            Circuit Breaker Assisted Congestion Control (CBACC): Protocol
                                 Specification
                      draft-jholland-cb-assisted-cc-01

Abstract

   This document specifies Circuit Breaker Assisted Congestion Control
   (CBACC), which provides bandwidth information from senders to
   intermediate network nodes to enable good decisions for fast-trip
   Network Transport Circuit Breaker activity
   ([I-D.ietf-tsvwg-circuit-breaker]) when necessary for network health.
   CBACC is specifically designed to support protocols using IP
   multicast, particularly as a supplement to receiver-driven congestion
   control protocols to help affected networks rapidly detect and
   mitigate the impact of scenarios in which a network is oversubscribed
   to flows which are not responsive to congestion.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

    This document specifies Circuit Breaker Assisted Congestion Control
    (CBACC).

    CBACC is a congestion control building block designed for use with IP
    traffic that has a known maximum bandwidth, which does not reduce its
    sending rate in response to congestion. CBACC is specifically
    designed to supplement protocols using receiver-driven multicast
    congestion control systems that rely on well-behaved receivers to
    achieve congestion control in a very highly scalable system (up to
    millions of receivers) without a feedback path that reduces sending

rates by senders. Examples of congestion control systems fitting
this description include PLM, RLM, RLC, FLID-DL, SMCC, ESMCC, QIRLM,
and WEBRC [RFC3738].

CBACC addresses a vulnerability to "overjoining", a condition in
which receivers (particularly malicious receivers) subscribe to
traffic which, from the sending side, is non-responsive to
congestion. Overjoining attacks and the challenges they present are
discussed in more detail in Appendix A.

A careful reading of the congestion control requirements of UDP Best
Practices [I-D.ietf-tsvwg-rfc5405bis] suggests that a network that
forwards multicast traffic is required to operate a circuit breaker
to maintain network health under a persistent overjoining condition,
at a cost of cutting off some or all multicast traffic across the
network during high congestion.

CBACC provides a mechanism for networks to mitigate the impact of
overjoining within a network by introducing a mechanism for
communicating the bandwidth of non-responsive flows from the sender
of the flow to the transit nodes forwarding the flow. The bandwidth
information is sufficient to implement a fast-trip circuit breaker
[I-D.ietf-tsvwg-circuit-breaker] within a single network node which
can specifically block or police flows when receivers have overjoined
the network's capacity.

In conjunction with receiver counts (e.g. via [RFC6807]) such nodes
can also provide much improved network fairness for circuit breaking
decisions during an overjoining condition.

In addition to streams using multicast receiver-driven congestion
control, CBACC may also be suitable for use with other traffic, both
unicast and multicast, that does not respond to congestion by
reducing sending rates, including certain profiles of RTP [RFC3550]
over either unicast or multicast, as well as several tunneling
protocols (e.g. AMT [RFC7450] and GRE [RFC2784]) when they are known
to carry traffic that would be suitable for CBACC. A complete
specification for use of CBACC with unicast protocols and with
tunneling protocols is out of scope for this document, though the
security issues section does mention a few special considerations for
potential unicast usage.

CBACC-compliant senders transmit Bandwidth Advertisements through the
same transport path as the data traffic, so that circuit breakers can
make informed decisions about how flows should be prioritized for
circuit breaking. Additionally, CBACC-compliant circuit breakers
transmit information to receivers about flows which have been or
might soon be circuit-broken, to encourage CBACC-aware applications

to use alternate methods to retrieve equivalent (though probably
lower-quality and possibly less efficient) data when possible.

This document describes a building block as defined in [RFC3048].
This document describes a congestion control building block that
conforms to [RFC2357]. This document follows the general guidelines
provided in [RFC3269], in addition to the requirements on RFCs from
[RFC5226] and [RFC3552].

2. Terminology

Term	Definition
circuit breaker	See [I-D.ietf-tsvwg-circuit-breaker]
controlled environment	See [I-D.ietf-tsvwg-rfc5405bis] Section 3.6
general internet	See [I-D.ietf-tsvwg-rfc5405bis] Section 3.6
flow	traffic for a single (source,destination) IP pair, including destinations that are group addresses
upstream	along a network topology path in the direction of a flow's sender
downstream	along a network topology path in the direction of a flow's receiver
ingress interface	the (single) upstream interface for a flow in a circuit breaker
egress interface	a downstream interface for a flow in a circuit breaker

Table 1

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 [RFC2119].

3. Rationale

CBACC is defined as an independent congestion control building block
because it would be a useful supplement a wide variety of receiver-
driven multicast congestion control schemes, such as [PLM] or other
methods based on receiver-driven conformance to a measurement of
available network bandwidth or congestion.

CBACC is also potentially valuable, even without other congestion
control systems, in controlled environments where congestion control

may not be required (e.g. for certain profiles of RTP [RFC3550]), since CBACC can provide protection for such a network against congestion due to sender or network mis-configuration.

CBACC provides a new form of communication between senders and network transit nodes to facilitate fast-trip circuit breakers as described in section 5.1 of [I-D.ietf-tsvwg-circuit-breaker] which are not available via previously existing methods. When used in conjunction with compatible circuit breakers, CBACC can greatly improve the safety of a network that accepts and delivers interdomain massively scalable multicast traffic to potentially untrusted receivers.

4. Applicability

   CBACC relies on the presence of CBACC-aware circuit breakers on a flow's transit path in order to provide congestion control in a network. In the absence of any CBACC-aware circuit breakers on a network path, CBACC constitutes a small extra overhead to a flow without providing any additional value.

   CBACC provides a form of congestion control for massively scalable protocols using the IP multicast service. CBACC is best used in conjunction with another receiver-driven multicast congestion control, but it is also suitable for use even without another congestion control mechanism, or when presence of another congestion control mechanism is unproven, such as when accepting multicast joins from untrusted receivers.

5. Protocol Specification

5.1. Overview

   CBACC senders send Bandwidth Advertisement packets to advertise the maximum sending bandwidth along the data path for a flow through a network.

   CBACC bandwidth information is monitored by CBACC circuit breakers along the network path, which may block the forwarding of traffic for some flows in order to maintain network health. When a flow is blocked, a CBACC circuit breaker sets a bit in Bandwidth Advertisement packets before they're forwarded downstream that indicates to subscribed receivers of that flow that traffic has been blocked.

   The protocol also defines a way to notify downstream receivers when a flow is in danger of being circuit broken in the near future. A CBACC-capable transport node SHOULD send this information when it is

known, as described in section [TBD]. This gives applications an
opportunity to gracefully shift to a lower-bandwidth version of the
same content, when possible, providing an early warning system for
avoiding congestion more smoothly.

A Bandwidth Advertisement packet constitutes an "ingress meter" as
described in section 3.1 of [I-D.ietf-tsvwg-circuit-breaker]. The
configured bandwidth caps of egress interfaces likewise constitute
"egress meters". However, the diagram in the referenced document is
simplified by running the ingress and egress on the same network
node. At the CBACC-aware circuit breaker, the CBACC node has both
pieces of information as soon as a Bandwidth Advertisement is
received, and can trip the circuit breaker if the aggregate
advertised CBACC bandwidth exceeds the actual bandwidth available on
any egress interfaces.

5.2. Packet Header Fields

5.2.1. Bandwidth Advertisement

5.2.1.1. As an IP header option

Bandwidth advertisements can appear as either an IPv4 header option
(as in Section 3.1 of [RFC0791]) or as an IPv6 extension header
option (as in section 4.2 of [RFC2460]). They have the same layout:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Length    |B|D|P|  Res  |    Priority    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Bandwidth                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
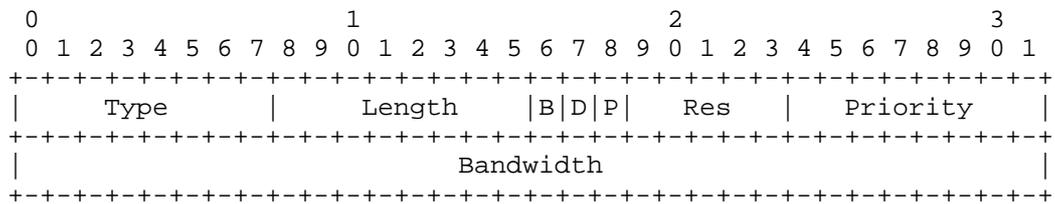
Figure 1

Bandwidth advertisements sent as IPv4 header options use option value
[TBD], with the "copied" bit set and the option class "control", as
specified in [RFC0791] section 3.1. Until and unless IANA assigns a
value, this will be option number 158 as described in section 8 of
[RFC4727] for experiments using IPv4 Option types. The length field
is 8.

Bandwidth advertisements sent as IPv6 header options use option value
[TBD], with the "action" bits set to "skip" and the "change" bit set
to 1, as specified in [RFC2460] section 4.2. Until and unless IANA
assigns a value, this will be option number 0x3e as described in

section 8 of [RFC4727] for experiments using IPv6 Option Types. The
length field is 6.

Using an IP header option has the benefit of exposing the bandwidth
to all CBACC-compatible routers, in much the same way the IP Router
Alert option would, but without being processed or causing undue load
in non-CBACC routers.

The IP Header encapsulations DO work with IPSEC. As described in
Appendix A of [RFC4302], the IP header fields are properly treated as
mutable and zeroed for the IPSEC ICV calculations. CBACC circuit
breakers MAY change bits in transit. The Bandwidth Advertisement
header itself IS NOT protected by IPSEC security services, but
protection of other parts of the packet remain unchanged.

5.2.1.2. Field definitions

5.2.1.2.1. Bandwidth

As in several other protocols sending bandwidth values such as OSPF-
TE [RFC3630], the bandwidth is expressed in bytes per second (not
bits), in IEEE floating point format. For quick reference, this
format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|    Exponent   |                  Fraction                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2

S is the sign, Exponent is the exponent base 2 in "excess 127"
notation, and Fraction is the mantissa - 1, with an implied binary
point in front of it. Thus, the above represents the value:

$(-1)**(S) * 2**(Exponent-127) * (1 + Fraction)$

For more details, refer to [IEEE.754.1985].

Figure 3

5.2.1.2.2. B (Blocked) bit

Indicates that the flow has been circuit-broken.

5.2.1.2.3. D (Danger) bit

   Indicates that the flow is in danger of being circuit-broken.

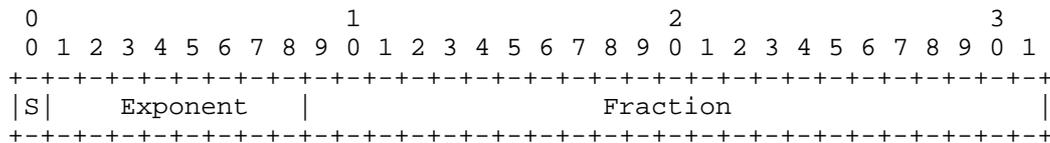### 5.2.1.2.4. P (Police) bit

   Indicates that the flow should be policed instead of blocked. Flows
   marked for policing by the sender should have traffic proportionally
   dropped when bandwidth is needed, according to their priority. [TBD]
   Flesh this concept out, and decide whether it's actually viable.
   This was my attempt at addressing a suggestion from Bob Briscoe at
   IETF 97 in ICCRG at the mic, IIRC. It probably requires more state,
   such as total desired policable bandwidth, total current policed
   bandwidth, and current policing bandwidth per-flow, plus some
   definition of how to decide between cutting off some flows and
   policing others. This may not be worth the hassle, but there are
   some use cases such as FEC repair traffic which might actually be
   nicer this way. However, it might also be possible to get the same
   effect by assigning priority to those repair flows. Things like
   video enhancement layers of course are probably better done as a
   complete cutoff.

5.2.1.2.5. Res (Reserved bits)

   The sender MUST set all reserved bits to 0 when sending a CBACC
   control packet. Receivers and CBACC-capable transit nodes MUST
   accept any value in the reserved bits.

5.2.1.2.6. Priority

   The sender MAY indicate relative priorities of different streams from
   the same sender with this field. This is an 8-bit unsigned integer,
   and higher values are kept preferentially over other traffic from the
   same sender with lower priority values, so all flows with a lower
   priority value are circuit-broken before any flows with a higher
   priority value. Among multiple flows from the same sender with the
   same priority, the highest bandwidth flows are circuit- broken first.

5.3. States

5.3.1. Interface State

   A CBACC circuit breaker holds the following state for each interface,
   for both the inbound and outbound directions on that interface:

   o aggregate bandwidth: The sum of the bandwidths of all non-
      circuit-broken CBACC flows which transit this interface in this
      direction.

o bandwidth limit: The maximum aggregate CBACC advertised bandwidth
   allowed, not including circuit-broken flows. This may depend on
   administrative configuration and congestion measurements for the
   network, whether from this node or other nodes. It's out of
   scope for this document to define such congestion measurements.
   Network operators should carefully consider that this bandwidth
   limit applies to flows that are unresponsive to congestion.

   When reducing the bandwidth limit due to congestion, the circuit
   breaker MUST NOT reduce the limit by more than half its value in
   10 seconds, and SHOULD use a smoothing function to reduce the
   limit gradually over time.

   It is RECOMMENDED that no more than half the capacity for a link
   be allocated to CBACC flows if the link might be shared with TCP
   or other traffic that is responsive to congestion.

   Depending on administrative configuration and the physical
   characteristics of the interface, the bandwidth limit may be
   either shared between upstream and downstream traffic, or it may
   be separate. Either a single shared value should be used, or two
   separate independent values should be used for the inbound and
   outbound directions for an interface.

o CBACC bandwidth warning threshold: A soft bandwidth threshold.
   When the aggregate CBACC advertised bandwidth exceeds this
   threshold, flows that would have been circuit-broken with a
   bandwidth limit at this threshold MUST have the Danger bit set in
   the Bandwidth Advertisement packets that are forwarded by this
   circuit breaker. This threshold SHOULD be configurable as a
   proportion of the bandwidth limit, and MUST remain at or below
   the bandwidth limit when the bandwidth limit changes. The
   recommended proportion value is .75, but specific networks may
   use a different value if deemed useful by the network operators.

5.3.2. Flow State

   The following state is kept for flows that are joined from at least
   one downstream interface and for which at least one CBACC Bandwidth
   Advertisement packet has been received:

o bandwidth: The bandwidth from the most receintly received
   Bandwidth Advertisement.

o ingress status: One of the following values:

       * 'subscribed'

Indicates that the circuit breaker is subscribed upstream to
the flow and forwarding data and control packets through zero
or more egress interfaces.

* 'pruned'
Indicates that the flow has been circuit-broken. A request to
unsubscribe from the flow has been sent upstream, e.g. a PIM
prune (section 3.5 of [RFC7761]) or a "leave" operation via
IGMP, MLD, or another appropriate group membership protocol.

* 'probing'
Indicates that the flow was circuit-broken previously, and is
currently joined upstream to refresh the most recent Bandwidth
Advertisement in order to evaluate reinstating the flow.

o probe timer: Used to periodically probe a flow in the 'pruned'
   state, to evaluate returning to 'forwarding'.

Flows additionally have a per-interface state for egress interfaces:

o egress status: One of the following values:

* 'forwarding'
Indicates that the flow is a non-circuit-broken flow in steady
state, forwarding data and control packets downstream.

* 'blocked'
Indicates that data packets for this flow are NOT forwarded
downstream via this interface. Bandwidth Advertisements are
still forwarded, each with the 'Blocked' bit set to 1. All
other flow traffic MUST be dropped.

5.4. Functionality

The CBACC building block on a sender MUST have access to the maximum
bandwidth that may be sent at any time in the following 3 seconds. A
CBACC sender MUST send this value in a Bandwidth Advertisement packet
once per second. The end result of the traffic sent on the wire for
a particular flow MUST honor this maximum bandwidth commitment, such
that bandwidth measurements taken over any sliding window one-second
period MUST NOT exceed any of prior 3 maximum Bandwidth
Advertisements (or any of them, if fewer than 3 have been sent).

A CBACC circuit breaker MUST order its monitored flows based on per-
flow estimates of network fairness and preferentially circuit break
less fair flows when bandwidth limits are exceeded. A normative
method to determine network fairness for a flow is out of scope for
this document, but CBACC circuit breaker implementations SHOULD

provide a capability for network operators to configure
administrative biases for specific sets of flows, and network
operators SHOULD consider fairness concerns as expressed in [RFC2914]
section 3.2 and other relevant documents describing best practices.

In particular, fairness metrics SHOULD favor multicast flows with
many receivers over multicast flows with few receivers and flows with
low bandwidth over flows with high bandwidth. When receiver counts
are known (for example via the experimental PIM extension specified
in [RFC6807]) a RECOMMENDED metric is (bandwidth/receiver count),
though other metrics MAY be used where deemed appropriate by network
operators following internet best practices, or when receiver counts
can't be determined.

A CBACC sender MUST send Bandwidth Advertisements once per second.
(Implementation-specific jitter in timer implementations not
exceeding .1s is acceptable.)

If a circuit breaker receives more than 5 Bandwidth Advertisement
packets for a flow in two seconds, the circuit breaker SHOULD set the
flow to "pruned" and leave the upstream channel, and MUST drop
Bandwidth Advertisement packets in excess of one per second.

Flows which are currently circuit-broken on an egress interface are
set to "blocked". When a flow on an egress interface is in blocked
state, Bandwidth Advertisement packets MUST be forwarded except as
described in the preceding paragraph, the "Blocked" bit MUST be set
to 1 before forwarding, and other traffic for that flow MUST NOT be
forwarded along that interface.

When a flow is blocked or pruned, the circuit breaker MAY truncate
the Bandwidth Advertisement packet, keeping only the headers of the
packet containing the Bandwidth Advertisement before forwarding.

When a flow is pruned, the circuit-breaker MUST generate and forward
a Bandwidth Advertisement packet once per second with the "Blocked"
bit set when there are still downstream receivers connected.

In flows which are not circuit-broken but which would be circuit-
broken if the bandwidth warning threshold were the bandwidth limit,
the Danger bit MUST be set to 1 before forwarding. Both data and
control packets are forwarded for flows in this situation. The
"Danger" bit MAY be used by receivers to take early action to avoid
getting circuit-broken by shifting to a lower-bandwidth
representation, if available.

When a flow is in the "blocked" state on every egress interface, the circuit breaker MAY set the flow to "pruned" on the ingress interface and leave the channel upstream.

In addition to monitoring the advertised bandwidth, a CBACC circuit breaker or other assisting nodes in the network SHOULD monitor the observed bandwidth per flow, and SHOULD circuit break "overactive" flows, defined as those which exceed their CBACC maximum bandwidth commitment. A circuit breaker MAY perform constant monitoring on all flows, or MAY use load sharing techniques such as random selection or round robin to monitor only a certain subset of flows at a time.

When detecting overactive flows, circuit breakers MUST use techniques to avoid false positives due to transient upstream network conditions such as packet compression or occasional packet duplication. For example, using an average of bandwidth measurements over the prior 3 seconds would qualify, where a half-second window would not. (A full listing of reasonable false-positive avoidance techniques is out of scope for this document.)

[TBD: examples with network diagrams and bandwidths?] [TBD: some internal structure on this section. "wall of text" was some feedback]

6. Requirements from other building blocks

The sender needs to know the bandwidth, including any upcoming changes, at least 3 seconds in advance. There is no requirement on how building blocks define this functionality except on the packets on the wire--the advance knowledge might, for example, be implemented by buffering and pacing on the sending machine. Specifics of the sending bandwidth implementations are out of scope for this document, as it's intended to provide requirements that will be applicable to a broad range of possible implementations, including RTP and WEBRC.

7. IANA Considerations

This draft requests IANA to allocate an IPv6 packet header option number with the "action" bits set to "skip" and the "change" bit set to 1, as specified in [RFC2460] section 4.2. [TO BE REMOVED: This registration should take place at the following location: http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#extension-header.]

This draft also requests IANA to allocate an IPv4 packet header option number with the "copied" bit set and the option class "control", as specified in [RFC0791] section 3.1. [TO BE REMOVED: This registration should take place at the following location:

http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml#ip-
parameters-1.]

If those are deemed unacceptable, as an alternative with some
compromises described in Section 5.2.1, this draft instead requests
IANA to allocate a UDP destination port number. [TO BE REMOVED: This
registration should take place at the following location:
http://www.iana.org/assignments/service-names-port-numbers/service-
names-port-numbers.xhtml.]

8. Security Considerations

8.1. Forged Packets

Forged Bandwidth Advertisement packets that get accepted by CBACC
circuit breakers which dramatically over-report or under-report the
correct bandwidth would present a potential DoS against a CBACC flow,
by making the circuit breaker believe the flow exceeds the node's
capacity when over-reporting, or by letting the node notice an
apparent violation of the commitment to remain under the advertised
bandwidth when under-reporting.

Similarly, it is possible to forge a CBACC Bandwidth Advertisement
for a non-CBACC flow, which likewise may constitute a DoS against
that flow.

For multicast, attacker would have to be on-path in order to deliver
a forged packet to a CBACC circuit breaker, because the join's
reverse path propagation will only reach the sender on a legitimate
network path to its source address.

For unicast, it's a bigger problem, because ANY sender along path
that doesn't have RPF check BCP 38 [RFC2827] permits attack on the
flow via forged packet that substantially under-reports or over-
reports bandwidth.

For AMT tunnels, when RPF checks along a path to the gateway are not
present, nothing stops forged packets from being forwarded by the
gateway. If these packets contain CBACC control packets, it's
possible to inject a forged packet into the network downstream from
the gateway, combining the unicast hole with the multicast hole.
This is a vulnerability that should probably be addressed by a new
AMT version with some defense against forgery of data.

For IPSEC, since the Bandwidth Advertisement IP header option is
mutable, it's not protected by the IPSEC security services, so the
Bandwidth Advertisement can be forged for consumption by the circuit
breakers, even though the packet will be rejected by the end host

with the security association. This could mount a DoS via the
intermediate circuit-breakers by over-reporting or under-reporting
flow bandwidth, when processing CBACC traffic through untrusted
network paths.

The unicast vulnerabilities would be much mitigated by RPF checks as
recommended by BCP 38 [RFC2827] at every hop, or otherwise maintained
by the network. Absent such checks, cheap DoS vulnerabilities may be
present from any permissive network locations.

8.2. Overloading of Slow Paths

CBACC control packets are sent as part of the data stream so that
they traverse the same intermediate network nodes as the rest of the
data, but they also carry control information that must be processed
by certain nodes along that path.

This creates potential problems very similar to the problems with the
Router Alert IP option discussed in Section 3 of [RFC6398], where a
circuit-breaker might have a "fast path" for forwarding that can
handle a much higher traffic volume than the "slow path" necessary to
process CBACC control packets, which is potentially vulnerable to
overloading.

If a CBACC-compatible circuit breaker receives a high rate of CBACC
control packets, the circuit breaker MUST maintain network health for
other flows. A circuit-breaker MAY drop all packets, including all
CBACC control packets, for a flow in which more than 5 CBACC control
packets were received in less than a second. (This number is
intended to allow for moderate IP packet duplication and packet
compression by upstream routers, while still being slow enough for
handling of packets on the slow path.)

8.3. Overloading of State

Since CBACC flows require state, it may be possible for a set of
receivers and/or senders, possibly acting in concert, to generate
many flows in an attempt to overflow the circuit breakers' state
tables.

It is permissible for a network node to behave as a CBACC circuit
breaker for some CBACC flows while treating other CBACC flows as non-
CBACC, as part of a load balancing strategy for the network as a
whole, or simply as defense against this concern when the number of
monitored flows exceeds some threshold.

The same techniques described in section 3.1 of [RFC4609] can be used
to help mitigate this attack, for much the same reasons. It is

RECOMMENDED that network operators implement measures to mitigate
such attacks.

9. Acknowledgements

Many thanks to Devin Anderson and Ben Kaduk for detailed reviews and
many great suggestions. Thanks also to Cheng Jin, Scott Brown,
Miroslav Kaduk, and Bob Briscoe for their thoughtful contributions.

10. References

10.1. Normative References

[IEEE.754.1985]
          Institute of Electrical and Electronics Engineers,
          "Standard for Binary Floating-Point Arithmetic",
          IEEE Standard 754, August 1985.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791,
          DOI 10.17487/RFC0791, September 1981,
          <http://www.rfc-editor.org/info/rfc791>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119,
          DOI 10.17487/RFC2119, March 1997,
          <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6
          (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
          December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3048] Whetten, B., Vicisano, L., Kermode, R., Handley, M.,
          Floyd, S., and M. Luby, "Reliable Multicast Transport
          Building Blocks for One-to-Many Bulk-Data Transfer",
          RFC 3048, DOI 10.17487/RFC3048, January 2001,
          <http://www.rfc-editor.org/info/rfc3048>.

[RFC3738] Luby, M. and V. Goyal, "Wave and Equation Based Rate
          Control (WEBRC) Building Block", RFC 3738,
          DOI 10.17487/RFC3738, April 2004,
          <http://www.rfc-editor.org/info/rfc3738>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302,
          DOI 10.17487/RFC4302, December 2005,
          <http://www.rfc-editor.org/info/rfc4302>.

   [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4,
              ICMPv6, UDP, and TCP Headers", RFC 4727,
              DOI 10.17487/RFC4727, November 2006,
              <http://www.rfc-editor.org/info/rfc4727>.

   [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I.,
              Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent
              Multicast - Sparse Mode (PIM-SM): Protocol Specification
              (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March
              2016, <http://www.rfc-editor.org/info/rfc7761>.

10.2. Informative References

   [I-D.ietf-tsvwg-circuit-breaker]
              Fairhurst, G., "Network Transport Circuit Breakers",
              draft-ietf-tsvwg-circuit-breaker-15 (work in progress),
              April 2016.

   [I-D.ietf-tsvwg-rfc5405bis]
              Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage
              Guidelines", draft-ietf-tsvwg-rfc5405bis-19 (work in
              progress), October 2016.

   [PLM]   A.Legout, E.W.Biersack, Institut EURECOM, "Fast
              Convergence for Cumulative Layered Multicast Transmission
              Schemes", 1999.

   [RFC2357] Mankin, A., Romanow, A., Bradner, S., and V. Paxson, "IETF
              Criteria for Evaluating Reliable Multicast Transport and
              Application Protocols", RFC 2357, DOI 10.17487/RFC2357,
              June 1998, <http://www.rfc-editor.org/info/rfc2357>.

   [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P.
              Traina, "Generic Routing Encapsulation (GRE)", RFC 2784,
              DOI 10.17487/RFC2784, March 2000,
              <http://www.rfc-editor.org/info/rfc2784>.

   [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering:
              Defeating Denial of Service Attacks which employ IP Source
              Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827,
              May 2000, <http://www.rfc-editor.org/info/rfc2827>.

   [RFC2914] Floyd, S., "Congestion Control Principles", BCP 41,
              RFC 2914, DOI 10.17487/RFC2914, September 2000,
              <http://www.rfc-editor.org/info/rfc2914>.

   [RFC3269] Kermode, R. and L. Vicisano, "Author Guidelines for
              Reliable Multicast Transport (RMT) Building Blocks and
              Protocol Instantiation documents", RFC 3269,
              DOI 10.17487/RFC3269, April 2002,
              <http://www.rfc-editor.org/info/rfc3269>.

   [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
              Jacobson, "RTP: A Transport Protocol for Real-Time
              Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550,
              July 2003, <http://www.rfc-editor.org/info/rfc3550>.

   [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC
              Text on Security Considerations", BCP 72, RFC 3552,
              DOI 10.17487/RFC3552, July 2003,
              <http://www.rfc-editor.org/info/rfc3552>.

   [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering
              (TE) Extensions to OSPF Version 2", RFC 3630,
              DOI 10.17487/RFC3630, September 2003,
              <http://www.rfc-editor.org/info/rfc3630>.

   [RFC4609] Savola, P., Lehtonen, R., and D. Meyer, "Protocol
              Independent Multicast - Sparse Mode (PIM-SM) Multicast
              Routing Security Issues and Enhancements", RFC 4609,
              DOI 10.17487/RFC4609, October 2006,
              <http://www.rfc-editor.org/info/rfc4609>.

   [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an
              IANA Considerations Section in RFCs", BCP 26, RFC 5226,
              DOI 10.17487/RFC5226, May 2008,
              <http://www.rfc-editor.org/info/rfc5226>.

   [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and
              Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October
              2011, <http://www.rfc-editor.org/info/rfc6398>.

   [RFC6807] Farinacci, D., Shepherd, G., Venaas, S., and Y. Cai,
              "Population Count Extensions to Protocol Independent
              Multicast (PIM)", RFC 6807, DOI 10.17487/RFC6807, December
              2012, <http://www.rfc-editor.org/info/rfc6807>.

   [RFC7450] Bumgardner, G., "Automatic Multicast Tunneling", RFC 7450,
              DOI 10.17487/RFC7450, February 2015,
              <http://www.rfc-editor.org/info/rfc7450>.

Appendix A. Overjoining

   [I-D.ietf-tsvwg-rfc5405bis] describes several remedies for unicast
   congestion control under UDP, even though UDP does not itself provide
   congestion control. In general, any network node under congestion
   could in theory collect evidence that a unicast flow's sending rate
   is not responding to congestion, and would then be justified in
   circuit-breaking it.

   With multicast IP, the situation is different, especially in the
   presence of malicious receivers. A well-behaved sender using a
   receiver-controlled congestion scheme such as WEBRC does not reduce
   its send rate in response to congestion, instead relying on receivers
   to leave the appropriate multicast groups.

   This leads to a situation where, when a network accepts inter-domain
   multicast traffic, as long as there are senders somewhere in the
   world with aggregate bandwidth that exceeds a network's capacity,
   receivers in that network can join the flows and overflow the network
   capacity. A receiver controlled by an attacker could do this at the
   IGMP/MLD level without running the application layer protocol that
   participates in the receiver-controlled congestion control.

   A network might be able to detect and defend against the most naive
   version of such an attack by blocking end users that try to join too
   many flows at once. However, an attacker can achieve the same effect
   by joining a few high-bandwidth flows, if those exist anywhere, and
   an attacker that controls a few machines in a network can coordinate
   the receivers so they join disjoint sets of non-responsive sending
   flows.

   This scenario will produce congestion in a middle node in the network
   that can't be easily detected at the edge where the IGMP/MLD join is
   accepted. Thus, an attacker with a small set of machines in a target
   network can always trip a circuit breaker if present, or can induce
   excessive congestion among the bandwidth allocated to multicast.
   This problem gets worse as more multicast flows become available.

   This is a significant barrier to multicast adoption because there is
   no present defense which does not itself constitute a denial of
   service attack.

   Although the same can apply to non-responsive unicast traffic,
   network operators can assume that non-responsive sending flows are in
   violation of congestion control best practices, and can therefore cut
   off such flows. However, non-responsive multicast senders are likely
   to be well-behaved participants in receiver-controlled congestion
   control schemes.

However, receiver controlled congestion control schemes also show the
most promise for efficient massive scale content distribution via
multicast, provided network health can be ensured. Therefore,
mechanisms to mitigate overjoining attacks while still permitting
receiver-controlled congestion control are necessary. [TBD: this
whole section should be expanded and moved to a separate
informational draft]

TBD: network diagram

Figure 4

Author's Address

Jacob Holland
Akamai Technologies, Inc.
150 Broadway
Cambridge, Massachusetts 02142
USA

Phone: +1 617 444 3000
Email: jholland@akamai.com
URI: https://www.akamai.com/

                             Multicast Model
                draft-zhang-mboned-multicast-info-model-02

Abstract

   This document intents to provide a general and all-round multicast
   model, which tries to stand at a high level to take full advantages
   of existed multicast protocol models to control the multicast
   network, and guides the deployment of multicast service. And also,
   there will define several possible RPCs about how to interact between
   multicast info model and multicast protocol models. This multicast
   information model is mainly used by the management tools run by the
   network operators in order to manage, monitor and debug the network
   resources used to deliver multicast service, as well as gathering
   some data from the network.

Status of This Memo

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

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

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

   This Internet-Draft will expire on February 22, 2018.

Copyright Notice

Table of Contents

1. Introduction

   Currently, there are many multicast YANG models, such as PIM, MLD,
   and BIER and so on. But all these models are distributed in
   different working groups as separate files and focus on the protocol
   itself. Furthermore, they cannot describe a high-level multicast
   service required by network operators.

   This document intents to provide a general and all-round multicast
   model, which tries to stand at a high level to take full advantages
   of these aforementioned models to control the multicast network, and
   guides the deployment of multicast service.

   This multicast information model is mainly used by the management
   tools run by the network operators in order to manage, monitor and
   debug the network resources used to deliver multicast service, as
   well as gathering some data from the network.

```
                  +------------------------+
                  |      Multicast Info     |
                  +------------------------+
                   |       |          |
                   |       |          |
                   |  +---------+  +----------+
                   |  | EMS/NMS |  |Controller|
                   |  +---------+  +----------+
                   |       |          |
                   |       |          |
              +-------------------------------------------------+
              |            Network Element1.....N               |
              +-------------------------------------------------+
```
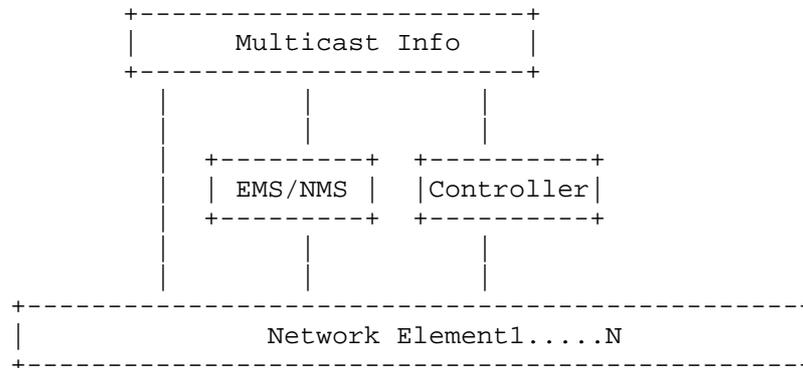
Figure 1: Example usage of Multicast Model

Detailly,in figure 1, there is an example of usage of this multicast
model. Network operators can input this model to a controller who is
responsible to translate the information and invoke the corresponding
protocol models into configurations to configure the network elements
through NETCONF/RESTCONF/CLI. Or network operators can input this
model to the EMS/NMS to manage the network elements or configure the
network elements directly. On the other hand, when the network
elements detect failure or some other changes, the network operators
can collect these kind of notifications through this model to assist
locating the exact failure and responsing immediatly. for example,
when the network element suffers a failure of one MVPN neighbor, it
can notify to the EMS/NMS or Controller or to other Multicast Model
management tool directly to let the network operator take actions
immediately.

Specifically, in section 3, it provides a human readability of the
whole multicast network through UML class diagram, which frames
different multicast components and correlates them in a readable
fashion. Then, based on this UML class diagram, there is
instantiated and detailed YANG model in Section 5.

In other words, this document does not define any specific protocol
model, instead, it depends on many existed multicast protocol models
and relates several multicast information together to fulfill
multicast service.

2. Design of the multicast model

   This model includes three layers: the multicast overlay, the
   transport layer and the multicast underlay information.

Multicast overlay defines the features of multicast flow, such as(vpnid, multicast source and multicast group) information, and (ingress-node,egress-nodes) nodes information. If the transport layer is BIER, there may define BIER information including (Subdomain, ingress-node BFR-id, egress-nodes BFR-id). In data center network, for fine-grained to gather the nodes belonging to the same virtual network, there may need VNI-related information to assist. If no (ingress-node, egress-nodes) information are defined directly, there may need overlay multicast signaling technology, such as MLD or MVPN, to collect these nodes information.

Multicast transport layer defines the type of transport technologies that can be used to forward multicast flow, including BIER forwarding type, MPLS forwarding type, or PIM forwarding type and so on. One or several transport technologies could be defined at the same time. As for the detailed parameters for each transport technology, this multicast information model can invoke the corresponding protocol model to define them.

Multicast underlay defines the type of underlay technologies, such as OSPF, ISIS, BGP, PIM or BABEL and so on. One or several underlay technologies could be defined at the same time. As for the specific parameters for each underlay technology, this multicast information model can depend the corresponding protocol model to configure them as well.

3. UML Class Diagram for Multicast Info Model

   The following is a UML diagram for Multicast Info Model.

```
                                       +----------------+
                                       | Multicast Info |
                                       +----------------+
                                          |    |    |Contain
                    +-----------------------+    |   +--------------------
------+
                    |                            |
     |
          +------------------+           +--------------------+      +----
----------------+
          | Multicast Overlay |          | Multicast Transport |      | Mul
ticast Underlay |
          +------------------+           +--------------------+      +----
----------------+
                   | |  |Contain               | | | | | invoke
 | | | | | invoke
          +--------+ |  +-------+          +----+ | | | +----+         +---
-+ | | | +----+
              |        |      |               |     | | |    |            |
   | | |     |
      +-------------+  |  +-------------+  +-----+  | | |  +------+    +------
+ | | |  +------+
      | Multi-Info   |  |  | Ing/Eg Nodes |  | PIM |  | | |  | MPLS |    | OSPF
| | | |   | PIM  |
      +-------------+  |  +-------------+  +-----+  | | |  +------+    +------
+ | | |  +------+
      |Group Address |  |  |Ingress Nodes |   +----+ | +-----+              +-
---+ | +-----+
      +-------------+  |  +--------------+       |     |    |              |
          |     |
      |Source Address|  |  |Egress Nodes  |  +----------+ | +--------+    +----
-+    |  +------+
      +-------------+  |  +--------------+  |Cisco Mode| | |BIER-TE |    |BABE
L|    |  | BGP  |
      |VPN Info      |  |          | relate  +----------+ | +--------+    +----
-+    |  +------+
      +-------------+  |          \|/                +----+
+----+
      |VNI Info      |  |  +----------------+              |
|
      +-------------+  |  | BIER Nodes Info|         +------+       +---
---+
                      |  +----------------+         | BIER |       | IS
IS |
          +--------+ |  | BFR-ID         |         +------+       +---
---+
              |        |  +----------------+
      +-------------+
      | Overlay Tech |
      +-------------+
      |    MLD       |
      +-------------+
      |    MVPN      |
      +-------------+
      |    BGP       |
      +-------------+
      | MLD-Snooping |
      +-------------+
```

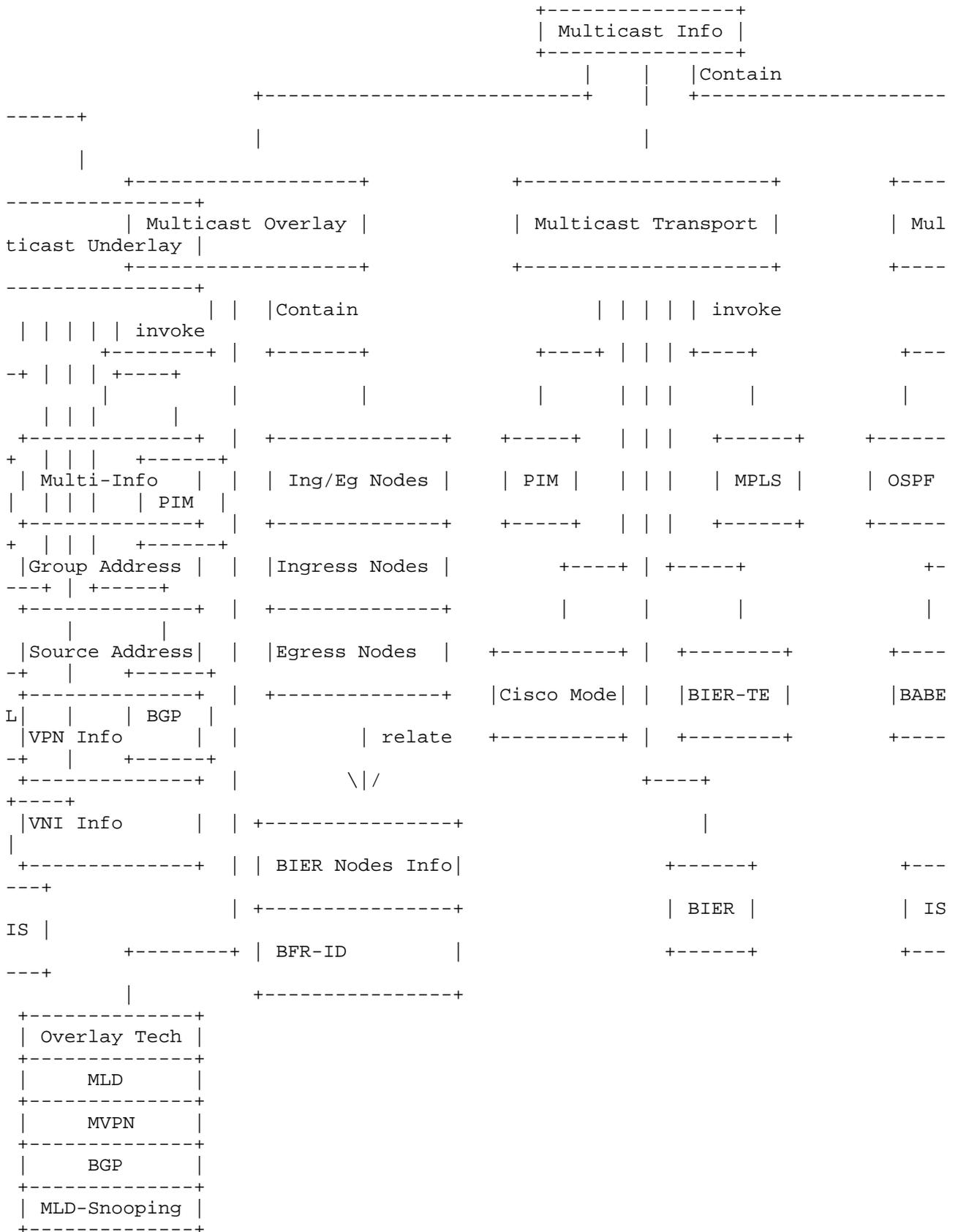   Figure 2: UML Class Diagram for Multicast Info Model

4. Model Structure

```
module: ietf-multicast-information
   +--rw multicast-information
      +--rw multicast-info* [vpn-id source-address source-wildcard group-addres
s group-wildcard vni-type vni-value]
         +--rw vpn-id                uint32
         +--rw source-address        inet:ip-address
         +--rw source-wildcard       uint8
```

```
         +--rw group-address         inet:ip-address
         +--rw group-wildcard        uint8
         +--rw vni-type              virtual-type
         +--rw vni-value             uint32
         +--rw multicast-overlay
         |  +--rw nodes-information
         |  |  +--rw ingress-node?   inet:ip-address
         |  |  +--rw egress-nodes* [egress-node]
         |  |     +--rw egress-node   inet:ip-address
         |  +--rw bier-information
         |  |  +--rw sub-domain?     sub-domain-id
         |  |  +--rw ingress-node?   bfr-id
         |  |  +--rw egress-nodes* [egress-node]
         |  |     +--rw egress-node   bfr-id
         |  +--rw overlay-technology
         |     +--rw (overlay-tech-type)?
         |        +--:(mld)
         |        +--:(mvpn)
         |        +--:(bgp)
         |        +--:(mld-snooping)
         +--rw multicast-transport
         |  +--rw bier
         |  |  +--rw sub-domain?       sub-domain-id
         |  |  +--rw (encap-type)?
         |  |  |  +--:(mpls)
         |  |  |  +--:(non-mpls)
         |  |  |  +--:(ipv6)
         |  |  +--rw bitstringlength?   uint16
         |  |  +--rw set-identifier?    si
         |  |  +--rw ecmp?              boolean
         |  |  +--rw frr?               boolean
         |  +--rw bier-te
         |  |  +--rw sub-domain?       sub-domain-id
         |  |  +--rw (encap-type)?
         |  |  |  +--:(mpls)
         |  |  |  +--:(non-mpls)
         |  |  +--rw bitstringlength?   uint16
         |  |  +--rw set-identifier?    si
         |  |  +--rw ecmp?              boolean
         |  |  +--rw frr?               boolean
         |  +--rw cisco-mode
         |  |  +--rw p-group?            inet:ip-address
         |  |  +--rw graceful-restart?   boolean
         |  |  +--rw bfd?                boolean
         |  +--rw mpls
         |  |  +--rw (mpls-tunnel-type)?
         |  |     +--:(mldp)
         |  |     |  +--rw mldp-tunnel-id?        uint32
```

```
        |  |     |  +--rw mldp-frr?            boolean
        |  |     |  +--rw mldp-backup-tunnel?  boolean
        |  |     +--:(p2mp-te)
        |  |        +--rw te-tunnel-id?        uint32
        |  |        +--rw te-frr?              boolean
        |  |        +--rw te-backup-tunnel?    boolean
        |  +--rw pim
        |     +--rw graceful-restart?   boolean
        |     +--rw bfd?                boolean
      +--rw multicast-underlay
         +--rw underlay-requirement?   boolean
         +--rw bgp
         +--rw ospf
         |  +--rw topology-id?   uint16
         +--rw isis
         |  +--rw topology-id?   uint16
         +--rw babel
         +--rw pim
```

5. Multicast Information Model

<CODE BEGINS> file "ietf-multicast-information.yang"
   module ietf-multicast-information {

   namespace "urn:ietf:params:xml:ns:yang:ietf-multicast-information";

   prefix multicast-info;

   import ietf-inet-types {
       prefix "inet";
   }

   organization " IETF MBONED(MBONE Deployment) Working Group";
   contact
       "WG List: <mailto:bier@ietf.org>
        WG Chair: Greg Shepherd
                  <mailto:gjshep@gmail.com>
        WG Chair: Leonard Giuliano
                  <mailto:lenny@juniper.net>

        Editor: Zheng Zhang
                  <mailto:zhang.zheng@zte.com.cn>
        Editor: Cui Wang
                  <mailto:wang.cui1@zte.com.cn>
        Editor: Ying Cheng
                  <mailto:chengying10@chinaunicom.cn>
        ";
```

```
    description
        "This module contains a collection of YANG definitions for
         managing multicast information.";

    revision 2017-08-20 {
        description
        "Add BGP and MLD-snooping overlay and BIER-TE transport.";
        reference "https://tools.ietf.org/html/draft-zhang-mboned-multicast-info
-model";
    }

    revision 2016-12-08 {
        description
        "Initial version.";
        reference "https://tools.ietf.org/html/draft-zhang-mboned-multicast-info
-model";
    }
/*feature*/
    grouping general-multicast {
        description "The general multicast address information.";
        leaf source-address {
            type inet:ip-address;
            description "The address of multicast source. The value set to zero
              means that the receiver interests in all source that relevant to
              one group.";
        }
        leaf source-wildcard {
            type uint8;
            description "The wildcard information of source.";
        }
        leaf group-address {
            type inet:ip-address;
            description "The address of multicast group.";
        }
        leaf group-wildcard {
            type uint8;
            description "The wildcard information of group.";
        }
    }

        grouping m-addr {
        description "The vpn multicast information.";
        leaf vpn-id {
            type uint32;
            description "The vpn-id of the multicast flow.
              If there is global instance, the vpnid value should be zero.";
        }
        uses general-multicast;
    }
```

```
     typedef virtual-type {
         type enumeration {
             enum "vxlan" {
                 description "The vxlan type.";
             }
             enum "virtual subnet" {
                 description "The nvgre type";
             }
             enum "vni" {
                 description "The geneve type";
             }
         }
         description "The collection of virtual network type.";
     }

     grouping multicast-nvo3 {
         description "The nvo3 multicast information.";
         leaf vni-type {
             type virtual-type;
              description "The type of virtual network identifier. Include the Vx
lan
               NVGRE and Geneve.";
         }
         leaf vni-value {
             type uint32;
             description "The value of Vxlan network identifier, virtual subnet I
D
             or virtual net identifier.";
         }
     }

     grouping multicast-feature {
         description
           "This group describe the different multicast information
            in various deployments.";
         uses m-addr;
         uses multicast-nvo3;
     }

     grouping ip-node {
         description "The IP information of multicast nodes.";
         leaf ingress-node {
             type inet:ip-address;
             description "The ingress node of multicast flow. Or the ingress
               node of MVPN and BIER. In MVPN, this is the address of ingress
               PE; in BIER, this is the BFR-prefix of ingress nodes.";
         }

         list egress-nodes {
             key "egress-node";
```

```
            description "This ID information of one adjacency.";

            leaf egress-node {
                type inet:ip-address;
                description
                  "The egress multicast nodes of multicast flow.
                  Or the egress node of MVPN and BIER. In MVPN, this is the
                  address of egress PE; in BIER, this is the BFR-prefix of
                  ingress nodes.";
            }
          }
        }
    }
    /* should import from BIER yang */
    typedef bfr-id {
        type uint16;
        description "The BFR id of nodes.";
    }

    typedef si {
        type uint16;
        description
          "The type for set identifier";
    }

    typedef sub-domain-id {
        type uint16;
        description
          "The type for sub-domain-id";
    }

    typedef bit-string {
        type uint16;
        description
          "The bit mask of one bitstring.";
    }

    grouping bier-node {
        description "The BIER information of multicast nodes.";
        leaf sub-domain {
            type sub-domain-id;
            description "The sub-domain that this multicast flow belongs to.";
        }
        leaf ingress-node {
            type bfr-id;
            description "The ingress node of multicast flow. This is the
              BFR-id of ingress nodes.";
        }
        list egress-nodes {
```

```
          key "egress-node";
          description "This ID information of one adjacency.";

          leaf egress-node {
              type bfr-id;
              description
                "The egress multicast nodes of multicast flow.
                 This is the BFR-id of egress nodes.";
          }
      }
  }

  grouping overlay-tech {
      description "The possible overlay technologies for multicast service.";
      choice overlay-tech-type {
          case mld {
              description "MLD technology is used for multicast overlay";
          }
          case mvpn {
              description "MVPN technology is used for multicast overlay";
          }
          case bgp {
              description "BGP technology is used for multicast overlay";
          }
          case mld-snooping {
              description "MLD snooping technology is used for multicast overl
ay";
          }
          description "The collection of multicast overlay technology";
      }
  }

  grouping multicast-overlay {
      description "The node information that connect the ingress multicast
        flow, and the nodes information that connect the egress multicast
        flow.";
      /*uses multicast-feature;*/
      container nodes-information {
          description "The ingress and egress nodes information.";
          uses ip-node;
      }
      container bier-information {
          description "The ingress and egress BIER nodes information.";
          uses bier-node;
      }
      container overlay-technology {
          description "The possible overlay technologies for multicast service
.";
          uses overlay-tech;
      }
```

```
    }


/*transport*/

    grouping transport-bier {
        description "The BIER transport information.";
        leaf sub-domain {
            type sub-domain-id;
            description "The subdomain id that this multicast flow belongs to.";
        }
        choice encap-type {
            case mpls {
                description "The BIER forwarding depend on mpls.";
            }
            case non-mpls {
                description "The BIER forwarding depend on non-mpls.";
            }
            case ipv6 {
                description "The BIER forwarding depend on IPv6.";
            }
            description "The encapsulation type in BIER.";
        }
         leaf bitstringlength {
            type uint16;
            description "The bitstringlength used by BIER forwarding.";
        }
        leaf set-identifier {
            type si;
            description "The set identifier used by this multicast flow.";
        }
        leaf ecmp {
            type boolean;
            description "The capability of ECMP.";
        }
        leaf frr {
            type boolean;
            description "The capability of fast re-route.";
        }
    }

        grouping transport-bier-te {
        description "The BIER-TE transport information.";
        leaf sub-domain {
            type sub-domain-id;
            description "The subdomain id that this multicast flow belongs to.";
        }
        choice encap-type {
```

```
            case mpls {
                description "The BIER-TE forwarding depend on mpls.";
            }
            case non-mpls {
                description "The BIER-TE forwarding depend on non-mpls.";
            }
            description "The encapsulation type in BIER-TE.";
        }
         leaf bitstringlength {
            type uint16;
            description "The bitstringlength used by BIER-TE forwarding.";
        }
        leaf set-identifier {
            type si;
            description "The set identifier used by this multicast flow, especia
lly in BIER TE.";
        }
        leaf ecmp {
            type boolean;
            description "The capability of ECMP.";
        }
        leaf frr {
            type boolean;
            description "The capability of fast re-route.";
        }
    }

    grouping transport-pim {
        description "The requirement information of pim transportion.";
        leaf graceful-restart {
            type boolean;
            description "If the graceful restart function should be supported.";
        }
        leaf bfd {
            type boolean;
            description "If the bfd function should be supported.";
        }
    }

    grouping mldp-tunnel-feature {
        description "The tunnel feature.";
        leaf mldp-tunnel-id {
            type uint32;
            description "The tunnel id that correspond this flow.";
        }
        leaf mldp-frr {
            type boolean;
            description "If the fast re-route function should be supported.";
        }
```

```
        leaf mldp-backup-tunnel {
            type boolean;
            description "If the backup tunnel function should be supported.";
        }
    }

    grouping p2mp-te-tunnel-feature {
        description "The tunnel feature.";
        leaf te-tunnel-id {
            type uint32;
            description "The tunnel id that correspond this flow.";
        }
        leaf te-frr {
            type boolean;
            description "If the fast re-route function should be supported.";
        }
        leaf te-backup-tunnel {
            type boolean;
            description "If the backup tunnel function should be supported.";
        }
    }

    /*typedef sub-domain-id {
        type uint16;
        description
          "The type for sub-domain-id";
    }*/

    grouping transport-mpls {
        description "The mpls transportion information.";
        choice mpls-tunnel-type {
            case mldp {
                uses mldp-tunnel-feature;
                description "The mldp tunnel.";
            }
            case p2mp-te {
                uses p2mp-te-tunnel-feature;
                description "The p2mp te tunnel.";
            }
            description "The collection types of mpls tunnels";
        }
    }

    grouping cisco-multicast {
        description "The Cisco MDT multicast information in RFC6037.";
        leaf p-group {
            type inet:ip-address;
            description "The address of p-group.";
```

```
        }
    }

        grouping transport-cisco-mode {
        description "The transport information of Cisco mode, RFC6037.";
        uses cisco-multicast;
        uses transport-pim;
    }

    grouping multicast-transport {
        description "The transport information of multicast service.";
        container bier {
            uses transport-bier;
            description "The transport technology is BIER.";
        }
        container bier-te {
            uses transport-bier-te;
            description "The transport technology is BIER-TE.";
        }
        container cisco-mode {
            uses transport-cisco-mode;
            description "The transport technology is cisco-mode.";
        }
        container mpls {
            uses transport-mpls;
            description "The transport technology is mpls.";
        }
        container pim {
            uses transport-pim;
            description "The transport technology is PIM.";
        }
    }

/*underlay*/
    grouping underlay-bgp {
        description "Underlay information of BGP.";
    }

    grouping underlay-ospf {
        description "Underlay information of OSPF.";
        leaf topology-id {
            type uint16;
            description "The topology id of ospf instance.";
        }
    }

    grouping underlay-isis {
        description "Underlay information of ISIS.";
```

```
        leaf topology-id {
            type uint16;
            description "The topology id of isis instance.";
        }
    }

    grouping underlay-babel {
        description "Underlay information of Babel.";
        /* If there are some necessary information should be defined? */
    }

    grouping underlay-pim {
        description "Underlay information of PIM.";
        /* If there are some necessary information should be defined? */
    }

    grouping multicast-underlay {
        description "The underlay information relevant multicast service.";
        leaf underlay-requirement {
            type boolean;
            description "If the underlay technology should be required.";
        }
        container bgp {
            uses underlay-bgp;
            description "The underlay technology is BGP.";
        }
        container ospf {
            uses underlay-ospf;
            description "The underlay technology is OSPF.";
        }
        container isis {
            uses underlay-isis;
            description "The underlay technology is ISIS.";
        }
        container babel {
            uses underlay-babel;
            description "The underlay technology is Babel.";
        }
        container pim {
            uses underlay-pim;
            description "The underlay technology is PIM.";
        }
    }

    container multicast-information {
        description "The model of multicast service. Include overlay, transport
and underlay.";

        list multicast-info{
```

```
            key "vpn-id source-address source-wildcard group-address group-wildc
ard vni-type vni-value";
            uses multicast-feature;
            description "The detail multicast information.";

            container multicast-overlay {
                description "The overlay information of multicast service.";
                uses multicast-overlay;
            }
            container multicast-transport {
                description "The transportion of multicast service.";
                uses multicast-transport;
            }
            container multicast-underlay {
                description "The underlay of multicast service.";
                uses multicast-underlay;
            }
        }
    }
}
<CODE ENDS>
```

6. Notifications

    TBD.

7. Acknowledgements

    The authors would like to thank Stig Venaas, Jake Holland for their
    valuable comments and suggestions.

8. Normative References

    [I-D.ietf-bier-architecture]
              Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and
              S. Aldrin, "Multicast using Bit Index Explicit
              Replication", draft-ietf-bier-architecture-07 (work in
              progress), June 2017.

    [I-D.ietf-bier-bier-yang]
              Chen, R., hu, f., Zhang, Z., dai.xianxian@zte.com.cn, d.,
              and M. Sivakumar, "YANG Data Model for BIER Protocol",
              draft-ietf-bier-bier-yang-02 (work in progress), August
              2017.

   [I-D.ietf-pim-yang]
              Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu,
              Y., and f. hu, "A YANG data model for Protocol-Independent
              Multicast (PIM)", draft-ietf-pim-yang-08 (work in
              progress), April 2017.

   [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF)", RFC 6020,
              DOI 10.17487/RFC6020, October 2010, <https://www.rfc-
              editor.org/info/rfc6020>.

   [RFC6037] Rosen, E., Ed., Cai, Y., Ed., and IJ. Wijnands, "Cisco
              Systems' Solution for Multicast in BGP/MPLS IP VPNs",
              RFC 6037, DOI 10.17487/RFC6037, October 2010,
              <https://www.rfc-editor.org/info/rfc6037>.

   [RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG
              Data Model Documents", RFC 6087, DOI 10.17487/RFC6087,
              January 2011, <https://www.rfc-editor.org/info/rfc6087>.

   [RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/
              BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February
              2012, <https://www.rfc-editor.org/info/rfc6513>.

   [RFC7223] Bjorklund, M., "A YANG Data Model for Interface
              Management", RFC 7223, DOI 10.17487/RFC7223, May 2014,
              <https://www.rfc-editor.org/info/rfc7223>.

Authors' Addresses

   Zheng Zhang
   ZTE Corporation
   No. 50 Software Ave, Yuhuatai Distinct
   Nanjing
   China

   Email: zhang.zheng@zte.com.cn

   Cui(Linda) Wang
   ZTE Corporation
   No. 50 Software Ave, Yuhuatai Distinct
   Nanjing
   China

   Email: lindawangjoy@gmail.com

Ying Cheng
China Unicom
Beijing
China

Email: chengying10@chinaunicom.cn