessively by the queuing process on the BOD device, are given the best chance to do so by being placed in a smaller more rapidly serviced queue. If this action does not occur, the packets become stale before reaching their destination due to the queue delay on the BOD.

Normal Queue

The final queue class is for packets that will likely reach the destination network within their respective lifespan provided there are no abnormal network fluctuations or extreme changes in network conditions. To qualify for this queue, the packet must possess a lifespan (TLTL) greater than the time needed for queuing and the time needed for the packet to traverse the network to the destination device. I.e. $TLTL > (TLIQ + TNTCN)$. Packets in this queue will be offered a guaranteed position and guaranteed bandwidth allocation ensuring the queue time is deterministic. Packets that are placed into this queue are placed there on a first-in-first-out (FIFO) basis and no packet is ever to be placed in front of a pre-existing packet. As this queue is given the majority share of available bandwidth and offers guaranteed service rates, it creates incentive for the application selecting the TTL values to actively aim to deliver packets that are placed into this queue as the likelihood of packets in this queue being delivered on time is probabilistically higher than the other two queues.

In the Figure 1, the categorization scheme described above is demonstrated graphically.

B. Probability Calculation

When a packet is received for transmission on an interface, the PBS mechanism parses the packet's DCCP and IP header to obtain three variables. The first two variables that are extracted are derived from the additional lifetime fields, namely the BT and the MTTL. In addition to this, the IP header is also accessed to obtain the packet's intended destination network. Having obtained these three variables, the probability

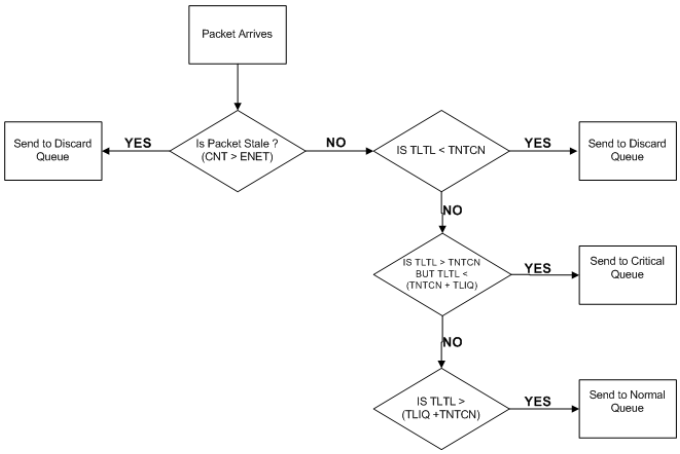

Fig. 1. Flow Diagram Representing Queue Categorization Scheme

of the packet arriving in time is calculated by performing a TNTCN lookup task. This lookup task will provide a value, in milliseconds, of the estimated time that it will take for the packet to traverse the network and reach its destination network. In the next section, the way EIGRP metrics are used to calculate the TNTCN value is presented.

1) Using EIGRP to calculate TNTCN: In order to determine the delay between the BOD and the destination network there are three elements that make up the TNTCN latency value. The first of these elements is commonly referred to as *wire line latency*. This value is the time it takes for the data signal to travel across wired or wireless medium between intermediate and end point devices including the serialization of packets into the necessary electrical signal format. This latency is governed by principals of physics and is almost always deterministic. Typically, compared to the queuing delay latency that will be discussed later, this value is negligible.

The second element that contributes to the TNTCN value will be referred herein to as *switching latency*. This value represents the minimum latency that is added by the intermediate device if no queuing takes place. This latency is created by the internal activities that occur on intermediate devices such as switches and routers with the exception of any queuing related activities. Examples of these activities include transferring packets between interfaces, packet encapsulation, MAC table and routing table lookup activities to name but a few. These values are also all typically deterministic and will normally remain constant throughout a flow. This value is the minimum latency that the intermediate device will add to the TNTCN.

The third component that contributes to the TNTCN value is *queuing latency*. Although queuing takes place on intermediate devices, this value will be treated separately from the switching latency variable described above. The reason for this is that incorporation of queues into the network path introduces a non-deterministic element into the delay calculation. Typically, queue sizes will fluctuate and cause variance in queuing latency values.

 ISBN: 978-1-61208-186-1

In order to provide an accurate TNTCN value to the PBS mechanism, all three of these values must be taken into consideration. The TNTCN value is therefore calculated using the following formula: $TNTCN = SwitchingLatency + WireLatency + QueuingLatency$

Having defined the three components that make up the TNTCN variable, the way in which the EIGRP metrics are used to calculate these elements will now be described. To estimate the time that will be required for the packet to travel across the remaining portion of the network, four of EIGRPs metrics are used by the PBS mechanism. These four metrics are EIGRP Delay, Load, Bandwidth and Hop count.

2) Use of EIGRP delay metrics: The EIGRP delay is the total delay that exists between the router and the final destination network. To calculate this value, each EIGRP device along the path assigns a predefined delay metric to all its interfaces based on the interfaces speed. Each of these calculated values along the network path is then combined in order to determine the EIGRP delay metric for the destination network. EIGRP assigns the delay value to the interface using a predefined table of values. For example, EIGRP will assign a 100Mbps Ethernet link a delay value of 100microseconds. A 1000Mbps link will be assigned a 10microsecond value and so on. The values used equate roughly to the propagation time that would be required to transmit a 1250byte packet across a network.

The issue with using CISCO Systems Inc.'s predefined delay values is that when smaller packets are transmitted across the network, the predefined delay values used by EIGRP are not indicative of the actual latency smaller packets would experience. As DCCP will be used for real-time data, which will typically use smaller more frequent data packets, the standard EIGRP delay values are not by default suitable.

In order to make the reported EIGRP Delay value more suitable to smaller packets, the delay value will be adjusted in this mechanism according to packets packet size, in order to provide more accurate and representative delay values. To do this the following formula is used.

$$Delay = \frac{PS}{1250} * RD \tag{2}$$

By factoring the reported EIGRP delay value (RD) by the actual packet size (PS), as shown in the formula above, realistic wire delay latency values are obtained.

3) Use of EIGRP Load Metric: Using the delay metric alone constitutes only the wire latency portion of the total expected latency that will occur between the BOD and the destination network. As mentioned earlier, the incorporation of queues in the network path introduces an indeterminable amount of latency due to queue size fluctuations. The inability to predict how long packets will be in queues for during transfer across the network makes the task of predicting their probability of arriving within their useful lifespan problematic. One approach that could be utilized to make queuing latency deterministic is to adopt a worst case scenario view and assume queues are always full and apply the maximum latency the queue can produce into the probability equation. While this could work, this approach was not deemed feasible because it would require that all devices have knowledge of upstream device queue sizes. If this approach was taken, it would be more beneficial to simply create a separate communication channel between devices to exchange queue depth information.

In order to overcome this non deterministic element in a more eloquent manner than described above, this mechanism makes use of the EIGRP load metric. The EIGRP load metric is calculated dynamically by EIGRP and represents the level of saturation that exists on links along the network path to the destination network. The load value is reported as a value between 1 and 255 with 1/255 representing a completely unsaturated network and 255/255 signifying a completely saturated network. In order to calculate the load value, EIGRP uses a moving average of saturation levels over a 5 minute period with sampling occurring every 5 seconds. This large sampling period prevents sudden saturation increases in network load from causing instability in this mechanism. While the frequent 5 second sampling rate ensures that there is accurate reflection of network conditions in the load value.

EIGRPs load metric is ultimately indicative of the level of congestion occurring in the network. Wherever congestion occurs, the subsequent result is virtually always an increase in queuing at various points along the network path. The larger the queue sizes become along the network path, the longer packets take to reach their intended destination. As queue sizes and link saturation values begin to increase, so too does the EIGRP load value. The PBS mechanism uses a combination of the EIGRP bandwidth metric, and the EIGRP load metric to estimate the queue time that a packet can expect to take in order to pass through the queues on the forward path to the destination network.

To calculate the queuing latency (QL), the following formula is used.

$$QL = \frac{LD}{255} * \frac{\frac{DQS*PKS}{EB}}{1} \tag{3}$$

From Formula 3 above, it can be seen that the first step taken is to obtain the default queue size of the slowest upstream device (DQS). On CISCO Systems Inc.'s devices, the default outgoing queue size is set at 40 packets. The next step taken is to calculate the size in bits that could reside in the DQS queue. This is problematic as the packets sizes can vary and therefore the exact value is not known. Assuming all the packets are of the MTU size would increase the queuing delay value inaccurately. For this reason, a bias that will result in smaller than potentially possible TNTCN values is implemented. The size of the current packet (PKS), in bits, is used in place of the MTU. If the packet is 180 bytes, the PBS mechanism will assume all the packets at the bottleneck are 180 bytes in size. In addition to this, the queue size on the local BOD is used instead of what the actual queue size may be on the device with the lowest bandwidth (slowest upstream device). There is strong support showing it would be beneficial for EIGRP to be modified to transmit queue size information of the devices in the topology.

 ISBN: 978-1-61208-186-1

Once the queue size in bits is determined, the PBS mechanism then takes the slowest link in the network path, obtained from EIGRP bandwidth metric (EB), and calculates the maximum time it will take a queue to be emptied based on the interface speed. Finally, this time is weighted against the reported EIGRP load value (LD) to provide a realistic queue delay value. In the experiment section of this paper, the accuracy of the above formula will be examined, and it will be shown that this formula and the weighting used produces a sufficiently accurate estimate of the actual delay that occurs as a result of queuing.

4) Use of EIGRP Hop Count Metric: The final element of latency that must be taken into consideration is switching latency. This latency is added through the processes that take place on intermediate devices along the network path such as routing and re-encapsulation. To calculate the switching latency value, the EIGRP hop count metric is used by the PBS mechanism. The value produced by this metric represents the number of layer 3 routing devices that exist between the BOD and the destination network. The difficulty here is determining what latency each hop or layer 3 device will add to the total latency as there can be substantial differences in layer 3 router performance. This value can unfortunately never be exactly deduced due to variations in router hardware and configurations.

In order to provide a future proof mechanism that allows the switching time to be accurately calculated when faster switching technologies eventuate, it is recommended that each BOD should perform a calculation of its own switching time during initialization. Once this is done, this value is multiplied by the EIGRP hop count value to determine the total switching latency. In practice, this value is minuscule compared to queuing latency and therefore completely accurate determination of this value is deemed unnecessary.

5) Final TNTCN Formula: Having described where the various elements that make up the total latency value are sourced from using the EIGRP metrics, the final equation used to calculate the TNTCN variable is shown in two variations below:

$$TNTCN = (WireLatency + QueuingLatency + SwitchingLatency)$$

OR

$$TNTCN = \frac{PS}{1250} * RD + \frac{LD}{255} * \frac{\frac{DQS*PKS}{EB}}{1} + HC * 8.2M \quad (4)$$

It is important to remember that these metrics are used to merely estimate the time a packet will need to reach its destination network. As network conditions are constantly changing, there is no way this scheme will provide perfect results. This scheme takes a moderate to best case scenario approach in determining the TNTCN. This means the TNTCN value is slightly ambitious because doing this gives the packets a higher probability of arriving in their specified lifespans. Adopting a worst case scenario approach, which provides higher estimated times, would mean that packets would have lower probabilities of being deemed to arrive in time and as a result a higher number of packets would be placed in the incorrect queue if network condition were not exactly as the calculation predicted. Being too lenient and providing too much leeway in the calculation, would cause reduce the PBS mechanism's effectiveness. The experimentation below will show that the weighting and selection of the various metrics described above provide a TNTCN value that is sufficiently accurate for use in the PBS mechanism.

6) Packets destined for unknown networks: If a packet is received and there is no information relating to the packet's intended destination in the EIGRP routing table, then the packet is placed into the normal queue and no priority is given to it. This ensures that all packets are serviced, even when they are destined for unknown networks. In addition, applications can append a 99999 millisecond lifespan to packets they do not wish to have categorized. Upon receiving such a packet, the BOD will automatically place the packet into the normal queue.

C. Scheduling implementation of BOD

For the purposes of implementing the mechanism, statistical time division multiplexing was selected as the method for scheduling packet delivery on outgoing interfaces. With statistical multiplexing, each of the queues essentially became a channel in the scheduling algorithm. Each channel was then assigned the predetermined amount of bandwidth and serviced at a fixed rate accordingly. The advantage of statistical multiplexing over traditional time division multiplexing is that where a particular channel does not need to transmit during its pre-allocated slot, this slot can then be used by the next channel waiting to transmit. In traditional time division multiplexing this does not occur and the slot remains unused rather than being allocated to the next channel. For the purposes of the experimentation, the Opnet Modeler simulation toolkit provided statistical time division multiplexing modules as well as the queuing modules that were added and configured to service the queues on outgoing interfaces on the BOD device. For the purposes of this scheme, each of the three queues were configured as an individual channel and serviced based on a strict bandwidth allocation percentage.

VI. Experimentation

In this section, two experiments that were conducted to examine and test the scheme described above are presented. The primary purpose of these experiments is to test the accuracy of the TNTCN formula and ensure the action of placing packets in the critical queue is of benefit.

A. Experiment 1: Accuracy of TNTCN Formula

In this experiment, the accuracy of the formula used to derive the TNTCN value was explored. This experimentation compares the computed TNTCN value to the actual time it took for the packets to reach their destination networks. This comparison determines if the weighting used in the formula is

 ISBN: 978-1-61208-186-1

suitable for the purposes of the PBS mechanism by ensuring the TNTCN and actual values are within acceptable range. In addition to this, where there was variance in the times, the TNTCN formula should provide a bias towards lower TNTCN values than the actual time it took for packets to cross the network. If the TNTCN formula produces excessively large TNTCN values then packets could inadvertently be placed into the critical queue, or even worse be deemed unable to reach the destination within the remaining lifespan and placed in the discard queue. By maintaining a bias towards smaller TNTCN values, the PBS mechanism will produce fewer incorrectly categorized packets.

1) Hypotheses: It was hypothesized that the TNTCN and actual times needed to cross the network would typically be within a 10% threshold of difference during normal network operation periods. In addition to this, it was also hypothesized that the TNTCN formula would provide a bias towards generating smaller TNTCN values than the actual time that is needed by packets to cross the network.

2) Experiment Methodology: In order to carry out this experimentation 20 unique simulated topologies were created. These topologies were modelled based on a medium sized wide area network (WAN). In each of these topologies between 10 and 30 CISCO Systems Inc. 2600 series routers were configured. A variety of LAN and WAN links were used to connect the devices with varying speeds. These speeds ranged from 128kbps to 44736kbps in order to add complexity to the topology. In addition to this, each of the experimental topologies were configured to initiate multiple simultaneous flows across the network throughout the simulated period. As a result, at any given time 10, 20, 50 or 100 simultaneous flows existed on the network. This complexity was necessary due to the inclusion of EIGRP into the mechanism and the need to have a network complex enough to provide large routing tables and traffic loads that were capable of creating congestion events. In order to gather the data needed for this analysis, five randomly selected flows were chosen from each simulation (where more than five flows existed). The experiments were run over a simulated period of five minutes. The statistical results data from the experiment was only collected during the final 30 seconds of the five minute simulated period. This was done to allow sufficient time for the routing protocol to converge and network load or congestion levels to reach stabilized levels. Using global variables in the Opnet Modeler simulator, each time a packet left the BOD, a timestamp entry was created in a predefined global variable array. In addition to this, the TNTCN value calculated by this scheme was also stored in the same global variable array. Once the packet arrived at the destination network, its sequence number was used to access the corresponding global variable and the current NTP time was entered into the variable store.

The last routing devices along the network path were configured to add the simulation_time_pkt_arrive value to the global variable array when the packet arrived at the incoming interface on the router. The reason for configuring the last router to perform this task and not the destination workstation, was because the EIGRP metric is based on the premise that it calculates the time for the packet to reach the destination network only and not the actual endpoint device. Although in practice the difference is likely to be negligible, in the interests of rigour, it must be noted the actual_time variable was recorded when the packet arrived at the destination network and not the final node. At the end of the simulation period the group of all global variable arrays were collected. From these variables the actual time the packets took to traverse the network to the final destination network was calculated using: $simulator\_pkt\_arrive simulator\_pkt\_depart$. The actual time that was taken and the calculated time needed (TNTCN) were then compared.

In a few instances some of the packets were discarded by an upstream BOD device where the packet was deemed unable to traverse the network within time or where the packet had become stale. In order to accommodate this phenomenon in the results presented below, wherever the Simulator_time_pkt_arrive value was not recorded, that entire array relating to that particular packet was discarded.

3) Results: In the experiment 20 simulated topologies were tested and 5 flows from each of these topologies were monitored specifically. This led to a collection of 100 separate result sets. In the interests of keeping this paper concise not all 100 flows will be discussed in great detail. Instead a summarized version of the results will be presented.

In Table 1, the percentage of average variation found between the TNTCN calculation and the actual time it took packets to reach the destination network is shown. To calculate the values shown in Table 1, the percentage of difference was first calculated from each packet flow statistic array ($TNTCActualtime * (100/1)$). This value represents the percentage difference in times between that occurred between the calculated TNTCN value and the actual time that was measured. Once this had been done for all the flows, the average of these values was then calculated. To do this all the percentage of differences values calculated in the previous step were summed and then averaged ($Sum(percentageofdifference)/numberofcalculations$). This value represents the average difference in the calculated TNTCN and actual times that occurred for the flow. Note, a + symbol indicates that the average TNTCN value was larger than the actual time taken value and a symbol indicates the TNTCN value was found to be smaller on average than the actual time taken.

In topologies 14 and 17, a T3 serial link was placed into the network topology which resulted in a major issue with the PBS mechanism being discovered. The delay value automatically assigned to serial links in EIGRP is 20000 microseconds by default, irrespective of the speed of the serial link. When this value was used in the TNTCN formula, there was a large discrepancy between the calculated TNTCN value and the actual time it took for the packet to reach the destination network. Note, for the remainder of this narrative results from topologies 14 and 17 will be excluded because of this anomaly. As can be seen from the results in Table 1, in the

 ISBN: 978-1-61208-186-1

Top No.	Flow 1	Flow 2	Flow 3	Flow 4	Flow 5
1	-3.210%	-2,922%	-3.749%	-4.542%	-2.8924%
2	-3.433%	-3.879%	-3.773%	-4.185%	-3.860%
3	-4.371%	-3.823%	-4.087%	-4.592%	-3.376%
4	-3.483%	-3.653%	-3.833%	-4.042%	-3.325%
5	-6.084%	-6.339%	-6.545%	-6.423%	-6.284%
6	-1.218%	-0.924%	-1.188%	-1.255%	-0.827%
7	-0.313%	-0.267%	-0.231%	+0,109%	-0.119%
8	-4.520%	-5.643%	-5.012%	-5.221%	-4.845%
9	-6.767%	-6.289%	-6.133%	-6.934%	-6.459%
10	-6.001%	-5.847%	-6.210%	-6.387%	-6.113%
11	-2.329%	-2.367%	-2.753%	-2.958%	-2.164%
12	-3.562%	-3.324%	-3.491%	-3.762%	-3.445%
13	-3.011%	-2.872%	-2.992%	-3.019%	-2.691%
14	+17.601%	+17.938%	+17.614%	+18.478%	+17.737%
15	-7.968%	-7.758%	-7387%	-8.544%	-7.891%
16	-8.021%	-8.762%	-8.539%	-8.893%	-8.142%
17	+12.116%	+12.529%	+12.418%	+12.933%	+12.674%
18	-2.985%	-2.754%	-2.531%	-2.997%	-2.655%
19	-3.749%	-3.564%	-3.285%	-3.857%	-3.651%
20	-0.871%	-0.654%	-0.211%	+0.097%	-0.054%

TABLE I
TNTCN ACCURACY - VARIATION OF TNTCN AND ACTUAL TIMES RECORDED (CORRECT TO 3 DECIMAL PLACES)

vast majority of flows the calculated TNTCN produced, on average, a TNTCN value that was within the 10% threshold that was hypothesized. In addition, in all flows the calculated TNTCN value had a distinct bias towards generating lower TNTCN values than the actual transmit times that occurred. Delving deeper into the results, there was clear evidence that in more complex topologies such as in topologies 7 and 20, the mechanism did at times begin to lose its bias towards generating smaller TNTCN values. Exploring the results from these topologies, it became apparent that where EIGRP load metric value increased above 200/255, the PBS mechanism began generating larger volumes of TNTCN values that were higher than the actual time it took for packets to cross the network. Where the EIGRP load value was below 200/255, the TNTCN formula had a clear bias towards generating smaller TNTCN values than the time packets took to cross the network, and only very rarely generated higher values. However, once the EIGRP load value exceeded 200 the TNTCN values showed no clear bias towards being lower than the actual time it was taking for packets to arrive at the destination network.

The next grouping of statistics took the results one step further to determine the percentage of times the packets calculated TNTC values were higher than the actual time taken for the packet to arrive at the destination network. This value shows the percentage of packets in the particular flow that are given higher than necessary TNTCN values. The danger with having a higher TNTCN value is that packets may be categorized into the critical or discard queue incorrectly because of this higher value. The reality for these packets is that they have arrived in a faster time than the TNTCN value had predicted. Using the same statistics collated in the first part of the experiment, all instances where the TNTCN value was larger than the actual time that was taken for the packet to reach the destination network were identified. The number of instances where this was found to be the case was then represented as a percentage of the total number of packets that were send during the collection period (no. of young packets / total number of packets).These results are shown in the Table 2.

Top No.	Flow 1	Flow 2	Flow 3	Flow 4	Flow 5
1	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
2	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
3	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
4	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
5	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
6	0.001013%	0.002112%	0.00129%	0.00135%	0.00285%
7	0.005282%	0.006001%	0.00609%	0.00724%	0.00585%
8	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
9	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
10	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
11	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
12	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
13	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
15	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
16	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
18	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
19	<0.001%	<0.001%	<0.001%	<0.001%	<0.001%
20	0.00341%	0.00244%	0.00407%	0.00553%	0.00285%

TABLE II
TNTCN ACCURACY - PERCENTAGE OF PACKETS WITH EXCESSIVELY HIGH TNTCN VALUES(CORRECT TO 3 DECIMAL PLACES)

In the all but three topologies, namely topology 6, 7 and 20, the percentage of packets that were given a higher TNTCN value than the actual time taken were below 0.001% of the total number of packets sent. In topology 6, the TNTCN formula calculated a higher TNTCN value than that was actually experienced for approximately 0.002% of the packets. In topology 7 and 20 this value was approximately 0.006% and 0.004% respectively. As can be seen in Table 2, the number of times the TNTCN formula generated higher values than those that were recorded in the simulation were extremely low.

4) Analysis and discussion or results: The results show that the TNTCN formula used in the experimentation, in the majority of cases ($> 90\%$), was on average, able to provide values that fell within the specified threshold sought by the PBS mechanism. In addition to this, the PBS mechanism also provided a bias towards generating lower TNTCN values than were actually experienced in all but 8 of the 100 streams tested. This suggests there is strong evidence to support the hypothesis, and that the given formula provides suitable values needed in the PBS mechanism. The formula used to calculate the TNTCN value could potentially be improved in order to provide more accurate TNTCN values. One way this could be achieved would be to incorporate an additional 6th metric into EIGRP, whereby queue depths are advertised and exchanged among EIGRP devices. By doing this, the need to try to estimate queuing times would be eliminated from the formula

 ISBN: 978-1-61208-186-1

calculation which would make this mechanism more accurate. While the TNTCN value could potentially be improved, the results categorically show that for the purposes of the PBS mechanism and this proof of concept experimentation, the current formula is adequate. The final issue discovered through this experimentation is that EIGRP assigns a delay value of 20000 microseconds to all serial links in the topology when calculating the delay metric. Using this value creates a very high TNTCN value and would lead to a large number of incorrectly categorized packets. While this is problematic, the reality is such that if high speed serial links are employed on networks, typically the delay value in EIGRP is configured manually in order to ensure EIGRP is given an accurate representation of actual network conditions. In such a case this PBS mechanism would simply make use of the manually configured delay value and this issue would be resolved.

B. Experiment 2 Effectiveness of the critical queue

The second experiment in this paper tested the effectiveness of the critical queue by exploring how many of the packets sent via the critical queue were received within their useful lifespan. The reasons for doing so are to ensure the function of prioritizing these packets performs its intended duty of ensuring packets reach their destination networks within the necessary time frame. If this is not achieved and packets that are placed into the critical queue arrive at the destination stale on a consistent basis, it will bring into question the effectiveness and benefits of the scheme.

1) Hypotheses: It is hypothesized that the majority of packets placed into the critical queue will arrive at their destination within their valid lifespan window.

2) Methodology: To carry out this analysis, the three topologies where the highest number of packets that were placed into the critical queue out of the 20 topologies used in the previous experiment were selected. The BOD was then modified in these three topologies to mark all of the packets that were placed into the critical queue using the COS field in the IP header for that packet. All packets which were placed into the critical queue were given a COS value of 8. The workstations in the topology were then configured to detect the COS marking when incoming packets were received. When a packet was received with a COS value of 8 in the IP header, the workstation then calculated if the packet was fresh or stale. This was done by checking the current NTP time with the lifetime information in the DCCP header for that packet. If the packet was found to be fresh a $global_fresh_pkt_rcvd_counter$ variable was incremented by one. Alternatively, if the packet was found to be stale a global $stale_pkt_rcvd_counter$ variable was incremented by one. In addition to these two variables, a third variable was also used which recorded the total number of packets placed into the critical queues on the BOD. This variable was called $critical_pkt_sent_counter$. This variable was used to ensure all packets placed in the critical queue were received in either a stale state or a fresh state. Each time a packet was placed into the critical queue, the $critical_pkt_sent_counter$ value was incremented by one. To simplify the analysis of results, only one BOD was used in the topologies in which this experimentation was performed.

Topology	Total Pkts Received Fresh	Total Pkts Received Stale	Total Pkts Reported by BOD	Percentage of Pkts Arrived fresh
14	6080	830	6910	87.988%
15	9970	1180	11150	89.417%
16	15330	1850	17180	89.232%

TABLE III
PERCENTAGE OF PACKETS FROM CRITICAL QUEUE RECEIVED ON TIME

3) Results: In Table 3, it can be seen that in all three topologies the number of critical packets that were received fresh was significantly larger than the number of critical packets that were received stale. Specifically, between approximately 87 and 89 percent of the packets transmitted from the critical queue were able to reach the final endpoint within their useful lifespan.

4) Analysis and discussion of results: While the results shown here indicate the PBS mechanism allows the majority of packets placed in the critical queue to be received within their useful lifespans, it must be noted that the network conditions are conducive to this occurring. It is simply not possible to compare the results that occur when the PBS mechanism is enabled to results taken from a standard DCCP topology where categorization does not take place. This is because the act of prioritizing packets has a compounding effect on all other subsequent packets in that flow. Direct comparison between a standard topology and a topology where the PBS mechanism is enabled would not be accurate because of the infinite variation that would occur, particularly due to changes that occur in congestion window rate values.

C. Limitations of the experimentation

One of the limitations of this experimentation is that the performance gains suggested in the results section of the experimentation are very much based on topology selected. During the simulation analysis phase, the results showed categorically that the level of improvement, as well any adverse effects this scheme caused, became almost entirely dependent on the network topology. While all efforts were made to generalize the result findings to make them applicable to a large a range of topologies, the results still should be treated as proof of concept rather than exactly what could be expected in any given topology.

An additional limitation with the experiments is that they do not take into consideration the impact EIGRP protocol has on the overall performance of the various streams. While EIGRP is designed to be as minimalistic as possible, there are overheads associated with its function that should be considered. This impact is not measured in any of the results above as the complexity of the multiple streams and contention between the streams was not explored on a per stream basis. While there would be some impact caused through the addition of EIGRP needed by the scheme, this impact is likely to be extremely minimal. The final limitation that will be discussed

 ISBN: 978-1-61208-186-1

in regards to the experiments was that the workstations were designed in such a way that they generated packets as fast as the DCCP protocol would allow them to transmit. This is not what would be expected in normal real world network operations as the application layer would likely limit the rate of transfer, especially in real-time applications. In theory, the volume of traffic generated by the 5 to 20 workstations would be representative of a much larger number of real world flows than depicted in the experimental topology.

D. Experimentation conclusion

The experiments above serve two main purposes. The first purpose of the various experiments was to validate that the PBS mechanism worked effectively, reliability and in a stable manner in a number of different topologies. The second main purpose was to identify scenarios and network conditions where the PBS mechanism would be most beneficial in improving DCCP performance. The results from the various experiments showed that both of these occurred and there is potential benefit that can be obtained through the implementation of the mechanism. In terms of reliability, the networks tested remained stable and throughout the simulations all links and nodes remained up. In these instances, the BOD devices remained stable and were able to service up to 20 simultaneous packet streams transferring data at rates governed only by the CCID3 protocol. Having stated these findings, it is important to note that this research did not explore topologies where changes to the topology structure occurred due to device or link failure during the statistical collection. Finally these experiments show that the selection of the probability based TNTCN value falls within the acceptable threshold and produced improvement to overall network performance. Now that the reliability and benefits of the mechanism have been shown, it is hoped a more advanced and accurate formula can be developed in future research. This would allow the 10% acceptable threshold to be greatly reduced. This will also mean greater accuracy can occur in the categorization process of packets and thus improve the scheme.

VII. Conclusion

This paper has introduced a novel scheduling mechanism for the DCCP protocol. Specifically, the research has introduced a PBS mechanism that utilizes an array of routing protocol information to predict the time that it is likely to take for packets to reach their destination networks and pro-actively sorts and prioritizes packets based on this prediction. By placing packets into one of three queues, packets that are likely to become stale or that have already become so, are pruned or given a lower priority to ensure they do not have an adverse effect on fresh packets utilizing the same contended resources. Packets that need to be prioritized in order to avoid becoming stale are given the best chance of being delivered on time. Finally, the mechanism provides a deterministic queue that ensures the majority of normal packets remain largely unaffected by the actions occurring in the other two aforementioned queues. Detailed discussion as to how the mechanism should be implemented have been presented which was followed by two experiments that were carried out to examine the accuracy of the PBS mechanism. These experiments showed that the proposed mechanism is stable, reliable and capable of offering benefit to CCID3 controlled DCCP. This research concludes that through the results obtained in the proof of concept implementation of the mechanism, not only is the mechanism workable, it is also provides highly accurate TNTCN predictions. In addition, the results also show that the mechanism is able to ensure delivery of packets placed into the critical queue which would likely have become stale had the intervention of the mechanism not taken place. While there is still some work required to optimize the mechanisms efficiency, the objectives of this paper to showcase this novel probability based packet optimization mechanism through proof of concept implementation and modelling have been achieved.

References

[1] E. Kohler, M. Handley, S. Floyd, and J. Padhye. "RFC 4340: Datagram congestion control protocol (DCCP)." Technical report, IETF, Request For Comments, 2006.

[2] D. Farinachi, Introduction to enhanced IGRP (EIGRP). Cisco Systems Inc Press, 1993.

[3] A. Gurtov and R. Ludwig, "Lifetime packet discard for efficient real-time transport over cellular links." *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(4) pp. 32-45. IEEE, 2003.

[4] C.W. Yuen and O.C. Yue, "Channel state dependent packet discard policy for 3g networks." *In Vehicular Technology Conference, 2006. VTC 2006-Spring.* IEEE 63rd, volume 1, pp. 405-409, 2006.

[5] K. Chebrolu and R.R. Rao, "Selective frame discard for interactive video." *In Communications, 2004 IEEE International Conference on*, volume 7, pp. 4097-4102. IEEE, 2004.

[6] S. Liang and D. Cheriton, "Tcp-rtm: Using tcp for real time multimedia applications" *In International Conference on Network Protocols*, 2002

[7] I. Radovanovic, R. Verhoeven, and J. Lukkien, "Improving tcp/ip performance over last-hop wireless networks for streaming video delivery." In *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers*. International Conference on, pp. 1-2. IEEE, 2007.

[8] R. Albrightson, JJ Garcia-Luna-Aceves, and J. Boyle, "Eigrp-a fast routing protocol based on distance vectors." In *Proc. Networld/Interop*, volume 94, pp. 136-147, 1994.

[9] Low Latency Queueing. 2001 [cited 2010 05/07/2011]; *Available from*: $http : //www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fsllq26.html$.

[10] S. Floyd, E. Kohler, and J. Padhye, "RFC 4342: Profile for datagram congestion control protocol (DCCP) congestion control id 3: Tcp-friendly rate control (TFRC)". Technical report, IETF, Request For Comments, 2006.

 ISBN: 978-1-61208-186-1