

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: January 09, 2014

G. Chen  
China Mobile  
D. Binet  
France Telecom-Orange  
July 08, 2013

Radius Attributes for Stateful NAT64  
draft-chen-behave-nat64-radius-extension-00

Abstract

This document proposes new radius attributes for stateful NAT64. The extensions are used to provide geo-location services with an exact IPv6 source address. The message flow to deliver the NAT64 binding information between radius clients and servers is also described. Therefore, accurate location could be traced out depending on the radius method.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 09, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Requirements Language . . . . .	3
3. Problem Statement . . . . .	3
4. Delivery Methods for NAT64 Binding Information . . . . .	3
5. Attributes . . . . .	4
5.1. NAT64-Binding-Capable . . . . .	4
5.2. Requested-Binding-Info . . . . .	5
5.3. NAT64-Binding-Information . . . . .	6
6. Diameter Considerations . . . . .	8
7. IANA Considerations . . . . .	8
8. Security Considerations . . . . .	8
9. References . . . . .	8
9.1. Normative References . . . . .	8
9.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

Stateful NAT64[RFC6146] has been specified to provide IPv4 services access when IPv6-only connectivity is enabled. This NAT64 function could be implemented into routers or firewalls and deployed in broadband access networks or mobile core networks. A public IPv4 pool is configured in a NAT64 device to represent IPv6 subscribers in the IPv4 realm. Since the public IPv4 address is shared by several subscribers, it's hardly for a geo-location service to retrieve accurate location information just depending on mapped IPv4 source address. Therefore, it may impact geo-location service provisioning in such case due to unsatisfactory inputs.

[RFC6269] mentions that in order to resolve the location of a host based on IP address, "It will be necessary for users of such systems to provide more information (e.g., TCP or UDP port numbers), and for the systems to use this information to query additional network resources (e.g., Network Address Translation - Protocol Translation (NAT-PT) binding tables)." Current geo-location systems may rely on a radius database to inspect location information, for example the information provided in [RFC5580]. A radius based method may be desirable to convey original IPv6 source address in such system because it's rather convenient for a geo-location to get actual IPv6 source address through the same message bus. This document proposes to provide those information using radius methods.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 3. Problem Statement

IP address sharing solution raises some issues dealing with the capability to reveal the actual source address. A use of stateful NAT64 likely has same issues once geo-location systems seek an accurate input of a source address. Unlike NAT44, it may make more significances to trace the IPv6 source address other than the mapped IPv4 address to locate the subscribers.[I-D.ietf-v6ops-nat64-experience] provided more descriptions on stateful NAT64 uses. Once the stateful NAT64 function is built at a load balancer, XFF (X-Forwarded-For) [I-D.ietf-appsawg-http-forwarded] is likely to be adopted to transmit the IPv6 source address in HTTP headers. Those messages would be passed on to web-servers for the geo-location processing. However, XFF only handles HTTP-based traffic and may not be implemented, for example if the NAT64 function is integrated within routers or firewall in an broadband fixed network or a mobile network. Requiring NAT64 devices providing some application aware functions to insert IPv6 source addresses for each data flow would introduce overwhelming complexity and performance degradation. It's also possible to extend Port Control Protocol (PCP) to support those network information queries from external servers. This method can be treated as an "out-band" approach. However, it may require additional correlations between different systems. Therefore, the document proposes a radius-based solution to fit into geo-location systems with following benefits.

- o It has few impacts to the NAT64 performance since the radius is a independent system which doesn't interact with NAT64 process
- o Geo-location systems already rely on the radius database. The extended attributes could be transmitted in the same message as already occurs over RADIUS[RFC5580]

## 4. Delivery Methods for NAT64 Binding Information

The Figure 1 takes an example to show the message exchanges when the NAT64 function is implemented in a Broadband Network Gateway(BNG).

If the RADIUS client provides a NAT64-Binding-Capable Attribute in the Access-Request, then the RADIUS server MAY request NAT64 Binding information from the RADIUS client. The inclusion of the Location-

Capable Attribute in an Access-Request message indicates that the BNG with stateful NAT64 functions is capable of providing binding information in response to an Access-Challenge. The subsequent Access-Challenge message sent from the RADIUS server to the BNG provides a hint regarding the desired NAT64 Binding Attributes. BNG would search the corresponding binding information regarding to mapped IPv4 address contained in the Requested-Binding-Info attribute. In the shown message flow, the NAT64-Binding-Information attributes including IPv6 source address and life-time of the binding are then provided in the subsequent Access-Request message. Afterwards, RADIUS server should take a authorization procedure to evaluate this Access-Request message.

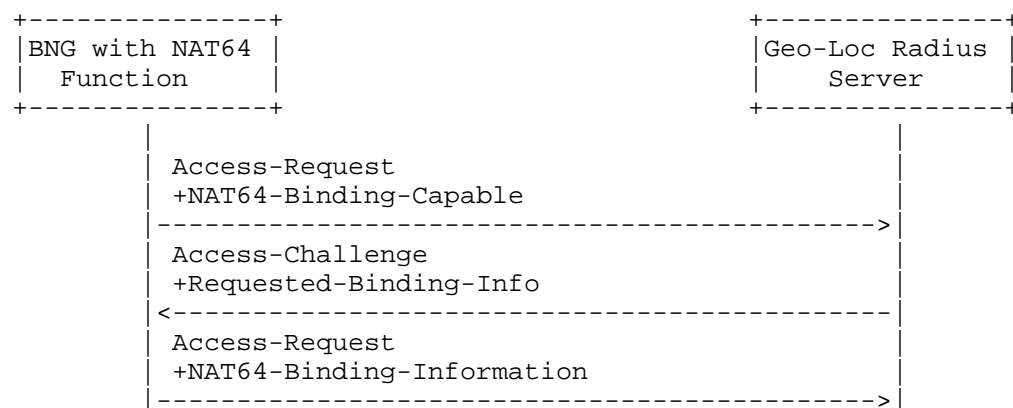
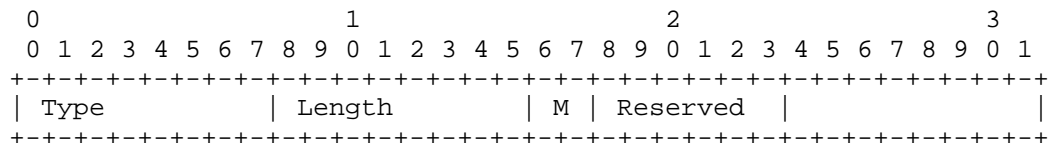


Figure 1: NAT64 Binding Information Delivery

## 5. Attributes

### 5.1. NAT64-Binding-Capable

The NAT64-Binding-Capable Attribute allows a network node with stateful NAT64 functions to indicate support for the functionality specified in this document. The NAT64-Binding-Capable Attribute MUST be sent with the Access-Request messages. A RADIUS server MAY challenge for additional network information once the NAT64-Binding-Capable Attribute has been received.



Type (8 bits)

TBD - NAT64-Binding-Capable

Length (8 bits)

4

M flag (2 bits)

This flag indicates the type of address mapping.

00 -- Endpoint-Independent Mapping

01 -- Address-Dependent Mapping

10 -- Address and Port-Dependent Mapping

Reserved (6 bits)

The bits are reserved for the future uses

## 5.2. Requested-Binding-Info

The Requested-Binding-Info Attribute MUST be sent with the Access-Challenge depending on the received NAT64-Binding-Capable attributes. A stateful NAT64 function with Radius clients SHOULD reply to the Access-Challenge with Access-Request message.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Type										Length										Protocol										Reserved									
Mapped IPv4 address																																							
Mapped Port Number															Peer's IPv4 address(Optional)																								
Peer's IPv4 address(Optional)															Peer's Port Number(Optional)																								

Type (8 bits)

TBD - Requested-Binding-Info

Length (8 bits)

It indicates the length of attributes

Protocol (8 bits)

It indicates the Upper-layer protocol associated with the mapping.

Values are taken from the IANA protocol registry. For example, 17 represents UDP mappings while 6 represents TCP mapping

Reserved (8 bits)

The bits are reserved for the future uses

Mapped IPv4 address (32 bits)

It contains the IPv4 address which represents the IPv6 host in the IPv4 Internet.

Mapped Port Number (16 bits)

It indicates the assigned port number correlated with the Mapped IPv4 address.

Peer's IPv4 address (32 bits)

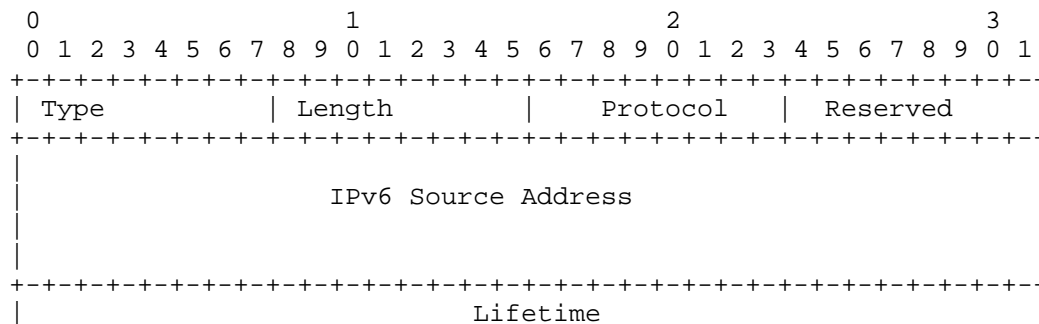
It indicates the IPv4 address that internal hosts connect with. The IPv4 address MAY be only carried when Address-Dependent Mapping and Address and Port-Dependent Mapping are indicated within the received Access-Request message.

Peer's Port Number (16 bits)

It indicates the port number correlated with the Peer's IPv4 address. The Peer's Port Number MAY be only carried when Port-Dependent Mapping are indicated within the received Access-Request message.

### 5.3. NAT64-Binding-Information

The NAT64-Binding-Information Attribute MUST be sent with the Access-Request responding to Access-Challenge from a radius client to a Radius server.



```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Mapped IPv4 address                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Mapped Port Number         | Peer's IPv4 address(Optional) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Peer's IPv4 address(Optional) | Peer's Port Number(Optional) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type (8 bits)

TBD - NAT64-Binding-Information

Length (8 bits)

It indicates the length of attributes

Protocol (8 bits)

It indicates the Upper-layer protocol associated with the mapping. Values are taken from the IANA protocol registry. For example, 17 represents UDP mappings while 6 represents TCP mapping

Reserved (8 bits)

The bits are reserved for the future uses

IPv6 Source Address (128 bits)

It indicates the IPv6 source address corresponding to mapped IPv4 address and port number

Lifetime (32 bits)

It indicates how long the geo-location should assume the IPv6 address is still correlated with the mapped IPv4 address and port.

Mapped IPv4 address (32 bits)

It contains the IPv4 address which represents the IPv6 host in the IPv4 Internet.

Mapped Port Number (16 bits)

It indicates the assigned port number correlated with the Mapped IPv4 address.

Peer's IPv4 address (32 bits)

It indicates the IPv4 address that internal hosts connect with. The IPv4 address MAY be only carried when Address-Dependent Mapping and Address and Port-Dependent Mapping are indicated within the received Access-Request message.

Peer's Port Number (16 bits)

It indicates the port number correlated with the Peer's IPv4 address. The Peer's Port Number MAY be only carried when Port-Dependent Mapping are indicated within the received Access-Request message.

## 6. Diameter Considerations

This attribute is usable within either RADIUS or Diameter[RFC6733] . Since the Attributes defined in this document will be allocated from the standard RADIUS type space, no special handling is required by Diameter entities.

## 7. IANA Considerations

This document requires the assignment of RADIUS Attribute Type in the "Radius Types" registry (currently located at <http://www.iana.org/assignments/radius-types> for the following attributes: o

- o NAT64-Binding-Capable TBD
- o Requested-Binding-Info TBD
- o NAT64-Binding-Information TBD

IANA should allocate the numbers from the standard RADIUS Attributes space using the "IETF Review" policy [RFC5226].

## 8. Security Considerations

The proposed method is RECOMMENDED to be used with [RFC5580]. Therefore, it shares all the considerations at Section 7 of [RFC5580].

## 9. References

### 9.1. Normative References

[I-D.ietf-appsawg-http-forwarded]



Petersson, A. and M. Nilsson, "Forwarded HTTP Extension", draft-ietf-appsawg-http-forwarded-10 (work in progress), October 2012.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5580] Tschofenig, H., Adrangi, F., Jones, M., Lior, A., and B. Aboba, "Carrying Location Objects in RADIUS and Diameter", RFC 5580, August 2009.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

## 9.2. Informative References

- [I-D.ietf-v6ops-nat64-experience]  
Chen, G., Cao, Z., Byrne, C., Xie, C., and D. Binet,  
"NAT64 Deployment Considerations", draft-ietf-v6ops-nat64-experience-01 (work in progress), January 2013.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

## Authors' Addresses

Gang Chen  
China Mobile  
53A,Xibianmennei Ave.,  
Xuanwu District,  
Beijing 100053  
China

Email: phdgang@gmail.com

David Binet  
France Telecom-Orange  
Rennes  
35000  
France

Email: david.binet@orange.com

Behave  
Internet-Draft  
Intended status: Standards Track  
Expires: July 13, 2017

S. Sivakumar  
R. Penno  
Cisco Systems  
January 9, 2017

IPFIX Information Elements for logging NAT Events  
draft-ietf-behave-ipfix-nat-logging-13

Abstract

Network operators require NAT devices to log events like creation and deletion of translations and information about the resources that the NAT device is managing. The logs are essential in many cases to identify an attacker or a host that was used to launch malicious attacks and for various other purposes of accounting. Since there is no standard way of logging this information, different NAT devices log the information using proprietary formats and hence it is difficult to expect a consistent behavior. The lack of a consistent way to log the data makes it difficult to write the collector applications that would receive this data and process it to present useful information. This document describes the formats for logging of NAT events.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 13, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Terminology . . . . .	3
1.2. Requirements Language . . . . .	4
2. Scope . . . . .	4
3. Deployment . . . . .	5
4. Event based logging . . . . .	5
4.1. Logging of destination information . . . . .	6
4.2. Information Elements . . . . .	6
4.3. Definition of NAT Events . . . . .	8
4.4. Quota exceeded Event types . . . . .	9
4.5. Threshold reached Event types . . . . .	10
4.6. Templates for NAT Events . . . . .	11
4.6.1. NAT44 create and delete session events . . . . .	11
4.6.2. NAT64 create and delete session events . . . . .	12
4.6.3. NAT44 BIB create and delete events . . . . .	13
4.6.4. NAT64 BIB create and delete events . . . . .	13
4.6.5. Addresses Exhausted event . . . . .	14
4.6.6. Ports Exhausted event . . . . .	14
4.6.7. Quota exceeded events . . . . .	15
4.6.7.1. Maximum session entries exceeded . . . . .	15
4.6.7.2. Maximum BIB entries exceeded . . . . .	15
4.6.7.3. Maximum entries per user exceeded . . . . .	15
4.6.7.4. Maximum active host or subscribers exceeded . . . . .	16
4.6.7.5. Maximum fragments pending reassembly exceeded . . . . .	16
4.6.8. Threshold reached events . . . . .	17
4.6.8.1. Address pool high or low threshold reached . . . . .	17
4.6.8.2. Address and port high threshold reached . . . . .	17
4.6.8.3. Per-user Address and port high threshold reached . . . . .	18
4.6.8.4. Global Address mapping high threshold reached . . . . .	18
4.6.9. Address binding create and delete events . . . . .	19
4.6.10. Port block allocation and de-allocation . . . . .	19
5. Management Considerations . . . . .	20
5.1. Ability to collect events from multiple NAT devices . . . . .	20
5.2. Ability to suppress events . . . . .	20
6. Acknowledgements . . . . .	21
7. IANA Considerations . . . . .	21
7.1. Information Elements . . . . .	21
7.1.1. natInstanceID . . . . .	21

7.1.2.	internalAddressRealm	21
7.1.3.	externalAddressRealm	22
7.1.4.	natQuotaExceededEvent	22
7.1.5.	natThresholdEvent	23
7.1.6.	natEvent	24
8.	Security Considerations	25
9.	References	25
9.1.	Normative References	25
9.2.	Informative References	26
	Authors' Addresses	27

## 1. Introduction

The IPFIX Protocol [RFC7011] defines a generic push mechanism for exporting information and events. The IPFIX Information Model [IPFIX-IANA] defines a set of standard IEs which can be carried by the IPFIX protocol. This document details the IPFIX Information Elements (IEs) that MUST be logged by a NAT device that supports NAT logging using IPFIX, and all the optional fields. The fields specified in this document are gleaned from [RFC4787] and [RFC5382].

This document and [I-D.ietf-behave-syslog-nat-logging] are written in order to standardize the events and parameters to be recorded, using IPFIX [RFC7011] and SYSLOG [RFC5424] respectively. The intent is to provide a consistent way to log information irrespective of the mechanism that is used.

This document uses IPFIX as the encoding mechanism to describe the logging of NAT events. However, the information that is logged should be the same irrespective of what kind of encoding scheme is used. IPFIX is chosen because it is an IETF standard that meets all the needs for a reliable logging mechanism. IPFIX provides the flexibility to the logging device to define the data sets that it is logging. The IEs specified for logging must be the same irrespective of the encoding mechanism used.

### 1.1. Terminology

The usage of the term "NAT device" in this document refers to any NAT44 and NAT64 devices. The usage of the term "collector" refers to any device that receives the binary data from a NAT device and converts that into meaningful information. This document uses the term "Session" as it is defined in [RFC2663] and the term Binding Information Base (BIB) as it is defined in [RFC6146]. The usage of the term Information Element (IE) is defined in [RFC7011]. The term Carrier Grade NAT refers to a large scale NAT device as described in [RFC6888]

The IPFIX Information Elements that are NAT specific are created with NAT terminology. In order to avoid creating duplicate IEs, IEs are reused if they convey the same meaning. This document uses the term timestamp for the Information element which defines the time when an event is logged, this is the same as IPFIX term `observationTimeMilliseconds` as described in [IPFIX-IANA]. Since `observationTimeMilliseconds` is not self explanatory for NAT implementors, this document uses the term `timeStamp`. This document refers to event templates, that refers to IPFIX template records. This document refers to log events that refers to IPFIX Flow records.

## 1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 2. Scope

This document provides the information model to be used for logging the NAT events including Carrier Grade NAT (CGN) events. [RFC7011] provides guidance on the choices of the transport protocols used for IPFIX and their effects. This document does not provide guidance on the transport protocol like TCP, UDP or SCTP that is to be used to log NAT events. The logs SHOULD be reliably sent to the collector to ensure that the log events are not lost. The choice of the actual transport protocol is beyond the scope of this document.

The existing IANA IPFIX IEs registry [IPFIX-IANA] already has assignments for most of the NAT logging events. This document uses the allocated IPFIX IEs and will request IANA for the ones that are defined in this document but not yet allocated.

This document assumes that the NAT device will use the existing IPFIX framework to send the log events to the collector. This would mean that the NAT device will specify the template that it is going to use for each of the events. The templates can be of varying length and there could be multiple templates that a NAT device could use to log the events.

The implementation details of the collector application is beyond the scope of this document.

The optimization of logging the NAT events is left to the implementation and is beyond the scope of this document.

### 3. Deployment

NAT logging based on IPFIX uses binary encoding and hence is very efficient. IPFIX based logging is recommended for environments where a high volume of logging is required, for example, where per-flow logging is needed or in case of Carrier Grade NAT. However, IPFIX based logging requires a collector that processes the binary data and requires a network management application that converts this binary data to a human readable format.

A collector may receive NAT events from multiple CGN devices. The collector distinguishes between the devices using the source IP address, source port, and Observation Domain ID in the IPFIX header. The collector can decide to store the information based on the administrative policies that are inline with the operator and the local jurisdiction. The retention policy is not dictated by the exporter and is left to the policies that are defined at the collector.

A collector may have scale issues if it is overloaded by a large number of simultaneous events. An appropriate throttling mechanism may be used to handle the oversubscription.

The logs that are exported can be used for a variety of reasons. An example use case is to do accounting based on when the users logged on and off. The translation will be installed when the user logs on and removed when the user logs off. These events create log records. Another use case is to identify an attacker or a host in a provider network. The network administrators can use these logs to identify the usage patterns, need for additional IP addresses etc. The deployment of NAT logging is not limited to just these cases.

### 4. Event based logging

An event in a NAT device can be viewed as a state transition as it relates to the management of NAT resources. The creation and deletion of NAT sessions and bindings are examples of events as they result in resources (addresses and ports) being allocated or freed. The events can happen through the processing of data packets flowing through the NAT device or through an external entity installing policies on the NAT router or as a result of an asynchronous event like a timer. The list of events are provided in Table 2. Each of these events SHOULD be logged, unless they are administratively prohibited. A NAT device MAY log these events to multiple collectors if redundancy is required. The network administrator will specify the collectors to which the log records are to be sent. It is necessary to preserve the list of collectors and its associated information like the IPv4/IPv6 address, port and protocol across

reboots so that the configuration information is not lost when the device is restarted. The NAT device implementing the IPFIX logging MUST follow the IPFIX specs as specified in RFC 7011.

#### 4.1. Logging of destination information

Logging of destination information in a NAT event has been discussed in [RFC6302] and [RFC6888]. Logging of destination information increases the size of each record and increases the need for storage considerably. It increases the number of log events generated because when the same user connects to a different destination, it results in a log record per destination address. Logging of the source and destination addresses result in loss of privacy. Logging of destination addresses and ports, pre or post NAT, SHOULD NOT be done [RFC6888]. However, this draft provides the necessary fields to log the destination information in cases where they must be logged.

#### 4.2. Information Elements

The templates could contain a subset of the IEs shown in Table 1 depending upon the event being logged. For example a NAT44 session creation template record will contain,

```
{sourceIPv4Address, postNATSourceIPv4Address, destinationIPv4Address,
postNATDestinationIPv4Address, sourceTransportPort,
postNATSourceTransportPort, destinationTransportPort,
postNAPTDestTransportPort, internalAddressRealm, natEvent, timeStamp}
```

An example of the actual event data record is shown below - in a human readable form

```
{192.0.2.1, 203.0.113.100, 192.0.2.104, 192.0.2.104, 14800, 1024, 80,
80, 0, 1, 09:20:10:789}
```

A single NAT device could be exporting multiple templates and the collector MUST support receiving multiple templates from the same source.



The following is the table of all the IEs that a NAT device would need to export the events. The formats of the IEs and the IPFIX IDs are listed below. Some of the IPFIX IEs are not yet assigned. The detailed description of these fields that are requested are in the IANA considerations section.

Field Name	Size (bits)	IANA IPFIX ID	Description
timeStamp	64	323	System Time when the event occurred.
natInstanceId	32	TBD	NAT Instance Identifier
vlanID	16	58	VLAN ID in case of overlapping networks
ingressVRFID	32	234	VRF ID in case of overlapping networks
sourceIPv4Address	32	8	Source IPv4 Address
postNATSourceIPv4Address	32	225	Translated Source IPv4 Address
protocolIdentifier	8	4	Transport protocol
sourceTransportPort	16	7	Source Port
postNAPTsourceTransportPort	16	227	Translated Source port
destinationIPv4Address	32	12	Destination IPv4 Address
postNATDestinationIPv4Address	32	226	Translated IPv4 destination address
destinationTransportPort	16	11	Destination port
postNAPTdestinationTransportPort	16	228	Translated Destination port

sourceIPv6Address	128	27	Source IPv6 address
destinationIPv6Address	128	28	Destination IPv6 address
postNATSourceIPv6Address	128	281	Translated source IPv6 addresss
postNATDestinationIPv6Address	128	282	Translated Destination IPv6 address
internalAddressRealm	OctetArray	TBD	Source Address Realm
externalAddressRealm	OctetArray	TBD	Destination Address Realm
natEvent	8	230	Type of Event
portRangeStart	16	361	Allocated port block start
portRangeEnd	16	362	Allocated Port block end
natPoolID	32	283	NAT pool Identifier
natQuotaExceededEvent	32	TBD	Limit event identifier
natThresholdEvent	32	TBD	Threshold event identifier

Table 1: Template format Table

#### 4.3. Definition of NAT Events

The following is the complete list of NAT events and the proposed event type values. The natEvent IE is defined in the IPFIX IANA registry in <http://www.iana.org/assignments/ipfix/ipfix.xml>. The list can be expanded in the future as necessary. The data record will have the corresponding natEvent value to indicate the event that is being logged.

Note that the first two events are marked historic. These values were defined prior to the existence of this draft and outside the

IETF working group. These events are not standalone and require more information need to be conveyed to qualify the event. For example, the NAT Translation create event does not specify if it is a NAT44 or NAT64. As a result the Behave WG decided to have explicit definition for each one of the unique events. The historic events are listed here for the purpose of completeness and are already defined in the IPFIX IANA registry. Any compliant implementation SHOULD NOT implement the events that are marked historic.

Event Name	Values
NAT Translation create (Historic)	1
NAT Translation Delete (Historic)	2
NAT Addresses exhausted	3
NAT44 Session create	4
NAT44 Session delete	5
NAT64 Session create	6
NAT64 Session delete	7
NAT44 BIB create	8
NAT44 BIB delete	9
NAT64 BIB create	10
NAT64 BIB delete	11
NAT ports exhausted	12
Quota exceeded	13
Address binding create	14
Address binding delete	15
Port block allocation	16
Port block de-allocation	17
Threshold reached	18

Table 2: NAT Event ID table

#### 4.4. Quota exceeded Event types

The Quota Exceeded event is a natEvent IE described in Table 2. The Quota exceeded events are generated when the hard limits set by the administrator has been reached or exceeded. The following table shows the sub event types for the Quota exceeded or limits reached event. The events that can be reported are the Maximum session entries limit reached, Maximum BIB entries limit reached, Maximum (session/BIB) entries per user limit reached, Maximum active hosts limit reached or maximum subscribers limit reached and Maximum Fragments pending reassembly limit reached.

Quota Exceeded Event Name	Values
Maximum Session entries	1
Maximum BIB entries	2
Maximum entries per user	3
Maximum active hosts or subscribers	4
Maximum fragments pending reassembly	5

Table 3: Quota Exceeded event table

#### 4.5. Threshold reached Event types

The following table shows the sub event types for the threshold reached event. The administrator can configure the thresholds and whenever the threshold is reached or exceeded, the corresponding events are generated. The main difference between Quota Exceeded and the Threshold reached events is that, once the Quota exceeded events are hit, the packets are dropped or mappings won't be created etc, whereas, the threshold reached events will provide the operator a chance to take action before the traffic disruptions can happen. A NAT device can choose to implement one or the other or both.

The address pool high threshold event will be reported when the address pool reaches a high water mark as defined by the operator. This will serve as an indication that the operator might have to add more addresses to the pool or an indication that the subsequent users may be denied NAT translation mappings.

The address pool low threshold event will be reported when the address pool reaches a low water mark as defined by the operator. This will serve as an indication that the operator can reclaim some of the global IPv4 addresses in the pool.

The address and port mapping high threshold event is generated, when the number of ports in the configured address pool has reached a configured threshold.

The per-user address and port mapping high threshold is generated when a single user uses more address and port mapping than a configured threshold. We don't track the low threshold for per-user address and port mappings, because as the ports are freed, the address will become available. The address pool low threshold event will then be triggered so that the IPv4 global address can be reclaimed.

The Global address mapping high threshold event is generated when the maximum mappings per-user is reached for a NAT device doing paired address pooling.

Threshold Exceeded Event Name	Values
Address pool high threshold event	1
Address pool low threshold event	2
Address and port mapping high threshold event	3
Address and port mapping per user high threshold event	4
Global Address mapping high threshold event	5

Table 4: Threshold event table

#### 4.6. Templates for NAT Events

The following is the template of events that will be logged. The events below are identified at the time of this writing but the set of events is extensible. A NAT device that implements a given NAT event MUST support the mandatory IE's in the templates. Depending on the implementation and configuration various IEs that are not mandatory can be included or ignored.

##### 4.6.1. NAT44 create and delete session events

These events will be generated when a NAT44 session is created or deleted. The template will be the same, the natEvent will indicate whether it is a create or a delete event. The following is a template of the event.

The destination address and port information is optional as required by [RFC6888]. However, when the destination information is suppressed, the session log event contains the same information as the BIB event. In such cases, the NAT device SHOULD NOT send both BIB and session events.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
sourceIPv4Address	32	Yes
postNATSourceIPv4Address	32	Yes
protocolIdentifier	8	Yes
sourceTransportPort	16	Yes
postNAPTsourceTransportPort	16	Yes
destinationIPv4Address	32	No
postNATDestinationIPv4Address	32	No
destinationTransportPort	16	No
postNAPTdestinationTransportPort	16	No
natInstanceID	32	No
vlanID/ingressVRFID	32	No
internalAddressRealm	OctetArray	No
externalAddressRealm	OctetArray	No

Table 5: NAT44 Session delete/create template

## 4.6.2. NAT64 create and delete session events

These events will be generated when a NAT64 session is created or deleted. The following is a template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
sourceIPv6Address	128	Yes
postNATSourceIPv4Address	32	Yes
protocolIdentifier	8	Yes
sourceTransportPort	16	Yes
postNAPTsourceTransportPort	16	Yes
destinationIPv6Address	128	No
postNATDestinationIPv4Address	32	No
destinationTransportPort	16	No
postNAPTdestinationTransportPort	16	No
natInstanceID	32	No
vlanID/ingressVRFID	32	No
internalAddressRealm	OctetArray	No
externalAddressRealm	OctetArray	No

Table 6: NAT64 session create/delete event template

## 4.6.3. NAT44 BIB create and delete events

These events will be generated when a NAT44 Bind entry is created or deleted. The following is a template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
sourceIPv4Address	32	Yes
postNATSourceIPv4Address	32	Yes
protocolIdentifier	8	No
sourceTransportPort	16	No
postNAPTsourceTransportPort	16	No
natInstanceID	32	No
vlanID/ingressVRFID	32	No
internalAddressRealm	OctetArray	No
externalAddressRealm	OctetArray	No

Table 7: NAT44 BIB create/delete event template

## 4.6.4. NAT64 BIB create and delete events

These events will be generated when a NAT64 Bind entry is created or deleted. The following is a template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
sourceIPv6Address	128	Yes
postNATSourceIPv4Address	32	Yes
protocolIdentifier	8	No
sourceTransportPort	16	No
postNAPTsourceTransportPort	16	No
natInstanceID	32	No
vlanID/ingressVRFID	32	No
internalAddressRealm	OctetArray	No
externalAddressRealm	OctetArray	No

Table 8: NAT64 BIB create/delete event template

## 4.6.5. Addresses Exhausted event

This event will be generated when a NAT device runs out of global IPv4 addresses in a given pool of addresses. Typically, this event would mean that the NAT device won't be able to create any new translations until some addresses/ports are freed. This event SHOULD be rate limited as many packets hitting the device at the same time will trigger a burst of addresses exhausted events.

The following is a template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natPoolID	32	Yes
natInstanceID	32	No

Table 9: Address Exhausted event template

## 4.6.6. Ports Exhausted event

This event will be generated when a NAT device runs out of ports for a global IPv4 address. Port exhaustion shall be reported per protocol (UDP, TCP etc). This event SHOULD be rate limited as many packets hitting the device at the same time will trigger a burst of port exhausted events.

The following is a template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
postNATSourceIPv4Address	32	Yes
protocolIdentifier	8	Yes
natInstanceID	32	No

Table 10: Ports Exhausted event template



#### 4.6.7. Quota exceeded events

This event will be generated when a NAT device cannot allocate resources as a result of an administratively defined policy. The quota exceeded event templates are described below.

##### 4.6.7.1. Maximum session entries exceeded

The maximum session entries exceeded event is generated when the administratively configured NAT session limit is reached. The following is the template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natQuotaExceededEvent	32	Yes
configuredLimit	32	Yes
natInstanceID	32	No

Table 11: Session Entries Exceeded event template

##### 4.6.7.2. Maximum BIB entries exceeded

The maximum BIB entries exceeded event is generated when the administratively configured BIB entry limit is reached. The following is the template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natQuotaExceededEvent	32	Yes
configuredLimit	32	Yes
natInstanceID	32	No

Table 12: BIB Entries Exceeded event template

##### 4.6.7.3. Maximum entries per user exceeded

This event is generated when a single user reaches the administratively configured NAT translation limit. The following is the template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natQuotaExceededEvent	32	Yes
configuredLimit	32	Yes
sourceIPv4 address	32	Yes for NAT44
sourceIPv6 address	128	Yes for NAT64
natInstanceID	32	No
vlanID/ingressVRFID	32	No

Table 13: Per-user Entries Exceeded event template

## 4.6.7.4. Maximum active host or subscribers exceeded

This event is generated when the number of allowed hosts or subscribers reaches the administratively configured limit. The following is the template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natQuotaExceededEvent	32	Yes
configuredLimit	32	Yes
natInstanceID	32	No

Table 14: Maximum hosts/subscribers Exceeded event template

## 4.6.7.5. Maximum fragments pending reassembly exceeded

This event is generated when the number of fragments pending reassembly reaches the administratively configured limit. Note that in case of NAT64, when this condition is detected in the IPv6 to IPv4 direction, the IPv6 source address is mandatory in the template. Similarly, when this condition is detected in IPv4 to IPv6 direction, the source IPv4 address is mandatory in the template below. The following is the template of the event.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natQuotaExceededEvent	32	Yes
configuredLimit	32	Yes
sourceIPv4 address	32	Yes for NAT44
sourceIPv6 address	128	Yes for NAT64
natInstanceID	32	No
vlanID/ingressVRFID	32	No
internalAddressRealm	OctetArray	No

Table 15: Maximum fragments pending reassembly Exceeded event template

#### 4.6.8. Threshold reached events

This event will be generated when a NAT device reaches a operator configured threshold when allocating resources. The threshold reached events are described in the section above. The following is a template of the individual events.

##### 4.6.8.1. Address pool high or low threshold reached

This event is generated when the high or low threshold is reached for the address pool. The template is the same for both high and low threshold events

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natThresholdEvent	32	Yes
natPoolID	32	Yes
configuredLimit	32	Yes
natInstanceID	32	No

Table 16: Address pool high/low threshold reached event template

##### 4.6.8.2. Address and port high threshold reached

This event is generated when the high threshold is reached for the address pool and ports.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natThresholdEvent	32	Yes
configuredLimit	32	Yes
natInstanceID	32	No

Table 17: Address port high threshold reached event template

## 4.6.8.3. Per-user Address and port high threshold reached

This event is generated when the high threshold is reached for the per-user address pool and ports.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natThresholdEvent	32	Yes
configuredLimit	32	Yes
sourceIPv4 address	32	Yes for NAT44
sourceIPv6 address	128	Yes for NAT64
natInstanceID	32	No
vlanID/ingressVRFID	32	No

Table 18: Per-user Address port high threshold reached event template

## 4.6.8.4. Global Address mapping high threshold reached

This event is generated when the high threshold is reached for the per-user address pool and ports. This is generated only by NAT devices that use a paired address pooling behavior.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
natThresholdEvent	32	Yes
configuredLimit	32	Yes
natInstanceID	32	No
vlanID/ingressVRFID	32	No

Table 19: Global Address mapping high threshold reached event template

#### 4.6.9. Address binding create and delete events

These events will be generated when a NAT device binds a local address with a global address and when the global address is freed. A NAT device will generate the binding events when it receives the first packet of the first flow from a host in the private realm.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
sourceIPv4 address	32	Yes for NAT44
sourceIPv6 address	128	Yes for NAT64
Translated Source IPv4 Address	32	Yes
natInstanceID	32	No

Table 20: NAT Address Binding template

#### 4.6.10. Port block allocation and de-allocation

This event will be generated when a NAT device allocates/de-allocates ports in a bulk fashion, as opposed to allocating a port on a per flow basis.

portRangeStart represents the starting value of the range.

portRangeEnd represents the ending value of the range.

NAT devices would do this in order to reduce logs and potentially to limit the number of connections a subscriber is allowed to use. In the following Port Block allocation template, the portRangeStart and portRangeEnd MUST be specified.

It is up to the implementation to choose to consolidate log records in case two consecutive port ranges for the same user are allocated or freed.

Field Name	Size (bits)	Mandatory
timeStamp	64	Yes
natEvent	8	Yes
sourceIPv4 address	32	Yes for NAT44
sourceIPv6 address	128	Yes for NAT64
Translated Source IPv4 Address	32	Yes
portRangeStart	16	Yes
portRangeEnd	16	No
natInstanceID	32	No

Table 21: NAT Port Block Allocation event template

## 5. Management Considerations

This section considers requirements for management of the log system to support logging of the events described above. It first covers requirements applicable to log management in general. Any additional standardization required to fulfill these requirements is out of scope of the present document. Some management considerations are covered in [I-D.ietf-behave-syslog-nat-logging]. This document covers the additional considerations.

### 5.1. Ability to collect events from multiple NAT devices

An IPFIX collector **MUST** be able to collect events from multiple NAT devices and be able to decipher events based on the Observation Domain ID in the IPFIX header.

### 5.2. Ability to suppress events

The exhaustion events can be overwhelming during traffic bursts and hence **SHOULD** be handled by the NAT devices to rate limit them before sending them to the collectors. For eg. when the port exhaustion happens during bursty conditions, instead of sending a port exhaustion event for every packet, the exhaustion events **SHOULD** be rate limited by the NAT device.

## 6. Acknowledgements

Thanks to Dan Wing, Selvi Shanmugam, Mohamed Boucadir, Jacni Qin Ramji Vaithianathan, Simon Perreault, Jean-Francois Tremblay, Paul Aitken, Julia Renouard, Spencer Dawkins and Brian Trammell for their review and comments.

## 7. IANA Considerations

### 7.1. Information Elements

IANA will register the following IEs in the IPFIX Information Elements registry at <http://www.iana.org/assignments/ipfix/ipfix.xml>

#### 7.1.1. natInstanceID

Name : natInstanceID

Description: This Information Element uniquely identifies an Instance of the NAT that runs on a NAT middlebox function after the packet passed the Observation Point. natInstanceID is defined in RFC 7659 [RFC7659]

Abstract Data Type: unsigned32

Data Type Semantics: identifier

Reference:

See RFC 791 [RFC0791] for the definition of the IPv4 source address field. See RFC 3022 [RFC3022] for the definition of NAT. See RFC 3234 [RFC3234] for the definition of middleboxes.

#### 7.1.2. internalAddressRealm

Name: internalAddressRealm

Description: This Information Element represents the internal address realm where the packet is originated from or destined to. By definition, a NAT mapping can be created from two address realms, one from internal and one from external. Realms are implementation dependent and can represent a VRF ID or a VLAN ID or some unique identifier. Realms are optional and when left unspecified would mean that the external and internal realms are the same.

Abstract Data Type: octetArray

Data Type Semantics: identifier

## Reference:

See RFC 791 [RFC0791] for the definition of the IPv4 source address field. See RFC 3022 [RFC3022] for the definition of NAT. See RFC 3234 [RFC3234] for the definition of middleboxes.

## 7.1.3. externalAddressRealm

Name: externalAddressRealm

Description: This Information Element represents the external address realm where the packet is originated from or destined to. The detailed definition is in the internal address realm as specified above.

Abstract Data Type: octetArray

Data Type Semantics: identifier

## Reference:

See RFC 791 [RFC0791] for the definition of the IPv4 source address field. See RFC 3022 [RFC3022] for the definition of NAT. See RFC 3234 [RFC3234] for the definition of middleboxes.

## 7.1.4. natQuotaExceededEvent

Values of this Information Element are defined in a registry maintained by IANA at <<http://www.iana.org/assignments/ipfix/ipfix.xml#TBD-by-IANA>>. New assignments of values will be administered by IANA, subject to Expert Review [RFC5226]. Experts need to check definitions of new values for completeness, accuracy, and redundancy.

Name : natQuotaExceededEvent

Description: This Information Element identifies the type of a NAT quota exceeded event. Values for this Information Element are listed in the NAT quota exceed event type registry, see [<http://www.iana.org/assignments/ipfix/ipfix.xml#TBD-by-IANA>] Initial values in the registry are defined by the table below.



Quota Exceeded Event Name	Values
Maximum Session entries	1
Maximum BIB entries	2
Maximum entries per user	3
Maximum active hosts or subscribers	4
Maximum fragments pending reassembly	5

Table 22

Abstract Data Type: unsigned32

Data Type Semantics: identifier

Reference:

See RFC 791 [RFC0791] for the definition of the IPv4 source address field. See RFC 3022 [RFC3022] for the definition of NAT. See RFC 3234 [RFC3234] for the definition of middleboxes.

#### 7.1.5. natThresholdEvent

Values of this Information Element are defined in a registry maintained by IANA at <http://www.iana.org/assignments/ipfix/ipfix.xml#TBD-by-IANA>. New assignments of values will be administered by IANA, subject to Expert Review [RFC5226]. Experts need to check definitions of new values for completeness, accuracy, and redundancy.

Name: natThresholdEvent

Description: This Information Element identifies a type of a NAT threshold event. Values for this Information Element are listed in the NAT threshold event type registry, see <http://www.iana.org/assignments/ipfix/ipfix.xml#TBD-by-IANA>. Initial values in the registry are defined by the table below.

Threshold Exceeded Event Name	Values
Address pool high threshold event	1
Address pool low threshold event	2
Address and port mapping high threshold event	3
Address and port mapping per user high threshold event	4
Global Address mapping high threshold event	5

Table 23

Abstract Data Type: unsigned32

Data Type Semantics: identifier

Reference:

See RFC 791 [RFC0791] for the definition of the IPv4 source address field. See RFC 3022 [RFC3022] for the definition of NAT. See RFC 3234 [RFC3234] for the definition of middleboxes.

#### 7.1.6. natEvent

The original definition of this Information Element specified only three values 1, 2, and 3. This definition is replaced by a registry, to which new values can be added. The semantics of the three originally defined values remains unchanged. IANA maintains the registry for values of this Information Element at <http://www.iana.org/assignments/ipfix/ipfix.xml#TBD-by-IANA>. New assignments of values will be administered by IANA, subject to Expert Review [RFC5226]. Experts need to check definitions of new values for completeness, accuracy, and redundancy.

Name : natEvent

Description: Description: This Information Element identifies a NAT event. This IE identifies the type of a NAT event. Examples of NAT events include but not limited to, creation or deletion of a NAT translation entry, a threshold reached or exceeded etc. Values for this Information Element are listed in the NAT event type registry, see [<http://www.iana.org/assignments/ipfix/ipfix.xml#TBD-by-IANA>] The NAT Event values in the registry are defined by the Table 2 in Section 5.3.

Abstract Data Type: unsigned8

Data Type Semantics: identifier

Element ID : 230

Reference:

See RFC 3022 [RFC3022] for the definition of NAT. See RFC 3234 [RFC3234] for the definition of middleboxes. See [thisRFC] for the definitions of values 4-16.

## 8. Security Considerations

The security considerations listed in detail for IPFIX in [RFC7011] applies to this draft as well. As described in [RFC7011] the messages exchanged between the NAT device and the collector MUST be protected to provide confidentiality, integrity and authenticity. Without those characteristics, the messages are subject to various kinds of attacks. These attacks are described in great detail in [RFC7011].

This document re-emphasizes the use of TLS or DTLS for exchanging the log messages between the NAT device and the collector. The log events sent in clear text can result in confidential data being exposed to attackers, who could then spoof log events based on the information in clear text messages. Hence, the log events SHOULD NOT be sent in clear text.

The logging of NAT events can result in privacy concerns as result of exporting information such as source address and port information. The logging of destination information can also cause privacy concerns but it has been well documented in [RFC6888]. A NAT device can choose to operate in various logging modes if it wants to avoid logging of private information. The collector that receives the information can also choose to mask the private information but generate reports based on abstract data. It is outside the scope of this document to address the implementation of logging modes for privacy considerations.

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <<http://www.rfc-editor.org/info/rfc4787>>.
- [RFC5382] Guha, S., Ed., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, DOI 10.17487/RFC5382, October 2008, <<http://www.rfc-editor.org/info/rfc5382>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<http://www.rfc-editor.org/info/rfc6146>>.
- [RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, DOI 10.17487/RFC6302, June 2011, <<http://www.rfc-editor.org/info/rfc6302>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<http://www.rfc-editor.org/info/rfc6888>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<http://www.rfc-editor.org/info/rfc7011>>.
- [RFC7659] Perreault, S., Tsou, T., Sivakumar, S., and T. Taylor, "Definitions of Managed Objects for Network Address Translators (NATs)", RFC 7659, DOI 10.17487/RFC7659, October 2015, <<http://www.rfc-editor.org/info/rfc7659>>.

## 9.2. Informative References

- [I-D.ietf-behave-syslog-nat-logging]  
Chen, Z., Zhou, C., Tsou, T., and T. Taylor, "Syslog Format for NAT Logging", draft-ietf-behave-syslog-nat-logging-06 (work in progress), January 2014.
- [IPFIX-IANA]  
IANA, "IPFIX Information Elements registry", <<http://www.iana.org/assignments/ipfix>>.

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<http://www.rfc-editor.org/info/rfc2663>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<http://www.rfc-editor.org/info/rfc3022>>.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <<http://www.rfc-editor.org/info/rfc3234>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <<http://www.rfc-editor.org/info/rfc5424>>.

#### Authors' Addresses

Senthil Sivakumar  
Cisco Systems  
7100-8 Kit Creek Road  
Research Triangle Park, North Carolina 27709  
USA

Phone: +1 919 392 5158  
Email: [ssenthil@cisco.com](mailto:ssenthil@cisco.com)

Renaldo Penno  
Cisco Systems  
170 W Tasman Drive  
San Jose, California 95035  
USA

Email: [repenno@cisco.com](mailto:repenno@cisco.com)

Network Working Group  
Internet-Draft  
Obsoletes: 4008 (if approved)  
Intended status: Standards Track  
Expires: July 28, 2014

S. Perreault  
Viagenie  
T. Tsou  
Huawei Technologies (USA)  
S. Sivakumar  
Cisco Systems  
January 24, 2014

Definitions of Managed Objects for Network Address Translators (NAT)  
draft-ietf-behave-nat-mib-11

Abstract

This memo defines a portion of the Management Information Base (MIB) for devices implementing Network Address Translator (NAT) function. This MIB module may be used for monitoring of a device capable of NAT function.

This document obsoletes RFC 4008.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 28, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. The Internet-Standard Management Framework . . . . .	2
3. Overview . . . . .	3
3.1. Deprecated Features . . . . .	3
3.2. New Features . . . . .	4
3.3. Realms . . . . .	5
4. Definitions . . . . .	5
5. Security Considerations . . . . .	86
6. IANA Considerations . . . . .	88
7. References . . . . .	88
7.1. Normative References . . . . .	88
7.2. Informative References . . . . .	89
Authors' Addresses . . . . .	90

## 1. Introduction

This memo defines a portion of the Management Information Base (MIB) for devices implementing NAT function. This MIB module may be used for monitoring of a device capable of NAT function. Using it for configuration is deprecated. NAT types and their characteristics are defined in [RFC2663]. Traditional NAT function, in particular is defined in [RFC3022]. This MIB does not address the firewall functions and must not be used for configuring or monitoring these. Section 2 provides references to the SNMP management framework, which was used as the basis for the MIB module definition. Section 3 provides an overview of the MIB features. Lastly, Section 4 has the complete NAT MIB definition.

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].

## 2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally

accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

### 3. Overview

#### 3.1. Deprecated Features

All objects defined in [RFC4008] have been marked with "STATUS deprecated" for the following reasons:

**Writability:** Experience with NAT has shown that implementations vary tremendously. The NAT algorithms and data structures have little in common across devices, and this results in wildly incompatible configuration parameters. Therefore, few implementations were ever able to claim full compliance.

Lesson learned: the MIB should be read-only as much as possible.

**Exposing configuration parameters:** Even in read-only mode, many configuration parameters were exposed by [RFC4008] (e.g. timeouts). Since implementations vary wildly in their sets of configuration parameters, few implementations could claim even basic compliance.

Lesson learned: the NAT MIB's purpose is not to expose configuration parameters.

**Interfaces:** Objects from [RFC4008] tie NAT state with interfaces (e.g. the interface table, the way map entries are grouped by interface). Many NAT implementations either never keep track of the interface or associate a mapping to a set of interfaces. Since interfaces are at the core of [RFC4008], many NAT devices were unable to have a proper implementation.

Lesson learned: NAT is a logical function that may be independent of interfaces. Do not tie NAT state with interfaces.

**NAT service types:** [RFC4008] used four categories of NAT service: basicNat, napt, bidirectionalNat, twiceNat. These are ill-defined and many implementations either use different categories or do not use categories at all.

Lesson learned: do not try to categorize NAT types.



Limited transport protocol set: The set of transport protocols was defined as: other, icmp, udp, tcp. Furthermore, the numeric values corresponding to those labels were arbitrary, without relation to the actual standard protocol numbers. This meant that NAT implementations were limited to those protocols and were unable to expose information about DCCP, SCTP, etc.

Lesson learned: use standard transport protocol numbers.

### 3.2. New Features

New features in this module are as follows:

Counters: Many new counters are introduced. Most of them are available in two variants: global and per-transport protocol.

Limits: A few limits on the quantity of state data stored by the NAT device. Some of them can trigger notifications.

Address+Port Pools: Pools of external addresses and ports are often used in enterprise and ISP settings. Pools are listed in a table, each with its range of addresses and ports. It is possible to inspect each pool's usage, to set limits, and to receive notifications when thresholds are crossed.

Address Mappings: NATs that have an "IP address pooling" behavior of "Paired" [RFC4787] maintain a mapping from internal address to external address. This module allows inspection of this mapping table.

Mapping table indexed by external 3-tuple: It is often necessary to determine the internal address that is mapped to a given external address and port. This MIB provides this table with an index to accomplish this efficiently, without having to iterate over all mappings.

Realms: See Section 3.3.

RFC 4787 terminology: Mapping table entries indicate the mapping behavior, the filtering behavior, and the address pooling behavior that were used to create the mapping.

Subscriber awareness: With the advent of CGN deployment, a set of subscriber specific counters, limits and parameters are added.

NAT instances: Multiple NAT instances may be managed by a single SNMP agent. All instance-specific objects (counters, limits, etc.) are indexed by NAT instance ID. In addition, NAT instances may be reliably identified using the `natInstanceAlias` object.

### 3.3. Realms

Current NAT devices commonly allow the internal and external parts of a mapping to come from different realms. The meaning of "realm" is implementation-dependent. On some implementations it can be equivalent to the name of a VPN Routing and Forwarding table (VRF). On others it is simply the numeric index of a virtual routing table. Note that this usage of "realm" is completely different from the one in [RFC4008].

This MIB allows the realm to be indicated where it makes sense. The format is an `SnmpAdminString`. On platforms that identify realms with integers, the string representation of the integer is used instead. The empty string has special meaning: it refers to the default realm.

Note that many MIBs implicitly support realms in one form or another by using SNMPv3 contexts. See for example the OSPFv2 MIB [RFC4750]. This method cannot be used for the NAT MIB because mappings can belong to two realms simultaneously: the internal part can be in one realm while the external part is in another. In such cases the NAT function acts like a "wormhole" between two realms. Using contexts would implicitly impose the restriction that all objects would have to belong to the same realm.

## 4. Definitions

This MIB module IMPORTs objects from [RFC2578], [RFC2579], and [RFC4001].

NAT-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY,  
OBJECT-TYPE,  
Integer32,  
Unsigned32,  
Gauge32,  
Counter64,  
TimeTicks,  
mib-2,  
NOTIFICATION-TYPE  
FROM SNMPv2-SMI  
TEXTUAL-CONVENTION,

```
    DisplayString,
    StorageType,
    RowStatus
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    NOTIFICATION-GROUP,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ifIndex,
    ifCounterDiscontinuityGroup,
    InterfaceIndex
        FROM IF-MIB
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    InetAddressType,
    InetAddress,
    InetAddressPrefixLength,
    InetPortNumber
        FROM INET-ADDRESS-MIB
    VPNIdOrZero
        FROM VPN-TC-STD-MIB;

natMIB MODULE-IDENTITY
    LAST-UPDATED "201304260000Z"
    -- RFC Ed.: set to publication date
    ORGANIZATION
        "IETF Behavior Engineering for Hindrance Avoidance
        (BEHAVE) Working Group"
    CONTACT-INFO
        "Working Group Email: behave@ietf.org

        Simon Perreault
        Viagenie
        246 Aberdeen
        Quebec, QC G1R 2E1
        Canada

        Phone: +1 418 656 9254
        Email: simon.perreault@viagenie.ca
        URI: http://viagenie.ca

        Tina Tsou
        Huawei Technologies (USA)
        2330 Central Expressway
        Santa Clara, CA 95050
        USA
```

Phone: +1 408 330 4424  
Email: tina.tsou.zouting@huawei.com

Senthil Sivakumar  
Cisco Systems  
7100-8 Kit Creek Road  
Research Triangle Park, North Carolina 27709  
USA

Phone: +1 919 392 5158  
Email: ssenthil@cisco.com"

## DESCRIPTION

"This MIB module defines the generic managed objects  
for NAT.

Copyright (C) The Internet Society (2013). This  
version of this MIB module is part of RFC yyyy; see  
the RFC itself for full legal notices."

-- RFC Ed.: replace yyyy with actual RFC number & remove this note"

REVISION "201304260000Z"

-- RFC Ed.: set to publication date

## DESCRIPTION

"Complete rewrite, published as RFC yyyy."

-- RFC Ed.: replace yyyy with actual RFC number & set date"

REVISION "200503210000Z" -- 21th March 2005

## DESCRIPTION

"Initial version, published as RFC 4008."

::= { mib-2 123 }

natMIBObjects OBJECT IDENTIFIER ::= { natMIB 1 }

NatProtocolType ::= TEXTUAL-CONVENTION

STATUS deprecated

## DESCRIPTION

"A list of protocols that support the network  
address translation. Inclusion of the values is  
not intended to imply that those protocols  
need to be supported. Any change in this  
TEXTUAL-CONVENTION should also be reflected in  
the definition of NatProtocolMap, which is a  
BITS representation of this."

SYNTAX INTEGER {  
    none (1), -- not specified  
    other (2), -- none of the following  
    icmp (3),  
    udp (4),  
    tcp (5)

```
}
```

```
NatProtocolMap ::= TEXTUAL-CONVENTION
    STATUS      deprecated
    DESCRIPTION
        "A bitmap of protocol identifiers that support
        the network address translation. Any change
        in this TEXTUAL-CONVENTION should also be
        reflected in the definition of NatProtocolType."
    SYNTAX      BITS {
        other (0),
        icmp (1),
        udp (2),
        tcp (3)
    }
```

```
NatAddrMapId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS deprecated
    DESCRIPTION
        "A unique id that is assigned to each address map
        by a NAT enabled device."
    SYNTAX      Unsigned32 (1..4294967295)
```

```
NatBindIdOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS deprecated
    DESCRIPTION
        "A unique id that is assigned to each bind by
        a NAT enabled device. The bind id will be zero
        in the case of a Symmetric NAT."
    SYNTAX      Unsigned32 (0..4294967295)
```

```
NatBindId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS deprecated
    DESCRIPTION
        "A unique id that is assigned to each bind by
        a NAT enabled device."
    SYNTAX      Unsigned32 (1..4294967295)
```

```
NatSessionId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS deprecated
    DESCRIPTION
        "A unique id that is assigned to each session by
        a NAT enabled device."
    SYNTAX      Unsigned32 (1..4294967295)
```

```
NatBindMode ::= TEXTUAL-CONVENTION
    STATUS deprecated
    DESCRIPTION
        "An indication of whether the bind is
        an address bind or an address port bind."
    SYNTAX    INTEGER {
                addressBind (1),
                addressPortBind (2)
            }

NatAssociationType ::= TEXTUAL-CONVENTION
    STATUS deprecated
    DESCRIPTION
        "An indication of whether the association is
        static or dynamic."
    SYNTAX    INTEGER {
                static (1),
                dynamic (2)
            }

NatTranslationEntity ::= TEXTUAL-CONVENTION
    STATUS deprecated
    DESCRIPTION
        "An indication of a) the direction of a session for
        which an address map entry, address bind or port
        bind is applicable, and b) the entity (source or
        destination) within the session that is subject to
        translation."
    SYNTAX    BITS {
                inboundSrcEndPoint (0),
                outboundDstEndPoint(1),
                inboundDstEndPoint (2),
                outboundSrcEndPoint(3)
            }

--
-- Default Values for the Bind and NAT Protocol Timers
--

natDefTimeouts OBJECT IDENTIFIER ::= { natMIBObjects 1 }

natNotifCtrl OBJECT IDENTIFIER ::= { natMIBObjects 2 }

--
-- Address Bind and Port Bind related NAT configuration
--
```

```
natBindDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (0..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      deprecated
    DESCRIPTION
        "The default Bind (Address Bind or Port Bind) idle
         timeout parameter.

         If the agent is capable of storing non-volatile
         configuration, then the value of this object must be
         restored after a re-initialization of the management
         system."
    DEFVAL { 0 }
    ::= { natDefTimeouts 1 }

--
-- UDP related NAT configuration
--

natUdpDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      deprecated
    DESCRIPTION
        "The default UDP idle timeout parameter.

         If the agent is capable of storing non-volatile
         configuration, then the value of this object must be
         restored after a re-initialization of the management
         system."
    DEFVAL { 300 }
    ::= { natDefTimeouts 2 }

--
-- ICMP related NAT configuration
--

natIcmpDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      deprecated
    DESCRIPTION
        "The default ICMP idle timeout parameter.

         If the agent is capable of storing non-volatile
```

```

        configuration, then the value of this object must be
        restored after a re-initialization of the management
        system."
    DEFVAL { 300 }
    ::= { natDefTimeouts 3 }

--
-- Other protocol parameters
--

natOtherDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS        "seconds"
    MAX-ACCESS   read-write
    STATUS        deprecated
    DESCRIPTION
        "The default idle timeout parameter for protocols
        represented by the value other (2) in
        NatProtocolType.

        If the agent is capable of storing non-volatile
        configuration, then the value of this object must be
        restored after a re-initialization of the management
        system."
    DEFVAL { 60 }
    ::= { natDefTimeouts 4 }

--
-- TCP related NAT Timers
--

natTcpDefIdleTimeout OBJECT-TYPE
    SYNTAX      Unsigned32  (1..4294967295)
    UNITS        "seconds"
    MAX-ACCESS   read-write
    STATUS        deprecated
    DESCRIPTION
        "The default time interval that a NAT session for an
        established TCP connection is allowed to remain
        valid without any activity on the TCP connection.

        If the agent is capable of storing non-volatile
        configuration, then the value of this object must be
        restored after a re-initialization of the management
        system."
    DEFVAL { 86400 }
    ::= { natDefTimeouts 5 }
```



## natTcpDefNegTimeout OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

UNITS "seconds"

MAX-ACCESS read-write

STATUS deprecated

DESCRIPTION

"The default time interval that a NAT session for a TCP connection that is not in the established state is allowed to remain valid without any activity on the TCP connection.

If the agent is capable of storing non-volatile configuration, then the value of this object must be restored after a re-initialization of the management system."

DEFVAL { 60 }

::= { natDefTimeouts 6 }

## natNotifThrottlingInterval OBJECT-TYPE

SYNTAX Integer32 (0 | 5..3600)

UNITS "seconds"

MAX-ACCESS read-write

STATUS deprecated

DESCRIPTION

"This object controls the generation of the natPacketDiscard notification.

If this object has a value of zero, then no natPacketDiscard notifications will be transmitted by the agent.

If this object has a non-zero value, then the agent must not generate more than one natPacketDiscard 'notification-event' in the indicated period, where a 'notification-event' is the generation of a single notification PDU type to a list of notification destinations. If additional NAT packets are discarded within the throttling period, then notification-events for these changes must be suppressed by the agent until the current throttling period expires.

If natNotifThrottlingInterval notification generation is enabled, the suggested default throttling period is 60 seconds, but generation of the natPacketDiscard notification should be disabled by default.

If the agent is capable of storing non-volatile configuration, then the value of this object must be

restored after a re-initialization of the management system.

The actual transmission of notifications is controlled via the MIB modules in RFC 3413."

DEFVAL { 0 }

::= { natNotifCtrl 1 }

--

-- The NAT Interface Table

--

natInterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatInterfaceEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"This table specifies the attributes for interfaces on a device supporting NAT function."

::= { natMIBObjects 3 }

natInterfaceEntry OBJECT-TYPE

SYNTAX NatInterfaceEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"Each entry in the natInterfaceTable holds a set of parameters for an interface, instantiated by ifIndex. Therefore, the interface index must have been assigned, according to the applicable procedures, before it can be meaningfully used. Generally, this means that the interface must exist.

When natStorageType is of type nonVolatile, however, this may reflect the configuration for an interface whose ifIndex has been assigned but for which the supporting implementation is not currently present."

INDEX { ifIndex }

::= { natInterfaceTable 1 }

NatInterfaceEntry ::= SEQUENCE {

natInterfaceRealm INTEGER,

natInterfaceServiceType BITS,

natInterfaceInTranslates Counter64,

natInterfaceOutTranslates Counter64,

natInterfaceDiscards Counter64,

natInterfaceStorageType StorageType,

```
    natInterfaceRowStatus      RowStatus
  }

natInterfaceRealm OBJECT-TYPE
    SYNTAX      INTEGER {
                    private (1),
                    public (2)
                }
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "This object identifies whether this interface is
        connected to the private or the public realm."
    DEFVAL      { public }
    ::= { natInterfaceEntry 1 }

natInterfaceServiceType OBJECT-TYPE
    SYNTAX      BITS {
                    basicNat (0),
                    napt (1),
                    bidirectionalNat (2),
                    twiceNat (3)
                }
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "An indication of the direction in which new sessions
        are permitted and the extent of translation done within
        the IP and transport headers."
    ::= { natInterfaceEntry 2 }

natInterfaceInTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "Number of packets received on this interface that
        were translated.
        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
    ::= { natInterfaceEntry 3 }

natInterfaceOutTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
```

## DESCRIPTION

"Number of translated packets that were sent out this interface.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natInterfaceEntry 4 }

## natInterfaceDiscards OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS deprecated

## DESCRIPTION

"Number of packets that had to be rejected/dropped due to a lack of resources for this interface.

Discontinuities in the value of this counter can occur at reinitialization of the management system and at other times as indicated by the value of ifCounterDiscontinuityTime on the relevant interface."

::= { natInterfaceEntry 5 }

## natInterfaceStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS deprecated

## DESCRIPTION

"The storage type for this conceptual row. Conceptual rows having the value 'permanent' need not allow write-access to any columnar objects in the row."

## REFERENCE

"Textual Conventions for SMIV2, Section 2."

DEFVAL { nonVolatile }

::= { natInterfaceEntry 6 }

## natInterfaceRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS deprecated

## DESCRIPTION

"The status of this conceptual row.

Until instances of all corresponding columns are appropriately configured, the value of the corresponding instance of the natInterfaceRowStatus

column is 'notReady'.

In particular, a newly created row cannot be made active until the corresponding instance of natInterfaceServiceType has been set.

None of the objects in this row may be modified while the value of this object is active(1)."

#### REFERENCE

"Textual Conventions for SMIV2, Section 2."

::= { natInterfaceEntry 7 }

--

-- The Address Map Table

--

natAddrMapTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatAddrMapEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"This table lists address map parameters for NAT."

::= { natMIBObjects 4 }

natAddrMapEntry OBJECT-TYPE

SYNTAX NatAddrMapEntry

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"This entry represents an address map to be used for NAT and contributes to the dynamic and/or static address mapping tables of the NAT device."

INDEX { ifIndex, natAddrMapIndex }

::= { natAddrMapTable 1 }

NatAddrMapEntry ::= SEQUENCE {

natAddrMapIndex

natAddrMapName

natAddrMapEntryType

natAddrMapTranslationEntity

natAddrMapLocalAddrType

natAddrMapLocalAddrFrom

natAddrMapLocalAddrTo

natAddrMapLocalPortFrom

natAddrMapLocalPortTo

natAddrMapGlobalAddrType

natAddrMapGlobalAddrFrom

NatAddrMapId,

SnmpAdminString,

NatAssociationType,

NatTranslationEntity,

InetAddressType,

InetAddress,

InetAddress,

InetPortNumber,

InetPortNumber,

InetAddressType,

InetAddress,

```

    natAddrMapGlobalAddrTo      InetAddress,
    natAddrMapGlobalPortFrom    InetPortNumber,
    natAddrMapGlobalPortTo      InetPortNumber,
    natAddrMapProtocol          NatProtocolMap,
    natAddrMapInTranslates      Counter64,
    natAddrMapOutTranslates     Counter64,
    natAddrMapDiscards          Counter64,
    natAddrMapAddrUsed          Gauge32,
    natAddrMapStorageType       StorageType,
    natAddrMapRowStatus         RowStatus
}

natAddrMapIndex OBJECT-TYPE
    SYNTAX      NatAddrMapId
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "Along with ifIndex, this object uniquely
        identifies an entry in the natAddrMapTable.
        Address map entries are applied in the order
        specified by natAddrMapIndex."
    ::= { natAddrMapEntry 1 }

natAddrMapName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "Name identifying all map entries in the table associated
        with the same interface. All map entries with the same
        ifIndex MUST have the same map name."
    ::= { natAddrMapEntry 2 }

natAddrMapEntryType OBJECT-TYPE
    SYNTAX      NatAssociationType
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "This parameter can be used to set up static
        or dynamic address maps."
    ::= { natAddrMapEntry 3 }

natAddrMapTranslationEntity OBJECT-TYPE
    SYNTAX      NatTranslationEntity
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "The end-point entity (source or destination) in

```

inbound or outbound sessions (i.e., first packets) that may be translated by an address map entry.

Session direction (inbound or outbound) is derived from the direction of the first packet of a session traversing a NAT interface. NAT address (and Transport-ID) maps may be defined to effect inbound or outbound sessions.

Traditionally, address maps for Basic NAT and NATPT are configured on a public interface for outbound sessions, effecting translation of source end-point. The value of this object must be set to outboundSrcEndPoint for those interfaces.

Alternately, if address maps for Basic NAT and NATPT were to be configured on a private interface, the desired value for this object for the map entries would be inboundSrcEndPoint (i.e., effecting translation of source end-point for inbound sessions).

If TwiceNAT were to be configured on a private interface, the desired value for this object for the map entries would be a bitmask of inboundSrcEndPoint and inboundDstEndPoint."

```
::= { natAddrMapEntry 4 }
```

natAddrMapLocalAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"This object specifies the address type used for natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo."

```
::= { natAddrMapEntry 5 }
```

natAddrMapLocalAddrFrom OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"This object specifies the first IP address of the range of IP addresses mapped by this translation entry. The value of this object must be less than or equal to the value of the natAddrMapLocalAddrTo object.

The type of this address is determined by the value of the natAddrMapLocalAddrType object."

```
::= { natAddrMapEntry 6 }
```

natAddrMapLocalAddrTo OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"This object specifies the last IP address of the range of IP addresses mapped by this translation entry. If only a single address is being mapped, the value of this object is equal to the value of natAddrMapLocalAddrFrom. For a static NAT, the number of addresses in the range defined by natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo must be equal to the number of addresses in the range defined by natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo. The value of this object must be greater than or equal to the value of the natAddrMapLocalAddrFrom object.

The type of this address is determined by the value of the natAddrMapLocalAddrType object."

```
::= { natAddrMapEntry 7 }
```

natAddrMapLocalPortFrom OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the first port number in the range of ports being mapped.

The value of this object must be less than or equal to the value of the natAddrMapLocalPortTo object. If the translation specifies a single port, then the value of this object is equal to the value of natAddrMapLocalPortTo."

DEFVAL { 0 }

```
::= { natAddrMapEntry 8 }
```

natAddrMapLocalPortTo OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"If this conceptual row describes a Basic NAT address



mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the last port number in the range of ports being mapped.

The value of this object must be greater than or equal to the value of the natAddrMapLocalPortFrom object. If the translation specifies a single port, then the value of this object is equal to the value of natAddrMapLocalPortFrom."

```
DEFVAL { 0 }  
::= { natAddrMapEntry 9 }
```

natAddrMapGlobalAddrType OBJECT-TYPE

```
SYNTAX      InetAddressType  
MAX-ACCESS  read-create  
STATUS      deprecated  
DESCRIPTION
```

"This object specifies the address type used for natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo."

```
::= { natAddrMapEntry 10 }
```

natAddrMapGlobalAddrFrom OBJECT-TYPE

```
SYNTAX      InetAddress  
MAX-ACCESS  read-create  
STATUS      deprecated  
DESCRIPTION
```

"This object specifies the first IP address of the range of IP addresses being mapped to. The value of this object must be less than or equal to the value of the natAddrMapGlobalAddrTo object.

The type of this address is determined by the value of the natAddrMapGlobalAddrType object."

```
::= { natAddrMapEntry 11 }
```

natAddrMapGlobalAddrTo OBJECT-TYPE

```
SYNTAX      InetAddress  
MAX-ACCESS  read-create  
STATUS      deprecated  
DESCRIPTION
```

"This object specifies the last IP address of the range of IP addresses being mapped to. If only a single address is being mapped to, the value of this object is equal to the value of natAddrMapGlobalAddrFrom. For a static NAT, the number of addresses in the range defined by natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo must be equal to the number of addresses in the range

defined by natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo. The value of this object must be greater than or equal to the value of the natAddrMapGlobalAddrFrom object.

The type of this address is determined by the value of the natAddrMapGlobalAddrType object."

::= { natAddrMapEntry 12 }

natAddrMapGlobalPortFrom OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the first port number in the range of ports being mapped to.

The value of this object must be less than or equal to the value of the natAddrMapGlobalPortTo object. If the translation specifies a single port, then the value of this object is equal to the value natAddrMapGlobalPortTo."

DEFVAL { 0 }

::= { natAddrMapEntry 13 }

natAddrMapGlobalPortTo OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"If this conceptual row describes a Basic NAT address mapping, then the value of this object must be zero. If this conceptual row describes NAPT, then the value of this object specifies the last port number in the range of ports being mapped to.

The value of this object must be greater than or equal to the value of the natAddrMapGlobalPortFrom object. If the translation specifies a single port, then the value of this object is equal to the value of natAddrMapGlobalPortFrom."

DEFVAL { 0 }

::= { natAddrMapEntry 14 }

natAddrMapProtocol OBJECT-TYPE  
SYNTAX NatProtocolMap  
MAX-ACCESS read-create  
STATUS deprecated  
DESCRIPTION  
    "This object specifies a bitmap of protocol identifiers."  
 ::= { natAddrMapEntry 15 }

natAddrMapInTranslates OBJECT-TYPE  
SYNTAX Counter64  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
    "The number of inbound packets pertaining to this address  
    map entry that were translated.  
  
    Discontinuities in the value of this counter can occur  
    at reinitialization of the management system and at  
    other times, as indicated by the value of  
    ifCounterDiscontinuityTime on the relevant interface."  
 ::= { natAddrMapEntry 16 }

natAddrMapOutTranslates OBJECT-TYPE  
SYNTAX Counter64  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
    "The number of outbound packets pertaining to this  
    address map entry that were translated.  
  
    Discontinuities in the value of this counter can occur  
    at reinitialization of the management system and at  
    other times, as indicated by the value of  
    ifCounterDiscontinuityTime on the relevant interface."  
 ::= { natAddrMapEntry 17 }

natAddrMapDiscards OBJECT-TYPE  
SYNTAX Counter64  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
    "The number of packets pertaining to this address map  
    entry that were dropped due to lack of addresses in the  
    address pool identified by this address map. The value  
    of this object must always be zero in case of static  
    address map.  
  
    Discontinuities in the value of this counter can occur

```
        at reinitialization of the management system and at
        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrMapEntry 18 }

natAddrMapAddrUsed OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of addresses pertaining to this address map
        that are currently being used from the NAT pool.
        The value of this object must always be zero in the case
        of a static address map."
 ::= { natAddrMapEntry 19 }

natAddrMapStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "The storage type for this conceptual row.
        Conceptual rows having the value 'permanent'
        need not allow write-access to any columnar objects
        in the row."
    REFERENCE
        "Textual Conventions for SMIV2, Section 2."
    DEFVAL { nonVolatile }
 ::= { natAddrMapEntry 20 }

natAddrMapRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "The status of this conceptual row.

        Until instances of all corresponding columns are
        appropriately configured, the value of the
        corresponding instance of the natAddrMapRowStatus
        column is 'notReady'.

        None of the objects in this row may be modified
        while the value of this object is active(1)."
```

```
REFERENCE
    "Textual Conventions for SMIV2, Section 2."
 ::= { natAddrMapEntry 21 }
```

```
--
-- Address Bind section
--

natAddrBindNumberOfEntries OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "This object maintains a count of the number of entries
         that currently exist in the natAddrBindTable."
    ::= { natMIBObjects 5 }

--
-- The NAT Address BIND Table
--

natAddrBindTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NatAddrBindEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This table holds information about the currently
         active NAT BINDs."
    ::= { natMIBObjects 6 }

natAddrBindEntry OBJECT-TYPE
    SYNTAX      NatAddrBindEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "Each entry in this table holds information about
         an active address BIND.  These entries are lost
         upon agent restart.

        This row has indexing which may create variables with
        more than 128 subidentifiers.  Implementers of this
        table must be careful not to create entries that would
        result in OIDs which exceed the 128 subidentifier limit.
        Otherwise, the information cannot be accessed using
        SNMPv1, SNMPv2c or SNMPv3."

    INDEX      { ifIndex,
                 natAddrBindLocalAddrType,
                 natAddrBindLocalAddr }
    ::= { natAddrBindTable 1 }

NatAddrBindEntry ::= SEQUENCE {
```

```

    natAddrBindLocalAddrType      InetAddressType,
    natAddrBindLocalAddr          InetAddress,
    natAddrBindGlobalAddrType     InetAddressType,
    natAddrBindGlobalAddr         InetAddress,
    natAddrBindId                 NatBindId,
    natAddrBindTranslationEntity  NatTranslationEntity,
    natAddrBindType               NatAssociationType,
    natAddrBindMapIndex           NatAddrMapId,
    natAddrBindSessions           Gauge32,
    natAddrBindMaxIdleTime        TimeTicks,
    natAddrBindCurrentIdleTime    TimeTicks,
    natAddrBindInTranslates       Counter64,
    natAddrBindOutTranslates      Counter64
}

natAddrBindLocalAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This object specifies the address type used for
         natAddrBindLocalAddr."
    ::= { natAddrBindEntry 1 }

natAddrBindLocalAddr OBJECT-TYPE
    SYNTAX      InetAddress (SIZE (4|16))
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This object represents the private-realm specific
         network layer address, which maps to the public-realm
         address represented by natAddrBindGlobalAddr.

         The type of this address is determined by the value of
         the natAddrBindLocalAddrType object."
    ::= { natAddrBindEntry 2 }

natAddrBindGlobalAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "This object specifies the address type used for
         natAddrBindGlobalAddr."
    ::= { natAddrBindEntry 3 }

natAddrBindGlobalAddr OBJECT-TYPE
    SYNTAX      InetAddress

```

MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
    "This object represents the public-realm network layer address that maps to the private-realm network layer address represented by natAddrBindLocalAddr.  
  
    The type of this address is determined by the value of the natAddrBindGlobalAddrType object."  
::= { natAddrBindEntry 4 }

natAddrBindId OBJECT-TYPE  
SYNTAX NatBindId  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
    "This object represents a bind id that is dynamically assigned to each bind by a NAT enabled device. Each bind is represented by a bind id that is unique across both, the natAddrBindTable and the natAddrPortBindTable."  
::= { natAddrBindEntry 5 }

natAddrBindTranslationEntity OBJECT-TYPE  
SYNTAX NatTranslationEntity  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION  
    "This object represents the direction of sessions for which this bind is applicable and the endpoint entity (source or destination) within the sessions that is subject to translation using the BIND.  
  
    Orientation of the bind can be a superset of translationEntity of the address map entry which forms the basis for this bind.  
  
    For example, if the translationEntity of an address map entry is outboundSrcEndPoint, the translationEntity of a bind derived from this map entry may either be outboundSrcEndPoint or it may be bidirectional (a bitmask of outboundSrcEndPoint and inboundDstEndPoint)."  
::= { natAddrBindEntry 6 }

natAddrBindType OBJECT-TYPE  
SYNTAX NatAssociationType  
MAX-ACCESS read-only

```
STATUS      deprecated
DESCRIPTION
    "This object indicates whether the bind is static or
    dynamic."
 ::= { natAddrBindEntry 7 }

natAddrBindMapIndex OBJECT-TYPE
SYNTAX      NatAddrMapId
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
    "This object is a pointer to the natAddrMapTable entry
    (and the parameters of that entry) which was used in
    creating this BIND.  This object, in conjunction with
    the ifIndex (which identifies a unique addrMapName)
    points to a unique entry in the natAddrMapTable."
 ::= { natAddrBindEntry 8 }

natAddrBindSessions OBJECT-TYPE
SYNTAX      Gauge32
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
    "Number of sessions currently using this BIND."
 ::= { natAddrBindEntry 9 }

natAddrBindMaxIdleTime OBJECT-TYPE
SYNTAX      TimeTicks
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
    "This object indicates the maximum time for
    which this bind can be idle with no sessions
    attached to it.

    The value of this object is of relevance only for
    dynamic NAT."
 ::= { natAddrBindEntry 10 }

natAddrBindCurrentIdleTime OBJECT-TYPE
SYNTAX      TimeTicks
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
    "At any given instance, this object indicates the
    time that this bind has been idle without any sessions
    attached to it."
```



```

        The value of this object is of relevance only for
        dynamic NAT."
 ::= { natAddrBindEntry 11 }

natAddrBindInTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of inbound packets that were successfully
        translated by using this bind entry.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrBindEntry 12 }

natAddrBindOutTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of outbound packets that were successfully
        translated using this bind entry.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrBindEntry 13 }

--
-- Address Port Bind section
--

natAddrPortBindNumberOfEntries OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "This object maintains a count of the number of entries
        that currently exist in the natAddrPortBindTable."
 ::= { natMIBObjects 7 }

--
-- The NAT Address Port Bind Table
--
```

```

natAddrPortBindTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NatAddrPortBindEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This table holds information about the currently
        active NAPT BINDs."
    ::= { natMIBObjects 8 }

natAddrPortBindEntry OBJECT-TYPE
    SYNTAX      NatAddrPortBindEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "Each entry in the this table holds information
        about a NAPT bind that is currently active.
        These entries are lost upon agent restart.

        This row has indexing which may create variables with
        more than 128 subidentifiers. Implementers of this
        table must be careful not to create entries which would
        result in OIDs that exceed the 128 subidentifier limit.
        Otherwise, the information cannot be accessed using
        SNMPv1, SNMPv2c or SNMPv3."
    INDEX      { ifIndex, natAddrPortBindLocalAddrType,
                  natAddrPortBindLocalAddr, natAddrPortBindLocalPort,
                  natAddrPortBindProtocol }
    ::= { natAddrPortBindTable 1 }

NatAddrPortBindEntry ::= SEQUENCE {
    natAddrPortBindLocalAddrType      InetAddressType,
    natAddrPortBindLocalAddr          InetAddress,
    natAddrPortBindLocalPort          InetPortNumber,
    natAddrPortBindProtocol            NatProtocolType,
    natAddrPortBindGlobalAddrType      InetAddressType,
    natAddrPortBindGlobalAddr          InetAddress,
    natAddrPortBindGlobalPort          InetPortNumber,
    natAddrPortBindId                  NatBindId,
    natAddrPortBindTranslationEntity    NatTranslationEntity,
    natAddrPortBindType                NatAssociationType,
    natAddrPortBindMapIndex             NatAddrMapId,
    natAddrPortBindSessions             Gauge32,
    natAddrPortBindMaxIdleTime          TimeTicks,
    natAddrPortBindCurrentIdleTime      TimeTicks,
    natAddrPortBindInTranslates         Counter64,
    natAddrPortBindOutTranslates        Counter64
}

```

**natAddrPortBindLocalAddrType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"This object specifies the address type used for  
natAddrPortBindLocalAddr."

::= { natAddrPortBindEntry 1 }

**natAddrPortBindLocalAddr OBJECT-TYPE**

SYNTAX InetAddress (SIZE(4|16))

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"This object represents the private-realm specific  
network layer address which, in conjunction with  
natAddrPortBindLocalPort, maps to the public-realm  
network layer address and transport id represented by  
natAddrPortBindGlobalAddr and natAddrPortBindGlobalPort  
respectively.

The type of this address is determined by the value of  
the natAddrPortBindLocalAddrType object."

::= { natAddrPortBindEntry 2 }

**natAddrPortBindLocalPort OBJECT-TYPE**

SYNTAX InetPortNumber

MAX-ACCESS not-accessible

STATUS deprecated

DESCRIPTION

"For a protocol value TCP or UDP, this object represents  
the private-realm specific port number. On the other  
hand, for ICMP a bind is created only for query/response  
type ICMP messages such as ICMP echo, Timestamp, and  
Information request messages, and this object represents  
the private-realm specific identifier in the ICMP  
message, as defined in RFC 792 for ICMPv4 and in RFC  
2463 for ICMPv6.

This object, together with natAddrPortBindProtocol,  
natAddrPortBindLocalAddrType, and  
natAddrPortBindLocalAddr, constitutes a session endpoint  
in the private realm. A bind entry binds a private  
realm specific endpoint to a public realm specific  
endpoint, as represented by the tuple of  
(natAddrPortBindGlobalPort, natAddrPortBindProtocol,  
natAddrPortBindGlobalAddrType, and

```
        natAddrPortBindGlobalAddr)."
 ::= { natAddrPortBindEntry 3 }

natAddrPortBindProtocol OBJECT-TYPE
    SYNTAX      NatProtocolType
    MAX-ACCESS   not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This object specifies a protocol identifier.  If the
         value of this object is none(1), then this bind entry
         applies to all IP traffic.  Any other value of this
         object specifies the class of IP traffic to which this
         BIND applies."
 ::= { natAddrPortBindEntry 4 }

natAddrPortBindGlobalAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   read-only
    STATUS      deprecated
    DESCRIPTION
        "This object specifies the address type used for
         natAddrPortBindGlobalAddr."
 ::= { natAddrPortBindEntry 5 }

natAddrPortBindGlobalAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   read-only
    STATUS      deprecated
    DESCRIPTION
        "This object represents the public-realm specific network
         layer address that, in conjunction with
         natAddrPortBindGlobalPort, maps to the private-realm

         network layer address and transport id represented by
         natAddrPortBindLocalAddr and natAddrPortBindLocalPort,
         respectively.

         The type of this address is determined by the value of
         the natAddrPortBindGlobalAddrType object."
 ::= { natAddrPortBindEntry 6 }

natAddrPortBindGlobalPort OBJECT-TYPE
    SYNTAX      InetPortNumber
    MAX-ACCESS   read-only
    STATUS      deprecated
    DESCRIPTION
        "For a protocol value TCP or UDP, this object represents
         the public-realm specific port number.  On the other
```

hand, for ICMP a bind is created only for query/response type ICMP messages such as ICMP echo, Timestamp, and Information request messages, and this object represents the public-realm specific identifier in the ICMP message, as defined in RFC 792 for ICMPv4 and in RFC 2463 for ICMPv6.

This object, together with natAddrPortBindProtocol, natAddrPortBindGlobalAddrType, and natAddrPortBindGlobalAddr, constitutes a session endpoint in the public realm. A bind entry binds a public realm specific endpoint to a private realm specific endpoint, as represented by the tuple of (natAddrPortBindLocalPort, natAddrPortBindProtocol, natAddrPortBindLocalAddrType, and natAddrPortBindLocalAddr)."

::= { natAddrPortBindEntry 7 }

natAddrPortBindId OBJECT-TYPE

SYNTAX NatBindId

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"This object represents a bind id that is dynamically assigned to each bind by a NAT enabled device. Each bind is represented by a unique bind id across both the natAddrBindTable and the natAddrPortBindTable."

::= { natAddrPortBindEntry 8 }

natAddrPortBindTranslationEntity OBJECT-TYPE

SYNTAX NatTranslationEntity

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"This object represents the direction of sessions for which this bind is applicable and the entity (source or destination) within the sessions that is subject to translation with the BIND.

Orientation of the bind can be a superset of the translationEntity of the address map entry that forms the basis for this bind.

For example, if the translationEntity of an address map entry is outboundSrcEndPoint, the translationEntity of a bind derived from this map entry may either be outboundSrcEndPoint or may be bidirectional (a bitmask of

```
        outboundSrcEndPoint and inboundDstEndPoint)."  
 ::= { natAddrPortBindEntry 9 }  
  
natAddrPortBindType OBJECT-TYPE  
    SYNTAX      NatAssociationType  
    MAX-ACCESS  read-only  
    STATUS      deprecated  
    DESCRIPTION  
        "This object indicates whether the bind is static or  
        dynamic."  
 ::= { natAddrPortBindEntry 10 }  
  
natAddrPortBindMapIndex OBJECT-TYPE  
    SYNTAX      NatAddrMapId  
    MAX-ACCESS  read-only  
    STATUS      deprecated  
    DESCRIPTION  
        "This object is a pointer to the natAddrMapTable entry  
        (and the parameters of that entry) used in  
        creating this BIND. This object, in conjunction with  
        the ifIndex (which identifies a unique addrMapName),  
        points to a unique entry in the natAddrMapTable."  
 ::= { natAddrPortBindEntry 11 }  
  
natAddrPortBindSessions OBJECT-TYPE  
    SYNTAX      Gauge32  
    MAX-ACCESS  read-only  
    STATUS      deprecated  
    DESCRIPTION  
        "Number of sessions currently using this BIND."  
 ::= { natAddrPortBindEntry 12 }  
  
natAddrPortBindMaxIdleTime OBJECT-TYPE  
    SYNTAX      TimeTicks  
    MAX-ACCESS  read-only  
    STATUS      deprecated  
  
    DESCRIPTION  
        "This object indicates the maximum time for  
        which this bind can be idle without any sessions  
        attached to it.  
        The value of this object is of relevance  
        only for dynamic NAT."  
 ::= { natAddrPortBindEntry 13 }  
  
natAddrPortBindCurrentIdleTime OBJECT-TYPE  
    SYNTAX      TimeTicks  
    MAX-ACCESS  read-only
```

```
STATUS      deprecated
DESCRIPTION
    "At any given instance, this object indicates the
    time that this bind has been idle without any sessions
    attached to it.

    The value of this object is of relevance
    only for dynamic NAT."
 ::= { natAddrPortBindEntry 14 }

natAddrPortBindInTranslates OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
    "The number of inbound packets that were translated as
    per this bind entry.

    Discontinuities in the value of this counter can occur
    at reinitialization of the management system and at
    other times, as indicated by the value of
    ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrPortBindEntry 15 }

natAddrPortBindOutTranslates OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
    "The number of outbound packets that were translated as
    per this bind entry.

    Discontinuities in the value of this counter can occur
    at reinitialization of the management system and at
    other times, as indicated by the value of
    ifCounterDiscontinuityTime on the relevant interface."
 ::= { natAddrPortBindEntry 16 }

--
-- The Session Table
--

natSessionTable OBJECT-TYPE
SYNTAX      SEQUENCE OF NatSessionEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION
    "The (conceptual) table containing one entry for each
```

```

        NAT session currently active on this NAT device."
 ::= { natMIBObjects 9 }

natSessionEntry OBJECT-TYPE
    SYNTAX      NatSessionEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "An entry (conceptual row) containing information
         about an active NAT session on this NAT device.
         These entries are lost upon agent restart."
    INDEX       { ifIndex, natSessionIndex }
 ::= { natSessionTable 1 }

NatSessionEntry ::= SEQUENCE {
    natSessionIndex                      NatSessionId,
    natSessionPrivateSrcEPBindId        NatBindIdOrZero,
    natSessionPrivateSrcEPBindMode      NatBindMode,
    natSessionPrivateDstEPBindId        NatBindIdOrZero,
    natSessionPrivateDstEPBindMode      NatBindMode,
    natSessionDirection                 INTEGER,
    natSessionUpTime                    TimeTicks,
    natSessionAddrMapIndex              NatAddrMapId,
    natSessionProtocolType              NatProtocolType,
    natSessionPrivateAddrType           InetAddressType,
    natSessionPrivateSrcAddr            InetAddress,
    natSessionPrivateSrcPort            InetPortNumber,
    natSessionPrivateDstAddr            InetAddress,
    natSessionPrivateDstPort            InetPortNumber,
    natSessionPublicAddrType            InetAddressType,
    natSessionPublicSrcAddr             InetAddress,
    natSessionPublicSrcPort             InetPortNumber,
    natSessionPublicDstAddr             InetAddress,
    natSessionPublicDstPort             InetPortNumber,
    natSessionMaxIdleTime               TimeTicks,
    natSessionCurrentIdleTime           TimeTicks,
    natSessionInTranslates              Counter64,
    natSessionOutTranslates             Counter64
}

natSessionIndex OBJECT-TYPE
    SYNTAX      NatSessionId
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "The session ID for this NAT session."
 ::= { natSessionEntry 1 }

```



```
natSessionPrivateSrcEPBindId OBJECT-TYPE
    SYNTAX      NatBindIdOrZero
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The bind id associated between private and public
        source end points.  In the case of Symmetric-NAT,
        this should be set to zero."
    ::= { natSessionEntry 2 }

natSessionPrivateSrcEPBindMode OBJECT-TYPE
    SYNTAX      NatBindMode
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "This object indicates whether the bind indicated
        by the object natSessionPrivateSrcEPBindId
        is an address bind or an address port bind."
    ::= { natSessionEntry 3 }

natSessionPrivateDstEPBindId OBJECT-TYPE
    SYNTAX      NatBindIdOrZero
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The bind id associated between private and public
        destination end points."
    ::= { natSessionEntry 4 }

natSessionPrivateDstEPBindMode OBJECT-TYPE
    SYNTAX      NatBindMode
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "This object indicates whether the bind indicated
        by the object natSessionPrivateDstEPBindId
        is an address bind or an address port bind."
    ::= { natSessionEntry 5 }

natSessionDirection OBJECT-TYPE
    SYNTAX      INTEGER {
                    inbound (1),
                    outbound (2)
                }

    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
```

```
        "The direction of this session with respect to the
        local network. 'inbound' indicates that this session
        was initiated from the public network into the private
        network. 'outbound' indicates that this session was
        initiated from the private network into the public
        network."
 ::= { natSessionEntry 6 }

natSessionUpTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS   read-only
    STATUS       deprecated
    DESCRIPTION
        "The up time of this session in one-hundredths of a
        second."
 ::= { natSessionEntry 7 }

natSessionAddrMapIndex OBJECT-TYPE
    SYNTAX      NatAddrMapId
    MAX-ACCESS   read-only
    STATUS       deprecated
    DESCRIPTION
        "This object is a pointer to the natAddrMapTable entry
        (and the parameters of that entry) used in
        creating this session. This object, in conjunction with
        the ifIndex (which identifies a unique addrMapName),
        points to a unique entry in the natAddrMapTable."
 ::= { natSessionEntry 8 }

natSessionProtocolType OBJECT-TYPE
    SYNTAX      NatProtocolType
    MAX-ACCESS   read-only
    STATUS       deprecated
    DESCRIPTION
        "The protocol type of this session."
 ::= { natSessionEntry 9 }

natSessionPrivateAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   read-only
    STATUS       deprecated
    DESCRIPTION
        "This object specifies the address type used for
        natSessionPrivateSrcAddr and natSessionPrivateDstAddr."
 ::= { natSessionEntry 10 }

natSessionPrivateSrcAddr OBJECT-TYPE
    SYNTAX      InetAddress
```

MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION

"The source IP address of the session endpoint that lies in the private network.

The value of this object must be zero only when the natSessionPrivateSrcEPBindId object has a zero value. When the value of this object is zero, the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPrivateAddrType object."

::= { natSessionEntry 11 }

natSessionPrivateSrcPort OBJECT-TYPE

SYNTAX InetPortNumber  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the source port in the first packet of session while in private-realm. On the other hand, when the protocol is ICMP, a NAT session is created only for query/response type ICMP messages such as ICMP echo, Timestamp, and Information request messages, and this object represents the private-realm specific identifier in the ICMP message, as defined in RFC 792 for ICMPv4 and in RFC 2463 for ICMPv6.

The value of this object must be zero when the natSessionPrivateSrcEPBindId object has zero value and value of natSessionPrivateSrcEPBindMode is addressPortBind(2). In such a case, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort, or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 12 }

natSessionPrivateDstAddr OBJECT-TYPE

SYNTAX InetAddress  
MAX-ACCESS read-only  
STATUS deprecated  
DESCRIPTION

"The destination IP address of the session endpoint that

lies in the private network.

The value of this object must be zero when the natSessionPrivateDstEPBindId object has a zero value. In such a scenario, the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPrivateAddrType object."

::= { natSessionEntry 13 }

natSessionPrivateDstPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the destination port in the first packet of session while in private-realm. On the other hand, when the protocol is ICMP, this object is not relevant and should be set to zero.

The value of this object must be zero when the natSessionPrivateDstEPBindId object has a zero value and natSessionPrivateDstEPBindMode is set to addressPortBind(2). In such a case, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort, or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 14 }

natSessionPublicAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"This object specifies the address type used for natSessionPublicSrcAddr and natSessionPublicDstAddr."

::= { natSessionEntry 15 }

natSessionPublicSrcAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The source IP address of the session endpoint that lies in the public network.

The value of this object must be zero when the natSessionPrivateSrcEPBindId object has a zero value. In such a scenario, the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPublicAddrType object."

::= { natSessionEntry 16 }

natSessionPublicSrcPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the source port in the first packet of session while in public-realm. On the other hand, when protocol is ICMP, a NAT session is created only for query/response type ICMP messages such as ICMP echo, Timestamp, and Information request messages, and this object represents the public-realm specific identifier in the ICMP message, as defined in RFC 792 for ICMPv4 and in RFC 2463 for ICMPv6.

The value of this object must be zero when the natSessionPrivateSrcEPBindId object has a zero value and natSessionPrivateSrcEPBindMode is set to addressPortBind(2). In such a scenario, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 17 }

natSessionPublicDstAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The destination IP address of the session endpoint that lies in the public network.

The value of this object must be non-zero when the natSessionPrivateDstePBindId object has a non-zero value. If the value of this object and the corresponding natSessionPrivateDstePBindId object value is zero, then the NAT session lookup will match any IP address to this field.

The type of this address is determined by the value of the natSessionPublicAddrType object."

::= { natSessionEntry 18 }

natSessionPublicDstPort OBJECT-TYPE

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"When the value of protocol is TCP or UDP, this object represents the destination port in the first packet of session while in public-realm. On the other hand, when the protocol is ICMP, this object is not relevant for translation and should be zero.

The value of this object must be zero when the natSessionPrivateDstePBindId object has a zero value and natSessionPrivateDstePBindMode is addressPortBind(2). In such a scenario, the NAT session lookup will match any port number to this field.

The value of this object must be zero when the object is not a representative field (SrcPort, DstPort, or ICMP identifier) of the session tuple in either the public realm or the private realm."

::= { natSessionEntry 19 }

natSessionMaxIdleTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

"The max time for which this session can be idle without detecting a packet."

::= { natSessionEntry 20 }

natSessionCurrentIdleTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS deprecated

```
DESCRIPTION
    "The time since a packet belonging to this session was
    last detected."
 ::= { natSessionEntry 21 }

natSessionInTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of inbound packets that were translated for
        this session.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natSessionEntry 22 }

natSessionOutTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of outbound packets that were translated for
        this session.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natSessionEntry 23 }

--
-- The Protocol table
--

natProtocolTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NatProtocolEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "The (conceptual) table containing per protocol NAT
        statistics."
 ::= { natMIBObjects 10 }

natProtocolEntry OBJECT-TYPE
    SYNTAX      NatProtocolEntry
```

```
MAX-ACCESS not-accessible
STATUS      deprecated
DESCRIPTION
    "An entry (conceptual row) containing NAT statistics
    pertaining to a particular protocol."
INDEX       { natProtocol }
 ::= { natProtocolTable 1 }

NatProtocolEntry ::= SEQUENCE {
    natProtocol                NatProtocolType,
    natProtocolInTranslates    Counter64,
    natProtocolOutTranslates    Counter64,
    natProtocolDiscards        Counter64
}

natProtocol      OBJECT-TYPE
    SYNTAX      NatProtocolType
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This object represents the protocol pertaining to which
        parameters are reported."
    ::= { natProtocolEntry 1 }

natProtocolInTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of inbound packets pertaining to the protocol
        identified by natProtocol that underwent NAT.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
    ::= { natProtocolEntry 2 }

natProtocolOutTranslates OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of outbound packets pertaining to the
        protocol identified by natProtocol that underwent NAT.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
```



```

        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natProtocolEntry 3 }

natProtocolDiscards OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "The number of packets pertaining to the protocol
        identified by natProtocol that had to be
        rejected/dropped due to lack of resources.  These
        rejections could be due to session timeout, resource
        unavailability, lack of address space, etc.

        Discontinuities in the value of this counter can occur
        at reinitialization of the management system and at
        other times, as indicated by the value of
        ifCounterDiscontinuityTime on the relevant interface."
 ::= { natProtocolEntry 4 }

--
-- Notifications section
--

natMIBNotifications OBJECT IDENTIFIER ::= { natMIB 0 }

--
-- Notifications
--

natPacketDiscard NOTIFICATION-TYPE
    OBJECTS { ifIndex }
    STATUS deprecated
    DESCRIPTION
        "This notification is generated when IP packets are
        discarded by the NAT function; e.g., due to lack of
        mapping space when NAT is out of addresses or ports.

        Note that the generation of natPacketDiscard
        notifications is throttled by the agent, as specified
        by the 'natNotifThrottlingInterval' object."
 ::= { natMIBNotifications 1 }

--
-- Conformance information.
```

```
--

natMIBConformance OBJECT IDENTIFIER ::= { natMIB 2 }

natMIBGroups      OBJECT IDENTIFIER ::= { natMIBConformance 1 }
natMIBCompliances OBJECT IDENTIFIER ::= { natMIBConformance 2 }

--
-- Units of conformance
--

natConfigGroup OBJECT-GROUP
    OBJECTS { natInterfaceRealm,
               natInterfaceServiceType,
               natInterfaceStorageType,
               natInterfaceRowStatus,
               natAddrMapName,
               natAddrMapEntryType,
               natAddrMapTranslationEntity,
               natAddrMapLocalAddrType,
               natAddrMapLocalAddrFrom,
               natAddrMapLocalAddrTo,
               natAddrMapLocalPortFrom,
               natAddrMapLocalPortTo,
               natAddrMapGlobalAddrType,
               natAddrMapGlobalAddrFrom,
               natAddrMapGlobalAddrTo,
               natAddrMapGlobalPortFrom,
               natAddrMapGlobalPortTo,
               natAddrMapProtocol,
               natAddrMapStorageType,
               natAddrMapRowStatus,
               natBindDefIdleTimeout,
               natUdpDefIdleTimeout,
               natIcmpDefIdleTimeout,
               natOtherDefIdleTimeout,
               natTcpDefIdleTimeout,
               natTcpDefNegTimeout,
               natNotifThrottlingInterval }
    STATUS deprecated
    DESCRIPTION
        "A collection of configuration-related information
         required to support management of devices supporting
         NAT."
    ::= { natMIBGroups 1 }

natTranslationGroup OBJECT-GROUP
    OBJECTS { natAddrBindNumberOfEntries,
```

```
natAddrBindGlobalAddrType,
natAddrBindGlobalAddr,
natAddrBindId,
natAddrBindTranslationEntity,
natAddrBindType,
natAddrBindMapIndex,
natAddrBindSessions,
natAddrBindMaxIdleTime,
natAddrBindCurrentIdleTime,
natAddrBindInTranslates,
natAddrBindOutTranslates,
natAddrPortBindNumberOfEntries,
natAddrPortBindGlobalAddrType,
natAddrPortBindGlobalAddr,
natAddrPortBindGlobalPort,
natAddrPortBindId,
natAddrPortBindTranslationEntity,
natAddrPortBindType,
natAddrPortBindMapIndex,
natAddrPortBindSessions,
natAddrPortBindMaxIdleTime,
natAddrPortBindCurrentIdleTime,
natAddrPortBindInTranslates,
natAddrPortBindOutTranslates,
natSessionPrivateSrcEPBindId,
natSessionPrivateSrcEPBindMode,
natSessionPrivateDstEPBindId,
natSessionPrivateDstEPBindMode,
natSessionDirection,
natSessionUpTime,
natSessionAddrMapIndex,
natSessionProtocolType,
natSessionPrivateAddrType,
natSessionPrivateSrcAddr,
natSessionPrivateSrcPort,
natSessionPrivateDstAddr,
natSessionPrivateDstPort,
natSessionPublicAddrType,
natSessionPublicSrcAddr,
natSessionPublicSrcPort,
natSessionPublicDstAddr,
natSessionPublicDstPort,
natSessionMaxIdleTime,
natSessionCurrentIdleTime,
natSessionInTranslates,
natSessionOutTranslates }
STATUS deprecated
```

```
DESCRIPTION
    "A collection of BIND-related objects required to support
    management of devices supporting NAT."
 ::= { natMIBGroups 2 }

natStatsInterfaceGroup OBJECT-GROUP
    OBJECTS { natInterfaceInTranslates,
               natInterfaceOutTranslates,
               natInterfaceDiscards }
    STATUS deprecated
    DESCRIPTION
        "A collection of NAT statistics associated with the
        interface on which NAT is configured, to aid
        troubleshooting/monitoring of the NAT operation."
 ::= { natMIBGroups 3 }

natStatsProtocolGroup OBJECT-GROUP
    OBJECTS { natProtocolInTranslates,
               natProtocolOutTranslates,
               natProtocolDiscards }
    STATUS deprecated
    DESCRIPTION
        "A collection of protocol specific NAT statistics,
        to aid troubleshooting/monitoring of NAT operation."
 ::= { natMIBGroups 4 }

natStatsAddrMapGroup OBJECT-GROUP
    OBJECTS { natAddrMapInTranslates,
               natAddrMapOutTranslates,
               natAddrMapDiscards,
               natAddrMapAddrUsed }
    STATUS deprecated
    DESCRIPTION
        "A collection of address map specific NAT statistics,
        to aid troubleshooting/monitoring of NAT operation."
 ::= { natMIBGroups 5 }

natMIBNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS { natPacketDiscard }
    STATUS deprecated
    DESCRIPTION
        "A collection of notifications generated by
        devices supporting this MIB."
 ::= { natMIBGroups 6 }

--
-- Compliance statements
```

--

natMIBFullCompliance MODULE-COMPLIANCE

STATUS deprecated

DESCRIPTION

"When this MIB is implemented with support for read-create, then such an implementation can claim full compliance. Such devices can then be both monitored and configured with this MIB.

The following index objects cannot be added as OBJECT clauses but nevertheless have the compliance requirements:

"

-- OBJECT natAddrBindLocalAddrType  
-- SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
-- DESCRIPTION  
-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

-- OBJECT natAddrBindLocalAddr  
-- SYNTAX InetAddress (SIZE(4|16))  
-- DESCRIPTION  
-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

-- OBJECT natAddrPortBindLocalAddrType  
-- SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
-- DESCRIPTION  
-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

-- OBJECT natAddrPortBindLocalAddr  
-- SYNTAX InetAddress (SIZE(4|16))  
-- DESCRIPTION  
-- "An implementation is required to support  
-- global IPv4 and/or IPv6 addresses, depending  
-- on its support for IPv4 and IPv6."

MODULE IF-MIB -- The interfaces MIB, RFC2863

MANDATORY-GROUPS {  
    ifCounterDiscontinuityGroup  
}

MODULE -- this module

```
MANDATORY-GROUPS { natConfigGroup, natTranslationGroup,
                    natStatsInterfaceGroup }

GROUP      natStatsProtocolGroup
DESCRIPTION
    "This group is optional."
GROUP      natStatsAddrMapGroup
DESCRIPTION
    "This group is optional."
GROUP      natMIBNotificationGroup
DESCRIPTION
    "This group is optional."

OBJECT natAddrMapLocalAddrType
SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
DESCRIPTION
    "An implementation is required to support global IPv4
    and/or IPv6 addresses, depending on its support
    for IPv4 and IPv6."

OBJECT natAddrMapLocalAddrFrom
SYNTAX  InetAddress (SIZE(4|16))
DESCRIPTION
    "An implementation is required to support global IPv4
    and/or IPv6 addresses, depending on its support
    for IPv4 and IPv6."

OBJECT natAddrMapLocalAddrTo
SYNTAX  InetAddress (SIZE(4|16))
DESCRIPTION
    "An implementation is required to support global IPv4
    and/or IPv6 addresses, depending on its support
    for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrType
SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
DESCRIPTION
    "An implementation is required to support global IPv4
    and/or IPv6 addresses, depending on its support
    for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrFrom
SYNTAX  InetAddress (SIZE(4|16))
DESCRIPTION
    "An implementation is required to support global IPv4
    and/or IPv6 addresses, depending on its support
    for IPv4 and IPv6."
```

OBJECT natAddrMapGlobalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natAddrBindGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natAddrBindGlobalAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natSessionPrivateAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natSessionPrivateSrcAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support  
    for IPv4 and IPv6."

OBJECT natSessionPrivateDstAddr  
 SYNTAX InetAddress (SIZE(4|16))  
 DESCRIPTION  
 "An implementation is required to support global IPv4  
 and/or IPv6 addresses, depending on its support  
 for IPv4 and IPv6."

OBJECT natSessionPublicAddrType  
 SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
 DESCRIPTION  
 "An implementation is required to support global IPv4  
 and/or IPv6 addresses, depending on its support  
 for IPv4 and IPv6."

OBJECT natSessionPublicSrcAddr  
 SYNTAX InetAddress (SIZE(4|16))  
 DESCRIPTION  
 "An implementation is required to support global IPv4  
 and/or IPv6 addresses, depending on its support  
 for IPv4 and IPv6."

OBJECT natSessionPublicDstAddr  
 SYNTAX InetAddress (SIZE(4|16))  
 DESCRIPTION  
 "An implementation is required to support global IPv4  
 and/or IPv6 addresses, depending on its support  
 for IPv4 and IPv6."

::= { natMIBCompliances 1 }

natMIBReadOnlyCompliance MODULE-COMPLIANCE

STATUS deprecated

DESCRIPTION

"When this MIB is implemented without support for  
 read-create (i.e., in read-only mode), then such an  
 implementation can claim read-only compliance.  
 Such a device can then be monitored but cannot be  
 configured with this MIB."

The following index objects cannot be added as OBJECT  
 clauses but nevertheless have the compliance  
 requirements:

"  
 -- OBJECT natAddrBindLocalAddrType  
 -- SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
 -- DESCRIPTION  
 -- "An implementation is required to support  
 -- global IPv4 and/or IPv6 addresses, depending



```
--          on its support for IPv4 and IPv6."

-- OBJECT  natAddrBindLocalAddr
-- SYNTAX  InetAddress (SIZE(4|16))

-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

-- OBJECT  natAddrPortBindLocalAddrType
-- SYNTAX  InetAddressType { ipv4(1), ipv6(2) }
-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."
-- OBJECT  natAddrPortBindLocalAddr
-- SYNTAX  InetAddress (SIZE(4|16))
-- DESCRIPTION
--          "An implementation is required to support
--          global IPv4 and/or IPv6 addresses, depending
--          on its support for IPv4 and IPv6."

MODULE IF-MIB -- The interfaces MIB, RFC2863
    MANDATORY-GROUPS {
        ifCounterDiscontinuityGroup
    }

MODULE -- this module
    MANDATORY-GROUPS { natConfigGroup, natTranslationGroup,
        natStatsInterfaceGroup }

    GROUP          natStatsProtocolGroup
    DESCRIPTION
        "This group is optional."
    GROUP          natStatsAddrMapGroup
    DESCRIPTION
        "This group is optional."
    GROUP          natMIBNotificationGroup
    DESCRIPTION
        "This group is optional."
    OBJECT natInterfaceRowStatus
    SYNTAX RowStatus { active(1) }
    MIN-ACCESS    read-only
    DESCRIPTION
        "Write access is not required, and active is the only
        status that needs to be supported."
```

OBJECT natAddrMapLocalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapLocalAddrFrom  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapLocalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrFrom  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapGlobalAddrTo  
SYNTAX InetAddress (SIZE(4|16))  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required. An implementation is required to support global IPv4 and/or IPv6 addresses, depending on its support for IPv4 and IPv6."

OBJECT natAddrMapRowStatus  
SYNTAX RowStatus { active(1) }  
MIN-ACCESS read-only  
DESCRIPTION  
    "Write access is not required, and active is the only  
    status that needs to be supported."

OBJECT natAddrBindGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natAddrBindGlobalAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natAddrPortBindGlobalAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natSessionPrivateAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natSessionPrivateSrcAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natSessionPrivateDstAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natSessionPublicAddrType  
SYNTAX InetAddressType { ipv4(1), ipv6(2) }  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natSessionPublicSrcAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

OBJECT natSessionPublicDstAddr  
SYNTAX InetAddress (SIZE(4|16))  
DESCRIPTION  
    "An implementation is required to support global IPv4  
    and/or IPv6 addresses, depending on its support for  
    IPv4 and IPv6."

::= { natMIBCompliances 2 }

-----  
-- END OF DEPRECATED OBJECTS. CURRENT OBJECTS FOLLOW.

-- textual conventions

ProtocolNumber ::= TEXTUAL-CONVENTION  
    DISPLAY-HINT "d"  
    STATUS current  
    DESCRIPTION  
        "A transport protocol number, from the 'protocol-numbers'  
        IANA registry."  
    SYNTAX Unsigned32 (0..255)

NatPoolId ::= TEXTUAL-CONVENTION  
    DISPLAY-HINT "d"  
    STATUS current

## DESCRIPTION

"A unique ID that is assigned to each pool."

SYNTAX Unsigned32 (1..4294967295)

NatBehaviorType ::= TEXTUAL-CONVENTION

STATUS current

## DESCRIPTION

"Behavior type as described in [RFC4787] sections 4.1 and 5."

SYNTAX INTEGER {

endpointIndependent (0),

addressDependent (1),

addressAndPortDependent (2)

}

NatPoolingType ::= TEXTUAL-CONVENTION

STATUS current

## DESCRIPTION

"Pooling type as described in [RFC4787] sections 4.1."

SYNTAX INTEGER {

arbitrary (0),

paired (1)

}

VlanIndexOrZero ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

## DESCRIPTION

"A value used to index per-VLAN tables: a value of 4095 is not permitted. A value of 0 indicates no index is present. If the value is between 1 and 4094 inclusive, it represents an IEEE 802.1Q VLAN-ID with global scope within a given bridged domain (see VlanId textual convention in [RFC4363]). If the value is greater than 4095, then it represents a VLAN with scope local to the particular agent, i.e., one without a global VLAN-ID assigned to it. Such VLANs are outside the scope of IEEE 802.1Q, but it is convenient to be able to manage them in the same way using this MIB."

SYNTAX Unsigned32

SubscriberIndex ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

## DESCRIPTION

"A unique ID that is assigned to each subscriber."

SYNTAX Unsigned32 (1..4294967295)

SubscriberIdentifierType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Type of additional classifying information used by the NAT to identify the subscriber from an incoming packet, when the packet source address is not sufficient to do so unambiguously.

null(0)

No additional information is needed.

interfaces(1)

A set of one or more ingress interface indexes specified by the [RFC2863] InterfaceIndex textual convention.

vlan(2)

An ingress VLAN index using the VlanIndexOrZero textual convention, which is the [RFC4363] VlanIndex textual convention modified for local use in this MIB.

vpn(3)

An ingress layer 3 VPN identifier using the [RFC4265] VPNIdOrZero textual convention.

ipencaps(4)

Incoming source address of an encapsulating IPv4 or IPv6 tunnel (e.g., IPv6 as used in DS-Lite, [RFC6333]) as defined by the InetAddressType and InetAddress textual conventions.

other(5)

The implementation supports other classifiers and/or combinations of classifier types. In the latter case the implementation MUST specify the semantics of the combination ('OR' or 'AND')."

SYNTAX INTEGER {  
    null(0),  
    interfaces(1),  
    vlan(2),  
    vpn(3),  
    ipencaps(4),  
    other(5)

```
    }

SubsInterfaceIdRowIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "A unique ID that is assigned to each row in the
        natSubsInterfaceIdentifierTable."
    SYNTAX Unsigned32 (1..4294967295)

-- notifications

natNotifPoolWatermarkLow NOTIFICATION-TYPE
    OBJECTS { natPoolWatermarkLow }
    STATUS current
    DESCRIPTION
        "This notification is generated when a pool's usage
        percentage becomes lower than or equal to the specified
        threshold. The threshold is specified by the
        natPoolWatermarkLow object"
    ::= { natMIBNotifications 2 }

natNotifPoolWatermarkHigh NOTIFICATION-TYPE
    OBJECTS { natPoolWatermarkHigh }
    STATUS current
    DESCRIPTION
        "This notification is generated when a pool's usage
        percentage becomes greater than or equal to the specified
        threshold. The threshold is specified by the
        natPoolWatermarkHigh object"
    ::= { natMIBNotifications 3 }

natNotifMappings NOTIFICATION-TYPE
    OBJECTS { natMappingCreations, natMappingRemovals }
    STATUS current
    DESCRIPTION
        "This notification is generated when the number of active
        mappings exceeds the value of natMappingsNotifyThreshold."
    ::= { natMIBNotifications 4 }

natNotifAddrMappings NOTIFICATION-TYPE
    OBJECTS { natAddressMappingCreations, natAddressMappingRemovals }
    STATUS current
    DESCRIPTION
        "This notification is generated when the number of active
        address mappings exceeds the value of
        natAddrMapNotifyThreshold."
```

```
 ::= { natMIBNotifications 5 }

natNotifSubscriberMappings NOTIFICATION-TYPE
  OBJECTS { natSubscriberMappingCreations,
             natSubscriberMappingRemovals }
  STATUS current
  DESCRIPTION
    "This notification is generated when the number of active
     mappings exceeds the value of natSubscriberMapNotifyThresh,
     unless natSubscriberMapNotifyThresh is zero.."
  ::= { natMIBNotifications 6 }

-- instance table

natInstanceTable OBJECT-TYPE
  SYNTAX SEQUENCE OF NatInstanceEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Table of NAT instances."
  ::= { natMIBObjects 11 }

natInstanceEntry OBJECT-TYPE
  SYNTAX NatInstanceEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Objects related to a single NAT instance."
  INDEX { natInstanceIndex }
  ::= { natInstanceTable 1 }

NatInstanceEntry ::=
  SEQUENCE {
    natInstanceIndex Unsigned32,
    natInstanceAlias DisplayString
  }

natInstanceIndex OBJECT-TYPE
  SYNTAX Unsigned32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "NAT instance index. Semantics of this number are
     implementation-specific. This object is used as an index for
     many tables defined below."
  ::= { natInstanceEntry 1 }
```



## natInstanceAlias OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..64))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is an 'alias' name for the NAT instance as specified by a network manager, and provides a non-volatile 'handle' for the instance.

On the first instantiation of a NAT instance, the value of natInstanceAlias associated with that instance is the zero-length string. As and when a value is written into an instance of natInstanceAlias through a network management set operation, then the agent must retain the supplied value in this object instance associated with the same interface for as long as that NAT instance remains instantiated, including across all re-initializations/reboots of the network management system, including those which result in a change of the interface's natInstanceIndex value.

An example of the value which a network manager might store in this object for a NAT instance is the name/identifier of the interface that brings in internal traffic for this NAT instance or the name of the VRF for internal traffic.

An agent may choose to provide read-only access if the agent itself assigns an identifier for the NAT instance. An agent which supports write access to this object is required to keep the value in non-volatile storage, but it may limit the length of new values depending on how much storage is already occupied by the current values for other NAT instances."

```
::= { natInstanceEntry 2 }
```

```
-- counters
```

```
natCounters OBJECT IDENTIFIER ::= { natMIBObjects 12 }
```

## natCountersTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatCountersEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of counters of a NAT instance. The counters are global across L4 protocols."

```
::= { natCounters 1 }
```

```
natCountersEntry OBJECT-TYPE
    SYNTAX NatCountersEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Counters related to a single NAT instance."
    INDEX { natInstanceIndex }
    ::= { natCountersTable 1 }

NatCountersEntry ::=
    SEQUENCE {
        natTranslations                Counter64,
        natOutOfPortErrors             Counter64,
        natResourceErrors              Counter64,
        natQuotaDrops                  Counter64,
        natMappingCreations             Counter64,
        natMappingRemovals             Counter64,
        natAddressMappingCreations     Counter64,
        natAddressMappingRemovals      Counter64
    }

natTranslations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets translated."
    ::= { natCountersEntry 1 }

natOutOfPortErrors OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets not translated because no external
        port was available, excluding quota limitations."
    ::= { natCountersEntry 2 }

natResourceErrors OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets not translated because of resource
        constraints (excluding out-of-ports error and quota drops)."
    ::= { natCountersEntry 3 }

natQuotaDrops OBJECT-TYPE
```

```
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The number of incoming packets not translated because of
    quota limitations. Quotas include absolute limits as well
    as limits on rate of allocation."
 ::= { natCountersEntry 4 }

natMappingCreations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of mapping creations. This includes static mappings."
    ::= { natCountersEntry 5 }

natMappingRemovals OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of mapping removals. This includes static mappings."
    ::= { natCountersEntry 6 }

natAddressMappingCreations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of address mapping creations. This includes static
        mappings."
    ::= { natCountersEntry 7 }

natAddressMappingRemovals OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of address mapping removals. This includes static
        mappings.

        The number of active mappings is equal to
        natAddressMappingCreations - natAddressMappingRemovals."
    ::= { natCountersEntry 8 }

natL4ProtocolTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NatL4ProtocolEntry
```

```
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Table of protocols with per-protocol counters."
 ::= { natCounters 2 }

natL4ProtocolEntry OBJECT-TYPE
    SYNTAX NatL4ProtocolEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Per-protocol counters."
    INDEX { natInstanceIndex, natL4ProtocolNumber }
    ::= { natL4ProtocolTable 1 }

NatL4ProtocolEntry ::=
    SEQUENCE {
        natL4ProtocolNumber          ProtocolNumber,
        natL4ProtocolTranslations    Counter64,
        natL4ProtocolOutOfPortErrors Counter64,
        natL4ProtocolResourceErrors  Counter64,
        natL4ProtocolQuotaDrops      Counter64,
        natL4ProtocolMappingCreations Counter64,
        natL4ProtocolMappingRemovals Counter64
    }

natL4ProtocolNumber OBJECT-TYPE
    SYNTAX ProtocolNumber
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Counters in this conceptual row apply to packets using the
        transport protocol identified by this object's value."
    ::= { natL4ProtocolEntry 1 }

natL4ProtocolTranslations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets translated."
    ::= { natL4ProtocolEntry 2 }

natL4ProtocolOutOfPortErrors OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "The number of packets not translated because no external
        port was available."
 ::= { natL4ProtocolEntry 3 }

natL4ProtocolResourceErrors OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets not translated because of resource
        constraints (excluding out-of-ports errors and quota
        drops)."
```

```
 ::= { natL4ProtocolEntry 4 }

natL4ProtocolQuotaDrops OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of incoming packets not translated because of
        exceeded quotas. Quotas include absolute limits as well as
        limits on rate of allocation."
```

```
 ::= { natL4ProtocolEntry 5 }

natL4ProtocolMappingCreations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of mapping creations. This includes static mappings."
```

```
 ::= { natL4ProtocolEntry 6 }

natL4ProtocolMappingRemovals OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of mapping removals. This includes static mappings.

        The number of active mappings is equal to
        natL4ProtocolMappingCreations -
        natL4ProtocolMappingRemovals."
```

```
 ::= { natL4ProtocolEntry 7 }

-- limits

natLimitsTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF NatLimitsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Table of limits for a NAT instance."
 ::= { natMIBObjects 13 }

natLimitsEntry OBJECT-TYPE
    SYNTAX NatLimitsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Limit related to a single NAT instance."
    INDEX { natInstanceIndex }
    ::= { natLimitsTable 1 }

NatLimitsEntry ::=
    SEQUENCE {
        natLimitMappings                Unsigned32,
        natMappingsNotifyThreshold      Unsigned32,
        natLimitAddressMappings         Unsigned32,
        natAddrMapNotifyThreshold       Unsigned32,
        natLimitFragments               Unsigned32,
        natLimitSubscribers              Unsigned32
    }

natLimitMappings OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Global limit on the total number of mappings. Zero means
        unlimited."
    ::= { natLimitsEntry 1 }

natMappingsNotifyThreshold OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See natNotifMappings."
    ::= { natLimitsEntry 2 }

natLimitAddressMappings OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

"Global limit on the total number of internal-to-external address mappings. Zero means unlimited.

This limit is only applicable to NATs that have an 'IP address pooling' behavior of 'Paired' [RFC4787]."  
 ::= { natLimitsEntry 3 }

natAddrMapNotifyThreshold OBJECT-TYPE

SYNTAX Unsigned32  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"See natNotifAddrMappings."  
 ::= { natLimitsEntry 4 }

natLimitFragments OBJECT-TYPE

SYNTAX Unsigned32  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"Global limit on the total number of fragments pending reassembly. Zero means unlimited.  
  
This limit is only applicable to NATs having 'Receive Fragments Out of Order' behavior [RFC4787]."  
 ::= { natLimitsEntry 5 }

natLimitSubscribers OBJECT-TYPE

SYNTAX Unsigned32  
MAX-ACCESS read-write  
STATUS current  
DESCRIPTION  
"Global limit on the number of subscribers with active mappings. Zero means unlimited."  
 ::= { natLimitsEntry 6 }

-- pools

natPoolObjects OBJECT IDENTIFIER ::= { natMIBObjects 14 }

natPoolTable OBJECT-TYPE

SYNTAX SEQUENCE OF NatPoolEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
"Table of pools."  
 ::= { natPoolObjects 1 }

```
natPoolEntry OBJECT-TYPE
    SYNTAX NatPoolEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Entry in the table of pools."
    INDEX { natInstanceIndex, natPoolIndex }
    ::= { natPoolTable 1 }

NatPoolEntry ::=
    SEQUENCE {
        natPoolIndex          NatPoolId,
        natPoolRealm          SnmpAdminString,
        natPoolWatermarkLow   Integer32,
        natPoolWatermarkHigh Integer32,
        natPoolPortMin        InetPortNumber,
        natPoolPortMax        InetPortNumber
    }

natPoolIndex OBJECT-TYPE
    SYNTAX NatPoolId
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index of an address pool."
    ::= { natPoolEntry 1 }

natPoolRealm OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE (0..32))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Realm to which this pool's addresses belong."
    ::= { natPoolEntry 2 }

natPoolWatermarkLow OBJECT-TYPE
    SYNTAX Integer32 (-1|0..100)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Low watermark on a pool's usage, in percentage of the total
        number of ports available. If set to -1, the watermark is
        disabled. Otherwise when the usage percentage becomes lower
        than or equal to natPoolWatermarkLow, a notification is
        sent. The NAT may also start behaving in low usage mode
        (this is implementation-defined).

        The pool's current usage percentage can be computed by
```



```
        summing (natPoolRangeAllocations -
        natPoolRangeDeallocations) over all address ranges
        belonging to this pool, then dividing by the total number of
        IP addresses in this pool and by the size of the port range
        in this pool (natPoolPortMax - natPoolPortMin + 1)."
```

::= { natPoolEntry 3 }

natPoolWatermarkHigh OBJECT-TYPE  
SYNTAX Integer32 (-1|0..100)  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
 "High watermark on a pool's usage, in percentage of the total  
 number of ports available. If set to -1, the watermark is  
 disabled. Otherwise, when the usage percentage becomes  
 higher than or equal to natPoolWatermarkHigh, a notification  
 is sent. The NAT may also start behaving in high usage mode  
 (this is implementation-defined)."

::= { natPoolEntry 4 }

natPoolPortMin OBJECT-TYPE  
SYNTAX InetPortNumber  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
 "Minimal port number to be allocated in this pool."

::= { natPoolEntry 5 }

natPoolPortMax OBJECT-TYPE  
SYNTAX InetPortNumber  
MAX-ACCESS read-create  
STATUS current  
DESCRIPTION  
 "Maximal port number to be allocated in this pool."

::= { natPoolEntry 6 }

natPoolRangeTable OBJECT-TYPE  
SYNTAX SEQUENCE OF NatPoolRangeEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
 "This table contains address ranges used by pool entries."

::= { natPoolObjects 2 }

natPoolRangeEntry OBJECT-TYPE  
SYNTAX NatPoolRangeEntry  
MAX-ACCESS not-accessible

```
STATUS current
DESCRIPTION
    "NAT pool address range."
INDEX { natInstanceIndex, natPoolRangePoolIndex }
 ::= { natPoolRangeTable 1 }

NatPoolRangeEntry ::=
SEQUENCE {
    natPoolRangePoolIndex      NatPoolId,
    natPoolRangeType           InetAddressType,
    natPoolRangeBegin          InetAddress,
    natPoolRangeEnd            InetAddress,
    natPoolRangeAllocations    Counter64,
    natPoolRangeDeallocations  Counter64
}

natPoolRangePoolIndex OBJECT-TYPE
SYNTAX NatPoolId
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Index of the address pool to which this address range
    belongs.  See natPoolIndex."
 ::= { natPoolRangeEntry 1 }

natPoolRangeType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The address type of natPoolRangeBegin and
    natPoolRangeEnd."
 ::= { natPoolRangeEntry 2 }

natPoolRangeBegin OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Lowest address included in this range."
 ::= { natPoolRangeEntry 3 }

natPoolRangeEnd OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Highest address included in this range."
```

```
 ::= { natPoolRangeEntry 4 }

natPoolRangeAllocations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of ports that have been allocated on the addresses in
         this range."
    ::= { natPoolRangeEntry 5 }

natPoolRangeDeallocations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of ports that have been allocated and then
         deallocated on the addresses in this range.

         The number of ports currently allocated on the addresses in
         this range can be computed by subtracting
         natPoolRangeDeallocations from natPoolRangeAllocations."
    ::= { natPoolRangeEntry 6 }

-- indexed mapping tables

natMapObjects OBJECT IDENTIFIER ::= { natMIBObjects 15 }

natMapIntAddrTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NatMapIntAddrEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Table of mappings from internal to external address.

         This table is only applicable to NATs that have an 'IP
         address pooling' behavior of 'Paired' [RFC4787]."
    ::= { natMapObjects 1 }

natMapIntAddrEntry OBJECT-TYPE
    SYNTAX NatMapIntAddrEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Mapping from internal to external address."
    INDEX { natInstanceIndex,
            natMapIntAddrIntRealm,
```

```
        natMapIntAddrIntType,
        natMapIntAddrInt }
 ::= { natMapIntAddrTable 1 }

NatMapIntAddrEntry ::=
SEQUENCE {
    natMapIntAddrIntRealm    SnmpAdminString,
    natMapIntAddrExtRealm    SnmpAdminString,
    natMapIntAddrIntType     InetAddressType,
    natMapIntAddrInt         InetAddress,
    natMapIntAddrExtType     InetAddressType,
    natMapIntAddrExt         InetAddress,
    natMapIntAddrSubsIndex   Unsigned32
}

natMapIntAddrIntRealm OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(0..32))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Realm to which natMapIntAddrInt belongs."
    ::= { natMapIntAddrEntry 1 }

natMapIntAddrExtRealm OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Realm to which natMapIntAddrExt belongs."
    ::= { natMapIntAddrEntry 2 }

natMapIntAddrIntType OBJECT-TYPE
    SYNTAX InetAddressType
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Address type for natMapIntAddrInt."
    ::= { natMapIntAddrEntry 3 }

natMapIntAddrInt OBJECT-TYPE
    SYNTAX InetAddress (SIZE (4|16))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Internal address."
    ::= { natMapIntAddrEntry 4 }

natMapIntAddrExtType OBJECT-TYPE
```

```
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Address type for natMapIntAddrExt."
 ::= { natMapIntAddrEntry 5 }

natMapIntAddrExt OBJECT-TYPE
    SYNTAX InetAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "External address."
    ::= { natMapIntAddrEntry 6 }

natMapIntAddrSubsIndex OBJECT-TYPE
    SYNTAX Unsigned32 (0|1..4294967295)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Subscriber to which this address mapping applies, or zero if
        it applies to all subscribers."
    ::= { natMapIntAddrEntry 7 }

natMappingTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NatMappingEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Table of mappings indexed by external 3-tuple."
    ::= { natMapObjects 2 }

natMappingEntry OBJECT-TYPE
    SYNTAX NatMappingEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A single NAT mapping."
    INDEX { natInstanceIndex,
            natMappingProto,
            natMappingExtRealm,
            natMappingExtAddressType,
            natMappingExtAddress,
            natMappingExtPort }
    ::= { natMappingTable 1 }

NatMappingEntry ::=
    SEQUENCE {
```

```
    natMappingProto          ProtocolNumber,
    natMappingExtRealm       SnmpAdminString,
    natMappingExtAddressType InetAddressType,
    natMappingExtAddress     InetAddress,
    natMappingExtPort        InetPortNumber,
    natMappingIntRealm       SnmpAdminString,
    natMappingIntAddressType InetAddressType,
    natMappingIntAddress     InetAddress,
    natMappingIntPort        InetPortNumber,
    natMappingPool           Unsigned32,
    natMappingMapBehavior    NatBehaviorType,
    natMappingFilterBehavior NatBehaviorType,
    natMappingAddressPooling NatPoolingType,
    natMappingSubsIndex      SubscriberIndex
}

natMappingProto OBJECT-TYPE
    SYNTAX ProtocolNumber
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The mapping's transport protocol number."
    ::= { natMappingEntry 1 }

natMappingExtRealm OBJECT-TYPE
    SYNTAX SnmpAdminString (SIZE(0..32))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The realm to which natMappingExtAddress belongs."
    ::= { natMappingEntry 2 }

natMappingExtAddressType OBJECT-TYPE
    SYNTAX InetAddressType
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Type of the mapping's external address."
    ::= { natMappingEntry 3 }

natMappingExtAddress OBJECT-TYPE
    SYNTAX InetAddress (SIZE (4|16))
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The mapping's external address. If this is the undefined
        address, all external addresses are mapped to the internal
        address."
```

```
 ::= { natMappingEntry 4 }

natMappingExtPort OBJECT-TYPE
    SYNTAX InetPortNumber
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The mapping's external port number. If this is zero, all
        external ports are mapped to the internal port."
    ::= { natMappingEntry 5 }

natMappingIntRealm OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The realm to which natMappingIntAddress belongs."
    ::= { natMappingEntry 6 }

natMappingIntAddressType OBJECT-TYPE
    SYNTAX InetAddressType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Type of the mapping's internal address."
    ::= { natMappingEntry 7 }

natMappingIntAddress OBJECT-TYPE
    SYNTAX InetAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The mapping's internal address. If this is the undefined
        address, addresses are not translated."
    ::= { natMappingEntry 8 }

natMappingIntPort OBJECT-TYPE
    SYNTAX InetPortNumber
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The mapping's internal port number. If this is zero, ports
        are not translated."
    ::= { natMappingEntry 9 }

natMappingPool OBJECT-TYPE
    SYNTAX Unsigned32 (0|1..4294967295)
    MAX-ACCESS read-only
```

```
STATUS current
DESCRIPTION
    "Index of the pool that contains this mapping's external
    address and port. If zero, no pool is associated with this
    mapping."
 ::= { natMappingEntry 10 }

natMappingMapBehavior OBJECT-TYPE
    SYNTAX NatBehaviorType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Mapping behavior as described in [RFC4787] section 4.1."
    ::= { natMappingEntry 11 }

natMappingFilterBehavior OBJECT-TYPE
    SYNTAX NatBehaviorType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Filtering behavior as described in [RFC4787] section 5."
    ::= { natMappingEntry 12 }

natMappingAddressPooling OBJECT-TYPE
    SYNTAX NatPoolingType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Type of address pooling behavior that was used to create
        this mapping."
    ::= { natMappingEntry 13 }

natMappingSubsIndex OBJECT-TYPE
    SYNTAX SubscriberIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Subscriber using this mapping."
    ::= { natMappingEntry 14 }

-- subscribers

natSubscribers OBJECT IDENTIFIER ::= { natMIBObjects 16 }

natSubscribersTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NatSubscribersEntry
    MAX-ACCESS not-accessible
```



```

STATUS current
DESCRIPTION
    "Table of CGN subscribers."
 ::= { natSubscribers 1 }

natSubscribersEntry OBJECT-TYPE
    SYNTAX NatSubscribersEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry describes a single CGN subscriber or a host
        served by a managed enterprise NAT."
    INDEX { natInstanceIndex,
            natSubscriberIndex }
    ::= { natSubscribersTable 1 }

NatSubscribersEntry ::=
    SEQUENCE {
        natSubscriberIndex          SubscriberIndex,
        natSubscriberIdentifierType  SubscriberIdentifierType,
        natSubscriberIntPrefixType   InetAddressType,
        natSubscriberIntPrefix       InetAddress,
        natSubscriberIntPrefixLength InetAddressPrefixLength,
        natSubscriberRealm           SnmpAdminString,
        natSubscriberTranslations    Counter64,
        natSubscriberOutOfPortErrors Counter64,
        natSubscriberResourceErrors  Counter64,
        natSubscriberQuotaDrops       Counter64,
        natSubscriberMappingCreations Counter64,
        natSubscriberMappingRemovals  Counter64,
        natSubscriberLimitMappings    Unsigned32,
        natSubscriberMapNotifyThresh  Unsigned32,
        natSubscriberVlanIdentifier   VlanIndexOrZero,
        natSubscriberVpnIdentifier    VPNIdOrZero,
        natSubscriberIPEncapsIdType   InetAddressType,
        natSubscriberIPEncapsIdAddr   InetAddress
    }

natSubscriberIndex OBJECT-TYPE
    SYNTAX SubscriberIndex
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index of the subscriber or host."
    ::= { natSubscribersEntry 1 }

natSubscriberIdentifierType OBJECT-TYPE
    SYNTAX SubscriberIdentifierType

```

MAX-ACCESS read-only  
STATUS current  
DESCRIPTION

"Type of additional information needed to identify the subscriber or host from incoming packets, when the packet source address does not do so unambiguously.

The implementation MUST ensure that the type and the identifier value provided are synchronized, as follows. Unused identifier values MUST be zero or equivalent.

Type	Identifier object
null(0)	None.
interfaces(1)	natSubsInterfaceIdentifierTable
vlan(2)	natSubscriberVlanIdentifier
vpn(3)	natSubscriberVpnIdentifier
ipencaps(4)	natSubscriberIPEncapsIdType and natSubscriberIPEncapsIdAddr
other(5)	As specified by the implementation"

::= { natSubscribersEntry 2 }

natSubscriberIntPrefixType OBJECT-TYPE  
SYNTAX InetAddressType  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"Subscriber's internal prefix type."  
::= { natSubscribersEntry 3 }

natSubscriberIntPrefix OBJECT-TYPE  
SYNTAX InetAddress  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"Prefix assigned to a subscriber's CPE."  
::= { natSubscribersEntry 4 }

natSubscriberIntPrefixLength OBJECT-TYPE  
SYNTAX InetAddressPrefixLength  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
"Length of the prefix assigned to a subscriber's CPE, in bits. In case a single address is assigned, this will be 32 for IPv4 and 128 for IPv6."  
::= { natSubscribersEntry 5 }

```
natSubscriberRealm OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The realm to which this subscriber belongs."
    ::= { natSubscribersEntry 6 }

natSubscriberTranslations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of translated packets received from or sent to
         this subscriber."
    ::= { natSubscribersEntry 7 }

natSubscriberOutOfPortErrors OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets received from this subscriber not
         translated because no external port was available, excluding
         quota limitations."
    ::= { natSubscribersEntry 8 }

natSubscriberResourceErrors OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of packets received from this subscriber not
         translated because of resource constraints (excluding
         out-of-port errors and quota drops)."
    ::= { natSubscribersEntry 9 }

natSubscriberQuotaDrops OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of incoming packets received from or destined to
         this subscriber not translated because of quota limitations.
         Quotas include absolute limits as well as limits on the rate
         of allocation."
    ::= { natSubscribersEntry 10 }
```

```
natSubscriberMappingCreations OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of mappings created by or for this subscriber."
    ::= { natSubscribersEntry 11 }

natSubscriberMappingRemovals OBJECT-TYPE
    SYNTAX Counter64
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of mappings removed by or for this subscriber."
    ::= { natSubscribersEntry 12 }

natSubscriberLimitMappings OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Limit on the number of active mappings created by or for
        this subscriber. Zero means unlimited."
    ::= { natSubscribersEntry 13 }

natSubscriberMapNotifyThresh OBJECT-TYPE
    SYNTAX Unsigned32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See natNotifSubscriberMappings."
    ::= { natSubscribersEntry 14 }

natSubscriberVlanIdentifier OBJECT-TYPE
    SYNTAX VlanIndexOrZero
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "When non-zero, VLAN index used to identify subscriber in
        combination with packet source address."
    ::= { natSubscribersEntry 15 }

natSubscriberVpnIdentifier OBJECT-TYPE
    SYNTAX VpnIdOrZero
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "When non-zero, VPN identifier used to identify subscriber
```

```

        in combination with packet source address."
 ::= { natSubscribersEntry 16 }

natSubscriberIPEncapsIdType OBJECT-TYPE
    SYNTAX InetAddressType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "When not unknown(0), type of address of encapsulating IP
        ingress tunnel."
 ::= { natSubscribersEntry 17 }

natSubscriberIPEncapsIdAddr OBJECT-TYPE
    SYNTAX InetAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Source address in outer header of packets incoming via IP
        tunnel, used to identify subscriber in combination with
        inner packet source address."
 ::= { natSubscribersEntry 18 }

natSubsInterfaceIdentifierTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NatSubsInterfaceIdentifierEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Table of interface indexes. If non-empty, used along with
        packet source address to identify the subscriber sending
        the packet. 'OR' semantics if multiple interface indexes
        are present."
 ::= { natSubscribers 2 }

natSubsInterfaceIdentifierEntry OBJECT-TYPE
    SYNTAX NatSubsInterfaceIdentifierEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Each entry provides a single interface index."
    INDEX { natInstanceIndex,
            natSubsInterfaceIdSubsIndex,
            natSubsInterfaceIdRowIndex }
 ::= { natSubsInterfaceIdentifierTable 1 }

NatSubsInterfaceIdentifierEntry ::=
    SEQUENCE {
        natSubsInterfaceIdSubsIndex      SubscriberIndex,
        natSubsInterfaceIdRowIndex       SubsInterfaceIdRowIndex,

```

```
        natSubsInterfaceIndex      InterfaceIndex
    }

natSubsInterfaceIdSubsIndex OBJECT-TYPE
    SYNTAX SubscriberIndex
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index of the subscriber to which this conceptual table is
        related."
    ::= { natSubsInterfaceIdentifierEntry 1 }

natSubsInterfaceIdRowIndex OBJECT-TYPE
    SYNTAX SubsInterfaceIdRowIndex
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Row index."
    ::= { natSubsInterfaceIdentifierEntry 2 }

natSubsInterfaceIndex OBJECT-TYPE
    SYNTAX InterfaceIndex
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Interface index of an ingress interface through which
        packets from this subscriber may flow."
    ::= { natSubsInterfaceIdentifierEntry 3 }

-- object groups

natGroupStatelessObjects OBJECT-GROUP
    OBJECTS { natInstanceAlias,
               natTranslations,
               natResourceErrors,
               natQuotaDrops,
               natMappingCreations,
               natMappingRemovals,
               natL4ProtocolTranslations ,
               natL4ProtocolResourceErrors,
               natL4ProtocolQuotaDrops,
               natL4ProtocolMappingCreations,
               natL4ProtocolMappingRemovals,
               natMappingIntRealm,
               natMappingIntAddressType,
               natMappingIntAddress,
               natMappingIntPort,
```

```
        natMappingPool,
        natMappingMapBehavior,
        natMappingFilterBehavior }
STATUS current
DESCRIPTION
    "Basic counters, limits, and thresholds that do not require
    stateful NAT. That is, they apply to both stateless and
    stateful NATs.

    For this MIB's purposes, stateless NATs are defined as NATs
    that do not create mappings dynamically (either implicitly
    or explicitly using, for instance, the Port Control
    Protocol). Their mappings are created statically by the NAT
    administrator."
 ::= { natMIBGroups 7 }

natGroupStatefulObjects OBJECT-GROUP
    OBJECTS { natOutOfPortErrors,
               natL4ProtocolOutOfPortErrors,
               natLimitMappings,
               natMappingsNotifyThreshold,
               natPoolRealm,
               natPoolWatermarkLow,
               natPoolWatermarkHigh,
               natPoolPortMin,
               natPoolPortMax,
               natPoolRangeType,
               natPoolRangeBegin,
               natPoolRangeEnd,
               natPoolRangeAllocations,
               natPoolRangeDeallocations,
               natMappingAddressPooling }
STATUS current
DESCRIPTION
    "Basic counters, limits, and thresholds that require stateful
    NAT."
 ::= { natMIBGroups 8 }

natGroupAddrMapObjects OBJECT-GROUP
    OBJECTS { natAddressMappingCreations,
               natAddressMappingRemovals,
               natLimitAddressMappings,
               natAddrMapNotifyThreshold,
               natMapIntAddrExtRealm,
               natMapIntAddrExtType,
               natMapIntAddrExt }
STATUS current
DESCRIPTION
```

```
        "Objects that require 'Paired IP address pooling' behavior
        [RFC4787]."
```

::= { natMIBGroups 9 }

natGroupFragmentObjects OBJECT-GROUP

OBJECTS { natLimitFragments }

STATUS current

DESCRIPTION

"Objects that require 'Receive Fragments Out of Order'
 behavior [RFC4787]."

::= { natMIBGroups 10 }

natGroupBasicNotifications NOTIFICATION-GROUP

NOTIFICATIONS { natNotifPoolWatermarkLow,

natNotifPoolWatermarkHigh,

natNotifMappings }

STATUS current

DESCRIPTION

"Basic notifications."

::= { natMIBGroups 11 }

natGroupAddrMapNotifications NOTIFICATION-GROUP

NOTIFICATIONS { natNotifAddrMappings }

STATUS current

DESCRIPTION

"Notifications about address mappings."

::= { natMIBGroups 12 }

natGroupSubscriberObjects OBJECT-GROUP

OBJECTS { natMapIntAddrSubsIndex,

natMappingSubsIndex,

natSubscriberIdentifierType,

natSubscriberIntPrefixType,

natSubscriberIntPrefix,

natSubscriberIntPrefixLength,

natSubscriberRealm,

natSubscriberTranslations,

natSubscriberOutOfPortErrors,

natSubscriberResourceErrors,

natSubscriberQuotaDrops,

natSubscriberMappingCreations,

natSubscriberMappingRemovals,

natSubscriberLimitMappings,

natSubscriberVlanIdentifier,

natSubscriberVpnIdentifier,

natSubscriberIPEncapsIdType,

natSubscriberIPEncapsIdAddr,

natSubsInterfaceIndex,



```
        natLimitSubscribers,
        natSubscriberMapNotifyThresh }
STATUS current
DESCRIPTION
    "Per-subscriber counters, limits, and thresholds."
 ::= { natMIBGroups 13 }

natGroupSubscriberNotifications NOTIFICATION-GROUP
NOTIFICATIONS { natNotifSubscriberMappings }
STATUS current
DESCRIPTION
    "Subscriber notifications."
 ::= { natMIBGroups 14 }

-- compliance statements

natBasicStatelessCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
    "Basic stateless compliance with this MIB is attained when
    the objects contained in the mandatory groups are
    implemented."
MODULE -- this module
    MANDATORY-GROUPS { natGroupStatelessObjects }

    OBJECT      natInstanceAlias
    MIN-ACCESS   read-only
    DESCRIPTION
        "Write access is not required."

 ::= { natMIBCompliances 3 }

natBasicStatefulCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
    "Basic stateful compliance with this MIB is attained when the
    objects contained in the mandatory groups are implemented."
MODULE -- this module
    MANDATORY-GROUPS { natGroupStatelessObjects,
                        natGroupStatefulObjects,
                        natGroupBasicNotifications }
 ::= { natMIBCompliances 4 }

natAddrMapCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
    "NATs that have 'Paired IP address pooling' behavior
```

```
        [RFC4787] and implement the objects in this group can claim
        this level of compliance."
MODULE -- this module
    MANDATORY-GROUPS { natGroupStatelessObjects,
                        natGroupStatefulObjects,
                        natGroupBasicNotifications,
                        natGroupAddrMapObjects,
                        natGroupAddrMapNotifications }
 ::= { natMIBCompliances 5 }

natFragmentsCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "NATs that have 'Receive Fragments Out of Order' behavior
        [RFC4787] and implement the objects in this group can claim
        this level of compliance."
MODULE -- this module
    MANDATORY-GROUPS { natGroupStatelessObjects,
                        natGroupStatefulObjects,
                        natGroupBasicNotifications,
                        natGroupFragmentObjects }
 ::= { natMIBCompliances 6 }

natCGNCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "NATs that have 'Paired IP address pooling' and 'Receive
        Fragments Out of Order' behavior [RFC4787] and implement the
        objects in this group can claim this level of compliance.

        This level of compliance is to be expected of a CGN
        compliant with [RFC6888]."
```

```
MODULE -- this module
    MANDATORY-GROUPS { natGroupStatelessObjects,
                        natGroupStatefulObjects,
                        natGroupBasicNotifications,
                        natGroupAddrMapObjects,
                        natGroupAddrMapNotifications,
                        natGroupFragmentObjects,
                        natGroupSubscriberObjects,
                        natGroupSubscriberNotifications }
 ::= { natMIBCompliances 7 }

END
```

## 5. Security Considerations

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

Limits: An attacker setting a very low or very high limit can easily cause a denial-of-service situation.

- \* natLimitMappings
- \* natLimitAddressMappings
- \* natLimitFragments
- \* natLimitSubscribers
- \* natSubscriberLimitMappings

Notification thresholds: An attacker setting an arbitrarily low threshold can cause many useless notifications to be generated. Setting an arbitrarily high threshold can effectively disable notifications, which could be used to hide another attack.

- \* natMappingsNotifyThreshold
- \* natAddrMapNotifyThreshold
- \* natSubscriberMapNotifyThresh

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

Objects that reveal host identities: Various objects can reveal the identity of private hosts that are engaged in a session with external end nodes. A curious outsider could monitor these to assess the number of private hosts being supported by the NAT device. Further, a disgruntled former employee of an enterprise could use the information to break into specific private hosts by

intercepting the existing sessions or originating new sessions into the host.

- \* natMapIntAddrType
- \* natMapIntAddrInt
- \* natMapIntAddrExt
- \* natMappingIntRealm
- \* natMappingIntAddressType
- \* natMappingIntAddress
- \* natMappingIntPort
- \* natMappingMapBehavior
- \* natMappingFilterBehavior
- \* natMappingAddressPooling
- \* natSubscriberIntPrefixType
- \* natSubscriberIntPrefix
- \* natSubscriberIntPrefixLength

Other objects that reveal NAT state: Other managed objects in this MIB may contain information that may be sensitive from a business perspective, in that they may represent NAT state information.

- \* natCntAddressMappings
- \* natCntProtocolMappings
- \* natPoolUsage
- \* natPoolRangeAllocatedPorts
- \* natSubscriberCntMappings

There are no objects that are sensitive in their own right, such as passwords or monetary amounts.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec),

there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

## 6. IANA Considerations

IANA has assigned object identifier 123 to the natMIB module, with prefix iso.org.dod.internet.mgmt.mib-2 in the Network Management Parameters registry [SMI-NUMBERS].

No IANA actions are required by this document.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.
- [RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC4265] Schliesser, B. and T. Nadeau, "Definition of Textual Conventions for Virtual Private Network (VPN) Management", RFC 4265, November 2005.
- [RFC4363] Levi, D. and D. Harrington, "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions", RFC 4363, January 2006.
- [RFC4750] Joyal, D., Galecki, P., Giacalone, S., Coltun, R., and F. Baker, "OSPF Version 2 Management Information Base", RFC 4750, December 2006.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.
- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 6353, July 2011.

## 7.2. Informative References

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC4008] Rohit, R., Srisuresh, P., Raghunarayan, R., Pai, N., and C. Wang, "Definitions of Managed Objects for Network Address Translators (NAT)", RFC 4008, March 2005.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6674] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment", RFC 6674, July 2012.
- [RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.
- [SMI-NUMBERS]  
 , "Network Management Parameters registry at IANA", ,  
 <<http://www.iana.org/assignments/smi-numbers>>.

#### Authors' Addresses

Simon Perreault  
Viagenie  
246 Aberdeen  
Quebec, QC G1R 2E1  
Canada

Phone: +1 418 656 9254  
Email: [simon.perreault@viagenie.ca](mailto:simon.perreault@viagenie.ca)  
URI: <http://viagenie.ca>

Tina Tsou  
Huawei Technologies (USA)  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

Phone: +1 408 330 4424  
Email: tina.tsou.zouting@huawei.com

Senthil Sivakumar  
Cisco Systems  
7100-8 Kit Creek Road  
Research Triangle Park, North Carolina 27709  
USA

Phone: +1 919 392 5158  
Email: ssenthil@cisco.com



BEHAVE  
Internet-Draft  
Intended status: Best Current Practice  
Expires: December 05, 2013

R. Penno  
Cisco  
S. Perreault  
Viagenie  
S. Kamiset  
Insieme Networks  
M. Boucadair  
France Telecom  
K. Naito  
NTT  
June 03, 2013

Network Address Translation (NAT) Behavioral Requirements Updates  
draft-ietf-behave-requirements-update-00

Abstract

This document clarifies and updates several requirements of RFC4787, RFC5382 and RFC5508 based on operational and development experience. The focus of this document is NAPT44.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

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

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

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

This Internet-Draft will expire on December 05, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Terminology . . . . .	3
2. Introduction . . . . .	3
2.1. Scope . . . . .	3
3. TCP Session Tracking . . . . .	3
3.1. TCP Transitory Connection Idle-Timeout . . . . .	4
3.2. TIME_WAIT State . . . . .	4
3.2.1. Proposal: Apply RFC6191 and PAWS to NAT . . . . .	5
3.3. TCP RST . . . . .	7
4. Port Overlapping behavior . . . . .	8
5. Address Pooling Paired (APP) . . . . .	9
6. EIF Security . . . . .	9
7. EIF Protocol Independence . . . . .	9
8. EIF Mapping Refresh . . . . .	9
8.1. Outbound Mapping Refresh and Error Packets . . . . .	10
9. EIM Protocol Independence . . . . .	10
10. Port Parity . . . . .	10
11. Port Randomization . . . . .	10
12. IP Identification (IP ID) . . . . .	10
13. ICMP Query Mappings Timeout . . . . .	11
14. Hairpinning Support for ICMP Packets . . . . .	11
15. IANA Considerations . . . . .	11
16. Security Considerations . . . . .	11
17. Acknowledgements . . . . .	11
18. References . . . . .	11
18.1. Normative References . . . . .	12
18.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Terminology

The reader should be familiar with all terms defined in RFC2663 [RFC2663], RFC4787 [RFC4787], RFC5382 [RFC5382], RFC5508 [RFC5508]

## 2. Introduction

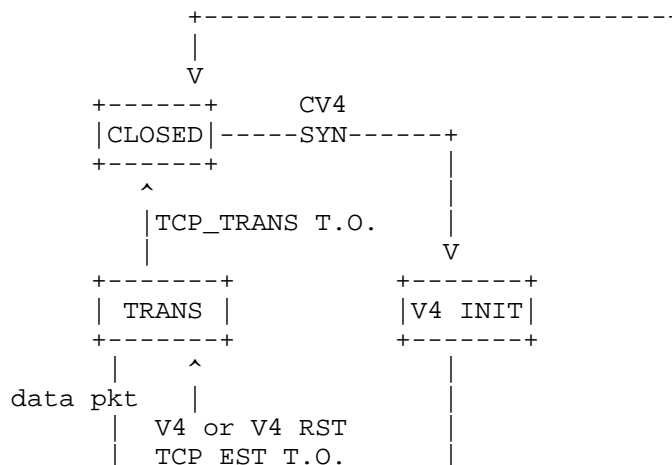
[RFC4787], [RFC5382] and [RFC5508] greatly advanced NAT interoperability and conformance. But with widespread deployment and evolution of NAT more development and operational experience was acquired some areas of the original documents need further clarification or updates. This documents provides such clarifications and updates.

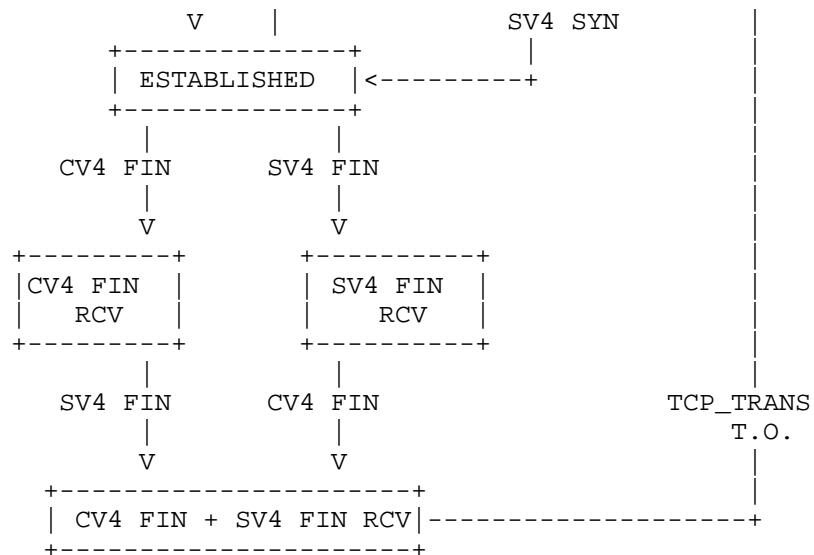
### 2.1. Scope

This document focuses solely on NAT44 and its goal is to clarify, fill gaps or update requirements of [RFC4787], [RFC5382] and [RFC5508]. It is out of the scope of this document the creation of completely new requirements not associated with the documents cited above. New requirements would be better served elsewhere and if they are CGN specific in an update to [RFC6888] [I-D.ietf-behave-lsn-requirements]

## 3. TCP Session Tracking

[RFC5382] specifies TCP timers associated with various connection states but does not specify the TCP state machine a NAT44 should use as a basis to apply such timers. The TCP state machine below, adapted from [RFC6146], provides guidance on how TCP session tracking could be implemented - it is non-normative.





(postamble)

### 3.1. TCP Transitory Connection Idle-Timeout

[RFC5382]:REQ-5 The transitory connection idle-timeout is defined as the minimum time a TCP connection in the partially open or closing phases must remain idle before the NAT considers the associated session a candidate for removal. But the document does not clearly states if these can be configured separately. This document clarifies that a NAT device SHOULD provide different knobs for configuring the open and closing idle timeouts. This document further acknowledges that most TCP flows are very short (less than 10 seconds) [FLOWRATE][TCPWILD] and therefore a partially open timeout of 4 minutes might be excessive if security is a concern. Therefore it MAY be configured to be less than 4 minutes in such cases. There also may be cases that a timeout of 4 minutes might be excessive. The case and the solution are written below.

### 3.2. TIME\_WAIT State

The TCP TIME\_WAIT state is described in [RFC0793]. The TCP TIME\_WAIT state needs to be kept for 2MSL before a connection is CLOSED, for the reasons below.

- 1: In the event that packets from a session are delayed in the in-between network, and delivered to the end relatively later, we should prevent the packets from being transferred and interpreted as a packet that belongs to a new session.
- 2: If the remote TCP has not received the acknowledgment of its connection termination request, it will re-send the FIN packet several times.

These points are important for the TCP to work without problems.

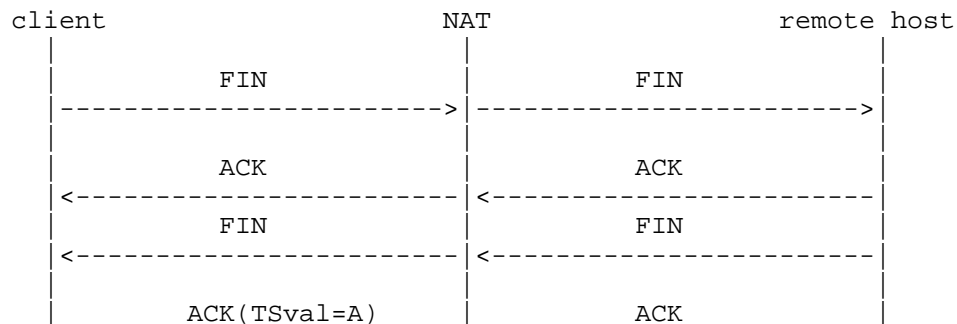
[RFC5283] leaves the handling of TCP connections in TIME\_WAIT state unspecified and mentions that TIME\_WAIT state is not part of the transitory connection idle-timeout. If the NAT device honors the TIME\_WAIT state, each TCP connection and its associated resources is kept for a certain period, typically for four minutes, which consumes port resources.

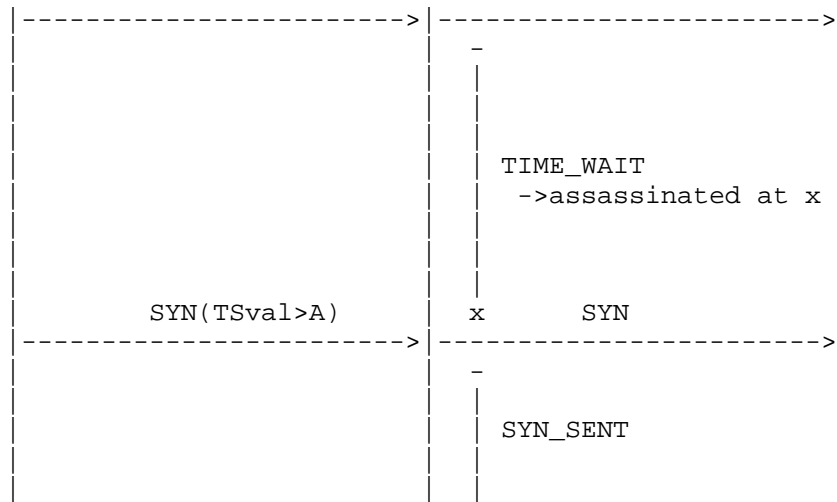
[RFC6191] explains that in certain situation it is necessary to reduce the TIME\_WAIT state and defines such a mechanism using TCP timestamps and sequence numbers. When a connection request is received with a four-tuple that is in the TIME\_WAIT state, the connection request may be accepted if the sequence number or the timestamp of the incoming SYN segment is greater than the last sequence number seen on the previous incarnation of the connection.

[N.E] This document specifies that a NAT device should keep TCP connections in TIME\_WAIT state unless it implements the proposal below?

### 3.2.1. Proposal: Apply RFC6191 and PAWS to NAT

This section proposes to apply [RFC6191] mechanism at NAT. This mechanism MAY be adopted for both clients' and remote hosts' TCP active close.





(postamble)

Also, PAWS works to discard old duplicate packets at NAT. A packet can be discarded as an old duplicate if it is received with a timestamp or sequence number value less than a value recently received on the connection.

To make these mechanisms work, we should concern the case that there are several clients with nonsuccessive timestamp or sequence number values are connected to a NAT device (i.e. not monotonically increasing among clients). Two mechanisms to solve this mechanism and applying [RFC6191] and PAWS to NAT are described below. These mechanisms are optional.

#### 3.2.1.1. Rewrite timestamp and sequence number values at NAT

Rewrite timestamp and sequence number values of outgoing packets at NAT to be monotonically increasing. This can be done by adopting following mechanisms at NAT.

- A: Store the newest rewritten value of timestamp and sequence number as the "max value at the time".
- B: NAT rewrite timestamp and sequence number values of incoming packets to be monotonically increasing.

When packets come back as replies from remote hosts, NAT rewrite again the timestamp and sequence number values to be the original values. This can be done by adopting following mechanisms at NAT.

C: Store the values of original timestamp and sequence number of packets, and rewritten values of those.

#### 3.2.1.2. Split an assignable number of port space to each client

Adopt following mechanisms at NAT.

A: Choose clients that can be assigned ports.

B: Split assignable port numbers between clients.

Packets from other clients which are not chosen by these mechanisms are rejected at NAT, unless there is unassigned port left.

#### 3.2.1.3. Resend the last ACK to the resended FIN

We should concern another case to make RFC6191 work at NAT. In case the remote TCP could not receive the acknowledgment of its connection termination request, the NAT device, on behalf of clients, resends the last ACK packet when it receives an FIN packet of the previous connection, and when the state of the previous connection is deleted from the NAT. This mechanism MAY be used when clients starts closing process, and the remote host could not receive the last ACK.

#### 3.2.1.4. Remote host behavior of several implementations

To solve the port shortage problem on the client side, the behavior of remote host should be compliant to [RFC6191] or the mechanism written in 4.2.2.13 of [RFC1122], since NAT may reuse the same 5 tuple for a new connection. We have investigated behaviors of OSes (e.g., Linux, FreeBSD, Windows, MacOS), and found that they implemented the server side behavior of the above two.

### 3.3. TCP RST

[RFC5382] leaves the handling of TCP RST packets unspecified. This document does not try standardize such behavior but clarifies based on operational experience that a NAT that receives a TCP RST for an active mapping and performs session tracking MAY immediately delete the sessions and remove any state associated with it. If the NAT device that performs TCP session tracking receives a TCP RST for the first session that created a mapping, it MAY remove the session and the mapping immediately.

#### 4. Port Overlapping behavior

[RFC4787] [RFC5382]: REQ-1 Current RFCs specify a specific port overlapping behavior, i.e., that the external IP:port can be reused for connections originating from the same internal source IP:port irrespective of the destination. This is known as endpoint-independent mapping. This document clarifies that this port overlapping behavior can be extended to connections originating from different internal source IP:ports as long as their destinations are different. This is known as EDM (Endpoint Dependent Mapping). The mechanism below MAY be one optional implement to NAT.

If destination addresses and ports are different for outgoing connections started by local clients, NAT MAY assign the same external port as the source ports for the connections. The port overlapping mechanism manages mappings between external packets and internal packets by looking at and storing their 5-tuple (protocol, source address, source port, destination address, destination port). This enables concurrent use of a single NAT external port for multiple transport sessions, which enables NAT to work correctly in IP address resource limited network.

#### Discussions:

[RFC4787] and [RFC5382] requires "endpoint-independent mapping" at NAT, and port overlapping NAT cannot meet the requirement. This mechanism can degrade the transparency of NAT in that its mapping mechanism is endpoint-dependent and makes NAT traversal harder. However, if a NAT adopts endpoint-independent mapping together with endpoint-dependent filtering, then the actual behavior of the NAT will be the same as port overlapping NAT. It should also be noted that a lot of existing NAT devices(e.g., SEIL, FITElnet Series) adopted this port overlapping mechanism.

A: Reference URL for SEIL -> [www.seil.jp](http://www.seil.jp)

B: Reference URL for FITElnet -> [www.furukawa.co.jp/fitelnet](http://www.furukawa.co.jp/fitelnet)



The netfilter, which is a popular packet filtering mechanism for Linux, also adopts port overlapping behavior.

#### 5. Address Pooling Paired (APP)

[RFC4787]: REQ-2 [RFC5382]:ND Address Pooling Paired behavior for NAT is recommended in previous documents but behavior when a public IPv4 run out of ports is left undefined. This document clarifies that if APP is enabled new sessions from a subscriber that already has a mapping associated with a public IP that ran out of ports SHOULD be dropped. The administrator MAY provide a knob that allows a NAT device to starting using ports from another public IP when the one that anchored the APP mapping ran out of ports. This is trade-off between subscriber service continuity and APP strict enforcement. (NE: It is sometimes referred as 'soft-APP')

#### 6. EIF Security

[RFC4787]:REQ-8 and [RFC5382]:REQ-3 End-point independent filtering could potentially result in security attacks from the public realm. In order to handle this, when possible there MUST be strict filtering checks in the inbound direction. A knob SHOULD be provided to limit the number of inbound sessions and a knob SHOULD be provided to enable or disable EIF on a per application basis. This is specially important in the case of Mobile networks where such attacks can consume radio resources and count against the user quota.

#### 7. EIF Protocol Independence

[RFC4787]:REQ-8 and[RFC5382]: REQ-3 Current RFCs do not specify whether EIF mappings are protocol independent. In other words, if an outbound TCP SYN creates a mapping, it is left undefined whether inbound UDP packets destined to that mapping should be forwarded. This document specifies that EIF mappings SHOULD be protocol independent in order allow inbound packets for protocols that multiplex TCP and UDP over the same IP: port through the NAT and also maintain compatibility with stateful NAT64 RFC6146 [RFC6146]. But the administrator MAY provide a configuration knob to make it protocol dependent.

#### 8. EIF Mapping Refresh

[RFC4787]: REQ-6 [RFC5382]: ND The NAT mapping Refresh direction MAY have a "NAT Inbound refresh behavior" of "True" but it does not clarifies how this applies to EIF mappings. The issue in question is whether inbound packets that match an EIF mapping but do not create a new session due to a security policy should refresh the mapping timer. This document clarifies that even when a NAT device has a

inbound refresh behavior of TRUE, such packets SHOULD NOT refresh the mapping. Otherwise a simple attack of a packet every 2 minutes can keep the mapping indefinitely.

#### 8.1. Outbound Mapping Refresh and Error Packets

In the case of NAT outbound refresh behavior there are certain types of packets that should not refresh the mapping even if their direction is outbound. For example, if the mapping is kept alive by ICMP Errors or TCP RST outbound packets sent as response to inbound packets, these SHOULD NOT refresh the mapping.

#### 9. EIM Protocol Independence

[RFC4787] [RFC5382]: REQ-1 Current RFCs do not specify whether EIM are protocol independent. In other words, if a outbound TCP SYN creates a mapping it is left undefined whether outbound UDP can reuse such mapping and create session. On the other hand, Stateful NAT64 [RFC6146] clearly specifies three binding information bases (TCP, UDP, ICMP). This document clarifies that EIM mappings SHOULD be protocol dependent. A knob MAY be provided in order allow protocols that multiplex TCP and UDP over the same source IP and port to use a single mapping.

#### 10. Port Parity

A NAT devices MAY disable port parity preservation for dynamic mappings. Nevertheless, A NAT SHOULD support means to explicitly request to preserve port parity (e.g., [I-D.pcp-port-set]).

#### 11. Port Randomization

A NAT SHOULD follow the recommendations specified in Section 4 of [RFC6056] especially: "A NAT that does not implement port preservation [RFC4787] [RFC5382] SHOULD obfuscate selection of the ephemeral port of a packet when it is changed during translation of that packet. A NAT that does implement port preservation SHOULD obfuscate the ephemeral port of a packet only if the port must be changed as a result of the port being already in use for some other session. A NAT that performs parity preservation and that must change the ephemeral port during translation of a packet SHOULD obfuscate the ephemeral ports. The algorithms described in this document could be easily adapted such that the parity is preserved (i.e., force the lowest order bit of the resulting port number to 0 or 1 according to whether even or odd parity is desired)."

#### 12. IP Identification (IP ID)

A NAT SHOULD handle the Identification field of translated IPv4 packets as specified in Section 9 of [I-D.ietf-intarea-ipv4-id-update].

### 13. ICMP Query Mappings Timeout

Section 3.1 of [RFC5508] says that ICMP Query Mappings are to be maintained by NAT device. However, RFC doesn't discuss about the Query Mapping timeout values. Section 3.2 of that RFC only discusses about ICMP Query Session Timeouts. ICMP Query Mappings MAY be deleted once the last the session using the mapping is deleted.

### 14. Hairpinning Support for ICMP Packets

[RFC5508]:REQ-7 This requirement specifies that NAT devices enforcing Basic NAT MUST support traversal of hairpinned ICMP Query sessions. This implicitly means that address mappings from external address to internal address (similar to Endpoint Independent Filters) MUST be maintained to allow inbound ICMP Query sessions. If an ICMP Query is received on an external address, NAT device can then translate to an internal IP. [RFC5508]:REQ-7 This requirement specifies that all NAT devices (i.e., Basic NAT as well as NAPT devices) MUST support the traversal of hairpinned ICMP Error messages. This too requires NAT devices to maintain address mappings from external IP address to internal IP address in addition to the ICMP Query Mappings described in section 3.1 of that RFC.

### 15. IANA Considerations

TBD

### 16. Security Considerations

In the case of EIF mappings due to high risk of resource crunch, a NAT device MAY provide a knob to limit the number of inbound sessions spawned from a EIF mapping.

[TCP-Security] contains a detailed discussion of the security implications of TCP Timestamps and of different timestamp generation algorithms.

### 17. Acknowledgements

Thanks to Dan Wing, Suresh Kumar, Mayuresh Bakshi, Rajesh Mohan and Senthil Sivamular for review and discussions

### 18. References

## 18.1. Normative References

- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.
- [RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, October 2003.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.
- [RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6191] Gont, F., "Reducing the TIME-WAIT State Using TCP Timestamps", BCP 159, RFC 6191, April 2011.
- [RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.

## 18.2. Informative References

## [FLOWRATE]

Zhang, Y., Breslau, L., Paxson, V., and S. Shenker, "On the Characteristics and Origins of Internet Flow Rates", .

## [I-D.ietf-pcp-port-set]

Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", draft-ietf-pcp-port-set-01 (work in progress), May 2013.

## [I-D.naito-nat-resource-optimizing-extension]

Kengo, K. and A. Matsumoto, "NAT TIME\_WAIT reduction", draft-naito-nat-resource-optimizing-extension-02 (work in progress), July 2012.

## [TCPWILD]

Qian, F., Subhabrata, S., Spatscheck, O., Morley Mao, Z., and W. Willinger, "TCP Revisited: A Fresh Look at TCP in the Wild", .

## Authors' Addresses

Reinaldo Penno  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, California 95134  
USA

Email: repenno@cisco.com

Simon Perreault  
Viagenie  
2875 boul. Laurier, suite D2-630  
Quebec, QC G1V 2M2  
Canada

Email: simon.perreault@viagenie.ca

Sarat Kamiset  
Insieme Networks  
California

Mohamed Boucadair  
France Telecom  
Rennes 35000  
France

Email: mohamed.boucadair@orange.com

Kengo Naito  
NTT  
Tokyo  
Japan

Email: kengo@lab.ntt.co.jp

Behave Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 29, 2014

Z. Chen  
China Telecom  
C. Zhou  
T. Tsou  
T. Taylor, Ed.  
Huawei Technologies  
January 25, 2014

Syslog Format for NAT Logging  
draft-ietf-behave-syslog-nat-logging-06

Abstract

NAT devices are required to log events like creation and deletion of translations and information about the resources the NAT is managing. The logs are required to identify an attacker or a host that was used to launch malicious attacks, and for various other purposes of accounting and management. Since there is no standard way of logging this information, different NAT devices behave differently. The lack of a consistent way makes it difficult to write the collector applications that would receive this data and process it to present useful information.

This document describes the information that is required to be logged by the NAT devices. It goes on to standardize formats for reporting these events and parameters using SYSLOG (RFC 5424). A companion document specifies formats for reporting the same events and parameters using IPFIX (RFC 7011). Applicability statements are provided in this document and its companion to guide operators and implementors in their choice of which technology to use for logging.

Status of This Memo

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

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

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

This Internet-Draft will expire on July 29, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	4
1.1. Terminology . . . . .	5
2. Deployment Considerations . . . . .	6
2.1. Static and Dynamic NATs . . . . .	6
2.2. Realms and Address Pools . . . . .	7
2.2.1. Address Pools . . . . .	7
2.3. NAT Logging Requirements For Different Transition Methods . . . . .	8
2.4. Subscriber Identification . . . . .	9
2.5. The Port Control Protocol (PCP) . . . . .	10
2.6. Logging At the Customer Edge . . . . .	10
3. NAT-Related Events and Parameters . . . . .	10
3.1. Events Relating To Allocation Of Resources To Hosts . . . . .	10
3.1.1. NAT Address Mapping Creation and Deletion . . . . .	11
3.1.2. NAT Address and Port Mapping Creation and Deletion . . . . .	12
3.1.3. NAT Session Creation and Deletion . . . . .	14
3.1.3.1. Destination Logging . . . . .	17
3.1.4. Port Range Allocation and Deallocation . . . . .	17
3.2. Threshold Events . . . . .	19
3.2.1. Address Pool High- and Low-Water-Mark Threshold Events . . . . .	19
3.2.2. Global Address Mapping High-Water-Mark Threshold Event . . . . .	20
3.2.3. Global Address and Port Mapping High-Water-Mark Threshold Event . . . . .	21
3.2.4. Subscriber-Specific Address and Port Mapping Threshold Event . . . . .	22
3.3. Limit-Related Events . . . . .	22
3.3.1. Global Address Mapping Limit Exceeded . . . . .	22
3.3.2. Global Address and Port Mapping Limit Exceeded . . . . .	23
3.3.3. Global Limit On Number of Active Hosts Exceeded . . . . .	24
3.3.4. Subscriber-Specific Limit On Number of Address and . . . . .	



Port Mappings Exceeded . . . . .	25
3.3.5. Global Limit On Number Of Fragments Pending Reassembly Exceeded . . . . .	26
4. SYSLOG Applicability . . . . .	27
5. SYSLOG Record Format For NAT Logging . . . . .	27
5.1. SYSLOG HEADER Fields . . . . .	28
5.2. Parameter Encodings . . . . .	29
5.2.1. General Encoding Rules . . . . .	32
5.2.2. Special Cases . . . . .	32
5.2.3. Relationship To Objects In the NAT MIB . . . . .	33
5.3. Encoding Of Complete Log Report For Each Event Type . . . . .	35
5.3.1. Encoding of Events Relating To Allocation Of Resources To Hosts . . . . .	35
5.3.1.1. NAT Address Mapping Creation and Deletion . . . . .	36
5.3.1.2. NAT Address and Port Mapping Creation and Deletion . . . . .	37
5.3.1.3. NAT Session Creation and Deletion . . . . .	39
5.3.1.4. Port Range Allocation and Deallocation . . . . .	41
5.3.2. Encoding of Threshold Events . . . . .	43
5.3.2.1. NAT Address Pool High- and Low-Water-Mark Threshold Events . . . . .	43
5.3.2.2. Global Address Mapping High-Water-Mark Threshold Exceeded . . . . .	44
5.3.2.3. Global Address and Port Mapping High-Water-Mark Threshold Event . . . . .	45
5.3.2.4. Subscriber-Specific Address and Port Mapping High-Water-Mark Threshold Event . . . . .	45
5.3.3. Encoding of Limit Events . . . . .	46
5.3.3.1. Global Address Mapping Limit Exceeded . . . . .	46
5.3.3.2. Global Address and Port Mapping Limit Exceeded . . . . .	47
5.3.3.3. Global Limit On Number of Active Hosts Exceeded . . . . .	48
5.3.3.4. Subscriber-Specific Limit On Number of Address and Port Mappings Exceeded . . . . .	49
5.3.3.5. Pending Fragment Limit Exceeded . . . . .	50
6. Management Considerations . . . . .	51
6.1. General Requirements For Control Of Logging . . . . .	51
6.1.1. Configuration of PRI Value . . . . .	51
6.1.2. Ability For Each Collector To Detect Lost Event Reports . . . . .	52
6.1.3. Ability To Rate Limit Or Disable Event Reports . . . . .	52
6.2. Setting Limits and Thresholds . . . . .	53
6.3. Other Management Requirements . . . . .	54
7. Security Considerations . . . . .	55
8. IANA Considerations . . . . .	55
9. References . . . . .	58
9.1. Normative References . . . . .	58
9.2. Informative References . . . . .	60
Authors' Addresses . . . . .	61

## 1. Introduction

This document deals with logging of NAT activity in two categories: NAT translations and NAT resource usage.

Operators already need to record the addresses assigned to subscribers at any point in time, for operational and regulatory reasons. When operators introduce NAT devices that support address sharing (e.g., Carrier Grade NATs (CGNs)) into their network, additional information has to be logged. This document and [I-D.behave-ipfix-nat-logging] are provided in order to standardize the events and parameters to be recorded, using SYSLOG [RFC5424] and IPFIX [RFC7011] respectively. The same content is proposed to be logged by both documents.

In addition to records of subscriber activity, some operators use logs to indicate when utilization of critical resources is approaching or has reached limits set by the operator or implementation. This document and the IPFIX document therefore provide logs in two categories: thresholds exceeded and limits exceeded. Operators have the alternative to receive the threshold limits as SNMP notifications (see the NAT MIB [I-D.behave-NAT-MIB]).

Detailed logging requirements will vary depending on the context in which they are used. For example, different methods for transition from IPv4 to IPv6 require different events and different parameters to be logged. Section 2 covers this topic.

Section 3 provides a detailed description of the events that need logging and the parameters that may be required in the logs. Section 3.1 describes events related to subscriber activity, Section 3.2 covers threshold events, and Section 3.3 covers events where hard limits have been reached.

The use of SYSLOG [RFC5424] has advantages and disadvantages compared with the use of IPFIX [RFC7011]. Section 4 provides a statement of applicability for the SYSLOG approach.

Section 5 specifies SYSLOG record formats for logging of the events and parameters described in Section 3. Section 5.1 describes the SYSLOG header format for each report, Section 5.2 lists and describes the encoding of parameters that can appear in the logs, and Section 5.3 specifies the encoding of the body of each event report. The definitions provide the flexibility to vary actual log contents based on the requirements of the particular deployment.

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

This document makes frequent reference to the NAT MIB. That reference is to the document [I-D.behave-NAT-MIB].

This document makes frequent reference to NAT behaviours defined in [RFC4784]. In particular it refers to

- o the recommended pooling behaviour "pooled" and its contrary pooling behaviour "arbitrary"; and
- o the recommended mapping behaviour "endpoint-independent" and its contrary mapping behaviour "endpoint-dependent".

This document uses the term "address mapping" to denote an association between an internal IP address and an IP address in a selected external realm. See Section 2.2 for a further discussion of this process.

The natMapIntAddrTable in the NAT MIB provides details on all currently active address mappings. Note that this table is applicable only when NAT pooling behaviour is "paired".

This document uses the [RFC4787] term "address and port mapping" to denote a three-tuple association between an internal IP address and port and an IP address and port in a selected external realm, or between an internal <IP address, ICMP identifier> pair and an <IP address, ICMP identifier> pair in the selected realm. For implementations which maintain a Binding Information Base (BIB) (as described in Section 2 of [RFC6146], for example), the content of a BIB entry is an address and port mapping.

The natMappingTable in the NAT MIB provides details on all currently active address and port mappings.

This document uses the term "session" as it is defined in [RFC2663], Section 2.3. From the point of view of this document, session creation involves the combination of a source address and port mapping with a mapping between internal and external destination address and port to create a full five-tuple mapping.

Except where a clear distinction is necessary, this document uses the abbreviation "NAT" to encompass both Network Address Translation (NAT

in the strict sense) and Network Address and Port Translation (NAPT). The event report descriptions provided in this document apply to NAPT, and can be simplified for pure NAT operation.

To match the terminology used by the NAT MIB, this document uses the term "subscriber" to denote any device being served by the NAT, whether individual host or customer edge router. That is, despite the carrier-oriented terminology, the intended scope of applicability of this document is both to NATs in the carrier network and managed NATs in the customer network.

Finally, with two exceptions, when the terms "source" or "destination" are used below, they denote the source and destination of packets that are flowing from the internal to the external realm, regardless of the direction of session establishment or the direction of flow of an individual packet. The exceptions relate to the global address and port mapping limit event and the pending fragment limit event, when the actual source and destination addresses in the header of the packet that hit the limit are reported.

## 2. Deployment Considerations

### 2.1. Static and Dynamic NATs

A NAT controls a set of resources in the form of one or more pools of external addresses. If the NAT also does port translation (i.e., it is a NAPT), it also controls the sets of UDP and TCP port numbers and ICMP identifiers associated with each external address.

Logging requirements for a NAT depend heavily on its resource allocation strategy. NATs can be classed as static or dynamic depending on whether the resources provided to individual users are pre-configured or allocated in real time as the NAT recognizes new flows.

Static assignments can be logged at configuration time by the NAT or by network infrastructure. The logging volume associated with static assignments will be relatively low, of the order of the volume of user logons.

Dynamic assignments typically require both more detail in the logs and a higher volume of logs in total. A traditional Network Address Port Translator (NAPT) as described in [RFC3022] and following the recommendations of [RFC4787] and [RFC5382] will generate a new address and port mapping each time it encounters a new internal <address, port> combination.

For statistical reasons, static assignments support lower address sharing ratios than fully dynamic assignments as exemplified by the traditional NAT. The sharing ratio can be increased while restraining log volumes by assigning ports to users in multi-port increments as required rather than assigning just one port at a time. A subscriber may start with no initial allocation, or may start with an initial permanent allocation to which temporary increments are added when the initial set is all being used. See [RFC6264] and [I-D.tsou-behave-natx4-log-reduction] for details. If this strategy is followed, logging will be required only when an increment is allocated or reclaimed rather than every time an internal <address, port> combination is mapped to an external <address, port>.

## 2.2. Realms and Address Pools

A realm identifies an IP numbering space. A NAT session always maps between an internal and an external realm. In simple NAT configurations, it may be possible to identify a default internal realm and/or a default external realm for all sessions. In more complex NAT configurations a given realm may be an internal realm for some sessions and an external realm for others. Realms without subscriber sites are always external.

Address pools are associated with specific realms in their external role.

It is necessary to define multiple realms when the NAT supports overlapping IP numbering spaces. In such a case, the NAT must determine the source realm and subscriber using additional information associated with the incoming packet. See further discussion in Section 2.4.

### 2.2.1. Address Pools

An address pool is a mechanism for configuring the set of addresses to which a given internal address can be mapped in a given realm. The pool may be used simply to ration the available addresses within that realm, or may be selected for other reasons such as to add additional semantics (e.g., type of service required) to the external address within the target realm. Clearly a given internal address may be mapped into more than one address pool at a given time.

The model of an address pool assumed in this document and in the NAT MIB is that the pool offers a fixed range of port/ICMP identifier values, the same over all addresses within the pool. How these are allocated to individual mappings depends on the pooling behaviour. With a pooling behaviour of "arbitrary", the NAT can select any address in the pool with a free port value for the required protocol

and map the internal address to it. With the recommended pooling behaviour of "paired", the NAT restricts itself to finding a free port at the address to which the internal address is already mapped, if there is one.

From this description, one can see that ports are a limited resource, subject to exhaustion at the pool level and, with "paired" behaviour, at the level of the individual address. Log events are defined in Section 3.2.1 that allow monitoring of port utilization at the pool level. Section 6.2 discusses how the thresholds for triggering these events should be varied depending on pooling behaviour.

### 2.3. NAT Logging Requirements For Different Transition Methods

A number of transition technologies have been or are being developed to aid in the transition from IPv4 to IPv6. 6rd [RFC5969] and DS-Lite [RFC6333] are at the deployment stage. Several 'stateless' technologies: Public IPv4 over IPv6 [RFC7040], MAP-E [I-D.softwire-map], and Lightweight 4over6 [I-D.softwire-lw4over6] have seen experimental deployment and are in the process of being standardized at the time of writing of this document.

Of the technologies just listed, 6rd and Public IPv4 over IPv6 do not involve NATs and hence need not be considered further. The other techniques involve NAT at the customer edge, at the border router, or both, and hence are in scope.

A DS-Lite Address Family Transition Router (AFTR) includes a large-scale session-stateful NAT44 processing potentially millions of sessions per second. The special character of AFTR operation over that of a traditional NAT44 is that the source IPv4 addresses of the internal hosts will not be unique. As a consequence, identification of the realm and subscriber from which the packet was sent needs to include an additional identifier associated with the subscriber host. For basic DS-Lite, this will be the IPv6 address used to encapsulate the packets outgoing from the host. See Section 6.6 of [RFC6333]. For gateway-initiated DS-Lite [RFC6674], two identifiers are needed: an identifier of the softwire from the gateway to the NAT, and an identifier associated with the incoming tunnel to the gateway.

The DS-Lite customer edge equipment (the 'B4') may also perform NAT44 functions, similar to the functions performed by traditional NAT44 devices.

As a NAT44, the DS-Lite AFTR may be fully dynamic, or may allocate ports in increments as described in the previous section.

Lightweight 4over6 [I-D.softwire-lw4over6] and MAP-E [I-D.softwire-map] both require NAT44 operation at the customer edge. In both cases the resource allocation strategy is static. Thus any logging of resource allocation for these two transition techniques can be done by the network at configuration time.

#### 2.4. Subscriber Identification

The ability to identify the particular subscriber involved in an event is required for the events defined in Section 3.1, and desirable for technician follow-up for those defined in Section 3.2.4 and Section 3.3.

As mentioned above, in some NAT configurations the source address is insufficient to identify an individual subscriber because of overlapping address space, and additional information is required. For example, if the NAT supports DS-Lite [RFC 6333], the source address of incoming packets from DS-Lite subscribers will always be in the range 192.0.0/29. The additional information required in this case is the IPv6 address of the encapsulating header.

The natSubscribersTable in the NAT MIB contains the additional information needed, if any, to identify each subscriber. Thus it is sufficient to include the index to this table in the event report to provide the needed identification. However, this implies that for full interpretation of the event report, the configuration information stored in the natSubscribersTable must be stored (along with AAA information relating the additional identifiers to the subscriber profiles, which must be stored in any event). To relieve the operator of the need to store the configuration data (given that the logs may be needed months or years after they were recorded), the reports specified in Section 3.1 include the additional identifying information that is found in the natSubscribersTable.

This document standardizes the presentation of the following possible additional classifying information within NAT-related log reports:

- o interface index [RFC2863];
- o VLAN index [RFC4363];
- o VPN identifier [RFC4265];
- o DS-Lite encapsulating IPv6 address [RFC6333].

Which of these is actually used in a given NAT depends on implementation and deployment.

Gateway-Initiated DS-Lite [RFC6674] identifiers could also be specified, but it seems premature to do so because it is not clear which of the variety of possibilities presented in Section 6 and Appendix A.2 of [RFC6674] are actually being deployed.

## 2.5. The Port Control Protocol (PCP)

The Port Control Protocol (PCP) [RFC6887] and its port set extension [I-D.pcp-port-set] can be viewed as a way to provision ports by other means. However, PCP can be invoked on a per-flow basis, so the volume of logs generated by a PCP server can be closer to the volume associated with a fully dynamic NAT. The volume really depends on how PCP is being used in a specific network.

## 2.6. Logging At the Customer Edge

Logging at the customer edge (or at the ISP edge for NATs protecting the ISP's internal networks) may be done by the customer for purposes of internal management, or by the ISP for its own administrative and regulatory purposes. Given the likelihood of a high internal community of interest, it is possible but unlikely that a NAT at the edge of a large enterprise network processes a number of new packet flows per second which is comparable to the volume handled by a carrier grade NAT. Most customer edge NATs will handle a much smaller volume of flows.

## 3. NAT-Related Events and Parameters

The events which follow were initially gleaned, in the words of the authors of [I-D.behave-ipfix-nat-logging], from [RFC4787] and [RFC5382]. Some details were subsequently informed by the discussion in Section 2 and by provisions within the NAT MIB. Section 4 of [RFC6888] also provides a brief statement of logging requirements for carrier grade NATs.

In SYSLOG, the timestamp and the event type will appear in the log header rather than as an explicit part of the structured data portion of the log. Hence they are omitted from the parameter tabulations that follow.

Parameters marked CONDITIONAL are REQUIRED under some circumstances but not others. Details are provided for each event.

### 3.1. Events Relating To Allocation Of Resources To Hosts

Setting up a NAT session proceeds in a series of logical steps: creation of an address mapping, creation of an address and port mapping, and finally, creation of the session.



The reports corresponding to these three steps are defined in Section 3.1.1, Section 3.1.2, and Section 3.1.3 respectively. Which of these reports is enabled depends on the NAT implementation and operator preferences, subject to the considerations of the next paragraph.

If the NAT implements the recommended pooling behaviour of "paired", address mapping creation is an event distinct in general from the creation of a subsequent address and port mapping based on that address mapping. However, if the pooling behaviour is "arbitrary" [RFC4787], the two events occur simultaneously and there is no point in reporting both. Similarly, if the NAT implements the recommended mapping behaviour of "endpoint-independent mapping", the two events of address and port mapping creation and session creation based on that mapping are distinct and may meaningfully be reported separately. However, if the mapping behaviour is "endpoint-dependent", the two events occur simultaneously and it is only meaningful to report session creation.

The fourth report type in this section describes the bulk allocation of ports to an address mapping, which the NAT may implement if the pooling behaviour is "paired" [RFC4787]. It, along with the other reports, is needed to provide complete accountability for resources allocated to the subscriber.

#### 3.1.1. NAT Address Mapping Creation and Deletion

Two specific events are provided:

- o NAT address mapping creation;
- o NAT address mapping deletion.

Implementations MUST NOT report these events unless pooling behaviour is "paired".

Address mapping is discussed in detail in Section 2.2.

One address mapping creation event is associated with potentially many succeeding address and port mapping creation events, as individual port values are mapped for specific protocols. Similarly, an address mapping deletion event may be associated with potentially many address and port mapping deletion events, which may have preceded it over a period of time or may occur at the same time as a result of the address unbinding.

The address mapping events take the following specific parameters:

- o NAT instance identifier (CONDITIONAL);
- o Source subscriber index (MANDATORY);
- o Additional source subscriber classifier value as recognized at the ingress to the internal realm (CONDITIONAL);
- o Internal realm (CONDITIONAL);
- o Internal address type (MANDATORY);
- o Internal source IP address (MANDATORY);
- o External realm (CONDITIONAL);
- o External address type (MANDATORY);
- o External source IP address (MANDATORY);
- o Trigger for address mapping creation or deletion (OPTIONAL):
  - \* outgoing packet;
  - \* administrative action (e.g., via the Port Control Protocol [RFC6887]); or
  - \* autonomous action of the NAT.

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.
- o Additional source subscriber classifier value REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o Internal or external realm REQUIRED if not the default internal or external realm, respectively, else MAY appear.

### 3.1.2. NAT Address and Port Mapping Creation and Deletion

The address and port mapping creation or deletion event reports the addition or deletion of an address and port mapping as defined in Section 1.1. If the implementation maintains a Binding Information Base (BIB), this is equivalent to the creation or deletion of a BIB entry. Implementations MUST support the generation of the address and port mapping creation/deletion event reports if they implement

the recommended mapping behaviour "endpoint-independent". They MAY support reporting of these events in the contrary case.

The address and port mapping creation/deletion event report provides the same information as the session creation/deletion event, except for the destination-related fields and (in general) timestamp values in the latter. With "endpoint-independent" mapping behaviour, one address and port mapping creation event is associated with potentially many succeeding session creation events. Similarly, an address and port mapping deletion event will be associated with potentially many session deletion events, which may have preceded it over a period of time or may occur at the same time as a result of the address and port mapping deletion.

Operators should disable the reporting of address and port mapping creation and deletion events when destination logging is enabled, because of the redundancy between the address and port mapping and session event reports. However, if destination logging is disabled and the NAT uses the recommended "endpoint-independent" mapping behaviour, it is the session events that are redundant and should be disabled.

The following specific events are defined:

- o NAT address and port mapping creation
- o NAT address and port mapping deletion

These take the same parameters for all types of NAT. The internal realm, subscriber-identifying information, internal source IP address, external realm, and external source IP address capture the underlying address mapping. The port values and protocol are unique to the address and port mapping.

The parameters for the address and port mapping creation/deletion event are:

- o NAT instance identifier (CONDITIONAL);
- o Source subscriber index (MANDATORY);
- o Additional source subscriber classifier value as recognized at the ingress to the internal realm (CONDITIONAL);
- o Internal realm (CONDITIONAL);
- o Internal address type (MANDATORY);

- o Internal source IP address (MANDATORY);
- o Internal source port or ICMP identifier (MANDATORY);
- o External realm (CONDITIONAL);
- o External address type (MANDATORY);
- o External source IP address (MANDATORY);
- o External source port or ICMP identifier (MANDATORY);
- o Protocol identifier (MANDATORY);
- o Trigger for address and port mapping creation or deletion (OPTIONAL):
  - \* outgoing packet received;
  - \* incoming packet received;
  - \* administrative action (e.g., via the Port Control Protocol [RFC6887]); or
  - \* deletion of the underlying address mapping (applicable only if pooling behaviour is "paired" [RFC4787]).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.
- o Additional source subscriber classifier value REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o Internal or external realm REQUIRED if not the default internal or external realm, respectively, else MAY appear.

### 3.1.3. NAT Session Creation and Deletion

A NAT session creation or deletion event is logged when a address and port mapping is further bound to or unbound from a specific destination address and port in the external realm. One to many sessions can be based on the same address and port mapping.

Implementations MUST provide a means for the operator to specify whether destination information is to be included in the reports of these events (see discussion below).

The following specific events are defined:

- o NAT session creation
- o NAT session deletion

These take the same parameters for all types of NAT. Parameters "internal realm" through "protocol identifier" capture the underlying address and port mapping. Subsequent parameters capture the destination address and destination subscriber identity (if applicable).

The parameters for the session creation/deletion event are:

- o NAT instance identifier (CONDITIONAL);
- o Internal source subscriber index, equal to the natSubscriberIndex value in the natSubscribersTable in the NAT MIB (MANDATORY);
- o Additional internal subscriber classifier value (CONDITIONAL);
- o Internal realm (CONDITIONAL);
- o Internal address type (MANDATORY);
- o Internal source IP address (MANDATORY);
- o Internal source port or ICMP identifier (MANDATORY);
- o External realm (CONDITIONAL);
- o External address type (MANDATORY);
- o External source IP address (MANDATORY);
- o External source port or ICMP identifier (MANDATORY);
- o Protocol identifier (MANDATORY);
- o Internal destination IP address (CONDITIONAL);
- o Internal destination port or ICMP identifier (CONDITIONAL);
- o Destination subscriber index (CONDITIONAL);

- o Additional destination subscriber classifier value as recognized at the ingress to the external realm (CONDITIONAL);
- o External destination IP address (CONDITIONAL);
- o External destination port or ICMP identifier (CONDITIONAL);
- o Trigger for session creation or deletion (OPTIONAL):
  - \* outgoing packet received;
  - \* incoming packet received;
  - \* administrative action (e.g., via the Port Control Protocol [RFC6887]); or
  - \* deletion of the underlying address and port mapping (applicable only if the NAT mapping behaviour is "endpoint-independent").

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.
- o Additional source subscriber classifier value REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o Internal or external realm REQUIRED if not the default internal or external realm, respectively, else MAY appear.
- o Internal destination address and port REQUIRED if destination logging is enabled and these need to be remapped to external destination address and port. Otherwise, if destination logging is disabled, they MUST NOT appear, and if destination logging is enabled, they SHOULD NOT appear because of redundancy.
- o External destination subscriber index REQUIRED if destination logging is enabled and the destination is a subscriber served by the NAT, else MUST NOT appear.
- o Additional external subscriber classifier value REQUIRED if destination logging is enabled and the destination is a subscriber served by the NAT and the external destination address is not enough to identify the external destination subscriber unambiguously, else MUST NOT appear.

- o External destination address and port REQUIRED if destination logging is enabled, else MUST NOT appear.

#### 3.1.3.1. Destination Logging

The logging of destination address and port is undesirable, for several reasons. [RFC6888] recommends against destination logging because of the privacy issues it creates. From an operator's point of view, destination logging is costly not just because of the volume of logs it will generate, but because the NAT now has to carry additional session state so that it only needs to log once per session between two transport end points rather than logging every packet. Finally, [RFC4787], etc. recommend the use of endpoint-independent mapping to maximize the ability of applications to operate through the NAT. In that case, most of the contents of the session creation event report will be repeated for one destination after another.

One possibility is that the implementation provides the operator with the ability to log destinations only for particular subscribers or particular mapped addresses on a special study basis. This facility could be used for trouble-shooting or malicious activity tracing in particular cases as required. If such a capability is provided, the implementation MUST report destination information for sessions matching the specified criteria, but MUST NOT report these events for other sessions.

#### 3.1.4. Port Range Allocation and Deallocation

This event is recorded at a hybrid NAT whenever the set of ports allocated to a given address mapping changes. It is assumed that when ports are allocated in bulk, the same values are allocated for all protocols.

The following specific events are defined:

- o Port range allocation;
- o Port range deallocation.

The parameters for these events are:

- o NAT instance identifier (CONDITIONAL);
- o Source subscriber index (MANDATORY);
- o Additional source subscriber classifier value as recognized at the ingress to the internal realm (CONDITIONAL);

- o Internal realm (CONDITIONAL);
- o Internal address type (MANDATORY);
- o Internal source IP address (MANDATORY);
- o External realm (CONDITIONAL);
- o External address type (MANDATORY);
- o External source IP address (MANDATORY);
- o Lowest port number of the range being allocated or deallocated (MANDATORY).
- o Highest port number of the range being allocated or deallocated (MANDATORY).
- o Trigger for port range allocation or deallocation (OPTIONAL):
  - \* outgoing packet received;
  - \* incoming packet received;
  - \* administrative action (e.g., via the Port Control Protocol [RFC6887]); or
  - \* autonomous action of the NAT.

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.
- o Additional source subscriber classifier value REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o Internal or external realm REQUIRED if not the default internal or external realm, respectively, else MAY appear.

It will be necessary to use multiple event reports to report more complex allocations or deallocations.



### 3.2. Threshold Events

The events of this section are based on thresholds set by the operator within the NAT MIB. Cross-references to the associated MIB objects are provided for each event. With the exception of the address pool low-water-mark event, the threshold events provide early warning of potential dropped packets due to resource exhaustion or administrator-imposed limits.

#### 3.2.1. Address Pool High- and Low-Water-Mark Threshold Events

Two specific events provide reports on address pool utilization:

- o High-water-mark threshold reached or exceeded;
- o Low-water-mark threshold reached or under-shot.

Depending on deployment the operator has the alternative of using the SNMP notifications `natNotifPoolWater-MarkHigh` and `natNotifPoolWater-MarkLow` defined in the NAT MIB rather than logging these events.

Address pools are discussed in Section 2.2.1. The `natPoolTable` object in the NAT MIB provides access to parameters describing the utilization level of address-port combinations within a given pool. Since a new mapping cannot be allocated unless a mappable address and a free port on that address are available, it is important to know when the available set of address-port combinations within a given pool is nearing exhaustion. Hence the `natPoolTable` contains a high-water-mark threshold settable by the operator. An address pool high-water-mark event report is generated when a new mapping into the pool is requested and aggregate address-port utilization is equal to or greater the threshold.

Similarly it can be of interest to know when a pool is under-utilized. Hence the `natPoolTable` also provides a low-water-mark threshold. An address pool low-water-mark event report is generated when aggregate address-port utilization is equal to or less than the low-water-mark threshold.

Section 6.2 discusses factors affecting the choice of the threshold values.

The high-water-mark threshold event provides a warning that the address-port combinations offered by the pool are nearing exhaustion. Upon exhaustion, subscribers may be unable to establish new connections because no address has enough free port values left to be allocated to an address mapping ("address exhaustion"). This applies to the case of "paired" pooling behaviour, where typically an address

will not be allocated unless it has a sufficient number of free ports. Alternatively, new connections cannot be established simply because no address in the pool has a free port number for the required protocol ("port exhaustion").

Packets triggering failed attempts to establish new connections due to address exhaustion are included in the following NAT MIB dropped packet counters:

- o globally, natResourceErrors in the natCounters table;
- o per protocol, natProtocolResourceErrors in natProtocolTable;
- o per subscriber, natSubscriberResourceErrors in natSubscribersTable.

Packets triggering failed attempts to establish new connections due to port exhaustion are counted in the following NAT MIB dropped packet counters:

- o globally, natOutOfPortErrors in the natCounters table;
- o per protocol, natProtocolOutOfPortErrors in natProtocolTable;
- o per subscriber, natSubscriberOutOfPortErrors in natSubscribersTable.

An address pool threshold event report contains the following specific parameters:

- o NAT instance identifier (CONDITIONAL);
- o Pool identifier (MANDATORY);
- o The threshold value set by the administrator (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

### 3.2.2. Global Address Mapping High-Water-Mark Threshold Event

One specific event allows monitoring of the total number of mappings between internal and external addresses:

- o Address mapping high-water-mark threshold exceeded.

Implementations MUST NOT generate this event report unless the pooling behaviour is "paired". Depending on deployment, operators can choose instead to use the SNMP notification `natNotifAddrMappings` defined in the NAT MIB.

The NAT MIB displays cumulative counts of address mappings created and removed in the `natCounters` table. When the difference between these two counters is greater than the threshold `natAddrMapNotifyThreshold` provided in the `natLimits` table the global address binding high-water-mark threshold event is reported.

The specific parameters provided by this event report are:

- o NAT instance identifier (CONDITIONAL);
- o Current number of active address mappings (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

### 3.2.3. Global Address and Port Mapping High-Water-Mark Threshold Event

One specific event allows monitoring of the total number of active address and port mappings. Where the NAT implements a BIB, this is equivalent to the total number of BIB entries.

- o address and port mapping high-water-mark threshold exceeded.

Depending on deployment, operators can choose instead to use the SNMP notification `natNotifMappings` defined in the NAT MIB.

The NAT MIB displays cumulative counts of address and port mappings created and removed in the `natCounters` table. When the difference between these two counters is greater than the threshold `natMappingsNotifyThreshold` provided in the `natLimits` table the global mapping high-water-mark threshold event is reported.

The specific parameters provided by this event report are:

- o NAT instance identifier (CONDITIONAL);
- o Current number of active address and port mappings (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

#### 3.2.4. Subscriber-Specific Address and Port Mapping Threshold Event

An event is provided to allow monitoring of the total number of active mappings per subscriber:

- o Subscriber-specific mapping high-water-mark threshold exceeded.

Depending on deployment, operators can choose instead to use the SNMP notification `natNotifSubscriberMappings` defined in the NAT MIB.

The NAT MIB displays cumulative counts of address and port mappings created and removed per subscriber in the `natSubscribersTable`. When the difference between these two counters is greater than the threshold `natSubscriberMapNotifyThresh` provided in that table the subscriber address and port mapping high-water-mark threshold event is reported.

The specific parameters provided by this event report are:

- o NAT instance identifier (CONDITIONAL);
- o Index of the affected subscriber (MANDATORY).
- o Current number of active mappings for this subscriber (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

#### 3.3. Limit-Related Events

The events of this section are generated when hard limits set by the operator are exceeded. The consequence for service will be dropped packets. As with the threshold events, the description of each report includes cross-references to the associated MIB objects.

##### 3.3.1. Global Address Mapping Limit Exceeded

The global address mapping limit exceeded event is reported when a new address mapping is requested but the total number of address mappings would exceed an administrative limit if it were added. The limit is given by object `natLimitAddressMappings` in the `natLimits` table of the NAT MIB. MIB counters giving number of packets dropped due to resource limitations including this one are:

- o globally, natResourceErrors in the natCounters table;
- o per protocol, natProtocolResourceErrors in natProtocolTable;
- o per subscriber, natSubscriberResourceErrors in natSubscribersTable.

Implementations MUST NOT generate this event report unless the pooling behaviour is "paired". Depending on deployment, operators can choose instead to use the SNMP notification natNotifAddrMappings defined in the NAT MIB.

The parameters for this event are:

- o NAT instance identifier (CONDITIONAL);
- o Index of the affected subscriber (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

The subscriber index is provided to allow the operator to correlate the event with any subscriber complaints or possible abuse.

### 3.3.2. Global Address and Port Mapping Limit Exceeded

The global address and port mapping limit exceeded event is reported when a new address and port mapping is requested but the total number of address and port mappings would exceed an administrative limit if it were added. The limit is given by object natLimitMappings in the natLimits table of the NAT MIB. MIB counters giving number of packets dropped due to resource limitations including this one are:

- o globally, natResourceErrors in the natCounters table;
- o per protocol, natProtocolResourceErrors in natProtocolTable;
- o per subscriber, natSubscriberResourceErrors in natSubscribersTable.

The parameters for this event are:

- o NAT instance identifier (CONDITIONAL);
- o Index of the internal subscriber (CONDITIONAL);

- o Index of the external subscriber (CONDITIONAL);
- o Source realm of the triggering packet (MANDATORY);
- o Incoming packet header IP address type (CONDITIONAL);
- o Incoming packet source IP address (CONDITIONAL).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.
- o The index of the internal subscriber is REQUIRED if the mapping was triggered by a packet outgoing from the internal to the external realm, else MUST NOT appear.
- o The index of the external subscriber is REQUIRED if the mapping was triggered by a packet incoming from a subscriber served by the NAT and located in the external realm (i.e., using an address mapping created previously by the internal subscriber), else MUST NOT appear.
- o The address type and source IP address from the initiating packet are REQUIRED if the mapping was triggered by a packet incoming from a purely external realm (i.e., using an address mapping created previously by the internal subscriber), else MAY appear.

The subscriber index or packet source address is provided to allow the operator to correlate the event with any subscriber complaints or possible abuse.

### 3.3.3. Global Limit On Number of Active Hosts Exceeded

The global limit on number of active hosts exceeded event is reported when an address mapping is requested (at least at the logical level) for a host with no previous active mappings, but the total number of active hosts would exceed an administrative limit if it were added. The limit is given by object natLimitSubscribers in the natLimits table of the NAT MIB. MIB counters giving number of packets dropped due to resource limitations including this one are:

- o globally, natResourceErrors in the natCounters table;
- o per protocol, natProtocolResourceErrors in natProtocolTable;
- o per subscriber, natSubscriberResourceErrors in natSubscribersTable.

The parameters for this event are:

- o NAT instance identifier (CONDITIONAL);
- o Index of the affected subscriber (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

The subscriber index is provided to allow the operator to correlate the event with any subscriber complaints.

#### 3.3.4. Subscriber-Specific Limit On Number of Address and Port Mappings Exceeded

The subscriber-specific limit on number of address and port mappings exceeded event is reported when a new mapping is requested, but the total number of active mappings for that subscriber would exceed an administrative limit if it were added. The limit is given by object natSubscriberLimitMappings in natSubscribersTable in the NAT MIB. MIB counters giving number of packets dropped due to resource limitations including this one are:

- o globally, natResourceErrors in the natCounters table;
- o per protocol, natProtocolResourceErrors in natProtocolTable;
- o per subscriber, natSubscriberResourceErrors in natSubscribersTable.

The parameters for this event are:

- o NAT instance identifier (CONDITIONAL);
- o Index of the affected subscriber (MANDATORY).

Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.

The subscriber index is provided to allow the operator to take administrative action or to correlate the event with any subscriber complaints or possible abuse.

### 3.3.5. Global Limit On Number Of Fragments Pending Reassembly Exceeded

The global limit on number of fragments pending reassembly exceeded event is reported when a new fragment is received and the number of fragments currently awaiting reassembly is already equal to an administrative limit. That limit is given by the `natLimitFragments` object in the `natLimits` table. This event **MUST NOT** be reported unless the NAT supports the "receive fragments out of order" behavior [RFC4787]. MIB counters giving number of packets dropped due to resource limitations including this one are:

- o globally, `natResourceErrors` in the `natCounters` table;
- o per protocol, `natProtocolResourceErrors` in `natProtocolTable`;
- o per subscriber, `natSubscriberResourceErrors` in `natSubscribersTable`.

The parameters for this event provide the contents of the IP header of the received fragment that triggered it. If the source of the packet is a subscriber served by the NAT and the subscriber index can be determined, it **MUST** also be included.

- o NAT instance identifier (CONDITIONAL);
- o Source realm of the packet (MANDATORY);
- o Packet header IP address type (MANDATORY);
- o Packet source IP address (MANDATORY);
- o Packet destination IP address (MANDATORY);
- o Source subscriber index (CONDITIONAL).

Conditions:

- o NAT instance identifier **REQUIRED** if device supports more than one instance, else **MAY** appear.
- o Source subscriber index **REQUIRED** if the source of the packet is a subscriber served by the NAT and can be determined, else **MUST NOT** appear.



#### 4. SYSLOG Applicability

The primary advantage of SYSLOG is the human readability and searchability of its contents. In addition, it has built-in priority and other header fields that allow for separate routing of reports requiring management action. Finally, it has a well-developed underpinning of transport and security protocol infrastructure.

SYSLOG presents two obstacles to scalability: the fact that the records will typically be larger than records based on a binary protocol such as IPFIX, and, depending on the architectural context, the reduced performance of a router that is forced to do text manipulation in the data plane. One has to conclude that for larger message volumes, IPFIX should be preferred as the reporting medium on the NAT itself. It is possible that SYSLOG could be used as a back-end format on an off-board device processing IPFIX records in real time, but this would give a limited boost to scalability. One concern expressed in list discussion is that when the SYSLOG formatting process gets overloaded records will be lost.

As a result, the key question is what the practical cutoff point is for the expected volume of SYSLOG records, on-board or off-board the NAT. This obviously depends on the computing power of the formatting platform, and also on the record lengths being generated.

Information has been provided to the BEHAVE list at the time of writing to the effect that one production application is generating an average of 150,000 call detail records per second, varying in length from 500 to 1500 bytes. Capacities several times this level have been reported involving shorter records, but this particular application has chosen to limit the average in order to handle peaks.

As illustrated by the example in Section 5.3.1.3, if destination logging is enabled, typical record sizes for session event logs are in the order of 300 bytes, so throughput capacity should be higher than in the call detail case for the same amount of computing power. However, note that bursts of session deletion events may occur as a result of deletion of the underlying mapping or address mapping.

In private communication, a discussant has noted a practical limit of a few hundred thousand SYSLOG records per second on a router.

#### 5. SYSLOG Record Format For NAT Logging

This section describes the SYSLOG record format for NAT logging in terms of the field names used in [RFC5424] and specified in Section 6 of that document. In particular, this section specifies values for the APP-NAME and MSGID fields in the record header, the SD-ID

identifying the STRUCTURED-DATA section, and the PARAM-NAMES and PARAM-VALUE types for the individual possible parameters within that section. The specification is in three parts, covering the header, encoding of the individual parameters, and encoding of the complete log record for each event type.

#### 5.1. SYSLOG HEADER Fields

Within the HEADER portion of the SYSLOG record, the priority (PRI) level is subject to local policy, but a Severity value of 6 (Informational) is suggested for the events relating to creation and deletion of sessions, mappings, address mappings, and port allocation, combined with a suitable Facility value in the range 16-23 (local use) to ensure routing to a secure collector. The Facility value(s) for the threshold and limit events will presumably be chosen to route them to maintenance for immediate action and/or to provisioning for less urgent consideration. The suggested value of Severity by event type is shown in Table 1, but in practice has a clear dependency on the context within which the NAT is operating.

The TIMESTAMP field SHOULD be expressed with sufficient precision to distinguish non-simultaneous event occurrences, subject to the accuracy of the local clock. This specification does not assume the ability to correlate the events reported by the subject device with events recorded by other devices, although that may be required for other reasons. Hence from the point of view of this specification only relative rather than absolute accuracy is of interest.

The HOSTNAME header field MUST identify the NAT device. The value of the HOSTNAME field is subject to the preferences given in Section 6.2.4 of [RFC5424].

The values of the APP-NAME and MSGID fields in the record header determine the semantics of the record. To simplify log collection procedures, the APP-NAME value "NAT" MUST be used for the event reports specified in Section 5.3.1, the APP-NAME value "NATTHR" MUST be used for the event types defined in Section 5.3.2, and the APP-NAME value "NATLIM" MUST be used for the event types defined in Section 5.3.3.

The MSGID values indicate the individual events. They are listed in Table 1 for each of the events defined in Section 3. The table also shows the SD-ID value used to label the event-specific STRUCTURED-DATA element.

Event	APP-NAME	MSGID	Severity	SD-ID
NAT address mapping creation	NAT	AMADD	6 info	namap
NAT address mapping deletion	NAT	AMDEL	6 info	namap
NAT address and port mapping creation	NAT	APMADD	6 info	napmap
NAT address and port mapping deletion	NAT	APMDEL	6 info	napmap
NAT session creation	NAT	SADD	6 info	nsess
NAT session deletion	NAT	SDEL	6 info	nsess
Port range allocation	NAT	PTADD	6 info	nprng
Port range deallocation	NAT	PTDEL	6 info	nprng
Address pool high threshold	NATTHR	POOLHT	4 warning	npool
Address pool low threshold	NATTHR	POOLLT	6 info	npool
Global address mapping high threshold	NATTHR	GAMHT	4 warning	ngamht
Global address and port mapping high threshold	NATTHR	GAPMHT	4 warning	ngapmht
Subscriber-specific mapping high threshold	NATTHR	SAPMHT	5 notice	nsapmht
Global address mapping limit	NATLIM	GAMLIM	3 error	ngaml
Global address and port mapping limit	NATLIM	GAPMLIM	3 error	ngapml
Global active subscriber limit	NATLIM	GSLIM	3 error	ngsl
Subscriber-specific address and port mapping limit	NATLIM	SAPMLIM	5 notice	nsapml
Pending fragment limit	NATLIM	FRAG	4 warning	nfpkt

Table 1: Recommended MSGID Encodings and Default Severity Values for the Events Defined In Section 3

## 5.2. Parameter Encodings

This section describes how to encode the individual parameters that can appear in NAT-related logs. The parameters are taken from the event descriptions in Section 3. The PARAM-NAMES, brief

descriptions, and encoding are listed in Table 2, with reference to the general and special case encoding rules which follow.

PARAM-NAME	Description	Encoding
	Miscellaneous	
NATINST	NAT instance identifier	Text
TRIG	Trigger for event	Special case
	Subscriber-identifying information	
SSUBIX	Source subscriber index	32-bit field
SIFIX	Source subscriber ingress interface index list	Special case
SVLAN	Source subscriber ingress VLAN index	32-bit field
SVPN	Source subscriber ingress VPN Id	Special case
SV6ENC	Source subscriber ingress RFC6333 encapsulating IPv6 address	IPv6 address
DSUBIX	Destination subscriber index	32-bit field
DIFIX	Destination subscriber ingress interface index list	Special case
DVLAN	Destination subscriber ingress VLAN index	32-bit field
DVPN	Destination subscriber ingress VPN Id	Special case
DV6ENC	Destination subscriber ingress RFC6333 encapsulating IPv6 address	IPv6 address
	Internal packet description	
IRLM	Internal realm	Text
IATYP	Internal IP address type	"IPv4" or "IPv6"
ISADDR	Internal source IP address value	IPv4 or IPv6 address
ISPORT	Internal source port or ICMP identifier value	16-bit field
IDADDR	Internal destination IP address value	IPv4 or IPv6 address

IDPORT	Internal destination port or ICMP identifier value	16-bit field
PROTO	Protocol identifier (from the IANA Assigned Internet Protocol Numbers registry)	8-bit field
	External (mapped) packet description	
XRLM	External realm	Text
XATYP	External IP address type	"IPv4" or "IPv6"
XSADDR	External source IP address value	IPv4 or IPv6 address
XSPORT	External source port or ICMP identifier value	16-bit field
XDADDR	External destination IP address value	IPv4 or IPv6 address
XDPORT	External destination port or ICMP identifier value	16-bit field
	Port range description	
PORTMN	Port range lowest value	16-bit field
PORTMX	Port range highest value	16-bit field
	Values related to thresholds	
POOLID	Address pool identifier	32-bit field
POOLHW	Address pool high water mark threshold	Unsigned decimal
POOLID	Address pool low water mark threshold	Unsigned decimal
GAMCNT	Current global number of address mappings	Unsigned decimal
GAPMCNT	Current global number of address and port mappings	Unsigned decimal
SAPMCNT	Current subscriber-specific number of address and port mappings	Unsigned decimal
	Specific incoming packet description	
PSRLM	Packet source realm	Text
PATYP	Packet IP address type	"IPv4" or

PSADDR	Packet source IP address	"IPv6" IPv4 or IPv6 address
PDADDR	Packet destination IP address	IPv4 or IPv6 address

Table 2: Parameters Used In NAT-Related Log Reports

#### 5.2.1. General Encoding Rules

All fields MUST be encoded as 7-bit US ASCII [US-ASCII].

Complete IPv6 addresses MUST be presented according to the rules specified in Sections 4 and 5 of [RFC5952], without a succeeding prefix length. The Section 5 rules MUST NOT be applied unless the address can be distinguished as having an IPv4 address embedded in the lower 32 bits solely from the IPv6 prefix portion (e.g., based on well-known prefix, flag), without external information. In such cases, the IPv6 prefix portion MUST be presented according to the Section 4 rules. Stand-alone IPv6 prefixes (i.e., outside of special addresses) MUST be presented according to the Section 4 rules, with the slash character (/) appended, followed by a decimal value with leading zeroes suppressed, giving the prefix length (0 to 127) in bits.

Similarly, complete IPv4 addresses MUST be presented in dotted decimal format, with no succeeding prefix length. IPv4 prefixes MUST be presented as if they were full addresses, with the slash character (/) appended, followed by a decimal value with leading zeroes suppressed, giving the prefix length (0 to 31) in bits.

N-bit fields and unsigned decimals are both presented as unsigned decimal integers with no leading zeroes.

#### 5.2.2. Special Cases

Three special cases are identified in Table 2: encoding of the interface index list (PARAM-NAMEs SIFIX and DIFIX), encoding of the VPN identifier (PARAM-NAMEs SVPN and DVPN), and encoding of the trigger for resource allocation events (PARAM-NAME TRIG).

The interface index list is presented as a series of individual interface indexes separated by commas, e.g., SIFIX="5,15". Each individual interface index is presented as a 32-bit field (i.e., as an unsigned decimal integer with no leading zeroes).

The VPN Identifier is standardized in [RFC2685], and consists of a three octet VPN Authority (Organizationally Unique Identifier, OUI) followed by a four octet VPN index identifying the VPN according to OUI. For SYSLOG, the OUI portion is presented as a string of six hexadecimal digits in lower case. The VPN index is presented as a 32-bit field. A colon (:) is used to separate the OUI from the succeeding index value. The OUI and separator MAY be omitted. If so, the applicable OUI is the default value for the NAT instance.

The trigger is an enumeration of text values which were not spelled out in the table itself for lack of space. The possible values for TRIG are:

"OPKT": outgoing packet received at NAT.

"IPKT": incoming packet received at NAT.

"ADMIN": administrative action.

"APMDEL": deletion of the underlying address and port mapping.

"AMDEL": deletion of the underlying address mapping.

"AUTO": autonomous action of the NAT.

The values applicable for any specific event are a subset of this list and are spelled out for each event in Section 5.3.

#### 5.2.3. Relationship To Objects In the NAT MIB

Table 3 lists the parameters in the same order as Table 2 and relates each parameter to its corresponding object in the NAT MIB.

PARAM-NAME	Related MIB Object(s)
Miscellaneous	
NATINST	natInstanceAlias in natInstanceTable
TRIG	None
Subscriber-identifying information	
SSUBIX	natSubscriberIndex in natSubscribersTable
SIFIX	natSubsInterfaceIndex in natSubsInterfaceIdentifierTable

SVLAN	natSubscriberVlanIdentifier in natSubscribersTable
SVPN	natSubscriberVpnIdentifier in natSubscribersTable
SV6ENC	natSubscriberIPEncapsIdType and natSubscriberIPEncapsIdAddr in natSubscribersTable
DSUBIX	natSubscriberIndex in natSubscribersTable
DIFIX	natSubsInterfaceIndex in natSubsInterfaceIdentifierTable
DVLAN	natSubscriberVlanIdentifier in natSubscribersTable
DVPN	natSubscriberVpnIdentifier in natSubscribersTable
DV6ENC	natSubscriberIPEncapsIdType and natSubscriberIPEncapsIdAddr in natSubscribersTable
Internal packet description	
IRLM	natSubscriberRealm in natSubscribersTable
IATYP	natMapIntAddrIntType in natMapIntAddrTable or natMappingIntAddressType in natMappingTable
ISADDR	natMapIntAddrInt in natMapIntAddrTable or natMappingIntAddress in natMappingTable
ISPORT	natMappingIntPort in natMappingTable
IDADDR	None
IDPORT	None
PROTO	natMappingProto in natMappingTable
External (mapped) packet description	
XRLM	natPoolRealm in natPoolTable
XATYP	natMapIntAddrExtType in natMapIntAddrTable or natMappingExtAddressType in natMappingTable
XSADDR	natMapIntAddrExt in natMapIntAddrTable or natMappingExtAddress in natMappingTable
XSPORT	natMappingExtPort in natMappingTable



XDADDR	None
XDPORT	None
Port range description	
PORTMN	None
PORTMX	None
Values related to thresholds	
POOLID	natPoolIndex in natPoolTable
POOLHW	natPoolWatermarkHigh in natPoolTable
POOLLW	natPoolWatermarkLow in natPoolTable
GAMCNT	natAddressMappingCreations - natAddressMappingRemovals in natCountersTable
GAPMCNT	natMappingCreations - natMappingRemovals in natCountersTable
SAPMCNT	natSubscriberMappingCreations - natSubscriberMappingRemovals in natSubscribersTable
Specific incoming packet description	
PSRLM	natSubscriberRealm in natSubscribersTable in the case of a packet originated by an identifiable subscriber
PATYP	None
PSADDR	None
PDADDR	None

Table 3: Relationship of Parameters To Objects In the NAT MIB

### 5.3. Encoding Of Complete Log Report For Each Event Type

This section describes the complete NAT-related contents of the logs used to report the events listed in Table 1.

#### 5.3.1. Encoding of Events Relating To Allocation Of Resources To Hosts

As indicated in Section 5.1, the event reports specified in this section MUST have APP-NAME="NAT" in the message header.

## 5.3.1.1. NAT Address Mapping Creation and Deletion

As shown in Table 1:

- o NAT address mapping creation event is indicated by MSGID set to "AMADD";
- o NAT address mapping deletion event is indicated by MSGID set to "AMDEL".

For both events, the associated SD-ELEMENT is tagged by SD-ID "namap". The contents of the namap SD-ELEMENT are shown in Table 4. The requirements for these contents are derived from the description in Section 3.1.1.

Description	PARAM-NAME	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Source subscriber index	SSUBIX	MANDATORY
Additional source subscriber classifier value as recognized at the ingress to the internal realm	One of SIFIX, SVLAN, SVPN, or SV6ENC	CONDITIONAL
Internal realm	IRLM	CONDITIONAL
Internal address type	IATYP	MANDATORY
Internal source IP address	ISADDR	MANDATORY
External realm	XRLM	CONDITIONAL
External address type	XATYP	MANDATORY
External source IP address	XSADDR	MANDATORY
Trigger for address mapping creation or deletion	TRIG	OPTIONAL

Table 4: Contents Of the SD-ELEMENT Section For Logging the Address Mapping Creation and Deletion Events

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.
- o One of SIFIX, SVLAN, SVPN, or SV6ENC REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o IRLM or XRLM REQUIRED if not the default internal or external realm, respectively, else MAY appear.

For the AMADD event type (MSGID), TRIG can take on the values "OPKT" or "ADMIN". For the AMDEL event type, TRIG can take on the values "ADMIN" or "AUTO".

Example: DS-Lite AFTR. One NAT instance. Multiple internal IPv4 realms containing the subscribers, divided by higher-level IPv6 prefix (details unnecessary). One default global IPv4 external realm. Intra-subscriber sessions use mappings into this realm.

Subscriber A in realm Internal05 sends an outgoing packet, causing the creation of an address mapping from the DS-Lite well-known address 192.0.0.2 to the global IPv4 address 198.51.100.127. Subscriber A's encapsulating IPv6 tunnel address is 2001:db8:a5e6:39b0:bd6a:35ad:1d33:6df6.

The event report for the address mapping creation is as follows (line folded into several for presentation):

```
<142>1 2013-05-07T22:14:15.03487Z record.example.net NAT 5063
AMADD [namap SSUBIX="489321"
SV6ENC="2001:db8:a5e6:3900:bd6a:35ad:1d33:6df6" IRLM="Internal05"
IATYP="IPv4" ISADDR="192.0.0.2" XATYP="IPv4"
XSADDR="198.51.100.127"
TRIG="OPKT"]
```

Character count is about 240.

#### 5.3.1.2. NAT Address and Port Mapping Creation and Deletion

As shown in Table 1:

- o NAT address and port mapping creation event is indicated by MSGID set to "APMADD";
- o NAT mapping deletion event is indicated by MSGID set to "APMDEL".

For both events, the associated SD-ELEMENT is tagged by SD-ID "napmap". The contents of the nmap SD-ELEMENT are shown in Table 5. The requirements for these contents are derived from the description in Section 3.1.2.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Source subscriber index	SSUBIX	MANDATORY
Additional source subscriber classifier value as recognized at the ingress to the internal realm	One of SIFIX, SVLAN, SVPN, or SV6ENC	CONDITIONAL
Internal realm	IRLM	CONDITIONAL
Internal address type	IATYP	MANDATORY
Internal source IP address	ISADDR	MANDATORY
Internal source port or ICMP identifier	ISPORT	MANDATORY
External realm	XRLM	CONDITIONAL
External address type	XATYP	MANDATORY
External source IP address	XSADDR	MANDATORY
External source port or ICMP identifier	XSPORT	MANDATORY
Protocol identifier	PROTO	MANDATORY
Trigger for address and port mapping creation or deletion	TRIG	OPTIONAL

Table 5: Contents Of the SD-ELEMENT Section For Logging the mapping Creation and Deletion Events

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.
- o One of SIFIX, SVLAN, SVPN, or SV6ENC REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o IRLM or XRLM REQUIRED if not the default internal or external realm, respectively, else MAY appear.

For the APMADD event type (MSGID), TRIG can take on the values "OPKT", "IPKT", or "ADMIN".

Note: it is not clear how the internal source port is selected if an address and port mapping is triggered by an incoming TCP packet. The NAT could select one based on its knowledge of subscriber port usage, but this knowledge may be incomplete. Some type of negotiation may be necessary, or else TCP address and port mappings can only be triggered by outbound packets as in the example below.

For the APMDEL event type, TRIG can take on the values "ADMIN", "AMDEL", or "AUTO".

Example: The triggering outgoing packet in the previous case was a TCP packet with internal source port 49178. As well as triggering the creation of an address mapping, the packet triggers the creation of an address and port mapping between that port and an external source port 6803. The corresponding mapping creation report would look like this:

```
<142>1 2013-05-07T22:14:15.03487Z record.example.net NAT 5063
APMADD [napmap SSUBIX="489321"
SV6ENC="2001:db8:a5e6:3900:bd6a:35ad:1d33:6df6" IRLM="Internal05"
IATYP="IPv4" ISADDR="192.0.0.2" ISPORT="49178"
XATYP="IPv4" XSADDR="198.51.100.127" XSPORT="6803"
PROTO="6" TRIG="OPKT"]
```

Character count is about 280.

#### 5.3.1.3. NAT Session Creation and Deletion

As shown in Table 1:

- o NAT session creation event is indicated by MSGID set to "SADD";
- o NAT session deletion event is indicated by MSGID set to "SDEL".

For both events, the associated SD-ELEMENT is tagged by SD-ID "nssess". The contents of the nssess SD-ELEMENT are shown in Table 6. The requirements for these contents are derived from the description in Section 3.1.3.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Source subscriber index	SSUBIX	MANDATORY
Additional source subscriber classifier value as recognized at the ingress to the internal realm	One of SIFIX, SVLAN, SVPN, or SV6ENC	CONDITIONAL
Internal realm	IRLM	CONDITIONAL
Internal address type	IATYP	MANDATORY
Internal source IP address	ISADDR	MANDATORY
Internal source port or ICMP identifier	ISPORT	MANDATORY
External realm	XRLM	CONDITIONAL
External address type	XATYP	MANDATORY
External source IP address	XSADDR	MANDATORY
External source port or ICMP identifier	XSPORT	MANDATORY
Protocol identifier	PROTO	MANDATORY
Internal destination IP address	IDADDR	CONDITIONAL
Internal destination port or ICMP identifier	IDPORT	CONDITIONAL
Destination subscriber index	DSUBIX	CONDITIONAL
Additional destination subscriber classifier value as recognized at the ingress to the external realm	One of DIFIX, DVLAN, DVPN, or DV6ENC	CONDITIONAL
External destination IP address	XDADDR	CONDITIONAL
External destination port or ICMP identifier	XDPORT	CONDITIONAL
Trigger for session creation or deletion	TRIG	OPTIONAL

Table 6: Contents Of the SD-ELEMENT Section For Logging the Session Creation and Deletion Events

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.
- o One of SIFIX, SVLAN, SVPN, or SV6ENC REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o IRLM or XRLM REQUIRED if not the default internal or external realm, respectively, else MAY appear.

- o IDADDR and IDPORT REQUIRED if destination logging is enabled and these need to be remapped to external destination address and port. Otherwise, if destination logging is disabled, they MUST NOT appear, and if destination logging is enabled, they SHOULD NOT appear because of redundancy.
- o DSUBIX REQUIRED if destination logging is enabled and the destination is a subscriber served by the NAT, else MUST NOT appear.
- o One of DIFIX, DVLAN, DVPN, or DV6ENC REQUIRED if destination logging is enabled and the destination is a subscriber served by the NAT and the external destination address is not enough to identify the external destination subscriber unambiguously, else MUST NOT appear.
- o XDADDR and XDPORT REQUIRED if destination logging is enabled, else MUST NOT appear.

For the SADD event type (MSGID), TRIG can take on the values "OPKT", "IPKT", or "ADMIN". For the SDEL event type, TRIG can take on the values "ADMIN", "MDEL", or "AUTO".

Example: destination logging is enabled. The outgoing packet that triggered the address and port mapping in the previous section was sent to 192.0.2.57 port 80. The session creation event report appears as follows:

```
<142>1 2013-05-07T22:14:15.03487Z record.example.net NAT 5063
SESSADD [nsess SSUBIX="489321"
SV6ENC="2001:db8:a5e6:3900:bd6a:35ad:1d33:6df6" IRLM="Internal05"
IATYP="IPv4" ISADDR="192.0.0.2" ISPORT="49178"
XATYP="IPv4" XSADDR="198.51.100.127" XSPORT=6803"
PROTO="6" XDADDR="192.0.2.57" XDPORT="80" TRIG="OPKT"]
```

Character count is about 310.

#### 5.3.1.4. Port Range Allocation and Deallocation

As shown in Table 1:

- o Port range allocation event is indicated by MSGID set to "PTADD";
- o Port range deallocation event is indicated by MSGID set to "PTDEL".

For both events, the associated SD-ELEMENT is tagged by SD-ID "nprng". The contents of the npset SD-ELEMENT are shown in Table 7.

The requirements for these contents are derived from the description in Section 3.1.4.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Source subscriber index	SSUBIX	MANDATORY
Additional source subscriber classifier value as recognized at the ingress to the internal realm	One of SIFIX, SVLAN, SVPN, or SV6ENC	CONDITIONAL
Internal realm	IRLM	CONDITIONAL
Internal address type	IATYP	MANDATORY
Internal source IP address	ISADDR	MANDATORY
External realm	XRLM	CONDITIONAL
External address type	XATYP	MANDATORY
External source IP address	XSADDR	MANDATORY
Port range lowest value	PORTMN	MANDATORY
Port range highest value	PORTMX	MANDATORY
Trigger for port range allocation or deallocation	TRIG	OPTIONAL

Table 7: Contents Of the SD-ELEMENT Section For Logging the Port Set Allocation and Deallocation Events

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.
- o One of SIFIX, SVLAN, SVPN, or SV6ENC REQUIRED if the internal source IP address is not enough to identify the subscriber unambiguously, else MUST NOT appear.
- o IRLM or XRLM REQUIRED if not the default internal or external realm, respectively, else MAY appear.

For the PTADD event type (MSGID), TRIG can take on the values "OPKT", "IPKT", "ADMIN", or "AUTO". For the PTDEL event type, TRIG can take on the values "ADMIN" or "AUTO".

Consider an example where the range 1024-1535 is allocated to the address mapping on which the example in Section 5.3.1.1 is based. The corresponding port range allocation report would look like this:

```
<142>1 2013-05-07T22:14:15.03487Z record.example.net NAT 5063
PTADD [nprng SSUBIX="489321"]
```



```
SV6ENC="2001:db8:a5e6:3900:bd6a:35ad:1d33:6df6" IRLM="Internal05"
IATYP="IPv4" ISADDR="192.0.0.2" XATYP="IPv4"
XSADDR="198.51.100.127"
PORTMN="1024" PORTMX="1535" TRIG="OPKT"]
```

Character count is about 270.

### 5.3.2. Encoding of Threshold Events

As indicated in Section 5.1, the event reports specified in this section MUST have APP-NAME="NATTHR" in the SYSLOG message header.

#### 5.3.2.1. NAT Address Pool High- and Low-Water-Mark Threshold Events

As shown in Table 1:

- o NAT address pool high-water-mark threshold event is indicated by MSGID set to "POOLHT";
- o NAT address pool low-water-mark threshold event is indicated by MSGID set to "POOLLT".

For both events, the associated SD-ELEMENT is tagged by SD-ID "npool". The contents of the npool SD-ELEMENT are shown in Table 8. The requirements for these contents are derived from the description in Section 3.2.1.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Pool identifier	POOLID	MANDATORY
The threshold value set by the administrator	POOLHW or POOLLW as applicable	CONDITIONAL

Table 8: Contents Of the SD-ELEMENT Section For Logging the Address Pool High- and Low-Water-Mark Threshold Events

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.
- o POOLHW REQUIRED for high-water-mark event, else MUST NOT appear.
- o POOLLW REQUIRED for low-water-mark event, else MUST NOT appear.

Example, assuming a high-water-mark threshold of 80% aggregate address-port utilization::

```
<132>1 2013-08-15T09:15:16.08716Z record.example.net NATTHR 5025
POOLHT [npool POOLID="13" POOLHW="80"]
```

Character count is about 105.

#### 5.3.2.2. Global Address Mapping High-Water-Mark Threshold Exceeded

As shown in Table 1:

- o Global address mapping high-water-mark threshold event is indicated by MSGID set to "GAMHT"; and
- o the associated SD-ELEMENT is tagged by SD-ID "ngamht".

The contents of the ngamht SD-ELEMENT are shown in Table 9. The requirements for these contents are derived from the description in Section 3.2.2.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Current number of active address mappings	GAMCNT	MANDATORY

Table 9: Contents Of the SD-ELEMENT Section For Logging the Global Address Map High-Water-Mark Threshold Event

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.

Example, assuming a threshold was set to 690000, already exceeded. As a result, prior events of this type were detected and logged, unless they were suppressed by the sort of controls discussed in Section 6.

```
<132>1 2013-08-15T09:15:16.08716Z record.example.net NATTHR 5025
GAMHT [ngamht GAMCNT="690015"]
```

Character count is about 95.

### 5.3.2.3. Global Address and Port Mapping High-Water-Mark Threshold Event

As shown in Table 1:

- o Global address and port mapping high-water-mark threshold event is indicated by MSGID set to "GAPMHT"; and
- o the associated SD-ELEMENT is tagged by SD-ID "ngapmht".

The contents of the ngmht SD-ELEMENT are shown in Table 10. The requirements for these contents are derived from the description in Section 3.2.3.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Current global number of address and port mappings	GAPMCNT	MANDATORY

Table 10: Contents Of the SD-ELEMENT Section For Logging the Global Address and Port Mapping High-Water-Mark Threshold Event

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.

Example: suppose the threshold was set to 2000000, so it has already been exceeded. As in the previous section, prior events of this type were detected and logged, unless they were suppressed by the sort of controls discussed in Section 6.

```
<132>1 2013-08-15T09:15:16.08716Z record.example.net NATTHR 5025
GAPMHT [ngapmht GAPMCNT="2000023"]
```

Character count is about 100.

### 5.3.2.4. Subscriber-Specific Address and Port Mapping High-Water-Mark Threshold Event

As shown in Table 1:

- o Subscriber-specific address and port mapping high-water-mark threshold event is indicated by MSGID set to "SAPMHT"; and

- o the associated SD-ELEMENT is tagged by SD-ID "nsapmht".

The contents of the nsapmht SD-ELEMENT are shown in Table 11. The requirements for these contents are derived from the description in Section 3.2.4.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Index of the affected subscriber	SSUBIX	MANDATORY
Current number of address and port mappings for this subscriber	SAPMCNT	MANDATORY

Table 11: Contents Of the SD-ELEMENT Section For Logging the Subscriber-Specific Address and Port Mapping High-Water-Mark Threshold Event

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.

Example: suppose the threshold was set to 1500 and the number of mappings for this subscriber has been increasing. Then this is the first threshold-exceeded event detected of what could possibly be a series of such events until subscriber consumption of outgoing ports drops below threshold again.

```
<133>1 2013-08-15T09:15:16.08853Z record.example.net NATTHR 5025
SAPMHT [nsapmht SSUBIX="489321" SAPMCNT="1501"]
```

Character count is about 115.

### 5.3.3. Encoding of Limit Events

As indicated in Section 5.1, the event reports specified in this section MUST have APP-NAME="NATLIM" in the SYSLOG message header.

#### 5.3.3.1. Global Address Mapping Limit Exceeded

As shown in Table 1:

- o Global address mapping limit exceeded event is indicated by MSGID set to "GAMLIM"; and
- o the associated SD-ELEMENT is tagged by SD-ID "ngaml".

The contents of the ngaml SD-ELEMENT are shown in Table 12. The requirements for these contents are derived from the description in Section 3.3.1.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Index of the affected subscriber	SSUBIX	MANDATORY

Table 12: Contents Of the SD-ELEMENT Section For Logging the Global Address Map Limit Exceeded Event

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.

Example:

```
<131>1 2013-08-15T09:15:16.08716Z record.example.net NATLIM 5025
GAMLIM [ngaml NATINST="VRF-Cust-X" SSUBIX="278067"]
```

Character count is about 115.

#### 5.3.3.2. Global Address and Port Mapping Limit Exceeded

As shown in Table 1:

- o Global address and port mapping limit exceeded event is indicated by MSGID set to "GAPMLIM"; and
- o the associated SD-ELEMENT is tagged by SD-ID "ngapml".

The contents of the ngapml SD-ELEMENT are shown in Table 13. The requirements for these contents are derived from the description in Section 3.3.2.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Index of the internal subscriber	SSUBIX	CONDITIONAL
Index of the external subscriber	DSUBIX	CONDITIONAL
Source realm of the triggering packet	PSRLM	MANDATORY
Incoming packet header IP address type	PATYP	CONDITIONAL
Incoming packet source IP address	PSADDR	CONDITIONAL

Table 13: Contents Of the SD-ELEMENT Section For Logging the Global Address and Port Mapping Limit Exceeded Event

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.
- o SSUBIX REQUIRED if the mapping was triggered by a packet outgoing from the internal to the external realm, else MUST NOT appear.
- o DSUBIX is REQUIRED if the mapping was triggered by a packet incoming from a subscriber served by the NAT and located in the external realm (i.e., using an address mapping created previously by the internal subscriber), else MUST NOT appear.
- o PATYP and PSADDR from the initiating packet are REQUIRED if the mapping was triggered by a packet incoming from a purely external realm (i.e., using an address mapping created previously by the internal subscriber), else MAY appear.

Example: limit event triggered by a packet coming from 192.0.2.57 in realm "externv4".

```
<131>1 2013-08-15T09:15:16.08716Z record.example.net NATLIM 5025
GAPMLIM [ngapml NATINST="VRF-Cust-X" PSRLM="externv4"
PATYP="IPv4" PSADDR="192.0.2.57"]
```

Character count is about 150.

#### 5.3.3.3. Global Limit On Number of Active Hosts Exceeded

As shown in Table 1:

- o Global active hosts limit exceeded event is indicated by MSGID set to "GSLIM"; and

- o the associated SD-ELEMENT is tagged by SD-ID "ngsl".

The contents of the ngsl SD-ELEMENT are shown in Table 14. The requirements for these contents are derived from the description in Section 3.3.3.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Index of the affected subscriber	SSUBIX	MANDATORY

Table 14: Contents Of the SD-ELEMENT Section For Logging the Global Active Host Limit Exceeded Event

Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.

An example would look exactly like that in Section 5.3.3.1 with the substitution of GSLIM for GAMLIM and ngsl for ngaml.

#### 5.3.3.4. Subscriber-Specific Limit On Number of Address and Port Mappings Exceeded

As shown in Table 1:

- o Subscriber-specific mapping limit exceeded event is indicated by MSGID set to "SMLIM"; and
- o the associated SD-ELEMENT is tagged by SD-ID "nsm1".

The contents of the nsm1 SD-ELEMENT are shown in Table 15. The requirements for these contents are derived from the description in Section 3.3.4.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Index of the affected subscriber	SSUBIX	MANDATORY

Table 15: Contents Of the SD-ELEMENT Section For Logging the Subscriber-Specific Mapping Limit Exceeded Event

## Conditions:

- o NATINST REQUIRED if device supports more than one instance, else MAY appear.

An example would look exactly like that in Section 5.3.3.1 with the substitution of SAPMLIM for GAMLIM and nsapml for ngaml.

## 5.3.3.5. Pending Fragment Limit Exceeded

As shown in Table 1:

- o Pending fragment limit exceeded event is indicated by MSGID set to "FRAG"; and
- o the associated SD-ELEMENT is tagged by SD-ID "nfpkt".

The contents of the nfpkt SD-ELEMENT are shown in Table 16. The requirements for these contents are derived from the description in Section 3.3.5.

PARAM-NAME	Description	Requirement
NAT instance identifier	NATINST	CONDITIONAL
Source realm of the packet	PSRLM	MANDATORY
Packet header IP address type	PATYP	MANDATORY
Packet source IP address	PSADDR	MANDATORY
Packet destination IP address	PDADDR	MANDATORY
Source subscriber index	SSUBIX	CONDITIONAL

Table 16: Contents Of the SD-ELEMENT Section For Logging the Pending Fragment Limit Exceeded Event

## Conditions:

- o NAT instance identifier REQUIRED if device supports more than one instance, else MAY appear.
- o Source subscriber index REQUIRED if the source of the packet is a subscriber served by the NAT and can be determined, else MUST NOT appear.

Example: assuming the packet passing the limit came from an internal host and was dropped as a result of the limit.

```
<132>1 2013-08-15T09:15:16.08Z record.example.net NATLIM 5025
```



```
FRAG [nfpkt PSRLM="DsLite-089" PATYP="IPv4" PSADDR="192.0.0.2"  
PDADDR="203.0.113.26" SSUBIX="32791"]
```

Character count is about 160.

## 6. Management Considerations

This section considers requirements for management of the log system to support logging of the events described above. It first covers requirements applicable to log management in general. Any additional standardization required to fulfil these requirements is out of scope of the present document. Subsequent sub-sections discuss management issues related to specific event report types. The identifiers PRI, APP-NAME, and MSGID used below refer to fields in the SYSLOG header [RFC5424]

### 6.1. General Requirements For Control Of Logging

This document assumes that any implementation provides the following capabilities, discussed in more detail below:

- o ability to configure the PRI value of each event report type at the granularity of (APP-NAME, MSGID) combination;
- o ability at each collector to determine that event reports that it should have received have been lost. The required granularity is at least at the level of PRI and may be finer for some event types.
- o ability to configure criteria to automatically suppress the generation of event reports while the criteria are met, at the granularity of (APP-NAME, MSGID) combination.

#### 6.1.1. Configuration of PRI Value

The PRI value is composed of two numbers, the Facility value and the Severity. It may be used at the origin for selecting logs to streams being dispatched to different collectors, and in applications beyond the collectors to prioritize display of logs to operators. The event reports in this document have been structured such that the Severity level varies between event types as represented by (APP-NAME, MSGID) combination. As an extreme example, the address pool high-water-mark threshold event (APP-NAME="NATTHR", MSGID="POOLHT") is obviously more urgent than the low-water-mark threshold event (APP-NAME="NATTHR", MSGID="POOLLT").

To some extent, this document tries to simplify message routing by making a general distinction between event types recording the

allocation of resources to hosts (with APP-NAME="NAT") and events of interest to operations and maintenance (with APP-NAME="NATTHR" and APP-NAME="NATLIM"). The need to provide different Severity levels for different event types remains.

#### 6.1.2. Ability For Each Collector To Detect Lost Event Reports

Operators have a need to know when a given collector has not received all of the event reports it should have. It probably does not matter if less-important events are tracked at the granularity of event type (APP-NAME, MSGID combination), by APP-NAME, or just by PRI value.

The event types defined in this document relating to allocation of resources to hosts are a special case. Regulatory requirements or the possibility that such reports might be introduced into court in cases such as abuse impose a requirement that the record of allocations to a particular host be complete. This requirement is important enough to be stated in the Security Considerations section (Section 7), where the implementation of signed SYSLOG messages [RFC5848], which also provides message sequencing, is mandated as part of this specification.

In deploying [RFC5848], the operator needs to decide the level of granularity of tracking, whether it should be over the whole set of reports covered by APP-NAME="NAT" or at a finer level. This judgement has to be tempered by local circumstances. One point to note is that since both creations/allocations and deletions/deallocations are recorded, a certain amount of redundancy is available in the reports being generated. However, without both the creation and deletion timestamps, there is no definitive evidence of the specific period of time during which the resources concerned were allocated to a specific host.

#### 6.1.3. Ability To Rate Limit Or Disable Event Reports

The event report types specified with APP-NAME="NATTHR" and APP-NAME="NATLIM" all relate to thresholds or limits. By their nature, events of this sort will come in bursts. The threshold or limit will be hit, the resource concerned will remain busy for a period, then pressure on the resource will ease. Depending on the resource, possibly hundreds of instances of the event concerned will be detected during a single busy period.

Where repeated events involve the same resource, it makes little sense to report all of them, since the NAT MIB counters provide the necessary information more succinctly. On the other hand, it can be useful to know that the fragmentation limit, for instance, is being hit by successive packets from the same source address.

As a result of these considerations, this document requires that implementations **MUST** provide means to configure limits on the rate at which event reports of a given type (APP-NAME, MSGID combination) are generated. It is **RECOMMENDED** that it be possible to specify two values per (APP-NAME, MSGID) combination:

- o minimum time between initial instances of a given event report type;
- o maximum number of instances of the event report to generate per busy period.

Regardless of the detailed method the implementation provides for specifying the rate limiting of individual event report types, all implementations **MUST** allow the operator to indicate through configuration that a given event report type is to be completely disabled. This is particularly required to disable logging of either session or mapping creations and deletions when not required (see discussion in Section 3.1.2). It is also required when the operator prefers to receive threshold event notifications via SNMP rather than SYSLOG.

The ability to rate limit or disable event reports **MUST NOT** interfere with the requirement to detect lost messages. This has implications for any sequence numbering used for that purpose. It is **RECOMMENDED** in any event that the implementation provide MIB counters of numbers of messages not generated due to rate limiting by event type supported. If this is done, counters for disabled event report types **SHOULD NOT** be incremented, since that could require keeping unnecessary additional state.

## 6.2. Setting Limits and Thresholds

The "NATTHR" and "NATLIM" events specified in this document depend on the thresholds and limits configured in the NAT MIB [I-D.behave-NAT-MIB]. The limits have to do with policy in some cases (e.g., most especially the subscriber-specific limits), but generally depend on the implementation and the device in which it is deployed.

The purpose of high-water-mark thresholds is, of course, to give sufficient advance warning that utilization of a particular resource is approaching its limit, so that appropriate provisioning or reconfiguration action can be undertaken to preserve target service levels on the NAT device. Thus the following general principles apply:

- o A high-water-mark threshold should be derived as a percentage of the relevant limit.
- o The more quickly that utilization of a given resource can build up, the lower the threshold must be to provide an adequate response time.
- o Some limits are more important than others in terms of their effect on overall service levels provided by the NAT device. To focus attention on the more important limits, their corresponding thresholds should be set lower than those for less-important limits, all other things being equal.

In practice, thresholds will require tuning to fit the particular characteristics of the NAT device and its users.

The setting of the high-water-mark-thresholds for address pools (Section 3.2.1) poses additional challenges. The problem is that the bottleneck for port availability will generally be a single protocol, which may vary from one time to another. However, the threshold is based on overall port utilization. If port usage is such that one protocol generally predominates, the required threshold value has to be lower than if usage is more balanced between protocols. Clearly the appropriate threshold value depends on the characteristics of the traffic handled by the particular address pool concerned.

Pooling behaviour adds another factor for consideration. With a pooling behaviour of "arbitrary" [RFC4787], port utilization for the bottleneck protocol can be quite high before service levels offered by the pool are in danger. On the other hand, with a pooling behaviour of "paired", possible utilization levels will be much lower because typically a number of port values will be reserved to each address mapping and only some of those will be in use on the average. The difference between "arbitrary" and "paired" utilization for a given level of service may be quite dramatic.

### 6.3. Other Management Requirements

The identification of internal realms is contingent on the the existence and applicability of default internal and external realms. If the implementation is capable of supporting more than one internal or external realm, it MUST provide the means for the operator to specify which realm is the default internal and/or external realm, as the case may be.

## 7. Security Considerations

When logs are being recorded for regulatory reasons or as potential evidence in abuse cases, preservation of their integrity and authentication of their origin is essential. To achieve this result, signed SYSLOG messages [RFC5848] MUST be implemented as part of this specification. It is RECOMMENDED that the operator deploy [RFC5848] where local requirements on integrity and authentication of origin are stringent. In conjunction with [RFC5848] and as recommended in Section 3 of that document, TLS transport as specified in [RFC5425] SHOULD be used between the origin and the collector(s) and MUST be implemented. Section 5.2.1 of [RFC5848] specifies the minimum support for Key Blob Type that must be provided by implementations of that specification.

Access to the logs defined in Section 3.1 and Section 5.3.1 while the reported assignments are in force could improve an attacker's chance of hijacking a session through port-guessing. Even after an assignment has expired, the information in the logs SHOULD be treated as confidential, since, if revealed, it could help an attacker trace sessions back to a particular user or user location. It is therefore RECOMMENDED that these logs be transported securely, using [RFC5425], for example, even if [RFC5848] is not deployed, that they be stored securely at the collector, and that access to them at the collector and in applications be tightly controlled.

The logs defined in Section 3.2 and Section 3.3 are less sensitive in general, but since many of them contain the subscriber identifier, they could be used to get some sense of subscriber activity. The fragmentation limit event provides actual packet header contents. Operators SHOULD at the least deploy secure transport to ensure that this information is not misused.

## 8. IANA Considerations

This document requests IANA to make the following assignments to the SYSLOG Structured Data ID Values registry. RFCxxxx refers to the present document when approved.

Some PARAM-NAMES appear under more than one SD-ID in Table 17. Formally, a parameter used with more than one event is registered as multiple separate parameters, one for each event report in which it is used. However, there is no reason to change either the PARAM-NAME or the encoding of the PARAM-VALUE between different instances of the same parameter if the parameters have the same meaning in both event reports.

While a number of parameters are marked CONDITIONAL in the body of this document, the SYSLOG registry provides only for MANDATORY and OPTIONAL parameters. All CONDITIONAL parameters have been placed in the OPTIONAL category in Table 17.

Structured Data ID	Structured Data Parameter	Required or Optional	Reference
namap	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	OPTIONAL	RFCxxxx
	SIFIX	MANDATORY	RFCxxxx
	SIFIX	OPTIONAL	RFCxxxx
	SVLAN	OPTIONAL	RFCxxxx
	SVPN	OPTIONAL	RFCxxxx
	SV6ENC	OPTIONAL	RFCxxxx
	IRLM	OPTIONAL	RFCxxxx
	IATYP	MANDATORY	RFCxxxx
	ISADDR	MANDATORY	RFCxxxx
	XRLM	OPTIONAL	RFCxxxx
	XATYP	MANDATORY	RFCxxxx
	XSADDR	MANDATORY	RFCxxxx
	TRIG	OPTIONAL	RFCxxxx
----	----	----	----
napmap	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
	SIFIX	OPTIONAL	RFCxxxx
	SIFIX	OPTIONAL	RFCxxxx
	SVLAN	OPTIONAL	RFCxxxx
	SVPN	OPTIONAL	RFCxxxx
	SV6ENC	OPTIONAL	RFCxxxx
	IRLM	OPTIONAL	RFCxxxx
	IATYP	MANDATORY	RFCxxxx
	ISADDR	MANDATORY	RFCxxxx
	ISPORT	MANDATORY	RFCxxxx
	XRLM	OPTIONAL	RFCxxxx
	XATYP	MANDATORY	RFCxxxx
nsess	XSADDR	MANDATORY	RFCxxxx
	XSPORT	MANDATORY	RFCxxxx
	PROTO	MANDATORY	RFCxxxx
	TRIG	OPTIONAL	RFCxxxx
	----	----	----
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
	SIFIX	OPTIONAL	RFCxxxx
	SIFIX	OPTIONAL	RFCxxxx
	SVLAN	OPTIONAL	RFCxxxx
	SVPN	OPTIONAL	RFCxxxx

	SV6ENC	OPTIONAL	RFCxxxx
	IRLM	OPTIONAL	RFCxxxx
	IATYP	MANDATORY	RFCxxxx
	ISADDR	MANDATORY	RFCxxxx
	ISPORT	MANDATORY	RFCxxxx
	XRLM	OPTIONAL	RFCxxxx
	XATYP	MANDATORY	RFCxxxx
	XSADDR	MANDATORY	RFCxxxx
	XSPORT	MANDATORY	RFCxxxx
	PROTO	MANDATORY	RFCxxxx
	IDADDR	OPTIONAL	RFCxxxx
	IDPORT	OPTIONAL	RFCxxxx
	DSUBIX	OPTIONAL	RFCxxxx
	DIFIX	OPTIONAL	RFCxxxx
	DVLAN	OPTIONAL	RFCxxxx
	DVPN	OPTIONAL	RFCxxxx
	DV6ENC	OPTIONAL	RFCxxxx
	XDADDR	OPTIONAL	RFCxxxx
	XDPORT	OPTIONAL	RFCxxxx
	TRIG	OPTIONAL	RFCxxxx
	----	----	----
nprng		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
	SIFIX	OPTIONAL	RFCxxxx
	SVLAN	OPTIONAL	RFCxxxx
	SVPN	OPTIONAL	RFCxxxx
	SV6ENC	OPTIONAL	RFCxxxx
	IRLM	OPTIONAL	RFCxxxx
	IATYP	MANDATORY	RFCxxxx
	ISADDR	MANDATORY	RFCxxxx
	XRLM	OPTIONAL	RFCxxxx
	XATYP	MANDATORY	RFCxxxx
	XSADDR	MANDATORY	RFCxxxx
	PORTMN	MANDATORY	RFCxxxx
	PORTMX	MANDATORY	RFCxxxx
	TRIG	OPTIONAL	RFCxxxx
	----	----	----
npool		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	POOLID	MANDATORY	RFCxxxx
	POOLLT	OPTIONAL	RFCxxxx
	POOLHT	OPTIONAL	RFCxxxx
	----	----	----
ngamht		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	GAMCNT	MANDATORY	RFCxxxx
	----	----	----

ngapmht		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	GAPMCNT	MANDATORY	RFCxxxx
----	----	----	----
nsapmht		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
	SAPMCNT	MANDATORY	RFCxxxx
----	----	----	----
ngaml		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
----	----	----	----
ngapml		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	OPTIONAL	RFCxxxx
	DSUBIX	OPTIONAL	RFCxxxx
	PSRLM	MANDATORY	RFCxxxx
	PATYP	OPTIONAL	RFCxxxx
	PSADDR	OPTIONAL	RFCxxxx
----	----	----	----
ngsl		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
----	----	----	----
nsapml		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	SSUBIX	MANDATORY	RFCxxxx
----	----	----	----
nfpkt		OPTIONAL	RFCxxxx
	NATINST	OPTIONAL	RFCxxxx
	PSRLM	MANDATORY	RFCxxxx
	PATYP	MANDATORY	RFCxxxx
	PSADDR	MANDATORY	RFCxxxx
	PDADDR	MANDATORY	RFCxxxx
	SSUBIX	OPTIONAL	RFCxxxx

Table 17: NAT-Related STRUCTURED-DATA Registrations

## 9. References

### 9.1. Normative References

[I-D.behave-NAT-MIB]

Perreault, S., Tsou, T., and S. Sivakumar, "Additional Managed Objects for Network Address Translators (NAT) (Work in progress)", September 2013.



- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC2685] Fox, B. and B. Gleeson, "Virtual Private Networks Identifier", RFC 2685, September 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC4265] Schliesser, B. and T. Nadeau, "Definition of Textual Conventions for Virtual Private Network (VPN) Management", RFC 4265, November 2005.
- [RFC4363] Levi, D. and D. Harrington, "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions", RFC 4363, January 2006.
- [RFC4784] Carroll, C. and F. Quick, "Verizon Wireless Dynamic Mobile IP Key Update for cdma2000(R) Networks", RFC 4784, June 2007.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.
- [RFC5425] Miao, F., Ma, Y., and J. Salowey, "Transport Layer Security (TLS) Transport Mapping for Syslog", RFC 5425, March 2009.
- [RFC5848] Kelsey, J., Callas, J., and A. Clemm, "Signed Syslog Messages", RFC 5848, May 2010.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [US-ASCII] American National Standards Institute, , "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

## 9.2. Informative References

- [I-D.behave-ipfix-nat-logging]  
Sivakumar, S. and R. Penno, "IPFIX Information Elements for logging NAT Events (Work in progress)", August 2013.
- [I-D.pcp-port-set]  
Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation (Work in progress)", July 2013.
- [I-D.softwire-lw4over6]  
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture (Work in progress)", July 2013.
- [I-D.softwire-map]  
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP) (Work in progress)", August 2013.
- [I-D.tsou-behave-natx4-log-reduction]  
Tsou, T., Li, W., and T. Taylor, "Port Management To Reduce Logging In Large-Scale NATs (Work in progress)", July 2013.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, March 2005.
- [RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264, June 2011.
- [RFC6674] Brockners, F., Gundavelli, S., Speicher, S., and D. Ward, "Gateway-Initiated Dual-Stack Lite Deployment", RFC 6674, July 2012.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.
- [RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.
- [RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.
- [RFC7040] Cui, Y., Wu, J., Wu, P., Vautrin, O., and Y. Lee, "Public IPv4-over-IPv6 Access Network", RFC 7040, November 2013.

#### Authors' Addresses

Zhonghua Chen  
China Telecom  
P.R. China

Email: 18918588897@189.cn

Cathy Zhou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Email: cathy.zhou@huawei.com

Tina Tsou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Email: [tina.tsou.zouting@huawei.com](mailto:tina.tsou.zouting@huawei.com)

T. Taylor (editor)  
Huawei Technologies  
Ottawa  
Canada

Email: [tom.taylor.stds@gmail.com](mailto:tom.taylor.stds@gmail.com)

behave WG  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2014

W. Meng  
ZTE Corporation  
July 15, 2013

Network Address Port Group Translator  
draft-meng-behave-napgt-01

Abstract

Currently, if an internal server and hosts are behind NAT, they cannot share a global IP address except adding lots of static NAT rule configuration. Because if a server wants to provide a service by constant port (i.e. HTTP and FTP), the destination port of packet sent by an external client should not be changed when it crosses NAT. This document specifies a new method to assign NAT global address and port, aiming to solve the problem that internal servers and hosts cannot share less global IP addresses.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Convention and Terminology . . . . .	3
3. Scenarios . . . . .	3
4. Configuration . . . . .	3
5. Mapping Item . . . . .	3
6. Security Considerations . . . . .	4
7. Normative References . . . . .	4
Author's Address . . . . .	4

## 1. Introduction

With the depletion of IPv4 addresses, many operators have begun to deploy NAPT in their network. However, the use of NAPT has many shortcomings. For example, a server is placed in an internal network, this may happen when an external client attempts to access a server in the internal network through HTTP. Dynamic NAPT cannot be used for translation except for NAT, because the translation MUST keep the consistency of the internal port and external port (PORT:80 should not be changed). This may be causing a public IP address being occupied to a server, but not for other users to access, resulting in a misuse of resources.

It appears that STATIC NAPT is a near-perfect solution to deal with this issue. However, However, if there are a lot of services or servers in the internal network, it may not be useful to configure a huge number of STATIC NAPT rules. This will increase the complexity of configuration without a corresponding increase in functionality.

The current existing solution works by changing the configuration due to user complaints. A user does not know whether his/her IP address is global or local. During the use of a global IP address, he/she can access the server placed in his home from external. Until one day, he/she cannot do that because he/she gets a local IP address. He/She is not satisfied and complains to the operator. Operator has to assign global IP address for him/her and still assign local IP address for others. Operator distinguishes him/her from others by embedding tags into the subscriber backend database.

Now, through the variant of traditional NAPT translation, we can achieve sharing a global IP address among a server and hosts placed in the same internal network. It is called NAPGT (Network Address Port Group Translator).

## 2. Convention and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Scenarios

There can be a typical scenario if NAT is involved.

In this scenario, a server and several hosts are behind NAT. NAT has only a global IP address. NAT needs to let client access to server without affecting any hosts accessing to the internet.

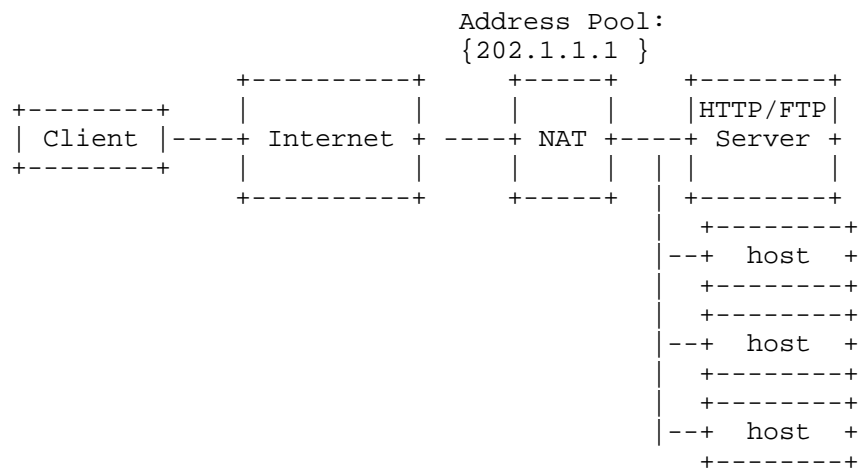


Figure 1: Server and Hosts Behind NAT

## 4. Configuration

The NAPGT needs to be configured in a NAT device. Port-ranges MUST be specified in NAT pool, such as '1-1024','7000-7100'. It means that a collection of ports MUST be utilized for binding. The rest of ports can be assigned to hosts.

Static or dynamic rules MUST be configured for server. Rules for hosts has no special requirement.

## 5. Mapping Item

To achieve client accessing server behind NAT by HTTP or FTP, mapping item MUST be generated in advance.

```

NAT(config)#show nat translations all
=====
=
  Protocol Type      Local Add:Port      global Add:Port      Destination Add:Port
=====
=
    ---  NAPGT      192.168.0.1:<1-1024>  202.1.1.1:<1-1024>      211.1.1.1:*
-----
-
    UDP STATIC      192.168.0.2:1024      202.1.1.1:1025          222.1.1.1
-----
-
    TCP DYNAMIC      192.168.0.3:2565      202.1.1.1:1030          ---
-----
-

```

Figure 2: Mapping Item in NAT(Example)

## 6. Security Considerations

To be added later on as-needed basis.

## 7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## Author's Address

Wei Meng  
 ZTE Corporation  
 No.50 Software Avenue, Yuhuatai District  
 Nanjing  
 China

Email: meng.wei2@zte.com.cn, vally.meng@gmail.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 1, 2014

K. Nishizuka  
NTT Communications  
D. Natsume  
NTT Neomeit  
Sep 28, 2013

Carrier-Grade-NAT (CGN) Deployment Considerations.  
draft-nishizuka-cgn-deployment-considerations-01

## Abstract

This document provides deployment considerations for Carrier-Grade-NAT (CGN). Due to emerging new web technologies such as Websocket, SPDY and HTTP2.0, the trend of the Internet traffic has been changing. The number of sessions of commonly-used applications were investigated to estimate the efficiency of IPv4 address sharing of CGN. Based on the result of the average number of sessions of subscribers, the verification of CGN was conducted in the large scale network experiment environment with one million emulated subscribers. It revealed that CGN can be used in more centralized location of a provider's network and it arose many considerations.

## Status of this Memo

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

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

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

This Internet-Draft will expire on April 1, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions used in this document . . . . .	3
3. Motivation . . . . .	3
4. The number of sessions of applications . . . . .	4
5. Feasibility of port assignment methods . . . . .	6
5.1. Port assignment methods . . . . .	6
5.2. Efficiency of address saving . . . . .	6
5.3. Logging design . . . . .	7
5.3.1. Amount of the NAT log . . . . .	7
5.3.2. Necessity for destination information . . . . .	9
6. Scalability of CGN . . . . .	9
6.1. Performance of CGN . . . . .	9
6.2. Redundancy features of CGN . . . . .	11
6.3. DNS query traffic considerations . . . . .	12
6.4. Separation of traffic . . . . .	13
7. Tested web sites and applications (Excerpts) . . . . .	13
8. IANA Considerations . . . . .	14
9. Security Considerations . . . . .	14
10. Acknowledgments . . . . .	14
11. References . . . . .	15
11.1. Normative References . . . . .	15
11.2. Informative References . . . . .	15
Authors' Addresses . . . . .	16

## 1. Introduction

IP address sharing is tentative technic to deal with the shortage of IPv4 addresses. As described in [RFC6269], IP address sharing causes many issues such as application failures and security vulnerabilities. A part of these issues is based on the assigned number of sessions per user and port allocation method of CGN. How many sessions are sufficient for users is one of the important considerations. Moreover, the efficiency of CGN is based on the average number of sessions of subscribers. To answer to these points, this document lists the number of port consumption of major application and web sites.

This document also describes the deployment considerations of CGN to specify the optimum place according to CGN performance. CGN performance was experimentally-verified with realistic traffic generated by amount of emulated users.

The growth of IPv6 is continual solution of the shortage of IPv4 addresses and frees these issues. By adopting the combination of the IPv4 shared address and native IPv6, the duty of CGN will decrease and as the result, the bad effect on applications which are caused by the limitation of available ports and address translation itself and security vulnerability will be resolved. The most effective way of deploying CGN is examined in this document. Further discussion about the integration of CGN into the existing network is studied in [I-D.ietf-opsawg-lsn-deployment].

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

## 3. Motivation

With a progressive exhaustion of IPv4 addresses, the demands for sharing IPv4 addresses with multiple customers are rapidly rising, thus many proposals are getting much attention include Carrier Grade NAT (CGN, or LSN for Large Scale NAT) [RFC6888], Dual-Stack Lite [RFC6333], NAT64 [RFC6146], Address+Port (A+P) [RFC6346], 464XLAT [RFC6877] and MAP [I-D.ietf-softwire-map]. The practical configuration of these method is based on the same considerations as follows:

- Stateful or Stateless
- Centralized or Distributed
- Dynamic port assignment or Static port assignment
- Log reduction strategy
- Security considerations

The best practice about these considerations should be derived from realistic experiment because there are pros and cons. Though we tested them in NAT444 environment, the result is applicable for other approaches. The investigation of number of sessions is described in this document and it can be also helpful for all of them.

#### 4. The number of sessions of applications

The number of concurrent sessions of applications is important factor of designing of CGN because there is trade-off between the efficiency of IPv4 address saving and the availability of those applications. In addition, for security and fairness, we should limit the number of sessions per user. As described in [RFC6269], infected devices could rapidly exhaust the available ports of global pool addresses, hence all the rest of users could not through the CGN anymore. In order to place the CGN to existing network, we should know how many sessions are sufficient for every user. Here is a list of applications and their average sessions. We selected and tested 50 sites from the list of top sites and remarkable applications. For web browsing, We used Chrome and Firefox which are capable of SPDY.

Application	Total sessions	TCP port80	TCP port443	UDP port53
Web mail	65	35	30	20
Video	83	77	6	20
Portal site	47	47	0	13
EC site	45	43	2	11
blog	61	59	2	17
Search Engine	8	8	0	4
Online Banking	20	2	18	4
Cloud Service	29	23	6	6
iTunes	20	1	19	7
Twitter	33	1	32	12
Twitter(mobile)	14	2	11	3
facebook	51	40	11	18
facebook(mobile)	18	11	7	10
Game	95	86	9	19

Figure 1: The number of sessions of applications.

Figure 1

The number of sessions of these applications are up to 100 sessions. There are no longer high-consumption applications. This observation implies that modern applications such as facebook have changed to use multiplexed requests. Previously, web technologies for achieving high-performance access consumed many HTTP sessions. Now, current cutting edge technologies such as WebSocket, SPDY and HTTP2.0 avoid such an abusing. Basically, all the requests are multiplexed into one TCP connection. However, a kind of game applications still consume many sessions.

The last factor of the estimation of number of sessions is how many applications are used simultaneously within a single CPE (Customer Premises Equipment) which includes non-PC devices like gaming devices. Our investigation shows that the average number of session of active subscriber is 400. We daresay the limitation of 1000 sessions per user would not affect the most of users while preventing the severe abuse from certain users.

## 5. Feasibility of port assignment methods

Basing on the investigation of the number of sessions of applications, the realistic parameter of each port assignment method was estimated by the verification.

### 5.1. Port assignment methods

The efficiency of IPv4 saving by CGN is highly depending on how to allocate the ports of pool addresses to each users. There are 2 major methods: dynamic assignment and static assignment [I-D.chen-sunset4-cgn-port-allocation]. There are combined problem involving efficiency of address saving and logging information reduction. Typical IP Network Address Translator (NAT) [RFC2663][RFC2993] implementation uses dynamic assignment, so NAT444, NAT64, DS-lite and 464XLAT are originally dynamic assignment approach. To avoid the huge amount of information needed to be recorded, those approaches have variations of static assignment [I-D.donley-behave-deterministic-cgn] and MAP is inherently static assignment approach. For taking advantage of both methods, the hybrid method that is dynamic assignment of port ranges has been implemented in some CGN. The merits of the port block assignment have been referred in [RFC6346], [I-D.donley-behave-deterministic-cgn] and [I-D.chen-sunset4-cgn-port-allocation].

### 5.2. Efficiency of address saving

In the dynamic assignment, the ports of pool address are allocated randomly for active users. This method can use pool addresses and ports most effectively. The average number of port consumption (N) per active subscriber is the key value for dynamic assignment. In the verification, the average number of port consumption (N) was estimated to be 400. At the same time, user-quota of 1000 sessions was set to avoid the abuse. The percentage of the active subscribers (a) was estimated to be 25% at the value during the busy hour of traffic (21:00 pm to 1:00am). In this time, "active" subscriber means who create a new session in certain period of time. Then, when a CGN adopt the dynamic assignment, the required number of the pool

address is as follows:

$$\# \text{ of pool address (P)} = \# \text{ of Subscriber (S)} * a * N / (65536 - R)$$

Here, (R) is reserved TCP/UDP port list referred in [I-D.donley-behave-deterministic-cgn]. CGN should eliminate the wellknown ports (0-1023 for TCP and UDP) to avoid the bad interpretation from destination servers. It is natural to translate source port of outgoing packet to ephemeral ports. Using the equation, 1550 pool addresses are sufficient for 1,000,000 subscribers.

On the other hand, in static assignment, the ports are allocated a priori for every users. The pool addresses and ports are reserved to every users, so most of them could be a dead stock because there are light users and heavy users in aspect of port consumption. The max number of port consumption in all subscribers is the key value for static assignment. The true peak number of the session by a heavy user could be over 10,000 sessions. However it can be assumed that such a severe consumption of ports to be an abuse, so the number of statically assigned port (M) is controllable parameter by each providers. In the static assignment, the required number of the pool address is as follows:

$$\# \text{ of pool address (P)} = \# \text{ of Subscriber (S)} * M / (65536 - R)$$

Taking account into the investigation of number of sessions of applications, the desirable value of (M) is over 1,000. As the result, no less than 15,501 pool addresses are needed for 1,000,000 subscribers. The compression ratio is one tenth of the case of dynamic assignment.

The feasibility of dynamic and static assignment configuration was confirmed in the verification.

### 5.3. Logging design

#### 5.3.1. Amount of the NAT log

The size of the log is important consideration of dynamic assignment because it demands a huge scale of logging ecosystem for CGN. There is a case that providers must identify a user to respond abuse or public safety requests. Conventionally, source IP address and a timestamp are needed. It was possible to identify a user by comparing IP address with authentication logs of the exact time. However, when IP address is shared by the CGN, it is necessary to compare the translated address and port information which are given by the destination host with the NAT log to identify the untranslated

IP address. According to the [RFC6888], following information is recommended to log (for NAT444):

- Transport Protocol - 1 byte
- Source IP address:port - 6 byte
- Source IP address:port after translation - 6 byte
- Timestamp - 8 byte

In addition, the indicator of the allocation and deallocation are needed because it assures that the identified subscriber certainly had been using the translated IP address and port. Plus, some identifier like the index or hostname of the CGN is needed to identify to which realm an address belongs.

- Add/Delete - 1 byte
- CGN device ID - 4 byte

As the result, the minimum size of NAT log is 26 bytes in binary. In ASCII format, the average size of NAT log is about 120 byte . Every active subscriber generate 400 sessions in average for a certain amount of time. It is assumed that the event happens every 5 minute in the most severe condition. The size of the log (L) for time frame (T) can be estimated as follows (for ASCII format):

The size of log (L) = # of Subscriber (S) \* a \* N \* 120byte \* 2 \* (Time frame(T) / 5 min. )

It should be noted that the log is generated at the timing of NAT table creation and freeing. As the result, for 1,000,000 users, the size of log is piled up to 6.4 terabytes per day. The verification result confirm the existing estimation referred in [I-D.donley-behave-deterministic-cgn].

The size of the log can be reduced without loss of information. Compact format is the technique of reducing the amount of log by using a notational change (hexadecimal number). It was confirmed by verification that the compact format can reduce amount of log to about 80% as compared with ASCII format. Though it was not tested, theoretically binary format is the smallest notation and amount of log can be reduced to 22%.

In static allocation, the amount of log is dramatically reduced even to zero because the untranslated IP address and the translated IP address / port range are mapped a priori.



### 5.3.2. Necessity for destination information

In [RFC6269], it is pointed out that only providing information about the external address to a service provider is no longer sufficient to identify customers unambiguously. One of the solutions is the method of recording the source port information (and exact time stamp) additionally by the destination server or FW, which is demanded in [RFC6302]. The other solution is the method of recording destination IP address and port information by CGN of service provider. The both solutions are imperfect. In [I-D.tsou-behave-natx4-log-reduction], it is noted that source port recording is not supported by every application. Thus, to increase the certainty, additional logging of destination address and port is effective measure to deal with the legal request from servers which are not compliant with [RFC6302]. In dynamic assignment, to log destination address is additional. It is confirmed by the verification that by logging destination address, only 4% of amount of log is increased in ASCII format. On the other hand, in static assignment, logging of every session is newly required and it has the same amount of log as the dynamic assignment. It completely breaks the merit of the static assignment.

## 6. Scalability of CGN

The estimation of efficiency of address saving and the logging design are depending on the number of subscribers accommodated with a CGN. The scalability of the current CGN was verified by the measurement of the performance.

### 6.1. Performance of CGN

According to the experimental results, there are three base capacities to indicate CGN performance as follows:

- Through put
- MCS: Max Concurrent Sessions
- CPS: Connections per Sec

These capacities are not independent of each other, but become mixed load for CGN. Each load will be combined in real network traffic, thus using subscriber emulated traffic is important for measuring the performance in realistic way.

Through put is forwarding performance of CGN. Currently CGN equipments with an IF of 1GigabitEthernet and 10GigabitEthernet are flagship models of the manufactures, but CGN has an upper limit internally because the performance depends on internal devices such

as CPUs. By ON / OFF of ALGs (Application Level Gateway), the forwarding performance will be affected because the traffic process is possibly changed to the path through CPU.

MCS shows an upper limit of the number of records kept in NAT table. The number of holding sessions depends on retention time of NAT table. That is because, even after the end of data transmission, the NAT table is held in a certain period of time to guarantee the behavior of an application. As described in [RFC6888] REQ-8, if the CGN tracks TCP sessions, NAT tables may be released when RST or FIN of TCP has been observed. In case of TCP session where RST or FIN session has not been observed, and UDP and ICMP communication, NAT table should retain a certain amount of time. Also, in case of Full Cone NAT, a table of Full Cone NAT also should retain a certain time to await communication from outside for a certain period of time. It is effective to shorten the time-out value in order to suppress the overflowing NAT table, but it is needed to be careful not to inhibit the behavior of the application. It is desirable that retention time of NAT table is configurable as time-out value. In the experiment, the time-out values are as follows:

Protocols	TCP	TCP SYN	UDP	DNS (port53)	ICMP
Time-out Value	300	60	300	3	2

Figure 2: The time-out values (sec) in the experiment.

Figure 2

These settings didn't break the behavior of applications we tested.

It is very difficult to estimate maximum number of concurrent sessions in the network where traffic already exists. By our assumption, maximum number of concurrent sessions was estimated to be 1M sessions per 10,000 users as follows:

Max Concurrent Sessions (MCS) = # of Subscriber (S) \* a \* N

As the result, it is verified that tested current CGN is able to have 16M sessions for 160,000 subscribers with the capability of the dynamic assignment and logging. It means that introducing CGN up to about 15G traffic section is capable, which implies that CGN can be placed to more centralized position of the network. In summary, the settings and the performance result are as follows:

	Assumed Values
average # of sessions(N)	400
% of the active subscribers (a)	25
	Verified Values
# of Subscriber (S)	160,000
Max Concurrent Sessions(MCS)	16,000,000
Connection Per Sec(CPS)	30,000
# of pool address (P)	4,000
size of log (L) (in 10min)	7.0GB

Figure 3: The performance results of tested CGN.

Figure 3

In the verification, session arrival rate by emulated subscribers was not so high because the load of concurrent sessions is noticeable in the equipment used in the experiment. There were no problems in weak load of about 30,000 CPS. In case that traffic flows suddenly change to standby equipment in redundant network, CPS performance becomes rate-limiting, so CPS performance is also important factor to minimize the effect of failures.

## 6.2. Redundancy features of CGN

It is often referred that introduction of CGN could create Single Point of Failure(SPOF) (ex. in [RFC6269]). CGN is stateful, in contrast to stateless BR of MAP, so the redundant configuration must be achieved by the synchronization of the NAT table between redundant equipments. Moreover, introduction of CGN creates layer 3 boundary to NATed traffic, so the redundancy features may work with routers via dynamic routing. Nevertheless, it is verified that current CGN can be configured and introduced to service providers network with the redundancy features. In the verification, CGN was able to switch to another CGN with sub-sec loss of traffic even in the situation that they holds 16M concurrent sessions.

### 6.3. DNS query traffic considerations

How to deal with the DNS query traffic is unignorable concern for deployment of CGN. In the test scenario, a control experiment was conducted to reveal the impact of the huge amount of DNS queries.

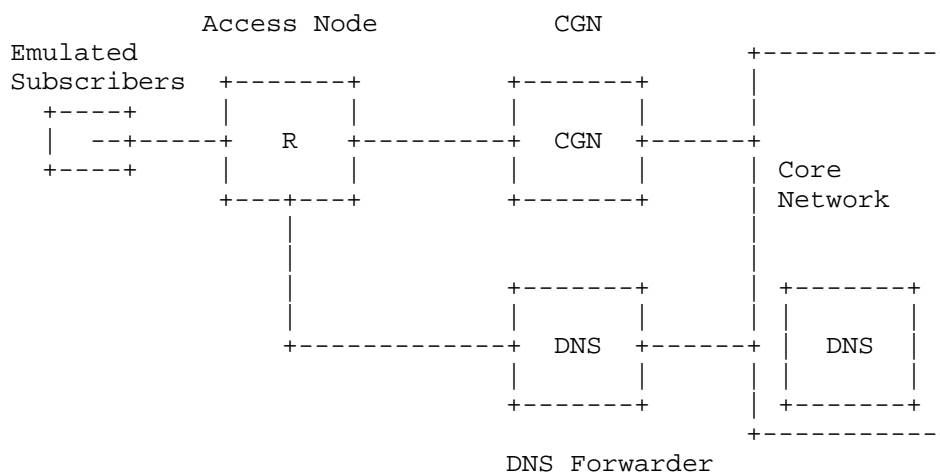


Figure 4: Bypassing of DNS queries using DNS forwarder.

In the first case, the original DNS server IP address in the service provider network is distributed to the subscribers. The emulated subscribers use the DNS server to get host IP address by name, so all query packets go through the CGN. The generated DNS query is 12M at the speed of 10k query per sec. In the second case, IP address of a DNS server placed in the bypassing position of CGN is distributed to users. The second DNS server works as a forwarder, so all queries are forwarded to the first DNS server. Therefore, all DNS queries are bypassed from the CGN while data traffic is still going through the CGN.

As the result, it was shown that DNS query almost does not affect the performance of the CGN. The max concurrent sessions of DNS packet was only 40k. NAT table of DNS (udp/53,tcp/53) timeouts in 3 seconds, thus It saves the consumption of NAT tables. However NAT log was generated for every query and it doubled the total amount of the log. It would be rare that the NAT log of DNS is needed to react to a legal request. The impact of the DNS query traffic is relatively small if DNS timeout is adjusted.

#### 6.4. Separation of traffic

In the existing network, IPv4 communication and IPv6 communication may already be mixed in the dual stack. In this case, by introducing CGN which can route IPv6 and existing IPv4 aside from NAT function, the influence for the network architecture could be suppressed and so a flexible design is possible. However, though current CGN is scalable enough to be deployed in core of the service providers network, the feature of routing is insufficient to replace the existing routers. Such a CGN is desirable, otherwise the design which makes IPv6 traffic and traditional IPv4 traffic bypass from CGN is effective choice for providers. In dividing NAT flows and non-NAT flows routers, VRF (Virtual Routing and Forwarding) and PBR (policy based routing) are needed at routers in front of CGN. In that case it is indispensable to configure routers so that the hairpinning communication between the NAT user and non-NAT user to be possible. The considerations about the separation of traffic and effective deployment configuration are discussed in detail in [I-D.ietf-opsawg-lsn-deployment].

#### 7. Tested web sites and applications (Excerpts)

- Web Mail
  - gmail
  - yahoo mail
  - hot mail
- Video
  - ustream
  - youtube
  - nicovideos
  - Hulu
  - dailymotion
  - daum
  - qq
  - fc2
  - xvideos
- Portal&EC site
  - yahoo
  - rakuten
  - amazon
  - apple
- Blog
  - livedoor blog

- ameba blog
- Search Engine
  - google
- Online Banking
  - mizuho bank
  - DC card
- Cloud Service
  - drop box
  - Evernote
- InstantMessenger & VoIP
  - skype
  - Line
- facebook
- twitter
- google map
- Online PC Game
  - aeria games
  - ameba pigg
  - nexon
  - hangame
- Consumer Game
  - Armored Core V (Play Station3)
  - Dark Souls 2 (Play Station3)
  - Gundam Extreme VS. (Play Station3)
  - Kinect adventure (XBox)
  - Persona 4 the ultimate in mayonaka arena (XBox)
  - Mingol 4 (WiiU)
  - Monster Hunter 3G (DS-lite)
  - Keri-hime sweets (iOS)
  - PuzzDra (iOS)

## 8. IANA Considerations

This document makes no request of IANA.

## 9. Security Considerations

TBD

## 10. Acknowledgments

This research and experiment are conducted under the great support of Ministry of Internal Affairs and Communications of Japan. Many thanks to MIC, JAIST members and Shin Miyakawa for their ideas and feedback in documentation.

## 11. References

### 11.1. Normative References

- [I-D.donley-behave-deterministic-cgn]  
Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier Grade NAT Deployments", draft-donley-behave-deterministic-cgn-06 (work in progress), July 2013.
- [I-D.ietf-opsawg-lsn-deployment]  
Kuarsingh, V. and J. Cianfarani, "CGN Deployment with BGP/MPLS IP VPNs", draft-ietf-opsawg-lsn-deployment-03 (work in progress), June 2013.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC2993] Hain, T., "Architectural Implications of NAT", RFC 2993, November 2000.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.
- [RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.

### 11.2. Informative References

- [I-D.chen-sunset4-cgn-port-allocation]  
Chen, G., "Analysis of NAT64 Port Allocation Method", draft-chen-sunset4-cgn-port-allocation-02 (work in progress), July 2013.
- [I-D.ietf-softwire-map]  
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-08 (work in progress), August 2013.
- [I-D.tsou-behave-natx4-log-reduction]  
Tsou, T., Li, W., Taylor, T., and J. Huang, "Port Management To Reduce Logging In Large-Scale NATs", draft-tsou-behave-natx4-log-reduction-04 (work in progress), July 2013.

progress), July 2013.

- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, April 2013.

#### Authors' Addresses

Kaname Nishizuka  
NTT Communications Corporation  
Granpark Tower  
3-4-1 Shibaura, Minato-ku  
Tokyo 108-8118  
Japan

Email: kaname@nttv6.jp

Daigo Natsume  
NTT-Neomeit Corporation  
3-15 Babacho, Chuo-ku, Osaka-shi  
Osaka 540-8511  
Japan

Email: daigo.natsume@ntt-neo.co.jp





BEHAVE  
Internet-Draft  
Intended status: Standards Track  
Expires: April 03, 2014

T. Reddy  
Ram. Ravindranath  
Muthu. Perumal  
Cisco  
A. Yegin  
Samsung  
September 30, 2013

Problems with STUN Authentication for TURN  
draft-reddy-behave-turn-auth-04

Abstract

This document discusses some of the issues with STUN authentication for TURN messages.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 03, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Notational Conventions . . . . .	3
3. Scope . . . . .	3
4. Problems with usage of STUN Authentication . . . . .	3
5. Security Considerations . . . . .	4
6. IANA Considerations . . . . .	5
7. Acknowledgments . . . . .	5
8. References . . . . .	5
8.1. Normative References . . . . .	5
8.2. Informative References . . . . .	5
Authors' Addresses . . . . .	6

## 1. Introduction

The TURN server is a building block to support interactive, real-time communication using audio, video, collaboration, games, etc., between two peer web browsers using the Web Real-Time communication (WebRTC) [I-D.ietf-rtcweb-overview] framework. The use-case explained in "Simple Video Communication Service, enterprise aspects" (Section 3.2.5 of [I-D.ietf-rtcweb-use-cases-and-requirements]) refers to deploying a TURN[RFC5766] server in the DMZ to audit all media sessions from inside an Enterprise premises to any external peer. TURN server could also be deployed for RTP Mobility [I-D.wing-mmusic-ice-mobility] etc.

TURN server is also used in the following scenarios:

- o Users of RTCWEB based web application may use TURN server to hide host candidate addresses from the remote peer for privacy.
- o Enterprise networks deploy firewalls which typically block UDP traffic. When SIP user agents or WebRTC endpoints are deployed behind such firewalls, media cannot be sent over UDP across the firewall, but must be sent using TCP (which causes a different user experience). In such cases a TURN server deployed in the DMZ MAY be used to traverse Firewalls.
- o TURN Server may be used for IPv4-to-IPv6, IPv6-to-IPv6, and IPv6-to-IPv4 relaying [RFC6156].
- o ICE connectivity checks using server-reflexive candidates could fail when the endpoint is behind NAT that performs Address-dependent mapping. In such cases relayed candidate allocated from the TURN server is used for media.

STUN [RFC5389] specifies an authentication mechanism called the long-term credential mechanism. TURN [RFC5766] in section 4 specifies that TURN servers and clients MUST implement this mechanism and the TURN server MUST demand that all requests from the client be authenticated using this mechanism, or that a equally strong or stronger mechanism for client authentication be used.

In the above scenarios RTCWEB based web applications would use Interactive Connectivity Establishment (ICE) protocol [RFC5245] for gathering candidates. ICE agent can use TURN to learn server-reflexive and relayed candidates. If the TURN server requires the TURN request to be authenticated then ICE agent will use the long-term credential mechanism explained in section 10 of [RFC5389] for authentication and message integrity. TURN specification [RFC5766] in section 10 explains the importance of long-term credential mechanism to mitigate various attacks. With proposals like[I-D.thomson-mmusic-rtcweb-bw-consent] that defines a STUN BANDWIDTH attribute for requesting bandwidth allocation at a TURN server, STUN authentication becomes further important to prevent unauthorized users from accessing the TURN server and misuse of credentials could impose significant cost on the victim TURN server.

This note focuses on listing the problems with current STUN authentication for TURN so that it can serve as the basis for stronger authentication mechanisms.

Compared to a Binding request the Allocate request is more likely to be identified by a server administrator as needing client authentication and integrity protection of messages exchanged. Hence, the issues discussed here in STUN authentication are applicable mainly in the context of TURN messages.

## 2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This note uses terminology defined in [RFC5389], [RFC5766].

## 3. Scope

This document can be used as an input to design solution(s) to address the problems with the current STUN authentication for TURN messages.

## 4. Problems with usage of STUN Authentication

1. The long-term credential mechanism in [RFC5389] could use traditional "log-in" username and password given to users which does not change for extended periods of time and uses the key derived from user credentials to generate message integrity for every TURN request/response. An attacker that is capable of eavesdropping on a message exchange between a client and server can determine the password by trying a number of candidate passwords and checking if one of them is correct by calculating the message-integrity of the message using these candidate passwords and comparing with the message integrity value in the MESSAGE-INTEGRITY attribute.
2. When TURN server is deployed in DMZ and requires requests to be authenticated using the long-term credential mechanism in [RFC5389], TURN server needs to be aware of the username and password to validate the message integrity of the requests and to provide message integrity for responses. This results in management overhead on the TURN server.
3. The long-term credential mechanism in [RFC5389] requires that the TURN client must include username value in the USERNAME STUN attribute. An adversary snooping the TURN messages between the TURN client and server can identify the users involved in the call resulting in privacy leakage. In certain scenarios TURN usernames need not be linked to any real usernames given to users as they are just provisioned on a per company basis.
4. An Attacker posing as a TURN server challenges the client to authenticate, learns the USERNAME of the client and later snoops the traffic from the client identifying the user activity resulting in privacy leakage.
5. Hosting multiple realms on a single IP address is challenging with TURN. When a TURN server needs to send the REALM attribute in response to an unauthenticated request, it has no useful information for determining which realm it should send, except the source transport address of the TURN request. Note this is a problem with multi-tenant scenarios only. This may not be a problem when TURN server is located in enterprise premises.
6. In WebRTC the Javascript needs to know the username and password to use in W3C RTCPeerConnection API to access the TURN server. This exposes the user credentials to the Javascript which could be malicious.

## 5. Security Considerations

This document lists problems with current STUN authentication for TURN so that it can serve as the basis for stronger authentication mechanisms.

## 6. IANA Considerations

This document does not require any action from IANA.

## 7. Acknowledgments

Authors would like to thank Dan Wing, Harald Alvestrand, Sandeep Rao, Prashanth Patil, Pal Martinsen and Simon Perreault for their comments and review.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.
- [RFC6156] Camarillo, G., Novo, O., and S. Perreault, "Traversal Using Relays around NAT (TURN) Extension for IPv6", RFC 6156, April 2011.

### 8.2. Informative References

- [I-D.ietf-rtcweb-overview] Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-08 (work in progress), September 2013.
- [I-D.ietf-rtcweb-use-cases-and-requirements] Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use-cases and Requirements", draft-ietf-rtcweb-use-cases-and-requirements-11 (work in progress), June 2013.
- [I-D.thomson-mmusic-rtcweb-bw-consent]

Thomson, M. and B. Aboba, "Bandwidth Constraints for Session Traversal Utilities for NAT (STUN)", draft-thomson-mmusic-rtcweb-bw-consent-00 (work in progress), October 2012.

[I-D.wing-mmusic-ice-mobility]

Wing, D., Reddy, T., Patil, P., and P. Martinsen, "Mobility with ICE (MICE)", draft-wing-mmusic-ice-mobility-05 (work in progress), September 2013.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, March 2012.

#### Authors' Addresses

Tirumaleswar Reddy  
Cisco Systems, Inc.  
Cessna Business Park, Varthur Hobli  
Sarjapur Marathalli Outer Ring Road  
Bangalore, Karnataka 560103  
India

Email: tiredddy@cisco.com

Ram Mohan Ravindranath  
Cisco Systems, Inc.  
Cessna Business Park, Varthur Hobli  
Sarjapur Marathalli Outer Ring Road  
Bangalore, Karnataka 560103  
India

Email: rmohanr@cisco.com

Muthu Arul Mozhi Perumal  
Cisco Systems, Inc.  
Cessna Business Park  
Sarjapur-Marathahalli Outer Ring Road  
Bangalore, Karnataka 560103  
India

Email: mperumal@cisco.com

Alper Yegin  
Samsung  
Istanbul  
Turkey

Email: alper.yegin@yegin.org



Behave WG  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2014

B. Rajtar  
Hrvatski Telekom  
I. Farrer  
Deutsche Telekom AG  
A. Vizdal  
T-Mobile CZ  
X. Li  
C. Bao  
CERNET Center/Tsinghua University  
July 15, 2013

Framework for accessing IPv6 content for IPv4-only clients  
draft-rfv1b-behave-v6-content-for-v4-clients-01

## Abstract

With the expansion of IPv6 usage and content available on IPv6, it is important that clients with legacy (i.e. non IPv6-capable) operating systems are able to access such content.

This document describes a method for achieving this, including how the method could be implemented in real-world scenarios.

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
1.1. Solution Requirements . . . . .	2
1.2. Covered Scenarios . . . . .	3
1.3. Functional elements . . . . .	3
2. Algorithm Description . . . . .	3
2.1. Flow diagram . . . . .	5
3. Usage scenarios . . . . .	5
4. IANA Considerations . . . . .	6
5. Security Considerations . . . . .	6
6. Acknowledgements . . . . .	6
7. Normative References . . . . .	6
Authors' Addresses . . . . .	6

## 1. Introduction

At the time of writing, IPv6 is still not widely deployed. There are several reasons for this, one of which is that IPv4-only operating systems are still commonplace with end-users and account for a large fraction of overall Internet traffic.

With the growth of IPv6 traffic, servers supporting only IPv6 are appearing on the Internet. An approach for enabling and IPv4-only clients to access this content is described below.

### 1.1. Solution Requirements

To clarify when this approach is applicable, the following requirements can be named:

1. The content MUST be reachable through IPv6, i.e. the server on which the content is stored must have a valid IPv6 address and a working IPv6 stack.
2. The server hosting the content MUST have a valid AAAA record
3. The client MUST support IPv4 only. The other alternative is also that it supports IPv6, but for some reason uses only IPv4 to access content on the Internet.
4. Client's DNS queries MUST be resolved by a dedicated appliance, i.e. a caching nameserver.
5. All traffic between the client and the server MUST be routed through a device capable of performing translation between IPv4 and IPv6, as described in [RFC6145] and [RFC6052].

It is feasible that requirements (4) and (5) can be combined in one device and managed by the service provider. That would simplify operations and remove the need for a control-plane protocol between the two devices.

#### 1.2. Covered Scenarios

[RFC6144] describes multiple scenarios for IPv4/IPv6 translation. This document is mainly concerned with Scenario 4: An IPv4 Network to the IPv6 Internet, but is also applicable to Scenario 6 (An IPv4 Network to an IPv6 Network). This scenario is not covered in this memo and can be elaborated in future documents, as necessary. Scenario 2, which faces similar challenges (The IPv4 Internet to an IPv6 Network), is covered by [I-D.draft-sun-behave-v4tov6-00].

#### 1.3. Functional elements

Client    User end-device, typically a personal computer or similar.

DNS proxy    Caching nameserver which proxies DNS queries from the client.

NAT46 translator    Translation device which translates incoming IPv4 traffic.

IPv6-only server    Device which holds content on an IPv6-only network.

## 2. Algorithm Description

This section describes how the algorithm works and the roles of every functional element. The steps are in chronological order, and display

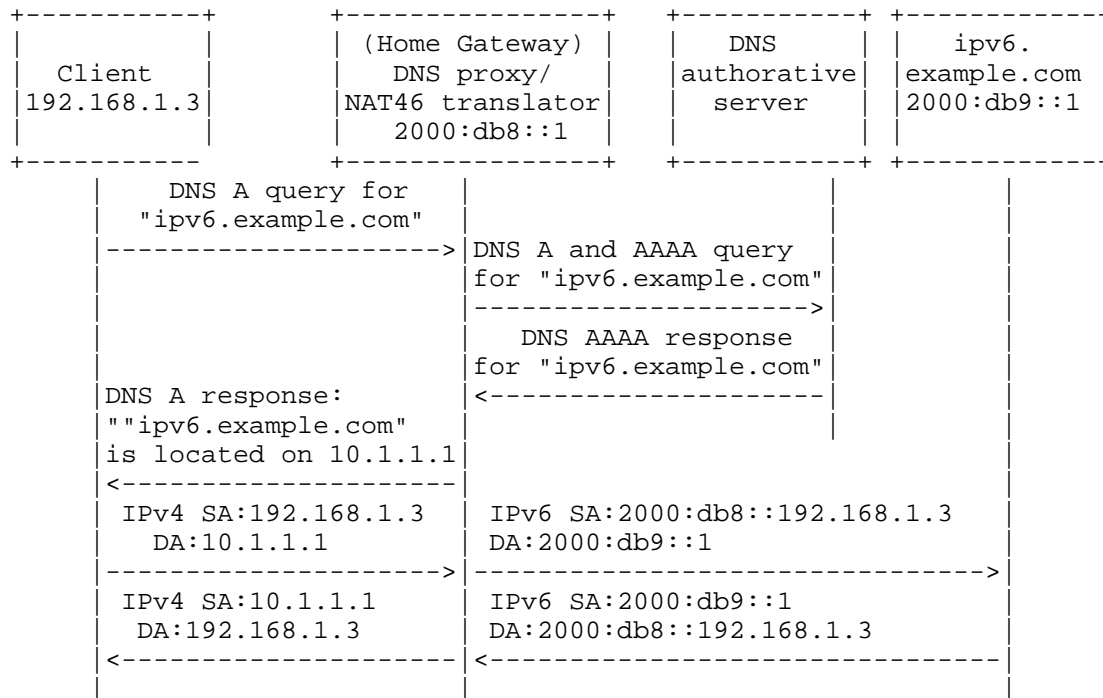
the scenario when the IPv4 client initiates a request for `ipv6.example.com` which is running on an IPv6-only server.

1. The customer types in "`ipv6.example.com`" into his web browser and initiates the request for the web page.
2. The client operating system initiates a DNS query for "`ipv6.example.com`". Since the client uses IPv4, the query is for an A record.
3. The DNS proxy receives the A record query and assumes the client is not IPv6 capable. Therefore, it initiates a DNS query for A and AAAA records for "`ipv6.example.com`" to the authoritative DNS server.
4. If a DNS response is received with only an AAAA record, the DNS proxy assumes that the server is IPv6-only. (In case the proxy receives both A or AAAA records, or just an A record, the A record is returned to the client and the process ends here.)
5. As a response to the client, the proxy returns a fake A record for "`ipv6.example.com`" pointing at an un-used IPv4 address from the private address space (as described in [RFC1918]).
6. The private IPv4 address and the resolved IPv6 address of "`ipv6.example.com`" must be kept in the translation table of the NAT46 translator. The time the translation would stay active in the table would be equal to the TTL field of the DNS response. How the DNS-related information is conveyed from the DNS proxy to the translator is out of the scope of this document. In the case the translator and the DNS proxy are functions of the same device, the logic is simplified.
7. All IPv4 traffic from the client to "`ipv6.example.com`" will be translated to IPv6 as described in [RFC6145]. Unlike NAT-PT described in [RFC2766] (moved to Historic Status by [RFC4966]), the translation is a learned state and not a session triggered state. The destination address of the translated IPv6 packet will be the resolved AAAA record of "`ipv6.example.com`", while the source IPv6 address will be created according to [RFC6052]. The IPv6 prefix used to create the source IPv6 address must be globally unique and allocated to the device. If there are more IPv6 prefixes on the device, defining which one will be used is out of the scope of this document. The IPv4 address used to create the source IPv6 address is the address of the client.
8. Return IPv6 traffic will be translated by the same device as the outgoing traffic, using IPv6 to IPv4 translation analogous to the

previous step. The source IPv4 address will be the private IPv4 address given by the DNS proxy to the client, while the destination IPv4 address would be the one of the client.

### 2.1. Flow diagram

In this example, the client is located behind a home gateway and is delegated an IPv4 address of 192.168.1.3. The home gateway is acting as a DNS proxy and as a NAT46 translator.



### 3. Usage scenarios

The typical scenario where such a solution can be used is the home network. The customer can have a broadband service with access to IPv6 Internet, but uses an IPv4-only client. The DNS proxy and the translation device would in that case be the home gateway, which would handle the decision-making process, as well as the translation.

However, other scenarios can also be foreseeable, such as mobile access, business customers, etc. It's applicable to all scenarios where a DNS proxy is used, as well as a default gateway which can act as a translation device.

#### 4. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

#### 5. Security Considerations

#### 6. Acknowledgements

#### 7. Normative References

[I-D.draft-sun-behave-v4tov6-00]  
 , .

[RFC1918] , "Address Allocation for Private Internets", .

[RFC2119] , "Key words for use in RFCs to Indicate Requirement Levels", .

[RFC2766] , "Network Address Translation - Protocol Translation (NAT-PT)", .

[RFC4966] , "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", .

[RFC6052] , "IPv6 Addressing of IPv4/IPv6 Translators", .

[RFC6144] , "Framework for IPv4/IPv6 Translation", .

[RFC6145] , "IP/ICMP Translation Algorithm", .

#### Authors' Addresses

Branimir Rajtar  
Hrvatski Telekom  
Zagreb  
Croatia

Email: branimir.rajtar@t.ht.hr

Ian Farrer  
Deutsche Telekom AG  
Bonn  
Germany

Email: [ian.farrer@telekom.de](mailto:ian.farrer@telekom.de)

Ales Vizdal  
T-Mobile CZ  
Prague  
Czech Republic

Email: [ales.vizdal@t-mobile.cz](mailto:ales.vizdal@t-mobile.cz)

Xing Li  
CERNET Center/Tsinghua University  
Beijing  
China

Email: [xing@cernet.edu.cn](mailto:xing@cernet.edu.cn)

Congxiao Bao  
CERNET Center/Tsinghua University  
Beijing  
China

Email: [congxiao@cernet.edu.cn](mailto:congxiao@cernet.edu.cn)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 30, 2014

C. Xie  
Q. Sun  
Q. He  
China Telecom  
C. Zhou  
Huawei Technologies  
X. Li  
C. Bao  
CERNET Center/Tsinghua  
University  
July 29, 2013

The Approach for IPv4-only users to access IPv6-only Content  
draft-sun-behave-v4tov6-01

Abstract

Current approaches can not solve the scenario that the users from IPv4 Internet to access IPv6-only content. When IPv6 content are becoming more and more popular, it is important to ensure that IPv6-only content can be reachable from legacy IPv4-only clients via some IPv4-only network. This document proposes two approaches for IPv4-only users to access IPv6-only content. It is designed to cover the Scenario 2 in [RFC6144].

Status of this Memo

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

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

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

This Internet-Draft will expire on January 30, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.



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

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. The NAT46 translator for IPv4 Internet to access IPv6 network . . . . .	4
4. Approach 1: DNS-based solution . . . . .	4
5. Approach 2: Redirect-based Solution . . . . .	6
6. IANA Considerations . . . . .	8
7. Acknowledgements . . . . .	8
8. References . . . . .	8
8.1. Normative References . . . . .	8
8.2. Informative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

In [RFC6144], Scenario 2 is an important use case. Not only could servers move directly to IPv6 without trudging through a difficult transition period, but they could do so without risk of losing connectivity with the IPv4-only Internet.

Existing solutions have not solved this scenario well. NAT-PT[RFC2766] can be used in this scenario, but it requires a tightly coupled DNS Application Level Gateway (ALG) in the translator, and have been deprecated by the IETF [RFC4966]. The stateless translation solution [RFC6219] can work too, but since each IPv6 server will consume one IPv4 public address, it is not suitable to deploy in situation that operators are running out of IPv4 address. [RFC6156] can be used for IPv4 client to communicate with IPv6 client. But this requires the IPv4 client and IPv6 client to implement a TURN client. Therefore, it is not suitable for C-S(Client-Server) and B-S (Browser-Server) mode.

[I-D.rfv1b-behave-v6-content-for-v4-clients] can work for IPv4-only user to access IPv6 content. But since it uses private IPv4 address to mapping the IPv6 server, it can only be used for IPv4 network to reach IPv6 network.

This document is designed for IPv4 Internet to reach IPv6 network. There are several requirements in this design:

- 1.Considering IPv4 address has been a scarce resource, the amount of public IPv4 addresses consumed by the translator should be less than that the number of IPv6 servers in the IPv6 network.
2. It should not require extra modifications on the server, e.g. by using a dynamic port number, implementing TURN client, etc.

In this document, we propose two approaches for this scenario. These two approaches can make use of existing DNS architecture. The binding table in these two approaches are static. Therefore, there will be no dynamic issue as in NAT-PT or DNS cache synchronization.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Terminology defined in [RFC6144] is used extensively in this document. Besides, this document uses the following terminologies:

IPv6-converted addresses: IPv4 addresses used to represent IPv6 nodes in an IPv4 Internet. They have an explicit mapping relationship to IPv6 addresses.

NAT46: a stateful IPv4/IPv6 translation functionality. It is consistent with IP/ICMP translation [RFC6145], and can also support IPv6-converted address selection and binding table maintenance.

### 3. The NAT46 translator for IPv4 Internet to access IPv6 network

The NAT46 solution is used for IPv4 clients in IPv4 Internet to reach IPv6 servers (depicted in Figure 1).

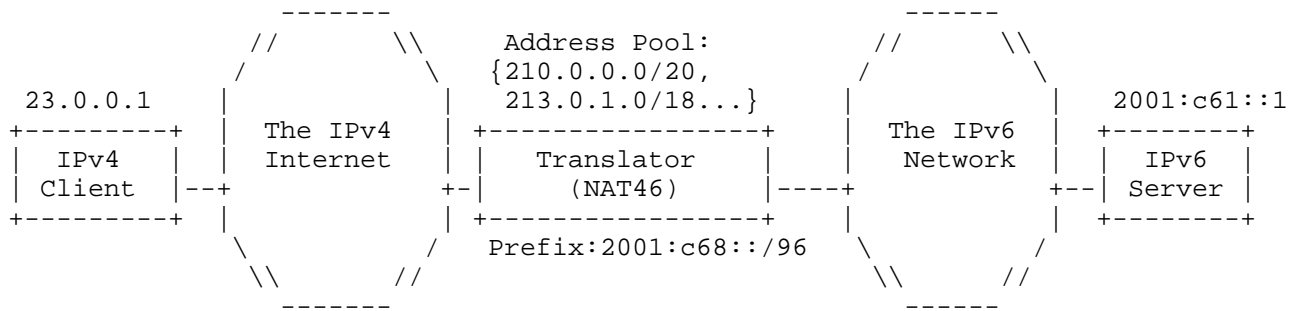


Figure 1: Overall solution for IPv4 Internet to IPv6 network

In order to achieve the translation initiated from IPv4 side, two addresses need to be determined by NAT46 translator. The first one is the IPv6-converted address of the IPv6 server, which is selected from the IPv4 address pool configured in NAT46. The second one is the IPv4-converted address for the IPv4 client, which can be synthesized using the stateless approach defined in [RFC6052].

In our approach, the mapping relationship between IPv6-converted address and the IPv6 address for the server is pre-determined in advance. As a result, the A record in the DNS server for a particular server is always the same for different IPv4 clients.

### 4. Approach 1: DNS-based solution

This approach is independent of translated protocol. For applications without DNS process can not be solved by this approach. In order to support IPv4 address sharing for multiple IPv6 servers, one IPv4 address can be shared by multiple servers with different service ports. If there are too many servers with the same service

port, the second approach can be used as a complement.

The overall solution is depicted in Figure 3.

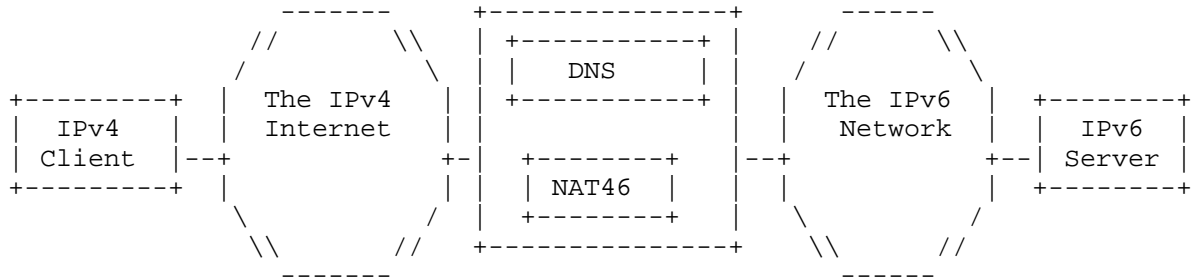


Figure 2: DNS-based approach

It consists of several functionalities:

1.NAT46: This functionality achieves the translation between IPv4 packet and IPv6 packet. It is consistent with [RFC6145]. Besides, it maintained the binding table including the IPv6 server address, IPv6-converted address for IPv6 server, and the service port statically.

2.DNS: The DNS server is configured with the IPv6-converted address as the A record for IPv6 server.

The workflow of this approach is as follows:

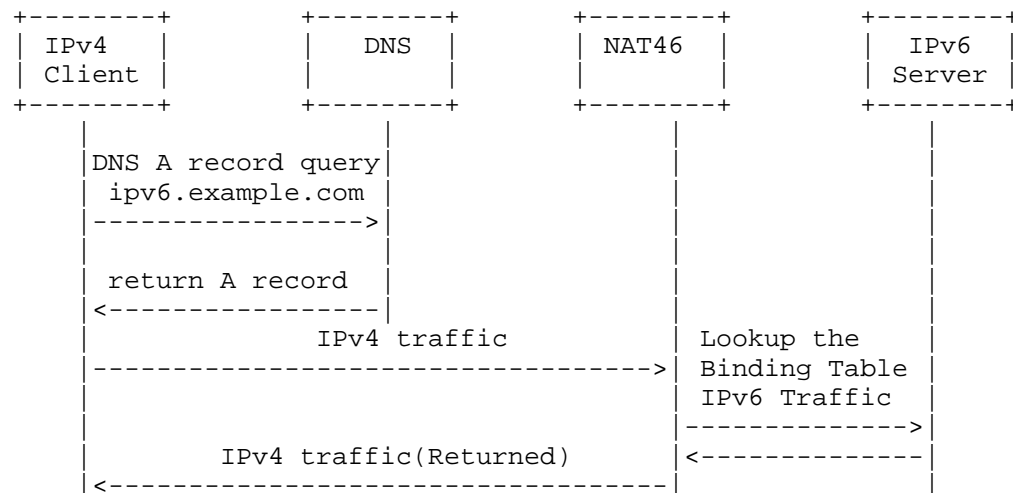


Figure 3: Workflow of DNS-based approach

1. An IPv4 client initiates a DNS query for A record (e.g. ipv6.example.com).
2. DNS receives the query. As it is configured with the IPv6-converted address in advance, the A record will be returned to the IPv4 client.
3. IPv4 client sends IPv4 traffic with the returned IPv6-converted address as the destination address.
4. When the IPv4 traffic arrives at the NAT46, NAT46 extracts the destination address and destination port in the IPv4 traffic. It will lookup the binding table maintained in NAT46 and NAT46 translates the IPv4 packet to IPv6 packet according to [RFC6145]. No port translation will be performed here.
5. The return traffic is treated in the same way.

#### 5. Approach 2: Redirect-based Solution

This approach is designed for HTTP application. It will have a high address sharing ratio. In HTTP, since the traffic can be redirected to a different service port, it is able to achieve address sharing for IPv6 servers by using different ports (denoted as IPv6-converted port). Therefore, one IPv4 address can support up to thousands of IPv6 servers in theory.

The overall solution is depicted in Figure 4.

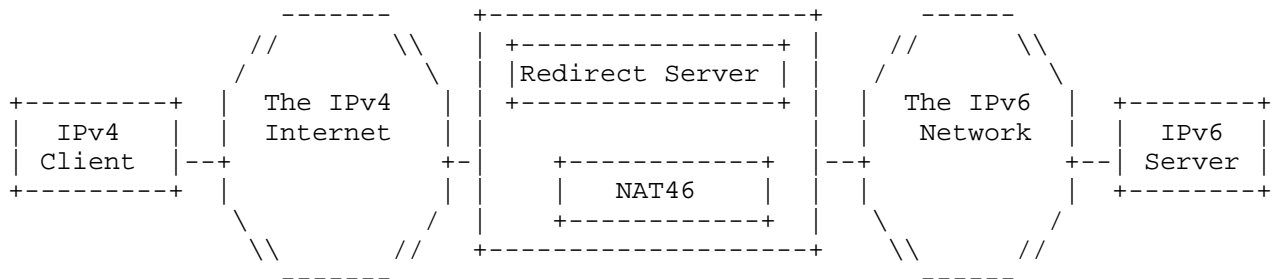


Figure 4: Redirect-based Solution

It consists of several functionalities:

1. NAT46: This functionality is basically the same as the first

approach, except for the binding table includes IPv6 server address, IPv6 service port, IPv6-converted address and IPv6-converted port.

2. Redirect Server: A Redirect server is used to redirect traffic to a different IPv6-converted address and IPv6-converted port. The redirect server may either store the binding table, or query for the redirected address and port from NAT46.

The workflow of this approach is as follows:

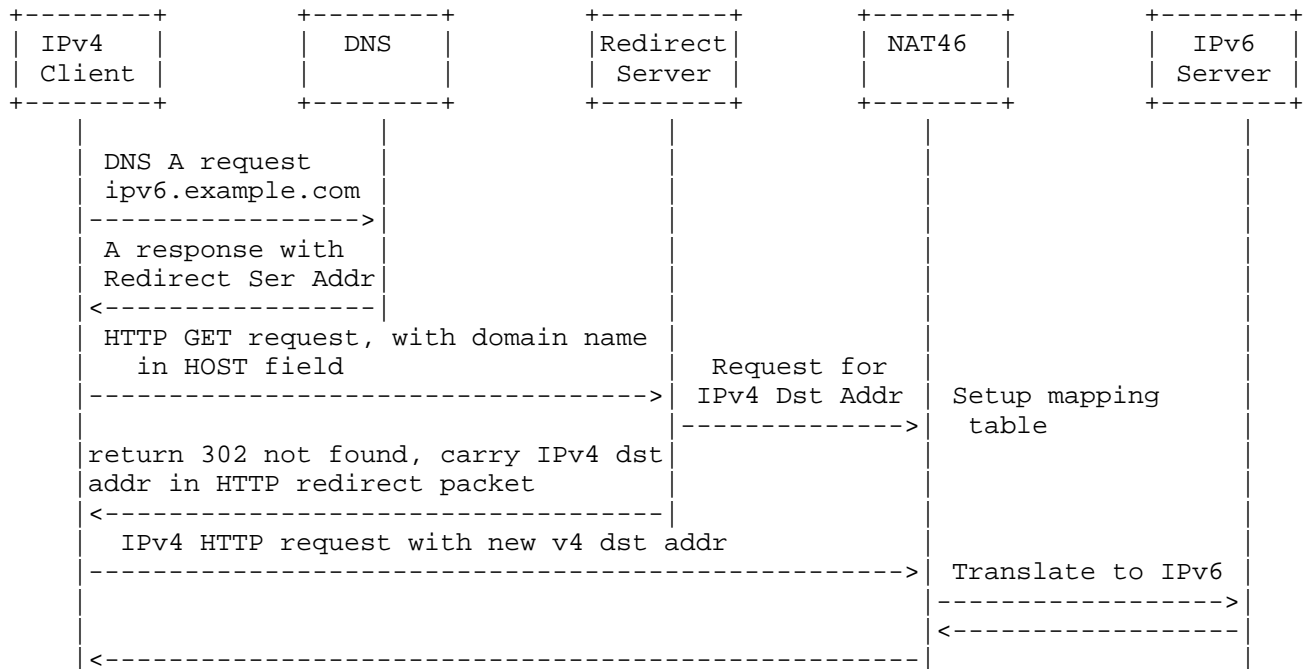


Figure 5: Workflow of Proxy-lite Approach

1. An IPv4 client initiates a DNS query for A record (e.g. ipv6.example.com).
2. In DNS server, the address of the redirect server is configured as the A record for ipv6.example.com and returns to the IPv4 client.
3. The IPv4 client sends HTTP GET request. The domain name (e.g. ipv6.example.com) is carried in HOST field.
4. The redirect server interprets the domain name, and sends the request to get IPv6-converted address to NAT46 (carrying the address

of IPv4 client and the destination port). The specific protocol for the request is now out of scope.

5. NAT46 selects IPv6-converted address and IPv6-converted port by lookuping the binding table. It will also keep the destination port in the binding table.

6. NAT46 returns the IPv6-converted address and IPv6-converted port to redirect server and the redirect server in turn returns IPv6-converted address in HTTP redirect packet with HTTP error "302 not found".

7. IPv4 client replaces the destination IPv4 address with the returned IPv6-converted address. The IPv4 traffic is routed to the NAT46.

8. extracts the destination address and destination port in the IPv4 traffic. It will lookup the binding table maintained in NAT46 and NAT46 translates the IPv4 packet to IPv6 packet according to [RFC6145].

## 6. IANA Considerations

No requirement on IANA.

## 7. Acknowledgements

The authors would like to thank Dan Wing, Fred Baker for their review and comments.

## 8. References

### 8.1. Normative References

- [I-D.rfv1b-behave-v6-content-for-v4-clients]  
Rajtar, B., Farrer, I., Ales, V., Li, X., and C. Bao,  
"Framework for accessing IPv6 content for IPv4-only  
clients", draft-rfv1b-behave-v6-content-for-v4-clients-01  
(work in progress), July 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address  
Translation - Protocol Translation (NAT-PT)", RFC 2766,

February 2000.

- [RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6156] Camarillo, G., Novo, O., and S. Perreault, "Traversal Using Relays around NAT (TURN) Extension for IPv6", RFC 6156, April 2011.
- [RFC6219] Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, "The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition", RFC 6219, May 2011.

## 8.2. Informative References

### Authors' Addresses

Chongfeng Xie  
China Telecom  
P.R.China

Phone: 86 10 58552116  
Email: xiechf@ctbri.com.cn



Qiong Sun  
China Telecom  
P.R.China

Phone: 86 10 58552936  
Email: sunqiong@ctbri.com.cn

Qi He  
China Telecom  
P.R.China

Phone: 86 10 58552332  
Email: heqi@ctbri.com.cn

Cathy Zhou  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Phone:  
Email: cathy.zhou@huawei.com

Xing Li  
CERNET Center/Tsinghua University  
Room 225, Main Building  
Beijing 100084  
P.R.China

Phone: +86 10 6278 5983  
Email: xing@cernet.edu.cn

Congxiao Bao  
CERNET Center/Tsinghua University  
Room 225, Main Building  
Beijing 100084  
P.R.China

Phone: +86 10 6278 5983  
Email: congxiao@cernet.edu.cn



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2014

J. Uberti  
Google  
July 15, 2013

A REST API For Access To TURN Services  
draft-uberti-behave-turn-rest-00

## Abstract

This document describes a proposed standard REST API for obtaining access to TURN services via ephemeral (i.e. time-limited) credentials. These credentials are vended by a web service over HTTP, and then supplied to and checked by a TURN server using the standard TURN protocol. The usage of ephemeral credentials ensures that access to the TURN server can be controlled even if the credentials can be discovered by the user, as is the case in WebRTC where TURN credentials must be specified in Javascript.

## Status of This Memo

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

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

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

This Internet-Draft will expire on January 16, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. HTTP Interactions . . . . .	3
2.1. Request . . . . .	3
2.2. Response . . . . .	4
3. WebRTC Interactions . . . . .	5
4. TURN Interactions . . . . .	5
4.1. Client . . . . .	5
4.2. Server . . . . .	5
5. Implementation Notes . . . . .	6
5.1. Revocation . . . . .	6
5.2. Key Rotation . . . . .	6
6. Security Considerations . . . . .	6
7. IANA Considerations . . . . .	7
8. Acknowledgements . . . . .	7
9. References . . . . .	7
9.1. Normative References . . . . .	7
9.2. Informative References . . . . .	7
Author's Address . . . . .	8

## 1. Introduction

TURN [RFC5766] is a protocol that is often used to improve the connectivity of P2P applications. By providing a cloud-based relay service, TURN ensures that a connection can be established even when one or both sides is incapable of a direct P2P connection. However, as a relay service, it imposes a nontrivial cost on the service provider. Therefore, access to a TURN service is almost always access-controlled.

TURN provides a mechanism to control access via [RFC5389] long-term credentials that are provided as part of the TURN protocol. It is expected that these credentials will be kept secret; if the credentials are discovered, the TURN server could be used by unauthorized users or applications. However, in web applications, ensuring this secrecy is typically impossible.

To address this problem, this document proposes an API that can be used to retrieve ephemeral TURN credentials from a web service. These credentials can then be used as long-term credentials with a standard TURN server with a custom authentication module. For simplicity, the design has been kept intentionally stateless; the only interaction needed between the web service and the TURN service is to share a secret key.

## 2. HTTP Interactions

To retrieve a new set of credentials, the client makes a HTTP GET request, specifying TURN as the service to allocate credentials for, and optionally specifying a user id parameter. The purpose of the user id parameter is to simplify debugging on the TURN server, as well as provide the ability to control the number of credentials handed out for a specific user, if desired. The TURN credentials and their lifetime are returned as JSON, along with URIs that indicate how to connect to the server using the TURN protocol.

To avoid the need for state passing between the web service and TURN server, the returned credentials consist of a TURN username that encodes all the necessary state (expiry time and application user id), and a TURN password that is a digest of this state, signed with the shared secret key.

Since the returned credentials are ephemeral, they will eventually expire. This does not affect existing TURN allocations, as they are tied to a specific 5-tuple, but requests to allocate new TURN ports will fail after the expiry time. This is significant in the case of an ICE restart, where the client will need to allocate a new set of candidates, including TURN candidates. To get a new set of ephemeral credentials, the client can simply re-issue the original HTTP request with the same parameters, which will return the new credentials in its JSON response.

To prevent unauthorized use, the HTTP requests can be ACLed by various means, e.g. IP address (if coming from a server), Origin header, User-Agent header, login cookie, API key, etc.

### 2.1. Request

The request includes the following parameters, specified in the URL:

- o service: specifies the desired service (turn)
- o username: an optional user id to be associated with the credentials

- o key: if an API key is used for authentication, the API key

Example:

```
GET /?service=turn&username=mbzrxpgjys
```

## 2.2. Response

The response is returned with content-type "application/json", and consists of a JSON object with the following parameters:

- o username: the TURN username to use, which is a colon-delimited combination of the expiration timestamp and the username parameter from the request (if specified). The timestamp is intended to be opaque to the web application, so its format is arbitrary, but for simplicity, use of UNIX timestamps is recommended.
- o password: the TURN password to use; this value is computed from the a secret key shared with the TURN server and the returned username value, by performing `base64(hmac(secret key, returned username))`. HMAC-SHA1 is one HMAC algorithm that can be used, but any algorithm that incorporates a shared secret is acceptable, as long as both the web server and TURN server use the same algorithm and secret.
- o ttl: the duration for which the username and password are valid, in seconds. A value of one day (86400 seconds) is recommended.
- o uris: an array of TURN URIs, in the form specified in [I-D.petithuguenin-behave-turn-uris]. This is used to indicate the different addresses and/or protocols that can be used to reach the TURN server. These URIs SHOULD specify a hostname, IPv4, or IPv6 address for the TURN server, as well as the port and transport to use; this avoids the need for a DNS SRV or S-NAPTR lookup as specified in [RFC5928].

Example:

```
{
  "username" : "12334939:mbzrxpgjys",
  "password" : "adfsaflsjfldssia",
  "ttl" : 86400,
  "uris" : [
    "turn:1.2.3.4:9991?transport=udp",
    "turn:1.2.3.4:9992?transport=tcp",
    "turns:1.2.3.4:443?transport=tcp"
  ]
}
```

```
}
```

### 3. WebRTC Interactions

The returned JSON is parsed into an `RTCIceServer` object, and supplied as part of the `RTCCConfiguration` object that is used when creating a `RTCPeerConnection`.

Example:

```
var iceServer = {  
  "username": response.username,  
  "credential": response.password,  
  "uris": response.uris  
};  
var config = {"iceServers": [iceServer]};  
var pc = new RTCPeerConnection(config);
```

When the credentials are updated (e.g. because they are about to expire), a new `RTCCConfiguration` with the updated credentials can be supplied to the existing `RTCPeerConnection` via the `updateIce` method. This update must not affect existing TURN allocations, because TURN requires that the username stay constant for an allocation, but the new credentials will be used for any new allocations.

[TODO: make sure this behavior is specified in the W3C API spec]

### 4. TURN Interactions

#### 4.1. Client

The WebRTC client will perform a standard TURN allocation sequence using the long-term credentials mechanism specified in [RFC5389], Section 10.2, using the "username" value from the returned JSON for its USERNAME attribute, and the "password" value for the password input to the MESSAGE-INTEGRITY hash.

#### 4.2. Server

The TURN server will process the request using the long-term credentials mechanism specified in [RFC5389]. Note that the REALM value supplied by the server is not meaningful in this context, and can be set to any valid value.

When processing ALLOCATE requests, the TURN server MUST split the USERNAME attribute into its timestamp and user id components, and

verify that the timestamp, which indicates when the credentials expire, has not yet been reached. If this verification fails, it SHOULD reject the request with a 401 (Unauthorized) error.

If desired, the TURN server can optionally verify that the parsed user id value corresponds to a currently valid user of an external service (e.g. is currently logged in to the web app that is making use of TURN). This requires proprietary communication between the TURN server and external service on each ALLOCATE request, and is not necessary for typical applications. If this external verification fails, it SHOULD reject the request with a 401 (Unauthorized) error.

For non-ALLOCATE requests, the TURN server merely verifies that the USERNAME matches the USERNAME that was used in the ALLOCATE (since it must remain constant).

As in RFC 5766, the TURN server MUST verify the MESSAGE-INTEGRITY using the password associated with the supplied USERNAME. For the usage outlined in this document, the password will always be constructed using the supplied username and the shared secret as indicated in the "HTTP Interactions" section above. Because the password is derived from the USERNAME, successful verification of the MESSAGE-INTEGRITY ensures that the USERNAME (and the expiration time contained within) is trustworthy.

## 5. Implementation Notes

### 5.1. Revocation

In the system as described here, revoking specific credentials is not possible. The assumption is that TURN services are of low enough value that waiting for the timeout to expire is a valid approach for dealing with possibly-compromised credentials.

In extreme abuse cases, TURN server blacklists of timestamp+username values can be supplied by an administrator to stop abuse of specific credential sets.

### 5.2. Key Rotation

As indicated in [RFC2104], periodic rotation of the shared secret to protect against key compromise is RECOMMENDED. To facilitate the rollover, the TURN server SHOULD be able to validate incoming MESSAGE-INTEGRITY tokens based on at least 2 shared secrets at any time.

## 6. Security Considerations



Because the USERNAME values in a TURN ALLOCATE request are typically visible to eavesdroppers, inclusion of an externally identifying user id, such as a login name, may allow a passive attacker to determine the identities of the parties in a conversation. To prevent this problem, use of opaque user id values is recommended.

This mechanism assumes that the clocks of the web server and TURN server are roughly in sync. Given the expected large TTLs for the vended credentials, clock skew on the order of seconds to minutes should not cause an issue. However, if the TURN server's clock was mistakenly set to a date significantly in the past, credentials could be accepted for far longer than their intended lifetime.

## 7. IANA Considerations

None.

## 8. Acknowledgements

Harald Alvestrand, Alfred Godoy, and Philipp Hancke provided key input on the initial design. Dave Cridland, Cullen Jennings, Oleg Moskalkenko, and Matthew Robertson pointed out several errors and omissions.

## 9. References

### 9.1. Normative References

- [I-D.petithuguenin-behave-turn-uris]  
Petit-Huguenin, M., Nandakumar, S., Salgueiro, G., and P. Jones, "Traversal Using Relays around NAT (TURN) Uniform Resource Identifiers", draft-petithuguenin-behave-turn-uris-03 (work in progress), January 2013.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

### 9.2. Informative References

- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,  
Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext  
Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC5928] Petit-Huguenin, M., "Traversal Using Relays around NAT  
(TURN) Resolution Mechanism", RFC 5928, August 2010.

Author's Address

Justin Uberti  
Google  
747 6th St S  
Kirkland, WA 98033  
USA

Email: [justin@uberti.name](mailto:justin@uberti.name)