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

J. Holland
Akamai Technologies, Inc.

7 March 2022

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 8 September 2022.

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]

Table 1

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

1.3. Motivation

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

1. Global IPv6 (S,G)s subscribed from within an IPv4-only network, or global IPv4 (S,G)s subscribed from within an IPv6-only network.

2. Networks with legacy devices that support only IGMPv2 or MLDv1, or otherwise do not support SSM and cannot discover the external sources without the use of non-standard services since interdomain any-source multicast has been deprecated (see [RFC8815]).

3. Networks that ingest external multicast traffic in a way that the route to the source of the traffic does not go through the ingest point may need to use a different source so that the Reverse Path Forwarding (RPF) can find the correct network location for the ingest.

4. Networks that provision multicast transport and packet replication channels with static routing instead of dynamic tree-building protocols like PIM-SM [RFC7761].

5. Networks using VLAN [IEEE-802.1Q] for traffic segregation and has Layer 2 access devices that assign VLAN tags according to MAC addresses will get MAC address collisions among multicast groups. Because the bits used for the multicast addresses come from the bottom 23 bits of the destination group address as described in [RFC1112] and those bits can collide between groups, especially in SSM. The technological limitations of VLAN assignment using MAC addresses at Layer 2 breaks the traffic segregation of multicast traffic for different services in such devices.

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

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

1.4. Notes for Contributors and Reviewers

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

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

1.4.1. Venues for Contribution and Discussion

This document is in the Github repository at:

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

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

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

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

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

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

1.4.2. Implementation status

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

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

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

2. Protocol Operation

2.1. Overview

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

* MNAT service

* ingress nodes

* egress nodes

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

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

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

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

2.1.1. Egress Node Operational Modes

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

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

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

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

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

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

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

2.2. Service Discovery

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

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

2.2.1. Detecting Invalid Services

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

2.3. RESTCONF Bootstrap

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

2.4. Message Handling

2.4.1. Notification Subscription

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

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

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

2.4.2. Watcher Keys

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

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

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

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

TBD: better explanation about how the service times out egress nodes that don’t refresh their egress key on schedule, and how egress nodes that reconnect can attempt to refresh the prior key they were using, but must request a new one on error. Probably define a state per egress key (e.g. active vs. recently expired vs. non-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
```

Figure 1: MNAT Tree Diagram

3.2. Yang Module

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

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

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

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

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

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

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

     description
       "Multicast Network Address Translation Model.

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

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

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

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

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

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

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

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

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

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

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

    identity unassigned {
      base assignment-state;
```

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

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

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

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

```
      list watcher {
        key "id";
        description
          "Mappings of traffic that correspond to the registered
           interest list for a given watch id (from the
           get-new-watcher-id rpc)";
        leaf id {
          type watcher-key;
          description
            "Identifier from get-new-watcher-id.  Tracks assignments
             of interest to the specific watcher.";
        }
        list monitor {
          key "id";
          leaf id {
            type string;
            description
              "id of the monitor definition";
          }
          uses monitor-definition;
        }
      }
    }
    container assigned-channels {
      config false;
      description
        "MNAT mappings of global (S,G)s into a local transport.";

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

          leaf id {
            type assignment-id;
```

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

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

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

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

4. IANA Considerations

4.1. The YANG Module Names Registry

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

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

4.2. The XML Registry

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

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

4.3. The Service Name and Transport Protocol Port Number Registry

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

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

5. Security Considerations

TBD. (What, me worry?)

Notable points to cover:

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

* separate authentication of the contents of the multicast traffic is recommended (e.g. with AMBI or TESLA). Probably it's not recommended for a network with MNAT to pass external traffic that does not provide authentication, and if the internal traffic is not authenticated, to segregate the internal from the external traffic in the MNAT assignment pools.

* mistaken mappings can result in receipt of payloads for the wrong channel. This can happen transiently even during normal operation. Recommend some steps to mitigate and avoid (e.g. the grace period and the authentication-TBD: explain how they help)

* Clients can (deliberately or accidentally) overload the service. Limits should be set to avoid disrupting traffic to the rest of the network.

6. Acknowledgements

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

7. References

7.1. Normative References

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <https://www.rfc-editor.org/info/rfc1112>.

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

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, DOI 10.17487/RFC3376, October 2002, <https://www.rfc-editor.org/info/rfc3376>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, DOI 10.17487/RFC4604, August 2006, <https://www.rfc-editor.org/info/rfc4604>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

7.2. Informative References

[IEEE-802.1Q] IEEE, "Local and Metropolitan Area Networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q, n.d., <https://standards.ieee.org/findstds/standard/802.1Q-2011.html>.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <https://www.rfc-editor.org/info/rfc2663>.

[RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, DOI 10.17487/RFC3397, November 2002, <https://www.rfc-editor.org/info/rfc3397>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

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

[RFC6202] Loreto, S., Saint-Andre, P., Salsano, S., and G. Wilkins, "Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP", RFC 6202, DOI 10.17487/RFC6202, April 2011, <https://www.rfc-editor.org/info/rfc6202>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

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

[RFC8815] Abrahamsson, M., Chown, T., Giuliano, L., and T. Eckert, "Deprecating Any-Source Multicast (ASM) for Interdomain Multicast", BCP 229, RFC 8815, DOI 10.17487/RFC8815, August 2020, <https://www.rfc-editor.org/info/rfc8815>.

Author's Address

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