Border Gateway Latency
draft-ietf-idr-bgp-distance-latency-00

Abstract

  A Border Gateway Protocol (BGP) speaker, defined in [RFC4271], often receives routes for the same prefix from multiple external peers. The BGP speaker must select a single "best path" for each prefix it uses and may advertise to its peers. The algorithm for selecting a best path includes several steps to decide which path is preferred. Often, the next step in the algorithm is to prefer the path with the lowest interior routing protocol metric. Sometimes, this is undesirable, for example, when the lowest metric path has excessive latency. This document describes an optional modification to the BGP best path selection algorithm that prefers paths with lower latency over paths with higher latency, when the latency is sufficiently different.

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 https://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 4 March 2023.

Copyright Notice

  Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

  This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Border Gateway Latency Model 5
  3.1. Flow Admission 5
  3.2. Packet Classification 6
  3.3. Queue Servicing 6
  3.4. Border Gateway Latency 6
    3.4.1. Queuing Delay Model 7
    3.4.2. Propagation Delay 8
    3.4.3. Processing Delay 8
    3.4.4. Serialization Delay 8
    3.4.5. Aggregate Queuing Optimization 8
    3.4.6. Border Gateway Latency Model Limitations 10
4. Configuration and Operation 10
  4.1. Configuration 10
    4.1.1. Queuing Delay Model 10
    4.1.2. Propagation Delay 10
    4.1.3. Flow Admit Threshold 10
    4.1.4. Flow Admit Threshold Hysteresis 10
    4.1.5. Flow Drop Threshold 10
    4.1.6. Flow Drop Hysteresis 10
    4.1.7. Queuing Delay Threshold 10
    4.1.8. Queuing Delay Hysteresis 10
    4.1.9. Aggregate Queuing Optimization 10
  4.2. Operation 10
  4.3. Interoperability 10
5. Implementation Status 10
6. Security Considerations 10
7. IANA Considerations 10
8. References 10
  8.1. Normative References 10
  8.2. Informative References 10
Author's Addresses 10

1. Introduction

  This document assumes that the following paragraph is used to select the best path:

  1. Routes are administratively ranked by the Weight (local router) or LOCAL_PREF (iBGP peers) attribute, and routes with a higher rank are always preferred over routes with a lower rank.

  2. Remaining routes are ranked by the AS_PATH length, and routes with a shorter length are preferred.

  3. Remaining routes are ranked by the ORIGIN attribute, and routes with a lower origin type (where IGP < EGP < INCOMPLETE) are preferred.

  4. Remaining routes are ranked by the MED attribute, and routes with a lower MED are preferred. This comparison is only performed between routes that have the same neighboring AS.

  5. Remaining routes are ranked by the IGP metric to the BGP NEXT_HOP, and routes with a lower IGP metric are preferred.

  6. Remaining routes are ranked by the BGP Identifier, and routes with a lower BGP Identifier are preferred.

  7. Remaining routes are ranked by the CLUSTER_LIST length, and routes with a shorter CLUSTER_LIST length are preferred.

  8. Remaining routes are ranked by the BGP peering address, and routes with a lower peering address are preferred.

  Step 5 is the step that often causes problems. Step 5 is usually the next step in the algorithm after step 4, and step 4 is frequently skipped. Step 4 is skipped when the MED attribute is absent, or when the MED attribute is present but the neighboring AS is different. Step 5 compares the metrics, which are usually configured by the network operator, to the IGP routing protocol. The network operator usually wants to minimize the IGP metric, which minimizes the sum of the link metrics along the path. This usually minimizes the latency, which is usually desirable.

  However, sometimes the lowest metric path has excessive latency. For example, consider the network in Figure 1. Router A has two paths to reach the destination prefix 192.0.2.0/24, via router B or via router C. The path via router B has a lower IGP metric, but the latency is 100 milliseconds. The path via router C has a higher IGP metric, but the latency is only 10 milliseconds. Router A will prefer the path via router B, because it has a lower IGP metric.

  Figure 1: Network Where Lowest IGP Metric Is Not Lowest Latency

  This document describes a modification to the BGP best path selection algorithm that prefers paths with lower latency over paths with higher latency, when the latency is sufficiently different. The modification is optional, and is disabled by default.

  This document is applicable to both eBGP and iBGP deployments. However, it is most commonly deployed in eBGP deployments, and is described in that context.

2. Terminology

  The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

  The following terms are used in this document. They are defined in [RFC4271] and are reproduced here for convenience:

  Autonomous System (AS): A set of routers under the same administration that use the Border Gateway Protocol and the decision process to determine the routes for packets that flow through the AS.

  BGP Identifier: A 4-octet unsigned integer that uniquely identifies a BGP speaker.

  BGP Speaker: A router that implements the Border Gateway Protocol.

  External Peer: A peer that is in a different AS.

  Internal Peer: A peer that is in the same AS.

  Loc-RIB: The combination of the routes selected by the local BGP speaker's Decision Process.

  Peer: A BGP speaker that has a transport protocol connection to another BGP speaker.

  Route: The information that is associated with a set of destination systems. The destination systems are specified by an IP address prefix. The information is contained in the BGP UPDATE message that is used to advertise the route.

  The following terms are used in this document. They are defined in [RFC4271] and are modified here:

  Adj-RIB-In: The set of routes that are received from a peer before the local BGP speaker's Decision Process has been applied to them. There is a separate Adj-RIB-In for each peer. The Adj-RIB-In is modified by this document.

  Adj-RIB-Out: The set of routes that are advertised to a peer after the local BGP speaker's Decision Process has been applied to them. There is a separate Adj-RIB-Out for each peer. The Adj-RIB-Out is modified by this document.

  The following terms are used in this document. They are defined in this document:

  Border Gateway Latency: The latency between two BGP speakers that are connected via a single BGP session.

  Border Gateway Latency Model: The model that is used to determine the Border Gateway Latency.

  Flow: A sequence of packets that are sent from a particular source to a particular unicast, anycast, or multicast destination that the source desires to label as a flow. A flow could consist of all packets in a specific transport connection or a single packet. Flows are identified by the combination of the source address, destination address, and flow label.

  Flow Admission: The process of admitting a flow into the Border Gateway Latency Model.

  Flow Admit Threshold: The threshold that is used to determine if a flow is admitted into the Border Gateway Latency Model.

  Flow Admit Threshold Hysteresis: The hysteresis that is used to determine if a flow is admitted into the Border Gateway Latency Model.

  Flow Drop Threshold: The threshold that is used to determine if a flow is dropped from the Border Gateway Latency Model.

  Flow Drop Hysteresis: The hysteresis that is used to determine if a flow is dropped from the Border Gateway Latency Model.

  Flow Table: The table that contains the flows that are admitted into the Border Gateway Latency Model.

  Packet Classification: The process of identifying which flow a packet belongs to.

  Queuing Delay: The delay that is caused by packets waiting in queues.

  Queuing Delay Hysteresis: The hysteresis that is used to determine if a flow is admitted into the Border Gateway Latency Model.

  Queuing Delay Threshold: The threshold that is used to determine if a flow is admitted into the Border Gateway Latency Model.

  Queue Servicing: The process of removing packets from queues and forwarding them.

3. Border Gateway Latency Model

  The Border Gateway Latency Model is a model that estimates the latency between two BGP speakers that are connected via a single BGP session. The Border Gateway Latency Model is used to determine if a flow is admitted into the Border Gateway Latency Model. The Border Gateway Latency Model is also used to determine if a flow is dropped from the Border Gateway Latency Model.

  The Border Gateway Latency Model is a conceptual model. It is not a requirement that an implementation use the Border Gateway Latency Model. However, the Border Gateway Latency Model is described in this document, because it is relatively simple to understand, and it is relatively simple to implement.

  The Border Gateway Latency Model is described in terms of its operations on packets. Implementations MAY perform these operations on packets, or MAY perform equivalent operations on groups of packets, or MAY perform equivalent operations in hardware.

  The Border Gateway Latency Model is described in terms of its operations on a single flow. Implementations MAY perform these operations on a single flow, or MAY perform equivalent operations on groups of flows, or MAY perform equivalent operations in hardware.

3.1. Flow Admission

  Flow Admission is the process of admitting a flow into the Border Gateway Latency Model. Flow Admission is performed before the Queuing Delay is measured for the flow.

  Flow Admission is performed as follows:

  1. Packet Classification is performed on the packet, to identify which flow it belongs to.

  2. If the flow is already admitted into the Border Gateway Latency Model, then Flow Admission is complete.

  3. If the flow is not already admitted into the Border Gateway Latency Model, then the Queuing Delay for the flow is measured.

  4. If the Queuing Delay is greater than or equal to the Flow Admit Threshold, then the flow is not admitted into the Border Gateway Latency Model.

  5. If the Queuing Delay is less than the Flow Admit Threshold minus the Flow Admit Threshold Hysteresis, then the flow is admitted into the Border Gateway Latency Model.

  6. If the Queuing Delay is less than the Flow Admit Threshold, but greater than or equal to the Flow Admit Threshold minus the Flow Admit Threshold Hysteresis, then the flow is not admitted into the Border Gateway Latency Model.

  Flow Admission is performed on packets that are received from peers. Flow Admission is not performed on packets that are received from the local router.

3.2. Packet Classification

  Packet Classification is the process of identifying which flow a packet belongs to. Packet Classification is performed on packets that are received from peers. Packet Classification is not performed on packets that are received from the local router.

  Packet Classification is performed on the source address, destination address, and flow label of packets. Packet Classification is performed on IPv6 packets. Packet Classification is not performed on IPv4 packets.

3.3. Queue Servicing

  Queue Servicing is the process of removing packets from queues and forwarding them. Queue Servicing is performed on packets that are received from peers. Queue Servicing is not performed on packets that are received from the local router.

  Queue Servicing is performed as follows:

  1. Packets are removed from queues in a round-robin fashion.

  2. Packets are removed from queues at a rate that is equal to the configured Queue Servicing Rate.

  3. Packets are removed from queues in the order that they were added to the queues.

  4. Packets are removed from queues until the queues are empty.

  5. Packets are removed from queues even if the Next Hop is not reachable.

  6. Packets are removed from queues even if the Next Hop's queues are full.

  7. Packets are removed from queues even if the packets will be dropped.

3.4. Border Gateway Latency

  The Border Gateway Latency is the latency between two BGP speakers that are connected via a single BGP session. The Border Gateway Latency is the latency between the two BGP speakers' Border Gateway Latency Models.

  The Border Gateway Latency is the sum of the following components:

  1. Queuing Delay

  2. Propagation Delay

  3. Processing Delay

  4. Serialization Delay

  The Border Gateway Latency is measured in units of time. The Border Gateway Latency is measured in milliseconds.

  The Border Gateway Latency is measured in one direction. The Border Gateway Latency is measured from the BGP speaker that sends packets to the BGP speaker that receives packets.

  The Border Gateway Latency is measured on packets that are received from peers. The Border Gateway Latency is not measured on packets that are received from the local router.

3.4.1. Queuing Delay Model

  The Queuing Delay is the delay that is caused by packets waiting in queues. The Queuing Delay is measured in milliseconds.

  The Queuing Delay is measured on packets that are received from peers. The Queuing Delay is not measured on packets that are received from the local router.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be dropped.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to the local router.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to other peers.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same AS.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other ASes.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP peer group.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP peer groups.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same VRF.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other VRFs.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same subsequent address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other subsequent address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same multicast address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other multicast address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same unicast address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other unicast address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same labeled unicast address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other labeled unicast address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same VPN address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other VPN address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same EVPN address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other EVPN address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same MVPN address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other MVPN address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same RTC address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other RTC address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same flowspec address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other flowspec address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LS address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LS address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same SR-TE address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other SR-TE address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same MUP address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other MUP address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same MCAST-VPN address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other MCAST-VPN address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same MDT address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other MDT address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same IPv4 address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other IPv4 address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same IPv6 address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other IPv6 address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same CLNS address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other CLNS address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same L2VPN address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other L2VPN address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LU address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LU address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-CT address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-CT address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-SD address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-SD address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-ID address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-ID address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-OPSF address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-OPSF address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-SRPM address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-SRPM address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-SRV6 address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-SRV6 address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LS-SPF address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LS-SPF address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LS-SRPM address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LS-SRPM address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LS-SRV6 address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LS-SRV6 address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LS-BGPSEC address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LS-BGPSEC address families.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in the same BGP-LS-OSPF address family.

  The Queuing Delay is measured on packets that are received from peers, even if the packets will be forwarded to peers in other BGP-LS-OS

Each of the queuing subsystems for classes A and B, contains Credit-
Based Shaper (CBS). The CBS serves a packet from a class
queue if the number of credits is greater than or equal to zero (defined in
Section 8.6.8.2 and Annex L.3 of [IEEE802.1Q]). The credit for each
class A or B increases based on the idleslope (dle slope (as guaranteed rate),
and decreases based on the sendslope (sle slope (the
difference between the guaranteed and the actual link rates), both of
which are parameters of the CBS (Section 8.6.8.2 of [IEEE802.1Q]). CBS. The CIT and BSS-BA flows are
mapped to class A and the rest of the flows are mapped to class B. Flows
are served by a transmission selection subsystem that serves packets
from each class based on its priority. All subsystems are non-preemptive. non-
preemptive. Consider a time interval of length t, in which a class A
traffic is bounded; it is assumed that the CBS traffic has a leaky
bucket arrival curve with two parameters σ as rate and b_A as
burst size, i.e., the amount of bits transmitted during a time within a time
interval t is bounded by $\rho_A * t + b_A$.

Additionally, it is assumed that the AVB flows are also replicated at
their source according to a leaky bucket arrival curve. At the
source, the traffic satisfies its regulation constraint, i.e. the
amount of bits generated within a time t is at most $\rho_A * t + b_A$.

At each Detnet transit node implementing an interleaved regulator,
packets of multiple flows are processed in one FIFO queue; the packet
that is stored for the longest time is processed first. Packets are
parameters; it is released at the earliest time at which this is
possible without violating the constraint.

The regulation parameters for a flow (leaky bucket rate and bucket
size) are the same at its source and at all Detnet transit nodes
along its path. This is because the flow may be merged with other flows; however, it
really there is still noticeably throughout the Detnet domain even
with clock synchronization. This phenomenon causes inaccuracy in the
estimation of the burst size of a flow, which leads to network
instability. To avoid that, when configuring the regulators, the
rates are set as the source rates with some positive margin.
[ThomasKhen] describes and provides solutions to this issue.

6.4.3. Delay Bound Calculation

A delay bound of the queuing subsystem (d_i) in Figure 5.1 of a given
Detnet node for an AVB flow of classes A or B can be computed if the
following are known:

- sum of leaky bucket rates of all flows of this class at this
  transit node == R, where R is given below for every class.

If the condition holds, the delay bounds for a flow of class X (A or
B) is d_X and calculated as:

$$d_X = T_X + (b_X + b_A + b_B)/(R - r_X)$$

where b_A is the minimum packet lengths of class A (A or B); r is
the output link transmission rate; T_X is the sum of the t term
(bucket size) for all the flows of the class X. Parameters T_X and
T_A are calculated as follows for class A and class B, separately:

If the flow is of class A:

$$T_A = I_A * (\sigma_A)/r$$
$$T_A = I_A * b_A * r_A * (1-r_A)/(\sigma_A)$$

where I_A is the idle slope for class A; I_A is the maximum packet
length of class A and M packets; I_A is the maximum packet length of
class A and M packets; σ_A and r_A are the burst size of
CBT traffic leaky bucket arrival curve.

If the flow is of class B:

$$T_B = I_B * (\sigma_B)/r + (I_A)/(r-r_A) * b_A * r_A * (1-r_A/r)/(\sigma_B)$$

where I_B is the idle slope for class B; I_B is the maximum packet
length of class B; I_AM is the maximum packet length of class M. where

**Then, as discussed in Section 4.2.2, an interleaved regulators does
not increase the delay bound of the upstream queuing subsystem;
therefore an end-to-end delay bound for a Detnet flow of class X is
the sum of d_X and L_X, where L_X is the latency of the link in the path the flow, where
d_X is the delay bound of queuing subsystem in node i which is
computed as above. According to the notation in Section 4.2.2, where h is the
number of hops (Detnet transit nodes) in a path i and interleaved
regulator in node i ($i = 1, 2, ..., h$), is:**

$$D_j = d_i + d_{i-1}$$

More information of delay analysis in such a Detnet transit node is
described in [ThomasInfocom2018].

6.4.4. Flow Admission

The delay bound calculation requires some information about each
node. For each node, it is required to know the idle slope of CBS
for each class A and B (I_A and I_B), as well as the transmission
rate of the output link (r). Also, for each input port, some flow
information on each class, i.e. maximum packet length of classes A,
B, and BE. Moreover, the leaky bucket parameters of CBT (σ_A,b_A)
and (σ_B,b_B)
must be known. To admit a flow/flows of classes A and B, their delay
requirements should must be guaranteed not to be violated. As described in
Section 6.4.3, the flows of different classes can be admitted
separately. In either of the problems, the rate and delay
should must be
guaranteed. Thus,

The static admission control:
: To guarantee performance of all AVB flows are known,
  therefore, for each AVB flow f_i, a delay bound can be
  calculated. The computed delay bound for every AVB flow
  can be made sure
  not to exceed its delay requirement. Moreover, the sum of
  the bandwidth assigned to all AVB flows must less than the rate
  allocated to each class (R). If these two conditions hold,
  the configuration is declared admissible.

The dynamic admission control:
: In the dynamic admission control, we allocate to every node and
  class A or B, static value for rate (R) and maximum packet
  size (b_X). In addition, for every node and every class A
  and B, two counters are maintained:

  - b_sat is equal to the sum of the leaky-bucket rates of all
    AVB flows already admitted at this node; At all times,
    we must have:

    $$b_sat <= R, \quad (Eq. 1)$$

  - b_sat is equal to the sum of the burst sizes of all flows
    already admitted at this node; At all times,
    we must have:

    $$b_sat <= b_X \quad (Eq. 2)$$

  A new AVB flow is admitted at this node, if Eqs. (1) and (2)
  continue to be satisfied after adding its leaky bucket rate and
  burst size to b_sat and b_sat, respectively. In the sum of
  the network, if it is admitted at all nodes along its
  path. When this happens, all variables b_sat and b_sat along
  the path must be incremented to reflect the admitted AVB
  flow. Initially, when an AVB flow leaves the network, all
  variables b_sat and b_sat along its path must be decremented
  to reflect the removal of the flow.

The choice of the static value of R and b_X at all nodes and classes
A and B is crucial. It determines the maximum bandwidth and burst size
allocated to this class at the node. b_X affects the delay bound and
the buffer requirement. **The value of R must** naturally be set such that

$$R >= I_X/(r-I_X)*b$$

where I_X is the maximum-slope-in
-----terval idleslope of credit-based shaper for class X(A,B),
r is the transmission rate of the output link and r_b is the leaky-
bucket rate of all the AVB flows.

6.4.5. Guaranteed Service Integrated

Guaranteed-Service Integrated service (intserv) is an architecture
that specifies the elements to guarantee quality of service (qos) on
networks [RFC2212].

The flow, at the source, has a leaky bucket arrival curve with two
parameters σ as rate and b as bucket size, i.e., the amount of bits
entering a node within a time interval t is bounded by $\sigma * t + m$.

If a resource reservation on a path is applied, a node provides a
service to guarantee that the bits arriving at time t will be
interpreted in a way that the bits might have to wait up to T before
being served with a rate greater or equal to R. The delay bound of
the flow traversing the node is $T + b / R$.

Consider a Guaranteed-Service Intserv path including a sequence of
h nodes. Let C_i and D_i be the rate and latency of the node's
service latency of i. Then, the end-to-end delay bound for a flow
on this can be calculated as $sum(T_i) + b / min(R_i)$.

The provided delay bound is based on a single case of Guaranteed-
Service Intserv. In practice, more complex cases, e.g. multiple
intserv and a leaky bucket arrival curve are available. If more
information about the nodes is known, e.g. the peak rate, the delay
bound can be computed. More detail is available in [RFC2212] and
Section 1.4.1 of [BerthaShah].

6.4. Cyclic Queuing and Forwarding

Annex T of [IEEE802Q] describes Cyclic Queuing and Forwarding (CQF),
which provides bounded latency and zero congestion loss using the
time-scheduled gates of [IEEE802Q] section 8.6.8.4. For a given
CQF node, the cycle time, T, is configured. Packets received in
the output queue layer of Figure 3. A cycle time T_c is configured
for each class of Detnet flows (, and all of the buffer area in a
node is divided equally among the set of cycles configured. A
Detnet domain at that cycle rate, all in phase. In such a mechanism,
the regulator, mentioned in Figure 1, is not required.

In the case of two-buffer CQF, each class of Detnet flows a has two
buffers, namely buffer$a$ and buffer$\bar{a}$. In a cycle (i) when buffer$a$
is selected to store the received packets, buffer$\bar{a}$ is selected to
transmit the already stored packets from the previous cycle ($i-1$).

In the next cycle ($i+1$), buffer$\bar{a}$ stores the received packets and
buffer$a$ transmits the packets received in cycle (i). The duration of
a cycle is T_c.

The per-hop latency is trivially determined by the cycle time T_c in
the packet transmission from a node at a cycle (i), is transmitted from the next
from the next node in cycle ($i+1$). Hence, the minimum delay
from the end of cycle (i) before the beginning of cycle (i) to
the end of cycle ($i+1$), or if it arrives the minimum delay is from the
end of cycle (i) to the beginning of cycle ($i+1$), i.e., more. Then,
if it has packets transmitted h hops, the maximum delay are

------($h+1$) T_c

and the minimum delay is

-------($h-1$) T_c

---which gives a latency variation of (h)

---The cycle length T_c should must be carefully chosen; it needs to be large
enough to accommodate all the Detnet traffic, plus at least one
maximum transmission unit (MTU) frame, that may be encountered, which
might be received within a cycle. Also, the value of T_c includes a
guard band to allow for any skew in the synchronization of the nodes
[1,2,3,4 defined in Figure 1. The value of T guarantees that the
delay bound is met for all Detnet flows. The minimum value for T
the next node in the same cycle. A two-buffer CQF is recommended if
T is small compared to T_c. For a large T, CQF with more buffers
can be used and a cycle identification label can be added to the
packets.

The per-hop latency is determined by the cycle time T_c: a packet
transmitted from a node at a cycle (i), is transmitted from the next
node at cycle ($i+1$). Then, if the packet transmit h hops, the
maximum delay are from the end of cycle (i) before the beginning of
cycle (i) to the end of cycle ($i+h$); else, the minimum latency is
from the end of cycle (i) before the H, to the beginning of cycle
($i+h$). Then, the maximum latency is:

($h+1$) T_c

and the minimum latency is:

($h-1$) T_c + H.

Ingress conditioning (Section 6.3) may be required if the source of a
Detnet flow does not, itself, employ CQF, since there are no per-
flow queues at the CQF transit nodes. A per-hop configuration is not
required in the CQF forwarding nodes.

7. Example application on Detnet IP network

This section provides an example application of this document on a
Detnet IP network. The network topology is taken from Section 1
of [RFC8578], that shows a simple IP network:

*

- The end-system 1 implements Guaranteed-Service Intserv as in
  Section 6.5 between Client and relay node 1.

*

- Sub-network 1 is a TSN network. The nodes in subnetwork 1
  implement credit-based shapers with asynchronous traffic shaping
  as in Section 6.4.

*

- Sub-network 2 is a TSN network. The nodes in subnetwork 2
  implement cyclic queuing and forwarding with two buffers as in
  Section 6.4

*

- The relay nodes 1 and 2 implement credit-based shapers with
  asynchronous traffic shaping as in Section 6.4. They also perform
  the aggregation and mapping of IP Detnet flows to TSN streams
  (Section 4.4 of [RFC7912]).

```
  Zeshen IP      Relay                          Relay     Zeshen IP
  End-System     Node 1                         Node 2    End-System
      |                                                       |
      |<-------------- Sub-network 1 ---------->|<-- Sub-network 2 -->|
      |<-------------- End-to-End Service ------------------->|
      |<-- Service --><-- Service --><-- Service --><-- Service -->|
      |  Function     |  Function     |  Function     |  Function  |
  +---+---------------+---------------+---------------+------------+---+
  |   |Forwarding|    |Forwarding|    |Forwarding|    |Forwarding|   |
  |   +----------+    +----------+    +----------+    +----------+   |
  |  L-Link  \        /--------\      /--------\      /        \     |
  |  ^^^^^^^^ \------/ Sub-     \----/ Sub-     \----/ Sub-     \--- |
  |           |Relay |  Network 1    |Relay |  Network 2    |Relay |  |
  |           |Node 1|               |Node 2|               |Node 3|  |
  |           \------/               \------/               \------/  |
  |                                                                   |
  |<-------------------- Zeshen IP ------------------------------->|
  |                                                                   |
  |<---- E2 --->|<------------- E2_p ---------------->|<-- E2_p -->|
```

Figure 5: A Simple Detnet-Enabled IP Network, taken from RFC8939

Consider a fully centralized control plane for the network of
Figure 5 as described in Section 3.2 of
[I-D.ietf-detnet-control-plane-framework]. Suppose end-system 1
wants to create a Detnet flow with traffic specification destined to
end-system 2 with end-to-end delay bound requirement D. Therefore,
the Detnet controller must find a feasible path consisting of
sub-networks a number of relay points through the network (Section 3.2
of [I-D.ietf-detnet-control-plane-framework]). To select a proper
path, the controller must know the route on end-to-end delay bound on
every node of each selected path p.

The end-to-end delay bound is $d1 + d2_p + d3_p$, where $d1$ is the delay
bound from end-system 1 to the entrance of relay node 1, $d2_p$ is the
delay bound from relay node 1 to the exit of relay node 2 through the
node in sub-network 2, and $d3_p$ the delay bound of path p from the
exit of relay node 2 to end-system 2. The delay bound computation in 1
is explained in Section 6.5. Since the relay node 1, sub-network 1
and relay node 2 implement aggregate queuing, we use the results in
Section 6.4.3 to compute the delay bound $d2_p$ of path p. And
finally, $d3_p$ is computed using the delay bound computation of
Section 6.4. Any path p such that $d1 + d2_p + d3_p <= D$ satisfies
the end-to-end delay requirement. If there is such a path p, then
the control plane may compare one set of valid paths and rank the
delay bound computation or do-not-admit reject the Detnet flow.

As soon as the control plane selects a path that satisfies the delay
bound constraint, it allocates and reserves the resources in the path
for the Detnet flow (Section 3.2.1
[I-D.ietf-detnet-control-plane-framework]).

8. Security considerations

Detailed security considerations for Detnet are cataloged in
[RFC8655]. For more general security considerations are described in
[RFC3552].

Security aspects that are unique to Detnet are those whose aim is to
provide Detnet service protection against failure to deliver packets
to-end delivery latency and zero congestion loss. Achieving such
loss rates and bounded latency may not be possible in the face of a
node failure or a network attack. Packet replication and elimination
Thread Model of RFC 7 [RFC7554] that can arbitrarily drop or delay
any or all traffic. In order to prevent meaningful security
compromises, Detnet must be operated in networks that are not
control the physical links of the Detnet domain but may have the
ability to control a network node within the boundary of the Detnet
domain.

A security consideration for this document is to ensure the resource
reservation request (e.g. bandwidth and burst size) and configuration
of packets during reservation may lead the flow not to be admitted or
have delay bound violation. Security mitigation for this issue is
described in Section 7.6 of [RFC8655].

9. IANA considerations

This document has no IANA actions.

10. Acknowledgment

**We would like to thank Lou Berger, Tony Przygienda and John Scudder
for their useful feedback on this document.**

11. References

11.1. Normative References

[RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification
of Guaranteed Quality of Service", RFC 2212,
DOI 10.17487/RFC2212, September 1997,
<https://www.rfc-editor.org/info/rfc2212>.

**[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z.,
and W. Weiss, "An Architecture for Differentiated
Services", RFC 2475, DOI 10.17487/RFC2475, December 1998,
<https://www.rfc-editor.org/info/rfc2475>.**

[RFC4654] Bryant, S., Ed., Martini, L., Swallow, G., and A. Malis,
"Pseudowire Emulation Edge-to-Edge (PWE3) Control Word",
RFC 4385, DOI 10.17487/RFC4385, July 2012,
<https://www.rfc-editor.org/info/rfc4385>.

[RFC7567] Baker, F. and R. Pan, "On Queuing, Marking, and Dropping",
RFC 7567, DOI 10.17487/RFC7567, April 2015,
<https://www.rfc-editor.org/info/rfc7567>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", RFC 8655,
DOI 10.17487/RFC8655, October 2019,
<https://www.rfc-editor.org/info/rfc8655>.

[RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S.
Bryant, "Deterministic Networking (DetNet) Data Plane:
IP", RFC 8939, DOI 10.17487/RFC8939, November 2020,
<https://www.rfc-editor.org/info/rfc8939>.

[RFC9964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant,
S., and J. Korhonen, "Deterministic Networking (DetNet)
Data Plane: MPLS", RFC 9964, DOI 10.17487/RFC9964, January
2021, <https://www.rfc-editor.org/info/rfc9964>.

[RFC9914] Varga, B., Farkas, J., Cummings, R., Jiang, Y., and D.
Fedyk, "Time-Sensitive Networking (TSN) Extensions to
Generalized Precision Time Protocol (gPTP) for the
Deterministic Networking (DetNet) IP", RFC 9914,
DOI 10.17487/RFC9914, March 2021,
<https://www.rfc-editor.org/info/rfc9914>.

11.2. Informative References

[AmericaLabs]
J.C.R. Bennett, K. Benson, A. Charny, W.F. Courtney, and
J. Le Boudec, "Delay Jitter Bounds and Packet Scale Rate
Guarantee for Expedited Forwarding",
<https://dl.acm.org/citation.cfm?id=504575>.

[CharnyDelay]
A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network
with Aggregate Scheduling", <https://link.springer.com/
chapter/10.1007/3-540-39939-9_15>.

[I-D.ietf-detnet-controller-plane-framework]
A. Malis, X. Geng, A. Chen, F. Qin, and B. Varga,
"Deterministic Networking (DetNet) Controller Plane
Framework draft-ietf-detnet-controller-plane-framework-
02", <https://datatracker.ietf.org/doc/html/draft-ietf-
detnet-controller-plane-framework-02>.

[IEEE8021Q]
"IEEE 802.1, "IEEE Std 802.1Q-2018 - IEEE Standard for Local
and Metropolitan area networks - Bridges and Bridged
Networks"", <http://ieeexplore.ieee.org/document/8403927>.

[IEEE802Qbv]
"IEEE 802.1, "IEEE P802.1Qbv-2015 - IEEE Draft Standard for Local
and metropolitan area networks - Bridges and Bridged
Networks - Amendment: Enhancements for Scheduled Traffic",
<https://www.ieee802.org/1/files/private/bv-drafts/>.

[IEEE8021TSN]
"IEEE 802.1, "IEEE 802.1 Time-Sensitive Networking (TSN)
Task Group", <http://www.ieee802.org/1/>.

[IEEE8023]
"IEEE 802.3, "IEEE Std 802.3-2018: IEEE Standard for
Ethernet"", <http://ieeexplore.ieee.org/document/8457449>.

[LeBoudecTheory]
J. Le Boudec and P. Thiran, "A Theory of Traffic Regulators for
Deterministic Networks with Application to Interleaved
Regulators",
<https://lemexplore.ieee.org/document/8519761>.

[BerthaShah]
F. Bouhafs, N. Mans, B. Shariati, F. Hussain, and M. Dohler,
"Interleaved Peristaltic Shaper", IEEE 802.1Qcr Time
Sensitive Networks",
<https://ieeexplore.ieee.org/abstract/document/7337870>.

[ThomasKhen]
J. Thomas and J.-Y. Le Boudec, "On Time Synchronization
Issues in Time-Sensitive Networks with Regulators and
Nonideal Clocks",
<https://dl.acm.org/doi/10.1145/3393691.3394206>.

[TSNwithATS]
J. Thomas, S. Mohammadpour, E. Stai, M. Mohiuddin, and J.-Y. Le
Boudec, "End-to-end Latency and Backlog Bounds in Time-
Sensitive Networks with Credit Based Shapers and
Asynchronous Traffic Shaping",
<https://arxiv.org/abs/1904.10864>.

Authors' Addresses

Norman Finn
Huawei Technologies Co. Ltd
3101 Rio Way
Spring Valley, California 91977
USA
Use Preferred Email Address

Phone: + 925 980 6430
Email: nfinn@LMCinnovations.com

Jean-Yves Le Boudec
EPFL
IC Station 14
CH-1015
Switzerland
+41 21 693 6695 1015
Switzerland

Email: jean-yves.leboudec@epfl.ch

Khaz Mohammadpour
EPFL
IC Station 14
CH-1015
Switzerland
+41 21 693 6695 1015
Switzerland

Email: mhaz.mohammadpour@epfl.ch

Jiayi Zheng
Huawei Technologies Co. Ltd
Huawei Base, Bantian Road
Beijing 100053
China

Email: zhengjiayi1@huawei.com

Balázs
Balazs Varga
Ericsson
Budapest
Konyves Kalman Kalman krt. 11/B
Budapest 1097
Hungary

Email: balazs.a.varga@ericsson.com

János

Janos Farkas
Ericsson
Budapest
Konyves Kalman Kalman krt. 11/B
Budapest 1097
Hungary

Email: janos.farkas@ericsson.com