

Evaluation of the Impact of Packet Drops due to AQM over Capacity Limited Paths

Eduard Grigorescu, Chamil Kulatunga, Gorry Fairhurst

Electronics Research Group, School of Engineering, University of Aberdeen, Aberdeen, AB24 3UE, United Kingdom
{eduard, chamil, gorry}@erg.abdn.ac.uk

Abstract - For many years Internet routers have been designed and benchmarked in ways that encourage the use of large buffers. When these buffers accumulate a large standing queue, this can lead to high network path latency. Two AQM algorithms: PIE and CoDel, have been recently proposed to reduce buffer latency by avoiding the drawbacks of previous AQM algorithms like RED. This paper explores the performance of these new algorithms in simulated rural broadband networks where capacity is limited. We compared the new algorithms using Adaptive RED as a reference. We observe that to achieve a small queuing delay PIE and CoDel both increase packet loss. We therefore explored this impact on the quality of experience for loss-sensitive unreliable multimedia applications, such as real-time and near-real-time video. The results from simulations show that PIE performs better than CoDel in terms of packet loss rates affecting video quality. We also noted that the performance of ARED is comparable to that of PIE and CoDel in constant capacity links. This suggests that AQM in general is useful for limited capacity network paths.

Keywords - Active Queue Management, Adaptive RED, PIE, CoDel, Real-time Voice, VoIP, Rural Broadband, QoE

I. INTRODUCTION

The cost and technical challenges of service provision for access networks mean that capacity is often limited, and nearly always shared, resulting in a bottleneck in the access part of the network path. The default queuing algorithm in most routers is FIFO queuing with taildrop. An increase traffic will lead to queuing in the bottleneck router buffers, resulting in congestion, and ultimately in packet loss. We suggest the choice of taildrop behaviour is often motivated by a desire to achieve a high forwarding rate, low packet losses, and high TCP bulk throughput.

TCP probes for capacity and only lowers its sending rate when it detects dropped or marked packets, interpreted as a signal of congestion. Therefore a bulk TCP flow increases its transmission rate until it fills router buffers. To optimise throughput, and hence minimise loss, operators typically configure large packet buffers in their routers. This trend has been encouraged by router vendors following the decreasing cost of memory. However, there is growing evidence that oversized buffers have resulted in large queues and a phenomenon that has come to be known as “bufferbloat” [1].

Bulk TCP flows can lead to a high queuing delay (latency) with a large router buffer. This disrupts the proper functioning of TCP [2], and may not improve the TCP throughput. The collateral effect on other flows that share a common router queue can be significant, degrading the experience of

interactive applications, and impairing the quality of real-time (e.g. voice) and near-real-time traffic (e.g. live streaming).

The solution to high latency requires two elements: 1) a strategy to signal the transport protocol to “slow down”, necessary to keep queues small; and 2) packet scheduling between flows (sharing capacity between different types of network flows). These solutions become more necessary at the edge of the network, where speed of links is less and therefore head of line blocking becomes more significant.

One way to prevent or minimise latency is to intelligently manage the queue of packets so that a router does not build a large standing queue. Active Queue Management (AQM) techniques have been developed over the last two decades to work in this way to improve capacity sharing in routers.

AQM algorithms, such as Random Early Detection (RED) [2] detect congestion by monitoring the router queue. If the queue exceeds an initial threshold (th_{min}), RED drops (or marks) some of the arriving packets with a probability function (p) based on the *average* router queue size. The dropped /marked packet effectively sends a congestion signal to the transport endpoints, encouraging them to reduce their sending rate before a large queue builds, and also preventing extensive packet loss. RED and its variants, such as Random Exponential Marking (REM) [3], can outperform DropTail schemes, but need careful configuration of parameters to reflect the network scenario, especially the target queue size.

Adaptive RED (ARED) [4] was proposed to overcome the difficulties of configuring network-dependent parameters in RED. It keeps the queue at an average length of two times a configured th_{min} , irrespective of the buffer size and the level of congestion. th_{min} is selected based on the user-defined target delay (D_{ARED}) and the link capacity (C), which is assumed constant.

In 2012, a new AQM scheme, Controlled Delay (CoDel) [5] was presented to address bufferbloat. CoDel directly controls the router/host queuing delay. CoDel timestamps packets as they are enqueued. If the time spent by a packet within the queue is higher than a defined threshold (D_{CoDel}), the algorithm sets a timer to drop a packet at dequeue. This dropping is only done when the queuing delay within a time window is above D_{CoDel} and the queue holds at least one MTU’s worth of bytes. This time window will be consequently reduced if the queuing delay continues to stay above D_{CoDel} , or otherwise will be cleared.

Recently, an alternate AQM scheme, Proportional Integral controller Enhanced (PIE) [6] was also proposed. Like RED, PIE uses a probability to drop packets at enqueue, based on the estimated level of congestion. The congestion is used to increase or decrease p , based on the deviation of the current queuing delay from a target delay (D_{PIE}). Queuing delay is calculated from the queue length and the packet departure rate. The PIE algorithm also considers the increasing and decreasing latency trend when calculating the probability.

The remainder of this paper is structured as follows. Section I provided an introduction to AQM algorithms. Section II analyses the characteristics of AQM algorithms and the impact of high packet loss under rural broadband networks, with simulation results presented in Section III. Section IV provides a discussion and Section V the conclusion.

II. DELAY AND PACKET LOSS WITH AQM

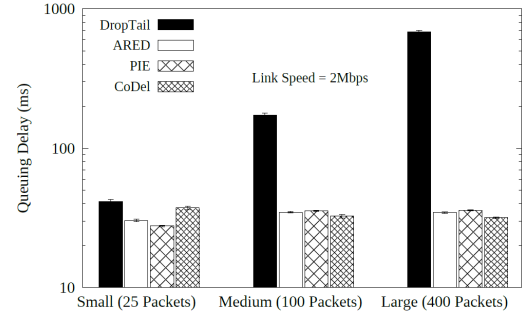
Congestion is generally not experienced in the Internet backbone, which is often over-provisioned. However, over provisioning cannot be expected in all the networks [8]. Many broadband access networks are therefore capacity-limited, especially when used for rural broadband services [11]. For example, AQM support in home gateways is expected to be beneficial [10]. Results from Akamai, suggest that despite increasing broadband speeds in urban areas, global average connection speeds remain below 3Mbps [9]. The gap between the global average and the peak has grown from 4Mbps to 13Mbps in 3 years. This growing divide indicates that recommendations to use AQM algorithms in the general Internet need to consider performance with a diversity of rates, and a range of applications.

The traditional performance metric of bulk applications e.g., file transfer or e-mail, measures the throughput. The TCP transport is impacted by loss/marketing, and insensitive to jitter and only slightly influenced by latency - since generally large windows are available. In contrast, other applications have very different requirements for latency, jitter, throughput and loss. Interactive applications: voice, video conferencing, some online gaming etc., often require low latency. Other applications can also benefit from reduced latency (e.g. live streaming). In general such applications use unreliable transport protocols (e. g. UDP/RTP, DCCP, RMCAT). Non-interactive streamed multimedia, can often accept more latency and jitter by buffering the content and can use reliable TCP protocol.

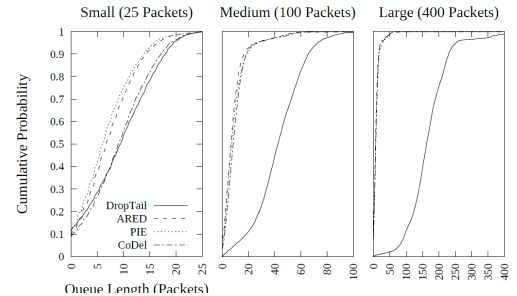
This paper compares ARED, PIE and CoDel as examples of AQM methods that seek to reduce the queuing delay irrespective of total buffer size. A side-effect of these methods is that they can increase loss and can reduce link utilization. One issue is that a router buffer may be shared both by delay-sensitive applications, and by applications that are sensitive to loss. This paper therefore evaluates the impact of high loss on unreliable video traffic that shares an ARED, PIE or CoDel router queue with latency sensitive applications (VoIP).

The topology in Fig. 2 was simulated with DropTail, ARED, PIE and CoDel at the bottleneck queue with a link

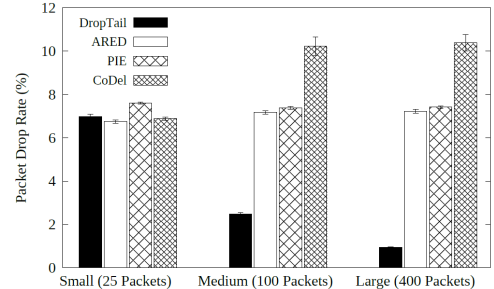
capacity of 2Mbps. It uses a simulated traffic mix of long-lived TCP flows, web TCP traffic, UDP video streaming and VoIP.



(a) Queuing delays for various AQM algorithms



(b) Cumulative probability distribution of the queue length



(c) Packet drop rates

Fig. 1. AQM performance for three buffer lengths at a bottleneck

The three graphs in Fig. 1 illustrate the properties of the queuing algorithms for three buffer sizes with 95 percent confidence intervals (over 50 seed values). Target delays for AQM algorithms were selected to achieve the same queuing delay for the traffic mix with a buffer size 100 packets (described in Part A of section III). According to Fig. 1 (a), a large buffer using a DropTail queue can result in high latency (as high as 680ms with a 400 packet buffer). ARED, PIE and CoDel maintain a small latency, irrespective of the buffer size, by setting a queue delay limit (Fig. 1 (b)). Fig. 1 (c) shows that ARED, PIE and CoDel all increase the packet drop rate at the bottleneck queue more than when using DropTail.

The published simulation results for PIE and CoDel considered TCP and UDP traffic, but we are not aware of results that monitor UDP considering the Quality of Experience (QoE) for multimedia applications. Current results have analysed the latency and throughput without elaborating the impact of packet loss. PIE has compared its performance with original RED [2], but CoDel evaluations used ARED [4] for comparison.

In [18], a simulation study compared CoDel, PIE, SFQ-CoDel and DropTail queues in DOCSIS 3.0 networks. It tested scenarios with light, moderate and heavy traffic that consisted of VoIP/gaming, web browsing, file upload and CBR traffic. In terms of VoIP and web traffic, very good performance was obtained using CoDel, PIE and SFQ-CoDel. The gaming performance results revealed that CoDel and PIE have good latency and loss performance with First Person Shooter (FPS) games that are sensitive to latency but not packet loss. However, it concludes that cloud-based games are more sensitive to loss than latency.

Some Internet traffic prediction reports [19] forecast that video streaming in the Internet could be as high as 60% of the total traffic by 2015. Many near-real-time applications (e.g. live streaming) provide good performance by using playout buffers to accommodate delay jitter introduced on the network path. AQM algorithms may significantly reduce the buffering needed, by reducing latency and jitter. The impact of loss depends on the design of video applications, including the choice of codec. This has been studied with various encodings including MPEG2 and MPEG4. Even though encoding/decoding algorithms can mitigate loss via retransmission or repair, they usually still benefit from low loss to achieve high perceptual quality [17].

Therefore it is important to investigate the impact of packet loss on different video applications. These motivated our further study of AQM algorithms with loss sensitive applications using unreliable transport protocols.

III. SIMULATION OF DROP IMPACT OF AQM

This section evaluates the impact of loss resulting from ARED, PIE and CoDel, by comparing the loss with a DropTail router queue. It analyses whether the loss rate induced by AQM has a negative impact on unreliable video applications, and whether AQM reduces the utilisation of the bottleneck link for bulk applications. Algorithms and protocols were evaluated for a scenario with limited capacity, simulating a rural broadband access path. Link capacities from 400kbps to 5Mbps were simulated with a range of path RTTs from 10ms to 300ms, representing a range of access network paths.

A set of ns2 simulations [12] used a dumbbell network topology with 8 nodes and a single bottleneck link (Fig. 2). The bottleneck capacity (C Mbps), the one-way link delay (D ms) and the queue buffer size (B packets) were varied in each experiment. All other links had excess capacity, ensuring they did not create an additional bottleneck between a sender and receiver pair. The queuing discipline at node 4 to bottleneck link 4-5 used one of the following AQM algorithms in packet mode: DropTail, RED configured for ARED, PIE and CoDel. ns2 implementations were provided by the authors of the PIE and CoDel (at their web site) for algorithm simulations.

Four long-lived FTP TCP NewReno/SACK ($IW = 3$) flows and two PACK-MIME [13] TCP web traffic loads (sent at a

rate of 8 requests/s to contribute about 80-100kbps of web traffic at the bottleneck) were added between node pairs 1-6 and 3-8. The TCP applications used a MSS of 1024 bytes.

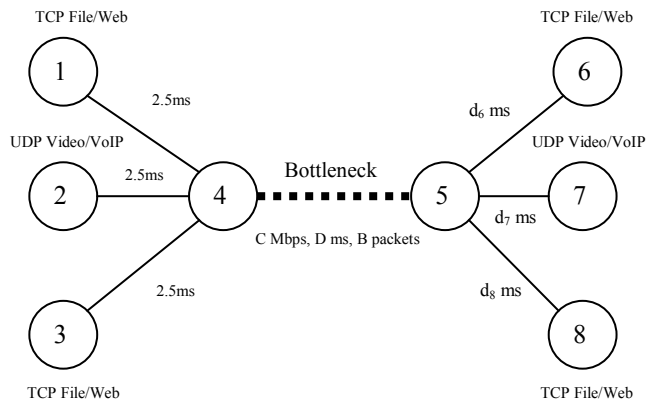


Fig. 2. Dumbbell simulation topology for different queuing disciplines

Video and voice traffic was introduced over the path between nodes 2 and 7. ns2 Evalvid [14] was used to model 450kbps of VBR video traffic (using the video sample given in TABLE I) over RTP/UDP. The Peak Signal to Noise Ratio (PSNR), which considered only loss, was calculated as an objective QoE measurement parameter for video application.

TABLE I. Evalvid video sample

Sample	PARIS (4:2:0 YUV)
Number of Frames	1065
Frame Rate	24 fps
Resolution	353 * 288
Simulation Duration	155s

Three bidirectional voice conversations (G.729A encoding) ran for the duration of the simulation between nodes 2 and 7 with traffic simulated using ns2voip++ [15]. The Mean Opinion Score (MOS), which considers both end-to-end latency and loss (R-factor converted using e-Model), provided a voice quality parameter.

A RED queue was configured using ARED (according to [4]) with minimum threshold (th_{min}) derived from the link capacity (C) and the target delay (D_{ARED}), and a maximum threshold (th_{max}) of $3th_{min}$ ($\alpha = 0.01$ and $\beta = 0.9$). Minimum and maximum dropping probabilities of ARED were 0% and 50%, using the ‘gentle’ RED algorithm. PIE was configured with a 30ms probability update interval and dequeue threshold of 10Kbyte. The CoDel update interval was 100ms. The target delays of the three AQM algorithms are described in the following section.

To compare the impact of loss on performance, the average queuing delay was measured, noting the dequeue rate and packet drop rate. The PSNR was measured for video and the MOS for voice. The buffer B was 100 packets in all the experiments. The one-way delay for all the links on the left side was 2.5ms. All simulations ran for 200s duration with results averaged over 10 seed values.

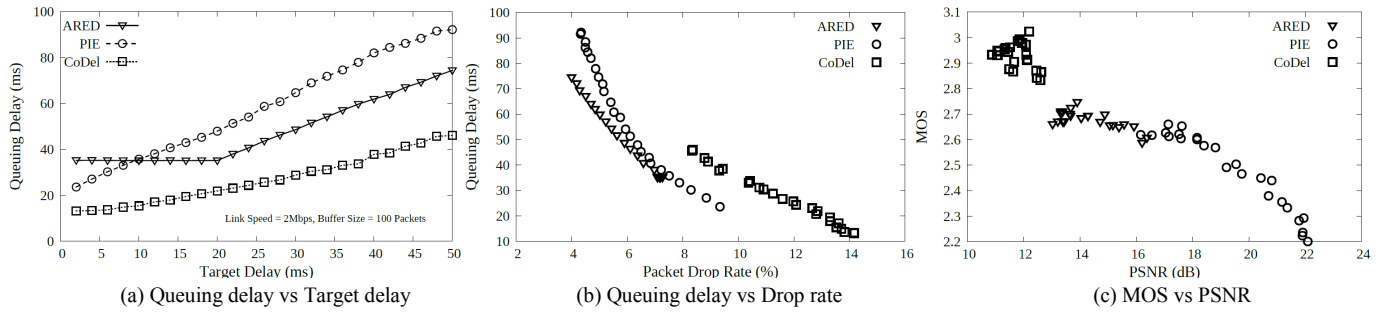


Fig. 3. Performance with different target delays (default 20ms for PIE and 5ms for ARED and CoDel)

A. Performance with different settings of Target Delay

The traffic mix was simulated, changing the target delay of ARED, PIE and CoDel from 2ms to 50ms. The bottleneck link capacity (C) and delay (D) were kept respectively at 2Mbps and 0ms. The one-way link delay parameters in Fig. 2 were $d_6=2.5\text{ms}$, $d_7=47.5\text{ms}$ and $d_8=147.5\text{ms}$. This created a path RTT of 100ms for video and voice traffic and 10ms for one half of the TCP FTP and web applications and 300ms for the other half.

According to Fig. 3 (a), ARED, PIE and CoDel achieved a queuing delay of 35ms for the simulated traffic scenario when the target delays were 10ms, 10ms and 35ms respectively. Therefore we selected these target delays ($D_{ARED}=D_{PIE}=10\text{ms}$, $D_{CoDel}=35\text{ms}$) in our experiments for a fair comparison.

Fig. 3 (b) shows the latency is increased (ARED: 35.2 to 74.4, PIE: 23.6 to 92.2, CoDel: 13.2 to 46.1) and the drop rate is reduced (ARED: 7.2 to 4.0, PIE: 9.3 to 4.3, CoDel: 14.2 to 8.4) by increasing the target delay (queuing delays on the y-axis of figures (a) and (b) map one-to-one). This resulted in a poor MOS for voice and improved PSNR for video with PIE, as shown in Fig. 3 (c). The MOS reduction in ARED and CoDel is not significant. However, ARED and CoDel failed to offer this improvement of PSNR.

B. Performance with Different path RTTs

The topology was simulated with the previously described traffic mix and target delays by changing the delay D of the bottleneck link from 0ms to 145ms. The one-way delay for the three links on the right side was 2.5ms ($d_6=d_7=d_8=2.5\text{ms}$). This resulted in varying path RTTs from 10ms to 300ms for all applications.

Fig. 4 (a) shows the bottleneck loss rate for the range of path RTTs. ARED, PIE and CoDel introduce higher loss for path RTTs lower than 100ms. CoDel increases loss significantly with small path RTTs. This occurs because small RTTs allow TCP to rapidly increase the congestion window as flows try to capture the remaining queue space; as a counter measure, these AQMs drop more packets to keep the queue small. Poor PSNR with CoDel can be seen in Fig 4 (b).

This mix of flows was simulated with different path delays for different flows sharing the bottleneck router queue. The link delay parameters in Fig. 2 were $D=0\text{ms}$, $d_6=2.5\text{ms}$ and

$d_7=47.5\text{ms}$. d_8 was changed from 2.5ms to 147.5ms. This created a varied path RTTs from 10ms to 300ms for one half of the TCP long-lived and web applications and 10ms fixed RTT for the other half.

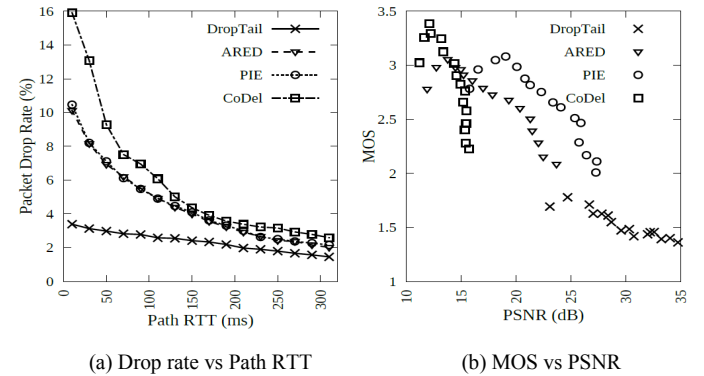


Fig. 4. Performance with different path RTTs but same for all the applications

Fig. 5 (a) shows persistent high packet loss at the router queue for ARED, PIE and CoDel, due to the low path RTT TCP applications between nodes 1 and 6. TCP congestion avoidance with a low RTT persistently increases the loss rate at the bottleneck using ARED, PIE or CoDel. The loss with CoDel is significantly higher than for PIE and ARED.

Fig. 5 (b) shows that even though CoDel has high drop rates its link utilisation is comparable to PIE and ARED in a range from 0.92 to 0.96. Link utilisation of DropTail stays as high as 0.98 in this simulation scenario.

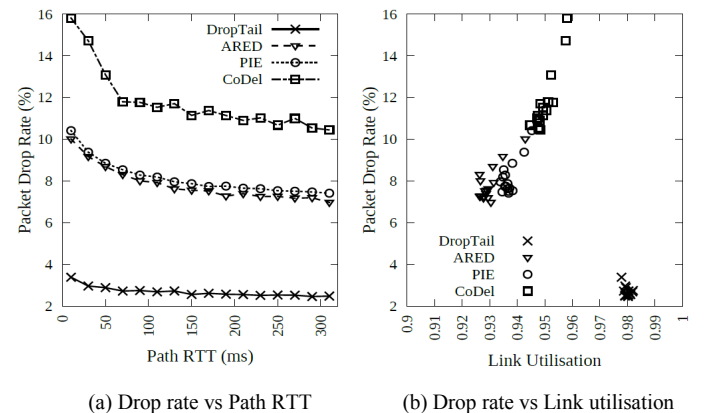


Fig. 5. Performance with a mix of flows with different path RTTs

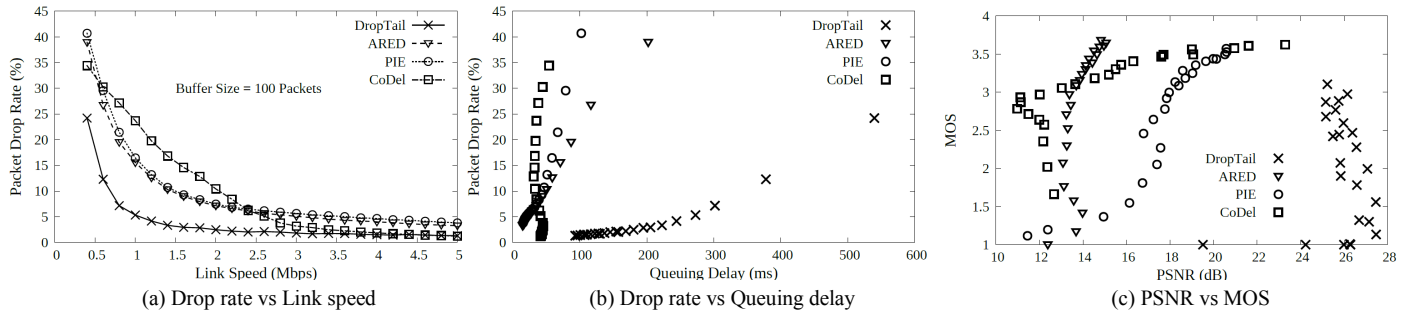


Fig. 6. Performance with different link speeds (400kbps to 5Mbps)

C. Performance with Different Link Speeds

The traffic mix was simulated with a range of bottleneck speeds (varying C from 400kbps to 5Mbps) with one-way link delays of $D=0\text{ms}$, $d_6=2.5\text{ms}$, $d_7=47.5\text{ms}$ and $d_8=147.5\text{ms}$. Fig. 6 (a) and (b) show that CoDel has a higher loss rates than ARED and PIE up to a link speed of 2.5Mbps (high level of congestion). However, CoDel reduces the loss rate under low congestion. PIE outperformed the other AQM algorithms with regard to PSNR, as shown in Fig. 6 (c).

The same scenario was simulated, but replacing NewReno with Cubic TCP, Compound TCP and LEDBAT (with a target delay of 25ms) using the ns2 Linux TCP implementation. The results (Fig. 7) show Cubic with ARED and PIE resulted in more loss, but this loss was not considered significant ($<2\%$). Excessive loss was not observed with Cubic using CoDel.

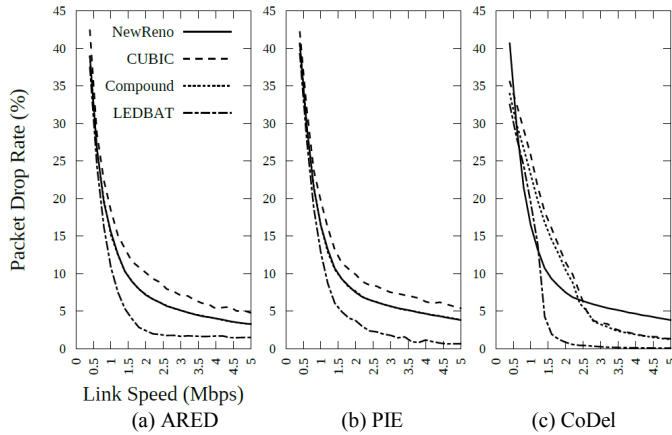


Fig. 7. Packet drop rates vs link speeds under different TCP variants

D. Performance with Fluctuating Link Speeds

The above scenario was simulated with a fluctuating link capacity to investigate the effect of the link capacity in the th_{min} calculation by ARED. The link speed (C) was randomly and uniformly changed at $[10:30\text{ms}]$ times selecting one of the capacities; 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 and 3.5 Mbps. These simulations were run (over 50 seed values) with three different initial link rates (C); Min (0.5Mbps), Avg (2Mbps) and Max (3.5Mbps). ARED considered these initial rates in th_{min} calculation for the entire duration of simulation.

Fig. 8 (a) shows that queuing delays of ARED and PIE are higher when link speed fluctuates (by comparing with Fig. 2 (a) for a buffer size of 100 packets). ARED increases queuing

delay considerably when it reads an upper initial rate in the fluctuating range.

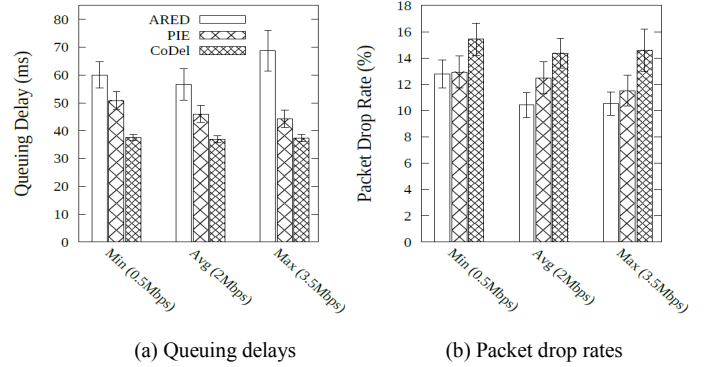


Fig. 8. Performance with fluctuating link speeds [0.5Mbps to 3.5Mbps]

All three AQM algorithms increase drop rates with fluctuating link capacity, as shown in Fig. 8 (b). These issues will be investigated in future simulations to evaluate ARED performance with fluctuating link capacity.

IV. DISCUSSION

Our results have shown that CoDel has the highest drop rate of all the simulated AQMs. This is because it tries to keep the queue small and the queuing delay below a target delay. ARED and PIE presented the lowest drop rates, performing better than CoDel in our simulation studies.

The PSNR was used as a popular metric for video. This measures the average video distortion across a video sequence [20]. However, it is accepted that the PSNR measure is perceptual and other metrics, such as subjective MOS may be better, but we suggest will likely show the same trend.

We considered low speed links to simulate rural home broadband networks (2Mbps) varying the RTT between 10 and 300ms. Our results have shown that an AQM will try to keep the queuing delay (latency) to a minimum, even if the RTT is low. However, there is a tolerance to increase the queuing delay for a low RTT. Furthermore, PIE improves the MOS and also results a high PSNR than CoDel. PIE reaches an optimal region (higher PSNR and higher MOS) more often. CoDel shows a low value for PSNR due to the high drop rate and a high value for MOS.

When traffic mixes include flows with a low RTT, the flows contributed to the high observed loss rates. This rendered

unreliable video applications unusable when CoDel was used, since it results in drop rates up to 16% in a moderate level of congestion. This loss rate is significantly higher than those experienced by flows where the bottleneck used PIE or ARED. Moreover, by increasing the target delay, we noted a decrease in drop rates, especially with CoDel. This indicates that CoDel performance can be tuned for specific types of traffic. ARED is comparable to PIE in many cases when used with a link of a constant speed. However, its performance is poor with fluctuating links.

Although AQM can control the queue size, a small standing queue is required to fully utilise the subsequent network link. AQM approaches can be especially useful in home gateways where traffic patterns may be correlated on user, time of the day, etc. Setting the target delay is a major performance index of ARED, PIE and CoDel. Appropriate values could be determined based on the end-to-end latency requirement of applications. New AQM algorithms may be extended to make them adaptive based on the network and the applications requirements/behaviour. Scheduling (flow isolation) may be used as a tool to achieve this. This could lead to latency-sensitive applications experiencing low latency while possibly accepting slightly higher loss and loss-sensitive applications being able to accommodate some additional latency.

Stochastic Fair Queue (SFQ) [7] is a scheduling mechanism that has been used in routers to fairly share the negative effects of dropping among queued flows. This helps to isolate the impact of one flow that experiences loss from other flows that share a common router queue. Such methods have benefit when the link capacity creates a bottleneck shared by a small number of flows, as in a home gateway, and flow synchronisation could otherwise be frequent. SFQ may also be combined with AQM. This can yield the joint benefits of scheduling and AQM. CoDel has already extended its implementation with SFQ in SFQ-CoDel [18]. We will further investigate this area to reduce the impact of packet loss due to new AQM algorithms.

We note that AQM enables the use of Explicit Congestion Notification (ECN) [16]. Although use of AQM with bulk flows may not offer significant advantage (e.g. in throughput), ECN is likely to offer significant advantage to latency/loss sensitive applications, such as video conferencing and adaptive voice/video flows. We therefore suggest that it is important to revisit the use of ECN, and to examine the real reasons why more operators have not offered ECN, with the goal of encouraging greater support of both AQM and ECN in networks.

V. CONCLUSIONS

Our results support other data that both PIE and CoDel can significantly reduce network latency. We have compared PIE, CoDel and ARED and presented new simulation results, which show ARED, PIE and CoDel can increase loss and how this can negatively impact some applications, for example unreliable video. CoDel acted aggressively, but PIE performance showed a trade-off between MOS and PSNR.

AQM algorithms requiring setting delay targets: Too small a value can increase loss for applications with a low RTT. However, increasing the delay target may significantly impair performance for interactive applications with a high path RTT. Since most routers are unaware of the actual path RTT between end hosts this requires a trade-off that depends on the traffic dynamics at the queue. PIE was observed to perform better. We also discussed areas that need to be addressed to minimise the loss induced by these algorithms.

ACKNOWLEDGEMENTS

This research was supported by an award made by the RCUK DE programme to the dot.rural Digital Economy Hub; EP/G066051/1. The work was part-funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). We also thank David Ross for his invaluable comments and the PIE and CoDel authors for making their ns2 implementations available.

REFERENCES

- [1] J. Gettys, K. Nichols, "Bufferbloat: Dark Buffers in the Internet", *ACM Queue Networks*, Vol 9, Issue 11, November 2011.
- [2] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol 1, Issue 4, August 1993
- [3] S. Athuraliya, S. H. Low, V. H. Li, Q. Yin, "REM: Active Queue Management", *IEEE Network*, Vol 15, Issue 3, May 2001.
- [4] Sally Floyd, Ramakrishna Gummadi, Scott Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management", August 2001.
- [5] K. Nichols, V. Jacobson, "A Modern AQM is just one Piece of the Solution to Bufferbloat", *ACM Queue Networks*, Vol 10, Issue 5, May 2012
- [6] R. Pan, P. Natarajan, C. Piglione, M. Suryanarayana Prabhu, V. Subramanian, F. Baker, B. VerSteeg, "PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem", *IEEE Conference on High Performance Switching and Routing (Taipei, Taiwan)*, July 2013.
- [7] Paul E. McKenney, "Stochastic Fairness Queue", *IEEE International Conference on Computer Communications (San Francisco, USA)*, June 1990
- [8] N. Leavitt, "Network-Usage Changes Push Internet Traffic to the Edge", *IEEE Computer*, Vol 43, Issue 10, October 2010.
- [9] Mat Ford, "Internet Connections Speeds – Up and to the Right, for Some" (<http://www.internetsociety.org/blog/2013/02/internet-connection-speeds-and-right-some>).
- [10] G. Fairhurst, B. Briscoe, "Advice on Network Buffering", *IETF Internet Draft*, March 2013 (work-in-progress).
- [11] G. Fairhurst, A. Sathiseelan, C. Baudoin, E. Callejo, "Delivery of Triple-play Services over Broadband Satellite Networks", *IET Communications*, Vol 4, Issue 13, September 2010.
- [12] Network Simulator 2 (<http://www.isi.edu/nsnam/ns>).
- [13] Web Traffic Generation in NS-2 with PackMime-HTTP (<http://www.cs.odu.edu/~mweigle/research/packmime>).
- [14] Evalvid: A Video Quality Evaluation Tool Set (<http://www.tkn.tu-berlin.de/menue/research/evalvid>).
- [15] ns2voip++ (<http://cng1.iet.unipi.it/wiki/index.php/Ns2voip++>).
- [16] K. Ramakrishnan, S. Floyd, D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", *IETF RFC 3168*, September 2001.
- [17] Jason Greengrass, John Evans, Ali C. Begen, "Not All Packets are Equal, Part 2: The Impact of Network Packet Loss on Video Quality", *IEEE Internet Computing*, Vol 13, Issue 2, March 2009.
- [18] Greg White, Dan Rice, "Active Queue Management Algorithms for DOCSIS 3.0", *Cable Television Laboratories, Inc*, April 2013.
- [19] Michael Kende, "Internet Global Growth: Lessons for the Future", *Analysys Mason Limited*, September 2012.
- [20] M. Vranjes, S. Rimac-Drlje, K. Grgic, "Average PSNR as a Simple Objective Video Quality Metric", *Int. Symposium of ELMAR (Zadar, Croatia)*, September 2008.