Mboned J. Holland
Internet-Draft K. Rose
Intended status: Standards Track Akamai Technologies, Inc.
Expires: May 3, 2021 October 30, 2020

Asymmetric Manifest Based Integrity
draft-ietf-mboned-ambi-01

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

Copyright Notice

Table of Contents

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, 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 computational overhead.

- 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.

Aymmetric 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 communicating packet "manifests" (described in Section 2.4) via an out-of-band communication channel that provides authentication and verifiable integrity.

Each manifest contains a message digest (described in Section 2.3) 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.

Each manifest MUST be delivered in a way that provides cryptographic integrity guarantees of the authenticity of the manifest. For example, TLS could be used to deliver a stream of manifests over a unicast data stream from a set of trusted senders to each receiver, or a protocol that asymmetrically signs each message could be used to transport authenticated manifests over a multicast channel. Note that a UDP-based protocol might drop or reorder manifests while still providing authentication.

Upon successful verification of a manifest and receipt of any subset of the corresponding data packets, the receiver has proof of the integrity of the contents of the data packets that are listed in the manifest.

Authenticating the integrity of the data packets depends on:

- o the authenticity of the manifests
- o the authenticity of the digest profile used for construction of the packet digests
- o the difficulty of generating a collision for the packet digests contained in the manifest.

This document defines a YANG [RFC7950] module that augments the DORMS [I-D.draft-ietf-mboned-dorms-00] YANG module to provide a way to communicate a digest profile, described in Section 2.3.1, for construction of the packet digests, described in Section 2.3. When obtaining the digest profile by using DORMS, the authenticity of the data stream relies on a trust relationship with the DORMS server, since that anchors the authenticity of the digest profile for constructing packet digests.

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, as measured in the amount of extra data required, than TESLA imposes. 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 7. By contrast, TESLA requires a MAC to be added to each authenticated message.

1.2. Threat Model

TBD: Summarize the applicable threat model this protects against. A diagram plus a cleaned-up version of the on-list explanation here is probably appropriate: https://mailarchive.ietf.org/arch/msg/mboned/CG9FLjPwuno3MtvYvgNcD5p69I4/

Reference [RFC3552] https://tools.ietf.org/html/rfc3552#section-3

1.3. 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.

1.4. 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/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/

2. Protocol Operation

2.1. Overview

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

- o the data packet

- o an authenticated manifest containing the packet digest for the data packet

- o 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.

2.2. Buffering and Validation Windows

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 manifest stream and can be authenticated, 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.

Once a data packet is authenticated, the corresponding packet digest can be discarded and the data packet can be further processed by the receiving application or forwarded through the receiving network. Authenticating a data packet consumes one packet digest and prevents re-learning, with a hold-down time equal to the hold time for packet digests. A different manifest might provide the same packet digest with the same packet sequence number, but the digest remains consumed if it has been used to authenticate a data packet.

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 SHOULD be discarded. (Note that in some cases, packet digests can be sent redundantly in more than one manifest. In such cases, the latest received time for an authenticated packet digest should be used for the expiration time.)

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 5 provides a mechanism for senders to communicate the sender's recommendation for buffering durations, when using DORMS.

Receivers SHOULD follow the recommendations for hold times provided by the sender, subject to their capabilities and any administratively configured limits on buffer sizes at the receiver.

However receivers MAY deviate from the values recommended by the sender for a variety of reasons. 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 constraints on some devices.

2.2.1. 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. Examples of such techniques include paced chirping and pathrate.

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.

2.3. Packet Digests

2.3.1. 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 2.3.1.1)

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: there should also be a way to specify that only packets to a specific UDP port are applicable. I think this is not quite right today and probably should be done with a grouping in the yang model, so that the profile appears either inside a "protocol" container inside the (S,G) or inside the udp-stream inside the "protocol", but am not sure. Follow-up on this after the first reference implementation...

TBD: As recommended by https://tools.ietf.org/html/rfc7696#section-2.2 [1], a companion document containing the mandatory-to-implement cipher suite should also be published separately and referenced by this document.

2.3.1.1. Payload Type

2.3.1.1.1. UDP vs. IP payload validation

When the digest profile indicates that UDP payloads are validated, the IP protocol for the packets MUST be 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 digest profile indicates that IP payloads are validated, the IP payload of the packet is used, using the outermost IP layer that contains the (S,G) corresponding to the (S,G) protected by the manifest. 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.

2.3.1.1.2. 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.

2.3.2. 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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Source Address (32 bits IPv4/128 bits IPv6)           |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Destination Address (32 bits IPv4/128 bits IPv6)        |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Zeroes     |   Protocol    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Manifest Identifier                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Payload Data                          |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.3.2.1. Source Address

The IPv4 or IPv6 source address of the packet.

2.3.2.2. Destination Address

The IPv4 or IPv6 destination address of the packet.

2.3.2.3. Zeroes

All bits set to 0.

2.3.2.4. Protocol

The IP Protocol field from IPv4, or the Next Header field for IPv6. When UDP payload is indicated, this value MUST be UDP (0x11).

2.3.2.5. Length

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

2.3.2.6. Source Port

The source port of the packet. Zeroes if using a protocol that does not use source ports.

2.3.2.7. Destination Port

The destination port of the packet. Zeroes if using a protocol that does not use destination ports.

TBD: there's something I hate about the source and destination ports. Maybe it should only be active in UDP-payload mode, instead of zeroes when not UDP? But I suspect there's a better approach than UDP-or-not, so it's this way for now, with hopes of finding something better in the next version.

2.3.2.8. Manifest Identifier

The 32-bit identifier for the manifest stream.

2.3.2.9. Payload Data

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

The payload type is configurable because when sending UDP, some legacy networks may strip the UDP option space, and it's necessary to provide a manifest stream capable of authentication that can interoperate with these networks. However, for non-UDP traffic or in order to authenticate the UDP options, some use cases may require support for authenticating the full IP payload.

2.4. Manifests

2.4.1. Manifest Layout

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Manifest Stream Identifier                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Manifest sequence number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 First packet sequence number                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Refresh Deadline        |      Packet Digest Count      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 ... Packet Content Expansions ...             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      ... Packet Digests ...                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.4.1.1. Manifest Stream Identifier

A 32-bit unsigned integer chosen by the sender.

2.4.1.2. 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.

2.4.1.3. 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 overapping sets of packet sequence numbers. When received, these reset the hold timer for the listed packet digests.

2.4.1.4. Refresh Deadline

A 16-bit unsigned integer number of seconds.

A zero value means the current digest profile for the current manifest stream is stable.

A nonzero value means that the authentication is transitioning to a new manifest stream, and the set of digest profiles SHOULD be refreshed by receivers that might stay joined longer than this duration, and a different manifest stream SHOULD be selected, before this many seconds have elapsed, in order to avoid a disruption. See Section 2.5.

2.4.1.5. Packet Digest Count

The count of packet digests in the manifest.

2.4.1.6. Packet Digests

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

2.5. 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 field 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 end.

The receivers SHOULD use a random value between now and one half the number of seconds in the deadline field, to spread the spike of load on the DORMS server during a large multicast event.

3. Transport Considerations

3.1. 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: extend the 'manifest-transport' in the YANG model to make an extensible mechanism to advertise different transport options for receiving manifest streams.

TBD: add ALTA to the list when and if it gets further along [I-D.draft-krose-mboned-alta-01]. 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: 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.

3.2. HTTPS

This document defines a new media type 'application/ambi' for use with HTTPS.

An HTTPS stream carrying the 'application/ambi' media type is composed of a sequence of binary AMBI manifests. It is RECOMMENDED to use Chunked encoding.

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

3.3. DTLS

TBD: DTLS [RFC6347] can provide authentication for datagrams, so if manifests can be constructed to fit within datagrams, it is an appropriate choice. (IPSec is similar-worth adding as an option?).

This option provides no native redundancy or retransmission, but packet digests can be repeated in different manifests to provide some resilience to loss. Lost manifests that result in the loss of blocks of packet digests can be expensive, since they would make received data packets unauthenticatable. TBD: should we therefore not support this case? (probably still worthwhile-the manifests can still contain redundant hashes)

3.4. DTLS + FECFRAME

DTLS for manifests that do not fit into single-packet payloads can still be delivered by using FECFRAME [RFC6363], particularly Reed-Solomon [RFC6865] or possibly Raptor [RFC6681]. This has some advantages compared to HTTPS because of the absence of HOL-blocking, while providing for tunable redundancy. This has some advantages relative to DTLS because of overhead reduction and non-integer redundancy tunability (e.g. 1.5 becomes a viable redundancy factor).

TBD: define this method, possibly in another RFC.

4. Examples

TBD: walk through some examples as soon as we have a build running. Likely to need some touching up.

5. YANG Module

5.1. Tree Diagram

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

```
module: ietf-ambi
  augment /dorms:metadata/dorms:sender/dorms:group/dorms:udp-stream:
    +--rw manifest-stream* [id]
       +--rw id                     uint32
       +--rw manifest-transport*    inet:uri
       +--rw hash-algorithm         iha:hash-algorithm-type
       +--rw payload-type           enumeration
       +--rw data-hold-time-ms?     uint32
       +--rw digest-hold-time-ms?   uint32
```

5.2. Module

```
<CODE BEGINS> file ietf-ambi@2020-10-31.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";
        }

        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 2019-08-25 {
            description "Initial revision as an extension.";
            reference
```

```
             "";
       }

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

           list manifest-stream {
               key id;
               description "Definition of a manifest stream.";
               leaf id {
                   type uint32;
                   mandatory true;
                   description
                     "The Manifest ID referenced in a manifest.";
               }
               leaf-list manifest-transport {
                   type inet:uri;
                   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 payload-type {
                   type enumeration {
                       enum udp {
                         description "The hash includes only the UDP
                             payload.";
                       }
                       enum ip {
                         description "The hash includes the full IP
                             payload.";
                       }
                   }
                   mandatory true;
                   description "The contents of the payload for the
                       digest profile";
               }
               leaf data-hold-time-ms {
                   type uint32;
                   default 2000;
                   description
```

```
                      "The number of milliseconds to hold data
                       packets waiting for a corresponding digest before
                       discarding";
                }
                leaf digest-hold-time-ms {
                    type uint32;
                    default 10000;
                    description
                      "The number of milliseconds to hold packet
                       digests waiting for a corresponding data packet
                       before discarding";
                }
            }
        }
    }

    <CODE ENDS>
```

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

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-ambi
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

6.3. Media Type

TBD: Register ’application/ambi’ according to advice from: https://www.iana.org/form/media-types

TBD: check guidelines in https://tools.ietf.org/html/rfc5226

7. Security Considerations

7.1. 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. Acknowledgements

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

9. References

9.1. Normative References

[I-D.draft-ietf-mboned-dorms-00]
Holland, J., "Discovery Of Restconf Metadata for Source-specific multicast", draft-ietf-mboned-dorms-00 (work in progress), March 2020.

[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>.

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, DOI 10.17487/RFC6363, October 2011, <https://www.rfc-editor.org/info/rfc6363>.

[RFC6681] Watson, M., Stockhammer, T., and M. Luby, "Raptor Forward Error Correction (FEC) Schemes for FECFRAME", RFC 6681, DOI 10.17487/RFC6681, August 2012, <https://www.rfc-editor.org/info/rfc6681>.

[RFC6865] Roca, V., Cunche, M., Lacan, J., Bouabdallah, A., and K. Matsuzono, "Simple Reed-Solomon Forward Error Correction (FEC) Scheme for FECFRAME", RFC 6865, DOI 10.17487/RFC6865, February 2013, <https://www.rfc-editor.org/info/rfc6865>.

[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>.

9.2. Informative References

[I-D.draft-krose-mboned-alta-01]
Rose, K. and J. Holland, "Asymmetric Loss-Tolerant Authentication", draft-krose-mboned-alta-01 (work in progress), July 2019.

[I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-08 (work in progress), September 2019.

[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., 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>.

9.3. URIs

[1] https://tools.ietf.org/html/rfc7696#section-2.2

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 J. Holland
Internet-Draft Akamai Technologies, Inc.
Intended status: Standards Track February 01, 2021
Expires: August 5, 2021

Circuit Breaker Assisted Congestion Control
draft-ietf-mboned-cbacc-02

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

Copyright Notice

Table of Contents

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-00] 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.

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-bits-per-second" 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 the interval given by the "data-rate-window" field, which is the measurement interval.

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 forwarding bursts. If a CB node observes a higher data rate within any measurement window, it MAY circuit-break that flow immediately.

In the terminology of [RFC8084], this 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 costraints 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-00], 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-00]. 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-bit-rate 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 [PathChirp], if available.

   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 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 detsils, 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:metadata/dorms:sender/dorms:group:
    +--rw cbacc!
       +--rw max-bits-per-second    uint32
       +--rw max-mss?               uint16
       +--rw data-rate-window?      uint32
       +--rw priority?              uint16
```

3.2. Module

```
<CODE BEGINS> file ietf-cbacc@2021-02-01.yang
module ietf-cbacc {
    yang-version 1.1;

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

```
       prefix "ambi";

       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-jholland-mboned-dorms).";

       revision 2021-01-15 {
           description "Initial revision as an extension.";
           reference
             "";
       }

       augment
         "/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-bits-per-second {
               type uint32;
               mandatory true;
               description "Maximum bitrate for this stream, in Kilobits
                   of IP packet data (including headers) of native
                   multicast traffic per second";
           }
           leaf max-mss {
               type uint16;
               default 1400;
               description "Maximum payload size, in bytes";
           }
           leaf data-rate-window {
               type uint32;
               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

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, and Dino Farinacci for their thoughtful comments and contributions.

7. References

7.1. Normative References

[I-D.draft-ietf-mboned-ambi-00]
Holland, J. and K. Rose, "Asymmetric Manifest Based Integrity", draft-ietf-mboned-ambi-00 (work in progress), March 2020.

[I-D.draft-ietf-mboned-dorms-00]
Holland, J., "Discovery Of Restconf Metadata for Source-specific multicast", draft-ietf-mboned-dorms-00 (work in progress), March 2020.

[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

[FLID-DL] Byers, J., 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., Riedi, R., Baraniuk, R., 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., "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes", 1999.

[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.

[RLM] McCanne, S., Jacobson, V., Vetterli, M., University of California, Berkeley, and Lawrence Berkeley National Laboratory, "Receiver-driven Layered Multicast", 1995.

[SMCC] Kwon, G., Byers, J., and Computer Science Department, Boston University, "Smooth Multirate Multicast Congestion Control", 2002.

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 J. Holland
Internet-Draft Akamai Technologies, Inc.
Intended status: Standards Track October 30, 2020
Expires: May 3, 2021

Discovery Of Restconf Metadata for Source-specific multicast
draft-ietf-mboned-dorms-01

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

Copyright Notice

Table of Contents

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 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]

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

DORMS provides a framework that can be extended to publish supplemental information about multicas traffic in a globally discoverable manner. This is useful so that entities engaged in delivery or processing of the traffic that are not affiliated with the sender of the traffic and who may not otherwise have any means to discover information about the traffic, such as forwarding ISPs or operators of firewalls providing security guarantees to their users, can discover the information they may need in order to process the traffic according to their requirements.

1.3.1. Use cases

For example, a network that is capable of forwarding multicast traffic may need to take provisioning actions or make admission control decisions at ingress points based on the expected bitrate of the traffic in order to prevent oversubscription of the network.

Other use cases may include metadata that can be used to authenticate the multicast traffic, metadata that describes the contents of the traffic, metadata that makes assertions about the legal status of the traffic within specific contexts, or metadata that describes the protocols or applications that can be used to consume the traffic. Extensions to DORMS to support these or other kinds of metadata can be defined by later specifications.

Detailing the specific of the 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.

1.3.2. Channel Selection

In general, a DORMS client might learn of an (S,G) by any means. Therefore, describing the full set of 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 to give a few examples, 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 futher upstream in the network.

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

1.4. 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/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).

o Join: https://www.ietf.org/mailman/listinfo/mboned

o Search: https://mailarchive.ietf.org/arch/browse/mboned/

1.4.2. 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.

o 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).

2. Discovery and Metdata 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 with information about the 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, a receiver or middlebox 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, while handling a join for (203.0.113.15, 232.1.1.1), a receiver would perform a DNS query for the SRV RRType for the domain:

```
_dorms._tcp.15.113.0.203.in-addr.arpa.
```

The DNS response for this domain might return a record such as:

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

This response informs the receiver 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.

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 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 an hour and no longer than 3 days.

2.3. RESTCONF Bootstrap

Once a DORMS host 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, 27 Aug 2019 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, 27 Aug 2019 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,2016-08-15
Host: dorms-restconf.example.com
Accept: application/yang-data+json
```

The server might respond as follows:

```
HTTP/1.1 200 OK
Date: Tue, 27 Aug 2019 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": "2019-08-25",
      "schema":
          "https://example.com/yang/ietf-dorms@2019-08-25.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:metadata/\
    sender=203.0.113.15/group=232.1.1.1
Host: dorms-restconf.example.com
Accept: application/yang-data+json
```

The server might respond as follows:

```
HTTP/1.1 200 OK
Date: Tue, 27 Aug 2019 20:56:02 GMT
Server: example-server
Cache-Control: no-cache
Content-Type: application/yang-data+json

{
  "ietf-dorms:group": [
    {
      "group-address":"232.1.1.1",
      "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 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

In the absence of contextual information, 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 MAY use heuristics or out of band information about the service to 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. 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. Example specified use cases include providing authentication information [I-D.draft-ietf-mboned-ambi-00] and bit-rate information [I-D.draft-ietf-mboned-cbacc-00] for use by receivers and middle boxes, but more use cases are anticipated.

4.1. Yang Tree

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

```
module: ietf-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
```

DORMS Tree Diagram

4.2. Yang Module

```
<CODE BEGINS> file ietf-dorms@2020-10-31.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 2019-08-25 {
           description "Initial revision.";
           reference
               "I-D.draft-ietf-mboned-dorms";
       }

       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).";
                   }

                   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, as expected in a browser deployment, this can expose new information about end users relative to services based solely on multicast streams.

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].

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 control access to data nodes.

No data nodes defined in this YANG module are writable, creatable, or deletable. This YANG module is intended for publication of read-only data according to a well-defined schema.

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:metadata" 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:metadata" 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 Boostrap 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 [RFC2845] 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-00]), 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-00]).

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, and Lenny Guiliano 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>.

[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>.

[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>.

[whatwg-fetch] "WHATWG Fetch Living Standard", October 2020, <https://fetch.spec.whatwg.org/>.

9.2. Informative References

[I-D.draft-ietf-mboned-ambi-00] Holland, J. and K. Rose, "Asymmetric Manifest Based Integrity", draft-ietf-mboned-ambi-00 (work in progress), March 2020.

[I-D.draft-ietf-mboned-cbacc-00] Holland, J., "Circuit Breaker Assisted Congestion Control", draft-ietf-mboned-cbacc-00 (work in progress), March 2020.

[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>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <https://www.rfc-editor.org/info/rfc2845>.

[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>.

[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>.

[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>.

[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>.

Author's Address

Jake Holland
Akamai Technologies, Inc.
150 Broadway
Cambridge, MA 02144
United States of America

Email: jakeholland.net@gmail.com

Internet Area C. Perkins
Internet-Draft Blue Meadow Networks
Intended status: Informational M. McBride
Expires: August 8, 2021 Futurewei
D. Stanley
HPE
W. Kumari
Google
JC. Zuniga
SIGFOX
February 4, 2021

Multicast Considerations over IEEE 802 Wireless Media
draft-ietf-mboned-ieee802-mcast-problems-13

Abstract

Well-known issues with multicast have prevented the deployment of multicast in 802.11 (wifi) and other local-area wireless environments. This document describes the problems of known limitations with wireless (primarily 802.11) Layer-2 multicast. Also described are certain multicast enhancement features that have been specified by the IETF, and by IEEE 802, for wireless media, as well as some operational choices that can be taken to improve the performance of the network. Finally, some recommendations are provided about the usage and combination of these features and operational choices.

Table of Contents

1. Introduction

Well-known issues with multicast have prevented the deployment of multicast in 802.11 [dot11] and other local-area wireless environments, as described in [mc-props], [mc-prob-stmt]. Performance issues have been observed when multicast packet transmissions of IETF protocols are used over IEEE 802 wireless media. Even though enhancements for multicast transmissions have been designed at both IETF and IEEE 802, incompatibilities still exist between specifications, implementations and configuration choices.

Many IETF protocols depend on multicast/broadcast for delivery of control messages to multiple receivers. Multicast allows sending data to multiple interested recipients without the source needing to send duplicate data to each recipient. With broadcast traffic, data is sent to every device regardless of their interest in the data. Multicast is used for various purposes such as neighbor discovery, network flooding, address resolution, as well minimizing media occupancy for the transmission of data that is intended for multiple receivers. In addition to protocol use of broadcast/multicast for control messages, more applications, such as push to talk in hospitals, or video in enterprises, universities, and homes, are sending multicast IP to end user devices, which are increasingly using Wi-Fi for their connectivity.

IETF protocols typically rely on network protocol layering in order to reduce or eliminate any dependence of higher level protocols on the specific nature of the MAC layer protocols or the physical media. In the case of multicast transmissions, higher level protocols have traditionally been designed as if transmitting a packet to an IP address had the same cost in interference and network media access, regardless of whether the destination IP address is a unicast address or a multicast or broadcast address. This model was reasonable for networks where the physical medium was wired, like Ethernet. Unfortunately, for many wireless media, the costs to access the medium can be quite different. Multicast over Wi-Fi has often been plagued by such poor performance that it is disallowed. Some enhancements have been designed in IETF protocols that are assumed to work primarily over wireless media. However, these enhancements are usually implemented in limited deployments and not widespread on most wireless networks.

IEEE 802 wireless protocols have been designed with certain features to support multicast traffic. For instance, lower modulations are used to transmit multicast frames, so that these can be received by all stations in the cell, regardless of the distance or path attenuation from the base station or access point. However, these lower modulation transmissions occupy the medium longer; they hamper efficient transmission of traffic using higher order modulations to nearby stations. For these and other reasons, IEEE 802 working groups such as 802.11 have designed features to improve the performance of multicast transmissions at Layer 2 [ietf_802-11]. In addition to protocol design features, certain operational and configuration enhancements can ameliorate the network performance issues created by multicast traffic, as described in Section 5.

There seems to be general agreement that these problems will not be fixed anytime soon, primarily because it's expensive to do so and due to multicast being unreliable. Compared to unicast over Wi-Fi, multicast is often treated as somewhat of a second class citizen, even though there are many protocols using multicast. Something needs to be provided in order to make them more reliable. IPv6 neighbor discovery saturating the Wi-Fi link is only part of the problem. Wi-Fi traffic classes may help. This document is intended to help make the determination about what problems should be solved by the IETF and what problems should be solved by the IEEE (see Section 8).

This document details various problems caused by multicast transmission over wireless networks, including high packet error rates, no acknowledgements, and low data rate. It also explains some enhancements that have been designed at the IETF and IEEE 802.11 to ameliorate the effects of multicast traffic. Recommendations are also provided to implementors about how to use and combine these enhancements. Some advice about the operational choices that can be taken is also included. It is likely that this document will also be considered relevant to designers of future IEEE wireless specifications.

2. Terminology

This document uses the following definitions:

ACK
   The 802.11 layer 2 acknowledgement

AP

IEEE 802.11 Access Point

basic rate
: The slowest rate of all the connected devices, at which multicast and broadcast traffic is generally transmitted

DTIM
: Delivery Traffic Indication Map (DTIM): An information element that advertises whether or not any associated stations have buffered multicast or broadcast frames

MCS
: Modulation and Coding Scheme

NOC
: Network Operations Center

PER
: Packet Error Rate

STA
: 802.11 station (e.g. handheld device)

TIM
: Traffic Indication Map (TIM): An information element that advertises whether or not any associated stations have buffered unicast frames

3. Identified multicast issues

3.1. Issues at Layer 2 and Below

In this section some of the issues related to the use of multicast transmissions over IEEE 802 wireless technologies are described.

3.1.1. Multicast reliability

Multicast traffic is typically much less reliable than unicast traffic. Since multicast makes point-to-multipoint communications, multiple acknowledgements would be needed to guarantee reception at all recipients. Since there are no ACKs for multicast packets, it is not possible for the Access Point (AP) to know whether or not a retransmission is needed. Even in the wired Internet, this characteristic often causes undesirably high error rates. This has contributed to the relatively slow uptake of multicast applications even though the protocols have long been available. The situation for wireless links is much worse, and is quite sensitive to the presence of background traffic. Consequently, there can be a high packet error rate (PER) due to lack of retransmission, and because the sender never backs off. It is not uncommon for there to be a packet loss rate of 5% or more, which is particularly troublesome for video and other environments where high data rates and high reliability are required.

3.1.2. Lower and Variable Data Rate

Multicast over wired differs from multicast over wireless because transmission over wired links often occurs at a fixed rate. Wi-Fi, on the other hand, has a transmission rate that varies depending upon the STA's proximity to the AP. The throughput of video flows, and the capacity of the broader Wi-Fi network, will change and will impact the ability for QoS solutions to effectively reserve bandwidth and provide admission control.

For wireless stations associated with an Access Point, the power necessary for good reception can vary from station to station. For unicast, the goal is to minimize power requirements while maximizing the data rate to the destination. For multicast, the goal is simply to maximize the number of receivers that will correctly receive the multicast packet; generally the Access Point has to use a much lower data rate at a power level high enough for even the farthest station to receive the packet, for example as briefly mentioned in section 2 of [RFC5757]. Consequently, the data rate of a video stream, for instance, would be constrained by the environmental considerations of the least reliable receiver associated with the Access Point.

Because more robust modulation and coding schemes (MCSs) have longer range but also lower data rate, multicast / broadcast traffic is generally transmitted at the slowest rate of all the connected devices. This is also known as the basic rate. The amount of additional interference depends on the specific wireless technology. In fact, backward compatibility and multi-stream implementations mean that the maximum unicast rates are currently up to a few Gbps, so there can be more than 3 orders of magnitude difference in the transmission rate between multicast / broadcast versus optimal unicast forwarding. Some techniques employed to increase spectral efficiency, such as spatial multiplexing in MIMO systems, are not available with more than one intended receiver; it is not the case that backwards compatibility is the only factor responsible for lower multicast transmission rates.

Wired multicast also affects wireless LANs when the AP extends the wired segment; in that case, multicast / broadcast frames on the wired LAN side are copied to the Wireless Local Area Network (WLAN).

Since broadcast messages are transmitted at the most robust MCS, many large frames are sent at a slow rate over the air.

3.1.3. Capacity and Impact on Interference

Transmissions at a lower rate require longer occupancy of the wireless medium and thus take away from the airtime of other communications and degrade the overall capacity. Furthermore, transmission at higher power, as is required to reach all multicast STAs associated to the AP, proportionately increases the area of interference.

3.1.4. Power-save Effects on Multicast

One of the characteristics of multicast transmission is that every station has to be configured to wake up to receive the multicast, even though the received packet may ultimately be discarded. This process can have a large effect on the power consumption by the multicast receiver station. For this reason there are workarounds, such as Directed Multicast Service (DMS) described in Section 4, to prevent unnecessarily waking up stations.

Multicast can work poorly with the power-save mechanisms defined in IEEE 802.11e, for the following reasons.

- o Clients may be unable to stay in sleep mode due to multicast control packets frequently waking them up.
- o Both unicast and multicast traffic can be delayed by power-saving mechanisms.
- o A unicast packet is delayed until an STA wakes up and requests it. Unicast traffic may also be delayed to improve power save, efficiency and increase probability of aggregation.
- o Multicast traffic is delayed in a wireless network if any of the STAs in that network are power savers. All STAs associated to the AP have to be awake at a known time to receive multicast traffic.
- o Packets can also be discarded due to buffer limitations in the AP and non-AP STA.

3.2. Issues at Layer 3 and Above

This section identifies some representative IETF protocols, and describes possible negative effects due to performance degradation when using multicast transmissions for control messages. Common uses of multicast include:

- o Control plane signaling
- o Neighbor Discovery
- o Address Resolution

- Service Discovery
- Applications (video delivery, stock data, etc.)
- On-demand routing
- Backbone construction
- Other L3 protocols (non-IP)

User Datagram Protocol (UDP) is the most common transport layer protocol for multicast applications. By itself, UDP is not reliable -- messages may be lost or delivered out of order.

3.2.1. IPv4 issues

The following list contains some representative discovery protocols, which utilize broadcast/multicast, that are used with IPv4.

- ARP [RFC5424]
- DHCP [RFC2131]
- mDNS [RFC6762]
- uPnP [RFC6970]

After initial configuration, ARP (described in more detail later) and DHCP occur much less commonly, but service discovery can occur at any time. Some widely-deployed service discovery protocols (e.g., for finding a printer) utilize mDNS (i.e., multicast) which is often the first service that operators drop. Even if multicast snooping [RFC4541] (which provides the benefit of conserving bandwidth on those segments of the network where no node has expressed interest in receiving packets addressed to the group address) is utilized, many devices can register at once and cause serious network degradation.

3.2.2. IPv6 issues

IPv6 makes extensive use of multicast, including the following:

- DHCPv6 [RFC8415]
- Protocol Independent Multicast (PIM) [RFC7761]
- IPv6 Neighbor Discovery Protocol (NDP) [RFC4861]
- multicast DNS (mDNS) [RFC6762]
- Router Discovery [RFC4286]

IPv6 NDP Neighbor Solicitation (NS) messages used in Duplicate Address Detection (DAD) and Address Lookup make use of Link-Scope multicast. In contrast to IPv4, an IPv6 node will typically use multiple addresses, and may change them often for privacy reasons. This intensifies the impact of multicast messages that are associated to the mobility of a node. Router advertisement (RA) messages are also periodically multicasted over the Link.

Neighbors may be considered lost if several consecutive Neighbor Discovery packets fail.

3.2.3. MLD issues

Multicast Listener Discovery (MLD) [RFC4541] is used to identify members of a multicast group that are connected to the ports of a switch. Forwarding multicast frames into a Wi-Fi-enabled area can use such switch support for hardware forwarding state information. However, since IPv6 makes heavy use of multicast, each STA with an IPv6 address will require state on the switch for several and possibly many multicast solicited-node addresses. Multicast addresses that do not have forwarding state installed (perhaps due to hardware memory limitations on the switch) cause frames to be flooded on all ports of the switch. Some switch vendors do not support MLD, for link-scope multicast, due to the increase it can cause in state.

3.2.4. Spurious Neighbor Discovery

On the Internet there is a "background radiation" of scanning traffic (people scanning for vulnerable machines) and backscatter (responses from spoofed traffic, etc). This means that routers very often receive packets destined for IPv4 addresses regardless of whether those IP addresses are in use. In the cases where the IP is assigned to a host, the router broadcasts an ARP request, gets back an ARP reply, and caches it; then traffic can be delivered to the host. When the IP address is not in use, the router broadcasts one (or more) ARP requests, and never gets a reply. This means that it does not populate the ARP cache, and the next time there is traffic for that IP address the router will rebroadcast the ARP requests.

The rate of these ARP requests is proportional to the size of the subnets, the rate of scanning and backscatter, and how long the router keeps state on non-responding ARPs. As it turns out, this rate is inversely proportional to how occupied the subnet is (valid ARPs end up in a cache, stopping the broadcasting; unused IPs never respond, and so cause more broadcasts). Depending on the address space in use, the time of day, how occupied the subnet is, and other unknown factors, thousands of broadcasts per second have been observed. Around 2,000 broadcasts per second have been observed at the IETF NOC during face-to-face meetings.

With Neighbor Discovery for IPv6 [RFC4861], nodes accomplish address resolution by multicasting a Neighbor Solicitation that asks the target node to return its link-layer address. Neighbor Solicitation messages are multicast to the solicited-node multicast address of the target address. The target returns its link-layer address in a unicast Neighbor Advertisement message. A single request-response pair of packets is sufficient for both the initiator and the target to resolve each other's link-layer addresses; the initiator includes its link-layer address in the Neighbor Solicitation.

On a wired network, there is not a huge difference between unicast, multicast and broadcast traffic. Due to hardware filtering (see, e.g., [Deri-2010]), inadvertently flooded traffic (or excessive ethernet multicast) on wired networks can be quite a bit less costly, compared to wireless cases where sleeping devices have to wake up to process packets. Wired Ethernets tend to be switched networks, further reducing interference from multicast. There is effectively no collision / scheduling problem except at extremely high port utilizations.

This is not true in the wireless realm; wireless equipment is often unable to send high volumes of broadcast and multicast traffic, causing numerous broadcast and multicast packets to be dropped. Consequently, when a host connects it is often not able to complete DHCP, and IPv6 RAs get dropped, leading to users being unable to use the network.

4. Multicast protocol optimizations

This section lists some optimizations that have been specified in IEEE 802 and IETF that are aimed at reducing or eliminating the issues discussed in Section 3.

4.1. Proxy ARP in 802.11-2012

The AP knows the MAC address and IP address for all associated STAs. In this way, the AP acts as the central "manager" for all the 802.11 STAs in its basic service set (BSS). Proxy ARP is easy to implement at the AP, and offers the following advantages:

- o Reduced broadcast traffic (transmitted at low MCS) on the wireless medium
- o STA benefits from extended power save in sleep mode, as ARP requests for STA's IP address are handled instead by the AP.
- o ARP frames are kept off the wireless medium.
- o No changes are needed to STA implementation.

Here is the specification language as described in clause 10.23.13 of [dot11-proxyarp]:

> When the AP supports Proxy ARP "[...] the AP shall maintain a Hardware Address to Internet Address mapping for each associated station, and shall update the mapping when the Internet Address of the associated station changes. When the IPv4 address being

resolved in the ARP request packet is used by a non-AP STA currently associated to the BSS, the proxy ARP service shall respond on behalf of the non-AP STA".

4.2. IPv6 Address Registration and Proxy Neighbor Discovery

As used in this section, a Low-Power Wireless Personal Area Network (6LoWPAN) denotes a low power lossy network (LLN) that supports 6LoWPAN Header Compression (HC) [RFC6282]. A 6TiSCH network [I-D.ietf-6tisch-architecture] is an example of a 6LowPAN. In order to control the use of IPv6 multicast over 6LoWPANs, the 6LoWPAN Neighbor Discovery (6LoWPAN ND) [RFC6775] standard defines an address registration mechanism that relies on a central registry to assess address uniqueness, as a substitute to the inefficient DAD mechanism found in the mainstream IPv6 Neighbor Discovery Protocol (NDP) [RFC4861][RFC4862].

The 6lo Working Group has specified an update [RFC8505] to RFC6775. Wireless devices can register their address to a Backbone Router [I-D.ietf-6lo-backbone-router], which proxies for the registered addresses with the IPv6 NDP running on a high speed aggregating backbone. The update also enables a proxy registration mechanism on behalf of the registered node, e.g. by a 6LoWPAN router to which the mobile node is attached.

The general idea behind the backbone router concept is that broadcast and multicast messaging should be tightly controlled in a variety of WLANs and Wireless Personal Area Networks (WPANs). Connectivity to a particular link that provides the subnet should be left to Layer-3. The model for the Backbone Router operation is represented in Figure 1.

```
                 |
               +-----+
               |     | Gateway (default) router
               |     |
               +-----+
                  |
                  |      Backbone Link
            +--------------------+------------------+
            |                    |                  |
         +-----+             +-----+             +-----+
         |     | Backbone    |     | Backbone    |     | Backbone
         |     | router 1    |     | router 2    |     | router 3
         +-----+             +-----+             +-----+
            o                o   o  o              o o
        o o   o  o       o o   o  o  o         o  o  o  o o
       o  o o  o o       o   o  o  o  o        o  o  o o o
       o   o  o  o          o    o  o           o  o   o
         o   o o               o  o                 o o

           LLN 1              LLN 2                LLN 3
```

Figure 1: Backbone Link and Backbone Routers

LLN nodes can move freely from an LLN anchored at one IPv6 Backbone Router to an LLN anchored at another Backbone Router on the same backbone, keeping any of the IPv6 addresses they have configured. The Backbone Routers maintain a Binding Table of their Registered Nodes, which serves as a distributed database of all the LLN Nodes. An extension to the Neighbor Discovery Protocol is introduced to exchange Binding Table information across the Backbone Link as needed for the operation of IPv6 Neighbor Discovery.

RFC6775 and follow-on work [RFC8505] address the needs of LLNs, and similar techniques are likely to be valuable on any type of link where sleeping devices are attached, or where the use of broadcast and multicast operations should be limited.

4.3. Buffering to Improve Battery Life

Methods have been developed to help save battery life; for example, a device might not wake up when the AP receives a multicast packet. The AP acts on behalf of STAs in various ways. To enable use of the power-saving feature for STAs in its BSS, the AP buffers frames for delivery to the STA at the time when the STA is scheduled for reception. If an AP, for instance, expresses a DTIM (Delivery Traffic Indication Message) of 3 then the AP will send a multicast packet every 3 packets. In fact, when any single wireless STA associated with an access point has 802.11 power-save mode enabled, the access point buffers all multicast frames and sends them only after the next DTIM beacon.

In practice, most AP's will send a multicast every 30 packets. For unicast the AP could send a TIM (Traffic Indication Message), but for multicast the AP sends a broadcast to everyone. DTIM does power management but STAs can choose whether or not to wake up and whether or not to drop the packet. Unfortunately, without proper administrative control, such STAs may be unable to determine why their multicast operations do not work.

4.4. Limiting multicast buffer hardware queue depth

The CAB (Content after Beacon) queue is used for beacon-triggered transmission of buffered multicast frames. If lots of multicast frames were buffered, and this queue fills up, it drowns out all regular traffic. To limit the damage that buffered traffic can do, some drivers limit the amount of queued multicast data to a fraction of the beacon_interval. An example of this is [CAB].

4.5. IPv6 support in 802.11-2012

IPv6 uses NDP instead of ARP. Every IPv6 node subscribes to a special multicast address for this purpose.

Here is the specification language from clause 10.23.13 of [dot11-proxyarp]:

> "When an IPv6 address is being resolved, the Proxy Neighbor Discovery service shall respond with a Neighbor Advertisement message [...] on behalf of an associated STA to an [ICMPv6] Neighbor Solicitation message [...]. When MAC address mappings change, the AP may send unsolicited Neighbor Advertisement Messages on behalf of a STA."

NDP may be used to request additional information

- Maximum Transmission Unit
- Router Solicitation
- Router Advertisement, etc.

NDP messages are sent as group addressed (broadcast) frames in 802.11. Using the proxy operation helps to keep NDP messages off the wireless medium.

4.6. Using Unicast Instead of Multicast

It is often possible to transmit multicast control and data messages by using unicast transmissions to each station individually.

4.6.1. Overview

In many situations, it's a good choice to use unicast instead of multicast over the Wi-Fi link. This avoids most of the problems specific to multicast over Wi-Fi, since the individual frames are then acknowledged and buffered for power save clients, in the way that unicast traffic normally operates.

This approach comes with the tradeoff of sometimes sending the same packet multiple times over the Wi-Fi link. However, in many cases, such as video into a residential home network, this can be a good tradeoff, since the Wi-Fi link may have enough capacity for the unicast traffic to be transmitted to each subscribed STA, even though multicast addressing may have been necessary for the upstream access network.

Several technologies exist that can be used to arrange unicast transport over the Wi-Fi link, outlined in the subsections below.

4.6.2. Layer 2 Conversion to Unicast

It is often possible to transmit multicast control and data messages by using unicast transmissions to each station individually.

Although there is not yet a standardized method of conversion, at least one widely available implementation exists in the Linux bridging code [bridge-mc-2-uc]. Other proprietary implementations are available from various vendors. In general, these implementations perform a straightforward mapping for groups or channels, discovered by IGMP or MLD snooping, to the corresponding unicast MAC addresses.

4.6.3. Directed Multicast Service (DMS)

There are situations where more is needed than simply converting multicast to unicast. For these purposes, DMS enables an STA to request that the AP transmit multicast group addressed frames destined to the requesting STAs as individually addressed frames [i.e., convert multicast to unicast]. Here are some characteristics of DMS:

o Requires 802.11n A-MSDUs

- Individually addressed frames are acknowledged and are buffered for power save STAs
- The requesting STA may specify traffic characteristics for DMS traffic
- DMS was defined in IEEE Std 802.11v-2011
- DMS requires changes to both AP and STA implementation.

DMS is not currently implemented in products. See [Tramarin2017] and [Oliva2013] for more information.

4.6.4. Automatic Multicast Tunneling (AMT)

AMT[RFC7450] provides a method to tunnel multicast IP packets inside unicast IP packets over network links that only support unicast. When an operating system or application running on an STA has an AMT gateway capability integrated, it's possible to use unicast to traverse the Wi-Fi link by deploying an AMT relay in the non-Wi-Fi portion of the network connected to the AP.

It is recommended that multicast-enabled networks deploying AMT relays for this purpose make the relays locally discoverable with the following methods, as described in [I-D.ietf-mboned-driad-amt-discovery]:

- DNS-SD [RFC6763]
- the well-known IP addresses from Section 7 of [RFC7450]

An AMT gateway that implements multiple standard discovery methods is more likely to discover the local multicast-capable network, instead of forming a connection to a non-local AMT relay further upstream.

4.7. GroupCast with Retries (GCR)

GCR (defined in [dot11aa]) provides greater reliability by using either unsolicited retries or a block acknowledgement mechanism. GCR increases probability of broadcast frame reception success, but still does not guarantee success.

For the block acknowledgement mechanism, the AP transmits each group addressed frame as conventional group addressed transmission. Retransmissions are group addressed, but hidden from non-11aa STAs. A directed block acknowledgement scheme is used to harvest reception status from receivers; retransmissions are based upon these responses.

GCR is suitable for all group sizes including medium to large groups. As the number of devices in the group increases, GCR can send block acknowledgement requests to only a small subset of the group. GCR does require changes to both AP and STA implementations.

GCR may introduce unacceptable latency. After sending a group of data frames to the group, the AP has to do the following:

- unicast a Block Ack Request (BAR) to a subset of members.
- wait for the corresponding Block Ack (BA).
- retransmit any missed frames.
- resume other operations that may have been delayed.

This latency may not be acceptable for some traffic.

There are ongoing extensions in 802.11 to improve GCR performance.

- BAR is sent using downlink MU-MIMO (note that downlink MU-MIMO is already specified in 802.11-REVmc 4.3).
- BA is sent using uplink MU-MIMO (which is a .11ax feature).
- Additional 802.11ax extensions are under consideration; see [mc-ack-mux]
- Latency may also be reduced by simultaneously receiving BA information from multiple STAs.

5. Operational optimizations

This section lists some operational optimizations that can be implemented when deploying wireless IEEE 802 networks to mitigate some of the issues discussed in Section 3.

5.1. Mitigating Problems from Spurious Neighbor Discovery

ARP Sponges

An ARP Sponge sits on a network and learns which IP addresses are actually in use. It also listens for ARP requests, and, if it sees an ARP for an IP address that it believes is not used, it will reply with its own MAC address. This means that the router now has an IP to MAC mapping, which it caches. If that IP is later assigned to a machine (e.g using DHCP), the ARP sponge will see this, and will stop replying for that address. Gratuitous ARPs (or the machine ARPing for its gateway) will replace the sponged address in the router ARP table. This technique is quite effective; but, unfortunately, the ARP sponge daemons were not really designed for this use (one of the most widely deployed arp sponges [arpsponge], was designed to deal with the disappearance of participants from an IXP) and so are not optimized for this purpose. One daemon is needed per subnet, the tuning is tricky (the scanning rate versus the population rate versus retires, etc.) and sometimes daemons just stop, requiring a restart of the daemon which causes disruption.

Router mitigations

Some routers (often those based on Linux) implement a "negative ARP cache" daemon. Simply put, if the router does not see a reply to an ARP it can be configured to cache this information for some interval. Unfortunately, the core routers in use often do not support this. When a host connects to a network and gets an IP address, it will ARP for its default gateway (the router). The router will update its cache with the IP to host MAC mapping learned from the request (passive ARP learning).

Firewall unused space

The distribution of users on wireless networks / subnets may change in various use cases, such as conference venues (e.g SSIDs are renamed, some SSIDs lose favor, etc). This makes utilization for particular SSIDs difficult to predict ahead of time, but usage can be monitored as attendees use the different networks. Configuring multiple DHCP pools per subnet, and enabling them sequentially, can create a large subnet, from which only addresses in the lower portions are assigned. Therefore input IP access lists can be applied, which deny traffic to the upper, unused portions. Then the router does not attempt to forward packets to the unused portions of the subnets, and so does not ARP for it. This method has proven to be very effective, but is somewhat of a blunt axe, is fairly labor intensive, and requires coordination.

Disabling/filtering ARP requests

In general, the router does not need to ARP for hosts; when a host connects, the router can learn the IP to MAC mapping from the ARP request sent by that host. Consequently it should be possible to disable and / or filter ARP requests from the router. Unfortunately, ARP is a very low level / fundamental part of the IP stack, and is often offloaded from the normal control plane. While many routers can filter layer-2 traffic, this is usually implemented as an input filter and / or has limited ability to filter output broadcast traffic. This means that the simple "just disable ARP or filter it outbound" seems like a really simple (and obvious) solution, but implementations / architectural issues make this difficult or awkward in practice.

NAT

Broadcasts can often be caused by outside wifi scanning / backscatter traffic. In order to reduce the impact of broadcasts, NAT can be used on the entire (or a large portion) of a network. This would eliminate NAT translation entries for unused addresses, and the router would never ARP for them. There are, however, many reasons to avoid using NAT in such a blanket fashion.

Stateful firewalls

Another obvious solution would be to put a stateful firewall between the wireless network and the Internet. This firewall would block incoming traffic not associated with an outbound request. But this conflicts with the need and desire of some organizations to have the network as open as possible and to honor the end-to-end principle. An attendee on a meeting network should be an Internet host, and should be able to receive unsolicited requests. Unfortunately, keeping the network working and stable is the first priority and a stateful firewall may be required in order to achieve this.

5.2. Mitigating Spurious Service Discovery Messages

In networks that must support hundreds of STAs, operators have observed network degradation due to many devices simultaneously registering with mDNS. In a network with many clients, it is recommended to ensure that mDNS packets designed to discover services in smaller home networks be constrained to avoid disrupting other traffic.

6. Multicast Considerations for Other Wireless Media

Many of the causes of performance degradation described in earlier sections are also observable for wireless media other than 802.11.

For instance, problems with power save, excess media occupancy, and poor reliability will also affect 802.15.3 and 802.15.4. Unfortunately, 802.15 media specifications do not yet include mechanisms similar to those developed for 802.11. In fact, the design philosophy for 802.15 is oriented towards minimality, with the result that many such functions are relegated to operation within higher layer protocols. This leads to a patchwork of non-interoperable and vendor-specific solutions. See [uli] for some additional discussion, and a proposal for a task group to resolve similar issues, in which the multicast problems might be considered for mitigation.

Similar considerations hold for most other wireless media. A brief introduction is provided in [RFC5757] for the following:

- o 802.16 WIMAX
- o 3GPP/3GPP2
- o DVB-H / DVB-IPDC
- o TV Broadcast and Satellite Networks

7. Recommendations

This section provides some recommendations about the usage and combinations of some of the multicast enhancements described in Section 4 and Section 5.

Future protocol documents utilizing multicast signaling should be carefully scrutinized if the protocol is likely to be used over wireless media.

Proxy methods should be encouraged to conserve network bandwidth and power utilization by low-power devices. The device can use a unicast message to its proxy, and then the proxy can take care of any needed multicast operations.

Multicast signaling for wireless devices should be done in a way compatible with low duty-cycle operation.

8. On-going Discussion Items

This section suggests two discussion items for further resolution.

First, standards (and private) organizations should develop guidelines to help clarify when multicast packets should be sent wired rather than wireless. For example, 802.1ak [1] works on both ethernet and Wi-Fi and organizations could help decision making by developing guidelines for multicast over Wi-Fi including options for when traffic should be sent wired.

Second, reliable registration to Layer-2 multicast groups, and a reliable multicast operation at Layer-2, might provide a good multicast over wifi solution. There shouldn't be a need to support 2^24 groups to get solicited node multicast working: it is possible to simply select a number of trailing bits that make sense for a given network size to limit the number of unwanted deliveries to reasonable levels. IEEE 802.1, 802.11, and 802.15 should be encouraged to revisit L2 multicast issues and provide workable solutions.

9. Security Considerations

This document does not introduce or modify any security mechanisms. Multicast deployed on wired or wireless networks as discussed in this document can be made more secure in a variety of ways. [RFC7761], for instance, specifies the use of IPsec to ensure authentication of the link-local messages in the Protocol Independent Multicast - Sparse Mode (PIM-SM) routing protocol. [RFC5796]specifies mechanisms to authenticate the PIM-SM link-local messages using the IP security (IPsec) Encapsulating Security Payload (ESP) or (optionally) the Authentication Header (AH).

As noted in [group_key], the unreliable nature of multicast transmission over wireless media can cause subtle problems with multicast group key management and updates. When WPA (TKIP) or WPA2 (AES-CCMP) encryption is in use, AP to client (From DS) multicasts have to be encrypted with a separate encryption key that is known to all of the clients (this is called the Group Key). Quoting further from that website, "... most clients are able to get connected and surf the web, check email, etc. even when From DS multicasts are broken. So a lot of people don't realize they have multicast problems on their network..."

This document encourages the use of proxy methods to conserve network bandwidth and power utilization by low-power devices. One such proxy method listed is an Arp Sponge which listens for ARP requests, and, if it sees an ARP for an IP address that it believes is not used, it will reply with its own MAC address. ARP poisoning and false advertising could potentially undermine (e.g. DoS) this, and other, proxy approaches.

10. IANA Considerations

This document does not request any IANA actions.

11. Acknowledgements

This document has benefitted from discussions with the following people, in alphabetical order: Mikael Abrahamsson, Bill Atwood, Stuart Cheshire, Donald Eastlake, Toerless Eckert, Jake Holland, Joel Jaeggli, Jan Komissar, David Lamparter, Morten Pedersen, Pascal Thubert, Jeffrey (Zhaohui) Zhang

12. References

12.1. Informative References

[arpsponge]
Wessel, M. and N. Sijm, "Effects of IPv4 and IPv6 address resolution on AMS-IX and the ARP Sponge", July 2009, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.182.4692>.

[bridge-mc-2-uc]
Fietkau, F., "bridge: multicast to unicast", Jan 2017, <https://github.com/torvalds/linux/commit/6db6f0eae6052b70885562e1733896647ec1d807>.

[CAB] Fietkau, F., "Limit multicast buffer hardware queue depth", 2013, <https://patchwork.kernel.org/patch/2687951/>.

[Deri-2010]
Deri, L. and J. Gasparakis, "10 Gbit Hardware Packet Filtering Using Commodity Network Adapters", RIPE 61, 2010, <http://ripe61.ripe.net/presentations/138-Deri_RIPE_61.pdf>.

[dot11] "IEEE 802 Wireless", "802.11-2016 - IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification (includes 802.11v amendment)", March 2016, <http://standards.ieee.org/findstds/standard/802.11-2016.html>.

[dot11-proxyarp]
Hiertz, G., Mestanov, F., and B. Hart, "Proxy ARP in 802.11ax", September 2015, <https://mentor.ieee.org/802.11/dcn/15/11-15-1015-01-00ax-proxy-arp-in-802-11ax.pptx>.

[dot11aa] "IEEE 802 Wireless", "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming", March 2012, <https://standards.ieee.org/standard/802_11aa-2012.html>.

[group_key]
Spiff, "Why do some WiFi routers block multicast packets going from wired to wireless?", Jan 2017, <https://superuser.com/questions/730288/why-do-some-wifi-routers-block-multicast-packets-going-from-wired-to-wireless>.

[I-D.ietf-6lo-backbone-router]
Thubert, P., Perkins, C., and E. Levy-Abegnoli, "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-20 (work in progress), March 2020.

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-30 (work in progress), November 2020.

[I-D.ietf-mboned-driad-amt-discovery]
Holland, J., "DNS Reverse IP AMT (Automatic Multicast Tunneling) Discovery", draft-ietf-mboned-driad-amt-discovery-13 (work in progress), December 2019.

[ietf_802-11]
Stanley, D., "IEEE 802.11 multicast capabilities", Nov 2015, <https://mentor.ieee.org/802.11/dcn/15/11-15-1261-03-0arc-multicast-performance-optimization-features-overview-for-ietf-nov-2015.ppt>.

[mc-ack-mux]
Tanaka, Y., Sakai, E., Morioka, Y., Mori, M., Hiertz, G., and S. Coffey, "Multiplexing of Acknowledgements for Multicast Transmission", July 2015, <https://mentor.ieee.org/802.11/dcn/15/11-15-0800-00-00ax-multiplexing-of-acknowledgements-for-multicast-transmission.pptx>.

[mc-prob-stmt]
Abrahamsson, M. and A. Stephens, "Multicast on 802.11", March 2015, <https://www.iab.org/wp-content/IAB-uploads/2013/01/multicast-problem-statement.pptx>.

[mc-props]
Stephens, A., "IEEE 802.11 multicast properties", March 2015, <https://mentor.ieee.org/802.11/dcn/15/11-15-1161-02-0arc-802-11-multicast-properties.ppt>.

[Oliva2013]
de la Oliva, A., Serrano, P., Salvador, P., and A. Banchs, "Performance evaluation of the IEEE 802.11aa multicast mechanisms for video streaming", 2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM) pp. 1-9, June 2013.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC4286] Haberman, B. and J. Martin, "Multicast Router Discovery", RFC 4286, DOI 10.17487/RFC4286, December 2005, <https://www.rfc-editor.org/info/rfc4286>.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <https://www.rfc-editor.org/info/rfc4541>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <https://www.rfc-editor.org/info/rfc5424>.

[RFC5757] Schmidt, T., Waehlisch, M., and G. Fairhurst, "Multicast Mobility in Mobile IP Version 6 (MIPv6): Problem Statement and Brief Survey", RFC 5757, DOI 10.17487/RFC5757, February 2010, <https://www.rfc-editor.org/info/rfc5757>.

[RFC5796] Atwood, W., Islam, S., and M. Siami, "Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages", RFC 5796, DOI 10.17487/RFC5796, March 2010, <https://www.rfc-editor.org/info/rfc5796>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[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>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC6970] Boucadair, M., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF)", RFC 6970, DOI 10.17487/RFC6970, July 2013, <https://www.rfc-editor.org/info/rfc6970>.

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

[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>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

[Tramarin2017]
Tramarin, F., Vitturi, S., and M. Luvisotto, "IEEE 802.11n for Distributed Measurement Systems", 2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) pp. 1-6, May 2017.

[uli] Kinney, P., "LLC Proposal for 802.15.4", Nov 2015, <https://mentor.ieee.org/802.15/dcn/15/15-15-0521-01-wng0-llc-proposal-for-802-15-4.pptx>.

12.2. URIs

[1] https://www.ieee802.org/1/pages/802.1ak.html

Authors' Addresses

Charles E. Perkins
Blue Meadow Networks

Phone: +1-408-330-4586
Email: charliep@computer.org

Mike McBride
Futurewei Technologies Inc.
2330 Central Expressway
Santa Clara, CA 95055
USA

Email: michael.mcbride@futurewei.com

Dorothy Stanley
Hewlett Packard Enterprise
2000 North Naperville Rd.
Naperville, IL 60566
USA

Phone: +1 630 979 1572
Email: dstanley1389@gmail.com

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
USA

Email: warren@kumari.net

Juan Carlos Zuniga
SIGFOX
425 rue Jean Rostand
Labege 31670
France

Email: j.c.zuniga@ieee.org

Mboned J. Holland
Internet-Draft Akamai Technologies, Inc.
Intended status: Standards Track February 01, 2021
Expires: August 5, 2021

Multicast Network Address Translation
draft-ietf-mboned-mnat-00

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.

Table of Contents

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]

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).

- o Join: https://www.ietf.org/mailman/listinfo/mboned

- o Search: https://mailarchive.ietf.org/arch/browse/mboned/

1.4.2. Implementation status

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

- o 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 folowing entities:

- o MNAT service

- o ingress nodes

- o 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-existant) 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 withing 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
```

MNAT Tree Diagram

3.2. Yang Module

```
   <CODE BEGINS> file ietf-mnat@2021-02-01.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:

- o communcation with the MNAT service should be secured. RESTCONF does this, alternate methods should also do it.

- o 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.

- o 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

MBONED H. Song
Internet-Draft M. McBride
Intended status: Standards Track Futurewei Technologies
Expires: August 26, 2021 G. Mirsky
ZTE Corp.
G. Mishra
Verizon Inc.
H. Asaeda
NICT
T. Zhou
Huawei
February 22, 2021

Multicast On-path Telemetry Solutions
draft-ietf-mboned-multicast-telemetry-00

Abstract

This document discusses the requirement of on-path telemetry for multicast traffic. The existing solutions are examined and their issues are identified. Solution modifications are proposed to allow the original multicast tree to be correctly reconstructed without unnecessary replication of telemetry information.

Requirements Language

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.

This Internet-Draft will expire on August 26, 2021.

Table of Contents

1. Introduction

Multicast traffic is an important traffic type in today's Internet. Multicast provides services that are often real time (e.g., online meeting) or have strict QoS requirements (e.g., IPTV, Market Data). Multicast packet drop and delay can severely affect the application performance and user experience.

It is important to monitor the performance of the multicast traffic. Existing OAM techniques cannot gain direct and accurate information about the multicast traffic. New on-path telemetry techniques such as In-situ OAM [I-D.ietf-ippm-ioam-data], Postcard-based Telemetry [I-D.song-ippm-postcard-based-telemetry], and Hybrid Two-Step (HTS) [I-D.mirsky-ippm-hybrid-two-step] provide promising means to directly monitor the network experience of multicast traffic. However, multicast traffic has some unique characteristics which pose some challenges on efficiently applying such techniques.

When a network contains multicast (p2mp) trees there will be redundant data as data is replicated at branch points. The IP Multicast S,G data is identical from one branch to another on it’s way to multiple receivers. When adding iOAM trace data, to multicast packets, we enlarge data packets thus consuming more network bandwidth. Instead of adding iOAM trace data, it could be more efficient to collect the telemetry information using solutions, such as iOAM postcard or HTS, to cut down on the redundant iOAM data. The problem is that a postcard type solution doesn’t have a branch identifier.

This draft proposes a set of solutions to this iOAM data redundancy problem. The requirements for multicast traffic telemetry are discussed along with the issues of the existing on-path telemetry techniques. We propose modifications to make these techniques adapt to multicast in order for the original multicast tree to be correctly reconstructed while eliminating redundant data.

2. Requirements for Multicast Traffic Telemetry

Multicast traffic is forwarded through a multicast tree. With PIM and P2MP (MLDP, RSVP-TE) the forwarding tree is established and maintained by the multicast routing protocol. With BIER, no state is created in the network to establish a forwarding tree, instead, a bier header provides the necessary information for each packet to know the egress points. Multicast packets are only replicated at each tree branch node for efficiency.

There are several requirements for multicast traffic telemetry, a few of which are:

- o Reconstruct and visualize the multicast tree through data plane monitoring.
- o Gather the multicast packet delay and jitter performance.
- o Find the multicast packet drop location and reason.
- o Gather the VPN state and tunnel information in case of P2MP multicast.

In order to meet these requirements, we need the ability to directly monitor the multicast traffic and derive data from the multicast packets. The conventional OAM mechanisms, such as multicast ping and trace, may not be sufficient to meet these requirements.

3. Issues of Existing Techniques

On-path Telemetry techniques that directly retrieve data from multicast traffic's live network experience are ideal to address the above mentioned requirements. The representative techniques include In-situ OAM (IOAM) Trace option [I-D.ietf-ippm-ioam-data], IOAM Direct Export (DEX) option [I-D.ioamteam-ippm-ioam-direct-export], and Postcard-based Telemetry with Packet Marking(PBT-M) [I-D.song-ippm-postcard-based-telemetry]. However, unlike unicast, multicast poses some unique challenges to applying these techniques.

Multicast packets are replicated at each branch node in the corresponding multicast tree. Therefore, there are multiple copies of packets in the network.

If the IOAM trace option is used for on-path data collection, the partial trace data will also be replicated into multiple copies. The end result is that each copy of the multicast packet has a complete trace. Most of the data, however, is redundant. Data redundancy introduces unnecessary header overhead, wastes network bandwidth, and complicates the data processing. In case the multicast tree is large, and the path is long, the redundancy problem becomes severe.

The PBT solutions, including the IOAM DEX option and PBT-M, can be used to eliminate such data redundancy, because each node on the tree only sends a postcard covering local data. However, they cannot track the tree branches properly so it can bring confusion about the multicast tree topology. For example, Node A has two branches, one to Node B and the other to node D, and Node B leads to Node C and Node D leads to Node E. From the received postcards, one cannot tell whether or not Node C(E) is the next hop of Node B(D).

The fundamental reason for this problem is that there is not an identifier (either implicit or explicit) to correlate the data on each branch.

4. Proposed Modifications to Existing Techniques

Two solutions are proposed to address the above issues. One is built on PBT and requires augmentation or modification to the instruction header of the IOAM Direct Export Option; the other combines the IOAM trace option and PBT for an optimized solution.

4.1. Per-hop postcard using IOAM DEX

One way to mitigate PBT's multiple tree tracking weakness is to augment it with a branch identifier field. Note that this works for the IOAM DEX option but not for PBT-M because the IOAM DEX option uses an instruction header. To make the branch identifier globally unique, the branch node ID plus an index is used. For example, if Node A has two branches, one to Node B and one to Node C, Node A will use [A, 0] as the branch identifier for the branch to B, and [A, 1] for the branch to C. The identifier is unchanged for each multicast tree instance and carried with the multicast packet until the next branch node. Each postcard needs to include the branch identifier in the export data. The branch identifier, along with the other fields such as flow ID and sequence number, is sufficient for the data analyzer to reconstruct the topology of the multicast tree.

Figure 1 shows an example of this solution. "P" stands for the postcard packet. The square brackets contains the branch identifier. The curly brace contains the telemetry data about a specific node.

```
         P:[A,0]{A}  P:[A,0]{B}  P:[B,1]{D} P:[B,0]{C}
              ^            ^          ^        ^
              :            :          :        :
              :            :          :        :
              :            :          :      +-:-+
              :            :          :      |   |
              :            :      +---:----->| C |--...
           +-:-+        +-:-+     |   :      |   |
           |   |        |   |-----+   :      +---+
           | A |------->| B |         :
           |   |        |   |--+    +-:-+
           +---+        +---+  |    |   |
                               +--->| D |--....
                                    |   |
                                    +---+
```

Figure 1: Per-hop Postcard

Each branch fork node need to generate the branch ID for each branch in its multicast tree instance and include it in the IOAM DEX option header so the downstream node can learn it. The branch ID contains two parts: the branch fork node ID and a unique branch index.

Figure 2 shows that the branch ID is carried as an optional field after the flow ID and sequence number optional fields in the IOAM DEX option header. A bit "M" in the Flags field is reserved to indicate the presence of the branch index field. The "M" flag position will be determined later after the other flags are specified in [I-D.ioamteam-ippm-ioam-direct-export].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Namespace-ID           |M|          Flags              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IOAM-Trace-Type                 |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Flow ID (optional)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sequence Number  (Optional)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Encoded Branch ID (optional)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Carry Branch Index in IOAM DEX option header

To avoid introducing a new type of data field to the IOAM DEX option header, we can encode the branch identifier using the existing node ID data field as defined in [I-D.ietf-ippm-ioam-data]. Currently, the node ID field occupies three octets. A simple solution is to shorten the node ID field so a number of bits can be saved to encode the branch index, as shown in Figure 3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Hop_Lim      |              node_id           | branch index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Encode Branch Index with Node ID Method 1

Another encoding method is to use the sum of the node ID and the branch index as the new node ID, as shown in Figure 4. As long as the node IDs are assigned with large enough gap, the telemetry data analyzer can still successfully recover the original node ID and branch index.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Hop_Lim     |              node_id + branch index           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Encode Branch Index with Node ID Method 2

Once a node gets the branch ID information from the upstream, it MUST carry this information in its telemetry data export postcards, so the original multicast tree can be correctly reconstructed based on the postcards.

4.2. Per-section postcard

The second solution is a combination of the IOAM trace mode and PBT. To avoid data redundancy at each branch node, the trace data accumulated, to that point, is exported by a postcard before the packet is replicated. In this case, each branch still needs to maintain some identifier to help correlate the postcards for each tree section. The natural way to accomplish this is to simply carry the branch node's data (including its ID) in the trace of each branch. This is also necessary because each replicated multicast packet can have different telemetry data pertaining to this particular copy (e.g., node delay, egress timestamp, and egress interface). As a consequence, the local data exported by each branch node can only contain partial data (e.g., ingress interface and ingress timestamp).

Figure 5 shows an example in a segment of a multicast tree. Node B and D are two branch nodes and they will export a postcard covering the trace data for the previous section. The end node of each path will also need to export the data of the last section as a postcard.

```
                P:{A,B'}            P:{B1,C,D'}
                   ^                     ^
                   :                     :
                   :                     :
                   :                     :    {D1}
                   :                     :    +--...
                   :        +---+      +---+  |
                   :   {B1} |   |{B1,C}|   |--+ {D2}
                   :    +-->| C |----->| D |-----...
       +---+     +---+  |   |   |      |   |--+
       |   | {A} |   |--+   +---+      +---+  |
       | A |---->| B |                        +--...
       |   |     |   |--+   +---+             {D3}
       +---+     +---+  |   |   |{B2,E}
                        +-->| E |--...
                       {B2} |   |
                            +---+
```

Figure 5: Per-section Postcard

There is no need to modify the IOAM trace mode header format. We just need to configure the branch node to export the postcard and refresh the IOAM header and data.

5. Considerations for Different Multicast Protocols

MTRACEv2 [RFC8487] provides an active probing approach for the tracing of an IP multicast routing path. Mtrace can also provide information such as the packet rates and losses, as well as other diagnostic information. New on-path telemetry techniques will enhance Mtrace, and other existing OAM solutions, with more granular and realtime network status data through direct measurements. There are various multicast protocols that are used to forward the multicast data. Each will require their own unique on-path telemetry solution.

5.1. Application in PIM

PIM-SM [RFC7761] is the most widely used multicast routing protocol deployed today. Of the various PIM modes (PIM-SM, PIM-DM, BIDIR-PIM, PIM-SSM), PIM-SSM is the preferred method due to its simplicity and removal of network source discovery complexity. With all PIM modes, control plane state is established in the network in order to forward multicast UDP data packets. But with PIM-SSM, the discovery of multicast sources is performed outside of the network via HTTP, SDN, etc. IP Multicast packets fall within the range of 224.0.0.0 through

239.255.255.255. The telemetry solution will need to work within this address range and provide telemetry data for this UDP traffic.

The proposed solutions for encapsulating the telemetry instruction header and metadata in IPv4/IPv6 UDP packets are described in [I-D.herbert-ipv4-udpencap-eh] and [I-D.ioametal-ippm-6man-ioam-ipv6-deployment].

5.2. Application in P2MP

Multicast Label Distribution Protocol (MLDP) and P2MP RSVP-TE are commonly used within a Multicast VPN (MVPN) environment. MLDP provides extensions to LDP to establish point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) in MPLS networks. P2MP RSVP-TE provides extensions to RSVP-TE for establish traffic-engineered P2MP LSPs in MPLS networks. The telemetry solution will need to be able to follow these P2MP paths. The telemetry instruction header and data should be encapsulated into MPLS packets on P2MP paths. A corresponding proposal is described in [I-D.song-mpls-extension-header].

5.3. Application in BIER

BIER [RFC8279] adds a new header to multicast packets and allows the multicast packets to be forwarded according to the header only. By eliminating the requirement of maintaining per multicast group state, BIER is more scalable than the traditional multicast solutions.

OAM Requirements for BIER [I-D.ietf-bier-oam-requirements] lists many of the requirements for OAM at the BIER layer which will help in the forming of on-path telemetry requirements as well.

There is also current work to provide solutions for BIER forwarding in ipv6 networks. For instance, a solution, BIER in Non-MPLS IPv6 Networks [I-D.xie-bier-ipv6-encapsulation], proposes a new bier Option Type codepoint from the "Destination Options and Hop-by-Hop Options" IPv6 sub-registry. This is similar to what IOAM proposes for IPv6 transport.

Depending on how the BIER header is encapsulated into packets with different transport protocols, the method to encapsulate the telemetry instruction header and metadata also varies. It is also possible to make the instruction header and metadata a part of the BIER header itself, such as in a TLV.

6. Security Considerations

No new security issues are identified other than those discovered by the IOAM and PBT drafts.

7. IANA Considerations

The document makes no request of IANA.

8. Contributors

TBD

9. Acknowledgments

The authors would like to thank Frank Brockners for the comments and advice.

10. References

10.1. 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>.

[RFC4687] Yasukawa, S., Farrel, A., King, D., and T. Nadeau, "Operations and Management (OAM) Requirements for Point-to-Multipoint MPLS Networks", RFC 4687, DOI 10.17487/RFC4687, September 2006, <https://www.rfc-editor.org/info/rfc4687>.

[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>.

[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>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

[RFC8487] Asaeda, H., Meyer, K., and W. Lee. Ed., "Mtrace Version 2: Traceroute Facility for IP Multicast", RFC 8487, DOI 10.17487/RFC8487, October 2018, <https://www.rfc-editor.org/info/rfc8487>.

10.2. Informative References

[I-D.herbert-ipv4-udpencap-eh]
Herbert, T., "IPv4 Extension Headers and UDP Encapsulated Extension Headers", draft-herbert-ipv4-udpencap-eh-01 (work in progress), March 2019.

[I-D.ietf-bier-oam-requirements]
Mirsky, G., Nainar, N., Chen, M., and S. Pallagatti, "Operations, Administration and Maintenance (OAM) Requirements for Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-oam-requirements-11 (work in progress), November 2020.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-11 (work in progress), November 2020.

[I-D.ioametal-ippm-6man-ioam-ipv6-deployment]
Bhandari, S., Brockners, F., Mizrahi, T., Kfir, A., Gafni, B., Spiegel, M., Krishnan, S., and M. Smith, "Deployment Considerations for In-situ OAM with IPv6 Options", draft-ioametal-ippm-6man-ioam-ipv6-deployment-03 (work in progress), March 2020.

[I-D.ioamteam-ippm-ioam-direct-export]
Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F., Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ OAM Direct Exporting", draft-ioamteam-ippm-ioam-direct-export-00 (work in progress), October 2019.

[I-D.mirsky-ippm-hybrid-two-step]
Mirsky, G., Lingqiang, W., Zhui, G., and H. Song, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-07 (work in progress), December 2020.

[I-D.song-ippm-postcard-based-telemetry]
Song, H., Zhou, T., Li, Z., Mirsky, G., Shin, J., and K. Lee, "Postcard-based On-Path Flow Data Telemetry using Packet Marking", draft-song-ippm-postcard-based-telemetry-08 (work in progress), October 2020.

[I-D.song-mpls-extension-header]
Song, H., Li, Z., Zhou, T., and L. Andersson, "MPLS Extension Header", draft-song-mpls-extension-header-02 (work in progress), February 2019.

[I-D.xie-bier-ipv6-encapsulation]
Xie, J., Geng, L., McBride, M., Asati, R., Dhanaraj, S., Zhu, Y., Qin, Z., Shin, M., Mishra, G., and X. Geng, "Encapsulation for BIER in Non-MPLS IPv6 Networks", draft-xie-bier-ipv6-encapsulation-09 (work in progress), January 2021.

Authors' Addresses

Haoyu Song
Futurewei Technologies
2330 Central Expressway
Santa Clara
USA

Email: hsong@futurewei.com

Mike McBride
Futurewei Technologies
2330 Central Expressway
Santa Clara
USA

Email: mmcbride@futurewei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Hitoshi Asaeda
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi
Koganei, Tokyo 184-8795
Japan

Email: asaeda@nict.go.jp

Tianran Zhou
Huawei

Email: zhoutianran@huawei.com