Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 6, 2015

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

March 5, 2015

AQM Characterization Guidelines
draft-ietf-aqm-eval-guidelines-01

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 September 6, 2015.

Table of Contents

1. Introduction 3
1.1. Guidelines for AQM designers 5
1.2. Reducing the latency and maximizing the goodput 5
1.3. Glossary 6
1.4. Requirements Language 6
2. End-to-end metrics 6
2.1. Flow completion time 6
2.2. Packet loss 7
2.3. Packet loss synchronization 7
2.4. Goodput 8
2.5. Latency and jitter 9
2.6. 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 11
3.3. Congestion controls 11
4. Various TCP variants 12
4.1. TCP-friendly Sender 12
4.2. Aggressive Transport Sender 13
4.3. Unresponsive Transport Sender 13
4.4. TCP initial congestion window 14
4.5. Traffic Mix 14
5. RTT fairness 15
5.1. Motivation 15
5.2. Required tests 15
5.3. Metrics to evaluate the RTT fairness 16
6. Burst absorption 16
6.1. Motivation 16
6.2. Required tests 17
7. Stability 17
7.1. Motivation 18

7.2. Required tests 18
7.2.1. Definition of the congestion Level 18
7.2.2. Mild Congestion 19
7.2.3. Medium Congestion 19
7.2.4. Heavy Congestion 19
7.2.5. Varying congestion levels 19
7.2.6. Varying Available Bandwidth 19
7.3. Parameter sensitivity and stability analysis 20
8. Implementation cost 21
8.1. Motivation 21
8.2. Required discussion 21
9. Operator control knobs and auto-tuning 21
10. Interaction with ECN 22
10.1. Motivation 22
10.2. Required discussion 22
11. Interaction with scheduling 22
11.1. Motivation 22
11.2. Required discussion 23
12. Discussion on methodology, metrics, AQM comparisons and packet sizes 23
12.1. Methodology 23
12.2. Comments on metrics measurement 23
12.3. Comparing AQM schemes 23
12.3.1. Performance comparison 24
12.3.2. Deployment comparison 25
12.4. Packet sizes and congestion notification 25
13. Acknowledgements 25
14. Contributors 25
15. IANA Considerations 25
16. Security Considerations 25
17. References 26
17.1. Normative References 26
17.2. Informative References 26
Authors' Addresses 27

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 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.

In order to meet mostly throughput-based Service-Level Agreement (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'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.

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 these two. The guidelines also help to discuss AQM's safe deployment, including self-adaptation, stability analysis, fairness, design and 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 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. These guidelines do not present context dependent scenarios (such as 802.11 WLANs, data-centers or rural broadband networks).

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 due to the introduction of the proposed scheme.

1.3. Glossary

- AQM: there may be a debate on whether a scheduling scheme is additional to an AQM mechanism or is a part of an AQM scheme. 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.

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 along 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 MAY be used to better quantify (1) the reduction of latency, (2) maximization of goodput and (3) the trade-off between these two. These metrics MAY 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 the following metrics in every scenario discussed in this document necessarily, if the chosen metric is not relevant to the context of the evaluation scenario (e.g. latency vs. goooodput trade-off in application-limited traffic scenarios). The tester SHOULD however measure and report on all the metrics relevant to the context of the evaluation scenario.

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)

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 also be frequently measured during the experiment, since the long-term loss probability is only of interest for steady-state scenarios.

- the interval between consecutive losses: the time between two losses should 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 [RFC2679].

- to keep a count of all packets sent, and a count of the non-dupplicate 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].

2.3. Packet loss synchronization

One goal of an AQM algorithm should be to help with avoiding global synchronization of flows sharing 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. 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.

2.4. 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 enables an 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/marking policy -- e.g. the smaller the number of packet drops, the fewer packets need retransmission, minimizing AQM’s impact 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 in minimizing packet drops that impact application performance and to estimate how well the AQM scheme works with ECN.

The measurement of the goodput let the tester evaluate to which extent the AQM is able to maintain a high link utilization. This metric should be also obtained frequently during the 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. It is worth pointing out that the 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.

2.5. 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 strongly depends on the traffic pattern and the topology as well. The introduction of an AQM scheme would impact on these metrics and therefore they SHOULD be considered in the end-to-end evaluation of performance.

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

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 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

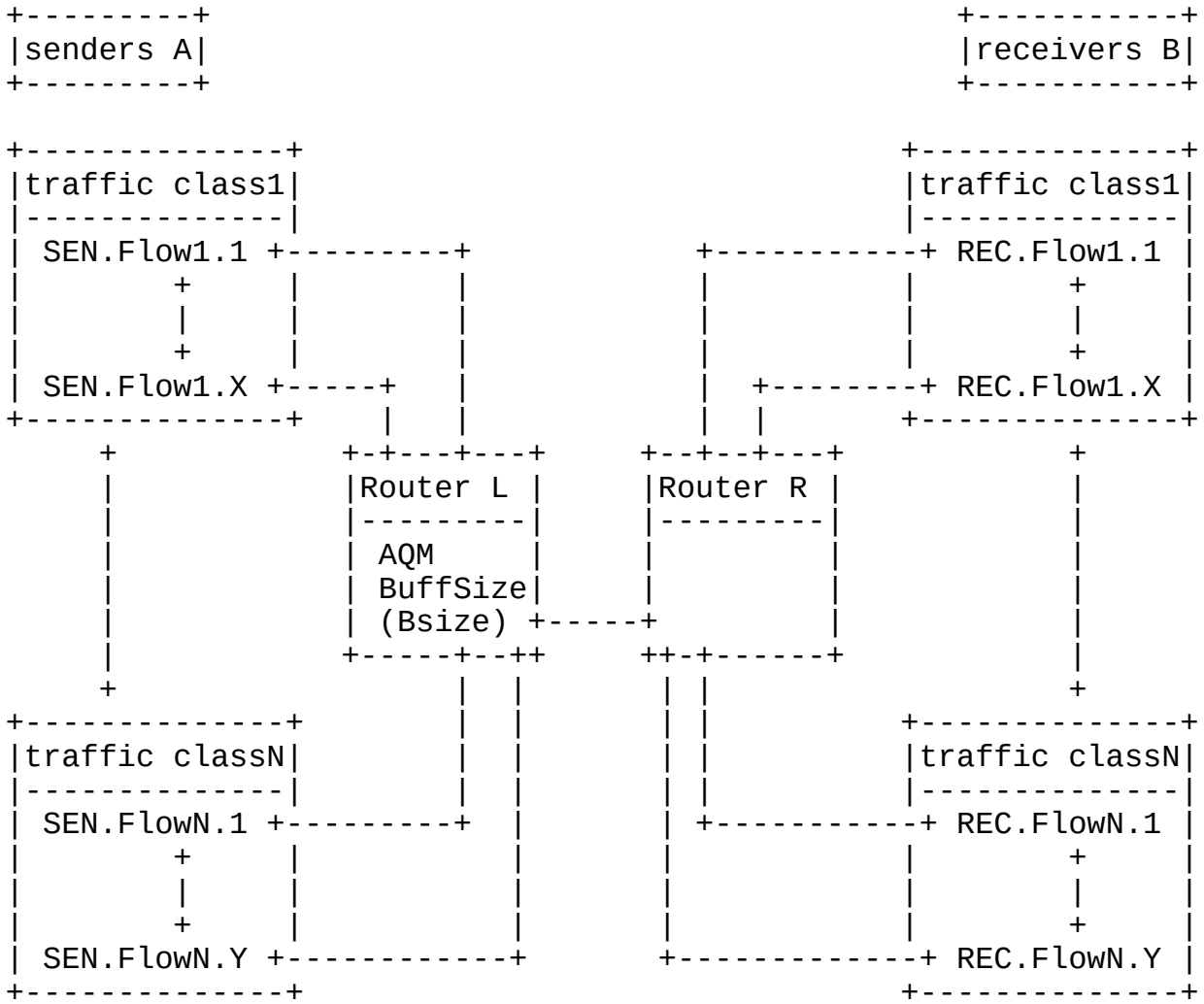

Figure 1: Topology and notations

Figure 1 is a generic topology where:

- various classes of traffic can be introduced;
- the timing of each flow (i.e., when does each flow start and stop) may be different;
- 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 bottleneck link SHOULD be asymmetric in terms of bandwidth: the capacity from senders to receivers is higher than the one from receivers to senders;
- The traffic SHOULD be bi-directional between A and B (downlink and uplink). The Tester MAY additionally evaluate uni-directional traffic scenarios as well (downlink-only or uplink-only).

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, similarly to the topology proposed in [TCPEVAL2013]. The tester should carefully choose the topology that is going to be used to evaluate the AQM scheme.

3.2. Buffer size

The size of the buffers should be carefully chosen, and MAY be set to 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 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]

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 maximum available buffer space in order to absorb transient bursts. In some situations, network providers tend to configure devices with large buffers in order to avoid packet drops triggered by a full buffer and to maximize the link utilization for standard loss-based TCP traffic. Loss-based TCP congestion controls (including standard NewReno TCP) fill up these unmanaged buffers until the TCP sender receives a signal (packet drop) to decrease the sending rate. The larger the buffer is, the higher the buffer occupancy, and therefore the queuing delay. On the other hand, an efficient AQM scheme SHOULD convey early congestion signals to TCP senders so that the average queuing delay is brought under control.

Not all applications run over the same flavor of TCP or even necessarily use TCP. Variety of applications generate different classes of traffic which may not react to congestion signals (a.k.a unresponsive flows) or may not decrease their sending rate as expected (a.k.a aggressive flows); AQM schemes aim at maintaining the queuing delay under control, which is challenged if aggressive or unresponsive traffics are present.

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), 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 NewReno flow 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 2.6 could be generated.

4.2. Aggressive Transport Sender

This scenario helps 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 2.6 could be generated.

4.3. Unresponsive Transport Sender

This scenario helps to evaluate how an AQM scheme reacts to a transport sender that 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 on the path that modify congestion signals in packet headers (e.g. modifying the ECN-related bitsets) [I-D.ietf-aqm-recommendation] may also lead to a similar situation, such that the AQM scheme needs to adapt to the unresponsive traffic. To this end, these guidelines propose the two following scenarios.

The first scenario aims at creating a test environment that results in constant queue build up; we consider unresponsive flow(s) with an overall sending rate that is greater than the bottleneck's link capacity between routers L and R. This scenario consists of a long-lived non application-limited UDP flow that transfers data between sender A and receiver B. Graphs described in Section 2.6 could be generated.

The second scenario aims to test to which extent the AQM scheme is able to keep the responsive fraction of overall traffic load 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 NewReno flow that transfer data between sender A and receiver B. As opposed to the first scenario, the rate of the UDP traffic should be less than or equal to 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 to evaluate how an AQM scheme adapts to a traffic mix consisting of TCP flows with different values of the Initial congestion Window (IW).

For this scenario, we consider two types of flows that 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 2.6 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.

4.5. Traffic Mix

This scenario helps 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 trafic mixes can be considered. These guidelines RECOMMEND to examine at least the following example: 1 bi-directionnal VoIP; 6 Webs; 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 2.6 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 reported.

5. RTT fairness

5.1. Motivation

The capability of AQM schemes to control the queuing delay highly depends on the way end-to-end transport protocols react to congestion signals. When network path's intrinsic RTT varies, the behaviour of congestion control is impacted and so the capability of AQM schemes to control the queueing level. It is therefore important to assess the AQM schemes against a set of intrinsic RTTs common in the Internet transfers (e.g. from 5 ms to 500 ms).

Also, asymmetry in terms of difference in intrinsic RTT between various paths sharing the same bottleneck SHOULD be considered and the fairness between the flows SHOULD 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.

Moreover, 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. Required tests

The topology that SHOULD be used is presented 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].

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 cumulative average goodput of the flow from Category I, goodput_Cat_I (Section 2.4); (2) the cumulative average goodput of the flow 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 cumulative average goodput of the two flows (Section 2.4); (2) the average packet drop rate for the two flows (Section 2.2).

6. Burst absorption

6.1. Motivation

Packet arrivals can be bursty due to various reasons. A packet burst can push the AQM schemes to drop/mark packets momentarily even though the average queue length may still be below the AQM's target queuing thresholds. Dropping/marking one or more packets within a burst may result in performance penalties for the corresponding flows since the dropped/marked packets cause unnecessary rate reduction by congestion control as well as retransmission in case of drop only. Performance penalties may turn into unmet SLAs and become disincentives for the AQM adoption. Therefore, an AQM scheme SHOULD be designed to accommodate transient bursts. AQM schemes do not present the same tolerance to packet bursts arriving at the buffer, therefore 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 average queues to allow the remaining space in the queue for temporary bursts of packets. The tolerance to packet bursts 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 such feature (a.k.a burst allowance) MAY be evaluated.

6.2. Required tests

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

- Web traffic with IW10: Web transfer of 100 packets with initial congestion window set to 10;
- 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

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. 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 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 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 maintain its control over the queuing delay and buffer occupancy is challenged. This document proposes guidelines to assess the behaviour of AQM schemes under varying congestion levels and varying draining rates.

7.2. Required 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 2.6 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 capacity and maximum buffer size based on Eq. 3 of [SCL-TCP]:

$$l_r = 0.76 * N^2 / S^2$$

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

7.2.2. 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 around 1%. The number of bulk flows required to achieve this congestion level, N_mild, is then:

N_mild = round(0.114*S)

7.2.3. Medium Congestion

This scenario helps to evaluate how an AQM scheme reacts to incoming traffic resulting in medium congestion -- packet drop rates around 5%. The number of bulk flows required to achieve this congestion level, N_med, is then:

N_med = round (0.256*S)

7.2.4. Heavy Congestion

This scenario helps to evaluate how an AQM scheme reacts to incoming traffic resulting in heavy congestion -- packet drop rates around 10%. The number of bulk flows required to achieve this congestion level, N_heavy, is then:

N_heavy = round (0.363*S)

7.2.5. 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 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 Bandwidth

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

To emulate 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 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 bandwidth 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 consists of TCP NewReno 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.

7.3. Parameter sensitivity and stability analysis

An AQM scheme's control law is the primary means by which the queuing delay is controlled. Hence understanding the control law is critical to understanding the AQM scheme's behavior. The control law may 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.

AQM proposals SHOULD provide background material showing control theoretic analysis of the 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 scheme's control law better and the network conditions/deployments under which the AQM scheme is performing stably.

8. Implementation cost

8.1. Motivation

An AQM scheme's successful deployment is directly related to its cost of implementation. Network devices may 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. Also, it helps the WG to 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 queueing-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 the queuing delay (for example: timestamps, dequeing rate etc), then the AQM behavior is sensitive to how accurate enough the queuing delay estimations are on that device. Highlighting the AQM scheme's sensitivity to queuing delay estimations helps implementers to identify optimal means of implementing the mechanism on the device.

9. Operator control knobs and auto-tuning

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 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 by the network operators.

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 a unsafe AQM deployment. Unexpected AQM behavior ensues when parameter

values are set improperly. A minimal number of control knobs minimizes the number of ways a possibly naive user can break a system where an AQM scheme is deployed at. Fewer control knobs make the AQM scheme more user-friendly and easier to deploy and debug.

We highly recommend that an AQM scheme SHOULD minimize the number of control knobs exposed for the operator's tuning. An AQM scheme SHOULD expose only those knobs that control the macroscopic AQM behavior such as queue delay threshold or queue length threshold and so on.

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 scheme 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 mean to signal the data senders about network congestion. The AQM recommendation document [I-D.ietf-aqm-recommendation] describes some of the benefits of using ECN coupled with an AQM mechanism.

10.2. Required discussion

An AQM scheme SHOULD support ECN and the 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 to its algorithm. This discussion as an instance, MAY explain whether the dropping policy is applied when packets are being enqueued or dequeued.

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

12.1. Methodology

A sufficiently detailed description of the test setup MUST be provided which facilitates other testers to replicate the tests if required. The test setup MAY include software and hardware specifications and versions. The tester is encouraged to make the detailed test setup and the results publicly available.

The proposals SHOULD be experimented 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.

12.2. Comments on metrics measurement

In this document, we presented the end-to-end metrics that SHOULD 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 RECOMMEND to collect 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 they highly depend on the evaluation toolset and/or hadrware.

12.3. Comparing AQM schemes

This memo recognizes that the guidelines mentioned above may be used for comparing AQM schemes. It 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 centers, 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 schemes such as RED or queueing-delay based scheme such as CoDel and PIE. Also, 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 schemes' knobs have different semantics resulting in different control points. Such differences in AQM schemes SHOULD not be 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, while comparing how well a queue-length based AQM scheme controls queueing delay vs. a queueing-delay based AQM scheme, identify the two schemes' parameters that control queueing delay and ensure that their input values are comparable. Similarly, to compare two AQM schemes on how well they accommodate packet 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 queueing delay has different semantics from CoDel's. In such situations, these schemes MUST be compared over a range of input configurations. For example, compare PIE vs. CoDel over the range of target delay input configurations.

12.3.2. Deployment comparison

AQM schemes MUST be compared against deployment criteria such as the parameter sensitivity (Section 7.3), auto-tuning (Section 9) or 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 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 disproportionate 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, A.C. Morton, 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-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.

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.

[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.

[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.

[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.

[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

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

Email: prenatar@cisco.com

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

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

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