Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: March 22, 2015

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

September 18, 2014

AQM Characterization Guidelines
draft-ietf-aqm-eval-guidelines-00

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), optionally in combination with a packet scheduling scheme such as fair queuing. The IETF AQM and packet scheduling working group was formed to standardize AQM schemes that are robust, easily implemented, and successfully deployed 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

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Guidelines for AQM designers 4
1.2. Reducing the latency and maximizing the goodput 5
1.3. Glossary 5
1.4. Requirements Language 6
2. End-to-end metrics 6
2.1. Flow Completion time 6
2.2. Packet loss 6
2.3. Packet loss synchronization 7
2.4. Goodput 7
2.5. Latency and jitter 8
2.6. Discussion on the trade-off between latency and goodput . 8
3. Generic set up for evaluations 8
3.1. Topology and notations 8
3.2. Buffer size 10
3.3. Congestion controls 10
4. Various TCP variants 10
4.1. TCP-friendly Sender 11
4.2. Aggressive Transport Sender 11
4.3. Unresponsive Transport Sender 11
4.4. TCP initial congestion window 12
4.5. Traffic Mix 13
5. RTT fairness 14
5.1. Motivation 14
5.2. Required tests 14
5.3. Metrics to evaluate the RTT fairness 14
6. Burst absorption 15
6.1. Motivation 15
6.2. Required tests 15
7. Stability 16
7.1. Motivation 16

7.2. Required tests 17
7.2.1. Mild Congestion 17
7.2.2. Medium Congestion 17
7.2.3. Heavy Congestion 17
7.2.4. Varying congestion levels 17
7.2.5. Varying Available Bandwidth 18
7.3. Parameter sensitivity and stability analysis 18
8. Implementation cost 19
8.1. Motivation 19
8.2. Required discussion 19
9. Operator control knobs and auto-tuning 19
10. Interaction with ECN 20
10.1. Motivation 20
10.2. Required discussion 20
11. Interaction with scheduling 20
11.1. Motivation 20
11.2. Required discussion 21
12. Discussion on methodology, metrics, AQM comparisons and packet sizes 21
12.1. Methodology 21
12.2. Comments on metrics measurement 21
12.3. Comparing AQM schemes 21
12.3.1. Performance comparison 22
12.3.2. Deployment comparison 22
12.4. Packet sizes and congestion notification 23
13. Acknowledgements 23
14. Contributors 23
15. IANA Considerations 23
16. Security Considerations 23
17. References 23
17.1. Normative References 23
17.2. Informative References 24
Authors' Addresses 25

1. Introduction

Active Queue Management (AQM) addresses the concerns arising from using unnecessarily large and unmanaged buffers in order to improve network and application performance. Several AQM algorithms have been proposed in the past years, most notable being 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 is 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, any potential benefits of the RED AQM 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 is 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 may 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.

In order to meet mostly throughput-based SLA requirements and to avoid packet drops, many home gateway manufacturers resort to increasing the available memory beyond "reasonable values". This increase is also referred to as Bufferbloat [BB2011]. Deploying large unmanaged buffers on the Internet has lead to the increase in end-to-end delay, resulting 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, produces problems even with commonly used web services. Active queue management is thus essential to control queuing delay and decrease network latency.

The AQM and Packet Scheduling working group 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 that 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.

1.1. Guidelines for AQM designers

One of the key objectives behind formulating the guidelines is to help ascertain whether a specific AQM is not only better than drop-tail but also safe to deploy. The guidelines help to quantify AQM schemes' performance in terms of latency reduction, goodput

maximization and the trade-off between the two. The guidelines also help to discuss AQM's safe deployment, including self adaptation, stability analysis, fairness, design/implementation complexity and robustness to different operating conditions.

This memo details generic characterization scenarios that any AQM proposal MUST be evaluated against. Irrespective of whether or not an AQM is standardized by the WG, we recommend the relevant scenarios and metrics discussed in this document to be considered. This document presents central aspects of an AQM algorithm that MUST be considered whatever the context is, such as burst absorption capacity, RTT fairness or resilience to fluctuating network conditions. These guidelines could 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. These guidelines do not present context dependent scenarios (such as Wi-Fi, data-centers or rural broadband).

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.

1.2. 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 should aim at both maximizing goodput and minimizing latency. This document proposes guidelines that enable the reader to quantify (1) reduction of latency, (2) maximization of goodput and (3) the trade-off between the two.

Testers SHOULD discuss in a reference document the performance of their proposal in terms of performance and deployment in regards with those of drop-tail: basically, these guidelines provide the tools to understand the deployment costs versus the potential gain in performance of the introduction of the proposed scheme.

1.3. Glossary

- AQM: there may be confusion whether a scheduling scheme is added to an AQM or is a part of the AQM. The rest of this memo refers to AQM as a dropping 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.

1.4. 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].

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 in the path. AQM algorithms 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 the two. These metrics SHOULD be considered to better assess the performance of an AQM scheme.

The metrics listed in this section are not necessarily suited to every type of traffic detailed in the rest of this document. It is therefore NOT REQUIRED to measure all of following metrics.

2.1. Flow Completion time

The flow completion time is an important performance metric for the end user. Considering the fact that an AQM scheme may drop packets, the flow completion time is directly linked to the dropping policy of the AQM scheme. This metric helps to better assess the performance of an AQM depending on the flow size.

2.2. Packet loss

Packet losses, that may occur in a queue, impact on the end-to-end performance at the receiver's side.

The tester MUST evaluate, at the receiver:

- the packet loss probability: this metric should be frequently measured during the experiment as the long term loss probability is of interests for steady state scenarios only;

- the interval between consecutive losses: the time between two losses should be measured. From the set of interval times, the tester should present the median value, the minimum and maximum values and the 10th and 90th percentiles.

2.3. Packet loss synchronization

One goal of an AQM algorithm should be to help with avoiding global synchronization of flows going through the bottleneck buffer on which the AQM operates ([RFC2309]). It is therefore important to assess the "degree" of packet-loss synchronization between flows, 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 may be imposed by practical considerations (e.g., is there 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].

2.4. Goodput

Measuring the goodput enables an end-to-end appreciation of how well the AQM improves transport and application performance. The measured end-to-end goodput is linked to the AQM scheme's dropping policy -- the smaller the packet drops, the fewer packets need retransmission, minimizing AQM's impact on transport and application performance. End-to-end goodput values help evaluate the AQM scheme's effectiveness in minimizing packet drops that impact application performance.

The measurement of the goodput let the tester evaluate to which extend the AQM is able to keep an high link utilization. This metric should be obtained frequently during the experiment: the long term goodput makes sense for steady-state scenarios only and may not reflect how the introduction of AQM actually impacts on the link utilization. It is worth pointing out that the fluctuations of this measurement may depend on other things than the introduction of an AQM, such as physical layer losses, fluctuating bandwidths (Wi-Fi), heavy congestion levels or transport layer congestion controls.

2.5. Latency and jitter

The end-to-end latency differs from the queuing delay: it is linked to the network topology and the path characteristics. Moreover, the jitter strongly depends on the traffic and the topology as well. The introduction of an AQM scheme would impact on these metrics and the end-to-end evaluation of performance SHOULD consider them to better assess the AQM schemes.

The guidelines advice that the tester SHOULD determine the minimum, average and maximum measurements for these metrics and the coefficient of variation for their average values as well.

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

The metrics presented in this section MAY be considered, 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 the section 5 of [TCPEVAL2013]. Each of the end-to-end delay and the goodput should be measured every second. From each of this sets of measurements, the 10th and 90th percentile and the median value should be computed. For each scenario, a graph can be generated, with the x-axis could show the end-to-end delay and the y-axis the goodput. This graph provides part of a better understanding (1) of the delay/goodput trade-off for a given congestion control mechanism, and (2) of 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 discuss various assumptions that have been made in the document.

3.1. Topology and notations

```
       +---------+                                        +-----------+
       |senders A|                                        |receivers B|
       +---------+                                        +-----------+

       +--------------+                                +--------------+
       |traffic class1|                                |traffic class1|
       |--------------|                                |--------------|
       | SEN.Flow1.1 +---------+            +-----------+ REC.Flow1.1 |
       |        +     |        |            |          |        +     |
       |        |     |        |            |          |        |     |
       |        +     |        |            |          |        +     |
       | SEN.Flow1.X +-----+   |            |  +--------+ REC.Flow1.X |
       +--------------+    |   |            |  |       +--------------+
            +            +-+---+---+     +--+--+---+            +
            |            |Router L |     |Router R |            |
            |            |---------|     |---------|            |
            |            | AQM     |     |         |            |
            |            | BuffSize|     |         |            |
            |            | (Bsize) +-----+         |            |
            |            +-----+--++     ++-+------+            |
            +                  |  |       | |                   +
       +--------------+        |  |       | |          +--------------+
       |traffic classN|        |  |       | |          |traffic classN|
       |--------------|        |  |       | |          |--------------|
       | SEN.FlowN.1 +---------+  |       | +-----------+ REC.FlowN.1 |
       |        +     |           |       |            |        +     |
       |        |     |           |       |            |        |     |
       |        +     |           |       |            |        +     |
       | SEN.FlowN.Y +------------+       +-------------+ REC.FlowN.Y |
       +--------------+                                +--------------+
```

Figure 1: Topology and notations

Figure 1 is a generic topology where:

- various classes of traffic can be introduced;

- the timing of each flow (i.e., when does each flow start and stop) may be different;

- each class of traffic can consider 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 links are supposed to be asymmetric in terms of bandwidth: the capacity from senders to receivers is higher than the one from receivers to senders.

This topology may not perfectly reflect actual topologies, however, this simple topology is commonly used in the world of simulations and small testbeds. This topology can be considered as adequate to evaluate AQM proposals, such as proposed in [TCPEVAL2013]. The tester should pay attention to the topology that has been used to evaluate the AQM scheme against which he compares his proposal.

3.2. Buffer size

The size of the buffers MAY be carefully set considering the bandwidth-delay product. However, if the context or the application requires a specific buffer size, the tester MUST justify and detail the way the maximum queue depth is set while presenting the results of its evaluation. Indeed, the size of the buffer may impact on the AQM performance and is a dimensioning parameter that will be considered for a fair comparison between AQM proposals.

3.3. Congestion controls

This memo features three kind of congestion controls:

- TCP-friendly congestion controls: a base-line congestion control for this category is TCP New Reno, 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]

Recent transport layer protocols are not mentioned in the following sections, for the sake of simplicity.

4. Various TCP variants

Network and end devices need to be configured with a reasonable amount of buffers in order to absorb transient bursts. In some situations, network providers configure devices with large buffers to avoid packet drops and increase goodput. Transmission Control Protocol (TCP) fills up these unmanaged buffers until the TCP sender receives a signal (packet drop) to cut down the sending rate. The larger the buffer, the higher the buffer occupancy, and therefore the

queuing delay. On the other hand, an efficient AQM scheme sends out early congestion signals to TCP senders so that the queuing delay is brought under control.

Not all applications run over the same flavor of TCP. Variety of senders generate different classes of traffic which may not react to congestion signals (aka unresponsive flows) or may not cut down their sending rate as expected (aka aggressive flows): AQM schemes aim at maintaining the queuing delay under control, which is challenged if blasting traffics are present.

This section provides guidelines to assess the performance of an AQM proposal based on various metrics presented in Section 2 irrespective of traffic profiles involved -- different senders (TCP variants, unresponsive, aggressive), traffic mix with different applications, etc.

4.1. TCP-friendly Sender

This scenario helps to evaluate how an AQM scheme reacts to a TCP-friendly transport sender. A single long-lived, non application limited, TCP New Reno flow transmits 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 2.6 could be generated.

4.2. Aggressive Transport Sender

This scenario helps to evaluate how an AQM scheme reacts to a transport sender whose sending rate is more aggressive than a single TCP-friendly sender. A single long-lived, non application limited, TCP Cubic flow transmits data between sender A and receiver B. Other aggressive congestion control schemes MAY also be considered.

For each flavor of aggressive transport, the graph described in Section 2.6 could be generated.

4.3. Unresponsive Transport Sender

This scenario helps evaluate how an AQM scheme reacts to a transport sender who is not responsive to congestion signals (ECN marks and/or packet drops) from the AQM scheme. Note that faulty transport implementations on end hosts 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 is the following. In order to create a test environment that results in queue build up, we consider 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 2.6 could be generated.

The second scenario is the following. In order to test to which extend the AQM scheme is able to keep responsive fraction under control, this scenario considers a mixture of TCP-friendly and unresponsive traffics. This scenario 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 2.6 COULD be generated.

4.4. TCP initial congestion window

This scenario helps evaluate how an AQM scheme adapts to a traffic mix consisting of TCP flows with different values for the initial congestion window (IW).

For this scenario, we consider two types of flow that MUST be generated between sender A and receiver B:

- a single long-lived non application limited TCP New Reno flow;

- a single long-lived application limited TCP New Reno flow, with an IW set to 3 or 10 packets. The size of the data transmitted MUST be strictly higher than 10 packets and should be lower than 100 packets.

The transmission of both flows must not start simultaneously: a steady state must be achieved before the transmission of the application limited flow. As a result, the transmission of the non application limited flow MUST start before the transmission of the application limited flow.

For each of these scenarios, the graph described in Section 2.6 could be generated for each class of traffic. The completion time of the application limited TCP flow could be measured.

4.5. Traffic Mix

This scenario helps to evaluate how an AQM scheme reacts to a traffic mix consisting of different applications such as bulk transfer, web, voice, video traffic. These testing cases presented in this subsection have been inspired by the table 2 of [DOCSIS2013]:

- Bulk TCP transfer

- Web traffic

- VoIP

- Constant bit rate UDP traffic

- Adaptive video streaming

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

Case	Number of flows				
	VoIP	Webs	CBR	AdaptVid	FTP
I	1	1	0	0	0
II	1	1	0	0	1
III	1	1	0	0	5
IV	1	1	1	0	5
V	1	1	0	1	5

Figure 2: Traffic Mix scenarios

For each of these scenarios, the graph described in Section 2.6 could be generated for each class of traffic. In addition, other metrics such as end-to-end latency, jitter and flow completion time MUST be generated.

5. RTT fairness

5.1. Motivation

The capability 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 congestion controls is impacted and so the capability of AQM schemes to control the queue. It is therefore important to assess the AQM schemes against a set of RTTs (e.g., from 5 ms to 200 ms).

Also, asymmetry in terms of RTT between various paths SHOULD be considered so that the fairness between the flows can be discussed as one may react faster to congestion than another. The introduction of AQM schemes may improve this fairness.

Moreover, introducing an AQM scheme may result in the absence of fairness between the flows, even when the RTTs are identical. This potential lack of fairness SHOULD be evaluated.

5.2. Required tests

The topology that SHOULD be used is detailed in Figure 1:

- to evaluate the inter-RTT fairness, for each run, ten flows divided into two categories. Category I (Flow1.1, ..., Flow1.5) which RTT between sender A and Router L SHOULD be 5ms. Category II (Flow2.1, ..., Flow 2.5) which RTT between sender A and Router L SHOULD be in [5ms;200ms].

- to evaluate the impact of the RTT value on the AQM performance and the intra-protocol fairness, for each run, ten flows (Flow1.1, ..., Flow1.5 and Flow2.1, ..., Flow2.5) SHOULD be introduced. For each experiment, the set of RTT SHOULD be the same for all the flows and in [5ms;200ms].

These flows MUST use the same congestion control algorithm.

5.3. Metrics to evaluate the RTT fairness

The output that MUST be measured is:

- for the inter-RTT fairness: (1) the cumulated average goodput of the flows from Category I, goodput_Cat_I (Section 2.4); (2) the cumulated average goodput of the flows from Category II, goodput_Cat_II (Section 2.4); (3) the ratio goodput_Cat_II/goodput_Cat_I; (4) the average packet drop rate for each category (Section 2.2).

- for the intra-protocol RTT fairness: (1) the cumulated averga goodput of the ten flows (Section 2.4); (2) the average packet drop rate for the ten flows(Section 2.2).

6. Burst absorption

6.1. Motivation

Packet arrivals can be bursty due to various reasons. Dropping one or more packets from a burst may result in performance penalties for the corresponding flows since the dropped packets have to be retransmitted. Performance penalties may turn into unmet SLAs and be disincentives to AQM adoption. Therefore, an AQM scheme SHOULD be designed to accommodate transient bursts. AQM schemes do not present the same tolerance to bursts of packets arriving in the buffer: this tolerance MUST be quantified.

Note that accommodating bursts translates to higher queue length and queuing delay. Naturally, it is important that the AQM scheme brings bursty traffic under control quickly. On the other hand, spiking packet drops in order to bring packet bursts quickly under control could result in multiple drops per flow and severely impact transport and application performance. Therefore, an AQM scheme SHOULD 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 maintains short queues to allow the remaining space in the queue for bursts of packets. The tolerance to bursts of packets depends on 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 may depend on the buffer occupancy or the currently estimated queuing delay. Also, the impact of the buffer size on the burst allowance MAY be evaluated.

6.2. Required tests

For this scenario, the following traffic MUST be generated from sender A to receiver B:

- IW10: TCP transfer with initial congestion window set to 10 of 5MB;

- Bursty video frames;

- Web traffic;

- Constant bit rate UDP traffic.

Figure 3 presents the various cases for the traffic that MUST be generated between sender A and receiver B.

Case	Number of traffic			
	Video	Webs	CBR	Bulk Traffic (IW10)
I	0	1	1	0
II	0	1	1	1
III	1	1	0	0
IV	1	1	1	0
V	1	1	1	1

Figure 3: Bursty traffic scenarios

For each of these scenarios, the graph described in Section 2.6 could be generated. In addition, other metrics such as end-to-end latency, jitter, flow completion time MUST be generated.

7. Stability

7.1. Motivation

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

- Traffic and congestion levels are higher during peak hours than off-peak hours.

- In the presence of scheduler, a queue's draining rate may vary depending on other queues: a low load on a high priority queue implies higher draining rate for lower priority queues.

- The available capacity on the physical layer may vary over time such as in the context of lossy channels.

Whether the target context is a not stable environment, the capability of an AQM scheme to actually maintain its control on the queuing delay and buffer occupancy is challenged. This document

propose guidelines to assess the behaviour of AQM schemes under varying congestion levels and varying draining rates.

7.2. Required tests

7.2.1. Mild Congestion

This scenario helps to evaluate how an AQM scheme reacts to a light load of incoming traffic resulting in mild congestion -- packet drop rates less than 1%. Each single-lived non application limited TCP flow transfers data.

For this scenario, the graph described in Section 2.6 could be generated.

7.2.2. Medium Congestion

This scenario helps to evaluate how an AQM scheme reacts to incoming traffic resulting in medium congestion -- packet drop rates between 1%-3%. Each single-lived non application limited TCP flow transfers data.

For this scenario, the graph described in Section 2.6 could be generated.

7.2.3. Heavy Congestion

This scenario helps to evaluate how an AQM scheme reacts to incoming traffic resulting in heavy congestion -- packet drop rates between 5%-10%. Each single lived non application limited TCP flow transfers data.

For this scenario, the graph described in Section 2.6 could be generated.

7.2.4. Varying congestion levels

This scenario helps 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 according to a large time scale. The following phases may be considered: phase I - mild congestion during 0-5s; phase II - medium congestion during 5-10s; phase III - heavy congestion during 10-15s; phase I again, ... and so on. Each single lived non application limited TCP flow transfers data.

For this scenario, the graph described in Section 2.6 could be generated. Moreover, one graph could be generated for each of the phases previously detailed.

7.2.5. Varying Available Bandwidth

This scenario helps evaluate how an AQM scheme adapts to varying available bandwidth on the outgoing link.

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

- Experiment 1: the capacity varies between two values according to a large time scale. As an example, the following phases may be considered: phase I - 100Mbps during 0-5s; phase II - 10Mbps during 5-10s: phase I again, ... and so on.

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

More realistic fluctuating bandwidth patterns MAY be considered.

The scenario consists of TCP New Reno flows between sender A and receiver B. In order to better assess the impact of draining rates on the AQM behavior, the tester MUST compare its performance with those of drop-tail.

For this scenario, the graph described in Section 2.6 could be generated. Moreover, one graph SHOULD be generated for each of the phases previously detailed.

7.3. Parameter sensitivity and stability analysis

An AQM scheme's control law is the primary means by which the AQM controls queuing delay. Hence understanding the AQM control law is critical to understanding AQM behavior. The AQM's control law may include several input parameters whose values affect the AQM output behavior and 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.

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 other ways to discuss its stability. 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's control law better and the network conditions/deployments under which the AQM is stable.

8. Implementation cost

8.1. Motivation

An AQM's successful deployment is directly related to its ease of implementation. Network devices may need hardware or software implementations of the AQM. 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 (current queuing delay, queue length), computations involved, need for timers etc. This helps identify costs associated with implementing the AQM on a particular hardware or software device. Also, it helps the WG understand which kind of devices can easily support the AQM and which cannot.

8.2. Required 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 queue-delay based AQM scheme requires current queuing delay as input from the device. If the device already maintains this value, then it is trivial to implement the AQM logic on the device. On the other hand, if the device provides indirect means to estimate queuing delay (for example: timestamps, dequeing rate etc.), then the AQM behavior is sensitive to how good the queuing delay estimate turns out on that device. Highlighting the AQM's sensitivity to queuing delay estimate helps implementers identify optimal means of implementing the AQM on a device.

9. Operator control knobs and auto-tuning

One of the biggest hurdles for RED deployment was/is its parameter sensitivity to operating conditions -- how difficult it is to tune important RED parameters for a deployment in order to get maximum benefit from the RED implementation. Fluctuating congestion levels and network conditions add to the complexity. Incorrect parameter

values lead to poor performance. This is one reason why RED is reported to be usually turned off.

Any AQM scheme is likely to have parameters whose values affect the AQM's control law and behavior. Exposing all these parameters as control knobs to a network operator (or user) can easily result in an unsafe AQM deployment. Unexpected AQM behavior ensues when parameter values are not set properly. A minimal number of control knobs minimizes the number of ways a, possible naive, user can break the AQM system. Fewer control knobs make the AQM scheme more user-friendly and easier to deploy and debug.

We recommend that an AQM scheme SHOULD minimize the number of control knobs exposed for operator tuning. An AQM scheme SHOULD expose only those knobs that control the macroscopic AQM behavior such as queue delay threshold, queue length threshold, etc.

Additionally, an AQM scheme's safety 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 it is necessary that the AQM MUST 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.

10. Interaction with ECN

10.1. Motivation

Apart from packet drops, Explicit Congestion Notification (ECN) is an alternative means to signal data senders about network congestion. The AQM recommendation document [I-D.ietf-aqm-recommendation] describes some of the benefits of using ECN with AQM.

10.2. Required discussion

An AQM scheme MAY support ECN, in which case testers MUST discuss and describe the support of ECN.

11. Interaction with scheduling

11.1. 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.

11.2. Required discussion

The scheduling and the AQM conjointly impact on the end-to-end performance. During the characterization process of a dropping policy, the tester MAY discuss the feasibility to add scheduling on top of its algorithm. This discussion MAY detail if the dropping policy is applied while packets are enqueued or dequeued.

12. Discussion on methodology, metrics, AQM comparisons and packet sizes

12.1. Methodology

A sufficiently detailed description of the test setup SHOULD be provided. Indeed, that would allow other to replicate the tests if needed. This test setup MAY include software and hardware versions. The tester MAY make its data available.

The proposals SHOULD be experimented on real 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.

12.2. Comments on metrics measurement

In this document, we present the end-to-end metrics that SHOULD be evaluated to evaluate the trade-off between latency and goodput. The queue-related metrics enable a better understanding of the AQM behavior under tests and the impact of its internal parameters. Whenever it is possible, these guidelines advice to consider queue-related metrics, such as link utilization, queuing delay, queue size or packet loss.

These guidelines could hardly detail the way the metrics can be measured depends highly on the evaluation toolset.

12.3. Comparing AQM schemes

This memo recognizes that the guidelines mentioned above may be used for comparing AQM schemes. This memo recommends that AQM schemes MUST 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.

12.3.1. 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 center, home networks etc. If an AQM scheme's parameter(s) were externally tuned for optimization or other purposes, these values MUST be disclosed.

Note that AQM schemes belong to different varieties such as queue-length based scheme (ex: RED) or queue-delay based scheme (ex: CoDel, PIE). Also, AQM schemes expose different control knobs associated with different semantics. For example, while both PIE and CoDel are queue-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 schemes' knobs 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 of two 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, while comparing how well a queue-length based AQM X controls queueing delay vs. queue-delay based AQM Y, identify the two schemes' parameters that control queue delay and ensure that their input values are comparable. Similarly, to compare two AQM schemes on how well they accommodate bursts, 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's knob to control queue delay has different semantics from CoDel's. In such situations, the schemes MUST be compared over a range of input configurations. For example, compare PIE vs. CoDel over the range of delay input configurations -- 5ms, 10ms, 15ms etc.

12.3.2. Deployment comparison

AQM schemes MUST be compared against deployment criteria such as the parameter sensitivity (Section 7.3), the auto-tuning (Section 9) or the implementation cost (Section 8).

12.4. Packet sizes and congestion notification

An AQM scheme may be considering packet sizes while generating congestion signals. [RFC7141] discusses the motivations behind the same. For example, control packets such as DNS requests/responses, TCP SYNs/ACKs are small, and their loss can severely impact 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 the 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 recommendations outlined in [RFC7141], and SHOULD NOT provide undue advantage to flows with smaller packets.

13. 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).

14. Contributors

Many thanks to S. Akhtar, A.B. Bagayoko, F. Baker, D. Collier-Brown, G. Fairhurst, T. Hoiland-Jorgensen, C. Kulatunga, W. Lautenschlager, R. Pan, D. Taht and M. Welzl for detailed and wise feedback on this document.

15. IANA Considerations

This memo includes no request to IANA.

16. Security Considerations

This document, by itself, presents no new privacy nor security issues.

17. References

17.1. Normative References

[I-D.ietf-aqm-recommendation]
Baker, F. and G. Fairhurst, "IETF Recommendations Regarding Active Queue Management", draft-ietf-aqm-recommendation-01 (work in progress), January 2014.

[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.

17.2. 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.

[DOCSIS2013] White, G. and D. Rice, "Active Queue Management Algorithms for DOCSIS 3.0", Technical report - Cable Television Laboratories , 2013.

[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.

[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.

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

[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.

[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

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

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

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