Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: December 21, 2015

N. Kuhn, Ed.
Telecom Bretagne
N. Khademi, Ed.
University of Oslo
P. Natarajan, Ed.
Cisco Systems
D. Ros
Simula Research Laboratory AS

June 19, 2015

AQM Characterization Guidelines
draft-ietf-aqm-eval-guidelines-04

Abstract

Unmanaged large buffers in today's networks have given rise to a slew of performance issues. These performance issues can be addressed by some form of Active Queue Management (AQM) mechanism, optionally in combination with a packet scheduling scheme such as fair queuing. The IETF Active Queue Management and Packet Scheduling working group was formed to standardize AQM schemes that are robust, easily implementable, and successfully deployable in today's networks. This document describes various criteria for performing precautionary characterizations of AQM proposals. This document also helps in ascertaining whether any given AQM proposal should be taken up for standardization by the AQM WG.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 21, 2015.

Table of Contents

1. Introduction 4
1.1. Reducing the latency and maximizing the goodput 5
1.2. Guidelines for AQM evaluation 5
1.3. Requirements Language 6
1.4. Glossary 6
2. End-to-end metrics 6
2.1. Flow completion time 7
2.2. Flow start up time 7
2.3. Packet loss 7
2.4. Packet loss synchronization 8
2.5. Goodput 8
2.6. Latency and jitter 9
2.7. Discussion on the trade-off between latency and goodput . 9
3. Generic set up for evaluations 10
3.1. Topology and notations 10
3.2. Buffer size 12
3.3. Congestion controls 12
4. Transport Protocols 13
4.1. TCP-friendly sender 13
4.1.1. TCP-friendly sender with the same initial congestion window 14
4.1.2. TCP-friendly sender with different initial congestion windows 14
4.2. Aggressive transport sender 14
4.3. Unresponsive transport sender 15
4.4. Less-than Best Effort transport sender 15
5. Round Trip Time Fairness 16
5.1. Motivation 16
5.2. Recommended tests 16
5.3. Metrics to evaluate the RTT fairness 17
6. Burst Absorption 17

6.1. Motivation 17
6.2. Recommended tests 18
7. Stability 19
7.1. Motivation 19
7.2. Recommended tests 19
7.2.1. Definition of the congestion Level 19
7.2.2. Mild congestion 20
7.2.3. Medium congestion 20
7.2.4. Heavy congestion 20
7.2.5. Varying congestion levels 20
7.2.6. Varying available capacity 21
7.3. Parameter sensitivity and stability analysis 22
8. Various Traffic Profiles 22
8.1. Traffic mix 22
8.2. Bi-directional traffic 23
9. Multi-AQM Scenario 23
9.1. Motivation 23
9.2. Details on the evaluation scenario 24
10. Implementation cost 24
10.1. Motivation 24
10.2. Recommended discussion 25
11. Operator Control and Auto-tuning 25
11.1. Motivation 25
11.2. Required discussion 26
12. Interaction with ECN 26
12.1. Motivation 26
12.2. Recommended discussion 26
13. Interaction with Scheduling 26
13.1. Motivation 27
13.2. Recommended discussion 27
13.3. Assessing the interaction between AQM and scheduling . . 27
14. Discussion on Methodology, Metrics, AQM Comparisons and Packet Sizes 27
14.1. Methodology 27
14.2. Comments on metrics measurement 28
14.3. Comparing AQM schemes 28
14.3.1. Performance comparison 28
14.3.2. Deployment comparison 29
14.4. Packet sizes and congestion notification 29
15. Conclusion 30
16. Acknowledgements 31
17. Contributors 31
18. IANA Considerations 32
19. Security Considerations 32
20. References 32
20.1. Normative References 32
20.2. Informative References 32
Authors' Addresses 34

1. Introduction

Active Queue Management (AQM) [I-D.ietf-aqm-recommendation] addresses the concerns arising from using unnecessarily large and unmanaged buffers to improve network and application performance. Several AQM algorithms have been proposed in the past years, most notably Random Early Detection (RED), BLUE, and Proportional Integral controller (PI), and more recently CoDel [CODEL] and PIE [PIE]. In general, these algorithms actively interact with the Transmission Control Protocol (TCP) and any other transport protocol that deploys a congestion control scheme to manage the amount of data they keep in the network. The available buffer space in the routers and switches should be large enough to accommodate the short-term buffering requirements. AQM schemes aim at reducing mean buffer occupancy, and therefore both end-to-end delay and jitter. Some of these algorithms, notably RED, have also been widely implemented in some network devices. However, the potential benefits of the RED scheme have not been realized since RED is reported to be usually turned off. The main reason of this reluctance to use RED in today's deployments comes from its sensitivity to the operating conditions in the network and the difficulty of tuning its parameters.

A buffer is a physical volume of memory in which a queue or set of queues are stored. In real implementations of switches, a global memory is shared between the available devices: the size of the buffer for a given communication does not make sense, as its dedicated memory may vary over the time and real-world buffering architectures are complex. For the sake of simplicity, when speaking of a specific queue in this document, "buffer size" refers to the maximum amount of data the buffer may store, which can be measured in bytes or packets. The rest of this memo therefore refers to the maximum queue depth as the size of the buffer for a given communication.

Bufferbloat [BB2011] is the consequence of deploying large unmanaged buffers on the Internet, which has lead to an increase in end-to-end delay: the buffering has often been measured to be ten times or hundred times larger than needed. This results in poor performance for latency-sensitive applications such as real-time multimedia (e.g., voice, video, gaming, etc). The degree to which this affects modern networking equipment, especially consumer-grade equipment's, produces problems even with commonly used web services. Active queue management is thus essential to control queuing delay and decrease network latency.

The Active Queue Management and Packet Scheduling Working Group (AQM WG) was recently formed within the TSV area to address the problems with large unmanaged buffers in the Internet. Specifically, the AQM

WG is tasked with standardizing AQM schemes that not only address concerns with such buffers, but also are robust under a wide variety of operating conditions.

In order to ascertain whether the WG should undertake standardizing an AQM proposal, the WG requires guidelines for assessing AQM proposals. This document provides the necessary characterization guidelines. There may be a debate on whether a scheduling scheme is additional to an AQM algorithm or is a part of an AQM algorithm. The rest of this memo refers to AQM as a dropping/marking policy that does not feature a scheduling scheme. This document may be complemented with another one on guidelines for assessing combination of packet scheduling and AQM. We note that such a document will inherit all the guidelines from this document plus any additional scenarios relevant for packet scheduling such as flow starvation evaluation or impact of the number of hash buckets.

1.1. Reducing the latency and maximizing the goodput

The trade-off between reducing the latency and maximizing the goodput is intrinsically linked to each AQM scheme and is key to evaluating its performance. This trade-off MUST be considered in various scenarios to ensure the safety of an AQM deployment. Whenever possible, solutions ought to aim at both maximizing goodput and minimizing latency. This document provides guidelines that enable the reader to quantify (1) reduction of latency, (2) maximization of goodput and (3) the trade-off between the two.

These guidelines provide the tools to understand the deployment costs versus the potential gain in performance from the introduction of the proposed scheme.

1.2. Guidelines for AQM evaluation

The guidelines help to quantify performance of AQM schemes in terms of latency reduction, goodput maximization and the trade-off between these two. The guidelines also help to discuss safe deployment of AQM, including self-adaptation, stability analysis, fairness, design and implementation complexity and robustness to different operating conditions.

This memo details generic characterization scenarios against which any AQM proposal must be evaluated, irrespective of whether or not an AQM is standardized by the IETF. This documents recommends the relevant scenarios and metrics to be considered.

The document presents central aspects of an AQM algorithm that must be considered whatever the context, such as burst absorption

capacity, RTT fairness or resilience to fluctuating network conditions. These guidelines do not cover every possible aspect of a particular algorithm. In addition, it is worth noting that the proposed criteria are not bound to a particular evaluation toolset.

This document details how an AQM designer can rate the feasibility of their proposal in different types of network devices (switches, routers, firewalls, hosts, drivers, etc) where an AQM may be implemented. However, these guidelines do not present context-dependent scenarios (such as 802.11 WLANs, data-centers or rural broadband networks).

1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.4. Glossary

- AQM: there may be a debate on whether a scheduling scheme is additional to an AQM algorithm or is a part of an AQM algorithm. The rest of this memo refers to AQM as a dropping/marking policy that does not feature a scheduling scheme.

- buffer: a physical volume of memory in which a queue or set of queues are stored.

- buffer size: the maximum amount of data that may be stored in a buffer, measured in bytes or packets.

2. End-to-end metrics

End-to-end delay is the result of propagation delay, serialization delay, service delay in a switch, medium-access delay and queuing delay, summed over the network elements along the path. AQM schemes may reduce the queuing delay by providing signals to the sender on the emergence of congestion, but any impact on the goodput must be carefully considered. This section presents the metrics that could be used to better quantify (1) the reduction of latency, (2) maximization of goodput and (3) the trade-off between these two. This section provides normative requirements for metrics that can be used to assess the performance of an AQM scheme.

Some metrics listed in this section are not suited to every type of traffic detailed in the rest of this document. It is therefore not necessary to measure all of the following metrics: the chosen metric may not be relevant to the context of the evaluation scenario (e.g.

latency vs. goodput trade-off in application-limited traffic scenarios). Guidance is provided for each metric.

2.1. Flow completion time

The flow completion time is an important performance metric for the end-user when the flow size is finite. Considering the fact that an AQM scheme may drop/mark packets, the flow completion time is directly linked to the dropping/marking policy of the AQM scheme. This metric helps to better assess the performance of an AQM depending on the flow size. The Flow Completion Time (FCT) is related to the flow size (Fs) and the goodput for the flow (G) as follows:

FCT [s] = Fs [B] / (G [Mbps] / 8)

If this metric is used to evaluate the performance of web transfers, we propose to rather consider the time needed to download all the objects that compose the web page, as this makes more sense in terms of user experience than assessing the time needed to download each object.

2.2. Flow start up time

The flow start up time is the time between the request has been sent from the client and the server starts to transmit data. The amount of packets dropped by an AQM may seriously affect the waiting period during which the data transfer has not started. This metric would specifically focus on the operations such as DNS lookups, TCP opens of SSL handshakes.

2.3. Packet loss

Packet loss can occur within a network device, this can impact the end-to-end performance measured at receiver.

The tester SHOULD evaluate loss experienced at the receiver using one of the two metrics:

- the packet loss probability: this metric is to be frequently measured during the experiment. The long-term loss probability is of interest for steady-state scenarios only;

- the interval between consecutive losses: the time between two losses is to be measured.

The packet loss probability can be assessed by simply evaluating the loss ratio as a function of the number of lost packets and the total

number of packets sent. This might not be easily done in laboratory testing, for which these guidelines advice the tester:

- to check that for every packet, a corresponding packet was received within a reasonable time, as explained in [RFC2680].

- to keep a count of all packets sent, and a count of the non-duplicate packets received, as explained in the section 10 of [RFC2544].

The interval between consecutive losses, which is also called a gap, is a metric of interest for VoIP traffic and, as a result, has been further specified in [RFC3611].

2.4. Packet loss synchronization

One goal of an AQM algorithm ought be to help to avoid global synchronization of flows sharing a bottleneck buffer on which the AQM operates ([RFC2309],[I-D.ietf-aqm-recommendation]). The "degree" of packet-loss synchronization between flows SHOULD be assessed, with and without the AQM under consideration.

As discussed e.g. in [LOSS-SYNCH-MET-08], loss synchronization among flows may be quantified by several slightly different metrics that capture different aspects of the same issue. However, in real-world measurements the choice of metric could be imposed by practical considerations -- e.g. whether fine-grained information on packet losses in the bottleneck available or not. For the purpose of AQM characterization, a good candidate metric is the global synchronization ratio, measuring the proportion of flows losing packets during a loss event. [YU06] used this metric in real-world experiments to characterize synchronization along arbitrary Internet paths; the full methodology is described in [YU06].

If an AQM scheme is evaluated using real-life network environments, it is worth pointing out that some network events, such as failed link restoration may cause synchronized losses between active flows and thus confuse the meaning of this metric.

2.5. Goodput

The goodput has been defined in the section 3.17 of [RFC2647] as the number of bits per unit of time forwarded to the correct destination interface of the Device Under Test (DUT) or the System Under Test (SUT), minus any bits lost or retransmitted. This definition induces that the test setup needs to be qualified to assure that it is not generating losses on its own.

Measuring the end-to-end goodput provides an appreciation of how well an AQM scheme improves transport and application performance. The measured end-to-end goodput is linked to the dropping/marking policy of the AQM scheme -- e.g. the fewer the number of packet drops, the fewer packets need retransmission, minimizing the impact of AQM on transport and application performance. Additionally, an AQM scheme may resort to Explicit Congestion Notification (ECN) marking as an initial means to control delay. Again, marking packets instead of dropping them reduces the number of packet retransmissions and increases goodput. End-to-end goodput values help to evaluate the AQM scheme's effectiveness of an AQM scheme in minimizing packet drops that impact application performance and to estimate how well the AQM scheme works with ECN.

The measurement of the goodput allows the tester evaluate to which extent an AQM is able to maintain a high bottleneck utilization. This metric should be also obtained frequently during an experiment as the long-term goodput is relevant for steady-state scenarios only and may not necessarily reflect how the introduction of an AQM actually impacts the link utilization during at a certain period of time. Fluctuations in the values obtained from these measurements may depend on other factors than the introduction of an AQM, such as link layer losses due to external noise or corruption, fluctuating bandwidths (802.11 WLANs), heavy congestion levels or transport layer's rate reduction by congestion control mechanism.

2.6. Latency and jitter

The latency, or the one-way delay metric, is discussed in [RFC2679]. There is a consensus on a adequate metric for the jitter, that represents the one-way delay variations for packets from the same flow: the Packet Delay Variation (PDV), detailed in [RFC5481], serves well all use cases.

The end-to-end latency differs from the queuing delay: it is linked to the network topology and the path characteristics. Moreover, the jitter also strongly depends on the traffic pattern and the topology. The introduction of an AQM scheme would impact these metrics and therefore they should be considered in the end-to-end evaluation of performance.

2.7. Discussion on the trade-off between latency and goodput

The metrics presented in this section may be considered as explained in the rest of this document, in order to discuss and quantify the trade-off between latency and goodput.

This trade-off can also be illustrated with figures following the recommendations of section 5 of [TCPEVAL2013]. Each of the end-to-end delay and the goodput SHOULD be measured frequently for every fixed time interval.

With regards to the goodput, and in addition to the long-term stationary goodput value, it is RECOMMENDED to take measurements every multiple of RTTs. We suggest a minimum value of 10 x RTT (to smooth out the fluctuations) but higher values are encouraged whenever appropriate for the presentation depending on the network's path characteristics. The measurement period MUST be disclosed for each experiment and when results/values are compared across different AQM schemes, the comparisons SHOULD use exactly the same measurement periods.

With regards to latency, it is highly RECOMMENDED to take the samples on per-packet basis whenever possible depending on the features provided by hardware/software and the impact of sampling itself on the hardware performance. It is generally RECOMMENDED to provide at least 10 samples per RTT.

From each of these sets of measurements, the 10th and 90th percentiles and the median value SHOULD be computed. For each scenario, a graph can be generated, with the x-axis showing the end-to-end delay and the y-axis the goodput. This graph provides part of a better understanding of (1) the delay/goodput trade-off for a given congestion control mechanism, and (2) how the goodput and average queue size vary as a function of the traffic load.

3. Generic set up for evaluations

This section presents the topology that can be used for each of the following scenarios, the corresponding notations and discusses various assumptions that have been made in the document.

3.1. Topology and notations

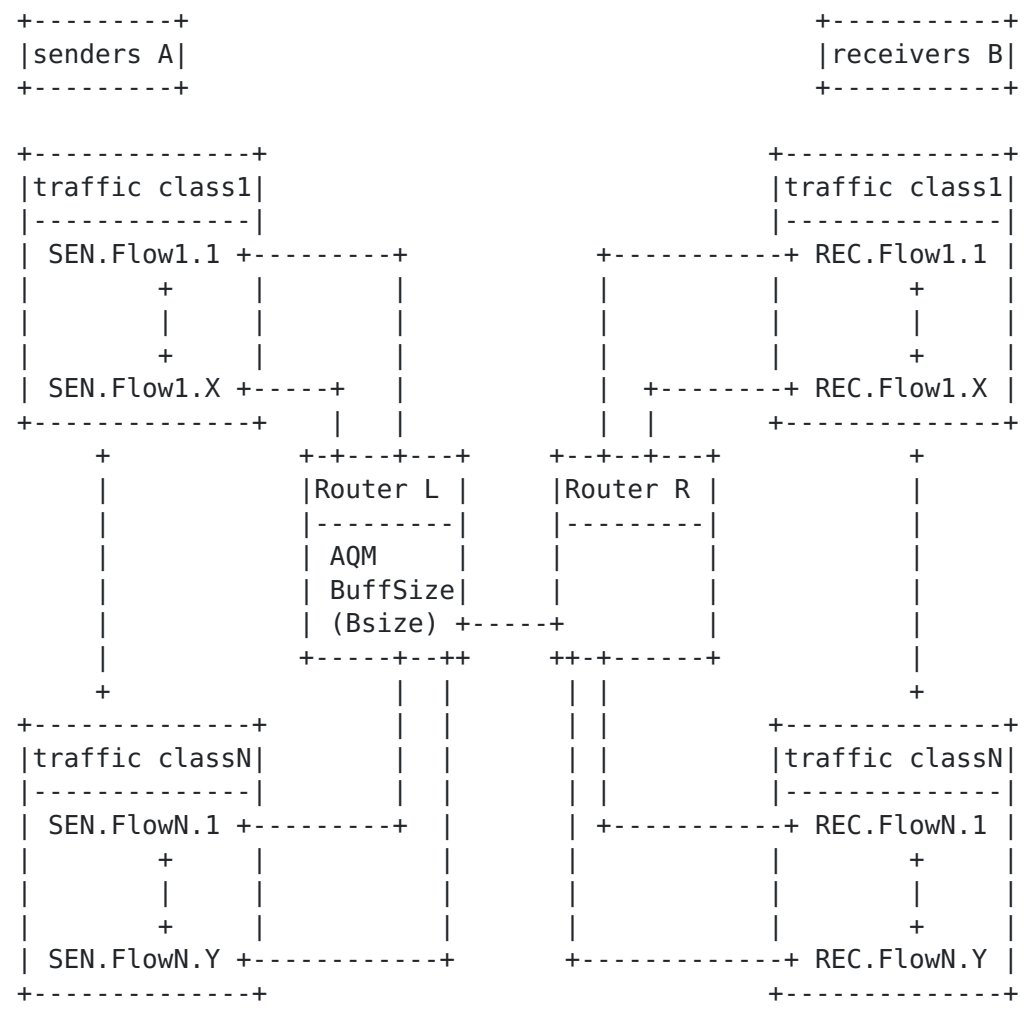

Figure 1: Topology and notations

Figure 1 is a generic topology where:

- various classes of traffic can be introduced;
- the timing of each flow could be different (i.e., when does each flow start and stop);
- each class of traffic can comprise various number of flows;
- each link is characterized by a couple (RTT,capacity);
- flows are generated between A and B, sharing a bottleneck (Routers L and R);

- the tester SHOULD consider both scenarios of asymmetric and symmetric bottleneck links in terms of bandwidth. In case of asymmetric link, the capacity from senders to receivers is higher than the one from receivers to senders; the symmetric link scenario provides a basic understanding of the operation of the AQM mechanism whereas the asymmetric link scenario evaluates an AQM mechanism in a more realistic setup;

- in asymmetric link scenarios, the tester SHOULD study the bi-directional traffic between A and B (downlink and uplink) with the AQM mechanism deployed on one direction only. The tester MAY additionally consider a scenario with AQM mechanism being deployed on both directions. In each scenario, the tester SHOULD investigate the impact of drop policy of the AQM on TCP ACK packets and its impact on the performance.

Although this topology may not perfectly reflect actual topologies, the simple topology is commonly used in the world of simulations and small testbeds. It can be considered as adequate to evaluate AQM proposals, similarly to the topology proposed in [TCPEVAL2013]. Testers ought to pay attention to the topology that has been used to evaluate an AQM scheme when comparing this scheme with a new proposed AQM scheme.

3.2. Buffer size

The size of the buffers should be carefully chosen, and is to be set to the bandwidth-delay product; the bandwith being the bottleneck capacity and the delay the larger RTT in the considered network. The size of the buffer can impact on the AQM performance and is a dimensioning parameter that will be considered when comparing AQM proposals.

If the context or the application requires a specific buffer size, the tester MUST justify and detail the way the maximum queue size is set. Indeed, the maximum size of the buffer may affect the AQM's performance and its choice SHOULD be elaborated for a fair comparison between AQM proposals. While comparing AQM schemes the buffer size SHOULD remain the same across the tests.

3.3. Congestion controls

This memo features three kind of congestion controls:

- Standard TCP congestion control: the base-line congestion control is TCP NewReno with SACK, as explained in [RFC5681].

- Aggressive congestion controls: a base-line congestion control for this category is TCP Cubic.

- Less-than Best Effort (LBE) congestion controls: an LBE congestion control 'results in smaller bandwidth and/or delay impact on standard TCP than standard TCP itself, when sharing a bottleneck with it.' [RFC6297]

Other transport congestion controls can OPTIONALLY be evaluatied in addition. Recent transport layer protocols are not mentioned in the following sections, for the sake of simplicity.

4. Transport Protocols

Network and end-devices need to be configured with a reasonable amount of buffer space to absorb transient bursts. In some situations, network providers tend to configure devices with large buffers to avoid packet drops triggered by a full buffer and to maximize the link utilization for standard loss-based TCP traffic.

AQM algorithms are often evaluated by considering Transmission Control Protocol (TCP) [RFC0793] with a limited number of applications. TCP is a widely deployed transport. It fills up unmanaged buffers until the TCP sender receives a signal (packet drop) that reduces the sending rate. The larger the buffer, the higher the buffer occupancy, and therefore the queuing delay. An efficient AQM scheme sends out early congestion signals to TCP to bring the queuing delay under control.

Not all applications using TCP use the same flavor of TCP. Variety of senders generate different classes of traffic which may not react to congestion signals (aka non-responsive flows [I-D.ietf-aqm-recommendation]) or may not reduce their sending rate as expected (aka Transport Flows that are less responsive than TCP[I-D.ietf-aqm-recommendation], also called "aggressive flows"). In these cases, AQM schemes seek to control the queuing delay.

This section provides guidelines to assess the performance of an AQM proposal for various traffic profiles -- different types of senders (with different TCP congestion control variants, unresponsive, aggressive).

4.1. TCP-friendly sender

4.1.1. TCP-friendly sender with the same initial congestion window

This scenario helps to evaluate how an AQM scheme reacts to a TCP-friendly transport sender. A single long-lived, non application-limited, TCP NewReno flow, with an Initial congestion Window (IW) set to 3 packets, transfers data between sender A and receiver B. Other TCP friendly congestion control schemes such as TCP-friendly rate control [RFC5348] etc MAY also be considered.

For each TCP-friendly transport considered, the graph described in Section 2.7 could be generated.

4.1.2. TCP-friendly sender with different initial congestion windows

This scenario can be used to evaluate how an AQM scheme adapts to a traffic mix consisting of TCP flows with different values of the IW.

For this scenario, two types of flows MUST be generated between sender A and receiver B:

- A single long-lived non application-limited TCP NewReno flow;
- A single long-lived application-limited TCP NewReno flow, with an IW set to 3 or 10 packets. The size of the data transferred must be strictly higher than 10 packets and should be lower than 100 packets.

The transmission of the non application-limited flow must start before the transmission of the application-limited flow and only after the steady state has been reached by non application-limited flow.

For each of these scenarios, the graph described in Section 2.7 could be generated for each class of traffic (application-limited and non application-limited). The completion time of the application-limited TCP flow could be measured.

4.2. Aggressive transport sender

This scenario helps testers to evaluate how an AQM scheme reacts to a transport sender that is more aggressive than a single TCP-friendly sender. We define 'aggressiveness' as a higher increase factor than standard upon a successful transmission and/or a lower than standard decrease factor upon a unsuccessful transmission (e.g. in case of congestion controls with Additive-Increase Multiplicative-Decrease (AIMD) principle, a larger AI and/or MD factors). A single long-lived, non application-limited, TCP Cubic flow transfers data between

sender A and receiver B. Other aggressive congestion control schemes MAY also be considered.

For each flavor of aggressive transports, the graph described in Section 2.7 could be generated.

4.3. Unresponsive transport sender

This scenario helps testers to evaluate how an AQM scheme reacts to a transport sender that is less responsive than TCP. Note that faulty transport implementations on an end host and/or faulty network elements en-route that "hide" congestion signals in packet headers [I-D.ietf-aqm-recommendation] may also lead to a similar situation, such that the AQM scheme needs to adapt to unresponsive traffic. To this end, these guidelines propose the two following scenarios.

The first scenario can be used to evaluate queue build up. It considers unresponsive flow(s) whose sending rate is greater than the bottleneck link capacity between routers L and R. This scenario consists of a long-lived non application limited UDP flow transmits data between sender A and receiver B. Graphs described in Section 2.7 could be generated.

The second scenario can be used to evaluate if the AQM scheme is able to keep responsive fraction under control. This scenario considers a mixture of TCP-friendly and unresponsive traffics. It consists of a long-lived non application-limited UDP flow and a single long-lived, non-application-limited, TCP New Reno flow that transmit data between sender A and receiver B. As opposed to the first scenario, the rate of the UDP traffic should not be greater than the bottleneck capacity, and should not be higher than half of the bottleneck capacity. For each type of traffic, the graph described in Section 2.7 could be generated.

4.4. Less-than Best Effort transport sender

This scenario helps to evaluate how an AQM scheme reacts to LBE congestion controls that 'results in smaller bandwidth and/or delay impact on standard TCP than standard TCP itself, when sharing a bottleneck with it.' [RFC6297]. The potential fateful interaction when AQM and LBE techniques are combined has been shown in [LBE-AQM]; this scenario helps to evaluate whether the coexistence of the proposed AQM and LBE techniques may be possible.

Single long-lived non application-limited TCP NewReno flows transfer data between sender A and receiver B. Other TCP-friendly congestion control schemes MAY also be considered. Single long-lived non application-limited LEDBAT [RFC6817] flows transfer data between

sender A and receiver B. We recommend to set the target delay and gain values of LEDBAT respectively to 5 ms and 10 [LEDBAT-PARAM]. Other LBE congestion control schemes, any of those listed in [RFC6297], MAY also be considered.

For each of the TCP-friendly and LBE transports, the graph described in Section 2.7 could be generated.

5. Round Trip Time Fairness

5.1. Motivation

The ability of AQM schemes to control the queuing delay highly depends on the way end-to-end protocols react to congestion signals. When the RTT varies, the behaviour of a congestion control is impacted and this impacts the ability of an AQM scheme to control the queue. It is therefore important to assess the AQM schemes for a set of RTTs (e.g., from 5 ms to 200 ms).

The asymmetry in terms of difference in intrinsic RTT between various paths sharing the same bottleneck SHOULD be considered so that the fairness between the flows can be discussed since in this scenario, a flow traversing on shorter RTT path may react faster to congestion and recover faster from it compared to another flow on a longer RTT path. The introduction of AQM schemes may potentially improve this type of fairness.

Introducing an AQM scheme may cause the unfairness between the flows, even if the RTTs are identical. This potential unfairness SHOULD be investigated as well.

5.2. Recommended tests

The RECOMMENDED topology is detailed in Figure 1:

- To evaluate the inter-RTT fairness, for each run, two flows divided into two categories. Category I which RTT between sender A and Router L SHOULD be 100ms. Category II which RTT between sender A and Router L should be in [5ms;560ms]. The maximum value for the RTT represents the RTT of a satellite link that, according to the section 2 of [RFC2488] should be at least 558ms.

- To evaluate the impact of the RTT value on the AQM performance and the intra-protocol fairness (the fairness for the flows using the same paths/congestion control), for each run, two flows (Flow1 and Flow2) should be introduced. For each experiment, the set of RTT SHOULD be the same for the two flows and in [5ms;560ms].

A set of evaluated flows MUST use the same congestion control algorithm.

5.3. Metrics to evaluate the RTT fairness

The outputs that MUST be measured are:

- for the inter-RTT fairness: (1) the cumulative average goodput of the flow from Category I, goodput_Cat_I (Section 2.5); (2) the cumulative average goodput of the flow from Category II, goodput_Cat_II (Section 2.5); (3) the ratio goodput_Cat_II/ goodput_Cat_I; (4) the average packet drop rate for each category (Section 2.3).

- for the intra-protocol RTT fairness: (1) the cumulative average goodput of the two flows (Section 2.5); (2) the average packet drop rate for the two flows (Section 2.3).

6. Burst Absorption

"AQM mechanisms need to control the overall queue sizes, to ensure that arriving bursts can be accommodated without dropping packets" [I-D.ietf-aqm-recommendation]

6.1. Motivation

An AQM scheme can result in bursts of packet arrivals due to various reasons. Dropping one or more packets from a burst can result in performance penalties for the corresponding flows, since dropped packets have to be retransmitted. Performance penalties can result in failing to meet SLAs and be a disincentive to AQM adoption.

The ability to accommodate bursts translates to larger queue length and hence more queuing delay. On the one hand, it is important that an AQM scheme quickly brings bursty traffic under control. On the other hand, a peak in the packet drop rates to bring a packet burst quickly under control could result in multiple drops per flow and severely impact transport and application performance. Therefore, an AQM scheme ought to bring bursts under control by balancing both aspects -- (1) queuing delay spikes are minimized and (2) performance penalties for ongoing flows in terms of packet drops are minimized.

An AQM scheme that maintains short queues allows some remaining space in the queue for bursts of arriving packets. The tolerance to bursts of packets depends upon the number of packets in the queue, which is directly linked to the AQM algorithm. Moreover, one AQM scheme may implement a feature controlling the maximum size of accepted bursts, that can depend on the buffer occupancy or the currently estimated

queuing delay. The impact of the buffer size on the burst allowance may be evaluated.

6.2. Recommended tests

For this scenario, tester MUST evaluate how the AQM performs with the following traffic generated from sender A to receiver B:

- Web traffic with IW10;
- Bursty video frames;
- Constant bit rate UDP traffic.
- A single bulk TCP flow as background traffic.

Figure 2 presents the various cases for the traffic that MUST be generated between sender A and receiver B.

Case	Traffic Type				
	Video	Webs (IW 10)	CBR	Bulk TCP Traffic	
I	0	1	1	0	
II	0	1	1	1	
III	1	1	1	0	
IV	1	1	1	1	

Figure 2: Bursty traffic scenarios

A new web page download could start after the previous web page download is finished. Each web page could be composed by at least 50 objects and the size of each object should be at least 1kB. 6 TCP parallel connections SHOULD be generated to download the objects, each parallel connections having an initial congestion window set to 10 packets.

For each of these scenarios, the graph described in Section 2.7 could be generated. Metrics such as end-to-end latency, jitter, flow completion time MAY be generated. For the cases of frame generation of bursty video traffic as well as the choice of web traffic pattern, we leave these details and their presentation to the testers.

7. Stability

7.1. Motivation

Network devices can experience varying operating conditions depending on factors such as time of the day, deployment scenario, etc. For example:

- Traffic and congestion levels are higher during peak hours than off-peak hours.
- In the presence of a scheduler, the draining rate of a queue can vary depending on the occupancy of other queues: a low load on a high priority queue implies a higher draining rate for the lower priority queues.
- The available capacity at the physical layer can vary over time (e.g., a lossy channel, a link supporting traffic in a higher diffserv class).

Whether the target context is a not stable environment, the ability of an AQM scheme to maintain its control over the queuing delay and buffer occupancy can be challenged. This document proposes guidelines to assess the behavior of AQM schemes under varying congestion levels and varying draining rates.

7.2. Recommended tests

Note that the traffic profiles explained below comprises non application-limited TCP flows. For each of the below scenarios, the results described in Section 2.7 SHOULD be generated. For Section 7.2.5 and Section 7.2.6 they SHOULD incorporate the results in per-phase basis as well.

Wherever the notion of time has explicitly mentioned in this subsection, time 0 starts from the moment all TCP flows have already reached their congestion avoidance phase.

7.2.1. Definition of the congestion Level

In these guidelines, the congestion levels are represented by the projected packet drop rate, had a drop-tail queue was chosen instead of an AQM scheme. When the bottleneck is shared among non-application-limited TCP flows. l_r, the loss rate projection can be expressed as a function of N, the number of bulk TCP flows, and S, the sum of the bandwidth-delay product and the maximum buffer size, both expressed in packets, based on Eq. 3 of [SCL-TCP]:

l_r = 0.76 * N^2 / S^2

N = S * sqrt(1/0.76) * sqrt (l_r)

These guidelines use the loss rate to define the different congestion levels, but they do not stipulate that in other circumstances, measuring the congestion level gives you an accurate estimation of the loss rate or vice-versa.

7.2.2. Mild congestion

This scenario can be used to evaluate how an AQM scheme reacts to a light load of incoming traffic resulting in mild congestion -- packet drop rates around 0.1%. The number of bulk flows required to achieve this congestion level, N_mild, is then:

N_mild = round(0.036*S)

7.2.3. Medium congestion

This scenario can be used to evaluate how an AQM scheme reacts to incoming traffic resulting in medium congestion -- packet drop rates around 0.5%. The number of bulk flows required to achieve this congestion level, N_med, is then:

N_med = round (0.081*S)

7.2.4. Heavy congestion

This scenario can be used to evaluate how an AQM scheme reacts to incoming traffic resulting in heavy congestion -- packet drop rates around 1%. The number of bulk flows required to achieve this congestion level, N_heavy, is then:

N_heavy = round (0.114*S)

7.2.5. Varying congestion levels

This scenario can be used to evaluate how an AQM scheme reacts to incoming traffic resulting in various level of congestions during the experiment. In this scenario, the congestion level varies within a large time-scale. The following phases may be considered: phase I - mild congestion during 0-20s; phase II - medium congestion during 20-40s; phase III - heavy congestion during 40-60s; phase I again, and so on.

7.2.6. Varying available capacity

This scenario can be used to help characterize how the AQM behaves and adapts to bandwidth changes. The experiments are not meant to reflect the exact conditions of Wi-Fi environments since its hard to design repetitive experiments or accurate simulations for such scenarios.

To emulate varying draining rates, the bottleneck capacity between nodes 'Router L' and 'Router R' varies over the course of the experiment as follows:

- Experiment 1: the capacity varies between two values within a large time-scale. As an example, the following phases may be considered: phase I - 100Mbps during 0-20s; phase II - 10Mbps during 20-40s; phase I again, and so on.

- Experiment 2: the capacity varies between two values within a short time-scale. As an example, the following phases may be considered: phase I - 100Mbps during 0-100ms; phase II - 10Mbps during 100-200ms; phase I again, and so on.

The tester MAY choose a phase time-interval value different than what is stated above, if the network's path conditions (such as bandwidth-delay product) necessitate. In this case the choice of such time-interval value SHOULD be stated and elaborated.

The tester MAY additionally evaluate the two mentioned scenarios (short-term and long-term capacity variations), during and/or including TCP slow-start phase.

More realistic fluctuating capacity patterns MAY be considered. The tester MAY choose to incorporate realistic scenarios with regards to common fluctuation of bandwidth in state-of-the-art technologies.

The scenario MAY consist of TCP NewReno flows between sender A and receiver B. To better assess the impact of draining rates on the AQM behavior, the tester MUST compare its performance with those of drop-tail and SHOULD provide a reference document for their proposal discussing performance and deployment compared to those of drop-tail. Burst traffic, such as presented in Section 6.2, could also be considered to assess the impact of varying available capacity on the burst absorption of the AQM.

7.3. Parameter sensitivity and stability analysis

The control law used by an AQM is the primary means by which the queuing delay is controlled. Hence understanding the control law is critical to understanding the behavior of the AQM scheme. The control law could include several input parameters whose values affect the AQM scheme's output behavior and its stability. Additionally, AQM schemes may auto-tune parameter values in order to maintain stability under different network conditions (such as different congestion levels, draining rates or network environments). The stability of these auto-tuning techniques is also important to understand.

Transports operating under the control of AQM experience the effect of multiple control loops that react over different timescales. It is therefore important that proposed AQM schemes are seen to be stable when they are deployed at multiple points of potential congestion along an Internet path. The pattern of congestion signals (loss or ECN-marking) arising from AQM methods also need to not adversely interact with the dynamics of the transport protocols that they control.

AQM proposals SHOULD provide background material showing control theoretic analysis of the AQM control law and the input parameter space within which the control law operates as expected; or could use another way to discuss the stability of the control law. For parameters that are auto-tuned, the material SHOULD include stability analysis of the auto-tuning mechanism(s) as well. Such analysis helps to understand an AQM control law better and the network conditions/deployments under which the AQM is stable.

8. Various Traffic Profiles

This section provides guidelines to assess the performance of an AQM proposal for various traffic profiles such as traffic with different applications or bi-directional traffic.

8.1. Traffic mix

This scenario can be used to evaluate how an AQM scheme reacts to a traffic mix consisting of different applications such as:

- Bulk TCP transfer
- Web traffic
- VoIP

- Constant Bit Rate (CBR) UDP traffic

- Adaptive video streaming

Various traffic mixes can be considered. These guidelines RECOMMEND to examine at least the following example: 1 bi-directional VoIP; 6 Webs pages download (such as detailed in Section 6.2); 1 CBR; 1 Adaptive Video; 5 bulk TCP. Any other combinations could be considered and should be carefully documented.

For each scenario, the graph described in Section 2.7 could be generated for each class of traffic. Metrics such as end-to-end latency, jitter and flow completion time MAY be reported.

8.2. Bi-directional traffic

Control packets such as DNS requests/responses, TCP SYNs/ACKs are small, but their loss can severely impact the application performance. The scenario proposed in this section will help in assessing whether the introduction of an AQM scheme increases the loss probability of these important packets.

For this scenario, traffic MUST be generated in both downlink and uplink, such as defined in Section 3.1. These guidelines RECOMMEND to consider a mild congestion level and the traffic presented in Section 7.2.2 in both directions. In this case, the metrics reported MUST be the same as in Section 7.2 for each direction.

The traffic mix presented in Section 8.1 MAY also be generated in both directions.

9. Multi-AQM Scenario

9.1. Motivation

Transports operating under the control of AQM experience the effect of multiple control loops that react over different timescales. It is therefore important that proposed AQM schemes are seen to be stable when they are deployed at multiple points of potential congestion along an Internet path. The pattern of congestion signals (loss or ECN-marking) arising from AQM methods also need to not adversely interact with the dynamics of the transport protocols that they control.

9.2. Details on the evaluation scenario

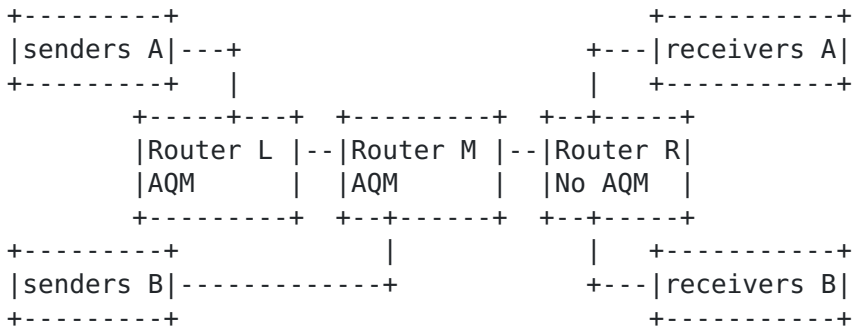

Figure 3: Topology for the Multi-AQM scenario

This scenario can be used to evaluate how having AQM schemes in sequence impact the induced latency reduction, the induced goodput maximization and the trade-off between these two. The topology presented in Figure 3 could be used. We recommend that the AQM schemes introduced in Router L and Router M should be the same; any other configurations could be considered. For this scenario, we recommend to consider a mild congestion level, the number of flows specified in Section 7.2.2 being equally shared among senders A and B. Any other relevant combination of congestion levels could be considered. We recommend to measure the metrics presented in Section 7.2.

10. Implementation cost

10.1. Motivation

Successful deployment of AQM is directly related to its cost of implementation. Network devices can need hardware or software implementations of the AQM mechanism. Depending on a device's capabilities and limitations, the device may or may not be able to implement some or all parts of the AQM logic.

AQM proposals SHOULD provide pseudo-code for the complete AQM scheme, highlighting generic implementation-specific aspects of the scheme such as "drop-tail" vs. "drop-head", inputs (e.g. current queuing delay, queue length), computations involved, need for timers, etc. This helps to identify costs associated with implementing the AQM scheme on a particular hardware or software device. This also helps the WG understand which kind of devices can easily support the AQM and which cannot.

10.2. Recommended discussion

AQM proposals SHOULD highlight parts of AQM logic that are device dependent and discuss if and how AQM behavior could be impacted by the device. For example, a queueing-delay based AQM scheme requires current queuing delay as input from the device. If the device already maintains this value, then it can be trivial to implement the AQM logic on the device. If the device provides indirect means to estimate the queuing delay (for example: timestamps, dequeuing rate), then the AQM behavior is sensitive to the precision of the queuing delay estimations are for that device. Highlighting the sensitivity of an AQM scheme to queuing delay estimations helps implementers to identify appropriate means of implementing the mechanism on a device.

11. Operator Control and Auto-tuning

11.1. Motivation

One of the biggest hurdles of RED deployment was/is its parameter sensitivity to operating conditions -- how difficult it is to tune RED parameters for a deployment to achieve acceptable benefit from using RED. Fluctuating congestion levels and network conditions add to the complexity. Incorrect parameter values lead to poor performance.

Any AQM scheme is likely to have parameters whose values affect the control law and behaviour of an AQM. Exposing all these parameters as control parameters to a network operator (or user) can easily result in a unsafe AQM deployment. Unexpected AQM behavior ensues when parameter values are set improperly. A minimal number of control parameters minimizes the number of ways a possibly naive user can break a system where an AQM scheme is deployed at. Fewer control parameters make the AQM scheme more user-friendly and easier to deploy and debug.

[I-D.ietf-aqm-recommendation] states "AQM algorithms SHOULD NOT require tuning of initial or configuration parameters in common use cases." A scheme ought to expose only those parameters that control the macroscopic AQM behavior such as queue delay threshold, queue length threshold, etc.

Additionally, the safety of an AQM scheme is directly related to its stability under varying operating conditions such as varying traffic profiles and fluctuating network conditions, as described in Section 7. Operating conditions vary often and hence the AQM needs to remain stable under these conditions without the need for additional external tuning. If AQM parameters require tuning under

these conditions, then the AQM must self-adapt necessary parameter values by employing auto-tuning techniques.

11.2. Required discussion

AQM proposals SHOULD describe the parameters that control the macroscopic AQM behavior, and identify any parameters that require require tuning to operational conditions. It could be interesting to also discuss that even if an AQM scheme may not adequatly auto-tune its parameters, the resulting performance may not be optimal, but close to something reasonable.

If there are any fixed parameters within the AQM, their setting SHOULD be discussed and justified.

If an AQM scheme is evaluated with parameter(s) that were externally tuned for optimization or other purposes, these values MUST be disclosed.

12. Interaction with ECN

Deployed AQM algorithms SHOULD support Explicit Congestion Notification (ECN) as well as loss to signal congestion to endpoints"[I-D.ietf-aqm-recommendation]. The benefits of providing ECN support for an AQM scheme are descibed in [ECN-Benefit].

12.1. Motivation

(ECN) [RFC3168] is an alternative that allows AQM schemes to signal receivers about network congestion that does not use packet drop.

12.2. Recommended discussion

An AQM scheme can support ECN [I-D.ietf-aqm-recommendation], in which case testers MUST discuss and describe the support of ECN.

13. Interaction with Scheduling

A network device may use per-flow or per-class queuing with a scheduling algorithm to either prioritize certain applications or classes of traffic, limit the rate of transmission, or to provide isolation between different traffic flows within a common class [I-D.ietf-aqm-recommendation].

13.1. Motivation

Coupled with an AQM scheme, a router may schedule the transmission of packets in a specific manner by introducing a scheduling scheme. This algorithm may create sub-queues and integrate a dropping policy on each of these sub-queues. Another scheduling policy may modify the way packets are sequenced, modifying the timestamp of each packet.

13.2. Recommended discussion

The scheduling and the AQM conjointly impact on the end-to-end performance. During the characterization process of a dropping policy, the tester MUST discuss the feasibility to add scheduling combined with the AQM algorithm. This discussion as an instance, MAY explain whether the dropping policy is applied when packets are being enqueued or dequeued.

13.3. Assessing the interaction between AQM and scheduling

These guidelines do not propose guidelines to assess the performance of scheduling algorithms. Indeed, as opposed to characterizing AQM schemes that is related to their capacity to control the queuing delay in a queue, characterizing scheduling schemes is related to the scheduling itself and its interaction with the AQM scheme. As one example, the scheduler may create sub-queues and the AQM scheme may be applied on each of the sub-queues, and/or the AQM could be applied on the whole queue. Also, schedulers might, such as FQ-CoDel [FQ-CoDel] or FavorQueue [FAVOUR], introduce flow prioritization. In these cases, specific scenarios should be proposed to ascertain that these scheduler schemes not only helps in tackling the bufferbloat, but also are robust under a wide variety of operating conditions. This is out of the scope of this document that focus on dropping and/or marking AQM schemes.

14. Discussion on Methodology, Metrics, AQM Comparisons and Packet Sizes

14.1. Methodology

One key objective behind formulating the guidelines is to help ascertain whether a specific AQM is not only better than drop-tail but also safe to deploy. Testers therefore need to provide a reference document for their proposal discussing performance and deployment compared to those of drop-tail.

A description of each test setup SHOULD be detailed to allow this test to be compared with other tests. This also allows others to

replicate the tests if needed. This test setup SHOULD detail software and hardware versions. The tester could make its data available.

The proposals SHOULD be evaluated on real-life systems, or they MAY be evaluated with event-driven simulations (such as ns-2, ns-3, OMNET, etc). The proposed scenarios are not bound to a particular evaluation toolset.

The tester is encouraged to make the detailed test setup and the results publicly available.

14.2. Comments on metrics measurement

The document presents the end-to-end metrics that ought to be used to evaluate the trade-off between latency and goodput in Section 2. In addition to the end-to-end metrics, the queue-level metrics (normally collected at the device operating the AQM) provide a better understanding of the AQM behavior under study and the impact of its internal parameters. Whenever it is possible (e.g. depending on the features provided by the hardware/software), these guidelines advice to consider queue-level metrics, such as link utilization, queuing delay, queue size or packet drop/mark statistics in addition to the AQM-specific parameters. However, the evaluation MUST be primarily based on externally observed end-to-end metrics.

These guidelines do not aim to detail on the way these metrics can be measured, since the way these metrics are measured is expected to depend on the evaluation toolset.

14.3. Comparing AQM schemes

This document recognizes that the guidelines mentioned above may be used for comparing AQM schemes.

AQM schemes need to be compared against both performance and deployment categories. In addition, this section details how best to achieve a fair comparison of AQM schemes by avoiding certain pitfalls.

14.3.1. Performance comparison

AQM schemes MUST be compared against all the generic scenarios presented in this memo. AQM schemes MAY be compared for specific network environments such as data centers, home networks, etc. If an AQM scheme has parameter(s) that were externally tuned for optimization or other purposes, these values MUST be disclosed.

AQM schemes belong to different varieties such as queue-length based schemes (ex. RED) or queueing-delay based scheme (ex. CoDel, PIE). AQM schemes expose different control knobs associated with different semantics. For example, while both PIE and CoDel are queueing-delay based schemes and each expose a knob to control the queueing delay -- PIE's "queueing delay reference" vs. CoDel's "queueing delay target", the two tuning paramerters of the two schemes have different semantics, resulting in different control points. Such differences in AQM schemes can be easily overlooked while making comparisons.

This document RECOMMENDS the following procedures for a fair performance comparison between the AQM schemes:

1. comparable control parameters and comparable input values: carefully identify the set of parameters that control similar behavior between the two AQM schemes and ensure these parameters have comparable input values. For example, to compare how well a queue-length based AQM scheme controls queueing delay vs. a queueing-delay based AQM scheme, a tester can identify the parameters of the schemes that control queue delay and ensure that their input values are comparable. Similarly, to compare how well two AQM schemes accommodate packet bursts, the tester can identify burst-related control parameters and ensure they are configured with similar values.

2. compare over a range of input configurations: there could be situations when the set of control parameters that affect a specific behavior have different semantics between the two AQM schemes. As mentioned above, PIE has tuning parameters to control queue delay that has a different semantics from those used in CoDel. In such situations, these schemes need to be compared over a range of input configurations. For example, compare PIE vs. CoDel over the range of target delay input configurations.

14.3.2. Deployment comparison

AQM schemes MUST be compared against deployment criteria such as the parameter sensitivity (Section 7.3), auto-tuning (Section 11) or implementation cost (Section 10).

14.4. Packet sizes and congestion notification

An AQM scheme may be considering packet sizes while generating congestion signals. [RFC7141] discusses the motivations behind this. For example, control packets such as DNS requests/responses, TCP SYNs/ACKs are small, but their loss can severely impact the application performance. An AQM scheme may therefore be biased

towards small packets by dropping them with smaller probability compared to larger packets. However, such an AQM scheme is unfair to data senders generating larger packets. Data senders, malicious or otherwise, are motivated to take advantage of such AQM scheme by transmitting smaller packets, and could result in unsafe deployments and unhealthy transport and/or application designs.

An AQM scheme SHOULD adhere to the recommendations outlined in [RFC7141], and SHOULD NOT provide undue advantage to flows with smaller packets [I-D.ietf-aqm-recommendation].

15. Conclusion

Figure 4 lists the scenarios and their requirements.

Scenario	Sec.	Requirement
Transport Protocols	4.	
TCP-friendly sender	4.1	Scenario MUST be considered
Aggressive sender	4.2	Scenario MUST be considered
Unresponsive sender	4.3	Scenario MUST be considered
LBE sender	4.4	Scenario MAY be considered
Round Trip Time Fairness	5.2	Scenario MUST be considered
Burst Absorption	6.2	Scenario MUST be considered
Stability	7.	
Varying congestion levels	7.2.5	Scenario MUST be considered
Varying available capacity	7.2.6	Scenario MUST be considered
Parameters and stability	7.3	This SHOULD be discussed
Various Traffic Profiles	8.	
Traffic mix	8.1	Scenario is RECOMMENDED
Bi-directional traffic	8.2	Scenario MAY be considered
Multi-AQM	9.2	Scenario MAY be considered
Implementation Cost	10.2	Pseudo-code SHOULD be provided
Operator Control	11.2	Tuning SHOULD NOT be required
Interaction with ECN	12.2	MUST be discussed if supported
Interaction with Scheduling	13.2	Feasibility MUST be discussed

Figure 4: Summary of the scenarios and their requirements

16. Acknowledgements

This work has been partially supported by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700).

17. Contributors

Many thanks to S. Akhtar, A.B. Bagayoko, F. Baker, D. Collier-Brown, G. Fairhurst, J. Gettys, T. Hoiland-Jorgensen, C.

Kulatunga, W. Lautenschlager, A.C. Morton, R. Pan, D. Taht and M. Welzl for detailed and wise feedback on this document.

18. IANA Considerations

This memo includes no request to IANA.

19. Security Considerations

Some security considerations for AQM are identified in [I-D.ietf-aqm-recommendation].This document, by itself, presents no new privacy nor security issues.

20. References

20.1. Normative References

[I-D.ietf-aqm-recommendation]
Baker, F. and G. Fairhurst, "IETF Recommendations Regarding Active Queue Management", draft-ietf-aqm-recommendation-11 (work in progress), February 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, 1997.

[RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", RFC 7141, 2014.

20.2. Informative References

[BB2011] "BufferBloat: what's wrong with the internet?", ACM Queue vol. 9, 2011.

[CODEL] Nichols, K. and V. Jacobson, "Controlling Queue Delay", ACM Queue , 2012.

[ECN-Benefit]
Welzl, M. and G. Fairhurst, "The Benefits to Applications of using Explicit Congestion Notification (ECN)", IETF (Work-in-Progress) , February 2014.

[FAVOUR] Anelli, P., Diana, R., and E. Lochin, "FavorQueue: a Parameterless Active Queue Management to Improve TCP Traffic Performance", Computer Networks vol. 60, 2014.

[FQ-CoDel]
Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "FlowQueue-Codel", IETF (Work-in-Progress) , January 2015.

[LBE-AQM] Gong, Y., Rossi, D., Testa, C., Valenti, S., and D. Taht, "Fighting the bufferbloat: on the coexistence of AQM and low priority congestion control", Computer Networks, Elsevier, 2014, 60, pp.115 - 128 , 2014.

[LEDBAT-PARAM]
Trang, S., Kuhn, N., Lochin, E., Baudoin, C., Dubois, E., and P. Gelard, "On The Existence Of Optimal LEDBAT Parameters", IEEE ICC 2014 - Communication QoS, Reliability and Modeling Symposium , 2014.

[LOSS-SYNCH-MET-08]
Hassayoun, S. and D. Ros, "Loss Synchronization and Router Buffer Sizing with High-Speed Versions of TCP", IEEE INFOCOM Workshops , 2008.

[PIE] Pan, R., Natarajan, P., Piglione, C., Prabhu, MS., Subramanian, V., Baker, F., and B. VerSteeg, "PIE: A lightweight control scheme to address the bufferbloat problem", IEEE HPSR , 2013.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[RFC2488] Allman, M., Glover, D., and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", BCP 28, RFC 2488, January 1999.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.

[RFC2647] Newman, D., "Benchmarking Terminology for Firewall Performance", RFC 2647, August 1999.

[RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

[RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, March 2009.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

[RFC6297] Welzl, M. and D. Ros, "A Survey of Lower-than-Best-Effort Transport Protocols", RFC 6297, June 2011.

[RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, December 2012.

[SCL-TCP] Morris, R., "Scalable TCP congestion control", IEEE INFOCOM , 2000.

[TCPEVAL2013] Hayes, D., Ros, D., Andrew, L., and S. Floyd, "Common TCP Evaluation Suite", IRTF ICCRG , 2013.

[YU06] Jay, P., Fu, Q., and G. Armitage, "A preliminary analysis of loss synchronisation between concurrent TCP flows", Australian Telecommunication Networks and Application Conference (ATNAC) , 2006.

Authors' Addresses

Nicolas Kuhn (editor)
Telecom Bretagne
2 rue de la Chataigneraie
Cesson-Sevigne 35510
France

Phone: +33 2 99 12 70 46
Email: nicolas.kuhn@telecom-bretagne.eu

Naeem Khademi (editor)
University of Oslo
Department of Informatics, PO Box 1080 Blindern
N-0316 Oslo
Norway

Phone: +47 2285 24 93
Email: naeemk@ifi.uio.no

Preethi Natarajan (editor)
Cisco Systems
510 McCarthy Blvd
Milpitas, California
United States

Email: prenatar@cisco.com

David Ros
Simula Research Laboratory AS
P.O. Box 134
Lysaker, 1325
Norway

Phone: +33 299 25 21 21
Email: dros@simula.no