Internet Engineering Task Force (IETF)
Request for Comments: 7928
Category: Informational
ISSN: 2070-1721

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

July 2016

Characterization Guidelines for Active Queue Management (AQM)

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. This document describes various criteria for performing characterizations of AQM schemes that can be used in lab testing during development, prior to deployment.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7928.

Table of Contents

1. Introduction

Active Queue Management (AQM) addresses the concerns arising from using unnecessarily large and unmanaged buffers to improve network and application performance, such as those presented in Section 1.2 of the AQM recommendations document [RFC7567]. Several AQM algorithms have been proposed in the past years, most notably Random Early Detection (RED) [FLOY1993], BLUE [FENG2002], Proportional Integral controller (PI) [HOLLO2001], and more recently, Controlled Delay (CoDel) [CODEL] and Proportional Integral controller Enhanced (PIE) [AQMPIE]. 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 buffer occupancy, and therefore the end-to-end delay. 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.

A buffer is a physical volume of memory in which a queue or set of queues are stored. When speaking of a specific queue in this document, "buffer occupancy" refers to the amount of data (measured in bytes or packets) that are in the queue, and the "maximum buffer size" refers to the maximum buffer occupancy. In switches and routers, a global memory space is often shared between the available interfaces, and thus, the maximum buffer size for any given interface may vary over time.

Bufferbloat [BB2011] is the consequence of deploying large, unmanaged buffers on the Internet -- the buffering has often been measured to be ten times or a hundred times larger than needed. Large buffer sizes in combination with TCP and/or unresponsive flows increases end-to-end delay. This results in poor performance for latency-sensitive applications such as real-time multimedia (e.g., voice, video, gaming, etc.). The degree to which this affects modern networking equipment, especially consumer-grade equipment, 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 chartered 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 are also robust under a wide variety of operating

conditions. This document provides characterization guidelines that can be used to assess the applicability, performance, and deployability of an AQM, whether it is a candidate for standardization at IETF or not.

The AQM algorithm implemented in a router can be separated from the scheduling of packets sent out by the router as discussed in the AQM recommendations document [RFC7567]. The rest of this memo refers to the AQM as a dropping/marking policy as a separate feature to any interface-scheduling scheme. This document may be complemented with another one on guidelines for assessing the combination of packet scheduling and AQM. We note that such a document will inherit all the guidelines from this document, plus any additional scenarios relevant for packet scheduling such as flow-starvation evaluation or the impact of the number of hash buckets.

1.1. Reducing the Latency and Maximizing the Goodput

The trade-off between reducing the latency and maximizing the goodput is intrinsically linked to each AQM scheme and is key to evaluating its performance. To ensure the safety deployment of an AQM, its behavior should be assessed in a variety of scenarios. Whenever possible, solutions ought to aim at both maximizing goodput and minimizing latency.

1.2. Goals of This Document

This document recommends a generic list of scenarios against which an AQM proposal should be evaluated, considering both potential performance gain and safety of deployment. The guidelines help to quantify performance of AQM schemes in terms of latency reduction, goodput maximization, and the trade-off between these two. The document presents central aspects of an AQM algorithm that should be considered, whatever the context, such as burst absorption capacity, RTT fairness, or resilience to fluctuating network conditions. The guidelines also discuss methods to understand the various aspects associated with safely deploying and operating the AQM scheme. Thus, one of the key objectives behind formulating the guidelines is to help ascertain whether a specific AQM is not only better than drop-tail (i.e., without AQM and with a BDP-sized buffer), but also safe to deploy: the guidelines can be used to compare several AQM proposals with each other, but should be used to compare a proposal with drop-tail.

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

presents central aspects of an AQM algorithm that should be considered whatever the context, such as burst absorption capacity, RTT fairness, or resilience to fluctuating network conditions.

These guidelines do not define and are not bound to a particular deployment scenario or evaluation toolset. Instead, the guidelines can be used to assert the potential gain of introducing an AQM for the particular environment, which is of interest to the testers. These guidelines do not cover every possible aspect of a particular algorithm. These guidelines do not present context-dependent scenarios (such as IEEE 802.11 WLANs, data centers, or rural broadband networks). To keep the guidelines generic, a number of potential router components and algorithms (such as Diffserv) are omitted.

The goals of this document can thus be summarized as follows:

- The present characterization guidelines provide a non-exhaustive list of scenarios to help ascertain whether an AQM is not only better than drop-tail (with a BDP-sized buffer), but also safe to deploy; the guidelines can also be used to compare several AQM proposals with each other.

- The present characterization guidelines (1) are not bound to a particular evaluation toolset and (2) can be used for various deployment contexts; testers are free to select a toolset that is best suited for the environment in which their proposal will be deployed.

- The present characterization guidelines are intended to provide guidance for better selecting an AQM for a specific environment; it is not required that an AQM proposal is evaluated following these guidelines for its standardization.

1.3. Requirements Language

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

1.4. Glossary

- Application-limited traffic: A type of traffic that does not have an unlimited amount of data to transmit.

- AQM: The Active Queue Management (AQM) algorithm implemented in a router can be separated from the scheduling of packets sent by the router. The rest of this memo refers to the AQM as a dropping/marking policy as a separate feature to any interface scheduling scheme [RFC7567].

- BDP: Bandwidth Delay Product.

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

- Buffer Occupancy: The amount of data stored in a buffer, measured in bytes or packets.

- Buffer Size: The maximum buffer occupancy, that is the maximum amount of data that may be stored in a buffer, measured in bytes or packets.

- Initial Window 10 (IW10): TCP initial congestion window set to 10 packets.

- Latency: One-way delay of packets across Internet paths. This definition suits transport layer definition of the latency, which should not be confused with an application-layer view of the latency.

- Goodput: Goodput is defined as the number of bits per unit of time forwarded to the correct destination, minus any bits lost or retransmitted [RFC2647]. The goodput should be determined for each flow and not for aggregates of flows.

- SQRT: The square root function.

- ROUND: The round function.

2. End-to-End Metrics

End-to-end delay is the result of propagation delay, serialization delay, service delay in a switch, medium-access delay, and queuing delay, summed over the network elements along the path. AQM schemes may reduce the queuing delay by providing signals to the sender on the emergence of congestion, but any impact on the goodput must be carefully considered. This section presents the metrics that could

be used to better quantify (1) the reduction of latency, (2) maximization of goodput, and (3) the trade-off between these two. This section provides normative requirements for metrics that can be used to assess the performance of an AQM scheme.

Some metrics listed in this section are not suited to every type of traffic detailed in the rest of this document. It is therefore not necessary to measure all of the following metrics: the chosen metric may not be relevant to the context of the evaluation scenario (e.g., latency vs. goodput trade-off in application-limited traffic scenarios). Guidance is provided for each metric.

2.1. Flow Completion Time

The flow completion time is an important performance metric for the end-user when the flow size is finite. The definition of the flow size may be a source of contradictions, thus, this metric can consider a flow as a single file. 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 [byte] / (G [bit/s] / 8 [bit/byte])

Where flow size is the size of the transport-layer payload in bits and goodput is the transport-layer payload transfer time (described in Section 2.5).

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

2.2. Flow Startup Time

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

2.3. Packet Loss

Packet loss can occur en route, this can impact the end-to-end performance measured at the receiver end.

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

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

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

The packet loss ratio 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 advise the tester:

- To check that for every packet, a corresponding packet was received within a reasonable time, as presented in the document that proposes a metric for one-way packet loss across Internet paths [RFC7680].

- To keep a count of all packets sent, and a count of the non-duplicate packets received, as discussed in [RFC2544], which presents a benchmarking methodology.

The interval between consecutive losses, which is also called a "gap", is a metric of interest for Voice over IP (VoIP) traffic [RFC3611].

2.4. Packet Loss Synchronization

One goal of an AQM algorithm is to help to avoid global synchronization of flows sharing a bottleneck buffer on which the AQM operates ([RFC2309] and [RFC7567]). The "degree" of packet-loss synchronization between flows should be assessed, with and without the AQM under consideration.

Loss synchronization among flows may be quantified by several slightly different metrics that capture different aspects of the same issue [HASS2008]. However, in real-world measurements the choice of metric could be imposed by practical considerations -- e.g., whether fine-grained information on packet losses at the bottleneck is 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. This metric can be used in real-world experiments to characterize synchronization along arbitrary Internet paths [JAY2006].

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

2.5. Goodput

The goodput has been defined as the number of bits per the unit of time forwarded to the correct destination interface, minus any bits lost or retransmitted, such as proposed in Section 3.17 of [RFC2647], which describes the benchmarking terminology for firewall performances. This definition requires that the test setup needs to be qualified to assure that it is not generating losses on its own.

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

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

2.6. Latency and Jitter

The latency, or the one-way delay metric, is discussed in [RFC7679]. There is a consensus on an adequate metric for the jitter that represents the one-way delay variations for packets from the same flow: the Packet Delay Variation (PDV) serves well in all use cases [RFC5481].

The end-to-end latency includes components other than just the queuing delay, such as the signal-processing delay, transmission delay, and processing delay. Moreover, the jitter is caused by variations in queuing and processing delay (e.g., scheduling effects). The introduction of an AQM scheme would impact end-to-end latency and jitter, and therefore these metrics should be considered in the end-to-end evaluation of performance.

2.7. Discussion on the Trade-Off between Latency and Goodput

The metrics presented in this section may be considered in order to discuss and quantify the trade-off between latency and goodput.

With regards to the goodput, and in addition to the long-term stationary goodput value, it is recommended to take measurements at every multiple of the minimum RTT (minRTT) between A and B. It is suggested to take measurements at least every K * minRTT (to smooth out the fluctuations), with K=10. Higher values for K can be considered whenever it is more appropriate for the presentation of the results, since the value for K may depend 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 recommended to take the samples on a per-packet basis whenever possible, depending on the features provided by the hardware and software and the impact of sampling itself on the hardware performance.

From each of these sets of measurements, the cumulative density function (CDF) of the considered metrics should be computed. If the considered scenario introduces dynamically varying parameters, temporal evolution of the metrics could also be generated. For each scenario, the following graph may be generated: the x-axis shows a queuing delay (that is, the average per-packet delay in excess of minimum RTT), the y-axis the goodput. Ellipses are computed as detailed in [WINS2014]: "We take each individual [...] run [...] as one point, and then compute the 1-epsilon elliptic contour of the maximum-likelihood 2D Gaussian distribution that explains the points. [...] we plot the median per-sender throughput and queueing delay as a circle. [...] The orientation of an ellipse represents the

covariance between the throughput and delay measured for the protocol." This graph provides part of a better understanding of (1) the delay/goodput trade-off for a given congestion control mechanism (Section 5), and (2) how the goodput and average queue delay vary as a function of the traffic load (Section 8.2).

3. Generic Setup 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

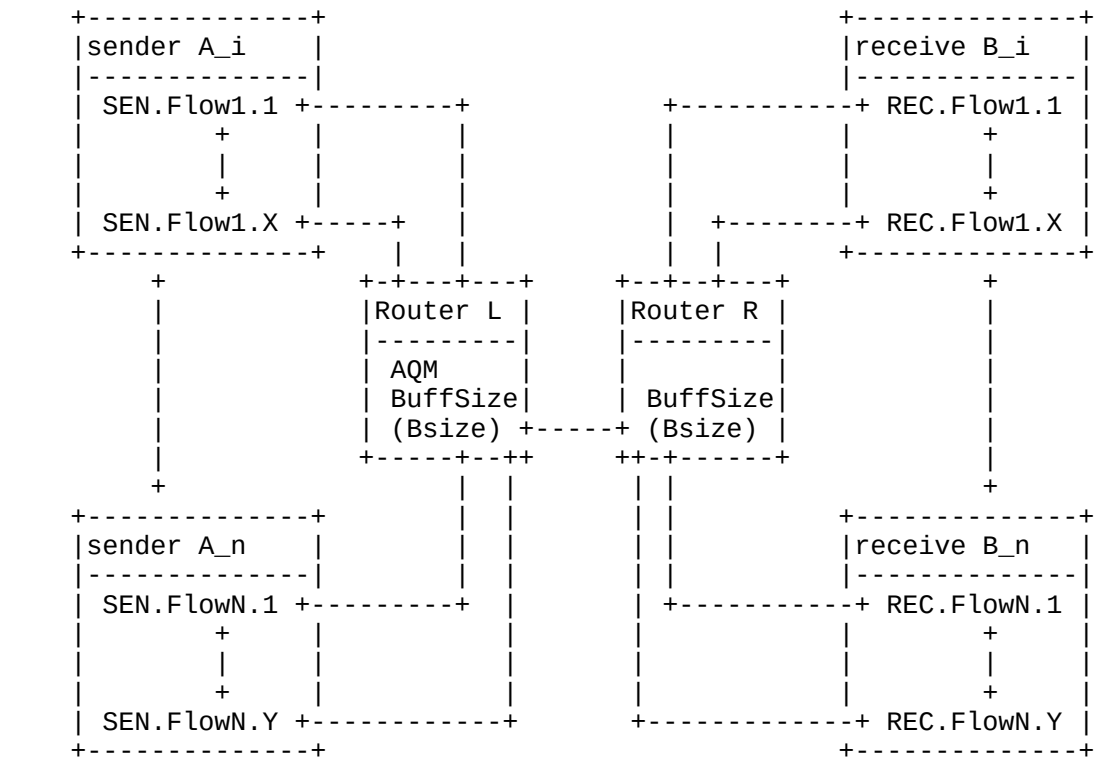

Figure 1: Topology and Notations

Figure 1 is a generic topology where:

- The traffic profile is a set of flows with similar characteristics -- RTT, congestion control scheme, transport protocol, etc.;
- Senders with different traffic characteristics (i.e., traffic profiles) can be introduced;
- The timing of each flow could be different (i.e., when does each flow start and stop?);
- Each traffic profile can comprise various number of flows;
- Each link is characterized by a couple (one-way delay, capacity);
- Sender A_i is instantiated for each traffic profile. A corresponding receiver B_i is instantiated for receiving the flows in the profile;
- Flows share a bottleneck (the link between routers L and R);
- The tester should consider both scenarios of asymmetric and symmetric bottleneck links in terms of bandwidth. In the case of an asymmetric link, the capacity from senders to receivers is higher than the one from receivers to senders; the symmetric link scenario provides a basic understanding of the operation of the AQM mechanism, whereas the asymmetric link scenario evaluates an AQM mechanism in a more realistic setup;
- In asymmetric link scenarios, the tester should study the bidirectional traffic between A and B (downlink and uplink) with the AQM mechanism deployed in one direction only. The tester may additionally consider a scenario with the AQM mechanism being deployed in both directions. In each scenario, the tester should investigate the impact of the drop policy of the AQM on TCP ACK packets and its impact on the performance (Section 9.2).

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

3.2. Buffer Size

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

If a specific buffer size is required, 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 document considers running three different congestion control algorithms between A and B:

- Standard TCP congestion control: The base-line congestion control is TCP NewReno with selective acknowledgment (SACK) [RFC5681].

- Aggressive congestion controls: A base-line congestion control for this category is CUBIC [CUBIC].

- Less-than-Best-Effort (LBE) congestion controls: Per [RFC6297], an LBE service "results in smaller bandwidth and/or delay impact on standard TCP than standard TCP itself, when sharing a bottleneck with it." A base-line congestion control for this category is Low Extra Delay Background Transport (LEDBAT) [RFC6817].

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

4. Methodology, Metrics, AQM Comparisons, Packet Sizes, Scheduling, and ECN

4.1. Methodology

A description of each test setup should be detailed to allow this test to be compared with other tests. This also allows others to replicate the tests if needed. This test setup should detail software and hardware versions. The tester could make its data available.

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

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

4.2. Comments on Metrics Measurement

This document presents the end-to-end metrics that ought to be used to evaluate the trade-off between latency and goodput as described 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 advise considering 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 the way these metrics can be measured, since that is expected to depend on the evaluation toolset.

4.3. Comparing AQM Schemes

This document recognizes that these guidelines may be used for comparing AQM schemes.

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

4.3.1. Performance Comparison

AQM schemes should be compared against the generic scenarios that are summarized in Section 13. AQM schemes may be compared for specific network environments such as data centers, home networks, etc. If an AQM scheme has parameter(s) that were externally tuned for optimization or other purposes, these values must be disclosed.

AQM schemes belong to different varieties such as queue-length based schemes (for example, RED) or queuing-delay based scheme (for example, CoDel, PIE). AQM schemes expose different control knobs associated with different semantics. For example, while both PIE and CoDel are queuing-delay based schemes and each expose a knob to

control the queuing delay -- PIE's "queuing delay reference" vs. CoDel's "queuing delay target", the two tuning parameters of the two schemes have different semantics, resulting in different control points. Such differences in AQM schemes can be easily overlooked while making comparisons.

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

1. Similar control parameters and implications: Testers should be aware of the control parameters of the different schemes that control similar behavior. Testers should also be aware of the input value ranges and corresponding implications. For example, consider two different schemes -- (A) queue-length based AQM scheme, and (B) queuing-delay based scheme. A and B are likely to have different kinds of control inputs to control the target delay -- the target queue length in A vs. target queuing delay in B, for example. Setting parameter values such as 100 MB for A vs. 10 ms for B will have different implications depending on evaluation context. Such context-dependent implications must be considered before drawing conclusions on performance comparisons. Also, it would be preferable if an AQM proposal listed such parameters and discussed how each relates to network characteristics such as capacity, average RTT, etc.

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

4.3.2. Deployment Comparison

AQM schemes must be compared against deployment criteria such as the parameter sensitivity (Section 8.3), auto-tuning (Section 12), or implementation cost (Section 11).

4.4. Packet Sizes and Congestion Notification

An AQM scheme may be considering packet sizes while generating congestion signals [RFC7141]. For example, control packets such as DNS requests/responses, TCP SYNs/ACKs are small, but their loss can severely impact application performance. An AQM scheme may therefore be biased towards small packets by dropping them with lower 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 an AQM scheme by transmitting smaller packets, and this could result in unsafe deployments and unhealthy transport and/or application designs.

An AQM scheme should adhere to the recommendations outlined in the Best Current Practice for dropping and marking packets [BCP41], and should not provide undue advantage to flows with smaller packets, such as discussed in Section 4.4 of the AQM recommendation document [RFC7567]. In order to evaluate if an AQM scheme is biased towards flows with smaller size packets, traffic can be generated, as defined in Section 8.2.2, where half of the flows have smaller packets (e.g., 500-byte packets) than the other half of the flow (e.g., 1500-byte packets). In this case, the metrics reported could be the same as in Section 6.3, where Category I is the set of flows with smaller packets and Category II the one with larger packets. The bidirectional scenario could also be considered (Section 9.2).

4.5. Interaction with ECN

ECN [RFC3168] is an alternative that allows AQM schemes to signal to receivers about network congestion that does not use packet drops. There are benefits to providing ECN support for an AQM scheme [WELZ2015].

If the tested AQM scheme can support ECN, the testers must discuss and describe the support of ECN, such as discussed in the AQM recommendation document [RFC7567]. Also, the AQM's ECN support can be studied and verified by replicating tests in Section 6.2 with ECN turned ON at the TCP senders. The results can be used not only to evaluate the performance of the tested AQM with and without ECN markings, but also to quantify the interest of enabling ECN.

4.6. Interaction with Scheduling

A network device may use per-flow or per-class queuing with a scheduling algorithm to either prioritize certain applications or classes of traffic, limit the rate of transmission, or to provide isolation between different traffic flows within a common class, such as discussed in Section 2.1 of the AQM recommendation document [RFC7567].

The scheduling and the AQM conjointly impact the end-to-end performance. Therefore, the AQM proposal must discuss the feasibility of adding scheduling combined with the AQM algorithm. It can be explained whether the dropping policy is applied when packets are being enqueued or dequeued.

These guidelines do not propose guidelines to assess the performance of scheduling algorithms. Indeed, as opposed to characterizing AQM schemes that is related to their capacity to control the queuing delay in a queue, characterizing scheduling schemes is related to the scheduling itself and its interaction with the AQM scheme. As one example, the scheduler may create sub-queues and the AQM scheme may be applied on each of the sub-queues, and/or the AQM could be applied on the whole queue. Also, schedulers might, such as FQ-CoDel [HOEI2015] or FavorQueue [ANEL2014], introduce flow prioritization. In these cases, specific scenarios should be proposed to ascertain that these scheduler schemes not only help in tackling the bufferbloat, but also are robust under a wide variety of operating conditions. This is out of the scope of this document, which focuses on dropping and/or marking AQM schemes.

5. Transport Protocols

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

AQM algorithms are often evaluated by considering the Transmission Control Protocol (TCP) [RFC793] with a limited number of applications. TCP is a widely deployed transport. It fills up available buffers until a sender transferring a bulk flow with TCP receives a signal (packet drop) that reduces the sending rate. The larger the buffer, the higher the buffer occupancy, and therefore the queuing delay. An efficient AQM scheme sends out early congestion signals to TCP to bring the queuing delay under control.

Not all endpoints (or applications) using TCP use the same flavor of TCP. A variety of senders generate different classes of traffic, which may not react to congestion signals (aka non-responsive flows in Section 3 of the AQM recommendation document [RFC7567]) or may not reduce their sending rate as expected (aka Transport Flows that are less responsive than TCP, such as proposed in Section 3 of the AQM recommendation document [RFC7567], also called "aggressive flows"). In these cases, AQM schemes seek to control the queuing delay.

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

5.1. TCP-Friendly Sender

5.1.1. TCP-Friendly Sender with the Same Initial Congestion Window

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

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

5.1.2. TCP-Friendly Sender with Different Initial Congestion Windows

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

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

- A single, long-lived non-application-limited TCP NewReno flow;
- A single, 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 first and the transmission of the application-limited flow starts after the non-application-limited flow has reached steady state. The steady state can be assumed when the goodput is stable.

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

5.2. Aggressive Transport Sender

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

long-lived, non-application-limited, CUBIC flow transfers data between sender A and receiver B. Other aggressive congestion control schemes may also be considered.

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

5.3. Unresponsive Transport Sender

This scenario helps testers evaluate how an AQM scheme reacts to a transport sender that is less responsive than TCP. Note that faulty transport implementations on an end host and/or faulty network elements en route that "hide" congestion signals in packet headers may also lead to a similar situation, such that the AQM scheme needs to adapt to unresponsive traffic (see Section 3 of the AQM recommendation document [RFC7567]). To this end, these guidelines propose the two following scenarios:

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

- The second scenario can be used to evaluate if the AQM scheme is able to keep the responsive fraction under control. This scenario considers a mixture of TCP-friendly and unresponsive traffic. It consists of a long-lived UDP flow from unresponsive application and a single long-lived, non-application-limited (unlimited data available to the transport sender from the application layer), 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 be higher than half of the bottleneck capacity. For each type of traffic, the graph described in Section 2.7 could be generated.

5.4. Less-than-Best-Effort Transport Sender

This scenario helps to evaluate how an AQM scheme reacts to LBE congestion control that "results in smaller bandwidth and/or delay impact on standard TCP than standard TCP itself, when sharing a bottleneck with it" [RFC6297]. There are potential fateful interactions when AQM and LBE techniques are combined [GONG2014]; this scenario helps to evaluate whether the coexistence of the proposed AQM and LBE techniques may be possible.

A single long-lived non-application-limited TCP NewReno flow transfers data between sender A and receiver B. Other TCP-friendly congestion control schemes may also be considered. Single long-lived non-application-limited LEDBAT [RFC6817] flows transfer data between sender A and receiver B. We recommend setting the target delay and gain values of LEDBAT to 5 ms and 10, respectively [TRAN2014]. Other LBE congestion control schemes may also be considered and are listed in the IETF survey of LBE protocols [RFC6297].

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

6. Round-Trip Time Fairness

6.1. Motivation

An AQM scheme’s congestion signals (via drops or ECN marks) must reach the transport sender so that a responsive sender can initiate its congestion control mechanism and adjust the sending rate. This procedure is thus dependent on the end-to-end path RTT. When the RTT varies, the onset of congestion control is impacted, and in turn impacts the ability of an AQM scheme to control the queue. It is therefore important to assess the AQM schemes for a set of RTTs between A and B (e.g., from 5 to 200 ms).

The asymmetry in terms of difference in intrinsic RTT between various paths sharing the same bottleneck should be considered, so that the fairness between the flows can be discussed. In this scenario, a flow traversing on a 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 the RTT fairness.

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

6.2. Recommended Tests

The recommended topology is detailed in Figure 1.

To evaluate the RTT fairness, for each run, two flows are divided into two categories. Category I whose RTT between sender A and receiver B should be 100 ms. Category II, in which the RTT between sender A and receiver B should be in the range [5 ms, 560 ms] inclusive. The maximum value for the RTT represents the RTT of a satellite link [RFC2488].

A set of evaluated flows must use the same congestion control algorithm: all the generated flows could be single long-lived non-application-limited TCP NewReno flows.

6.3. Metrics to Evaluate the RTT Fairness

The outputs that must be measured are: (1) the cumulative average goodput of the flow from Category I, goodput_Cat_I (see Section 2.5 for the estimation of the goodput); (2) the cumulative average goodput of the flow from Category II, goodput_Cat_II (see Section 2.5 for the estimation of the goodput); (3) the ratio goodput_Cat_II/ goodput_Cat_I; and (4) the average packet drop rate for each category (Section 2.3).

7. Burst Absorption

"AQM mechanisms might need to control the overall queue sizes to ensure that arriving bursts can be accommodated without dropping packets" [RFC7567].

7.1. Motivation

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

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

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

7.2. Recommended Tests

For this scenario, the tester must evaluate how the AQM performs with a traffic mix. The traffic mix could be composed of (from sender A to receiver B):

- Burst of packets at the beginning of a transmission, such as web traffic with IW10;
- Applications that send large bursts of data, such as bursty video frames;
- Background traffic, such as Constant Bit Rate (CBR) UDP traffic and/or A single non-application-limited 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	Web (IW 10)	CBR	Bulk TCP Traffic
I	0	1	1	0
II	0	1	1	1
III	1	1	1	0
IV	1	1	1	1

Figure 2: Bursty Traffic Scenarios

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

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

8. Stability

8.1. Motivation

The safety of an AQM scheme is directly related to its stability under varying operating conditions such as varying traffic profiles and fluctuating network conditions. Since operating conditions can vary often, the AQM needs to remain stable under these conditions without the need for additional external tuning.

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

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

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

8.2. Recommended Tests

Note that the traffic profiles explained below comprises non-application-limited TCP flows. For each of the below scenarios, the graphs described in Section 2.7 should be generated, and the goodput of the various flows should be cumulated. For Section 8.2.5 and Section 8.2.6, they should incorporate the results in a per-phase basis as well.

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

8.2.1. Definition of the Congestion Level

In these guidelines, the congestion levels are represented by the projected packet drop rate, which is determined when there is no AQM scheme (i.e., a drop-tail queue). When the bottleneck is shared among non-application-limited TCP flows, l_r (the loss rate projection) can be expressed as a function of N, the number of bulk TCP flows, and S, the sum of the bandwidth-delay product and the maximum buffer size, both expressed in packets, based on Eq. 3 of [MORR2000]:

l_r = 0.76 * N^2 / S^2

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

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

8.2.2. Mild Congestion

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

N_mild = ROUND (0.036*S)

8.2.3. Medium Congestion

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

N_med = ROUND (0.081*S)

8.2.4. Heavy Congestion

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

N_heavy = ROUND (0.114*S)

8.2.5. Varying the Congestion Level

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

8.2.6. Varying Available Capacity

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

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

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

- Experiment 2: The capacity varies between two values within a short timescale. As an example, the following phases may be considered: phase I -- 100 Mbps during 0-100 ms; phase II -- 10 Mbps during 100-200 ms; 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 a 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 the TCP slow-start phase.

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

The scenario consists of TCP NewReno flows between sender A and receiver B. To better assess the impact of draining rates on the AQM behavior, the tester must compare its performance with those of drop-

tail and should provide a reference document for their proposal discussing performance and deployment compared to those of drop-tail. Burst traffic, such as presented in Section 7.2, could also be considered to assess the impact of varying available capacity on the burst absorption of the AQM.

8.3. Parameter Sensitivity and Stability Analysis

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

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

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

9. Various Traffic Profiles

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

9.1. Traffic Mix

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

- Bulk TCP transfer
- Web traffic
- VoIP
- Constant Bit Rate (CBR) UDP traffic
- Adaptive video streaming (either unidirectional or bidirectional)

Various traffic mixes can be considered. These guidelines recommend examining at least the following example: 1 bidirectional VoIP; 6 web page downloads (such as those detailed in Section 7.2); 1 CBR; 1 Adaptive Video; 5 bulk TCP. Any other combinations could be considered and should be carefully documented.

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

9.2. Bidirectional Traffic

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

For this scenario, traffic must be generated in both downlink and uplink, as defined in Section 3.1. The amount of asymmetry between the uplink and the downlink depends on the context. These guidelines recommend considering a mild congestion level and the traffic presented in Section 8.2.2 in both directions. In this case, the metrics reported must be the same as in Section 8.2 for each direction.

The traffic mix presented in Section 9.1 may also be generated in both directions.

10. Example of a Multi-AQM Scenario

10.1. Motivation

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

10.2. Details on the Evaluation Scenario

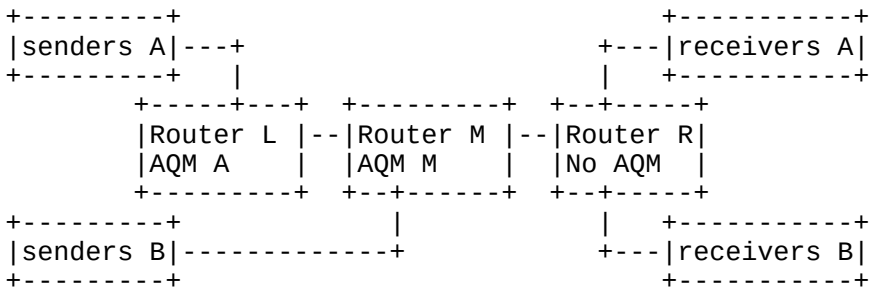

Figure 3: Topology for the Multi-AQM Scenario

Figure 3 describes topology options for evaluating multi-AQM scenarios. The AQM schemes are applied in sequence and impact the induced latency reduction, the induced goodput maximization, and the trade-off between these two. Note that AQM schemes A and B introduced in Routers L and M could be (I) same scheme with identical parameter values, (ii) same scheme with different parameter values, or (iii) two different schemes. To best understand the interactions and implications, the mild congestion scenario as described in Section 8.2.2 is recommended such that the number of flows is equally shared among senders A and B. Other relevant combinations of congestion levels could also be considered. We recommend measuring the metrics presented in Section 8.2.

11. Implementation Cost

11.1. Motivation

Successful deployment of AQM 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 their AQM logic.

AQM proposals should provide pseudocode 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, and queue length), computations involved, need for timers, etc. This helps to identify costs associated with implementing the AQM scheme on a particular hardware or software device. This also facilitates discussions around which kind of devices can easily support the AQM and which cannot.

11.2. Recommended Discussion

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

12. Operator Control and Auto-Tuning

12.1. Motivation

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

Any AQM scheme is likely to have parameters whose values affect the control law and behavior of an AQM. Exposing all these parameters as control parameters to a network operator (or user) can easily result

in an unsafe AQM deployment. Unexpected AQM behavior ensues when parameter values are set improperly. A minimal number of control parameters minimizes the number of ways a user can break a system where an AQM scheme is deployed at. Fewer control parameters make the AQM scheme more user-friendly and easier to deploy and debug.

"AQM algorithms SHOULD NOT require tuning of initial or configuration parameters in common use cases." such as stated in Section 4 of the AQM recommendation document [RFC7567]. A scheme ought to expose only those parameters that control the macroscopic AQM behavior such as queue delay threshold, queue length threshold, etc.

Additionally, the safety of an AQM scheme is directly related to its stability under varying operating conditions such as varying traffic profiles and fluctuating network conditions, as described in Section 8. Operating conditions vary often and hence the AQM needs to remain stable under these conditions without the need for additional external tuning. If AQM parameters require tuning under these conditions, then the AQM must self-adapt necessary parameter values by employing auto-tuning techniques.

12.2. Recommended Discussion

In order to understand an AQM’s deployment considerations and performance under a specific environment, AQM proposals should describe the parameters that control the macroscopic AQM behavior, and identify any parameters that require tuning to operational conditions. It could be interesting to also discuss that, even if an AQM scheme may not adequately auto-tune its parameters, the resulting performance may not be optimal, but close to something reasonable.

If there are any fixed parameters within the AQM, their setting should be discussed and justified to help understand whether a fixed parameter value is applicable for a particular environment.

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

13. Summary

Figure 4 lists the scenarios for an extended characterization of an AQM scheme. This table comes along with a set of requirements to present more clearly the weight and importance of each scenario. The requirements listed here are informational and their relevance may depend on the deployment scenario.

Scenario	Sec.	Informational requirement
Interaction with ECN	4.5	must be discussed if supported
Interaction with Scheduling	4.6	should be discussed
Transport Protocols	5	
TCP-friendly sender	5.1	scenario must be considered
Aggressive sender	5.2	scenario must be considered
Unresponsive sender	5.3	scenario must be considered
LBE sender	5.4	scenario may be considered
Round-Trip Time Fairness	6.2	scenario must be considered
Burst Absorption	7.2	scenario must be considered
Stability	8	
Varying congestion levels	8.2.5	scenario must be considered
Varying available capacity	8.2.6	scenario must be considered
Parameters and stability	8.3	this should be discussed
Various Traffic Profiles	9	
Traffic mix	9.1	scenario is recommended
Bidirectional traffic	9.2	scenario may be considered
Multi-AQM	10.2	scenario may be considered

Figure 4: Summary of the Scenarios and their Requirements

14. Security Considerations

Some security considerations for AQM are identified in [RFC7567]. This document, by itself, presents no new privacy or security issues.

15. References

15.1. Normative References

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

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <http://www.rfc-editor.org/info/rfc2544>.

[RFC2647] Newman, D., "Benchmarking Terminology for Firewall Performance", RFC 2647, DOI 10.17487/RFC2647, August 1999, <http://www.rfc-editor.org/info/rfc2647>.

[RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <http://www.rfc-editor.org/info/rfc5481>.

[RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <http://www.rfc-editor.org/info/rfc7567>.

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <http://www.rfc-editor.org/info/rfc7679>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <http://www.rfc-editor.org/info/rfc7680>.

15.2. Informative References

[ANEL2014] Anelli, P., Diana, R., and E. Lochin, "FavorQueue: a Parameterless Active Queue Management to Improve TCP Traffic Performance", Computer Networks Vol. 60, DOI 10.1016/j.bjp.2013.11.008, 2014.

[AQMPIE] Pan, R., Natarajan, P., Baker, F., and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", Work in Progress, draft-ietf-aqm-pie-08, June 2016.

[BB2011] Cerf, V., Jacobson, V., Weaver, N., and J. Gettys, "BufferBloat: what’s wrong with the internet?", ACM Queue Vol. 55, DOI 10.1145/2076450.2076464, 2012.

[BCP41] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141, February 2014.

<http://www.rfc-editor.org/info/bcp41>

[CODEL] Nichols, K., Jacobson, V., McGregor, A., and J. Iyengar, "Controlled Delay Active Queue Management", Work in Progress, draft-ietf-aqm-codel-04, June 2016.

[CUBIC] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., and R. Scheffenegger, "CUBIC for Fast Long-Distance Networks", Work in Progress, draft-ietf-tcpm-cubic-01, January 2016.

[FENG2002] Feng, W., Shin, K., Kandlur, D., and D. Saha, "The BLUE active queue management algorithms", IEEE Transactions on Networking Vol.10 Issue 4, DOI 10.1109/TNET.2002.801399, 2002, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1026008>.

[FLOY1993] Floyd, S. and V. Jacobson, "Random Early Detection (RED) Gateways for Congestion Avoidance", IEEE Transactions on Networking Vol. 1 Issue 4, DOI 10.1109/90.251892, 1993, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=251892>.

[GONG2014] Gong, Y., Rossi, D., Testa, C., Valenti, S., and D. Taht, "Fighting the bufferbloat: on the coexistence of AQM and low priority congestion control", Computer Networks, Elsevier, 2014, pp.115-128 Vol. 60, DOI 10.1109/INFCOMW.2013.6562885, 2014.

[HASS2008] Hassayoun, S. and D. Ros, "Loss Synchronization and Router Buffer Sizing with High-Speed Versions of TCP", IEEE INFOCOM Workshops, DOI 10.1109/INFOCOM.2008.4544632, 2008, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4544632>.

[HOEI2015] Hoeiland-Joergensen, T., McKenney, P., dave.taht@gmail.com, d., Gettys, J., and E. Dumazet, "The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm", Work in Progress, draft-ietf-aqm-fq-codel-06, March 2016.

[HOLLO2001] Hollot, C., Misra, V., Towsley, V., and W. Gong, "On Designing Improved Controller for AQM Routers Supporting TCP Flows", IEEE INFOCOM, DOI 10.1109/INFCOM.2001.916670, 2001, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=916670>.

[JAY2006] 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.

[MORR2000] Morris, R., "Scalable TCP congestion control", IEEE INFOCOM, DOI 10.1109/INFCOM.2000.832487, 2000, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=832487>.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[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, DOI 10.17487/RFC2309, April 1998, <http://www.rfc-editor.org/info/rfc2309>.

[RFC2488] Allman, M., Glover, D., and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", BCP 28, RFC 2488, DOI 10.17487/RFC2488, January 1999, <http://www.rfc-editor.org/info/rfc2488>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <http://www.rfc-editor.org/info/rfc3168>.

[RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <http://www.rfc-editor.org/info/rfc3611>.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, DOI 10.17487/RFC5348, September 2008, <http://www.rfc-editor.org/info/rfc5348>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <http://www.rfc-editor.org/info/rfc5681>.

[RFC6297] Welzl, M. and D. Ros, "A Survey of Lower-than-Best-Effort Transport Protocols", RFC 6297, DOI 10.17487/RFC6297, June 2011, <http://www.rfc-editor.org/info/rfc6297>.

[RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, DOI 10.17487/RFC6817, December 2012, <http://www.rfc-editor.org/info/rfc6817>.

[RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141, DOI 10.17487/RFC7141, February 2014, <http://www.rfc-editor.org/info/rfc7141>.

[TCPEVAL] Hayes, D., Ros, D., Andrew, L., and S. Floyd, "Common TCP Evaluation Suite", Work in Progress, draft-irtf-iccrg-tcpeval-01, July 2014.

[TRAN2014] Trang, S., Kuhn, N., Lochin, E., Baudoin, C., Dubois, E., and P. Gelard, "On The Existence Of Optimal LEDBAT Parameters", IEEE ICC 2014 - Communication QoS, Reliability and Modeling Symposium, DOI 10.1109/ICC.2014.6883487, 2014, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6883487>.

[WELZ2015] Welzl, M. and G. Fairhurst, "The Benefits to Applications of using Explicit Congestion Notification (ECN)", Work in Progress, draft-welzl-ecn-benefits-02, March 2015.

[WINS2014] Winstein, K., "Transport Architectures for an Evolving Internet", PhD thesis, Massachusetts Institute of Technology, June 2014.

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).

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

Authors' Addresses

Nicolas Kuhn (editor)
CNES, Telecom Bretagne
18 avenue Edouard Belin
Toulouse 31400
France

Phone: +33 5 61 27 32 13
Email: nicolas.kuhn@cnes.fr

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

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