

Mboned
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

J. Holland
K. Rose
Akamai Technologies, Inc.
7 March 2022

Asymmetric Manifest Based Integrity
draft-ietf-mboned-ambi-03

Abstract

This document defines Asymmetric Manifest-Based Integrity (AMBI). AMBI allows each receiver or forwarder of a stream of multicast packets to check the integrity of the contents of each packet in the data stream. AMBI operates by passing cryptographically verifiable hashes of the data packets inside manifest messages, and sending the manifests over authenticated out-of-band communication channels.

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 8 September 2022.

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
1.1. Comparison with TESLA	4
1.2. Terminology	4
1.3. Notes for Contributors and Reviewers	4
1.3.1. Venues for Contribution and Discussion	5
1.3.2. Non-obvious doc choices	5
2. Threat Model	6
2.1. Security Anchors	6
2.1.1. Alternatives and Their Requirements	7
2.2. System Security	8
3. Protocol Operation	8
3.1. Overview	8
3.2. Buffering of Packets and Digests	9
3.2.1. Validation Windows	10
3.2.2. Preserving Inter-packet Gap	11
3.3. Packet Digests	11
3.3.1. Digest Profile	11
3.3.2. Pseudoheader	13
3.4. Manifests	14
3.4.1. Manifest Layout	14
3.5. Transitioning to Other Manifest Streams	18
4. Transport Considerations	18
4.1. Overview	18
4.2. HTTPS	19
4.3. TLS	19
4.4. DTLS	19
5. Examples	20
6. YANG Module	20
6.1. Tree Diagram	20
6.2. Module	20
7. IANA Considerations	24
7.1. The YANG Module Names Registry	24
7.2. The XML Registry	24
7.3. Media Type	24
7.4. URI Schemes	25
7.4.1. TLS	25
7.4.2. DTLS	25
8. Security Considerations	25
8.1. Predictable Packets	25
8.2. Attacks on Side Applications	25
9. Acknowledgements	26
10. References	26
10.1. Normative References	26
10.2. Informative References	27
Authors' Addresses	29

1. Introduction

Multicast transport poses security problems that are not easily addressed by the same security mechanisms used for unicast transport.

The "Introduction" sections of the documents describing TESLA [RFC4082], and TESLA in SRTP [RFC4383], and TESLA with ALC and NORM [RFC5776] present excellent overviews of the challenges unique to multicast authentication for use cases like wide scale software or video distribution with a high data transfer rate. The challenges are briefly summarized here:

- * A MAC based on a symmetric shared secret cannot be used because each packet has multiple receivers that do not trust each other, and using a symmetric shared secret exposes the same secret to each receiver.
- * Asymmetric per-packet signatures can handle only very low bit-rates because of the transport and computational overhead associated with signature transmission and verification.
- * An asymmetric signature of a larger message comprising multiple packets requires reliable receipt of all such packets, something that cannot be guaranteed in a timely manner even for protocols that do provide reliable delivery, and the retransmission of which may anyway exceed the useful lifetime for data formats that can otherwise tolerate some degree of loss.

Asymmetric Manifest-Based Integrity (AMBI) defines a method for receivers or middle boxes to cryptographically authenticate and verify the integrity of a stream of packets by comparing the data packets to a stream of packet "manifests" (described in Section 3.4) received via an out-of-band communication channel that provides authentication and verifiable integrity.

Each manifest contains a message digest (described in Section 3.3) for each packet in a sequence of packets from the data stream, hereafter called a "packet digest". The packet digest incorporates a cryptographic hash of the packet contents and some identifying data from the packet, according to a defined digest profile for the data stream.

Upon receipt of a packet digest inside a manifest conveyed in a secure channel and verification that the packet digest of a received data packet matches, the receiver has proof of the integrity of the contents of the data packet corresponding to that digest.

This document defines the "ietf-ambi" YANG [RFC7950] model in Section 6 as an extension of the "ietf-dorms" model defined in [I-D.draft-ietf-mboned-dorms]. Also defined are new URI schemes for transport of manifests over TLS or DTLS, and a new media type for transport of manifests over HTTPS. The encodings for these are defined in Section 4.

1.1. Comparison with TESLA

AMBI and TESLA [RFC4082] and [RFC5776] attempt to achieve a similar goal of authenticating the integrity of streams of multicast packets. AMBI imposes a higher overhead than TESLA imposes, as measured in the amount of extra data required. In exchange, AMBI relaxes the requirement for establishing an upper bound on clock synchronization between sender and receiver, and allows for the use case of authenticating multicast traffic before forwarding it through the network, while also allowing receivers to authenticate the same traffic. By contrast, this is not possible with TESLA because the data packets can't be authenticated until a key is disclosed, so either the middlebox has to forward data packets without first authenticating them so that the receiver has them prior to key disclosure, or the middlebox has to hold packets until the key is disclosed, at which point the receiver can no longer establish their authenticity.

The other new capability is that because AMBI provides authentication information out of band, authentication can be retrofitted into some pre-existing deployments without changing the protocol of the data packets under some restrictions outlined in Section 8. By contrast, TESLA requires a MAC to be added to each authenticated message.

1.2. 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 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Notes for Contributors and Reviewers

Note to RFC Editor: Please remove this section and its subsections before publication.

This section is to provide references to make it easier to review the development and discussion on the draft so far.

1.3.1. Venues for Contribution and Discussion

This document is in the Github repository at:

<https://github.com/GrumpyOldTroll/ietf-dorms-cluster>
(<https://github.com/GrumpyOldTroll/ietf-dorms-cluster>)

Readers are welcome to open issues and send pull requests for this document.

Please note that contributions may be merged and substantially edited, and as a reminder, please carefully consider the Note Well before contributing: <https://datatracker.ietf.org/submit/note-well/>
(<https://datatracker.ietf.org/submit/note-well/>)

Substantial discussion of this document should take place on the MBONED working group mailing list (mboned@ietf.org).

- * Join: <https://www.ietf.org/mailman/listinfo/mboned>
(<https://www.ietf.org/mailman/listinfo/mboned>)
- * Search: <https://mailarchive.ietf.org/arch/browse/mboned/>
(<https://mailarchive.ietf.org/arch/browse/mboned/>)

1.3.2. Non-obvious doc choices

- * TBD: we need a way to assert that we provide the full set of packets for an (S,G) on all UDP ports and non-UDP protocols. Naively authenticating UDP for specified ports and ignoring other ports means that an attacker could attack a separate UDP port by injecting traffic directed at it, potentially hitting a different application that listens on 0.0.0.0, so an (S,G) with legitimately authenticated UDP traffic on one port could be used to transport UDP-based attacks to apps on another port or protocol unless they are firewalled. Passing traffic for an (S,G) subscription would open a new channel to such targets that otherwise would not be reachable from the internet for users behind e.g. a CPE with nat or connection-state-based firewalling.
- * Dropped intent to support DTLS+FECFRAME in this spec because RFC 6363 seems incomprehensible on a few points, most notably demux strategy between repair and source ADUs, which as written seems to require specifying another layer. So support for this will have to be a later separate RFC. However, for future extensibility made manifest-stream into a list instead of a leaf-list so that it can be an augment target for a later YANG extension with FEC selection from the likewise-very-confusing semi-overlapping registries at <https://www.iana.org/assignments/rmt-fec-parameters/>

rmt-fec-parameters.xhtml (<https://www.iana.org/assignments/rmt-fec-parameters/rmt-fec-parameters.xhtml>) defined by RFCs 5052 and 6363. See also RFC 6363, RFC 6681, and RFC 6865

2. Threat Model

AMBI is designed to operate over the internet, under the Internet Threat Model described in [RFC3552].

AMBI aims to provide Data Integrity for a multicast data stream, building on the security anchors described in Section 2.1 to do so. The aim is to enable receivers to subscribe to and receive multicast packets from a trusted sender without damage to the Systems Security (Section 2.3 of [RFC3552]) for those receivers or other entities.

Thus, we assume there might be attackers on-path or off-path with the capability to inject or modify packets, but that the attackers have not compromised the sender or discovered any of the sender's secret keys. We assume that an attacker may have compromised some receivers of the multicast traffic, but still aim to provide the above security properties for receivers that have not been compromised.

Those sending multicast traffic to receivers that include untrusted receivers should avoid transmitting sensitive information that requires strong confidentiality guarantees, due to the risk of compromise from those receivers. Since multicast transmits the same packets to potentially many receivers, in the presence of potentially compromised receivers confidentiality of the content cannot be assured.

However, any protocol that provides encryption of the packet data before generating the packet digest can provide confidentiality against on-path passive observers who do not possess the decryption key. This level of confidentiality can be provided by any such protocols without impact on AMBI's operation.

2.1. Security Anchors

Establishing the desired security properties for the multicast data packets relies on secure delivery of some other information:

- * Secured unicast connections (providing Data Integrity) to one or more trusted DORMS [I-D.draft-ietf-mboned-dorms] servers that use the AMBI extensions to the DORMS YANG model as defined in Section 6
- * Secure delivery (providing Data Integrity) of a stream of manifests (Section 3.4)

The secured unicast connection to the DORMS server provides the Peer Entity authentication of the DORMS server that's needed to establish the Data Integrity of the data it sends.

Note that DORMS provides a method for using DNS to bootstrap discovery of the DORMS server. In contexts where secure DNS lookup cannot be provided, it's still possible to establish a secure connection to a trusted DORMS server as long as the trusted DORMS server's hostname is known to the receivers (removing the need to use DNS for that discovery). Once the server name is known, the ordinary certificate verification of that hostname while establishing a secure https connection provides the needed security properties to anchor the rest.

Receiving unauthenticated data packets and knowing how to generate packet digests from the manifest profile provided by the AMBI extensions in the DORMS metadata allows the receiver to generate packet digests based on the contents of the received packet, which can be compared against the packet digests that were securely received.

Comparing the digests and finding the same answer then provides Data Integrity for the data packets that relies on one more property of the digest generation algorithm:

- * the difficulty of generating a collision for the packet digests contained in the manifest.

Taken together, successful validation of the multicast data packets proves within the above constraints that someone with control of the manifest URI streams provided by the DORMS server has verified the sending of the packets corresponding to the digests sent in that stream of manifests.

2.1.1. Alternatives and Their Requirements

Other protocols that can provide authentication could also be used for manifest delivery if defined later in another specification. For example a protocol that asymmetrically signs each packet, as the one defined in Section 3 of [RFC6584] does, would be a viable candidate for a delivery protocol for manifests that could be delivered over a multicast transport, which could have some important scalability benefits.

Other methods of securely transmitting metadata equivalent to the metadata provided by the "ietf-ambi" YANG model could also be used to provide the same security guarantees with the manifest channels. Defining other such possibilities is out of scope for this document.

2.2. System Security

By providing the means to authenticate multicast packets, AMBI aims to avoid giving attackers who can inject or modify packets the ability to attack application vulnerabilities that might be possible to exercise if those applications process the attack traffic. Many of the entries in the Common Vulnerabilities and Exposures (CVE) list at [CVE] (an extensive industry-wide database of software vulnerabilities) have documented a variety of system security problems that can result from maliciously generated UDP packets.

TBD: Fold in a mention of how off-path attacks are possible from most places on the internet for interdomain multicast over AMT at an ingest point, and how the multicast fanout downstream of that can make it a good target if multicast sees more use. A diagram plus a cleaned-up version of the on-list explanation here is probably appropriate: <https://mailarchive.ietf.org/arch/msg/mboned/CG9FLjPwuno3MtvYvgNcD5p69I4/> (<https://mailarchive.ietf.org/arch/msg/mboned/CG9FLjPwuno3MtvYvgNcD5p69I4/>). Nightmare scenario is zero-day RCE by off-path attacker that takes over a significant number of the devices watching a major sports event.

See also work-in-progress: <https://squarooticus.github.io/draft-krose-multicast-security/draft-krose-multicast-security.html> (<https://squarooticus.github.io/draft-krose-multicast-security/draft-krose-multicast-security.html>)

3. Protocol Operation

3.1. Overview

In order to authenticate a data packet, AMBI receivers need to hold these three pieces of information at the same time:

- * the data packet
- * an authenticated manifest containing the packet digest for the data packet
- * a digest profile defining the transformation from the data packet to its packet digest

The manifests are delivered as a stream of manifests over an authenticated data channel. Manifest contents MUST be authenticated before they can be used to authenticate data packets.

The manifest stream is composed of an ordered sequence of manifests that each contain an ordered sequence of packet digests, corresponding to the original packets as sent from their origin, in the same order.

Note that a manifest contains potentially many packet digests, and its size can be tuned to fit within a convenient PDU (Protocol Data Unit) of the manifest transport stream. By doing so, many packet digests for the multicast data stream can be delivered per packet of the manifest transport. The intent is that even with unicast-based manifest transport, multicast-style efficiencies of scale can still be realized with only a relatively small unicast overhead, when manifests use a unicast transport.

3.2. Buffering of Packets and Digests

Using different communication channels for the manifest stream and the data stream introduces a possibility of desynchronization in the timing of the received data between the different channels, so receivers hold data packets and packet digests from the manifest stream in buffers for some duration while awaiting the arrival of their counterparts.

While holding a data packet, if the corresponding packet digest for that packet arrives in the secured manifest stream, the data packet is authenticated.

While holding an authenticated packet digest, if the corresponding data packet arrives with a matching packet digest, the data packet is authenticated.

Authenticating a data packet consumes one packet digest and prevents re-learning a digest for the same sequence number with a hold-down time equal to the hold time for packet digests. The hold-down is necessary because a different manifest can send a duplicate packet digest for the same packet sequence number, either when repeating of packet digests is used for resilience to loss or when rotating authentication keys, so re-learning the packet digest could allow a replay of a data packet. After authenticating a packet, the digest and any future digests for the same data packet remain consumed if it has been used to authenticate a data packet, ignoring repeated digests for the same sequence number until after the holddown timer expires.

Once the data packet is authenticated it can be further processed by the receiving application or forwarded through the receiving network.

If the receiver's hold duration for a data packet expires without authenticating the packet, the packet SHOULD be dropped as unauthenticated. If the hold duration of a manifest expires, packet digests last received in that manifest MUST be discarded.

When multiple digests for the same packet sequence number are received, the latest received time for an authenticated packet digest should be used for the expiration time.

3.2.1. Validation Windows

Since packet digests are usually smaller than the data packets, it's RECOMMENDED that senders generate and send manifests with timing such that the packet digests in a manifest will typically be received by subscribed receivers before the data packets corresponding to those digests are received.

This strategy reduces the buffering requirements at receivers, at the cost of introducing some buffering of data packets at the sender, since data packets are generated before their packet digests can be added to manifests.

The RECOMMENDED default hold times at receivers are:

- * 2 seconds for data packets
- * 10 seconds for packet digests

The sender MAY recommend different values for specific data streams, in order to tune different data streams for different performance goals. The YANG model in Section 6 provides a mechanism for senders to communicate the sender's recommendation for buffering durations. These parameters are "data-hold-time" and "digest-hold-time", expressed in milliseconds.

Receivers MAY deviate from the values recommended by the sender for a variety of reasons, including their own memory constraints or local administrative configuration (for example, it might improve user experience in some situations to hold packets longer than the server recommended when there are receiver-specific delays in the manifest stream that exceed the server's expectations). Decreasing the buffering durations recommended by the server increases the risk of losing packets, but can be an appropriate tradeoff for specific network conditions and hardware or memory constraints on some devices.

Receivers SHOULD follow the recommendations for hold times provided by the sender (including the default values from the YANG model when unspecified), subject to their capabilities and any administratively configured overrides at the receiver.

3.2.2. Preserving Inter-packet Gap

It's RECOMMENDED that middle boxes forwarding buffered data packets preserve the inter-packet gap between packets in the same data stream, and that receiving libraries that perform AMBI-based authentication provide mechanisms to expose the network arrival times of packets to applications.

The purpose for this recommendation is to preserve the capability of receivers to use techniques for available bandwidth detection or network congestion based on observation of packet times and packet dispersal, making use of known patterns in the sending. Examples of such techniques include those described in [PathChirp], [PathRate], and [WEBRC].

Note that this recommendation SHOULD NOT prevent the transmission of an authenticated packet because the prior packet is unauthenticated. This recommendation only asks implementations to delay the transmission of an authenticated packet to correspond to the interpacket gap if an authenticated packet was previously transmitted and the authentication of the subsequent packet would otherwise burst the packets more quickly.

This does not prevent the transmission of packets out of order according to their order of authentication, only the timing of packets that are transmitted, after authentication, in the same order they were received.

For receiver applications, the time that the original packet was received from the network SHOULD be made available to the receiving application.

3.3. Packet Digests

3.3.1. Digest Profile

A packet digest is a message digest for a data packet, built according to a digest profile defined by the sender.

The digest profile is defined by the sender, and specifies:

1. A cryptographically secure hash algorithm (REQUIRED)

2. A manifest stream identifier

3. Whether to hash the IP payload or the UDP payload. (see Section 3.3.1.1)

The hash algorithm is applied to a pseudoheader followed by the packet payload, as determined by the digest profile. The computed hash value is the packet digest.

TBD: As recommended by <https://tools.ietf.org/html/rfc7696#section-2.2> (<https://tools.ietf.org/html/rfc7696#section-2.2>), a companion document containing the mandatory-to-implement cipher suite should also be published separately and referenced by this document.

3.3.1.1. Payload Type

3.3.1.1.1. UDP vs. IP payload validation

When the manifest definition is at the UDP layer, it applies only to packets with IP protocol of UDP (0x11) and the payload used for calculating the packet digest includes only the UDP payload with length as the number of UDP payload octets, as calculated by subtracting the size of the UDP header from the UDP payload length.

When the manifest definition is at the IP layer, the payload used for calculating the packet digest includes the full IP payload of the data packets in the (S,G). There is no restriction on the IP protocols that can be authenticated. The length field in the pseudoheader is calculated by subtracting the IP Header Length from the IP length, and is equal to the number of octets in the payload for the digest calculation.

3.3.1.1.2. Motivation

Full IP payloads often aren't available to receivers without extra privileges on end user operating systems, so it's useful to provide a way to authenticate only the UDP payload, which is often the only portion of the packet available to many receiving applications.

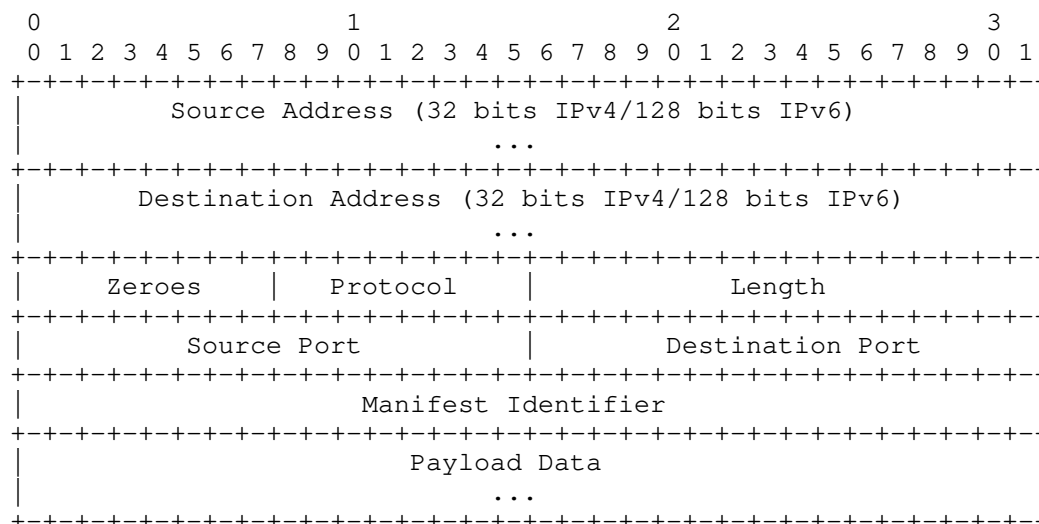
However, for some use cases a full IP payload is appropriate. For example, when retrofitting some existing protocols, some packets may be predictable or frequently repeated. Use of an IPSec Authentication Header [RFC4302] is one way to disambiguate such packets. Even though the shared secret means the Authentication Header can't itself be used to authenticate the packet contents, the sequence number in the Authentication Header can ensure that specific packets are not repeated at the IP layer, and so it's useful for AMBI to have the capability to authenticate such packets.

Another example: some services might need to authenticate the UDP options [I-D.ietf-tsvwg-udp-options]. When using the UDP payload, the UDP options would not be part of the authenticated payload, but would be included when using the IP payload type.

Lastly, since (S,G) subscription operates at the IP layer, it's possible that some non-UDP protocols will need to be authenticated, and the IP layer allows for this. However, most user-space transport applications are expected to use the UDP layer authentication.

3.3.2. Pseudoheader

When calculating the hash for the packet digest, the hash algorithm is applied to a pseudoheader followed by the payload from the packet. The complete sequence of octets used to calculate the hash is structured as follows:



3.3.2.1. Source Address

The IPv4 or IPv6 source address of the packet.

3.3.2.2. Destination Address

The IPv4 or IPv6 destination address of the packet.

3.3.2.3. Zeroes

All bits set to 0.

3.3.2.4. Protocol

The IP Protocol field from IPv4, or the Next Header field for IPv6. When using UDP-layer authentication, this value is always UDP (0x11) but for IP-layer authentication it can vary per-packet.

3.3.2.5. Length

The length in octets of the Payload Data field, expressed as an unsigned 16-bit integer.

3.3.2.6. Source Port

The source port of the packet. Zeroes if using IP-layer authentication for a non-UDP protocol.

3.3.2.7. Destination Port

The UDP destination port of the packet. Zeroes if using IP-layer authentication for a non-UDP protocol.

3.3.2.8. Manifest Identifier

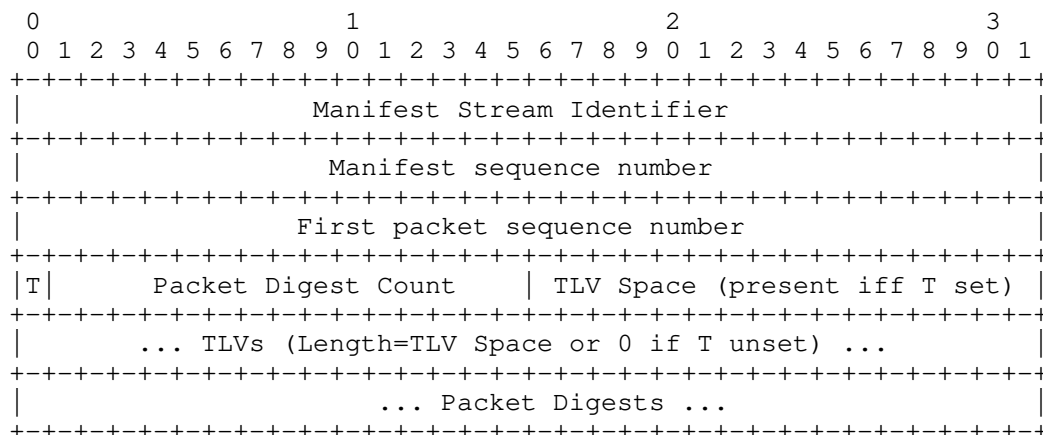
The 32-bit identifier for the manifest stream.

3.3.2.9. Payload Data

The payload data includes either the IP payload or the UDP payload, as indicated by the digest profile.

3.4. Manifests

3.4.1. Manifest Layout



3.4.1.1. Manifest Stream Identifier

A 32-bit unsigned integer chosen by the sender. This value **MUST** be equal to the "id" field in the manifest-stream in the "ietf-ambi" model. If a manifest is seen that does not have the expected value from the metadata provided for the manifest, the receiver **MUST** stop processing this manifest and disconnect from this manifest stream. It **MAY** reconnect with an exponential backoff starting at 1s, or it **MAY** connect to an alternative manifest stream if one is known.

3.4.1.2. Manifest Sequence Number

A monotonically increasing 32-bit unsigned integer. Each manifest sent by the sender increases this value by 1. On overflow it wraps to 0.

It's **RECOMMENDED** to expire the manifest stream and start a new stream for the data packets before a sequence number wrap is necessary.

3.4.1.3. First Packet Sequence Number

A monotonically increasing 32-bit unsigned integer. Each packet in the data stream increases this value by 1.

It's **RECOMMENDED** to expire the manifest stream and start a new stream for the data packets before a sequence number wrap is necessary.

Note: for redundancy, especially if using a manifest stream with unreliable transport, successive manifests **MAY** provide duplicates of the same packet digest with the same packet sequence number, using overlapping sets of packet sequence numbers. When received, these reset the hold timer for the listed packet digests.

3.4.1.4. T bit (TLVs Present)

If 1, this indicates the TLV Length and TLV space fields are present.
If 0, this indicates neither field is present.

3.4.1.5. Packet Digest Count

A 15-bit unsigned integer equal to the count of packet digests in the manifest.

3.4.1.6. TLV Space

A 16-bit unsigned integer with the length of the TLVs section.

3.4.1.7. TLVs

These are Type-Length-Value blocks, back to back. These may be extended by future specifications.

These are composed of 3 fields:

- * Type: an 8-bit unsigned integer indicating the type. Type values in 0-127 have an 8-bit length, and type values in 128-255 have a 16-bit length.
- * Length: a 8-bit or 16-bit unsigned integer indicating the length of the value
- * Value: a value with semantics defined by the Type field.

Defined values:

Type	Name	Value
0	Pad	Length can be 0-255. Value is filled with 0 and ignored by receiver.
128	Refresh Deadline	Length MUST be 2. Value is a 16-bit unsigned integer number of seconds. When this field is absent or zero, it means the current digest profile for the current manifest stream is stable. A nonzero value means the authentication is transitioning to a new manifest stream, and the set of digest profiles SHOULD be refreshed by receivers before this much time has elapsed in order to avoid a disruption. See Section 3.5.

Table 1

1-120 and 129-248 are unassigned 121-127 and 249-255 are reserved for experiments

Any unknown values MUST be skipped and ignored by the receiver, using the Length field to skip.

The total size of the manifest in octets is exactly equal to:

Size of digests * packet count + 14 if T is 0
Size of digests * packet count + 16 + TLV Length if T is 1

The total size of the TLV space is exactly equal to:

(2 + Length) summed for each TLV

The total size of the TLV space MUST exactly equal TLV Length. If the TLV space exceeds the TLV Length, the receiver MUST disconnect, and behave as if the Manifest Stream Identifier was wrong. This state indicates a failed decoding of the TLV space.

3.4.1.8. Packet Digests

Packet digests appended one after the other, aligned to 8-bit boundaries with 0-bit padding at the end if the bit length of the digests are not multiples of 8 bits.

3.5. Transitioning to Other Manifest Streams

It's possible for multiple manifest streams authenticating the same data stream to be active at the same time. The different manifest streams can have different hash algorithms, manifest ids, and current packet sequence numbers for the same data stream. These result in different sets of packet digests for the same data packets, one digest per packet per digest profile.

It's necessary sometimes to transition gracefully from one manifest stream to another. The Refresh Deadline TLV from the manifest is used to signal to receivers the need to transition.

When a receiver gets a nonzero refresh deadline in a manifest the sender SHOULD have an alternate manifest stream ready and available, and the receiver SHOULD learn the alternate manifest stream, join the new one, and leave the old one before the number of seconds given in the refresh deadline. After the refresh deadline has expired, a manifest stream MAY stop transmitting and close connections from the server side. When multiple manifest-streams are provided in the metadata, all or all but one SHOULD contain an expire-time, and new or refreshing receivers SHOULD choose a manifest stream without an expire-time, or with the latest expire-time if all manifests have an expire-time.

The receivers SHOULD start the refresh after a random time delay between now and one half the number of seconds in the deadline field after the first manifest they receive containing a nonzero refresh deadline. This time delay is to desynchronize the refresh attempts in order to spread the spike of load on the DORMS server while changing manifest profiles during a large multicast event.

4. Transport Considerations

4.1. Overview

AMBI manifests MUST be authenticated, but any transport protocol providing authentication can be used. This section discusses several viable options for the use of an authenticating transport, and some associated design considerations.

TBD: add ALTA to the list when and if it gets further along [I-D.draft-krose-mboned-alta]. Sending an authenticatable multicast stream (instead of the below unicast-based proposals) is a worthwhile goal, else a 1% unicast authentication overhead becomes a new unicast limit to the scalability.

TBD: probably should add quic also? Or maybe https is sufficient?

TBD: add a recommendation about scalability, like with DORMS, when using a unicast hash stream. CDN or other kind of fanout solution that can scale the delivery, and still generally hit the time window.

4.2. HTTPS

This document defines a new media type 'application/ambi' for use with HTTPS. URIs in the manifest-transport list with the scheme 'https' use this transport.

An HTTPS stream carrying the 'application/ambi' media type is composed of a sequence of binary AMBI manifests, sent back to back in the payload body (payload body is defined in Section 3.3 of [RFC7230]).

Complete packet digests from partially received manifests MAY be used by the receiver for authentication of data packets from the multicast channel, even if the full manifest is not yet delivered.

4.3. TLS

This document defines the new uri scheme 'ambi+tls' for use with TLS [RFC8446]. URIs in the manifest-transport list with the scheme 'ambi+tls' use this transport.

A TLS stream carrying AMBI manifests is composed of a sequence of binary AMBI manifests, transmitted back to back.

Complete packet Digests from partially received manifests MAY be used by the receiver for authentication, even if the full manifest is not yet delivered.

4.4. DTLS

This document defines the new uri scheme 'ambi+dtls' for use with DTLS [RFC6347].

Manifests transported with DTLS have the tradeoff (relative to TLS or HTTPS) that they might be lost and not retransmitted or reordered, but they will not cause head-of-line blocking or delay in processing data packets that arrived later. For some applications this is a worthwhile tradeoff.

Note that loss of a single DTLS packet can result in the loss of multiple packet digests, which can mean failure to authenticate multiple data packets.

DTLS transport for manifests supports one manifest per packet. It's OPTIONAL to provide for some redundancy in packet digests by providing overlap in the packet sequence numbers across different manifests, thereby sending some or all packet digests multiple times to avoid loss.

Future extensions might define extensions that can provide more efficient redundancy via FEC. Those future extensions will require a different URI scheme.

5. Examples

TBD: walk through some examples as soon as we have a build running. Likely to need some touching up of the spec along the way...

6. YANG Module

6.1. Tree Diagram

The tree diagram below follows the notation defined in [RFC8340].

```
module: ietf-ambi
```

```
augment /dorms:dorms/dorms:metadata/dorms:sender/dorms:group
  /dorms:udp-stream:
```

```
  +--rw ambi
    +--rw manifest-stream* [id]
      +--rw id                uint32
      +--rw manifest-stream* [uri]
        | +--rw uri          inet:uri
        +--rw hash-algorithm  iha:hash-algorithm-type
        +--rw data-hold-time?  uint32
        +--rw digest-hold-time? uint32
        +--rw expiration?     yang:date-and-time
```

```
augment /dorms:dorms/dorms:metadata/dorms:sender/dorms:group:
```

```
  +--rw ambi
    +--rw manifest-stream* [id]
      +--rw id                uint32
      +--rw manifest-stream* [uri]
        | +--rw uri          inet:uri
        +--rw hash-algorithm  iha:hash-algorithm-type
        +--rw data-hold-time?  uint32
        +--rw digest-hold-time? uint32
        +--rw expiration?     yang:date-and-time
```

6.2. Module

```
<CODE BEGINS>
file ietf-ambi@2022-03-07.yang
module ietf-ambi {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-ambi";
  prefix "ambi";

  import ietf-dorms {
    prefix "dorms";
    reference "I-D.jholland-mboned-dorms";
  }

  import ietf-inet-types {
    prefix "inet";
    reference "RFC6991 Section 4";
  }

  import iana-hash-algs {
    prefix "iha";
    reference "draft-ietf-netconf-crypto-types";
  }

  import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
  }

  organization "IETF";

  contact
    "Author:   Jake Holland
              <mailto:jholland@akamai.com>";

  description
    "Copyright (c) 2019 IETF Trust and the persons identified as
    authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject to
    the license terms contained in, the Simplified BSD License set
    forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (https://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
```

for full legal notices.

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 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.

This module contains the definition for the AMBI data types. It provides metadata for authenticating SSM channels as an augmentation to DORMS.";

```
revision 2021-07-08 {
  description "Draft version.";
  reference
    "draft-ietf-mboned-ambi";
}

grouping manifest-stream-definition {
  description
    "This grouping specifies a manifest stream for
    authenticating a multicast data stream with AMBI";
  leaf id {
    type uint32;
    mandatory true;
    description
      "The Manifest ID referenced in a manifest.";
  }
  list manifest-stream {
    key uri;
    leaf uri {
      type inet:uri;
      mandatory true;
      description
        "The URI for a stream of manifests.";
    }
    description "A URI that provides a location for the
      manifest stream";
  }
  leaf hash-algorithm {
    type iha:hash-algorithm-type;
    mandatory true;
    description
      "The hash algorithm for the packet hashes within
      manifests in this stream.";
  }
  leaf data-hold-time {
    type uint32;
```

```
    default 2000;
    units "milliseconds";
    description
        "The number of milliseconds to hold data packets
        waiting for a corresponding digest before
        discarding";
}
leaf digest-hold-time {
    type uint32;
    default 10000;
    units "milliseconds";
    description
        "The number of milliseconds to hold packet
        digests waiting for a corresponding data packet
        before discarding";
}
leaf expiration {
    type yang:date-and-time;
    description
        "The time after which this manifest stream may
        stop providing authentication for the data stream.
        When not present or empty there is no known expiration.";
}
}

augment
    "/dorms:dorms/dorms:metadata/dorms:sender/dorms:group/"+
    "dorms:udp-stream" {
    description "AMBI extensions for securing UDP multicast.";

    container ambi {
        description "UDP-layer AMBI container for DORMS extension.";
        list manifest-stream {
            key id;
            description "Manifest stream definition list.";
            uses manifest-stream-definition;
        }
    }
}

augment
    "/dorms:dorms/dorms:metadata/dorms:sender/dorms:group" {
    description "AMBI extensions for securing IP multicast.";

    container ambi {
        description "IP-layer AMBI container for DORMS extension.";
        list manifest-stream {
            key id;
```

```
        description "Definition of a manifest stream.";
        uses manifest-stream-definition;
    }
}
}
}
<CODE ENDS>
```

7. IANA Considerations

7.1. The YANG Module Names Registry

This document adds one YANG module to the "YANG Module Names" registry maintained at <https://www.iana.org/assignments/yang-parameters>. The following registrations are made, per the format in Section 14 of [RFC6020]:

```
name:      ietf-ambi
namespace: urn:ietf:params:xml:ns:yang:ietf-ambi
prefix:    ambi
reference: I-D.draft-jholland-mboned-ambi
```

7.2. The XML Registry

This document adds the following registration to the "ns" subregistry of the "IETF XML Registry" defined in [RFC3688], referencing this document.

```
URI: urn:ietf:params:xml:ns:yang:ietf-ambi
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

7.3. Media Type

TBD: Register 'application/ambi' according to advice from: <https://www.iana.org/form/media-types> (<https://www.iana.org/form/media-types>)

TBD: check guidelines in <https://tools.ietf.org/html/rfc8126> (<https://tools.ietf.org/html/rfc8126>)

TBD: comments from Amanda: The first is that the current IANA Considerations RFC is RFC 8126 rather than 5226. The other point, which you may be aware of, is that while <https://www.iana.org/form/media-types> provides guidance, standards-tree registrations submitted through RFCs shouldn't be submitted through that form and (unlike vendor-tree subtypes and standards-tree subtypes documented in other standards organization specs) won't need to be explicitly approved by

the IESG-designated experts. Instead, the advice in RFC 6838 is that media type registrations requested by IETF-stream I-Ds be informally reviewed on the media-types@iana.org mailing list, which the IESG-designated experts participate in.

7.4. URI Schemes

7.4.1. TLS

TBD: register 'ambi+tls' as a uri scheme according to advice from:
<https://datatracker.ietf.org/doc/html/rfc7595>
(<https://datatracker.ietf.org/doc/html/rfc7595>)

7.4.2. DTLS

TBD: register 'ambi+dtls' as a uri scheme according to advice from:
<https://datatracker.ietf.org/doc/html/rfc7595>
(<https://datatracker.ietf.org/doc/html/rfc7595>)

8. Security Considerations

8.1. Predictable Packets

Protocols that have predictable packets run the risk of offline attacks for hash collisions against those packets. When authenticating a protocol that might have predictable packets, it's RECOMMENDED to use a hash function secure against such attacks or to add content to the packets to make them unpredictable, such as an Authentication Header ([RFC4302]), or the addition of an ignored field with random content to the packet payload.

TBD: explain attack from generating malicious packets and then looking for collisions, as opposed to having to generate a collision on packet contents that include a sequence number and then hitting a match.

TBD: follow the rest of the guidelines: <https://tools.ietf.org/html/rfc3552>

8.2. Attacks on Side Applications

A multicast receiver subscribes to an (S,G) and if it's a UDP application, listens on a socket with a port number for packets to arrive.

UDP applications sometimes bind to an "unspecified" address ("::" or "0.0.0.0") for a particular UDP port, which will make the application receive and process any packet that arrives on said port.

Forwarding multicast traffic opens a new practical attack surface against receivers that have bound sockets using the "unspecified" address and were operating behind a firewall and/or NAT. Such applications will receive traffic from the internet only after sending an outbound packet, and usually only for return packets with the reversed source and destination port and IP addresses.

Multicast subscription and routing operates at the IP layer, so when a multicst receive application subscribes to a channel, traffic with the IP addresses for that channel will start arriving. There is no selection for the UDP port at the routing layer that prevents multicast IP traffic from arriving.

When an insecure application with a vulnerability is listening to a UDP port on an unspecified address, it will receive multicast packets arriving at the device and with that destination UDP port. Although the primary problem lies in the insecure application, accepting multicast subscriptions increases the attack scope against those applications to include attackers who can inject a packet into a properly subscribed multicast stream.

It's RECOMMENDED that senders using AMBI to secure their traffic include all IP traffic that they send in their DORMS metadata information, and that firewalls using AMBI to provide secure access to multicast traffic block multicast traffic destined to unsecured UDP ports on (S,G)s that have AMBI-based security for any traffic. This mitigation prevents new forwarding of multicast traffic from providing attackers with a packet inject capability access to new attack surfaces from pre-existing insecure apps.

9. Acknowledgements

Many thanks to Daniel Franke, Eric Rescorla, Christian Worm Mortensen, Max Franke, Albert Manfredi, and Amanda Baber for their very helpful comments and suggestions.

10. References

10.1. Normative References

[I-D.draft-ietf-mboned-dorms]
Holland, J., "Discovery Of Restconf Metadata for Source-specific multicast", Work in Progress, Internet-Draft, draft-ietf-mboned-dorms-01, 31 October 2020, <<https://www.ietf.org/archive/id/draft-ietf-mboned-dorms-01.txt>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

10.2. Informative References

- [CVE] MITRE, "Common Vulnerabilities and Exposures", September 1999, <<https://cve.mitre.org/>>.
- [I-D.draft-krose-mboned-alta] Rose, K. and J. Holland, "Asymmetric Loss-Tolerant Authentication", Work in Progress, Internet-Draft, draft-krose-mboned-alta-01, 8 July 2019, <<https://www.ietf.org/archive/id/draft-krose-mboned-alta-01.txt>>.
- [I-D.ietf-tsvwg-udp-options] Touch, J., "Transport Options for UDP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-udp-options-15, 3 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-tsvwg-udp-options-15.txt>>.

- [PathChirp] Ribeiro, V.J., Riedi, R.H., Baraniuk, R.G., Navratil, J., Cottrell, L., Department of Electrical and Computer Engineering Rice University, and SLAC/SCS-Network Monitoring, Stanford University, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", 2003.
- [PathRate] Dovrolis, C., Ramanathan, P., and D. Moore, "Packet dispersion techniques and a capacity estimation methodology", IEEE/ACM Transactions on Networking, Volume 12, Issue 6, pp. 963-977. , December 2004.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J. D., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, DOI 10.17487/RFC4082, June 2005, <<https://www.rfc-editor.org/info/rfc4082>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4383] Baugher, M. and E. Carrara, "The Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Secure Real-time Transport Protocol (SRTP)", RFC 4383, DOI 10.17487/RFC4383, February 2006, <<https://www.rfc-editor.org/info/rfc4383>>.
- [RFC5776] Roca, V., Francillon, A., and S. Faurite, "Use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA) in the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols", RFC 5776, DOI 10.17487/RFC5776, April 2010, <<https://www.rfc-editor.org/info/rfc5776>>.
- [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>>.

- [RFC6584] Roca, V., "Simple Authentication Schemes for the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols", RFC 6584, DOI 10.17487/RFC6584, April 2012, <<https://www.rfc-editor.org/info/rfc6584>>.
- [WEBRC] Luby, M. and V. Goyal, "Wave and Equation Based Rate Control Using Multicast Round Trip Time: Extended Report", Digital Fountain Technical Report no. DF2002-07-001 , September 2002.

Authors' Addresses

Jake Holland
Akamai Technologies, Inc.
150 Broadway
Cambridge, MA 02144,
United States of America
Email: jakeholland.net@gmail.com

Kyle Rose
Akamai Technologies, Inc.
150 Broadway
Cambridge, MA 02144,
United States of America
Email: krose@krose.org

Mboned
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

J. Holland
Akamai Technologies, Inc.
7 March 2022

Circuit Breaker Assisted Congestion Control
draft-ietf-mboned-cbacc-04

Abstract

This document specifies Circuit Breaker Assisted Congestion Control (CBACC). CBACC enables fast-trip Circuit Breakers by publishing rate metadata about multicast channels from senders to intermediate network nodes or receivers. The circuit breaker behavior is defined as a supplement to receiver driven congestion control systems, to preserve network health if misbehaving or malicious receiver applications subscribe to a volume of traffic that exceeds capacity policies or capability for a network or receiving device.

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 8 September 2022.

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
1.1. Background and Terminology	4
1.2. Venues for Contribution and Discussion	4
1.3. Non-obvious doc choices	4
2. Circuit Breaker Behavior	5
2.1. Functional Components	5
2.1.1. Bitrate Advertisement	5
2.1.2. Circuit Breaker Node	6
2.1.3. Communication Method	7
2.1.4. Measurement Function	7
2.1.5. Trigger Function	8
2.1.6. Reaction	9
2.1.7. Feedback Control Mechanism	10
2.2. States	10
2.2.1. Interface State	10
2.2.2. Flow State	11
2.3. Implementation Design Considerations	11
2.3.1. Oversubscription Thresholds	12
2.3.2. Fairness Functions	12
3. YANG Module	12
3.1. Tree Diagram	12
3.2. Module	12
4. IANA Considerations	14
4.1. YANG Module Names Registry	14
4.2. The XML Registry	15
5. Security Considerations	15
5.1. Metadata Security	15
5.2. Denial of Service	15
5.2.1. State Overload	15
6. Acknowledgements	16
7. References	16
7.1. Normative References	16
7.2. Informative References	17
Appendix A. Overjoining	19
Author's Address	20

1. Introduction

This document defines Circuit Breaker Assisted Congestion Control (CBACC). CBACC defines a Network Transport Circuit Breaker (CB), as described by [RFC8084].

The CB behavior defined in this document uses bit-rate metadata about multicast data streams coupled with policy, capacity, and load information at a network location to prune multicast channels so that the network's aggregate capacity at that location is not exceeded by the subscribed channels.

To communicate the required metadata, this document defines a YANG [RFC7950] module that augments the DORMS [I-D.draft-ietf-mboned-dorms] YANG module. DORMS provides a mechanism for senders to publish metadata about the multicast streams they're sending through a RESTCONF service, so that receivers or forwarding nodes can discover and consume the metadata with a set of standard methods. The CBACC metadata MAY be communicated to receivers or forwarding nodes by some other method, but the definition of any alternative methods is out of scope for this document.

The CB behavior defined in this document matches the description provided in Section 3.2.3 of [RFC8084] of a unidirectional CB over a controlled path. The control messages from that description are composed of the messages containing the metadata required for operation of the CB.

CBACC is designed to supplement protocols that use multicast IP and rely on well-behaved receivers to achieve congestion control. Examples of congestion control systems fitting this description include [PLM], [RLM], [RLC], [FLID-DL], [SMCC], and WEBRC [RFC3738].

CBACC addresses a problem with "overjoining" by untrusted receivers.

In an overjoining condition, receivers (either malicious, misconfigured, or with implementation errors) subscribe to multicast channels but do not respond appropriately to congestion. When sufficient multicast traffic is available for subscription by such receivers, this can overload any network.

The overjoining problem is relevant to misbehaving receivers for both receiver-driven and feedback-driven congestion control strategies, as described in Section 4.1 of [RFC8085].

Overjoining attacks and the challenges they present are discussed in more detail in Appendix A.

CBACC offers a solution for the recommendation in Section 4 of [RFC8085] that circuit breaker solutions be used even where congestion control is optional.

1.1. Background and 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.

1.2. Venues for Contribution and Discussion

This document is in the Github repository at:

<https://github.com/GrumpyOldTroll/ietf-dorms-cluster>

Readers are welcome to open issues and send pull requests for this document.

Please note that contributions may be merged and substantially edited, and as a reminder, please carefully consider the Note Well before contributing: <https://datatracker.ietf.org/submit/note-well/>

Substantial discussion of this document should take place on the MBONED working group mailing list (mboned@ietf.org).

* Join: <https://www.ietf.org/mailman/listinfo/mboned>

* Search: <https://mailarchive.ietf.org/arch/browse/mboned/>

1.3. Non-obvious doc choices

- * Since nothing is necessarily being actively measured by a network component at the ingress, referring to the bitrate advertisement as an "ingress meter" for this context was considered confusing by reviewers, so the section was renamed with just a note pointing to the link. Likewise the egress meter and "CB node".
- * TBD: might need more and better examples explaining the point in Section 2.1.5.1? Some reason to believe it's not sufficiently clear...
- * Another TBD: consider Dino's suggestion from 2020-04-09 to include an operational considerations section that addresses some possible optimizations for CB placement and configuration.

- * TBD: add a section walking through the requirements in <https://datatracker.ietf.org/doc/html/rfc8084#section-4> (<https://datatracker.ietf.org/doc/html/rfc8084#section-4>) and explaining how this matches.
- * I'm unclear on whether <https://datatracker.ietf.org/doc/html/rfc8407#section-3.8.2> (<https://datatracker.ietf.org/doc/html/rfc8407#section-3.8.2>) applies here, such that providing an augmentation inside the DORMS namespace causes an update to the DORMS document.

2. Circuit Breaker Behavior

2.1. Functional Components

This section maps the functional components described in Section 3.1 of [RFC8084] to the operational components of the CBACC CB defined by this document.

2.1.1. Bitrate Advertisement

The metadata provides an advertised maximum data bit-rate, namely the "max-speed" field in the YANG model in Section 3. This is a self-report by the sender about the maximum amount of traffic a sender will send within any time interval given by the "data-rate-window" field, which is the measurement interval for the CB. This value refers to the total IP Payload data for all packets in the same (S,G), and its units are in kilobits per second.

The sender MUST NOT send more data for a data stream than the amount of data declared according to its advertised data rate within any measurement window, and it's RECOMMENDED for the sender to provide some margin to account for the possibility of burst forwarding after traffic encounters a non-empty queue, e.g. as sometimes observed with ACK compression (see [ZSC91] for a description of the phenomenon). If a CB node observes a higher data rate transmitted within any measurement window, it MAY circuit-break that flow immediately.

In the terminology of [RFC8084], the bitrate advertisement qualifies as an ingress meter.

2.1.2. Circuit Breaker Node

A circuit breaker node (CB node) is a location in a network where the constraints of the network and the observations about active traffic are compared to the bitrate advertisement in order to make the decision loop about when and whether to perform the circuit breaking behavior. In the terminology of [RFC8084], the CB node qualifies as an egress meter.

The CB node has access to several pieces of information that can be used as relevant egress metrics that may include:

1. Physical capacity limits on each interface.
2. Configured capacity limits for multicast traffic for each interface.
3. The observed received data rates of subscribed multicast channels with CBACC metadata.
4. The observed received data rates of subscribed multicast channels without CBACC metadata.
5. The observed received data rates of competing non-multicast traffic.
6. The loss rate for subscribed multicast channels, when available. The loss rate is only sometimes observable at a CB node; for example, when using AMBI [I-D.draft-ietf-mboned-ambi], or when the data stream carries a protocol that is known to the CB node by some out of band means, and whose traffic can be monitored for loss. When available, the loss rates may be used.

Note that any on-path router can behave as a CB node, even though there may be other CB nodes downstream or upstream covering the same data streams. When viewing CB nodes as egress meters in the context of [RFC8084], it's important to recall there's not a single egress meter in the network, but rather an egress meter per CB node, representing potentially multiple overlaid circuit breakers that may redundantly cover parts of the same path, with potentially different constraints based on the network location where the egress meter operates. All of the CB nodes anywhere on a path constitute separate circuit breakers that may trip independently of other circuit breakers.

Also note that other kinds of components besides on-path routers forwarding the traffic can act as CB nodes, for example the operating system or browser on a device receiving the traffic, or the receiving application itself.

2.1.3. Communication Method

CBACC generally operates at a CB node, where metrics such as those described in Section 2.1.2 are available through system calls, or by communication with various locally deployable system monitoring applications. However, the CBACC processing can equivalently occur on a separate device that can monitor statistics gathered at a CB node, as long as the necessary control functions to trigger the CB can be invoked.

The communication path defined in this document for the CB node to obtain the bitrate advertisement in Section 2.1.1 is the use of DORMS [I-D.draft-ietf-mboned-dorms]. Other methods MAY be used as well or instead, but are out of scope for this document.

2.1.4. Measurement Function

The measurement function maintains a few values for each interface, computed from the metrics described in Section 2.1.2 and Section 2.1.1:

1. The aggregate advertised maximum bit-rate capacity consumed by CBACC data streams. This is the sum of the max-speed values in the CBACC metadata for all data streams subscribed through an interface
2. An oversubscription threshold for each interface. The oversubscription threshold will be determined differently for CB nodes in different contexts. In some network devices, it might be as simple as an administratively configured absolute value or proportion of an interface's capacity. For other situations, like a CB node operating in a context with loss visibility, it could be a dynamically changing value that grows when data streams are successfully subscribed and receiving data without loss, and shrinks as loss is observed across subscribed data streams. The oversubscription threshold calculation could also incorporate other information like out-of-band path capacity measurements with bandwidth detection techniques such as [PathChirp] or [CapProbe].

This document covers some non-normative examples of valid oversubscription threshold functions in Section 2.3.1. In general, the oversubscription threshold is the primary parameter that different CBs in different contexts can tune to provide the safety guarantees necessary for their context.

2.1.5. Trigger Function

The trigger function fires when the aggregate advertised maximum bit-rate exceeds the oversubscription threshold for any interface.

When oversubscribed, the trigger function changes the states of subscribed channels to "blocked" until the aggregate subscribed bit-rate is below the oversubscription threshold again.

2.1.5.1. Fairness and Inter-flow Ordering

The trigger function orders the monitored flows according to a fairness function and a within-sender priority ordering (chosen by the sender as part of the CBACC metadata). When flows are blocked, they're blocked in order until the aggregate bitrate of the permitted flows do not exceed the oversubscription thresholds monitored by the CB node.

Flows from a single sender **MUST** be ordered according to their priority field from the CBACC metadata when compared with each other. This takes precedence over the fairness function ordering, since certain flows from the same sender may need strict priority over others.

For example, consider a sender using File Delivery over Unidirectional Transport (FLUTE, defined in [RFC6726]) that sends File Delivery Table (FDT) Instances (see section 3.2 of [RFC6726]) in one (S,G) and data for the various referenced files in other (S,G)s. In this case the data for the files will not be consumable without the (S,G) containing the FDT. Other transport protocols may similarly send control information (often with a lower bitrate) on one channel, and data information on another. In these cases, the sender may need to ensure that data channels are only available when the control channels are also available.

When comparing flows between senders, (S,G)s from the same sender with different priorities should be treated as aggregated (S,G)s with regard to their declared bitrate consumption, to ensure that if any flows from the same sender need to be pruned by the circuit-breaker, the least preferred priority flows from that sender are pruned first.

Between-sender flows and flows from the same sender with the same priority are ordered according to the fairness function. TBD: need to work thru details, this does not work as written. Sample fairness function would reward senders for splitting a flow in 2 (more total subscribers). Maybe should count offload instead? This has trouble from favoring padding in your flow, but is (i think?) dominated by subscriber count where that's known. The fairness function can be different for CBs in different contexts.

A CBACC CB implementation SHOULD provide mechanisms for administrative controls to configure explicit biases, as this may be necessary to support Service Level Agreements for specific events or providers, or to block or de-prioritize channels with historically known misbehavior.

Subject to the above constraints, where possible the default fairness behavior SHOULD favor streams with many receivers over streams with few receivers, and streams with a low bit-rate over streams with a high bit-rate. See Section 2.3.2 for further considerations and examples.

2.1.6. Reaction

When the trigger function fires and a subscribed channel becomes blocked, the reaction depends on whether it's an upstream interface or a downstream interface.

If a channel is blocked on one or more downstream interfaces, it may still be unblocked on other downstream interfaces. When this is the case, traffic is simply not forwarded along blocked interfaces, even though clients might still be joined downstream of those interfaces.

When a channel is blocked on all downstream interfaces or when the upstream interface is oversubscribed, the channel is pruned so that data no longer arrives from the network on the upstream interface. The prune would be performed with a PIM prune (Section 3.5 of [RFC7761]), or a "leave" operation to be communicated via IGMP, MLD, or another multicast group signaling mechanism, according to the expected signaling within the network.

Once initially pruned, a flow SHOULD remain pruned for a minimum amount of time. The minimum hold-down duration SHOULD be no less than 2.5 minutes by default, even if available bitrate space clears up, to ensure downstream subscriptions will notice and respond. The hold-down duration SHOULD be extended from the minimum by a randomly chosen number of seconds uniformly distributed over a configurable desynchronization period, to avoid synchronized recovery of different circuit breakers along the path. The default length of the desynchronization period should be at least 30 seconds.

2.5 minutes is chosen to exceed the default maximum lifetime of 2 minutes that can occur if an IGMP responder suddenly stops operation, and ceases responding to IGMP queries with membership reports, and 30 seconds is chosen to allow for some flexibility in lost packets. The values MAY be administratively tuned as needed by network operators to meet performance goals specific to their networks or to the traffic they're forwarding.

When enough capacity is available for a circuit-broken stream to be unblocked and the circuit-breaker hold-down time is expired, flows SHOULD be unblocked according to the priority order until no more flows can be unblocked without exceeding the circuit breaker limits.

2.1.7. Feedback Control Mechanism

The bitrate advertisement metadata from Section 2.1.1 should be refreshed as needed to maintain up to date values. When using DORMS and RESTCONF, the Subscription to YANG Notifications for Datastore Updates [RFC8641] is the preferred method to receive changes if available.

If datastore subscriptions are not supported by the client or server, the HTTP Cache Control headers provide valid refresh time properties from the server, and SHOULD be used if present. If No-Cache is used, the default refresh timing SHOULD be 30 seconds. A uniformly distributed random value between 0 and 10 seconds SHOULD be added to the Cache Control or the default refresh timing to avoid synchronization across multiple clients.

2.2. States

2.2.1. Interface State

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

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

- * **bandwidth limit:** The maximum aggregate CBACC advertised bandwidth allowed, not including circuit-broken flows.

When reducing the bandwidth limit due to congestion, the circuit breaker SHOULD 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 unicast traffic that is responsive to congestion.

2.2.2. Flow State

Data streams with CBACC metadata have a state for the upstream interface through which the stream is joined:

- * **'subscribed'**

Indicates that the circuit breaker is subscribed upstream to the flow and forwarding 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 communicated via IGMP, MLD, or another group membership management mechanism.

Data streams also have a per-interface state for downstream interfaces with subscribers, where the data is being forwarded. It's one of:

- * **'forwarding'**

Indicates that the flow is a non-circuit-broken flow in steady state, forwarding packets downstream.

- * **'blocked'**

Indicates that data packets for this flow are NOT forwarded downstream via this interface.

2.3. Implementation Design Considerations

2.3.1. Oversubscription Thresholds

TBD.

2.3.2. Fairness Functions

As an example fairness function that makes good sense for a general case of unknown traffic:

Consider a network where the receiver count for multicast channels is known, for example via the experimental PIM extension for population count defined in [RFC6807].

A good fairness metric for a flow is max-bandwidth divided by receiver-count, with lower values of the fairness metric favored over higher values.

An overview of some other approaches to appropriate fairness metrics is given in Section 2.3 of [RFC5166].

3. YANG Module

3.1. Tree Diagram

The tree diagram below follows the notation defined in [RFC8340].

```
module: ietf-cbacc
```

```
augment /dorms:dorms/dorms:metadata/dorms:sender/dorms:group:
  +--rw cbacc!
    +--rw max-speed          uint32
    +--rw max-packet-size?   uint16
    +--rw data-rate-window?  uint32
    +--rw priority?          uint16
```

3.2. Module

```
<CODE BEGINS>
file ietf-cbacc@2022-03-07.yang
module ietf-cbacc {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-cbacc";
  prefix "cbacc";

  import ietf-dorms {
    prefix "dorms";
    reference "I-D.jholland-mboned-dorms";
```

```
}

organization "IETF";

contact
  "Author:   Jake Holland
            <mailto:jholland@akamai.com>";

description
  "Copyright (c) 2019 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of
  draft-jholland-mboned-cbacc. See the internet draft for full
  legal notices.

  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 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.

  This module contains the definition for bandwidth consumption
  metadata for SSM channels, as an extension to DORMS
  (draft-ietf-mboned-dorms).";

revision 2021-07-08 {
  description "Draft version, post-early-review.";
  reference
    "draft-ietf-mboned-cbacc";
}

augment
  "/dorms:dorms/dorms:metadata/dorms:sender/dorms:group" {
    description "Definition of the manifest stream providing
    integrity info for the data stream";

    container cbacc {
      presence "CBACC-enabled flow";
      description
```

```
    "Information to enable fast-trip circuit breakers";
  leaf max-speed {
    type uint32;
    units "kilobits/second";
    mandatory true;
    description "Maximum bitrate for this stream, in Kilobits
      of IP packet data (including headers) of native
      multicast traffic per second";
  }
  leaf max-packet-size {
    type uint16;
    default 1400;
    description "Maximum IP payload size, in octets.";
  }
  leaf data-rate-window {
    type uint32;
    units "milliseconds";
    default 2000;
    description
      "Time window over which data rate is guaranteed,
      in milliseconds.";
    /* TBD: range limits? */
  }
  leaf priority {
    type uint16;
    default 256;
    description
      "The relative preference level for keeping this flow
      compared to other flows from this sender (higher
      value is more preferred to keep)";
  }
}
}
}
}
<CODE ENDS>
```

4. IANA Considerations

4.1. YANG Module Names Registry

This document adds one YANG module to the "YANG Module Names" registry maintained at <https://www.iana.org/assignments/yang-parameters>. The following registrations are made, per the format in Section 14 of [RFC6020]:

```
name:      ietf-cbacc
namespace: urn:ietf:params:xml:ns:yang:ietf-cbacc
prefix:    cbacc
reference: I-D.draft-ietf-mboned-cbacc
```

4.2. The XML Registry

This document adds the following registration to the "ns" subregistry of the "IETF XML Registry" defined in [RFC3688], referencing this document.

```
URI: urn:ietf:params:xml:ns:yang:ietf-cbacc
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

5. Security Considerations

TBD: Yang Doctor review from Reshad said this should "mention the YANG data nodes". I think this means "do what <https://tools.ietf.org/html/rfc8407#section-3.7> says"?

5.1. Metadata Security

Be sure to authenticate the metadata. See DORMS security considerations, and don't accept unauthenticated metadata if using an alternative means.

5.2. Denial of Service

5.2.1. State Overload

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.

6. Acknowledgements

Many thanks to Devin Anderson, Ben Kaduk, Cheng Jin, Scott Brown, Miroslav Ponec, Bob Briscoe, Lenny Giuliani, Christian Worm Mortensen, Dino Farinacci, and Reshad Rahman for their thoughtful comments and contributions.

7. References

7.1. Normative References

- [I-D.draft-ietf-mboned-ambi]
Holland, J. and K. Rose, "Asymmetric Manifest Based Integrity", Work in Progress, Internet-Draft, draft-ietf-mboned-ambi-01, 31 October 2020, <<https://www.ietf.org/archive/id/draft-ietf-mboned-ambi-01.txt>>.
- [I-D.draft-ietf-mboned-dorms]
Holland, J., "Discovery Of Restconf Metadata for Source-specific multicast", Work in Progress, Internet-Draft, draft-ietf-mboned-dorms-01, 31 October 2020, <<https://www.ietf.org/archive/id/draft-ietf-mboned-dorms-01.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <<https://www.rfc-editor.org/info/rfc8084>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

7.2. Informative References

- [CapProbe] Kapoor, R., Chen, L., Lao, L., Gerla, M., and M.Y. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique", September 2004, <<https://dl.acm.org/doi/pdf/10.1145/1015467.1015476>>.
- [FLID-DL] Byers, J.W., Horn, G., Luby, M., Mitzenmacher, M., Shaver, W., and IEEE, "FLID-DL: congestion control for layered multicast", DOI 10.1109/JSAC.2002.803998, n.d., <<https://ieeexplore.ieee.org/document/1038584>>.
- [PathChirp] Ribeiro, V.J., Riedi, R.H., Baraniuk, R.G., Navratil, J., Cottrell, L., Department of Electrical and Computer Engineering Rice University, and SLAC/SCS-Network Monitoring, Stanford University, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths", 2003.
- [PLM] Biersack, Institut EURECOM, A.Legout, E.W., "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes", 1999, <<http://www.eurecom.fr/en/publication/340/download/ce-legoar-000601.pdf>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3738] Luby, M. and V. Goyal, "Wave and Equation Based Rate Control (WEBRC) Building Block", RFC 3738, DOI 10.17487/RFC3738, April 2004, <<https://www.rfc-editor.org/info/rfc3738>>.
- [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, <<https://www.rfc-editor.org/info/rfc4609>>.
- [RFC5166] Floyd, S., Ed., "Metrics for the Evaluation of Congestion Control Mechanisms", RFC 5166, DOI 10.17487/RFC5166, March 2008, <<https://www.rfc-editor.org/info/rfc5166>>.

- [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>>.
- [RFC6726] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", RFC 6726, DOI 10.17487/RFC6726, November 2012, <<https://www.rfc-editor.org/info/rfc6726>>.
- [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, <<https://www.rfc-editor.org/info/rfc6807>>.
- [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, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RLC] Rizzo, L., Vicisano, L., and J. Crowcroft, "The RLC multicast congestion control algorithm", 1999, <<http://www.iet.unipi.it/~a007834/rlc99.ps.gz>>.
- [RLM] McCanne, S., Jacobson, V., Vetterli, M., University of California, Berkeley, and Lawrence Berkeley National Laboratory, "Receiver-driven Layered Multicast", 1995, <<http://www1.cs.columbia.edu/~danr/courses/6761/Fall00/week9/layering.pdf>>.
- [SMCC] Kwon, G., Byers, J.W., and Computer Science Department, Boston University, "Smooth Multirate Multicast Congestion Control", 2002, <<http://www.cs.bu.edu/techreports/pdf/2002-025-smcc.pdf>>.
- [ZSC91] Zhang, L., Shenker, S., and D.D. Clark, "Observations and Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic", Proc. ACM SIGCOMM, ACM Computer Communications Review (CCR), Vol 21, No 4, pp.133-147. , 1991.

Appendix A. Overjoining

[RFC8085] 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.

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 flows associated with the misbehaving senders. By contrast, 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.

Author's Address

Jake Holland
Akamai Technologies, Inc.
150 Broadway
Cambridge, MA 02144,
United States of America
Email: jakeholland.net@gmail.com

Mboned
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

J. Holland
Akamai Technologies, Inc.
7 March 2022

Discovery Of Restconf Metadata for Source-specific multicast
draft-ietf-mboned-dorms-04

Abstract

This document defines DORMS (Discovery Of Restconf Metadata for Source-specific multicast), a method to discover and retrieve extensible metadata about source-specific multicast channels using RESTCONF. The reverse IP DNS zone for a multicast sender's IP address is configured to use SRV resource records to advertise the hostname of a RESTCONF server that publishes metadata according to a new YANG module with support for extensions. A new service name and the new YANG module are defined.

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 8 September 2022.

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
1.1. Background	3
1.2. Terminology	4
1.3. Motivation and Use Cases	5
1.3.1. Provisioning and Oversubscription Protection	5
1.3.2. Authentication	5
1.3.3. Content Description	5
1.4. Channel Discovery	5
1.5. Notes for Contributors and Reviewers	6
1.5.1. Venues for Contribution and Discussion	6
1.5.2. Non-obvious doc choices	7
2. Discovery and Metadata Retrieval	7
2.1. DNS Bootstrap	7
2.2. Ignore List	9
2.3. RESTCONF Bootstrap	9
2.3.1. Root Resource Discovery	9
2.3.2. Yang Library Version	10
2.3.3. Yang Library Contents	11
2.3.4. Metadata Retrieval	12
2.3.5. Cross Origin Resource Sharing (CORS)	13
3. Scalability Considerations	13
3.1. Provisioning	13
3.2. Data Scoping	13
4. YANG Model	14
4.1. Yang Tree	14
4.2. Yang Module	14
5. Privacy Considerations	16
5.1. Linking Content to Traffic Streams	17
5.2. Linking Multicast Subscribers to Unicast Connections	17
6. IANA Considerations	17
6.1. The YANG Module Names Registry	17
6.2. The XML Registry	18
6.3. The Service Name and Transport Protocol Port Number Registry	18
7. Security Considerations	18
7.1. YANG Model Considerations	18
7.2. Exposure of Metadata	20

7.3. Secure Communications	20
7.4. Record-Spoofing	21
7.5. CORS considerations	21
8. Acknowledgements	22
9. References	22
9.1. Normative References	22
9.2. Informative References	24
Author's Address	26

1. Introduction

This document defines DORMS (Discovery Of Restconf Metadata for Source-specific multicast).

A DORMS service is a RESTCONF [RFC8040] service that provides read access to data in the "ietf-dorms" YANG [RFC7950] model defined in Section 4. This model, along with optional extensions defined in other documents, provide an extensible set of information about multicast data streams. A review of some example use cases that can be enabled by this kind of metadata is given in Section 1.3.

This document does not prohibit the use of the "ietf-dorms" model with other protocols such as NETCONF [RFC6241], CORECONF [I-D.draft-ietf-core-comi], or gNMI [I-D.draft-openconfig-rtgw-gnmi-spec], but the semantics of using the model over those protocols is out of scope for this document. This document only defines the discovery and use of the "ietf-dorms" YANG model in RESTCONF.

This document defines the "dorms" service name for use with the SRV DNS Resource Record (RR) type [RFC2782]. A sender using a DORMS service to publish metadata SHOULD configure at least one SRV RR for the "_dorms._tcp" subdomain in the reverse IP DNS zone for the source IP used by some active multicast traffic. The domain name in one of these SRV records provides a hostname corresponding to a DORMS server that can provide metadata for the sender's source-specific multicast traffic. Publishing such a RR enables DORMS clients to discover and query a DORMS server as described in Section 2.

1.1. Background

The reader is assumed to be familiar with the basic DNS concepts described in [RFC1034], [RFC1035], and the subsequent documents that update them, as well as the use of the SRV Resource Record type as described in [RFC2782].

The reader is also assumed to be familiar with the concepts and terminology regarding source-specific multicast as described in [RFC4607] and the use of IGMPv3 [RFC3376] and MLDv2 [RFC3810] for group management of source-specific multicast channels, as described in [RFC4604].

The reader is also assumed to be familiar with the concepts and terminology for RESTCONF [RFC8040] and YANG [RFC7950].

1.2. Terminology

Term	Definition
(S,G)	A source-specific multicast channel, as described in [RFC4607]. A pair of IP addresses with a source host IP and destination group IP.
DORMS client	An application or system that can communicate with DORMS servers to fetch metadata about (S,G)s.
DORMS server	A RESTCONF server that implements the ietf-dorms YANG model defined in this document.
RR	A DNS Resource Record, as described in [RFC1034]
RRType	A DNS Resource Record Type, as described in [RFC1034]
SSM	Source-specific multicast, as described in [RFC4607]

Table 1

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 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Motivation and Use Cases

DORMS provides a framework that can be extended to publish supplemental information about multicast traffic in a globally discoverable manner. This supplemental information is sometimes needed by entities engaged in delivery or processing of the traffic to handle the traffic according to their requirements.

Detailing the specifics of all known possible extensions is out of scope for this document except to note that a range of possible use cases are expected and they may be supported by a variety of different future extensions. But a few example use cases are provided below for illustration.

1.3.1. Provisioning and Oversubscription Protection

One use case for DORMS is when a network that is capable of forwarding multicast traffic may need to take provisioning actions or make admission control decisions based on the expected bitrate of the traffic in order to prevent oversubscription of constrained devices in the network. [I-D.draft-ietf-mboned-chacc] defines some DORMS extensions to support this use case.

1.3.2. Authentication

Another use case for DORMS is providing information for use in authenticating the multicast traffic before accepting it for forwarding by a network device, or for processing by a receiving application. [I-D.draft-ietf-mboned-ambi] defines some DORMS extensions to support this use case.

1.3.3. Content Description

Another use case for DORMS is describing the contents carried by a multicast traffic channel. The content description could include information about the protocols or applications that can be used to consume the traffic, or information about the media carried (e.g. information based on the Dublin Core Metadata Element Set [RFC5013]), or could make assertions about the legal status of the traffic within specific contexts.

1.4. Channel Discovery

DORMS provides a method for clients to fetch metadata about (S,G)s that are already known to the clients. In general, a DORMS client might learn of an (S,G) by any means, so describing all possible methods a DORMS client might use to discover a set of (S,G)s for which it wants metadata is out of scope for this document.

But for example, a multicast receiver application that is a DORMS client might learn about an (S,G) by getting signals from inside the application logic, such as a selection made by a user, or a scheduled API call that reacts to updates in a library provided by a service operator.

As another example, an on-path router that's a DORMS client might instead learn about an (S,G) by receiving a PIM message or an IGMP or MLD membership report indicating a downstream client has tried to subscribe to an (S,G). Such a router might use information learned from the DORMS metadata to make an access control decision about whether to propagate the join further upstream in the network.

Other approaches for learning relevant (S,G)s could be driven by monitoring a route reflector to discover channels that are being actively forwarded, for a purpose such as monitoring network health.

1.5. Notes for Contributors and Reviewers

Note to RFC Editor: Please remove this section and its subsections before publication.

This section is to provide references to make it easier to review the development and discussion on the draft so far.

1.5.1. Venues for Contribution and Discussion

This document is in the Github repository at:

<https://github.com/GrumpyOldTroll/ietf-dorms-cluster>

Readers are welcome to open issues and send pull requests for this document.

Please note that contributions may be merged and substantially edited, and as a reminder, please carefully consider the Note Well before contributing: <https://datatracker.ietf.org/submit/note-well/>

Substantial discussion of this document should take place on the MBONED working group mailing list (mboned@ietf.org).

* Join: <https://www.ietf.org/mailman/listinfo/mboned>

* Search: <https://mailarchive.ietf.org/arch/browse/mboned/>

1.5.2. Non-obvious doc choices

Log of odd things that need to be the way they are because of some reason that the author or reviewers may want to know later.

- * building the draft without this line produces a warning about no reference to [RFC6991] or [RFC8294], but these are imported in the yang model. RFC 8407 requires the normative reference to 8294 (there's an exception for 6991 but I'm not sure why and it doesn't seem forbidden).
- * Although it's non-normative, I chose the boundaries in the recommendation for default setting of DNS expiry time in Section 2.2 based on the best practices advice at <https://www.varonis.com/blog/dns-ttl/> for "Short" and "Long" times.
- * Section 7.1 is intended to be the template from <https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines> (<https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>), as required by <https://datatracker.ietf.org/doc/html/rfc8407#section-3.7> (<https://datatracker.ietf.org/doc/html/rfc8407#section-3.7>). Individual nodes are not listed because blanket statements in that section cover them.
- * The 'must' constraint in the group list seems awkward, but seems to work. Its intent is to require source & group to be either both IPv4 or both IPv6, without mixing & matching. It requires that either both the group address and its source parent's address must contain a colon or both must NOT contain a colon, where presence of a colon is used to distinguish IPv4 from IPv6. Maybe there's a better way?

2. Discovery and Metadata Retrieval

A client that needs metadata about an (S,G) MAY attempt to discover metadata for the (S,G) using the mechanisms defined here, and MAY use the metadata received to manage the forwarding or processing of the packets in the channel.

2.1. DNS Bootstrap

The DNS Bootstrap step is how a client discovers an appropriate RESTCONF server, given the source address of an (S,G). Use of the DNS Bootstrap is OPTIONAL for clients with an alternate method of obtaining a hostname of a trusted DORMS server that has information about a target (S,G).

This mechanism only works for source-specific multicast (SSM) channels. The source address of the (S,G) is reversed and used as an index into one of the reverse mapping trees (in-addr.arpa for IPv4, as described in Section 3.5 of [RFC1035], or ip6.arpa for IPv6, as described in Section 2.5 of [RFC3596]).

When a DORMS client needs metadata for an (S,G), for example when handling a new join for that (S,G) and looking up the authentication methods that are available, the DORMS client can issue a DNS query for a SRV RR using the "dorms" service name with the domain from the reverse mapping tree, combining them as described in [RFC2782].

For example, a client looking for metadata about the channel with a source IP of 2001:db8::a and the group address of ff3e::8000:d, the client would start the DNS bootstrap step by performing a query for the SRV RRTYPE for the following domain (after removing the line break inserted for editorial reasons):

```
_dorms._tcp.a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.  
0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
```

Or for an IPv4 (S,G) with a source address of 203.0.113.4, the DORMS client would request the SRV record from the in-addr.arpa tree instead:

```
_dorms._tcp.4.113.0.203.in-addr.arpa.
```

In either case, the DNS response for this domain might return a record such as this:

```
SRV 0 1 443 dorms-restconf.example.com.
```

This response informs the client that a DORMS server should be reachable at `dorms-restconf.example.com` on port 443, and should contain metadata about multicast traffic from the given source IP. Multiple SRV records are handled as described by [RFC2782].

A sender providing DORMS discovery SHOULD publish at least one SRV record in the reverse DNS zone for each source address of the multicast channels it is sending in order to advertise the hostname of the DORMS server to DORMS clients. The DORMS servers advertised SHOULD be configured with metadata for all the groups sent from the same source IP address that have metadata published with DORMS.

When performing the SRV lookup, any CNAMEs or DNAMEs found MUST be followed. This is necessary to support zone delegation. Some examples outlining this need are described in [RFC2317].

2.2. Ignore List

If a DORMS client reaches a DORMS server but determines through examination of responses from that DORMS server that it may not understand or be able to use the responses of the server (for example due to an issue like a version mismatch or modules that are missing but are required for the DORMS client's purposes), the client MAY add this server to an ignore list and reject servers in its ignore list during future discovery attempts.

A client using the DNS Bootstrap discovery method in Section 2.1 would treat servers in its ignore list as unreachable for the purposes of processing the SRV RR as described in [RFC2782]. (For example, a client might end up selecting a server with a less-preferred priority than servers in its ignore list, even if an HTTPS connection could have been formed successfully with some of those servers.)

If an ignore list is maintained, entries SHOULD time out and allow for re-checking after either the cache expiration time from the DNS response that caused the server to be added to the ignore list, or for a configurable hold-down time that has a default value no shorter than 1 hour and no longer than 24 hours.

2.3. RESTCONF Bootstrap

Once a DORMS server has been chosen (whether via an SRV RR from a DNS response or via some other method), RESTCONF provides all the information necessary to determine the versions and url paths for metadata from the server. A walkthrough is provided here for a sequence of example requests and responses from a receiver connecting to a new DORMS server.

2.3.1. Root Resource Discovery

As described in Section 3.1 of [RFC8040] and [RFC6415], the RESTCONF server provides the link to the RESTCONF api entry point via the `"/.well-known/host-meta"` or `"/.well-known/host-meta.json"` resource.

Example:

The receiver might send:

```
GET /.well-known/host-meta.json HTTP/1.1
Host: dorms-restconf.example.com
Accept: application/json
```

The server might respond as follows:

```
HTTP/1.1 200 OK
Date: Tue, 09 Jul 2021 20:56:00 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/json
```

```
{
  "links":[
    {
      "rel":"restconf",
      "href":"/top/restconf"
    }
  ]
}
```

2.3.2. Yang Library Version

As described in Section 3.3.3 of [RFC8040], the yang-library-version leaf is required by RESTCONF, and can be used to determine the schema of the ietf-yang-library module:

Example:

The receiver might send:

```
GET /top/restconf/yang-library-version HTTP/1.1
Host: dorms-restconf.example.com
Accept: application/yang-data+json
```

The server might respond as follows:

```
HTTP/1.1 200 OK
Date: Tue, 09 Jul 2021 20:56:01 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/yang-data+json

{
  "ietf-restconf:yang-library-version": "2016-06-21"
}
```

If a DORMS client determines through examination of the yang-library-version that it may not understand the responses of the server due to a version mismatch, the server qualifies as a candidate for adding to an ignore list as described in Section 2.2.

2.3.3. Yang Library Contents

After checking that the version of the yang-library module will be understood by the receiver, the client can check that the desired metadata modules are available on the DORMS server by fetching the module-state resource from the ietf-yang-library module.

Example:

The receiver might send:

```
GET /top/restconf/data/ietf-yang-library:modules-state/\
    module=ietf-dorms,2021-07-08
Host: dorms-restconf.example.com
Accept: application/yang-data+json
```

The server might respond as follows:

```
HTTP/1.1 200 OK
Date: Tue, 09 Jul 2021 20:56:02 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/yang-data+json

{
  "ietf-yang-library:module": [
    {
      "conformance-type": "implement",
      "name": "ietf-dorms",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-dorms",
      "revision": "2021-07-08",
      "schema":
        "https://example.com/yang/ietf-dorms@2021-07-08.yang"
    }
  ]
}
```

Other modules required or desired by the client also can be checked in a similar way, or the full set of available modules can be retrieved by not providing a key for the "module" list. If a DORMS client that requires the presence of certain modules to perform its function discovers the required modules are not present on a server, that server qualifies for inclusion in an ignore list according to Section 2.2.

2.3.4. Metadata Retrieval

Once the expected DORMS version is confirmed, the client can retrieve the metadata specific to the desired (S,G).

Example:

The receiver might send:

```
GET /top/restconf/data/ietf-dorms:dorms/metadata/\
  sender=2001:db8::a/group=ff3e::8000:1
Host: dorms-restconf.example.com
Accept: application/yang-data+json
```

The server might respond as follows:

```
HTTP/1.1 200 OK
Date: Tue, 09 Jul 2021 20:56:02 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/yang-data+json
```

```
{
  "ietf-dorms:group": [
    {
      "group-address": "ff3e::8000:1",
      "udp-stream": [
        {
          "port": "5001"
        }
      ]
    }
  ]
}
```

Note that when other modules are installed on the DORMS server that extend the ietf-dorms module, other fields MAY appear inside the response. This is the primary mechanism for providing extensible metadata for an (S,G), so clients SHOULD ignore fields they do not understand.

As mentioned in Section 3.2, most clients SHOULD use data resource identifiers in the request URI as in the above example, in order to retrieve metadata for only the targeted (S,G)s.

2.3.5. Cross Origin Resource Sharing (CORS)

It is RECOMMENDED that DORMS servers use the Access-Control-Allow-Origin header field, as specified by [whatwg-fetch], and that they respond appropriately to Preflight requests.

The use of '*' for allowed origins is NOT RECOMMENDED for publicly reachable DORMS servers. A review of some of the potential consequences of unrestricted CORS access is given in Section 7.5.

3. Scalability Considerations

3.1. Provisioning

In contrast to many common RESTCONF deployments that are intended to provide configuration management for a service to a narrow set of authenticated administrators, DORMS servers often provide read-only metadata for public access or for a very large set of end receivers, since it provides metadata in support of multicast data streams and multicast can scale to very large audiences.

Operators are advised to provision the DORMS service in a way that will scale appropriately to the size of the expected audience. Specific advice on such scaling is out of scope for this document, but some of the mechanisms outlined in [RFC3040] or other online resources might be useful, depending on the expected number of receivers.

3.2. Data Scoping

Except as outlined below, clients SHOULD issue narrowed requests for DORMS resources by following the format from Section 3.5.3 of [RFC8040] to encode data resource identifiers in the request URI. This avoids downloading excessive data, since the DORMS server may provide metadata for many (S,G)s, possibly from many different senders.

However, clients with out of band knowledge about the scope of the expected contents MAY issue requests for (S,G) metadata narrowed only by the source-address, or not narrowed at all. Depending on the request patterns and the contents of the data store, this may result in fewer round trips or less overhead, and can therefore be helpful behavior for scaling purposes in some scenarios. In general, engaging in this behavior requires some administrative configuration or some optimization heuristics that can recover from unexpected results.

Servers MAY restrict or throttle client access based on the client certificate presented (if any), or based on heuristics that take note of client request patterns.

A complete description of the heuristics for clients and servers to meet their scalability goals is out of scope for this document.

4. YANG Model

The primary purpose of the YANG model defined here is to serve as a scaffold for the more useful metadata that will extend it. See Section 1.3 for some example use cases that can be enabled by the use of DORMS extensions.

4.1. Yang Tree

The tree diagram below follows the notation defined in [RFC8340].

```

module: ietf-dorms
  +--rw dorms
    +--rw metadata
      +--rw sender* [source-address]
      +--rw source-address  inet:ip-address
      +--rw group* [group-address]
      +--rw group-address
      |      rt-types:ip-multicast-group-address
      +--rw udp-stream* [port]
      +--rw port  inet:port-number

```

Figure 1: DORMS Tree Diagram

4.2. Yang Module

```

<CODE BEGINS>
file ietf-dorms@2022-03-07.yang
module ietf-dorms {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-dorms";
  prefix "dorms";

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991 Section 4";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

```

```
    reference "RFC 8294";
}

organization "IETF MBONED (Multicast Backbone
  Deployment) Working Group";

contact
  "Author:   Jake Holland
            <mailto:jholland@akamai.com>
  ";

description
  "Copyright (c) 2019 IETF Trust and the persons identified as
  authors of the code.  All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
  forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX
  (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
  for full legal notices.

  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 (RFC 2119) (RFC 8174) when, and only when,
  they appear in all capitals, as shown here.

  This module contains the definition for the DORMS data type.
  It provides out of band metadata about SSM channels."

revision 2021-07-08 {
  description "Draft version, post-early-review.";
  reference
    "draft-ietf-mboned-dorms";
}

container dorms {
  description "Top-level DORMS container.";
  container metadata {
    description "Metadata scaffold for source-specific multicast
      channels.";
    list sender {
      key source-address;
```



```
description "Sender for DORMS";

leaf source-address {
    type inet:ip-address;
    mandatory true;
    description
        "The source IP address of a multicast sender.";
}

list group {
    key group-address;
    description "Metadata for a DORMS (S,G).";

    leaf group-address {
        type rt-types:ip-multicast-group-address;
        mandatory true;
        description "The group IP address for an (S,G).";
    }
    must '(re-match(..group-address, "[^:]*") and \' +
        \'re-match(..source-address, "[^:]*")) or \' +
        \'(re-match(..group-address, ".*:.*)" and \' +
        \'re-match(..source-address, ".*:.*)" )\' {
        error-message 'A group-address type must match '+
            'its parent source-address type';
    }
}

list udp-stream {
    key "port";
    description
        "Metadata for UDP traffic on a specific port.";
    leaf port {
        type inet:port-number;
        mandatory true;
        description
            "The UDP port of a data stream.";
    }
}
}
}
}
}
<CODE ENDS>
```

5. Privacy Considerations

5.1. Linking Content to Traffic Streams

In the typical case, the mechanisms defined in this document provide a standardized way to discover information that is already available in other ways.

However, depending on the metadata provided by the server, observers may be able to more easily associate traffic from an (S,G) with the content contained within the (S,G). At the subscriber edge of a multicast-capable network, where the network operator has the capability to localize an IGMP [RFC3376] or MLD [RFC3810] channel subscription to a specific user or location, for example by MAC address or source IP address, the structured publishing of metadata may make it easier to automate collection of data about the content a receiver is consuming.

5.2. Linking Multicast Subscribers to Unicast Connections

Subscription to a multicast channel generally only exposes the IGMP or MLD membership report to others on the same LAN, and as the membership propagates through a multicast-capable network, it ordinarily gets aggregated with other end users.

However, a RESTCONF connection is a unicast connection, and exposes a different set of information to the operator of the RESTCONF server, including IP address and timing about the requests made. Where DORMS access becomes required to succeed a multicast join (for example, as expected in a browser deployment), this can expose new information about end users relative to services based solely on multicast streams. The information disclosure occurs by giving the DORMS service operator information about the client's IP and the channels the client queried.

In some deployments it may be possible to use a proxy that aggregates many end users when the aggregate privacy characteristics are needed by end users.

6. IANA Considerations

6.1. The YANG Module Names Registry

This document adds one YANG module to the "YANG Module Names" registry maintained at <<https://www.iana.org/assignments/yang-parameters>>. The following registrations are made, per the format in Section 14 of [RFC6020]:

```
name:      ietf-dorms
namespace: urn:ietf:params:xml:ns:yang:ietf-dorms
prefix:    dorms
reference:  I-D.draft-ietf-mboned-dorms
```

6.2. The XML Registry

This document adds the following registration to the "ns" subregistry of the "IETF XML Registry" defined in [RFC3688], referencing this document.

```
URI: urn:ietf:params:xml:ns:yang:ietf-dorms
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

6.3. The Service Name and Transport Protocol Port Number Registry

This document adds one service name to the "Service Name and Transport Protocol Port Number Registry" maintained at <https://www.iana.org/assignments/service-names-port-numbers>. The following registrations are made, per the format in Section 8.1.1 of [RFC6335]:

```
Service Name:      dorms
Transport Protocol(s): TCP, UDP
Assignee:          IESG <iesg@ietf.org>
Contact:           IETF Chair <chair@ietf.org>
Description:       The DORMS service (RESTCONF that
                   includes ietf-dorms YANG model)
Reference:         I-D.draft-ietf-mboned-dorms
Port Number:       N/A
Service Code:      N/A
Known Unauthorized Uses: N/A
Assignment Notes:  N/A
```

7. Security Considerations

7.1. YANG Model Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via RESTCONF [RFC8040]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config)

to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

Subtrees:

- * /dorms/metadata
- * /dorms/metadata/sender
- * /dorms/metadata/sender/group
- * /dorms/metadata/sender/group/udp-stream

Data nodes:

- * /dorms/metadata/sender/source-address
- * /dorms/metadata/sender/group/group-address
- * /dorms/metadata/sender/group/udp-stream/port

These data nodes refer to the characteristics of a stream of data packets being sent on a multicast channel. If an unauthorized or incorrect edit is made, receivers would no longer be able to associate the data stream to the correct metadata, resulting in a denial of service for end users that rely on the metadata to properly process the data packets. Therefore DORMS servers MUST constrain write access to ensure that unauthorized users cannot edit the data published by the server.

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content. DORMS servers MAY use NACM to constrain write accesses.

However, note that scalability considerations described in Section 3.1 might make the naive use of NACM intractable in many deployments, for a broadcast use case. So alternative methods to constrain write access to the metadata MAY be used instead of or in addition to NACM. For example, some deployments that use a CDN or caching layer of discoverable DORMS servers might uniformly provide read-only access through the caching layer, and might require the trusted writers of configuration to use an alternate method of accessing the underlying database such as connecting directly to the origin, or requiring the use of a non-RESTCONF mechanism for editing the contents of the metadata.

The data nodes defined in this YANG module are writable because some deployments might manage the contents in the database by using normal RESTCONF editing operations with NACM, but in typical deployments it's expected that DORMS clients will generally have read-only access. For the reasons and requirements described in Section 7.2, none of the data nodes in the DORMS module or its extensions contain sensitive data.

DORMS servers MAY provide read-only access to clients for publicly available metadata without authenticating the clients. That is, under the terms in Section 2.5 of [RFC8040] read-only access to publicly available data MAY be treated as unprotected resources.

7.2. Exposure of Metadata

Although some DORMS servers MAY restrict access based on client identity, as described in Section 2.5 of [RFC8040], many DORMS servers will use the ietf-dorms YANG model to publish information without restriction, and even DORMS servers requiring client authentication will inherently, because of the purpose of DORMS, be providing the DORMS metadata to potentially many receivers.

Accordingly, future YANG modules that augment data paths under "ietf-dorms:dorms" MUST NOT include any sensitive data unsuitable for public dissemination in those data paths.

Because of the possibility that scalable read-only access might be necessary to fulfill the scalability goals for a DORMS server, data under these paths MAY be cached or replicated by numerous external entities, so owners of such data SHOULD NOT assume such data can be kept secret when provided by DORMS servers anywhere under the "ietf-dorms:dorms" path even if access controls are used with authenticated clients unless additional operational procedures and restrictions are defined and implemented that can effectively control the dissemination of the secret data. DORMS alone does not provide any such mechanisms, and users of DORMS can be expected not to be following any such mechanisms in the absence of additional assurances.

7.3. Secure Communications

The provisions of Section 2 of [RFC8040] provide secure communication requirements that are already required of DORMS servers, since they are RESTCONF servers. All RESTCONF requirements and security considerations remain in force for DORMS servers.

It is intended that security related metadata about the SSM channels such as public keys for use with cryptographic algorithms may be delivered over the RESTCONF connection, and that information available from this connection can be used as a trust anchor. The secure transport provided by these minimum requirements are relied upon to provide authenticated delivery of these trust anchors, once a connection with a trusted DORMS server has been established.

7.4. Record-Spoofing

When using the DNS Bootstrap method of discovery described in Section 2.1, the SRV resource record contains information that SHOULD be communicated to the DORMS client without being modified. The method used to ensure the result was unmodified is up to the client.

There must be a trust relationship between the end consumer of this resource record and the DNS server. This relationship may be end-to-end DNSSEC validation or a secure connection to a trusted DNS server that provides end-to-end safety to prevent record-spoofing of the response from the trusted server. The connection to the trusted server can use any secure channel, such as with a TSIG [RFC8945] or SIG(0) [RFC2931] channel, a secure local channel on the host, DNS over TLS [RFC7858], DNS over HTTPS [RFC8484], or some other mechanism that provides authentication of the RR.

If a DORMS client accepts a maliciously crafted SRV record, the client could connect to a server controlled by the attacker, and use metadata provided by them. The consequences of trusting maliciously crafted metadata could range from attacks against the DORMS client's parser of the metadata (via malicious constructions of the formatting of the data) to arbitrary disruption of the decisions the DORMS client makes as a result of processing validly constructed metadata.

Clients MAY use other secure methods to explicitly associate an (S,G) with a set of DORMS server hostnames, such as a configured mapping or an alternative trusted lookup service.

7.5. CORS considerations

As described in Section 2.3.5, it's RECOMMENDED that DORMS servers provide appropriate restrictions to ensure only authorized web pages access metadata for their (S,G)s from the widely deployed base of secure browsers that use the CORS protocol according to [whatwg-fetch].

Providing '*' for the allowed origins exposes the DORMS-based metadata to access by scripts in all web pages, which opens the possibility of certain kinds of attacks against networks where browsers have support for joining multicast (S,G)s.

If the authentication for an (S,G) relies on DORMS-based metadata (for example, as defined in [I-D.draft-ietf-mboned-ambi]), an unauthorized web page that tries to join an (S,G) not permitted by the CORS headers for the DORMS server will be prevented from subscribing to the channels.

If an unauthorized site is not prevented from subscribing, code on the site (for example a malicious advertisement) could request subscriptions from many different (S,G)s, overflowing limits on the joining of (S,G)s and disrupting the delivery of multicast traffic for legitimate use.

Further, if the malicious script can be distributed to many different users within the same receiving network, the script could coordinate an attack against the network as a whole by joining disjoint sets of (S,G)s from different users within the receiving network. The distributed subscription requests across the receiving network could overflow limits for the receiving network as a whole, essentially causing the websites displaying the ad to participate in an overjoining attack (see Appendix A of [I-D.draft-ietf-mboned-cbacc]).

Even if network safety mechanisms protect the network from the worst effects of oversubscription, the population counts for the multicast subscriptions could be disrupted by this kind of attack, and therefore push out legitimately requested traffic that's being consumed by real users. For a legitimately popular event, this could cause a widespread disruption to the service if it's successfully pushed out.

A denial of service attack of this sort would be thwarted by restricting the access to (S,G)s to authorized websites through the use of properly restricted CORS headers.

8. Acknowledgements

Thanks to Christian Worm Mortensen, Dino Farinacci, Lenny Guiliano, and Reshad Rahman for their very helpful comments and reviews.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2317] Eidnes, H., de Groot, G., and P. Vixie, "Classless IN-ADDR.ARPA delegation", BCP 20, RFC 2317, DOI 10.17487/RFC2317, March 1998, <<https://www.rfc-editor.org/info/rfc2317>>.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <<https://www.rfc-editor.org/info/rfc2782>>.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <<https://www.rfc-editor.org/info/rfc3596>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [whatwg-fetch] "WHATWG Fetch Living Standard", October 2020, <<https://fetch.spec.whatwg.org/>>.

9.2. Informative References

- [I-D.draft-ietf-core-comi] Veillette, M., Stok, P. V. D., Pelov, A., Bierman, A., and I. Petrov, "CoAP Management Interface (CORECONF)", Work in Progress, Internet-Draft, draft-ietf-core-comi-11, 17 January 2021, <<https://www.ietf.org/archive/id/draft-ietf-core-comi-11.txt>>.
- [I-D.draft-ietf-mboned-ambi] Holland, J. and K. Rose, "Asymmetric Manifest Based Integrity", Work in Progress, Internet-Draft, draft-ietf-mboned-ambi-01, 31 October 2020, <<https://www.ietf.org/archive/id/draft-ietf-mboned-ambi-01.txt>>.
- [I-D.draft-ietf-mboned-cbacc] Holland, J., "Circuit Breaker Assisted Congestion Control", Work in Progress, Internet-Draft, draft-ietf-mboned-cbacc-02, 1 February 2021, <<https://www.ietf.org/archive/id/draft-ietf-mboned-cbacc-02.txt>>.
- [I-D.draft-openconfig-rtgwg-gnmi-spec] Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", Work in Progress, Internet-Draft, draft-openconfig-rtgwg-gnmi-spec-01, 5 March 2018, <<https://www.ietf.org/archive/id/draft-openconfig-rtgwg-gnmi-spec-01.txt>>.

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, DOI 10.17487/RFC2931, September 2000, <<https://www.rfc-editor.org/info/rfc2931>>.
- [RFC3040] Cooper, I., Melve, I., and G. Tomlinson, "Internet Web Replication and Caching Taxonomy", RFC 3040, DOI 10.17487/RFC3040, January 2001, <<https://www.rfc-editor.org/info/rfc3040>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, DOI 10.17487/RFC4604, August 2006, <<https://www.rfc-editor.org/info/rfc4604>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.
- [RFC5013] Kunze, J. and T. Baker, "The Dublin Core Metadata Element Set", RFC 5013, DOI 10.17487/RFC5013, August 2007, <<https://www.rfc-editor.org/info/rfc5013>>.

- [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>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC6415] Hammer-Lahav, E., Ed. and B. Cook, "Web Host Metadata", RFC 6415, DOI 10.17487/RFC6415, October 2011, <<https://www.rfc-editor.org/info/rfc6415>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [RFC8945] Dupont, F., Morris, S., Vixie, P., Eastlake 3rd, D., Gudmundsson, O., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", STD 93, RFC 8945, DOI 10.17487/RFC8945, November 2020, <<https://www.rfc-editor.org/info/rfc8945>>.

Author's Address

Jake Holland
Akamai Technologies, Inc.
150 Broadway
Cambridge, MA 02144,
United States of America
Email: jakeholland.net@gmail.com

Mboned
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

J. Holland
Akamai Technologies, Inc.
7 March 2022

Multicast Network Address Translation
draft-ietf-mboned-mnat-01

Abstract

This document defines a method for a network to maintain Network Address Translation address mappings for the transport of globally addressed multicast traffic within a network that can't otherwise forward the globally addressed traffic. A new Multicast Network Address Translation (MNAT) service is defined to communicate the address mappings to ingress and egress points within the network, and considerations for operation of the MNAT service are described.

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 8 September 2022.

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
1.1. Background	3
1.2. Terminology	4
1.3. Motivation	4
1.4. Notes for Contributors and Reviewers	5
1.4.1. Venues for Contribution and Discussion	6
1.4.2. Implementation status	6
2. Protocol Operation	6
2.1. Overview	6
2.1.1. Egress Node Operational Modes	7
2.2. Service Discovery	8
2.2.1. Detecting Invalid Services	8
2.3. RESTCONF Bootstrap	8
2.4. Message Handling	9
2.4.1. Notification Subscription	9
2.4.2. Watcher Keys	9
2.4.3. Egress Group Management	10
2.4.4. Ingress Considerations	10
2.4.5. MNAT Service Considerations	11
2.4.6. Example Messaging Walkthrough	12
3. YANG Model	12
3.1. Yang Tree	12
3.2. Yang Module	13
4. IANA Considerations	19
4.1. The YANG Module Names Registry	19
4.2. The XML Registry	20
4.3. The Service Name and Transport Protocol Port Number Registry	20
5. Security Considerations	20
6. Acknowledgements	21
7. References	21
7.1. Normative References	21
7.2. Informative References	22
Author's Address	23

1. Introduction

Network Address Translation is very widely used for unicast traffic in a variety of networks and according to a variety of mechanisms. [RFC2663] is recommended reading for background on the ways unicast NAT is used.

The handling of multicast traffic can pose a variety of additional problems for a network, some of which can be mitigated or avoided if traffic can be mapped to a different address space than its original addressing. This document defines a new service, Multicast Network Address Translation (MNAT) as a mechanism to administer network address mappings for multicast traffic within a network, for the purpose of working around various addressing-related issues. An overview of some of the motivating use cases that can be resolved by network address remapping for multicast traffic is given in Section 1.3. An explanation of the protocol operation is given in Section 2.

Messaging to and from the MNAT service is defined with RESTCONF [RFC8040] using the YANG [RFC7950] model in Section 3.

Unlike traditional unicast NAT, MNAT performs address translation at both an ingress point to the network (where the traffic is transformed to use an address scheme local to the network), and also at an egress point from the network (where the traffic is transformed back to the original address scheme for further forwarding, or for further processing by a receiving application).

1.1. Background

The reader is assumed to be familiar with the concepts and terminology regarding source-specific multicast as described in [RFC4607] and the use of IGMPv3 [RFC3376] and MLDv2 [RFC3810] for group management of source-specific multicast channels, as described in [RFC4604].

The reader is also assumed to be familiar with the concepts and terminology for RESTCONF [RFC8040] and YANG [RFC7950].

The reader is also assumed to be familiar with the use of DNS-SD [RFC6763] for discovery of services provided by the network to end hosts.

1.2. Terminology

Term	Definition
(S,G)	A source-specific multicast channel, as described in [RFC4607]. A pair of IP addresses with a source host IP and destination group IP.
egress node	A MNAT client operating at a point where NATted multicast traffic exits the network (close to the receiver)
ingress node	A MNAT client operating at a point where multicast traffic enters the network and gets NATted (close to the sender)
MNAT client	A client using the ietf-mnat YANG model via RESTCONF, or a client with equivalent signaling to an MNAT service.
NATted traffic	Multicast traffic that has been translated to use addressing or encapsulation assigned locally within the network, rather than its original global addressing.
SSM	Source-specific multicast, as described in [RFC4607]

Table 1

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 [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.3. Motivation

This section lists use cases where a global (S,G) may not be possible to transport within a network, requiring the use of some kind of encapsulation or address translation in order to adequately communicate the group membership for packet replication within the network, or in order to perform the forwarding for the subscribed traffic within the network.

1. Global IPv6 (S,G)s subscribed from within an IPv4-only network, or global IPv4 (S,G)s subscribed from within an IPv6-only network.
2. Networks with legacy devices that support only IGMPv2 or MLDv1, or otherwise do not support SSM and cannot discover the external sources without the use of non-standard services since interdomain any-source multicast has been deprecated (see [RFC8815]).
3. Networks that ingest external multicast traffic in a way that the route to the source of the traffic does not go through the ingest point may need to use a different source so that the Reverse Path Forwarding (RPF) can find the correct network location for the ingest.
4. Networks that provision multicast transport and packet replication channels with static routing instead of dynamic tree-building protocols like PIM-SM [RFC7761].
5. Networks using VLAN [IEEE-802.1Q] for traffic segregation and has Layer 2 access devices that assign VLAN tags according to MAC addresses will get MAC address collisions among multicast groups. Because the bits used for the multicast addresses come from the bottom 23 bits of the destination group address as described in [RFC1112] and those bits can collide between groups, especially in SSM. The technological limitations of VLAN assignment using MAC addresses at Layer 2 breaks the traffic segregation of multicast traffic for different services in such devices.

A note elaborating on the use of static routing for multicast groups:

Some networks have found that there are good use cases to deliver a limited set of packet-replicating flows, including sometimes the use of externally sourced multicast traffic, but have struggled with the operational complexity of operating a dynamic tree-building system based on PIM-SM [RFC7761]. Operating an MNAT service can allow these networks to provide for the limited use of packet-replicating data channels while keeping the operational complexity of handling a dynamically changing set of channels confined to a single service that implements their business logic for admission control, rather than trying to apply access control lists for group membership propagation spread across the network.

1.4. Notes for Contributors and Reviewers

Note to RFC Editor: Please remove this section and its subsections before publication.

This section is to provide references to make it easier to review the development and discussion on the draft so far.

1.4.1. Venues for Contribution and Discussion

This document is in the Github repository at:

<https://github.com/GrumpyOldTroll/draft-ietf-mnat>

Readers with feedback are invited to open issues and send pull requests for this document.

Please note that contributions may be merged and substantially edited, and as a reminder, please carefully consider the Note Well before contributing: <https://datatracker.ietf.org/submit/note-well/>

Substantial discussion of this document should take place on the MBONED working group mailing list (mboned@ietf.org).

* Join: <https://www.ietf.org/mailman/listinfo/mboned>

* Search: <https://mailarchive.ietf.org/arch/browse/mboned/>

1.4.2. Implementation status

There is an implementation prototype (MIT-licensed) at:

* <https://github.com/GrumpyOldTroll/mnat>

Pull requests, comments, testing and deployment reports, etc. are very welcome. Contributors before the final stages of RFC publication will be credited in this document unless requested otherwise.

2. Protocol Operation

2.1. Overview

The use of MNAT within a network is defined in terms the following entities:

- * MNAT service
- * ingress nodes
- * egress nodes

Address translation is performed at the ingress (closest to the sender) and egress (closest to the receiver) nodes. Ingress is where an external (S,G) is mapped to locally assigned address mapping before being forwarded for transport within the network. Egress is where the traffic received on locally assigned addresses is translated back to the corresponding external (S,G) address before being forwarded for further transmission or processed by a receiving application.

The MNAT service maintains the mapping between external (S,G)s and the local network addresses used to transport traffic of those (S,G)s within the network. The address mapping is performed according to the needs of the network operating the MNAT service, to satisfy whatever constraints and restrictions may be necessary or desirable according to the operational considerations within that network. Some example considerations that have motivated the design of MNAT are described in Section 1.3.

Ingress and egress nodes communicate with the MNAT service according to the schema defined by the YANG model in Section 3. Based on the messages exchanged with the MNAT service, each ingress or egress node maintains an up-to-date table of the mappings between the external (S,G)s and the locally assigned addresses for transport within the network. The table of mappings is used to perform the corresponding network address translations.

TBD: probably add a diagram here. Probably something roughly similar to page 7 of the IETF 108 mboned presentation touching on this: <https://www.ietf.org/proceedings/108/slides/slides-108-mboned-status-update-on-multicast-to-the-browser-00.pdf#page=7>

2.1.1. Egress Node Operational Modes

Egress nodes can run in at least two separate modes of operation.

One of the modes is "bump in the wire", which refers to a node that receives traffic using the network-assigned locally chosen addresses, and translates the traffic back to the associated externally addressed (S,G) before forwarding the traffic along the rest of the network paths to the receiving applications that tried to join the external (S,G).

The second mode is "bump in the host", which refers to a virtual node operating inside a client application.

As a "bump in the host" egress node, the virtual egress node can discover and connect to the MNAT service from a receiving application. The receiving application would then use the knowledge

about the address mapping within the network to perform a join for the mapped addresses in the local network, rather than for the external (S,G). The payloads of the traffic received with the locally mapped addresses are treated by the application as though they arrived with the external (S,G) addressing.

A common scenario for a bump in the wire egress node deployment might be to have egress nodes operating in Customer Premises Equipment (CPE), such as a Cable Modem or Wi-Fi router inside the home of a customer to a multicast-capable Internet Service Provider (ISP). In this scenario, the egress node discovery mechanism for the MNAT service might be a static configuration for the MNAT service's hostname, pushed by the ISP to the CPE devices.

For a bump in the host egress node, the discovery of the MNAT service might either operate via DNS-SD [RFC6763] using a search domain for the ISP distributed to hosts via a DHCP Domain Search option [RFC3397], or via configuration instructions the ISP gives to their customers to configure a search domain for their devices, or to configure the MNAT service's hostname for that ISP in their applications.

2.2. Service Discovery

It is RECOMMENDED that egress devices in end-user operating systems or applications use DNS-SD [RFC6763] by default to discover an MNAT service within their containing networks. However, a network may require the use of other mechanisms, including options such as manual configuration, so implementors are advised to offer manual configuration options in addition to automatic discovery with DNS-SD.

As long as an MNAT client can find a valid hostname to use, it can connect to the given MNAT service and monitor changes to the address assignments within the network.

2.2.1. Detecting Invalid Services

TBD: recommendations for noticing and discontinuing use of MNAT services that report mappings that don't correspond to the mappings apparently in use in the client's local network (particularly from egress nodes).

2.3. RESTCONF Bootstrap

TBD: describe the RESTCONF validation and bootstrapping steps. Use the same section name from I-D.draft-ietf-mboned-dorms as a template, assuming it passes a wider review.

2.4. Message Handling

2.4.1. Notification Subscription

When possible, changes to the group assignments should be communicated with subscriptions to data model updates using a server push mechanism, for example as described in [RFC8641].

Where clients or servers do not support server push updates, long polling can be used instead to provide timely updates. See [RFC6202] for an explanation of the approach and a discussion of its pros and cons.

If long polling and server push are both unavailable, MNAT clients may need to poll the server to monitor updates instead. This approach is likely to encounter delays in the detection of changes to mapping decisions within the MNAT service, but can be used as a last resort for providing multicast connectivity where the use of MNAT is required by a network to enable multicast forwarding.

2.4.2. Watcher Keys

MNAT clients open a persistent connection to the MNAT service and request allocation of a watcher key with the get-new-watcher-key Remote Procedure Call (RPC). Watcher keys are identifiers chosen by the MNAT service and communicated to client nodes in the response to a successful get-new-egress-key RPC. Watcher keys SHOULD be based on a random value and unique per new key requested.

Egress nodes communicate an interest in global (S,G)s by posting updates to the egress-global-joined container under a watcher with id equal to their watcher-key.

Ingress nodes communicate an interest in sets of global (S,G)s by providing a monitor object with a matching filter under a watcher with id equal to their watcher-key.

Watcher-keys expire if the refresh-watcher-id rpc is not invoked within the refresh-period given in the response to the get-new-watcher-id rpc.

TBD: better explanation about how the service times out egress nodes that don't refresh their egress key on schedule, and how egress nodes that reconnect can attempt to refresh the prior key they were using, but must request a new one on error. Probably define a state per egress key (e.g. active vs. recently expired vs. non-existent) for the MNAT service to maintain. Explain how the MNAT service should use population count from the egress joins to make prioritization

decisions for the assignment of flows when there is limited flow space. Probably reference CBACC in that explanation (I-D.draft-ietf-mboned-cbacc).

2.4.3. Egress Group Management

The egress-global-joined container in the YANG model provides a mechanism for egress nodes to directly advertise their group membership to the MNAT service for externally addressed (S,G)s.

Egress nodes advertise their group membership to external (S,G)s to the MNAT service and also advertise group membership to their next-hop router using IGMP or MLD for the locally mapped addressing within the network. Joins and leaves for the locally mapped network addresses occur in response to downstream joins for an external (S,G) that has or gains a mapping according to the MNAT service, when the join or leave propagates to the egress node.

Payloads of the locally mapped traffic should be treated as though they were carried in packets addressed as the external (S,G), including any authentication checks that should be performed for the traffic. Egress nodes that forward traffic (non-virtual egress nodes) will perform an address translation from the locally mapped addressing to the original (S,G) (according to the address mapping the MNAT service provides) before forwarding packets matching a locally mapped address. It is the responsibility of the MNAT service and the network that operates it to ensure that multiple different traffic streams are not merged to the same locally mapped addresses in a way that collides.

TBD: describe the effects of transient and persistent collisions?

2.4.4. Ingress Considerations

Like egress nodes, ingress nodes monitor the assignments provided by the MNAT service and perform network address translation and group membership propagation. Ingress nodes perform the translation from an external (S,G) to the internally mapped addressing for the local network transport.

In general, ingress nodes are translating traffic before the in-network multicast fanout to multiple egress nodes. So an ingress node is generally assumed to be feeding one or more egress nodes. Because one ingress node can feed many egress nodes, ingress nodes should be given priority ahead of egress nodes for notifications about changes to the address mapping from the MNAT service.

2.4.5. MNAT Service Considerations

The details of the address assignment strategies used by the internal logic of the MNAT service are out of scope for this document. Different instances of MNAT services are expected to use a wide range of considerations specific to the networks in which the instances operate.

However, outside of address assignment there are some operational points an MNAT service instance should take into consideration:

1. Assignment Transition Grace Period

It's recommended to provide a grace period between reassigning a local address mapping to a new external (S,G) after unassigning its mapping to an old (S,G). The grace period should account for the expected time for the connected ingress and egress nodes to process the unassigning of the external (S,G) and for egress nodes to perform leave operations for the old locally mapped address, and for the leave operations to propagate through the network. For most networks, 250 seconds is a good default, as this allows a usually sufficient time for IGMP and MLD membership to time out and for any resulting prune operations to propagate through the network. However, different networks may tune the grace period differently for a variety of operational considerations.

2. Scaling

The MNAT service should be appropriately provisioned to support the expected number of ingress and egress nodes within the network. In an eyeball network, restrictions on the number of egress nodes per shared receiver IP address may be appropriate in order to prevent a rogue client application from forming an excessive number of egress connections. Alternately, for bump-in-the-wire deployments of egress nodes in CPE devices it may be appropriate to authenticate the egress connections with a client certificate for each home to avoid denial of service attacks based on overloading the MNAT service with egress connections.

Additionally, it's RECOMMENDED to provide per-egress limits on the number of external simultaneous (S,G)s permitted per egress at a level appropriate to the scaling limitations for the network, to prevent denial of service attacks based on overloading the group assignments from a single malicious egress node.

2.4.6. Example Messaging Walkthrough

TBD: show what an expected example message sequence or 2 would look like.

3. YANG Model

3.1. Yang Tree

The tree diagram below uses the notation defined in [RFC8340].

```

module: ietf-mnat
  +--rw egress-global-joined
    |   +--rw watcher* [id]
    |     +--rw id          watcher-key
    |     +--rw joined-sg* [id]
    |       +--rw id          string
    |       +--rw (channel-type)?
    |         +--:(ssm-channel)
    |           +--rw source      inet:ip-address
    |           +--rw group
    |             rt-types:ip-multicast-group-address
    |         +--:(asm-channel)
    |           +--rw asm-group
    |             rt-types:ip-multicast-group-address
  +--rw ingress-watching
    |   +--rw watcher* [id]
    |     +--rw id          watcher-key
    |     +--rw monitor* [id]
    |       +--rw id          string
    |       +--rw (monitor-type)?
    |         +--:(monitor-global-sources)
    |           +--rw global-source-prefix  inet:ip-prefix
  +--ro assigned-channels
    |   +--ro watcher* [id]
    |     +--ro id          watcher-key
    |     +--ro mapped-sg* [id]
    |       +--ro id          assignment-id
    |       +--ro state        assignment-state
    |     +--ro global-subscription
    |       +--ro (channel-type)?
    |         +--:(ssm-channel)
    |           +--ro source      inet:ip-address
    |           +--ro group
    |             rt-types:ip-multicast-group-address
    |         +--:(asm-channel)
    |           +--ro asm-group
    |             rt-types:ip-multicast-group-address

```

```

+--ro local-mapping
+--ro (mapping-type)?
+--:(local-multicast-mapping)
+--ro (channel-type)?
+--:(ssm-channel)
|   +--ro source          inet:ip-address
|   +--ro group
|       rt-types:ip-multicast-group-address
+--:(asm-channel)
+--ro asm-group
    rt-types:ip-multicast-group-address

rpcs:
+---x get-new-watcher-id
|   +--ro output
|   +--ro watcher-id      watcher-key
|   +--ro refresh-period? uint16
+---x refresh-watcher-id
|   +---w input
|   |   +---w watcher-id    watcher-key
+--ro output
+--ro refresh-period?    uint16

```

Figure 1: MNAT Tree Diagram

3.2. Yang Module

```
<CODE BEGINS>
file ietf-mnat@2022-03-07.yang
module ietf-mnat {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-mnat";
  prefix mnat;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference "RFC 8294";
  }

  organization
    "IETF MBONED (Multicast Backbone Deployment) Working Group";
```


contact

"WG Web: <<https://datatracker.ietf.org/wg/mboned/>>
WG List: <<mailto:mboned@ietf.org>>

Author: Jake Holland
<<mailto:jakeholland.net@gmail.com>>;

description

"Multicast Network Address Translation Model.

Copyright (c) 2012 - 2020 IETF Trust and the persons
identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject
to the license terms contained in, the Simplified BSD
License set forth in Section 4.c of the IETF Trust's
Legal Provisions Relating to IETF Documents
(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see
the RFC itself for full legal notices.";

```
revision "2020-10-22" {  
  description  
    "Initial version.";  
}
```

```
grouping multicast-channel {  
  choice channel-type {  
    description  
      "ASM or SSM multicast channels can be represented.";  
    case ssm-channel {  
      leaf source {  
        type inet:ip-address;  
        mandatory true;  
        description  
          "Source address of a multicast channel";  
      }  
      leaf group {  
        type rt-types:ip-multicast-group-address;  
        mandatory true;  
        description "The global (S,G)'s group address";  
      }  
    }  
    case asm-channel {  
      leaf asm-group {  
        type rt-types:ip-multicast-group-address;  
      }  
    }  
  }  
}
```

```
        mandatory true;
        description "The global (S,G)'s group address";
    }
}
}

grouping monitor-definition {
    choice monitor-type {
        description
            "Definition of monitor characteristics.";
        case monitor-global-sources {
            leaf global-source-prefix {
                type inet:ip-prefix;
                mandatory true;
                description
                    "Prefix to match for source IPs.";
            }
        }
    }
}

typedef watcher-key {
    type string;
    description
        "A key for egress identification.";
}

typedef assignment-id {
    type uint32;
    description
        "A type for assignment identifiers.";
}

identity assignment-state {
    description
        "Base identity to represent assignment states";
}

typedef assignment-state {
    type identityref {
        base assignment-state;
    }
    description "Status of an assigned (S,G).";
}

identity unassigned {
    base assignment-state;
}
```

```
    description
      "Represents an unassigned global (S,G) that cannot be
       received in the local network.";
  }

  identity assigned-local-multicast {
    base assignment-state;
    description
      "Represents an assigned global (S,G) that can be
       received in the local network by joining the associated
       local-mapping.";
  }

  container egress-global-joined {
    description
      "Declarations of subscriptions to global (S,G)s per
       egress.";

    list watcher {
      key "id";
      description
        "Mappings of traffic that correspond to the registered
         interest list for a given watch id (from the
         get-new-watcher-id rpc)";
      leaf id {
        type watcher-key;
        description
          "Identifier from get-new-watcher-id. Tracks assignments
           of interest to the specific watcher.";
      }
      list joined-sg {
        key "id";
        leaf id {
          type string;
          description
            "id of the joined (S,G)";
        }
        description
          "(S,G)s in the global address space that an egress is
           joined to. These should get corresponding entries in
           the assigned-channels lists.";
        uses multicast-channel;
      }
    }
  }

  container ingress-watching {
    description
      "Matches on (S,G)s that get ingested from this ingress.";
```

```
list watcher {
  key "id";
  description
    "Mappings of traffic that correspond to the registered
    interest list for a given watch id (from the
    get-new-watcher-id rpc)";
  leaf id {
    type watcher-key;
    description
      "Identifier from get-new-watcher-id. Tracks assignments
      of interest to the specific watcher.";
  }
  list monitor {
    key "id";
    leaf id {
      type string;
      description
        "id of the monitor definition";
    }
    uses monitor-definition;
  }
}

container assigned-channels {
  config false;
  description
    "MNAT mappings of global (S,G)s into a local transport.";

  list watcher {
    key "id";
    description
      "Mappings of traffic that correspond to the registered
      interest list for a given watch id (from the
      get-new-watcher-id rpc)";
    leaf id {
      type watcher-key;
      description
        "Identifier from get-new-watcher-id. Tracks assignments
        of interest to the specific watcher.";
    }
  }
  list mapped-sg {
    key "id";
    description
      "The local network's assignment of global channels to
      local transport characteristics.";

    leaf id {
      type assignment-id;
    }
  }
}
```

```
        mandatory true;
        description
            "Identifier for this assignment.";
    }
    leaf state {
        type assignment-state;
        mandatory true;
        description
            "Status of the global (S,G)s that are assigned in the
            local network.";
    }
    container global-subscription {
        description
            "The global channel that's mapped.";
        uses multicast-channel;
    }
    container local-mapping {
        choice mapping-type {
            description
                "The description of how the global channel is
                transported within the local network";

            case local-multicast-mapping {
                description
                    "Defines the use of a local multicast (S,G) or
                    (*,G).";
                uses multicast-channel;
            }
        }
    }
}

rpc get-new-watcher-id {
    description
        "Obtain a secret key unique to an individual mnat-egress
        instance, assigned by the server and used for subscription
        management.";
    output {
        leaf watcher-id {
            type watcher-key;
            mandatory true;
            description
                "Identifier for assignment monitoring.";
        }
        leaf refresh-period {
            type uint16;
        }
    }
}
```

```
        default 10;
        description
            "Number of seconds to wait between refresh messages.";
    }
}
}
rpc refresh-watcher-id {
    description
        "A secret key unique to an individual mnat-egress instance,
        assigned by the server and used for subscription
        management.";
    input {
        leaf watcher-id {
            type watcher-key;
            mandatory true;
            description
                "Egress identifier for assignment monitoring.";
        }
    }
    output {
        leaf refresh-period {
            type uint16;
            default 10;
            description
                "Number of seconds to wait between refresh messages.";
        }
    }
}
}
<CODE ENDS>
```

4. IANA Considerations

4.1. The YANG Module Names Registry

This document adds one YANG module to the "YANG Module Names" registry maintained at <https://www.iana.org/assignments/yang-parameters>. The following registrations are made, per the format in Section 14 of [RFC6020]:

```
name:      ietf-mnat
namespace: urn:ietf:params:xml:ns:yang:ietf-mnat
prefix:    mnat
reference: I-D.draft-jholland-mboned-mnat
```

4.2. The XML Registry

This document adds the following registration to the "ns" subregistry of the "IETF XML Registry" defined in [RFC3688], referencing this document.

URI: urn:ietf:params:xml:ns:yang:ietf-mnat

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

4.3. The Service Name and Transport Protocol Port Number Registry

This document adds one service name to the "Service Name and Transport Protocol Port Number Registry" maintained at <https://www.iana.org/assignments/service-names-port-numbers>. The following registrations are made, per the format in Section 8.1.1 of [RFC6335]:

Service Name:	mnat
Transport Protocol(s):	TCP, UDP
Assignee:	IESG <iesg@ietf.org>
Contact:	IETF Chair <chair@ietf.org>
Description:	The MNAT service (RESTCONF that includes ietf-mnat YANG model)
Reference:	I-D.draft-jholland-mboned-mnat
Port Number:	N/A
Service Code:	N/A
Known Unauthorized Uses:	N/A
Assignment Notes:	N/A

5. Security Considerations

TBD. (What, me worry?)

Notable points to cover:

- * communication with the MNAT service should be secured. RESTCONF does this, alternate methods should also do it.
- * separate authentication of the contents of the multicast traffic is recommended (e.g. with AMBI or TESLA). Probably it's not recommended for a network with MNAT to pass external traffic that does not provide authentication, and if the internal traffic is not authenticated, to segregate the internal from the external traffic in the MNAT assignment pools.

- * mistaken mappings can result in receipt of payloads for the wrong channel. This can happen transiently even during normal operation. Recommend some steps to mitigate and avoid (e.g. the grace period and the authentication-TBD: explain how they help)
- * Clients can (deliberately or accidentally) overload the service. Limits should be set to avoid disrupting traffic to the rest of the network.

6. Acknowledgements

Thanks to Lenny Giuliano and Sandy Zhang for their very helpful comments on this document.

7. References

7.1. Normative References

- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <<https://www.rfc-editor.org/info/rfc1112>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <<https://www.rfc-editor.org/info/rfc3376>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, DOI 10.17487/RFC4604, August 2006, <<https://www.rfc-editor.org/info/rfc4604>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.

- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<https://www.rfc-editor.org/info/rfc6763>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.

7.2. Informative References

- [IEEE-802.1Q] IEEE, "Local and Metropolitan Area Networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q, n.d., <<https://standards.ieee.org/findstds/standard/802.1Q-2011.html>>.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<https://www.rfc-editor.org/info/rfc2663>>.
- [RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, DOI 10.17487/RFC3397, November 2002, <<https://www.rfc-editor.org/info/rfc3397>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

- [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>>.
- [RFC6202] Loreto, S., Saint-Andre, P., Salsano, S., and G. Wilkins, "Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP", RFC 6202, DOI 10.17487/RFC6202, April 2011, <<https://www.rfc-editor.org/info/rfc6202>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [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, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC8815] Abrahamsson, M., Chown, T., Giuliano, L., and T. Eckert, "Deprecating Any-Source Multicast (ASM) for Interdomain Multicast", BCP 229, RFC 8815, DOI 10.17487/RFC8815, August 2020, <<https://www.rfc-editor.org/info/rfc8815>>.

Author's Address

Jake Holland
Akamai Technologies, Inc.
150 Broadway
Cambridge, MA 02144,
United States of America
Email: jakeholland.net@gmail.com