State Migration
draft-gu-opsawg-policies-migration-01

Abstract

While Virtual Machine (VM) lively migrate around, not only the OS, memory, and the states on Hypervisor need to be migrated with VM, but also the states on the network side, e.g. on Firewall. Otherwise, the running services on the migrated VM could be disrupted, even stopped. In this draft, we describe the background and use cases of this proposal. We also raise a clear scope for the proposal.

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 May 3, 2012.

Copyright Notice

Copyright (c) 2011 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Terminologies and concepts</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>States On Firewalls</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>Session Table</td>
<td>4</td>
</tr>
<tr>
<td>3.2</td>
<td>Cumulative Data</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Scenarios for Migration of States on Firewall</td>
<td>5</td>
</tr>
<tr>
<td>4.1</td>
<td>VM Migration between different DCs</td>
<td>5</td>
</tr>
<tr>
<td>4.2</td>
<td>VM Migration under Distributed Deployed Firewalls</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Scope</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Security Considerations</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Acknowledgments</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>References</td>
<td>11</td>
</tr>
<tr>
<td>8.1</td>
<td>Normative Reference</td>
<td>11</td>
</tr>
<tr>
<td>8.2</td>
<td>Informative Reference</td>
<td>12</td>
</tr>
<tr>
<td>Authors’ Addresses</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>
1. Introduction

VM live Migration enable us to migrate a VM from one place to another place without significant interruption to the running service on the VM. VMware lists some benefits that VMotion, VMware’s VM live migration solution, can provide:

VMotion allows you to[VMotion]:

- Perform live migrations with zero downtime, undetectable to the user.
- Continuously and automatically optimize virtual machines within resource pools.
- Perform hardware maintenance without scheduling downtime and disrupting business operations.
- Proactively move virtual machines away from failing or underperforming servers.

VM Live Migration is a wonderful function to have for DC operators. However, some preconditions must be satisfied in order to make a successful live migration. One of the preconditions is that the flow-coupled state on network must be kept after VM migrates. A very obvious example of flow-coupled state is session table on Firewall. Assume that a VM migrates to a new place, which is under different Firewall from the original Firewall. If the session table, which records the existing connections to the VM, is lost, the following packets belonging to the existing connections will be dropped by the new Firewall, and the connections will finally be disconnected.

In the following sections, we will give more detail description of the problem with flow-coupled state in VM live migration. And we will conclude a feasible scope for further effort in IETF.

2. Terminologies and concepts

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].

Source Network Device, Source switch, or Source device: the network device/switch/device from where the VM migrates. I.E. VM is originally located under the source network device/switch/device.

Destination Network Device, Destination switch, or Destination
device: the network device/switch/device to where the VM migrates. I.E. VM is relocated to the destination network device/switch/device.

Virtual Machine (VM), A completely isolated operation system which is installed by software on a normal operation system. An normal operation system can be virtualized into several VM.

Firewall (FW), A policy based security device, typically used for restricting access to/from specific devices and applications.

3. States On Firewalls

There are two kinds of physical Firewall deployment in DCs.

One is to place a pair of centralized powerful Firewalls at WAN connect point. In this case, any traffic, even the traffic between VMs within the same LAN, need to pass the Firewall.

The third way is distributed deployed Firewall. In stead of place a powerful centralized Firewall at the WAN connect point, Firewall is distributively deployed at aggregation switches, even lower on access switches. The goal of this kind of distributed deployed Firewall is not to separate different security zones, but to off load the huge workload on centralized Firewall. This case is especially reasonable for large layer 2 network with tens even hundreds of thousands of Virtual Machines. To rely a centralized pair of Firewall to deal with traffic from such volume of VMs are not reliable and Firewall could be the bottleneck.

The following states are dynamically generated on Firewall.

3.1. Session Table

Firewall will establish session state for each connection to host within the DC. The host could a physical server or a VM. The session state includes most related information of the connection.

<table>
<thead>
<tr>
<th>Item</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src IP</td>
<td>Source IP Address of the connection</td>
</tr>
<tr>
<td>Dst IP</td>
<td>Destination IP Address of the connection</td>
</tr>
<tr>
<td>Src Port</td>
<td>Source Port Number used to establish the session</td>
</tr>
<tr>
<td>Dst Port</td>
<td>Destination Port Number used to establish the session</td>
</tr>
<tr>
<td>Protocol</td>
<td>Protocol type</td>
</tr>
<tr>
<td>VLAN</td>
<td>VLAN ID</td>
</tr>
</tbody>
</table>

3.2. Cumulative Data

In order to protect DC from attacks, Firewall will cumulate various kinds of data. Assuming a use case, where there are both individual clients and enterprise servers in the DC. An untrust client might attack the servers in the same DC. One example of attacks is SYN Flooding. The client keeps sending SYN message to a specific server, which will be a DOS attack to the server, or to any server, which will become a DOS attack to the Firewall. Firewall cumulate the SYN message from a client. If the frequency of SYN message exceed a pre-defined rate, the IP address of this client will be drawn into a black list. Same situation to DNS Flooding attack.

4. Scenarios for Migration of States on Firewall

The following are scenarios that we need to migrate Firewall states with VM when proceed VM live migration.

4.1. VM Migration between different DCs

China Telecom deploys several separated DCs in one province in West China. These DCs have been built for several years and been upgraded during these years. But any single DC is limited in scale because most of the DCs are built in downtown. When facing with the requirements from large Service Provider, none of any single DC can accommodate the huge requirements for racks of servers by itself. So China Telecom has to split SP’s requirements into multiple DCs. Interconnection between DCs must be provided to simulate a single DC, which is in order to enable inter-communication among the SP’s VMs and to enable VM live migration.

Here we provide an example architecture of above situation. A DC provider has two DCs on different locations. One is at City A and the other is at City B, which is 30 kilometers away form City A. We assume that the physical distance and network bandwidth between City A and B satisfy the requirements of VM live migration. Two DCs are interconnected by VPLS to make them in the same LAN. Each DC has a pair of Firewalls on Core Switch. VRRP(Virtual Router Redundancy Protocol) [VRRP]is deployed on GW1 and GW1’.

At the very beginning, VMs are evenly created on Pod1 and Pod2. With time past, Pod1 and Pod2 becomes overloaded. In order to guarantee SLA, and to accommodate more service, Pod3 is created and some of the
VMs on Pod1 and Pod2 are migrated to Pod3, and the running service must be kept during the migration.

... VM&#12288;Traffic

Figure 1: Example architecture

At payment day, a burst of access requests come to Finance Zone, the volume exceeds Server capability at Finance Zone 1. VM13 and some
other VM are migrated to Finance Zone 2 to utilize the idle resources in Finance Zone 2. The existing service on VM13 should be kept without disruption. So that the states on Firewall-2 that is related to VM13 should be migrated to Firewall-2'.

---

| GW1 |                                   | GW1' |
---|---|-------|

| /

---

VPLS-PE1

| /

---

States on FW1 for VM1

| FW1 | ******************************> |
---|---|

| /

---

CE1

| /

---

Aggregation Switch1

| /

---

Access Switch1

| /

---

VM1  VM2  VM3

| VM4' VM5' VM6'

| VM7  VM8  VM9 |

---

Pod1

---

Pod2

---

Pod3

Figure 2: VM and State Migration stage
4.2. VM Migration under Distributed Deployed Firewalls

In a DC with distributed deployed Firewalls on Aggregation Switches, an enterprise customer lease hundreds of physical servers, and each physical server carries 10 plus Virtual Machines (VM). The VMs provide VDI service to employees. At day time, the VMs are evenly deployed on each Pod3.
While at night, most of the VMs are shut down. Only a few VMs still working. In order to save energy, the active VMs are migrated to a few physical servers and the source physical servers, on which the migrated VM used to run, are shut down. The states on FW1’ need to be migrated to FW1, otherwise the running service on migrating VM will be disrupted.
5. Scope

SAMI (StAte MIgration) only considers the scenarios in which network conditions can satisfy the requirements raised by VM live migration. No matter VM is migrated within or between DCs, the scenario is in scope, as long as the network requirements for VM live migration can be satisfied. VM migration between L3 subnet, for now, is not in the scope. The solutions we develop in SAMI should enable both state migration within DC and between DCs, which is logically in the same Layer 2 network.

For the first stage, we only migrate Session tables on Firewall. But the solution should be extensible to enable migration of other states...
we may find that is necessary to be migrated during VM live migration.

We should always try to reuse existing IETF work to resolve SAMI problem. Only when there is no existing IETF work can use, with suitable extension, to achieve State migration, shall we develop a new mechanism to do this.

6. Security Considerations

The states described above are all about security. Besides, we need to be careful to avoid poisoned states from untrusted source. That means no matter how the states are migrated, authentication and verification are required.

7. Acknowledgments

The authors would like to thank the following people for contributing to this draft: Ning Zong, David Harrington, Linda Dunbar, Susan Hares, Serge Manning, Barry Leiba, Jiang Xingfeng, Song Wei, Robert Sultan.

8. References

8.1. Normative Reference


8.2. Informative Reference

[Data_Center_Fundamentals]
  "Data Center Fundamentals", 2003.

[I-D.wang-opsawg-policy-migration-gap-analysis]
  Wang, D. and Y. Gu, "I-D.wang-opsawg-policy-migration-gap-
  analysis", 2011.

[Vmotion_between_DCs]
  VMware, "VMotion between Data Centers--a VMware and Cisco
  networking/2009/06/
  vmotion-between-data-centersa-vmware-and-cisco-proof-of-
  concept.html)", June 2009.

           Virtualization", July 2011.

[Amazon_VPC_User_Guide]
  "http://docs.amazonwebservices.com/AmazonVPC/2011-07-15/
  UserGuide".


[LISP]     "Location/ID separation protocol,
           http://tools.ietf.org/wg/lisp/".

           Virtual_Router_Redundancy_Protocol".

Authors’ Addresses

Gu Yingjie
Huawei
No. 101 Software Avenue
Nanjing, Jiangsu Province  210001
P.R.China

Email: guyingjie@huawei.com
Yang Jingtao  
Huawei  
No. 101 Software Avenue  
Nanjing, Jiangsu Province  210001  
P.R.China  
Email: yangjingtao@huawei.com

Li Chen  
China Mobile  
Email: lichenyj@chinamobile.com

Xu Huiyang  
China Mobile  
Email: xuhuiyang@chinamobile.com

Li Kai  
China Telecom  
Email: leekai@ctbri.com.cn

Fan Yongbing  
China Telecom  
No. 109 Zhongshan Road West  
Guangzhou, Guangdong Province  
P.R.China  
Phone: 86-20-38639121  
Fax: 86-20-38639487  
Email: fanyb@gsta.com

Zhuo Zhiqiang  
Ruijie Network  
Email: zhuozq@ruijie.com.cn
Liu Ming
Ruijie Network

Email: lium@ruijie.com.cn
Abstract

This document defines common requirements for Carrier-Grade NAT (CGN). It updates RFC 4787.

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 June 9, 2013.

Copyright Notice

Copyright (c) 2012 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. Requirements for CGNs ....................................... 5
4. Logging .......................................................... 10
5. Port Allocation Scheme ..................................... 11
6. Deployment Considerations ................................ 12
7. IANA Considerations .......................................... 12
8. Security Considerations ..................................... 12
9. Acknowledgements ............................................. 13
10. References ....................................................... 13
    10.1. Normative References .............................. 13
    10.2. Informative Reference .......................... 14
Authors’ Addresses .................................................. 15
1. Introduction

With the shortage of IPv4 addresses, it is expected that more Internet Service Providers (ISPs) may want to provide a service where a public IPv4 address would be shared by many subscribers. Each subscriber is assigned a private address, and a Network Address Translator (NAT) [RFC2663] situated in the ISP’s network translates between private and public addresses. When a second IPv4 NAT is located at the customer edge, this results in two layers of NAT. This service can conceivably be offered alongside others, such as IPv6 services or regular IPv4 service assigning public addresses to subscribers. Some ISPs started offering such a service long before there was a shortage of IPv4 addresses, showing that there are driving forces other than the shortage of IPv4 addresses. One approach to CGN deployment is described in [RFC6264].

This document describes behavior that is required of those multi-subscriber NATs for interoperability. It is not an IETF endorsement of CGN or a real specification for CGN, but rather just a minimal set of requirements that will increase the likelihood of applications working across CGNs.

Because subscribers do not receive unique IPv4 addresses, Carrier Grade NATs introduce substantial limitations in communications between subscribers and with the rest of the Internet. In particular, it is considerably more involved to establish proxy functionality at the border between internal and external realms. Some applications may require substantial enhancements, while some others may not function at all in such an environment. Please see "Issues with IP Address Sharing" [RFC6269] for details.

This document builds upon previous works describing requirements for generic NATs [RFC4787][RFC5382][RFC5508]. These documents, and their updates if any, still apply in this context. What follows are additional requirements, to be satisfied on top of previous ones.

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].

Readers are expected to be familiar with "NAT Behavioral Requirements for Unicast UDP" [RFC4787] and the terms defined there. The following additional term is used in this document:
Carrier-Grade NAT (CGN): A NAT-based [RFC2663] logical function used to share the same IPv4 address among several subscribers. A CGN is not managed by the subscribers.

Note that the term "carrier-grade" has nothing to do with the quality of the NAT; that is left to discretion of implementers. Rather, it is to be understood as a topological qualifier: the NAT is placed in an ISP’s network and translates the traffic of potentially many subscribers. Subscribers have limited or no control over the CGN, whereas they typically have full control over a NAT placed on their premises.

Note also that the CGN described in this document is IPv4-only. IPv6 address translation is not considered.

However, the scenario in which the IPv4-only CGN logical function is used may include IPv6 elements. For example, DS-Lite [RFC6333] uses an IPv4-only CGN logical function in a scenario making use of IPv6 encapsulation. Therefore, this document would also apply to the CGN part of DS-Lite.

Figure 1 summarizes a common network topology in which a CGN operates.

```
                Internet
                ............ | ...................
                ISP network
    External pool:  |       ISP network
  192.0.2.1/26    |  Customer premises
                |    10.0.0.100    |    10.0.0.101
                |  CPE1   |  CPE2   | etc.
                | +-------++ +-------++
                +------++ +------++

(IP addresses are only for example purposes)
```

Figure 1: CGN network topology
Another possible topology is one for hotspots, where there is no customer premise or customer-premises equipment (CPE), but where a CGN serves a bunch of customers who don’t trust each other and hence fairness is an issue. One important difference with the previous topology is the absence of a second layer of NAT. This, however, has no impact on CGN requirements since they are driven by fairness and robustness in the service provided to customers, which applies in both cases.

3. Requirements for CGNs

What follows is a list of requirements for CGNs. They are in addition to those found in other documents such as [RFC4787], [RFC5382], and [RFC5508].

REQ-1: If a CGN forwards packets containing a given transport protocol, then it MUST fulfill that transport protocol’s behavioral requirements. Current applicable documents are as follows:

A. "NAT Behavioral Requirements for Unicast UDP" [RFC4787]
B. "NAT Behavioral Requirements for TCP" [RFC5382]
C. "NAT Behavioral Requirements for ICMP" [RFC5508]
D. "NAT Behavioral Requirements for DCCP" [RFC5597]

Any future NAT behavioral requirements documents for IPv4 transport protocols will impose additional requirements for CGNs on top of those stated here.

Justification: It is crucial for CGNs to maximize the set of applications that can function properly across them. The IETF has documented the best current practices for UDP, TCP, ICMP, and DCCP.

REQ-2: A CGN MUST have a default "IP address pooling" behavior of "Paired" (as defined in [RFC4787] section 4.1). A CGN MAY provide a mechanism for administrators to change this behavior on an application protocol basis.

* When multiple overlapping internal IP address ranges share the same external IP address pool (e.g., DS-Lite [RFC6333]), the "IP address pooling" behavior applies to mappings between external IP addresses and internal subscribers rather than between external and internal IP...
addresses.

Justification: This stronger form of REQ-2 from [RFC4787] is justified by the stronger need for not breaking applications that depend on the external address remaining constant.

Note that this requirement applies regardless of the transport protocol. In other words, a CGN must use the same external IP address mapping for all sessions associated with the same internal IP address, be they TCP, UDP, ICMP, something else, or a mix of different protocols.

The justification for allowing other behaviors is to allow the administrator to save external addresses and ports for application protocols that are known to work fine with other behaviors in practice. However, the default behavior MUST be "Paired".

REQ-3: The CGN function SHOULD NOT have any limitations on the size nor the contiguity of the external address pool. In particular, the CGN function MUST be configurable with contiguous or non-contiguous external IPv4 address ranges.

Justification: Given the increasing rarity of IPv4 addresses, it is becoming harder for an operator to provide large contiguous address pools to CGNs. Additionally, operational flexibility may require non-contiguous address pools for reasons such as differentiated services, routing management, etc.

The reason for having SHOULD instead of MUST is to account for limitations imposed by available resources as well as constraints imposed for security reasons.

REQ-4: A CGN MUST support limiting the number of external ports (or, equivalently, "identifiers" for ICMP) that are assigned per subscriber.

A. Per-subscriber limits MUST be configurable by the CGN administrator.

B. Per-subscriber limits MAY be configurable independently per transport protocol.

C. Additionally, it is RECOMMENDED that the CGN include administrator-adjustable thresholds to prevent a single subscriber from consuming excessive CPU resources from the CGN (e.g., rate limit the subscriber’s creation of new mappings).
Justification: A CGN can be considered a network resource that is shared by competing subscribers. Limiting the number of external ports assigned to each subscriber mitigates the DoS attack that a subscriber could launch against other subscribers through the CGN in order to get a larger share of the resource. It ensures fairness among subscribers. Limiting the rate of allocation mitigates a similar attack where the CPU is the resource being targeted instead of port numbers, however this requirement is not a MUST because it is very hard to explicitly call out all CPU-consuming events.

REQ-5: A CGN SHOULD support limiting the amount of state memory allocated per mapping and per subscriber. This may include limiting the number of sessions, the number of filters, etc., depending on the NAT implementation.

A. Limits SHOULD be configurable by the CGN administrator.

B. Additionally, it SHOULD be possible to limit the rate at which memory-consuming state elements are allocated.

Justification: A NAT needs to keep track of TCP sessions associated to each mapping. This state consumes resources for which, in the case of a CGN, subscribers may compete. It is necessary to ensure that each subscriber has access to a fair share of the CGN’s resources. Limiting the rate of allocation is intended to prevent CPU resource exhaustion. Item "B" is at the SHOULD level to account for the fact that means other than rate limiting may be used to attain the same goal.

REQ-6: It MUST be possible to administratively turn off translation for specific destination addresses and/or ports.

Justification: It is common for a CGN administrator to provide access for subscribers to servers installed in the ISP’s network in the external realm. When such a server is able to reach the internal realm via normal routing (which is entirely controlled by the ISP), translation is unneeded. In that case, the CGN may forward packets without modification, thus acting like a plain router. This may represent an important efficiency gain.

Figure 2 illustrates this use-case.
REQ-7: It is RECOMMENDED that a CGN use an "Endpoint-Independent Filtering" behavior (as defined in [RFC4787] section 5). If it is known that "Address-Dependent Filtering" does not cause the application-layer protocol to break (how to determine this is out of scope for this document), then it MAY be used instead.

Justification: This is a stronger form of REQ-8 from [RFC4787]. This is based on the observation that some games and peer-to-peer applications require EIF for the NAT traversal to work. In the context of a CGN it is important to minimize application breakage.

REQ-8: Once an external port is deallocated, it SHOULD NOT be reallocated to a new mapping until at least 120 seconds have passed, with the exceptions being:

A. If the CGN tracks TCP sessions (e.g., with a state machine, as in [RFC6146] section 3.5.2.2), TCP ports MAY be reused immediately.

B. If external ports are statically assigned to internal addresses (e.g., address X with port range 1000-1999 is assigned to subscriber A, 2000-2999 to subscriber B, etc.), and the assignment remains constant across state loss, then ports MAY be reused immediately.

C. If the allocated external ports used address-dependent or address-and-port-dependent filtering before state loss, they MAY be reused immediately.

The length of time and the maximum number of ports in this state MUST be configurable by the CGN administrator.
Justification: This is necessary in order to prevent collisions between old and new mappings and sessions. It ensures that all established sessions are broken instead of redirected to a different peer.

The exceptions are for cases where reusing a port immediately does not create a possibility that packets would be redirected to the wrong peer. One can imagine other exceptions where mapping collisions are avoided, thus justifying the SHOULD level for this requirement.

The 120 seconds value corresponds to the Maximum Segment Lifetime (MSL) from [RFC0793].

Note that this requirement also applies to the case when a CGN loses state (due to a crash, reboot, failover to a cold standby, etc.). In that case, ports that were in use at the time of state loss SHOULD NOT be reallocated until at least 120 seconds have passed.

REQ-9: A CGN MUST implement a protocol giving subscribers explicit control over NAT mappings. That protocol SHOULD be the Port Control Protocol [I-D.ietf-pcp-base].

Justification: Allowing subscribers to manipulate the NAT state table with PCP greatly increases the likelihood that applications will function properly.

A study of PCP-less CGN impacts can be found in [I-D.donley-nat444-impacts]. Another study considering the effects of PCP on a peer-to-peer file sharing protocol can be found in [I-D.boucadair-pcp-bittorrent].

REQ-10: CGN implementers SHOULD make their equipment manageable. Standards-based management using standards such as "Definitions of Managed Objects for NAT" [RFC4008] is RECOMMENDED.

Justification: It is anticipated that CGNs will be primarily deployed in ISP networks where the need for management is critical. This requirement is at the SHOULD level to account for the fact that some CGN operators may not need management functionality.

Note also that there are efforts within the IETF toward creating a MIB tailored for CGNs (e.g., [I-D.ietf-behave-nat-mib]).
REQ-11: When a CGN is unable to create a dynamic mapping due to resource constraints or administrative restrictions (i.e., quotas):

A. it MUST drop the original packet;

B. it SHOULD send an ICMP Destination Unreachable message with code 1 (Host Unreachable) to the sender;

C. it SHOULD send a notification (e.g., SNMP trap) towards a management system (if configured to do so);

D. and it MUST NOT delete existing mappings in order to "make room" for the new one. (This only applies to normal CGN behavior, not to manual operator intervention.)

Justification: This is a slightly different form of REQ-8 from [RFC5508]. Code 1 is preferred to code 13 because it is listed as a "soft error" in [RFC1122], which is important because we don’t want TCP stacks to abort the connection attempt in this case. See [RFC5461] for details on TCP’s reaction to soft errors.

Sending ICMP errors and SNMP traps may be rate-limited for security reasons, which is why requirements B and C are SHOULDs, not a MUSTs.

Applications generally handle connection establishment failure better than established connection failure. This is why dropping the packet initiating the new connection is preferred over deleting existing mappings. See also the rationale in [RFC5508] section 6.

4. Logging

It may be necessary for CGN administrators to be able to identify a subscriber based on external IPv4 address, port, and timestamp in order to deal with abuse. When multiple subscribers share a single external address, the source address and port that are visible at the destination host have been translated from the ones originated by the subscriber.

In order to be able to do this, the CGN would need to log the following information for each mapping created (this list is for informational purposes only and does not constitute a requirement):
o transport protocol
o subscriber identifier (e.g., internal source address or tunnel endpoint identifier)
o external source address
o external source port
o timestamp

By "subscriber identifier" we mean information that uniquely identifies a subscriber. For example, in a traditional NAT scenario, the internal source address would be sufficient. In the case of DS-Lite, many subscribers share the same internal address and the subscriber identifier is the tunnel endpoint identifier (i.e., the B4’s IPv6 address).

A disadvantage of logging mappings is that CGNs under heavy usage may produce large amounts of logs, which may require large storage volume.

REQ-12: A CGN SHOULD NOT log destination addresses or ports unless required to do so for administrative reasons.

Justification: Destination logging at the CGN creates privacy issues. Furthermore, readers should be aware of logging recommendations for Internet-facing servers [RFC6302]. With compliant servers, the destination address and port do not need to be logged by the CGN. This can help reduce the amount of logging.

This requirement is at the SHOULD level to account for the fact that there may be other reasons for logging destination addresses or ports. One such reason might be that the remote server is not following [RFC6302].

5. Port Allocation Scheme

A CGN’s port allocation scheme is subject to three competing requirements:

REQ-13: A CGN’s port allocation scheme SHOULD maximize port utilization.
Justification: External ports is one of the resources being shared by a CGN. Efficient management of that resource directly impacts the quality of a subscriber’s Internet connection.

Some schemes are very efficient in their port utilization. In that sense, they have good scaling properties (nothing is wasted). Others will systematically waste ports.

REQ-14: A CGN’s port allocation scheme SHOULD minimize log volume.

Justification: Huge log volumes can be problematic to CGN operators.

Some schemes create one log entry per mapping. Others allow multiple mappings to generate a single log entry, which sometimes can be expressed very compactly. With some schemes the logging frequency can approach that of DHCP servers.

REQ-15: A CGN’s port allocation scheme SHOULD make it hard for attackers to guess port numbers.

Justification: Easily guessed port numbers put subscribers at risk of the attacks described in [RFC6056].

Some schemes provide very good security in that ports numbers are not easily guessed. Others provide poor security to subscribers.

A CGN implementation’s choice of port allocation scheme optimizes to satisfy one requirement at the expense of another. Therefore, these are soft requirements (SHOULD as opposed to MUST).

6. Deployment Considerations

Several issues are encountered when CGNs are used [RFC6269]. There is current work in the IETF toward alleviating some of these issues. For example, see [I-D.ietf-intarea-nat-reveal-analysis].

7. IANA Considerations

There are no IANA considerations.

8. Security Considerations

If a malicious subscriber can spoof another subscriber’s CPE, it may cause a DoS to that subscriber by creating mappings up to the allowed limit. An ISP can prevent this with ingress filtering, as described
This document recommends Endpoint-Independent Filtering (EIF) as the default filtering behavior for CGNs. EIF has security considerations which are discussed in [RFC4787].

NATs sometimes perform fragment reassembly. CGNs would do so at presumably high data rates. Therefore, the reader should be familiar with the potential security issues described in [RFC4963].

9. Acknowledgements

Thanks for the input and review by Alexey Melnikov, Arifumi Matsumoto, Barry Leiba, Benson Schliesser, Dai Kuwabara, Dan Wing, Dave Thaler, David Harrington, Francis Dupont, Jean-Francois Tremblay, Joe Touch, Lars Eggert, Kousuke Shishikura, Mohamed Boucadair, Martin Stiemerling, Meng Wei, Nejc Skoberne, Pete Resnick, Reinaldo Penno, Ron Bonica, Sam Hartman, Sean Turner, Senthil Sivakumar, Stephen Farrell, Stewart Bryant, Takanori Mizuguchi, Takeshi Tomochika, Tina Tsou, Tomohiro Fujisaki, Tomohiro Nishitani, Tomoya Yoshida, Wes George, Wesley Eddy, and Yasuhiro Shirasaki.

10. References

10.1. Normative References


[RFC5597] Denis-Courmont, R., "Network Address Translation (NAT)

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-26 (work in progress), June 2012.

10.2. Informative Reference


RFC 2663, August 1999.


BCP 162, RFC 6302, June 2011.


Authors’ Addresses

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

Phone: +1 418 656 9254
Email: simon.perreault@viagenie.ca
URI: http://www.viagenie.ca
Problem Statement for the Automated Configuration of Large IP Networks
draft-ietf-opsawg-automated-network-configuration-02

Abstract

This memo discusses the steps required to bring a large number of devices into service in IP networks in an automated fashion. The goal of this document is to list known solutions where they exist and to identify gaps that require further specifications.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 3, 2012.

Copyright Notice
Table of Contents

1. Introduction .................................................. 3
2. Intra-domain and Inter-domain Scenarios ....................... 4
3. Model of the Automated Configuration Process ................ 6
4. Phase 1: Pre-configuration ................................... 7
5. Phase 2: Bootstrapping ........................................ 7
   5.1. Establishment of Link Layer Connectivity ................ 8
   5.2. Acquisition of IP Addresses and Basic Routing Information 8
   5.3. Finding the Configuration Server ....................... 8
   5.4. Establishing a Secure Channel to the Configuration Server 10
6. Phase 3: Initial Configuration .................................. 12
7. Phase 4: Configuration Auditing ................................ 15
8. Phase 5: Configuration Update .................................. 15
9. Missing Specifications ......................................... 16
10. Security Considerations ....................................... 16
11. IANA Considerations .......................................... 17
12. Acknowledgements ............................................ 17
13. Informative References ...................................... 17
Appendix A. Changes since -01 .................................. 22
Authors’ Addresses .............................................. 22
1. Introduction

Many large IP networks are being deployed that entail the installation of tens of thousands of new network devices. To keep costs down, it is desirable to automate the establishment of such networks to the maximum extent possible. This naturally raises the question how new devices can pick up the configuration information they need to operate properly in an automated fashion. The goal of this document is to list known solutions where they exist and to identify gaps that require further specifications.

A certain basic amount of configuration information must be pre-configured by the vendor or network operator before the devices are physically deployed. This pre-provisioned configuration can either be stored directly on the device itself or it can be provided to the device during the deployment operation via pluggable memory cards or near field communication technologies. Further device configuration information is best delivered after startup, to ensure that it is consistent with the physical deployment and the desired network configuration.

One example where automated configuration is important are new service provider networks. 3GPP work in progress describes requirements [TS_32_500] and an architectural specification [TS_36_300] for the self-configuration of edge node entities called eNodeBs. (The expansion of eNodeB is too unwieldy to spell out.) Specifically, procedures are specified for establishing transport connections to and for exchanging configuration data with control entities called MMEs (Mobility Management Entities) and with neighbouring eNodeBs. [TS_36_300] currently assumes as a starting precondition that the eNodeB knows its own IP address and knows IP address endpoints for the target MMEs and neighbouring eNodeBs.

The Broadband Forum has defined a CPE WAN Management Protocol (running over SOAP/HTTP/TLS) to manage customer premise equipment (CPE) terminating broadband access networks (typically DSL access networks) [TR_069]. CPE devices locate and connect to an Auto-Configuration Server (ACS), which provides configuration data and software/firmware images and modules. The ACS also performs status and performance monitoring and diagnostic functions. CPE devices use DHCP to locate an ACS and since both peers, the ACS and CPE, can initiate connections, the protocol can work across network address translators (NATs).

Next to service provider networks, many large enterprise networks face the same challenge to roll out a large number of network devices, which often connect to a 3rd party network provider. The current development of IP-based home automation and utility
monitoring technologies might carry the problem to roll out large numbers of devices that need to automatically configure themselves to private households.

IETF work on automated configuration goes back to BOOTP [RFC0951], followed eight years later by DHCP [RFC1541] and successors. The years since have seen a steady growth in the number of DHCP options. The Simple Network Management Protocol (SNMP) [RFC3410] was designed to convey management information between SNMP entities such as managers and agents. The number of SNMP MIB modules grew steadily, but SNMP has historically seen only limited use for configuration [RFC3535]. For a period, IETF configuration efforts were focussed on the distribution of policy information in the network. [RFC3139] provides a good insight into this period. More recently, the network configuration protocol NETCONF [RFC6241] was devised as an alternative to SNMP, but the development of standard NETCONF configuration data models is just beginning.

Recent IETF work closest in spirit to the 3GPP self-organizing network effort cited above is embodied in CAPWAP [RFC5415]. Like the 3GPP work, CAPWAP focusses on the configuration of edge nodes, in a Wi-Fi rather than cellular network. The CAPWAP work goes beyond that of 3GPP by specifying the process of AC (Access Controller) discovery rather than leaving discovery out of scope. With regard to the configuration process itself, CAPWAP provides for the download of new images to the WTP (Wireless Termination Point). In contrast, [TS_32_500] assumes that this has already been completed for the eNodeB.

2. Intra-domain and Inter-domain Scenarios

There are two different scenarios to consider. In the first scenario, called the Intra-domain Scenario, the new network device N is attached to the network operated by the service provider which is also operating the new device. In the second scenario, called the Inter-domain Scenario, the new device N is attached to a third party network providing connectivity to the network of the service provider operating the new device.
Figure 1 depicts the Intra-domain Scenario. We assume that the new device N attaches to a link connected to router R. Furthermore, we assume that the service provider provides a Domain Name System (DNS) server, a reachable DHCP server, and a Configuration Server (CONF). Overall, this scenario does not differ much from conventional network scenarios.

Figure 2 depicts the Inter-domain Scenario where the new device N attaches to a router R owned by a different service provider X. The service provider X might offer its own DNS service and a reachable DHCP service. We assume that the service provider X has connectivity to the service provider planning to operate the new device.

It should be noted that handing out DHCP options specific to N’s service provider via X’s DHCP service requires some close coordination between the two parties involved. This might be difficult in practice. A more general alternative might be to have X’s service provider establish a tunnel such that the new device logically appears to be part of N’s service provider network.
In both scenarios, the new device N is either directly reachable or it may be behind a middlebox such as a Network Address Translator (NAT) or a firewall. Middleboxes may impose restrictions on which party is able to initiate communication. As detailed in [I-D.kwatsen-reverse-ssh], it is often desirable to allow device-initiated connections.

3. Model of the Automated Configuration Process

We introduce a model of the configuration process in order to identify the parts that have well-known solutions. The remainder may be worth studying to see if the industry can agree on a solution.

Some basic terminology is needed for the discussion. Depending on the implementation, let us agree that "configuration data" consist of software and sets of configured parameters in some combination. This includes firmware, licenses, certificates, and other configuration data. Also, the system that provides the configuration data is called the "configuration server". Finally, the term "joining device" is used to denote a network device that is in the process of being incorporated into the network.

Broadly speaking, the configuration process can be broken into five phases:

1. Pre-configuration: configuration carried out either by the vendor or by the service provider prior to physical installation. One possible example is the pre-configuration of certificates or licenses or specific firmware.

2. Bootstrapping: the portion of the process from the time that physical installation is complete until a secure connection is established between the joining device and the configuration server.

3. Initial configuration: downloading of the configuration data that the joining device needs to carry out its function in the network.

4. Configuration auditing: tracking image versions and configuration parameters for each network device and verifying that the installed configuration data matches the physical installation, the network plan, and the records of what data was downloaded. It is possible that an initial audit of the physical installation is done before initial configuration, so that the validity of the intended download can be verified.
5. Configuration update: transferring configuration data to a fully configured and operating device from time to time as the need arises.

4. Phase 1: Pre-configuration

This memo identifies a specific requirement for pre-configuration of an invariant device identity and authentication-related material in the form of pre-shared secrets or certificates. There is, as one alternative, also a requirement for pre-configuration of information that permits the joining device to discover the address of the configuration server.

Note that pre-configuration may be carried out on the joining device itself or it may be provided to the joining device during the deployment process via pluggable memory cards or nearfield communication.

5. Phase 2: Bootstrapping

[I-D.sarikaya-core-sbootstrapping] deals with the process of security bootstrapping, with particular emphasis on the requirements for highly resource-constrained devices. The document makes a distinction between a data channel, which is used during network operation, and a control channel, which is used during bootstrapping. While both channels can be the same physical channel, they can also be different (e.g., a wireless access point using an infrared control channel to receive bootstrapping information). The draft discusses a number of possible security bootstrapping protocols for resource constrained devices that can be executed in several bootstrapping rounds and can be adapted to the specific contexts in terms of the resources available within individual devices and for the network as a whole.

For network devices in service provider networks or large enterprise networks, bootstrapping consists of several stages:

1. establishment of link layer connectivity with neighbouring nodes;
2. acquisition of IP addresses and basic routing information;
3. discovery of the configuration server;
4. establishment of a secure channel to the configuration server.

Each of these stages is further discussed below.
5.1. Establishment of Link Layer Connectivity

The protocol aspects of this phase are out of scope, since it involves non-IETF protocols only. While some link-layer technologies may provide authentication and access control, this cannot be assumed to be available in the general case.

5.2. Acquisition of IP Addresses and Basic Routing Information

For IPv4, DHCPv4 [RFC2131] is widely deployed and the usual way to obtain an IPv4 address, the IPv4 address of a link-local router and the IPv4 address of a DNS server. For IPv6, a choice has to be made between stateful DHCPv6 [RFC3315] versus stateless DHCPv6 [RFC3736] combined with stateless address autoconfiguration [RFC4862]. In the latter case, DHCPv6 is needed to configure parameters such as DNS server addresses. A routing advertisement option to configure the IPv6 address of a DNS server as part of the stateless address autoconfiguration is defined in [RFC6106].

Some security protection is provided in this stage by using DHCP authentication [RFC3118]. However, security of the configuration process as a whole has to be assured by other means. This is discussed further below.

Currently the lack of a stable identifier for use in DHCPv6 messaging is an impediment to authentication of the joining device. [RFC6355] discusses the problems with the current DHCPv6 identifiers (DUIDs) and proposes a new form that could be a more stable alternative.

A joining device can also choose to use a pre-configured IP address, a pre-configured link-local router address and a pre-configured DNS server address. This pre-configuration may be hard wired into the device or provided by a pluggable memory card or nearfield communication. However, a static pre-configuration hard-wires assumption about the network a devices operates in and is therefore brittle and not recommended.

5.3. Finding the Configuration Server

Four alternatives are available for finding the configuration server:

- pre-configuration;
- DHCP configuration;
- Service Location Protocol [RFC2608]; or
Pre-configuration of an IP address is brittle and not recommended. The pre-configuration of a Uniform Resource Identifier (URI) or fully qualified domain name (FQDN) is a slightly better approach since this allows for a limited dynamic mapping of the name to an IP address. One variant that has been suggested is to burn the URI of a vendor server into the device’s firmware along with a device identifier, and have that server redirect to the URI of the service provider’s configuration server based on the device identity. Such an approach requires that the device vendor’s redirection server is always reachable, that the device vendor offers such a redirection service for the lifetime of their devices and that service providers are able to update the URI of the service provider’s redirection server. Furthermore, this approach can lead to problems if certificates are used to authenticate the involved parties if a service provider tries to prevent the usage of a vendor’s redirection service. Finally, this approach also requires a trust relationship between the vendor and the service provider and agreement on a protocol to update the redirect information on the vendor’s server. As a consequence of these considerations, using this approach is not recommended.

DHCP configuration can use the usual DHCP options and is technically straightforward since DHCP is widely used by end user devices to obtain basic configuration information. There is, however, no standardized DHCP option to communicate the address of a configuration server.

The Service Location Protocol (SLP) has seen some usage to locate services such as printers or file system shares. Usage of SLP to locate configuration servers requires to define a new service template [RFC2609].

The use of DNS SRV records requires the joining device to obtain the correct domain suffix first, presumably from DHCP or via Routing Advertisements in the case of IPv6 or pre-configuration. A service type for the desired configuration protocol would have to be defined in the DNS for the purpose. See Section 3.3 of [RFC5415] for a discussion of the corresponding discovery process for CAPWAP.

The Inter-domain Scenario requires that the DHCP server or the SLP server of service provider X’s network is able to provide the correct information to the joining devices. To accomplish this, the discovery servers need to be able to match a device identification against a list of possible configuration servers. Furthermore, there needs to be a mechanism for the service provider operating the joining device to provision the configuration server’s address, e.g., by using an extension of the Extensible Provisioning Protocol (EPP).
Internet-Draft   Automated Configuration of IP Networks     October 2011

However, if the joining device has pre-configured information about the name of the service provider’s network, DNS SRV records may be queried after obtaining IP connectivity, avoiding the need to provision information in service provider X’s network.

5.4. Establishing a Secure Channel to the Configuration Server

It is essential that the configuration server and the joining device authenticate themselves to each other, since the steps leading up to this point in the process may not be fully secure. This raises two issues: how the joining device identifies itself, and how authentication takes place.

It seems best if the device has an invariant identity built in and accessible to whatever operating system is running on it. [RFC6355] provides such an identity in the form of a Universally Unique IDentifier (UUID). The vendor should make that identity available in a form that can be read and transferred into a database accessible to the configuration server along with the associated configuration data in advance of the bootstrapping stage (e.g., in bar-coded format on the device packaging).

Serial numbers may be used for identification purposes if UUIDs are not available. However, serial numbers often encode information such as model-numbers or manufacturing dates. Hence, it is not recommended to pass serial-numbers in the clear for security reasons. Similar precautions apply to Common Language Equipment Identifier (CLEI) codes that encode information about properties of the device.

This leaves the mutual authentication process itself. This has two aspects: the security protocol used to perform authentication, and initial keying methodology. The security protocol is tied together with the choice of configuration data transport, but the basic choices are:

- IP Security (IPsec) [RFC4301];
- Transport Layer Security (TLS) [RFC5246];
- Datagram Transport Layer Security (DTLS) [RFC4347];
- Secure Shell (SSH) [RFC4251], [RFC4252], [RFC4253], and [RFC4254]; and
- SNMPv3’s User-based Security Model (USM) [RFC3414].

For initial keying methodology, the two basic choices are between pre-shared secrets and certificates. All of the security protocols...
listed above except USM support both methods. USM supports pre-shared secrets only.

The usual concern with pre-shared secrets is scalability. In the bootstrapping case, the scale of operation required is linear with the number of devices to be configured, so it would definitely be a feasible approach if connection to the configuration system were the only consideration. The most likely procedure would be for the secret to be configured in the device during pre-configuration and also captured in a database along with the device identity, for use by the configuration server.

The problem with the use of pre-shared secrets is that the device needs to authenticate itself at an earlier stage, while it is establishing communications with its neighbours and acquiring IP addresses. It seems undesirable to use the same secret that is used to authenticate the device to the configuration server for that purpose as well, on the basic principle of limiting the potential damage from disclosure of a particular key.

This need for additional pre-shared secrets argues for consideration of certificates as an alternative. One issue for certificates is where the trust anchor resides. It seems logical that it should reside with the service provider rather than the vendor, to make it easy to install equipment from multiple vendors. On that basis, pre-configuration requires service provider input. On the other hand, if devices are drop-shipped to the destination from the vendor, having the trust anchor reside with the vendor might be acceptable as well.

CAPWAP (Section 2.4.4.3 of [RFC5415]) makes use of the Extended Key Usage (EKU) certificate extension [RFC5280] to distinguish certificates identifying the Access Controllers (i.e., the configuration servers in the CAPWAP case) from the Wireless Transfer Points (the configured devices in the CAPWAP case). Thought should be given to whether such distinctions are required in the general case of network device configuration.

CAPWAP (Section 12.8 of [RFC5415]) also discusses the use of the Common Name rather than SubjectAltName field of the certificate to carry device identity, due to lack of a Uniform Resource Name (URN) specification allowing the use of SubjectAltName to carry MAC addresses. This encoding of device identifiers in certifications needs to be investigated further if a new form of device unique identity is used, as discussed above.

Middleboxes such as NATs or firewalls may impose restriction on which party is able to initiate communication. In the common case of NATs in IPv4 access networks, communication can only be established from
the device to the configuration server. Not all secure transports, in particular those where authentication is not symmetric, support this "call home" mode of operation. A recent proposal to reverse the establishment of the TCP connection for SSH can be found in [I-D.kwatsen-reverse-ssh].

6. Phase 3: Initial Configuration

As mentioned at the beginning, the configuration data being downloaded may be a combination of software/firmware and configuration parameters. Some of the data will be vendor-specific and not subject to standardization. It appears that there is a continuing debate on whether the configuration data should be pushed to the joining device or whether the device should pull the configuration data from the configuration server. In the latter case, the device needs to know about the existence of the data and the path to reach it before it can act. One way to acquire this information is through DHCP. DHCPv4 has provided the necessary options from its beginnings, inheriting them from BOOTP. They have been recently added to DHCPv6 [RFC5970].

Protocols that can transport configuration data can be classified as follows: The first class consists of generic file transfer protocols that can carry configuration data serialized into configuration files. The second class consists of protocols that manipulate structured configuration data directly. The structure of the configuration data is defined by some data model.

In the first class, we find the following file transfer protocols:

- The File Transfer Protocol (FTP) [RFC0959] can be used to move files containing configuration data. It can be secured by running FTP over TLS [RFC4217].

- The Trivial File Transfer Protocol (TFTP) [RFC1350] has been used extensively to load boot images over the network. However, it does not provide security and the only option is to rely on IP layer security (IPsec).

- The Hypertext Transfer Protocol (HTTP) [RFC2616] can be used to transfer documents containing configuration data. It is commonly secured by running HTTP over TLS [RFC2817] [RFC2818].

- The SSH File Transfer Protocol (SFTP) [I-D.ietf-secsh-filexfer] provides roughly the same services as FTP but runs over SSH and thus utilizes the security services provided by SSH.
o UNIX utilities to transfer files such as RCP and SCP provide limited flexibility and they differ in their degree of integration with SSH.

o The Control And Provisioning of Wireless Access Points (CAPWAP) protocol [RFC5415] can be used to control the download of images. CAPWAP can be secured by running CAPWAP over DTLS.

In the second class, we find the following configuration protocols:

o Version 3 of the Simple Network Management Protocol (SNMPv3) [RFC3411]-[RFC3418] can be used to manipulate MIB objects and to carry event notifications. It has its own security protocol (USM) but can also run over SSH [RFC5592], TLS, or DTLS [RFC6353].

o The Common Open Policy Service for Policy Provisioning protocol (COPS-PR) [RFC3084] was designed to provision structured policy information from a Policy Decision Point (PDP) to a Policy Enforcement Point (PEP). The COPS protocol [RFC2748] provides an integrity object that can achieve authentication, message integrity, and replay prevention. Optionally, COPS and COPS-PR can run over TLS.

o The NETCONF protocol [RFC6241] provides mechanisms to install, manipulate, and delete the configuration of network devices. A protocol extension provides an asynchronous event notification delivery mechanism [RFC5277]. NETCONF by default runs over SSH but can also run over transports secured by TLS.

o The Control And Provisioning of Wireless Access Points protocol (CAPWAP) [RFC5415] supports the discovery of so called Access Controller (AC) by Wireless Termination Points (WTPs) and the configuration of WTPs by an AC. While CAPWAP can be extended to configure other devices, its main focus are WTPs. The CAPWAP protocol is protected by using DTLS after the discovery phase.

Table 1 lists the protocols plus their basic properties while Table 2 lists the security options available for each protocol.
Table 1: Protocols for transporting configuration data

<table>
<thead>
<tr>
<th>Transport</th>
<th>Data Transfer Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>TFTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>HTTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>SFTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>RCP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>SCP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>CAPWAP</td>
<td>AC pushes configuration parameters, WTP pulls software</td>
</tr>
<tr>
<td>SNMPv3</td>
<td>Push of structured configuration parameters, event notifications</td>
</tr>
<tr>
<td>COPS-PR</td>
<td>Push of structured policy information</td>
</tr>
<tr>
<td>NETCONF</td>
<td>Push of structured configuration data, event notifications</td>
</tr>
</tbody>
</table>

Table 2: Security options for configuration transport protocols

SNMPv3, NETCONF, and COPS-PR carry structured data specified in predefined data models. SNMPv3 and COPS-PR have size limitations on the data objects and thus make the transport of larger software images difficult. NETCONF does not suffer from hard size restrictions and can in principle carry software images inline. However, there is currently no work in progress to standardize the transfer of software images over NETCONF. CAPWAP combines the functions of configuration parameter transport and software download. The parameter transport aspect lacks the generality offered by SNMP, NETCONF, and COPS-PR, since the parameters are specified within the protocol specification itself. The remaining transports are independent of the nature of...
the information being transferred.

7. Phase 4: Configuration Auditing

To complete the process, it must be possible to audit the configuration status of the device in some detail. This is likely to begin even before all the configuration data has been downloaded. For instance, configuration management may wish to collect basic information such as the MAC addresses of the device’s interfaces, the link-local addresses assigned to them, and similar information for the neighbours of the joining device.

SNMP and SNMP MIB modules are obviously one way to collect this information. NETCONF [RFC6241] is an alternative, but the necessary data models have to be defined. YANG modules for NETCONF [RFC6020] can be generated from existing SNMP MIB modules by translating the SNMP modules into YANG modules [I-D.ietf-netmod-smi-yang].

Another important auditing activity is the analysis of system events. The SYSLOG protocol [RFC5424] is widely used for this purpose but SNMPv3 and NETCONF can ship event notifications as well. Translations of SNMP notifications into structured SYSLOG messages and vice versa do exist [RFC5675] [RFC5676]. NETCONF can carry SYSLOG content as well [RFC5277].

NETCONF provides generic notifications that help with tracking configuration changes [I-D.ietf-netconf-system-notifications]. Similar standardized configuration change notifications do not exist for SNMP or SYSLOG.

8. Phase 5: Configuration Update

Configuration updates can in principle be handled with the same protocol that delivered the initial configuration. However, in some deployments, the mechanism used for initial configuration might be different.

An advantage of NETCONF over SNMPv3 and CAPWAP in the context of configuration updates is the support of concurrent updates through explicit locking mechanisms and the support of network wide configuration change transactions through the confirmed commit capability.
9. Missing Specifications

This document discussed the automated configuration of devices in service provider networks. Several gaps were identified requiring further specification:

G1: Definition of a DHCP option to provide the IPv4/IPv6 address of a configuration server. Such an option allows a joining device to pickup the configuration server’s address as part of the DHCP exchange. This is particularly interesting for Intra-domain Scenarios.

G2: Definition of DNS SRV records for locating configuration servers. Such an option allows a joining device to lookup the configuration server’s in the DNS; this is particularly useful in an Inter-domain Scenario.

G3: Definition of a SLP template for discovering configuration servers. Such a template is useful only in environments where SLP is used also for other purposes.

G4: Definition of NETCONF data models to support the download / update of software images through NETCONF.

G5: Definition of NETCONF data models for collecting basic system information and integrity information (e.g., checksums of software images).

G6: Some management protocols lack a mechanisms for devices to initiate a secure communication channel with a management system ("call home").

10. Security Considerations

The security of a configuration management solution is of crucial importance. Section 6 discusses the security options of several protocols that might be used. The relevant protocol definitions should be consulted to learn more about the specific security aspects of the various protocols.

It should be noted that some steps in the described process, in particular the bootstrapping phase, may not be secure and it is thus important to verify the identity of the device and the identity of the configuration server when a secure connection to a configuration server is established. Usage of IPsec, which focuses on securing the IP layer, may not be sufficient for this.
During the choice of protocols, the available security mechanisms and the required key management infrastructures may play a major role in the selection of protocols. Easy integration into existing Authentication, Authorization and Accounting (AAA) infrastructures can significantly reduce the operational costs associated with the security management of the configuration system.

While [I-D.sarikaya-core-sbootstrapping] discusses security bootstrapping mechanisms in the context of constrained devices, many of the mechanisms are also applicable for bootstrapping security in normal devices.

Finally, [RFC6092] discusses security capabilities for customer premises equipment providing residential IPv6 Internet service.

11. IANA Considerations

This memo includes no request to IANA.

12. Acknowledgements

Thanks to Mehmet Ersue, Wesley George, Yiu Lee, Kent Watsen, and Cathy Zhou for their help in preparing this memo.

13. Informative References

[I-D.ietf-netconf-system-notifications]
Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications",
draft-ietf-netconf-system-notifications-06 (work in progress), October 2011.

[I-D.ietf-netmod-smi-yang]
Schoenwaelder, J., "Translation of SMIv2 MIB Modules to YANG Modules",
draft-ietf-netmod-smi-yang-01 (work in progress), July 2011.

[I-D.ietf-secsh-filexfer]
Galbraith, J. and O. Saarenmaa, "SSH File Transfer Protocol",
draft-ietf-secsh-filexfer-13 (work in progress), July 2006.

[I-D.kwatsen-reverse-ssh]
Watsen, K., "Reverse Secure Shell (Reverse SSH)",
draft-kwatsen-reverse-ssh-01 (work in progress),

June 2011.

[I-D.sarikaya-core-sbootstrapping]
Sarikaya, B., Ohba, Y., Moskowitz, R., Cao, Z., and R.


[RFC3084] Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie,


[RFC6020]  Bjorklund, M., "YANG - A Data Modeling Language for the
Network Configuration Protocol (NETCONF)", RFC 6020,
October 2010.

[RFC6092]  Woodyatt, J., "Recommended Simple Security Capabilities in
Customer Premises Equipment (CPE) for Providing
Residential IPv6 Internet Service", RFC 6092,
January 2011.

[RFC6106]  Jeong, J., Park, S., Beloeil, L., and S. Madanapalli,
"IPv6 Router Advertisement Options for DNS Configuration",
RFC 6106, November 2010.

[RFC6241]  Enns, R., Bjorklund, M., Schoenwaelder, J., and A.
Bierman, "Network Configuration Protocol (NETCONF)",
RFC 6241, June 2011.

Model for the Simple Network Management Protocol (SNMP)",
RFC 6353, July 2011.

[RFC6355]  Narten, T. and J. Johnson, "Definition of the UUID-Based
DHCPv6 Unique Identifier (DUID-UUID)", RFC 6355,
August 2011.

"CPE WAN Management Protocol", Broadband Forum TR-069,
November 2010.

[TS_32_500]  3rd Generation Partnership Project, "3rd Generation
Partnership Project; Technical Specification Group
Services and System Aspects; Telecommunication Management;
Self-Organizing Networks (SON); Concepts and requirements

[TS_36_300]  3rd Generation Partnership Project, "3rd Generation
Partnership Project; Technical Specification Group Radio
Access Network; Evolved Universal Terrestrial Radio Access
(E-UTRA) and Evolved Universal Terrestrial Radio Access
Network (E-UTRAN); Overall description; Stage 2 (Release
Appendix A. Changes since -01

Incorporated feedback from Kent Watsen and Wesley George.

Editorial improvements, updated references, etc.

Authors’ Addresses

Tina Tsou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: tena@huawei.com

Juergen Schoenwaelder (editor)
Jacobs University Bremen
Campus Ring 1
Bremen 28759
Germany

Email: j.schoenwaelder@jacobs-university.de

Yang Shi
Hangzhou H3C Tech. Co., Ltd.
Beijing R&D Center of H3C, Digital Technology Plaza,
No. 9 Shangdi 9th Street, Haidian District
Beijing 100085
P.R. China

Phone: +86 010 82775276
Email: young@h3c.com

Tom Taylor (editor)
Huawei Technologies
1852 Lorraine Ave.
Ottawa K1H 6Z8
Canada

Email: tom111.taylor@bell.net
Guoliang Yang
China Telecom
No. 109 Zhongshan Ave. (West), Tianhe District
Guangzhou
P.R. China

Phone: +86 020 38639615
Email: yanggl@gsta.com
Survey of Possibilities for the Automated Configuration of Large IP Networks
draft-ietf-opsawg-automated-network-configuration-05

Abstract

This memo discusses the steps required to bring a large number of devices into service in IP networks in an automated fashion. The goal of this document is to list known solutions where they exist, to point out approaches proven to be problematic, and to identify gaps that require further specifications.

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 27, 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
Table of Contents

1. Introduction ...................................................... 3
2. Intra-domain and Inter-domain Scenarios ......................... 5
4. Phase 1: Pre-configuration ........................................ 7
5. Phase 2: Bootstrapping ............................................. 8
   5.1. Establishment of Link Layer Connectivity .................... 8
   5.2. Acquisition of IP Addresses and Basic Routing Information ............................................. 8
   5.3. Finding the Configuration Server ............................. 9
   5.4. Establishing a Secure Channel to the Configuration Server ............................................. 10
6. Phase 3: Initial Configuration ..................................... 12
7. Phase 4: Configuration Auditing .................................... 15
8. Phase 5: Configuration Update ...................................... 16
9. Gap Analysis ........................................................ 16
10. Security Considerations ............................................ 17
11. IANA Considerations ................................................. 17
12. Acknowledgements .................................................. 18
13. Informative References ............................................ 18
14. Authors’ Addresses .................................................. 22
1. Introduction

Many large IP networks are being deployed that entail the installation of tens of thousands of new network devices. To keep costs down, it is desirable to automate the establishment of such networks to the maximum extent possible. This naturally raises the question how new devices can pick up the configuration information they need to operate properly in an automated fashion. The goal of this document is to list known solutions where they exist, to point out approaches proven to be problematic, and to identify gaps that require further specifications.

The document primarily targets (a) network operators (in the generic sense) who are facing the challenge to roll out a large number of new devices and think about how to implement things properly, (b) network equipment vendors who like to add features to their products that make the roll out of lots of new devices simpler for their customers, and (c) people active in the IETF by identifying gaps where further standards may be useful to develop. The aim of the document is to provide guidance to actors who have not already experienced success in this area by informing about the trade-offs of different approaches.

A certain basic amount of configuration information must be pre-configured by the vendor or network operator before the devices are physically deployed. This pre-provisioned configuration can either be stored directly on the device itself or it can be provided to the device during the deployment operation via pluggable memory cards or near field communication technologies. Further device configuration information is best delivered after startup, to ensure that it is consistent with the physical deployment and the desired network configuration.

One example where automated configuration is important are new service provider networks. 3GPP work in progress describes requirements [TS_32_500] and an architectural specification [TS_36_300] for the self-configuration of edge node entities called eNodeBs. (The expansion of eNodeB is too unwieldy to spell out.) Specifically, procedures are specified for establishing transport connections to and for exchanging configuration data with control entities called MMEs (Mobility Management Entities) and with neighbouring eNodeBs. [TS_36_300] currently assumes as a starting precondition that the eNodeB knows its own IP address and knows IP address endpoints for the target MMEs and neighbouring eNodeBs.

The Broadband Forum has defined a CPE WAN Management Protocol (running over SOAP/HTTP/TLS) to manage customer premise equipment (CPE) terminating broadband access networks (typically DSL access
networks) [TR_069]. CPE devices locate and connect to an Auto-
Configuration Server (ACS), which provides configuration data and
software/firmware images and modules. The ACS also performs status
and performance monitoring and diagnostic functions. CPE devices use
DHCP to locate an ACS and since both peers, the ACS and CPE, can
initiate connections, the protocol can work across network address
translators (NATs). The DHCP exchange uses vendor-specific options
defined by the Broadband Forum (number 3561 in the IANA Enterprise
Numbers registry).

Next to service provider networks, many large enterprise networks
face the same challenge to roll out a large number of network
deVICES, which often connect to a 3rd party network provider. The
current development of IP-based home automation and utility
monitoring technologies might carry the problem to roll out large
numbers of devices that need to automatically configure themselves to
private households.

IETF work on automated configuration goes back to BOOTP [RFC0951],
followed eight years later by DHCP ([RFC1541] and successors). The
years since have seen a steady growth in the number of DHCP options.
The Simple Network Management Protocol (SNMP) [RFC3410] was designed
to convey management information between SNMP entities such as
managers and agents. The number of SNMP MIB modules grew steadily,
but SNMP has historically seen only limited use for configuration
[RFC3535]. For a period, IETF configuration efforts were focussed on
the distribution of policy information in the network. [RFC3139]
provides a good insight into this period. More recently, the network
configuration protocol NETCONF [RFC6241] was devised as an
alternative to SNMP, but the development of standard NETCONF
configuration data models is just beginning.

Recent IETF work closest in spirit to the 3GPP self-organizing
network effort cited above is embodied in CAPWAP [RFC5415]. Like the
3GPP work, CAPWAP focusses on the configuration of edge nodes, in a
Wi-Fi rather than cellular network. The CAPWAP work goes beyond that
of 3GPP by specifying the process of Access Controller (AC) discovery
rather than leaving discovery out of scope. A CAPWAP Wireless
Termination Point (WTP) may use broadcasts and multicasts to discover
local ACs, it may use CAPWAP DHCP options [RFC5417] to obtain IP
addresses of ACs, or it may utilize CAPWAP DNS SRV records if a
domain name is known. With regard to the configuration process
itself, CAPWAP provides for the download of new images to the WTP
(Wireless Termination Point). In contrast, [TS_32_500] assumes that
this has already been completed for the eNodeB.

As can seen, standards for the automated configuration of devices in
IP networks have so far been primarily developed for specific network
access technologies (3GPP, Broadband, 802.11 WLANs) and the various solutions make different assumptions about the services that are available and they are designed to support a configuration protocol that is specific to a certain access technology. The aim of this document is to analyse the various phases of an automated configuration process and to identify gaps that are currently not covered in standard and general purpose configuration management protocols of the IETF.

2. Intra-domain and Inter-domain Scenarios

There are two different scenarios to consider. In the first scenario, called the Intra-domain Scenario, the new network device N is attached to the network operated by the service provider which is also operating the new device. In the second scenario, called the Inter-domain Scenario, the new device N is attached to a third party network providing connectivity to the network of the service provider operating the new device.

```
+------+
| CONF |
+--+---+
+---+     +---+          |
| N +-...-+ R +------------...+
+-----+ +------+
| DNS | | DHCP |
+-----+ +------+
|-- N's Service Provider --|
```

Figure 1: Intra-domain Scenario

Figure 1 depicts the Intra-domain Scenario. We assume that the new device N attaches to a link connected to router R. Furthermore, we assume that the service provider provides a Domain Name System (DNS) server, a reachable DHCP server, and a Configuration Server (CONF). Overall, this scenario does not differ much from conventional network scenarios.
Figure 2: Inter-domain Scenario

Figure 2 depicts the Inter-domain Scenario where the new device N attaches to a router R owned by a different service provider X. The service provider X might offer its own DNS service and a reachable DHCP service. We assume that the service provider X has connectivity to the service provider planning to operate the new device.

It should be noted that handing out DHCP options specific to N’s service provider via X’s DHCP service requires some close coordination between the two parties involved. This might be difficult in practice. A more general alternative might be to have X’s service provider establish a tunnel such that the new device logically appears to be part of N’s service provider network.

In both scenarios, the new device N is either directly reachable or it may be behind a middlebox such as a Network Address Translator (NAT) or a firewall. Middleboxes may impose restrictions on which party is able to initiate communication. As detailed in [I-D.kwatsen-reverse-ssh], it is often desirable to allow device-initiated connections.

3. Model of the Automated Configuration Process

We introduce a model of the configuration process in order to identify the parts that have well-known solutions. The remainder may be worth studying to see if the industry can agree on a solution.

Some basic terminology is needed for the discussion. Depending on the implementation, let us agree that "configuration data" consist of software and sets of configured parameters in some combination. This includes firmware, licenses, certificates, and other configuration data. Also, the system that provides the configuration data is called the "configuration server". Finally, the term "joining
device" is used to denote a network device that is in the process of being incorporated into the network.

Broadly speaking, the configuration process can be broken into five phases:

1. Pre-configuration: configuration carried out either by the vendor or by the service provider prior to physical installation. One possible example is the pre-configuration of certificates or licenses or specific firmware.

2. Bootstrapping: the portion of the process from the time that physical installation is complete until a secure connection is established between the joining device and the configuration server.

3. Initial configuration: downloading of the configuration data that the joining device needs to carry out its function in the network.

4. Configuration auditing: tracking image versions and configuration parameters for each network device and verifying that the installed configuration data matches the physical installation, the network plan, and the records of what data was downloaded. It is possible that an initial audit of the physical installation is done before initial configuration, so that the validity of the intended download can be verified.

5. Configuration update: transferring configuration data to a fully configured and operating device from time to time as the need arises.

4. Phase 1: Pre-configuration

This memo identifies a specific requirement for pre-configuration of an invariant device identity and authentication-related material in the form of pre-shared secrets or certificates. There is, as one alternative, also a requirement for pre-configuration of information that permits the joining device to discover the address of the configuration server.

Note that pre-configuration may be carried out on the joining device itself or it may be provided to the joining device during the deployment process via pluggable memory cards or nearfield communication.
5. Phase 2: Bootstrapping

[I-D.sarikaya-core-sbootstrapping] deals with the process of security bootstrapping, with particular emphasis on the requirements for highly resource-constrained devices. The document makes a distinction between a data channel, which is used during network operation, and a control channel, which is used during bootstrapping. While both channels can be the same physical channel, they can also be different (e.g., a wireless access point using an infrared control channel to receive bootstrapping information). The draft discusses a number of possible security bootstrapping protocols for resource constrained devices that can be executed in several bootstrapping rounds and can be adapted to the specific contexts in terms of the resources available within individual devices and for the network as a whole.

For network devices in service provider networks or large enterprise networks, bootstrapping consists of several stages:

1. establishment of link layer connectivity with neighbouring nodes;
2. acquisition of IP addresses and basic routing information;
3. discovery of the configuration server;
4. establishment of a secure channel to the configuration server.

Each of these stages is further discussed below.

5.1. Establishment of Link Layer Connectivity

The protocol aspects of this phase are out of scope, since it involves non-IETF protocols only. While some link-layer technologies may provide authentication and access control, this cannot be assumed to be available in the general case.

5.2. Acquisition of IP Addresses and Basic Routing Information

For IPv4, DHCPv4 [RFC2131] is widely deployed and the usual way to obtain an IPv4 address, the IPv4 address of a link- local router and the IPv4 address of a DNS server. For IPv6, a choice has to be made between stateful DHCPv6 [RFC3315] versus stateless DHCPv6 [RFC3736] combined with stateless address autoconfiguration [RFC4862]. In the latter case, DHCPv6 is needed to configure parameters such as DNS server addresses. A routing advertisement option to configure the IPv6 address of a DNS server as part of the stateless address autoconfiguration is defined in [RFC6106].
Some security protection is provided in this stage by using DHCP authentication [RFC3118]. However, security of the configuration process as a whole has to be assured by other means. This is discussed further below.

Currently the lack of a stable identifier for use in DHCPv6 messaging is an impediment to authentication of the joining device. [RFC6355] discusses the problems with the current DHCPv6 identifiers (DUIDs) and proposes a new form that could be a more stable alternative.

A joining device can also choose to use a pre-configured IP address, a pre-configured link-local router address and a pre-configured DNS server address. This pre-configuration may be hard wired into the device or provided by a pluggable memory card or nearfield communication. However, a static pre-configuration hard-wires assumption about the network a device operates in and is therefore brittle and not recommended.

5.3. Finding the Configuration Server

Four alternatives are available for finding the configuration server:

  o pre-configuration;
  o DHCP configuration;
  o Service Location Protocol [RFC2608]; or
  o DNS service discovery using DNS SRV records [RFC2782].

Pre-configuration of an IP address is brittle and not recommended unless the IP address is used as an anycast address. In the case of an IP anycast address, the routing system will select one out of an anycast cluster of configuration servers the devices connects to. For this to work well, all configuration servers in the anycast cluster should provide the same configuration data.

The pre-configuration of a Uniform Resource Identifier (URI) or fully qualified domain name (FQDN) is a slightly better approach than pre-configuring non-anycast IP addresses since this allows for a limited dynamic mapping of the name to an IP address. One variant that has been suggested is to burn the URI of a vendor server into the device’s firmware along with a device identifier, and have that server redirect to the URI of the service provider’s configuration server based on the device identity. Such an approach requires that the device vendor’s redirection server is always reachable, that the device vendor offers such a redirection service for the lifetime of their devices and that service providers are able to update the URI.
of the service provider’s redirection server. Furthermore, this approach can lead to problems if certificates are used to authenticate the involved parties if a service provider tries to prevent the usage of a vendor’s redirection service. Finally, this approach also requires a trust relationship between the vendor and the service provider and agreement on a protocol to update the redirect information on the vendor’s server. As a consequence of these considerations, using this approach is not recommended.

DHCP configuration can use the usual DHCP options and is technically straightforward since DHCP is widely used by end user devices to obtain basic configuration information. There is, however, no standardized DHCP option to communicate the address of a configuration server.

The Service Location Protocol (SLP) has seen some usage to locate services such as printers or file system shares. Usage of SLP to locate configuration servers requires to define a new service template [RFC2609].

The use of DNS SRV records requires the joining device to obtain the correct domain suffix first, presumably from DHCP or via Routing Advertisements in the case of IPv6 or pre-configuration. A service type for the desired configuration protocol would have to be defined in the DNS for the purpose. See Section 3.3 of [RFC5415] for a discussion of the corresponding discovery process for CAPWAP.

The Inter-domain Scenario requires that the DHCP server or the SLP server of service provider X’s network is able to provide the correct information to the joining devices. To accomplish this, the discovery servers need to be able to match a device identification against a list of possible configuration servers. Furthermore, there needs to be a mechanism for the service provider operating the joining device to provision the configuration server’s address, e.g., by using an extension of the Extensible Provisioning Protocol (EPP) [RFC5730]. However, if the joining device has pre-configured information about the name of the service provider’s network, DNS SRV records may be queried after obtaining IP connectivity, avoiding the need to provision information in service provider X’s network.

5.4. Establishing a Secure Channel to the Configuration Server

It is essential that the configuration server and the joining device authenticate themselves to each other, since the steps leading up to this point in the process may not be fully secure. This raises two issues: how the joining device identifies itself, and how authentication takes place.
It seems best if the device has an invariant identity built in and accessible to whatever operating system is running on it. [RFC6355] provides such an identity in the form of a Universally Unique IDentifier (UUID). The vendor should make that identity available in a form that can be read and transferred into a database accessible to the configuration server along with the associated configuration data in advance of the bootstrapping stage (e.g., in bar-coded format on the device packaging).

Serial numbers may be used for identification purposes if UUIDs are not available. However, serial numbers often encode information such as model-numbers or manufacturing dates. Hence, it is not recommended to pass serial-numbers in the clear for security reasons. Similar precautions apply to Common Language Equipment Identifier (CLEI) codes that encode information about properties of the device.

This leaves the mutual authentication process itself. This has two aspects: the security protocol used to perform authentication, and initial keying methodology. The security protocol is tied together with the choice of configuration data transport, but the basic choices are:

- IP Security (IPsec) [RFC4301];
- Transport Layer Security (TLS) [RFC5246];
- Datagram Transport Layer Security (DTLS) [RFC6347];
- Secure Shell (SSH) [RFC4251], [RFC4252], [RFC4253], and [RFC4254]; and
- SNMPv3’s User-based Security Model (USM) [RFC3414].

For initial keying methodology, the two basic choices are between pre-shared secrets and certificates. All of the security protocols listed above except USM support both methods. USM supports pre-shared secrets only.

The usual concern with pre-shared secrets is scalability. In the bootstrapping case, the scale of operation required is linear with the number of devices to be configured, so it would definitely be a feasible approach if connection to the configuration system were the only consideration. The most likely procedure would be for the secret to be configured in the device during pre-configuration and also captured in a database along with the device identity, for use by the configuration server.

The problem with the use of pre-shared secrets is that the device
needs to authenticate itself at an earlier stage, while it is establishing communications with its neighbours and acquiring IP addresses. It seems undesirable to use the same secret that is used to authenticate the device to the configuration server for that purpose as well, on the basic principle of limiting the potential damage from disclosure of a particular key.

This need for additional pre-shared secrets argues for consideration of certificates as an alternative. One issue for certificates is where the trust anchor resides. It seems logical that it should reside with the service provider rather than the vendor, to make it easy to install equipment from multiple vendors. On that basis, pre-configuration requires service provider input. On the other hand, if devices are drop-shipped to the destination from the vendor, having the trust anchor reside with the vendor might be acceptable as well.

CAPWAP (Section 2.4.4.3 of [RFC5415]) makes use of the Extended Key Usage (EKU) certificate extension [RFC5280] to distinguish certificates identifying the Access Controllers (i.e., the configuration servers in the CAPWAP case) from the Wireless Transfer Points (the configured devices in the CAPWAP case). Thought should be given to whether such distinctions are required in the general case of network device configuration.

CAPWAP (Section 12.8 of [RFC5415]) also discusses the use of the Common Name rather than SubjectAltName field of the certificate to carry device identity, due to lack of a Uniform Resource Name (URN) specification allowing the use of SubjectAltName to carry MAC addresses. This encoding of device identifiers in certifications needs to be investigated further if a new form of device unique identity is used, as discussed above.

Middleboxes such as NATs or firewalls may impose restriction on which party is able to initiate communication. In the common case of NATs in IPv4 access networks, communication can only be established from the device to the configuration server. Not all secure transports, in particular those where authentication is not symmetric, support this "call home" mode of operation. A recent proposal to reverse the establishment of the TCP connection for SSH can be found in [I-D.kwatsen-reverse-ssh].

6. Phase 3: Initial Configuration

As mentioned at the beginning, the configuration data being downloaded may be a combination of software/firmware and configuration parameters. Some of the data will be vendor-specific and not subject to standardization. It appears that there is a
continuing debate on whether the configuration data should be pushed to the joining device or whether the device should pull the configuration data from the configuration server. In the latter case, the device needs to know about the existence of the data and the path to reach it before it can act. One way to acquire this information is through DHCP. DHCPv4 has provided the necessary options from its beginnings, inheriting them from BOOTP. They have been recently added to DHCPv6 [RFC5970].

Protocols that can transport configuration data can be classified as follows: The first class consists of generic file transfer protocols that can carry configuration data serialized into configuration files. The second class consists of protocols that manipulate structured configuration data directly. The structure of the configuration data is defined by some data model.

In the first class, we find the following file transfer protocols:

- The File Transfer Protocol (FTP) [RFC0959] can be used to move files containing configuration data. It can be secured by running FTP over TLS [RFC4217].
- The Trivial File Transfer Protocol (TFTP) [RFC1350] has been used extensively to load boot images over the network. However, it does not provide security and the only option is to rely on IP layer security (IPsec).
- The Hypertext Transfer Protocol (HTTP) [RFC2616] can be used to transfer documents containing configuration data. It is commonly secured by running HTTP over TLS [RFC2817], [RFC2818].
- The SSH File Transfer Protocol (SFTP) [I-D.ietf-secsh-filexfer] provides roughly the same services as FTP but runs over SSH and thus utilizes the security services provided by SSH.
- UNIX utilities to transfer files such as RCP and SCP provide limited flexibility and they differ in their degree of integration with SSH.
- The Control And Provisioning of Wireless Access Points (CAPWAP) protocol [RFC5415] can be used to control the download of images. CAPWAP can be secured by running CAPWAP over DTLS.

In the second class, we find the following configuration protocols:

- Version 3 of the Simple Network Management Protocol (SNMPv3) [RFC3411] can be used to manipulate MIB objects and to carry event notifications. SNMPv3 has its own security protocol (USM)
The Common Open Policy Service for Policy Provisioning protocol (COPS-PR) [RFC3084] was designed to provision structured policy information from a Policy Decision Point (PDP) to a Policy Enforcement Point (PEP). The COPS protocol [RFC2748] provides an integrity object that can achieve authentication, message integrity, and replay prevention. Optionally, COPS and COPS-PR can run over TLS.

The NETCONF protocol [RFC6241] provides mechanisms to install, manipulate, and delete the configuration of network devices. A protocol extension provides an asynchronous event notification delivery mechanism [RFC5277]. NETCONF by default runs over SSH but can also run over transports secured by TLS.

The Control And Provisioning of Wireless Access Points protocol (CAPWAP) [RFC5415] supports the discovery of so called Access Controller (AC) by Wireless Termination Points (WTPs) and the configuration of WTPs by an AC. While CAPWAP can be extended to configure other devices, its main focus are WTPs. The CAPWAP protocol is protected by using DTLS after the discovery phase.

Table 1 lists the protocols plus their basic properties while Table 2 lists the security options available for each protocol.

<table>
<thead>
<tr>
<th>Transport</th>
<th>Data Transfer Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>TFTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>HTTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>SFTP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>RCP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>SCP</td>
<td>Push or pull of (configuration) files</td>
</tr>
<tr>
<td>CAPWAP</td>
<td>AC pushes configuration parameters, WTP pulls software</td>
</tr>
<tr>
<td>SNMPv3</td>
<td>Push of structured configuration parameters, event notifications</td>
</tr>
<tr>
<td>COPS-PR</td>
<td>Push of structured policy information</td>
</tr>
<tr>
<td>NETCONF</td>
<td>Push of structured configuration data, event notifications</td>
</tr>
</tbody>
</table>

Table 1: Protocols for transporting configuration data
SNMPv3, NETCONF, and COPS-PR carry structured data specified in predefined data models. SNMPv3 and COPS-PR have size limitations on the data objects and thus make the transport of larger software images difficult. NETCONF does not suffer from hard size restrictions and can in principle carry software images inline. However, there is currently no work in progress to standardize the transfer of software images over NETCONF. CAPWAP combines the functions of configuration parameter transport and software download. The parameter transport aspect lacks the generality offered by SNMP, NETCONF, and COPS-PR, since the parameters are specified within the protocol specification itself. The remaining transports are independent of the nature of the information being transferred.

7. Phase 4: Configuration Auditing

To complete the process, it must be possible to audit the configuration status of the device in some detail. This is likely to begin even before all the configuration data has been downloaded. For instance, configuration management may wish to collect basic information such as the MAC addresses of the device’s interfaces, the link-local addresses assigned to them, and similar information for the neighbours of the joining device.

SNMP and SNMP MIB modules are obviously one way to collect this information. NETCONF [RFC6241] is an alternative, but the necessary data models have to be defined. YANG modules for NETCONF [RFC6020] can be generated from existing SNMP MIB modules by translating the SNMP modules into YANG modules [RFC6643].

Another important auditing activity is the analysis of system events.
The SYSLOG protocol [RFC5424] is widely used for this purpose but SNMPv3 and NETCONF can ship event notifications as well. Translations of SNMP notifications into structured SYSLOG messages and vice versa do exist [RFC5675], [RFC5676]. NETCONF can carry SYSLOG content as well [RFC5277].

NETCONF provides generic notifications that help with tracking configuration changes [RFC6470]. Similar standardized configuration change notifications do not exist for SNMP or SYSLOG.

8. Phase 5: Configuration Update

Configuration updates can in principle be handled with the same protocol that delivered the initial configuration. However, in some deployments, the mechanism used for initial configuration might be different.

An advantage of NETCONF over SNMPv3 and CAPWAP in the context of configuration updates is the support of concurrent updates through explicit locking mechanisms and the support of network wide configuration change transactions through the confirmed commit capability.

9. Gap Analysis

This document discussed the automated configuration of devices in large IP networks. Several gaps were identified requiring further specification:

G1: Definition of a DHCP option to provide the IPv4/IPv6 address of a configuration server. Such an option allows a joining device to pickup the configuration server’s address as part of the DHCP exchange. This is particularly interesting for Intra-domain Scenarios.

G2: Definition of DNS SRV records for locating configuration servers. Using SRV records, a joining device can lookup the configuration server’s address in the DNS. This is particularly useful in an Inter-domain Scenario.

G3: Definition of a SLP template for discovering configuration servers. Such a template is useful only in environments where SLP is used also for other purposes.
G4: Definition of NETCONF data models to support the download/update of software images through NETCONF.

G5: Definition of NETCONF data models for collecting basic system information and integrity information (e.g., checksums of software images).

G6: Some management protocols lack a mechanisms for devices to initiate a secure communication channel with a management system ("call home").

10. Security Considerations

The security of a configuration management solution is of crucial importance. Section 6 discusses the security options of several protocols that might be used. The relevant protocol definitions should be consulted to learn more about the specific security aspects of the various protocols.

It should be noted that some steps in the described process, in particular the bootstrapping phase, may not be secure and it is thus important to verify the identity of the device and the identity of the configuration server when a secure connection to a configuration server is established. Usage of IPsec, which focuses on securing the IP layer, may not be sufficient for this.

During the choice of protocols, the available security mechanisms and the required key management infrastructures may play a major role in the selection of protocols. Easy integration into existing Authentication, Authorization and Accounting (AAA) infrastructures can significantly reduce the operational costs associated with the security management of the configuration system.

While [I-D.sarikaya-core-sbootstrapping] discusses security bootstrapping mechanisms in the context of constrained devices, many of the mechanisms are also applicable for bootstrapping security in normal devices.

Finally, [RFC6092] discusses security capabilities for customer premises equipment providing residential IPv6 Internet service.

11. IANA Considerations

This memo includes no request to IANA.
12. Acknowledgements

Thanks to Ronald Bonica, Mehmet Ersue, Wesley George, Yiu Lee, Christopher Liljenstolpe, Kent Watsen, and Cathy Zhou for their comments during the preparation of this memo.

13. Informative References

[I-D.ietf-secsh-filexfer]

[I-D.kwatsen-reverse-ssh]
Watsen, K., "Reverse Secure Shell (Reverse SSH)", draft-kwatsen-reverse-ssh-01 (work in progress), June 2011.

[I-D.sarikaya-core-sbootstrapping]


[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,


[RFC6643] Schoenwaelder, J., "Translation of Structure of Management Information Version 2 (SMIv2) MIB Modules to YANG


Authors’ Addresses

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

Phone: 
Email: tina.tsou.zouting@huawei.com

Juergen Schoenwaelder (editor)
Jacobs University Bremen
Campus Ring 1
Bremen 28759
Germany

Phone: 
Email: j.schoenwaelder@jacobs-university.de
Yang Shi
Huawei Technologies
156, Beiqing Road, Zhongguancun, Haidian District
Beijing
P.R. China
Phone: +86 10 60614043
Email: shiyang1@huawei.com

Tom Taylor
Huawei Technologies
Ottawa, Ontario
Canada
Phone:
Email: tom.taylor.stds@gmail.com

Guoliang Yang
China Telecom
No. 109 Zhongshan Ave. (West), Tianhe District
Guangzhou,
P.R. China
Phone: +86 020 38639615
Email: iamyanggl@gmail.com
Abstract

This document gives an overview of the IETF network management standards and summarizes existing and ongoing development of IETF standards-track network management protocols and data models. The purpose of this document is on the one hand to help system developers and users to select appropriate standard management protocols and data models to address relevant management needs. On the other hand the document can be used as an overview and guideline by other Standard Development Organizations or bodies planning to use IETF management technologies and data models.

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 May 3, 2012.

Copyright Notice

Copyright (c) 2011 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. Scope and Target Audience .............................. 4
   1.2. Related Work ........................................... 5
   1.3. Terminology ............................................. 6
2. Core Network Management Protocols ............................ 7
       2.1.1. Architectural Principles of SNMP ................ 7
       2.1.2. SNMP and its Versions ........................... 9
       2.1.3. Structure of Managed Information (SMI) .......... 10
       2.1.4. SNMP Security and Access Control Models ....... 11
       2.1.5. SNMP Transport Subsystem and Transport Models .. 13
   2.2. SYSLOG Protocol ......................................... 14
   2.3. IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols .................................. 16
   2.4. Network Configuration .................................. 19
       2.4.1. Network Configuration Protocol (NETCONF) ....... 19
       2.4.2. YANG - NETCONF Data Modeling Language .......... 21
3. Network Management Protocols and Mechanisms with specific Focus ....................................................... 22
   3.1. IP Address Management ................................... 23
       3.1.1. Dynamic Host Configuration Protocol (DHCP) .... 23
       3.1.2. Ad-Hoc Network Autoconfiguration ................ 24
   3.2. IPv6 Network Operations ................................ 24
   3.3. Policy-based Management ................................ 25
       3.3.1. IETF Policy Framework ............................ 25
       3.3.2. Use of Common Open Policy Service (COPS) for Policy Provisioning (COPS-PR) ...................... 25
   3.4. IP Performance Metrics (IPPM) ........................ 26
   3.5. Remote Authentication Dial In User Service (RADIUS) .. 28
   3.6. Diameter Base Protocol (DIAMETER) ..................... 30
   3.7. Control And Provisioning of Wireless Access Points (CAPWAP) ................................................. 33
   3.8. Access Node Control Protocol (ANCP) ................... 34
   3.9. Application Configuration Access Protocol (ACAP) .... 34
   3.10. XML Configuration Access Protocol (XCAP) ............ 35
4. Network Management Data Models ................................ 35
   4.1. Fault Management ....................................... 36
   4.2. Configuration Management ................................ 38
   4.3. Accounting Management .................................. 40
   4.4. Performance Management .................................. 41
4.5. Security Management ................................... 43
5. IANA Considerations ........................................ 44
6. Security Considerations ................................. 44
7. Contributors .................................................. 45
8. Acknowledgements ........................................... 45
9. Informative References ................................... 45
Appendix A. High Level Classification of Management Protocols
and Data Models ........................................... 63
  A.1. Protocols classified by the Standard Maturity at IETF  .. 64
  A.2. Protocols Matched to Management Tasks ............... 65
  A.3. Push versus Pull Mechanism ............................ 65
  A.4. Passive versus Active Monitoring ..................... 66
  A.5. Supported Data Model Types and their Extensibility .... 67
Appendix B. New Work related to IETF Management Standards .... 69
  B.1. Energy Management (EMAN) ........................... 69
Appendix C. Open issues .................................... 71
Appendix D. Change Log .................................... 71
  D.1. 01-02 ................................................. 71
  D.2. 00-01 ................................................. 72
  D.3. draft-ersue-opsawg-management-fw-03-00 ............... 72
  D.4. Change Log from draft-ersue-opsawg-management-fw .... 73
      D.4.1. 02-03 ........................................... 73
      D.4.2. 01-02 ........................................... 73
      D.4.3. 00-01 ........................................... 73
1. Introduction

1.1. Scope and Target Audience

This document gives an overview of the IETF network management standards and summarizes existing and ongoing development of IETF standards-track network management protocols and data models.

The target audience of the document is on the one hand IETF working groups, which aim to select appropriate standard management protocols and data models to address their needs concerning network management. On the other hand the document can be used as an overview and guideline by non-IETF Standard Development Organizations (SDO) planning to use IETF management technologies and data models for the realization of management applications. The document can be also used to initiate a discussion between the bodies with the goal to gather new requirements and to detect possible gaps. Finally, this document is directed to all interested parties, which seek to get an overview of the current set of the IETF network management protocols such as network administrators or newcomers to IETF.

Section 2 gives an overview of the IETF core network management standards with a special focus on Simple Network Management Protocol (SNMP), SYSLOG, IP Flow Information Export/Packet Sampling (IPFIX/PSAMP), and Network Configuration (NETCONF). Section 3 discusses IETF management protocols and mechanisms with a specific focus, e.g. IP address management or IP performance management. Section 4 discusses Proposed, Draft and Standard Level data models, such as MIB modules, IPFIX Information Elements, SYSLOG Structured Data Elements, and YANG modules designed to address specific set of management issues. The data models are structured following the management application view and mapped to the network management tasks fault, configuration, accounting, performance, and security management.

Appendix A guides the reader for the high-level selection of management standards. For this, the section classifies the protocols according to high level criteria such as push versus pull mechanism, passive versus active monitoring, as well as categorizes the protocols concerning the network management task they address and their data model extensibility. If the reader is interested only in a subset of the IETF network management protocols and data models described in this document, Appendix A can be used as a dispatcher to the corresponding chapter. Appendix B gives an overview of the new work on Energy Management at IETF.

This document mainly refers to Proposed, Draft or Full Standard documents at IETF (see [RFCSEARCH]). As far as valuable Best Current Practice (BCP) documents are referenced. In exceptional cases and if
the document provides substantial guideline for standard usage or fills an essential gap, Experimental and Informational RFCs are noticed and ongoing work is mentioned.

Information on active and concluded IETF working groups (e.g., their charters, published or currently active documents and mail archive) can be found at [IETF-WGS]).

Note: The final document will not contain any references to Internet-Drafts. Current references in the document are assumed to be published soon.

RFC Editor: Please delete the note above before publication.

1.2. Related Work


[RFC3535] "Overview of the 2002 IAB Network Management Workshop" documented strengths and weaknesses of some IETF management protocols. In choosing existing protocol solutions to meet the management requirements, it is recommended that these strengths and weaknesses be considered, even though some of the recommendations from the 2002 IAB workshop have become outdated, some have been standardized, and some are being worked on at the IETF.

[RFC5706] "Guidelines for Considering Operations and Management of New Protocols and Extensions" recommends working groups to consider operations and management needs, and then select appropriate management protocols and data models. This document can be used to ease surveying the IETF standards-track network management protocols and management data models.

Note that IETF so far has not developed specific technologies for the management of sensor networks. IP-based sensors or constrained devices in such an environment, i.e. with very limited memory and CPU resources, can use e.g. application layer protocols to do simple resource management and monitoring.

Note that the document does not cover OAM technologies on the data-path, e.g. OAM of tunnels, MPLS-TP OAM, Pseudowire, etc. [RFC6371] describes the OAM Framework for MPLS-based Transport Networks. There is an ongoing work on the overview of the OAM toolset for detecting and reporting connection failures or measurement of connection performance parameters [I-D.ietf-opsawg-oam-overview].
1.3. Terminology

This document does not describe standard requirements. Therefore key words from RFC2119 are not used in the document.

- 3GPP: 3rd Generation Partnership Project, a collaboration between groups of telecommunications associations, to prepare the third-generation (3G) mobile phone system specification.

- Agent: A software module that performs the network management functions requested by network management stations. An agent may be implemented in any network element that is to be managed, such as a host, bridge, or router. The 'management server' in NETCONF terminology.

- CLI: Command Line Interface. A management interface that system administrators can use to interact with networking equipment.

- Data model: A mapping of the contents of an information model into a form that is specific to a particular type of data store or repository (see [RFC3444]).

- Event: An occurrence of something in the "real world". Events can be indicated to managers through an event message or notification.

- IAB: Internet Architecture Board

- IANA: Internet Assigned Numbers Authority, an organization that oversees global IP address allocation, autonomous system number allocation, media types, and other Internet Protocol-related code point allocations.

- Information model: An abstraction and representation of entities in a managed environment, their properties, attributes and operations, and the way they relate to each other. Independent of any specific repository, protocol, or platform (see [RFC3444]).

- ITU-T: International Telecommunication Union - Telecommunication Standardization Sector

- Managed object: A management abstraction of a resource; a piece of management information in a MIB module. In the context of SNMP, a structured set of data variables that represent some resource to be managed or other aspect of a managed device.

- Manager: An entity that acts in a manager role, either a user or an application. The counterpart to an agent. A 'management client' in NETCONF terminology.
Management Information Base (MIB): An information repository with related collection of objects that represent an aggregation of resources to be managed. MIB modules are defined by using the modeling language SMI.

MIB module: A MIB definition, typically for a particular network technology feature, that constitutes a subtree in an object identifier tree. A MIB that is provided by a management agent is typically composed of multiple instantiated MIB modules.

Modeling language: A modeling language is any artificial language that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. Examples are SMIV2, XSD, and YANG.

Notification: An event message.

OAM: Operations, Administration, and Maintenance

PDU: Protocol Data Unit, a unit of data, which is specified in a protocol of a given layer consisting protocol-control information and possibly layer-specific data.

Relax NG: REgular LAnguage for XML Next Generation, a schema language for XML.

SDO: Standard Development Organization

Trap: An unsolicited message sent by an agent to a management station to notify an unusual event.

URI: Uniform Resource Identifier, a string of characters used to identify a name or a resource on the Internet. Can be classified as locators (URLs), or as names (URNs), or as both.

XPATH: XML Path Language, a query language for selecting nodes from an XML document.

2. Core Network Management Protocols

2.1. Simple Network Management Protocol (SNMP)

2.1.1. Architectural Principles of SNMP

The SNMPv3 Framework [RFC3410], builds upon both the original SNMPv1 and SNMPv2 framework. The basic structure and components for the SNMP framework did not change between its versions and comprises following components:
managed nodes, each with an SNMP entity providing remote access to
management instrumentation (the agent),

- at least one SNMP entity with management applications (the
  manager), and

- a management protocol used to convey management information
  between the SNMP entities, and management information.

During its evolution, the fundamental architecture of the SNMP
Management Framework remained consistent based on a modular
architecture, which consists of:

- a generic protocol definition independent of the data it is
carrying, and

- a protocol-independent data definition language,

- an information repository containing a data set of management
  information definitions (the Management Information Base, or MIB),
  and

- security and administration.

As such following standards build up the basis of the current SNMP
Management Framework:

- SNMPv3 protocol [STD62],

- the modeling language SMIv2 [RFC2578][RFC2579], and

- MIB modules for different management issues.

The SNMPv3 Framework extends the architectural principles of SNMPv1
and SNMPv2 by:

- building on these three basic architectural components, in some
  cases incorporating them from the SNMPv2 Framework by reference,
  and

- by using the same layering principles in the definition of new
  capabilities in the security and administration portion of the
  architecture.
2.1.2. SNMP and its Versions

SNMP is based on three conceptual entities: Manager, Agent, and the Management Information Base (MIB). In any configuration, at least one manager node runs SNMP management software. Typically, network devices such as bridges, routers, and servers are equipped with an agent. The agent is responsible for providing access to a local MIB of objects that reflects the resources and activity at its node. Following the manager-agent paradigm, an agent can generate notifications and send them as unsolicited messages to the management application.

SNMPv2 enhances this basic functionality with a Trap PDU, an Inform message, a bulk transfer capability and other functional extensions like an administrative model for access control, security extensions, and Manager-to-Manager communication. SNMPv2 entities can have a dual role as manager and agent. However, neither SNMPv1 nor SNMPv2 offers sufficient security features. To address the security deficiencies of SNMPv1/v2, SNMPv3 was issued as a set of Proposed Standards (see [STD62]).

[BCP74][RFC3584] "Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework" gives an overview of the relevant standard documents on the three SNMP versions. The BCP document furthermore describes how to convert MIB modules from SMIv1 to SMIv2 format and how to translate notification parameters as well as describes the mapping between the message processing and security models (see [RFC3584]).

SNMP utilizes the Management Information Base, a virtual information store of modules of managed objects. Generally, standard MIB modules support common functionality in a device. Operators often define additional MIB modules for their enterprise or use the Command Line Interface (CLI) to configure non-standard data in managed devices and their interfaces.

SNMPv2 trap and inform PDUs can alert an operator or an application when some aspect of a protocol fails or encounters an error condition, and the contents of a notification can be used to guide subsequent SNMP polling to gather additional information about an event.

SNMP is widely used for monitoring of fault and performance data and with its stateless nature SNMP also works well for status polling and determining the operational state of specific functionality. The widespread use of counters in standard MIB modules permits the interoperable comparison of statistics across devices from different vendors. Counters have been especially useful in monitoring bytes
and packets going in and out over various protocol interfaces. SNMP is often used to poll basic parameter of a device (e.g. sysUpTime, which reports the time since the last reinitialization of the device) to check for operational liveliness, and to detect discontinuities in counters. Some operators use SNMP also for configuration management in their environment (e.g. for DOCSIS-based systems such as cable modems).

SNMPv1 [RFC1157] is a Full Standard that the IETF has declared Historic and it is not recommended due to its lack of security features. "Community-based SNMPv2" [RFC1901] is an Experimental RFC, which IETF has declared Historic and it is not recommended due to its lack of security features.

SNMPv3 [STD62] is a Full Standard that is recommended due to its security features, including support for authentication, encryption, message timeliness and integrity checking, and fine-grained data access controls. An overview of the SNMPv3 document set is in [RFC3410].

Standards exist to use SNMP over diverse transport and link layer protocols, including Transmission Control Protocol (TCP) [STD7][RFC0793], User Datagram Protocol (UDP) [STD6][RFC0768], Ethernet [RFC4789], and others (see Section 2.1.5.1).

2.1.3. Structure of Managed Information (SMI)

SNMP MIB modules are defined with the notation and grammar specified as the Structure of Managed Information (SMI). The SMI uses an adapted subset of Abstract Syntax Notation One (ASN.1) [ITU-X680].

The SMI is divided into three parts: module definitions, object definitions, and notification definitions.

- Module definitions are used when describing information modules. An ASN.1 macro, MODULE-IDENTITY, is used to concisely convey the semantics of an information module.

- Object definitions are used when describing managed objects. An ASN.1 macro, OBJECT-TYPE, is used to concisely convey the syntax and semantics of a managed object.

- Notification definitions are used when describing unsolicited transmissions of management information. An ASN.1 macro, NOTIFICATION-TYPE, is used to concisely convey the syntax and semantics of a notification.

SMIPv1 is specified in [STD16][RFC1155] "Structure and Identification
of Management Information for TCP/IP-based Internets" and
[STD16][RFC1212] "Concise MIB Definitions". [RFC1215] specifies
conventions for defining SNMP traps. Note that SMIv1 is outdated and
is not recommended to use.

SMIv2 is the new notation for managed information definition and
should be used to define MIB modules. SMIv2 is specified in
following RFCs:

- [STD58][RFC2578] defines Version 2 of the Structure of Management
  Information (SMIv2),
- [STD58][RFC2579] defines common MIB "Textual Conventions",
- [STD58][RFC2580] defines Conformance Statements and requirements
  for defining agent and manager capabilities, and
- [RFC3584] defines the mapping rules for and the conversion of MIB
  modules between SMIv1 and SMIv2 formats.

2.1.4. SNMP Security and Access Control Models

2.1.4.1. Security Requirements on the SNMP Management Framework

Several of the classical threats to network protocols are applicable
to management problem space and therefore applicable to any security
model used in an SNMP Management Framework. This section lists
principal threats, secondary threats, and threats which are of lesser
importance (see [RFC3411] for the detailed description of the
security threats).

The principal threats against which SNMP Security Models can provide
protection are, "modification of information" by an unauthorized
entity, and "masquerade", i.e. the danger that management operations
not authorized for some principal may be attempted by assuming the
identity of another principal.

Secondary threats against which SNMP Security Models within this
architecture can provide protection are "message stream
modification", e.g. re-ordering, delay or replay of messages, and
"disclosure", i.e. the danger of eavesdropping on the exchanges
between SNMP engines.

There are two threats against which a Security Model within this
architecture does not protect, since they are deemed to be of lesser
importance in this context: "Denial of Service" and "Traffic
Analysis" (see [RFC3411]).
2.1.4.2. User-Based Security Model (USM)

SNMPv3 [STD62] introduced the User Security Model (USM). USM provides authentication and privacy services for SNMP and is specified in [RFC3414]. Specifically, USM is designed to secure against the principal and secondary threats discussed in Section 2.1.4.1. USM does not secure against Denial of Service and attacks based on Traffic Analysis.

The security services the USM security model supports are:

- **Data Integrity** is the provision of the property that data has not been altered or destroyed in an unauthorized manner, nor have data sequences been altered to an extent greater than can occur non-maliciously.

- **Data Origin Authentication** is the provision of the property that the claimed identity of the user on whose behalf received data was originated is supported.

- **Data Confidentiality** is the provision of the property that information is not made available or disclosed to unauthorized individuals, entities, or processes.

- **Message timeliness and limited replay protection** is the provision of the property that a message whose generation time is outside of a specified time window is not accepted.

See [RFC3414] for a detailed description of SNMPv3 USM.

2.1.4.3. View-Based Access Control Model (VACM)

SNMPv3 [STD62] introduced the View-Based Access Control (VACM) facility. The VACM [RFC3415] enables the configuration of agents to provide different levels of access to the agent’s MIB. An agent entity can restrict access to its MIB for a particular manager entity in two ways:

- The agent entity can restrict access to a certain portion of its MIB, e.g., an agent may restrict most manager principals to viewing performance-related statistics and allow only a single designated manager principal to view and update configuration parameters.

- The agent can limit the operations that a principal can use on that portion of the MIB. E.g., a particular manager principal could be limited to read-only access to a portion of an agent’s MIB.
VACM defines five elements that make up the Access Control Model: groups, security level, contexts, MIB views, and access policy. Access to a MIB module is controlled by means of a MIB view.

See [RFC3415] for a detailed description of SNMPv3 VACM.

2.1.5. SNMP Transport Subsystem and Transport Models

The User-based Security Model (USM) was designed to be independent of other existing security infrastructures to ensure it could function when third-party authentication services were not available. As a result, USM utilizes a separate user and key-management infrastructure. Operators have reported that the deployment of a separate user and key-management infrastructure in order to use SNMPv3 is costly and hinders the deployment of SNMPv3.

SNMP Transport Subsystem [RFC5590] extends the original SNMP architecture and transport model and enables the use of transport protocols to provide message security unifying the administrative security management for SNMP, and other management interfaces.

Transport Models are tied into the SNMP framework through the Transport Subsystem. The Transport Security Model [RFC5591] has been designed to work on top of lower-layer, secure Transport Models.

The SNMP Transport Model defines an alternative to existing standard transport mappings described in [RFC3417] e.g. for SNMP over UDP, in [RFC4789] for SNMP over IEEE 802 networks as well as in the Experimental RFC [RFC3430] defining SNMP over TCP.

2.1.5.1. SNMP Transport Security Model

The SNMP Transport Security Model [RFC5591] is an alternative to the existing SNMPv1 Security Model [RFC3584], the SNMPv2c Security Model [RFC3584], and the User-based Security Model [RFC3414].

The Transport Security Model utilizes one or more lower-layer security mechanisms to provide message-oriented security services. These include authentication of the sender, encryption, timeliness checking, and data integrity checking.

A secure transport model sets up an authenticated and possibly encrypted session between the Transport Models of two SNMP engines. After a transport-layer session is established, SNMP messages can be sent through this session from one SNMP engine to the other. The new Transport Model supports the sending of multiple SNMP messages through the same session to amortize the costs of establishing a security association.

The SSH Transport Model makes use of the commonly deployed SSH security and key-management infrastructure. [RFC5592] furthermore defines MIB objects for monitoring and managing the SSH Transport Model for SNMP.

The Transport Layer Security (TLS) transport model [RFC6353] uses either the TLS protocol or the Datagram TLS (DTLS) protocol. The TLS and DTLS protocols provide authentication and privacy services for SNMP applications. TLS transport model supports the sending of SNMP messages over TLS and TCP and over DTLS and UDP. [RFC6353] furthermore defines MIB objects for managing the TLS Transport Model for SNMP.

Note: Different IETF standards use security layers to address security threads (e.g. TLS [RFC5246], Simple Authentication and Security Layer (SASL) [RFC4422], and SSH [RFC4251]). Diverse management interfaces from IETF use a secure transport layer to provide secure information and message exchange to build management applications, e.g. SYSLOG [RFC5424], IPFIX [RFC5101] and NETCONF [RFC4741].

[RFC5608] describes the use of a 'Remote Authentication Dial-In User Service' (RADIUS) service by SNMP secure Transport Models for authentication of users and authorization of services. Access control authorization, i.e. how RADIUS attributes and messages are applied to the specific application area of SNMP Access Control Models, and VACM in particular has been specified in [RFC6065].

2.2. SYSLOG Protocol

SYSLOG is a mechanism for distribution of logging information initially used on Unix systems. IETF documented the status quo of the BSD SYSLOG protocol in the Informational [RFC3164]. The IETF SYSLOG protocol [RFC5424] obsoletes [RFC3164] and introduces a layered architecture allowing the use of any number of transport protocols, including reliable and secure transports, for transmission of SYSLOG messages.

The body of an BSD SYSLOG message has traditionally been unstructured text. This content is human-friendly, but difficult to parse for applications. The content of BSD SYSLOG messages correlate across vendors and with other event reporting such as SNMP traps.

The SYSLOG protocol enables a machine to send system log messages
across networks to event message collectors. The protocol is simply designed to transport and distribute these event messages. By default, no acknowledgements of the receipt are made, except the reliable delivery extensions specified in [RFC3195] are used. The SYSLOG protocol and process does not require a stringent coordination between the transport sender and the receiver. Indeed, the transmission of SYSLOG messages may be started on a device without a receiver being configured, or even actually physically present. Conversely, many devices will most likely be able to receive messages without explicit configuration or definitions.

BSD SYSLOG had little uniformity for the message format and the content of SYSLOG messages. The IETF has standardized a new message header format, including timestamp, hostname, application, and message ID, to improve filtering, interoperability and correlation between compliant implementations.

The SYSLOG protocol [RFC5424] introduces a mechanism for defining Structured Data Elements (SDEs). The SDEs allow vendors to define their own structured data elements to supplement standardized elements. [RFC5675] defines a mapping from SNMP notifications to SYSLOG messages. [RFC5676] defines a SNMP MIB module to represent SYSLOG messages for sending SYSLOG messages as notifications to SNMP notification receivers. [RFC5674] defines the way alarms are sent in SYSLOG, which includes the mapping of ITU perceived severities onto SYSLOG message fields and a number of alarm-specific definitions from ITU-T X.733 and the IETF Alarm MIB.

[RFC5848] "Signed Syslog Messages" defines a mechanism to add origin authentication, message integrity, replay resistance, message sequencing, and detection of missing messages to the transmitted SYSLOG messages to be used in conjunction with the SYSLOG protocol.

The SYSLOG protocol layered architecture provides support for any number of transport mappings. However, for interoperability purposes, SYSLOG protocol implementers are required to support the transmission of SYSLOG Messages over UDP as defined in [RFC5426].

[RFC3195] describes mappings of the SYSLOG protocol to TCP connections, useful for reliable delivery of event messages. As such the specification provides robustness and security in message delivery with encryption and authentication over a connection-oriented protocol that is unavailable to the usual UDP-based SYSLOG protocol.

IETF furthermore defined the TLS transport mapping for SYSLOG in [RFC5425], which provides a secure connection for the transport of SYSLOG messages. [RFC5425] describes the security threats to SYSLOG
and how TLS can be used to counter such threats. [RFC6012] defines the Datagram Transport Layer Security (DTLS) Transport Mapping for SYSLOG, which can be used if a connectionless transport is desired.

For information on MIB modules related to SYSLOG see Section 4.1.

2.3. IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols

The IPFIX protocol [RFC5101], IP Flow Information eXport, is a Proposed Standard, which defines a push-based data export mechanism for transferring IP flow information in a compact binary format from an exporter to a collector.

The IPFIX architecture [RFC5470] defines the components involved in IP flow measurement and reporting of information on IP flows, particularly, a metering process generating flow records, an exporting process that sends metered flow information using the IPFIX protocol, and a collecting process that receives flow information as IPFIX data records.

After listing the IPFIX requirements in [RFC3917], NetFlow Version 9 [RFC3954] was taken as the basis for the IPFIX protocol and the IPFIX architecture.

IPFIX can run over different transport protocols. The IPFIX protocol [RFC5101] specifies Stream Control Transmission Protocol (SCTP) [RFC4960] as the mandatory transport protocol to implement. Optional alternatives are TCP [STD7] and UDP [STD6].

SCTP is used with its Partial Reliability extension (PR-SCTP) specified in [RFC3758]. [I-D.ietf-ipfix-export-per-sctp-stream] specifies an extension to RFC 5101, when using the PR-SCTP [RFC3758]. The extension offers several advantages over IPFIX export, e.g. the ability to calculate Data Record losses for PR-SCTP, immediate reuse of Template IDs within an SCTP stream, reduced likelihood of Data Record loss, and reduced demands on the Collecting Process.

IPFIX transmits IP flow information in data records containing IPFIX Information Elements (IEs) defined by the IPFIX information model [RFC5102]. IPFIX information elements are quantities with unit and semantics defined by the information model. When transmitted over the IPFIX protocol, only their values need to be carried in data records. This compact encoding allows efficient transport of large numbers of measured flow values. Remaining redundancy in data records can be further reduced by methods described in [RFC5473] (for further discussion on IPFIX IEs see Section 4).
The IPFIX information model is extensible. New information elements can be registered at IANA (see 'IPFIX Information Elements' in [IANA-PROT]). IPFIX also supports the use of proprietary, i.e. enterprise-specific information elements.

The PSAMP protocol [RFC5476] extends the IPFIX protocol by means of transferring information on individual packets. [RFC5475] specifies a set of sampling and filtering techniques for IP packet selection, based on the PSAMP framework [RFC5474]. The PSAMP information model [RFC5477] provides a set of basic information elements for reporting packet information with the IPFIX/PSAMP protocol.

The IPFIX model of an IP traffic flow is uni-directional. [RFC5103] adds means to IPFIX for reporting bi-directional flows, for example both directions of packet flows of a TCP connection.

When enterprise-specific information elements are transmitted with IPFIX, a collector receiving data records may not know the type of received data and cannot choose the right format for storing the contained information. [RFC5610] provides means for providing type information of enterprise-specific information Elements from an exporter to a collector.

Collectors may store received flow information in files. The IPFIX file format [RFC5655] can be used for storing IP flow information in a way that facilitates exchange of traffic flow information between different systems and applications.

In terms of IPFIX and PSAMP configurations, the metering and exporting processes are configured out of band. As the IPFIX protocol is a push mechanism only, IPFIX cannot configure the exporter. The actual configuration of selection processes, caches, exporting processes, and collecting processes of IPFIX and PSAMP compliant monitoring devices is executed using the NETCONF protocol [RFC4741] (see Section 2.4.1). The ‘Configuration Data Model for IPFIX and PSAMP’ is ongoing work and is specified using Unified Modeling Language (UML) class diagrams. The data model is formally defined using the YANG modeling language [RFC6020] in [I-D.draft-ietf-ipfix-configuration-model] (see Section 2.4.2).

At the time of this writing a framework for IPFIX flow mediation is in preparation, which addresses the need for mediation of flow information in IPFIX applications in large operator networks, e.g. for aggregating huge amounts of flow data and for anonymization of flow information (see the problem statement in [RFC5982]).

The IPFIX Mediation Framework [RFC6183] defines the intermediate device between exporters and collectors, which provides an IPFIX
mediation by receiving a record stream from e.g. a collecting process, hosting one or more intermediate processes to transform this stream, and exporting the transformed record stream into IPFIX messages via an exporting process.

Examples for mediation functions are flow aggregation, flow selection, and anonymization of traffic information (see [RFC6235]).

Privacy, integrity, and authentication of exporter and collector are important security requirements for IPFIX [RFC3917]. Confidentiality, integrity, and authenticity of IPFIX data transferred from an exporting process to a collecting process must be ensured. The IPFIX and PSAMP protocols do not define any new security mechanism and rely on the security mechanism of the underlying transport protocol, such as TLS [RFC5246] and DTLS [RFC4347].

The primary goal of IPFIX is the reporting of the flow accounting for flexible flow definitions and usage-based accounting. As described in the IPFIX Applicability Statement [RFC5472], there are also other applications such as traffic profiling, traffic engineering, intrusion detection, and QoS monitoring, that require flow-based traffic measurements and can be realized using IPFIX. IPFIX Applicability Statement explains furthermore the relation of IPFIX to other framework and protocols such as PSAMP, RMON, IPPM. Similar flow information could be also used for security monitoring. The addition of performance metrics in the IPFIX IANA registry [IANA-IPFIX], will extend the IPFIX use case to performance management.

With further information elements, IPFIX can also be applied to monitoring of application-level protocols, for example, Session Initiation Protocol (SIP) [RFC3261] and related media transfer protocols. Requirements to such a monitoring on the application level include measuring signaling quality (e.g., session request delay, session completion ratio, or hops for request), media Quality of Service (QoS) (e.g., jitter, delay or bit rate), and user experience (e.g., Mean Opinion Score).

Note that even if the initial IPFIX focus has been around IP flow information exchange, non IP-related information elements are now specified in IPFIX IANA registration (e.g. MAC (Media Access Control) address, MPLS (Multiprotocol Label Switching) labels, etc.). At the time of this writing, there are requests to widen the focus of IPFIX and to export also non-IP related information elements (such as SIP monitoring IEs).

The IPFIX Structured Data [RFC6313] is an extension to the IPFIX
protocol, which supports hierarchical structured data and lists (sequences) of Information Elements in data records. This extension allows the definition of complex data structures such as variable-length lists and specification of hierarchical containment relationships between templates. Furthermore the extension provides the semantics to express the relationship among multiple list elements in a structured data record.

For information on data models related to the management of the IPFIX and PSAMP protocols see Section 4.1 and Section 4.2. For information on IPFIX/PSAMP IEs see Section 4.3.

2.4. Network Configuration

2.4.1. Network Configuration Protocol (NETCONF)

The IAB workshop on Network Management [RFC3535] determined advanced requirements for configuration management:

- Robustness: Minimizing disruptions and maximizing stability,
- Support of task-oriented view,
- Extensible for new operations,
- Standardized error handling,
- Clear distinction between configuration data and operational state,
- Distribution of configurations to devices under transactional constraints,
- Single and multi-system transactions and scalability in the number of transactions and managed devices,
- Operations on selected subsets of management data,
- Dump and reload a device configuration in a textual format in a standard manner across multiple vendors and device types,
- Support a human interface and a programmatic interface,
- Data modeling language with a human friendly syntax,
- Easy conflict detection and configuration validation, and
Secure transport, authentication, and robust access control.

The NETCONF protocol [RFC4741] is a Proposed Standard that provides mechanisms to install, manipulate, and delete the configuration of network devices and aims to address the configuration management requirements pointed in the IAB workshop. It uses an XML-based data encoding for the configuration data as well as the protocol messages. The NETCONF protocol operations are realized on top of a simple and reliable Remote Procedure Call (RPC) layer. A key aspect of NETCONF is that it allows the functionality of the management protocol to closely mirror the native command line interface of the device.

The NETCONF working group developed the NETCONF Event Notifications Mechanism as an optional capability, which provides an asynchronous message notification delivery service for NETCONF [RFC5277]. NETCONF notification mechanism enables using general purpose notification streams, where the originator of the notification stream can be any managed device (e.g. SNMP notifications).

NETCONF Partial Locking specification introduces fine-grained locking of the configuration datastore to enhance NETCONF for fine-grained transactions on parts of the datastore [RFC5717].

The NETCONF working group also defined the necessary data model to monitor the NETCONF protocol by using the modeling language YANG [RFC6022] (see Section 2.4.2). The monitoring data model includes information about NETCONF datastores, sessions, locks, and statistics, which facilitate the management of a NETCONF server.

NETCONF connections are required to provide authentication, data integrity, confidentiality, and replay protection. NETCONF depends on the underlying transport protocol for this capability. For example, connections can be encrypted in TLS or SSH, depending on the underlying protocol.

The NETCONF working group defined the SSH transport protocol as the mandatory transport binding [RFC4742]. Other optional transport bindings are TLS [RFC5539], BEEP (over TLS) [RFC4744], and SOAP (over HTTP over TLS) [RFC4743].

The NETCONF working group updated the NETCONF base protocol standard as [RFC6241] and the SSH transport protocol mapping as [RFC6242].

At the time of this writing NETCONF Access Control Model (NACM) is being specified. NACM proposes standard mechanisms to restrict protocol access to particular users with a pre-configured subset of operations and content.
2.4.2. YANG - NETCONF Data Modeling Language

Following the guidelines of the IAB management workshop [RFC3535], the NETMOD working group developed a data modeling language defining the semantics of operational and configuration data, notifications, and operations [RFC6020]. The new data modeling language maps directly to XML-encoded content (on the wire) and will serve as the normative description of NETCONF data models.

YANG has following properties addressing specific requirements on a modeling language for configuration management:

- **YANG provides the means to define hierarchical data models.** It supports reusable data types and groupings, i.e., a set of schema nodes that can be reused across module boundaries.

- **YANG supports the distinction between configuration and state data.** In addition, it provides support for modeling event notifications and the specification of operations that extend the base NETCONF operations.

- **YANG allows to express constraints on data models by means of type restrictions and XPATH 1.0 [XPATH] expressions.** XPATH expressions can also be used to make certain portions of a data model conditional.

- **YANG supports the integration of standard and vendor defined data models.** YANG's augmentation mechanism allows to seamlessly augment standard data models with proprietary extensions.

- **YANG data models can be partitioned into collections of features,** allowing low-end devices to only implement the core features of a data model while high-end devices may choose to support all features. The supported features are announced via the NETCONF capability exchange to management applications.

- **The syntax of the YANG language is compact and optimized for human readers.** An associated XML-based syntax called the YANG Independent Notation (YIN) [RFC6020] is available to allow the processing of YANG data models with XML-based tools. The mapping rules for the translation of YANG data models into Document Schema Definition Languages (DSDL), of which Relax NG is a major component, are defined in [RFC6110].

- **Devices implementing standard data models can document deviations from the data model in separate YANG modules.** Applications capable of discovering deviations can make allowances that would otherwise not be possible.
A collection of common data types for IETF-related standards is provided in [RFC6021]. This standard data type library has been derived to a large extent from common SMIv2 data types, generalizing them to a less constrained NETCONF framework.

The document "An Architecture for Network Management using NETCONF and YANG" describes how NETCONF and YANG can be used to build network management applications that meet the needs of network operators [RFC6244].

The Experimental RFC [RFC6095] specifies extensions for YANG introducing language abstractions such as class inheritance and recursive data structures.

[RFC6087] gives guidelines for the use of YANG within IETF and other standardization organizations.

Work is underway to standardize a translation of SMIv2 data models into YANG data models preserving investments into SNMP MIB modules, which are widely available for monitoring purposes.

Several independent and open source implementations of the YANG data modeling language and associated tools are available.

While YANG is a relatively recent data modeling language, some data models have already been produced. The specification of the base NETCONF protocol operations has been revised and uses YANG as the normative modeling language to specify its operations [RFC6241]. The IPFIX working group is currently preparing the normative model for configuring and monitoring IPFIX and PSAMP compliant monitoring devices using the YANG modeling language [I-D.draft-ietf-ipfix-configuration-model].

At the time of this writing the NETMOD working group is developing core system and interface data models. Following the example of the IPFIX configuration model, IETF working groups will prepare models for their specific needs.

For information on data models developed using the YANG modeling language see Section 4.1 and Section 4.2.

3. Network Management Protocols and Mechanisms with specific Focus

This section reviews additional protocols IETF offers for management and discusses for which applications they were designed and/or already successfully deployed. These are protocols that have mostly reached Proposed Standard status or higher within the IETF.
3.1. IP Address Management

3.1.1. Dynamic Host Configuration Protocol (DHCP)

The Draft Standard Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information to hosts on a TCP/IP network and enables as such auto-configuration in IP networks. In addition to IP address management, DHCP can also provide other configuration information, such as default routers, the IP addresses of recursive DNS servers and the IP addresses of NTP servers. As described in [RFC6272] DHCP can be used for IPv4 and IPv6 Address Allocation and Assignment as well as for Service Discovery.

There are two versions of DHCP, one for IPv4 (DHCPv4) [RFC2131] and one for IPv6 (DHCPv6) [RFC3315]. DHCPv4 was defined as an extension to BOOTP (Bootstrap Protocol) [RFC0951]. DHCPv6 was subsequently defined to accommodate new functions required by IPv6 such as assignment of multiple addresses to an interface and to address limitations in the design of DHCPv4 resulting from its origins in BOOTP. While both versions bear the same name and perform the same functionality, the details of DHCPv4 and DHCPv6 are sufficiently different that they can be considered separate protocols.

In addition to the assignment of IP addresses and other configuration information, DHCP options like the Relay Agent Information option (DHCPv4) [RFC3046] and, the Interface-Id Option (DHCPv6) [RFC3315] are widely used by ISPs.

DHCPv6 includes Prefix Delegation [RFC3633], which is used to provision a router with an IPv6 prefix for use in the subnetwork supported by the router.

Following are examples of DHCP options that provide configuration information or access to specific servers. A complete lists of DHCP options are available at [IANA-PROT].

- [RFC3646] describes DHCPv6 options for passing a list of available DNS recursive name servers and a domain search list to a client.
- [RFC2610] describes DHCPv4 options and methods through which entities using the Service Location Protocol can find out the address of Directory Agents in order to transact messages and how the assignment of scope for configuration of SLP User and Service Agents can be achieved.
3.1.2. Ad-Hoc Network Autoconfiguration

Ad-hoc nodes need to configure their network interfaces with locally unique addresses as well as globally routable IPv6 addresses, in order to communicate with devices on the Internet. The IETF AUTOCONF working group developed [RFC5889], which describes the addressing model for ad-hoc networks and how nodes in these networks configure their addresses.

The ad-hoc nodes under consideration are expected to be able to support multi-hop communication by running MANET (Mobile ad-hoc network) routing protocols as developed by the IETF MANET working group.

From the IP layer perspective, an ad hoc network presents itself as a layer 3 multi-hop network formed over a collection of links. The addressing model aims to avoid problems for ad-hoc-unaware parts of the system, such as standard applications running on an ad-hoc node or regular Internet nodes attached to the ad-hoc nodes.

3.2. IPv6 Network Operations

The IPv6 Operations Working Group develops guidelines for the operation of a shared IPv4/IPv6 Internet and provides operational guidance on how to deploy IPv6 into existing IPv4-only networks, as well as into new network installations.

- The Proposed Standard [RFC4213] specifies IPv4 compatibility mechanisms for dual stack and configured tunneling that can be implemented by IPv6 hosts and routers. Dual stack implies providing complete implementations of both IPv4 and IPv6, and configured tunneling provides a means to carry IPv6 packets over unmodified IPv4 routing infrastructures.

- [RFC3574] lists different scenarios in 3GPP defined packet network that would need IPv6 and IPv4 transition, where [RFC4215] does a more detailed analysis of the transition scenarios that may come up in the deployment phase of IPv6 in 3GPP packet networks.

- [RFC4029] describes and analyzes different scenarios for the introduction of IPv6 into an ISP’s existing IPv4 network.

- [RFC4038] specifies scenarios and application aspects of IPv6 transition considering how to enable IPv6 support in applications running on IPv6 hosts, and giving guidance for the development of IP version-independent applications.

- [I-D.weil-shared-transition-space-request] updates RFC 5735 and requests the allocation of an IPv4/10 address block to be used as "Shared Carrier Grade Network Address Translation (CGN) Space" by service providers to number the interfaces that connect CGN devices to Customer Premise Equipment (CPE).

3.3. Policy-based Management

3.3.1. IETF Policy Framework

IETF specified a general policy framework [RFC2753] for managing, sharing, and reusing policies in a vendor independent, interoperable, and scalable manner. [RFC3460] specifies the Policy Core Information Model (PCIM) as an object-oriented information model for representing policy information. PCIM has been developed jointly in the IETF Policy Framework working group and the Common Information Model (CIM) activity in the Distributed Management Task Force (DMTF). PCIM has been published as extensions to CIM [DMTF-CIM].

The IETF Policy Framework is based on a policy-based admission control specifying two main architectural elements, the Policy Enforcement Point (PEP) and the Policy Decision Point (PDP). For the purpose of network management, policies allow an operator to specify how the network is to be configured and monitored by using a descriptive language. Furthermore, it allows the automation of a number of management tasks, according to the requirements set out in the policy module.

IETF Policy Framework has been accepted by the industry as a standard-based policy management approach and has been adopted by different SDOs e.g. for 3GPP charging standards.

3.3.2. Use of Common Open Policy Service (COPS) for Policy Provisioning (COPS-PR)

[RFC3159] defines the Structure of Policy Provisioning Information (SPPI), an extension to the SMIV2 modeling language used to write
Policy Information Base (PIB) modules. COPS-PR [RFC3084] uses the Common Open Policy Service (COPS) protocol [RFC2748] for provisioning of policy information. The COPS-PR specification is independent of the type of policy being provisioned (QoS, Security, etc.) but focuses on the mechanisms and conventions used to communicate provisioned information between policy-decision-points (PDPs) and policy enforcement points (PEPs). Policy data is modeled using Policy Information Base (PIB) modules.

COPS-PR has not been widely deployed, and operators have stated that its use of binary encoding (BER) for management data makes it difficult to develop automated scripts for simple configuration management tasks in most text-based scripting languages. In the IAB Workshop on Network Management [RFC3535], the consensus of operators and protocol developers indicated a lack of interest in PIB modules for use with COPS-PR.

As a result, even if COPS-PR and the Structure of Policy Provisioning Information (SPPI) were initially approved as Proposed Standards, the IESG has not approved any PIB modules as IETF standard, and the use of COPS-PR is not recommended.

3.4. IP Performance Metrics (IPPM)

The IPPM working group has defined metrics for accurately measuring and reporting the quality, performance, and reliability of Internet data delivery. The metrics include connectivity, one-way delay and loss, round-trip delay and loss, delay variation, loss patterns, packet reordering, bulk transport capacity, and link bandwidth capacity.

These metrics are designed for use by network operators and their customers, and provide unbiased quantitative measures of performance. The IPPM metrics have been developed inside an active measurement context, that is, the devices used to measure the metrics produce their own traffic. However, most of the metrics can be used inside a passive context as well. At the time of this writing there is no work planned in the area of passive measurement.

As a property individual IPPM performance and reliability metrics need to be well-defined and concrete thus implementable. Furthermore, the methodology used to implement a metric needs to be repeatable with consistent measurements.

IETF IP Performance Metrics have been introduced widely in the industry and adopted by different SDOs such as the Metro Ethernet Forum.
Following are examples of essential IPPM documents published as Proposed Standard:

- [RFC2330] IPPM Framework document defines a general framework for particular metrics developed by IPPM working group and defines the fundamental concepts of 'metric' and 'measurement methodology' and discusses the issue of measurement uncertainties and errors as well as introduces the notion of empirically defined metrics and how metrics can be composed.


- [RFC2681] "Round-trip Delay Metric for IPPM", defines a metric for round-trip delay of packets across network paths and follows closely the corresponding metric for One-way Delay.

- [RFC3393] "IP Packet Delay Variation Metric", refers to a metric for variation in delay of packets across network paths and is based on the difference in the One-Way-Delay of selected packets called "IP Packet Delay Variation (ipdv)".

- [RFC2680] "One-way Packet Loss Metric for IPPM", defines a metric for one-way packet loss across Internet paths.

- [RFC5560] "One-Way Packet Duplication Metric", defines a metric for the case, where multiple copies of a packet are received and discusses methods to summarize the results of streams.

- [RFC4737] "Packet Reordering Metrics", defines metrics to evaluate whether a network has maintained packet order on a packet-by-packet basis and discusses the measurement issues, including the context information required for all metrics.

- [RFC2678] "IPPM Metrics for Measuring Connectivity", defines a series of metrics for connectivity between a pair of Internet hosts.


- [BCP170] [RFC6390] "Guidelines for Considering New Performance Metric Development" describes the framework and process for developing Performance Metrics of protocols and applications transported over IETF-specified protocols.

To measure these metrics two protocols have been standardized:
3.5. Remote Authentication Dial In User Service (RADIUS)

RADIUS [RFC2865], the Remote Authentication Dial In User Service, is a Draft Standard that describes a client/server protocol for carrying authentication, authorization, and configuration information between a Network Access Server (NAS), which desires to authenticate its links and a shared Authentication Server. The companion document [RFC2866] 'Radius Accounting' describes a protocol for carrying accounting information between a network access server and a shared accounting server. [RFC2867] adds required new RADIUS accounting attributes and new values designed to support the provision of tunneling in dial-up networks.

The RADIUS protocol is widely used in environments like enterprise networks, where a single administrative authority manages the network, and protects the privacy of user information. RADIUS is deployed in fixed broadband access provider networks as well as in cellular broadband operators’ networks.

RADIUS uses attributes to carry the specific authentication, authorization, information and configuration details. RADIUS is extensible with a known limitation of maximum 255 attribute codes and 253 octets as attribute content length. RADIUS has Vendor-Specific Attributes (VSA), which have been used both for vendor-specific purposes as an addition to standardized attributes as well as to extend the limited attribute code space.

The RADIUS protocol uses a shared secret along with the MD5 (Message-Digest algorithm 5) hashing algorithm to secure passwords [RFC1321]. Based on the known threads additional protection like IPsec tunnels are used to further protect the RADIUS traffic. However, building and administering large IPsec protected networks may become a
management burden, especially when IPsec protected RADIUS infrastructure should provide inter-provider connectivity. A trend has been moving towards TLS-based security solutions and establishing dynamic trust relationships between RADIUS servers. Since the introduction of TCP transport for RADIUS, it became natural to have TLS support for RADIUS. An ongoing work specifies the ’TLS encryption for RADIUS’.

[RFC2868] ‘RADIUS Attributes for Tunnel Protocol Support’ defines a number of RADIUS attributes designed to support the compulsory provision of tunneling in dial-up network access. Some applications involve compulsory tunneling i.e. the tunnel is created without any action from the user and without allowing the user any choice in the matter. In order to provide this functionality, specific RADIUS attributes are needed to carry the tunneling information from the RADIUS server to the tunnel end points. [RFC3868] defines the necessary attributes, attribute values and the required IANA registries.

[RFC3162] ‘RADIUS and IPv6’ specifies the operation of RADIUS over IPv6 and the RADIUS attributes used to support the IPv6 network access. [RFC4818] describes how to transport delegated IPv6 prefix information over RADIUS.

[RFC4675] ‘RADIUS Attributes for Virtual LAN and Priority Support’ defines additional attributes for dynamic Virtual LAN assignment and prioritization, for use in provisioning of access to IEEE 802 local area networks usable with RADIUS and DIAMETER.

[RFC5080] ‘Common RADIUS Implementation Issues and Suggested Fixes’ describes common issues seen in RADIUS implementations and suggests some fixes. Where applicable, unclear statements and errors in previous RADIUS specifications are clarified. People designing extensions to RADIUS protocol for various deployment cases should get familiar with RADIUS Design Guidelines [RFC6158] in order to avoid e.g. known interoperability challenges.

[RFC5090] ‘RADIUS Extension for Digest Authentication’ defines an extension to the RADIUS protocol to enable support of Digest Authentication, for use with HTTP-style protocols like the Session Initiation Protocol (SIP) and HTTP.

[RFC5580] ‘Carrying Location Objects in RADIUS and DIAMETER’ describes procedures for conveying access-network ownership and location information based on civic and geospatial location formats in RADIUS and DIAMETER.

[RFC5607] specifies required RADIUS attributes and their values for
authorizing a management access to a NAS. Both local and remote management are supported, with access rights and management privileges. Specific provisions are made for remote management via Framed Management protocols, such as SNMP and NETCONF, and for management access over a secure transport protocols.

[RFC3579] describes how to use RADIUS to convey Extensible Authentication Protocol (EAP) payload between the authenticator and the EAP server using RADIUS. RFC3579 is widely implemented, for example, in WLAN and 802.1X environment. [RFC3580] describes how to use RADIUS with IEEE 802.1X authenticators. In the context of 802.1X and EAP-based authentication, the Vendor Specific Attributes described in [RFC2458] have been widely accepted by the industry. [RFC2869] ‘RADIUS extensions’ is another important RFC related to EAP use. RFC2869 describes additional attributes for carrying AAA information between a NAS and a shared Accounting Server using RADIUS. It also defines attributes to encapsulate EAP message payload.

There are different MIB modules defined for multiple purposes to use with RADIUS (see Section 4.3 and Section 4.5).

3.6. Diameter Base Protocol (DIAMETER)

DIAMETER [RFC3588] is a Proposed Standard that provides an Authentication, Authorization and Accounting (AAA) framework for applications such as network access or IP mobility. DIAMETER is also intended to work in local AAA and in roaming scenarios. DIAMETER provides an upgrade path for RADIUS but is not directly backwards compatible.

DIAMETER is designed to resolve a number of known problems with RADIUS. DIAMETER supports server failover, reliable transport over TCP and SCTP, well documented functions for proxy, redirect and relay agent functions, server-initiated messages, auditability, and capability negotiation. DIAMETER also provides a larger attribute space for Attribute-Value Pairs (AVP) and identifiers than RADIUS. DIAMETER features make it especially appropriate for environments, where the providers of services are in different administrative domains than the maintainer (protector) of confidential user information.

Other notable differences to RADIUS are:

- Network and transport layer security (IPsec or TLS),
- Stateful and stateless models,
Dynamic discovery of peers (using DNS SRV and NAPTR),

Concept of an application that describes how a specific set of commands and Attribute-Value Pairs (AVPs) are treated by DIAMETER nodes. Each application has an IANA assigned unique identifier,

Support of application layer acknowledgements, failover methods and state machines,

Basic support for user-sessions and accounting,

Better roaming support,

Error notification, and

Easy extensibility.

The DIAMETER protocol is designed to be extensible to support e.g. proxies, brokers, mobility and roaming, Network Access Servers (NASREQ), and Accounting and Resource Management. DIAMETER applications extend the DIAMETER base protocol by adding new commands and/or attributes. Each application is defined by an unique IANA assigned application identifier and can add new command codes and/or new mandatory AVPs.

The DIAMETER application identifier space has been divided into Standards Track and 'First Come First Served' vendor-specific applications. Following are examples for DIAMETER applications published at IETF:

Diameter Base Protocol Application [RFC3588],

Diameter Base Accounting Application [RFC3588],

Diameter Mobile IPv4 Application [RFC4004],

Diameter Network Access Server Application (NASREQ, [RFC4005]),

Diameter Extensible Authentication Protocol Application [RFC4072],

Diameter Credit-Control Application [RFC4006],

Diameter Session Initiation Protocol Application [RFC4740], and

Diameter Quality-of-Service Application [RFC5866].

Diameter Mobile IPv6 IKE (MIP6I) Application [RFC5778].
The large majority of DIAMETER applications are vendor-specific and mainly used in various SDOs outside IETF. One example SDO using DIAMETER extensively is 3GPP (e.g. 3GPP ‘IP Multimedia Subsystem’ (IMS) uses DIAMETER based interfaces (e.g. Cx) [3GPPIMS]). Recently, during the standardization of the ’3GPP Evolved Packet Core’ [3GPP-EPC], DIAMETER was chosen as the only AAA signaling protocol.

One part of DIAMETER’s extensibility mechanism is an easy and consistent way of creating new commands for the need of applications. RFC3588 proposed to define DIAMETER command code allocations with a new RFC. This policy decision caused undesired use and redefinition of existing Command Codes within SDOs. Diverse RFCs have been published as simple command code allocations for other SDO purposes (see [RFC3589], [RFC5224], [RFC5431] and [RFC5516]). [RFC5719] changed the Command Code policy and added a range for vendor-specific Command Codes to be allocated on a ‘First Come First Served’ basis by IANA.

The implementation and deployment experience of DIAMETER has led to the currently ongoing development of an update of the base protocol [I-D.ietf-dime-rfc3588bis]. One of the major changes is the introduction of TLS as the preferred security mechanism and deprecating the in-band security negotiation for TLS.

Some DIAMETER protocol enhancements and clarifications that logically fit better into [I-D.ietf-dime-rfc3588bis], are also needed on the existing RFC3588 based deployments. Therefore, protocol extensions specifically usable in large inter-provider roaming network scenarios are made available for RFC3588. Two currently existing specifications are mentioned below:

- "Clarifications on the Routing of DIAMETER Requests Based on the Username and the Realm" [RFC5729] defines the behavior required for DIAMETER agents to route requests when the User-Name AVP contains a Network Access Identifier formatted with multiple realms. These multi-realm Network Access Identifiers are used in order to force the routing of request messages through a predefined list of mediating realms.

- The ongoing work on "Diameter Extended NAPTR" [I-D.ietf-dime-extended-naptr] describes an improved DNS-based dynamic DIAMETER Agent discovery mechanism without having to do DIAMETER capability exchange beforehand with a number of agents.
There have been a growing number of DIAMETER framework documents at IETF that basically are just a collection of AVPs for a specific purpose or a system architecture with semantical AVP descriptions and a logic for "imaginary" applications. From standardization point of view, this practice allows the development of larger system architecture documents that do not need to reference AVPs or application logic outside IETF. Below are examples of a few recent AVP and framework documents:


- 'Traffic Classification and Quality of Service (QoS) Attributes for Diameter' [RFC5777] defines a number of DIAMETER AVPs for traffic classification with actions for filtering and QoS treatment.

- 'Diameter Proxy Mobile IPv6: Mobile Access Gateway and Local Mobility Anchor Interaction with Diameter Server' [RFC5779] defines AAA interactions between Proxy Mobile IPv6 (PMIPv6) entities (Mobile Access Gateway and Local Mobility Anchor) and an AAA server within a PMIPv6 Domain. For information on MIB modules related to DIAMETER see Section 4.5.

3.7. Control And Provisioning of Wireless Access Points (CAPWAP)

Wireless LAN (WLAN) product architectures have evolved from single autonomous Access Points to systems consisting of a centralized Access Controller (AC) and Wireless Termination Points (WTPs). The general goal of centralized control architectures is to move access control, including user authentication and authorization, mobility management, and radio management from the single access point to a centralized controller, where an Access Points pulls the information from the Access Controller.

Based on the CAPWAP Architecture Taxonomy work [RFC4118] the CAPWAP working group developed the CAPWAP protocol [RFC5415] to facilitate control, management and provisioning of WTPs specifying the services, functions and resources relating to 802.11 WLAN Termination Points in order to allow for interoperable implementations of WTPs and ACs. The protocol defines the CAPWAP control plane including the primitives to control data access. The protocol document also specifies how configuration management of WTPs can be done and
defines CAPWAP operations responsible for debugging, gathering
statistics, logging, and firmware management as well as discusses
operational and transport considerations.

The CAPWAP protocol is prepared to be independent of Layer 2
technologies, and meets the objectives in "Objectives for Control and
Provisioning of Wireless Access Points (CAPWAP)" [RFC4564]. Separate
binding extensions enable the use with additional wireless
technologies. [RFC5416] defines CAPWAP Protocol Binding for IEEE
802.11.

CAPWAP Control messages, and optionally CAPWAP Data messages, are
secured using DTLS [RFC4347]. DTLS is used as a tightly integrated,
secure wrapper for the CAPWAP protocol.

For information on MIB modules related to CAPWAP see Section 4.2.

3.8. Access Node Control Protocol (ANCP)

The Access Node Control Protocol (ANCP) [RFC6320] realizes a control
plane between a service-oriented layer 3 edge device, the Network
Access Server (NAS) and a layer 2 Access Node (AN), e.g., Digital
Subscriber Line Access Module (DSLAM). As such ANCP operates in a
multi-service reference architecture and communicates QoS-, service-
and subscriber-related configuration and operation information
between a NAS and an Access Node.

The main goal of this protocol is to configure and manage access
equipments and allow them to report information to the NAS in order
to enable and optimize configuration.

The framework and requirements for an Access Node control mechanism
and the use cases for ANCP are documented in [RFC5851].

The ANCP protocol offers authentication, and authorization between AN
and NAS nodes and provides replay protection and data-origin
authentication. ANCP protocol solution is also robust against
Denial-of-Service (DoS) attacks. Furthermore, the ANCP protocol
solution is recommended to offer confidentiality protection.
Security Threats and Security Requirements for ANCP are discussed in
[RFC5713].

3.9. Application Configuration Access Protocol (ACAP)

The Application Configuration Access Protocol (ACAP) [RFC2244] is a
Proposed Standard protocol designed to support remote storage and
access of program option, configuration and preference information.
The data store model is designed to allow a client relatively simple
access to interesting data, to allow new information to be easily added without server re-configuration, and to promote the use of both standardized data and custom or proprietary data. Key features include "inheritance" which can be used to manage default values for configuration settings and access control lists which allow interesting personal information to be shared and group information to be restricted.

ACAP’s primary purpose is to allow applications access to their configuration data from multiple network-connected computers. Users can then use any network-connected computer, run any ACAP-enabled application and have access to their own configuration data. To enable wide usage client simplicity has been preferred to server or protocol simplicity whenever reasonable.

The ACAP ‘authenticate’ command uses Simple Authentication and Security Layer (SASL) [RFC4422] to provide basic authentication, authorization, integrity and privacy services. All ACAP implementations are required to implement the CRAM-MD5 (Challenge-Response Authentication Mechanism) [RFC2195] for authentication, which can be disabled based on the site security policy.

3.10. XML Configuration Access Protocol (XCAP)

The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) [RFC4825] is a Proposed Standard protocol that allows a client to read, write, and modify application configuration data stored in XML format on a server.

XCAP is a protocol that can be used to manipulate per-user data. XCAP is a set of conventions for mapping XML documents and document components into HTTP URIs, rules for how the modification of one resource affects another, data validation constraints, and authorization policies associated with access to those resources. Because of this structure, normal HTTP primitives can be used to manipulate the data. Like ACAP, XCAP supports the configuration needs for a multiplicity of applications.

All XCAP servers are required to implement HTTP Digest Authentication [RFC2617]. Furthermore, XCAP servers are required to implement HTTP over TLS (HTTPS) [RFC2818]. It is recommended that administrators use an HTTPS URI as the XCAP root URI, so that the digest client authentication occurs over TLS.

4. Network Management Data Models

This section lists management data models standardized at IETF, which can be reused and applied to different management solutions. The
subsections below are structured following the management application view and focus mainly on the management data models for the network management tasks fault, configuration, accounting, performance, and security management.

The advancement process for management data models beyond Proposed Standard status, has been defined in [BCP27][RFC2438] with a more pragmatic approach and special considerations on data model specification interoperability. However, most IETF management data models never advance beyond Proposed Standard.

This section gives an overview of management data models that have reached Draft or Proposed Standard status at the IETF. In exceptional cases important Informational RFCs are referred.

The different data models covered in this section are MIB modules, IPFIX Information Elements, SYSLOG Structured Data Elements, and YANG modules.

Note that IETF does not use the FCAPS view as an organizing principle for its data models. However, FCAPS view is used widely outside of IETF for the realization of management tasks and applications. This document provides an overview of IETF data models with an FCAPS view to enable people outside of IETF to understand the relevant data models. There are many technology-specific IETF data models, such as transmission and protocol MIBs, which are not mentioned in this document and can be found at [RFCSEARCH].

4.1. Fault Management

Draft Standards:

[RFC3418], part of SNMPv3 standard [STD62], contains objects in the system group that are often polled to determine if a device is still operating, and sysUpTime can be used to detect if a system has rebooted, and counters have been reinitialized.

[RFC3413], part of SNMPv3 standard [STD62], includes objects designed for managing notifications, including tables for addressing, retry parameters, security, lists of targets for notifications, and user customization filters.

The Interfaces Group MIB [RFC2863] builds on MIB II [RFC1229] and is used for managing and monitoring of network interfaces. The ‘interfaces’ group in MIB II [RFC1229] defines a generic set of managed objects and provides the means for additional managed objects specific to particular types of network interfaces, such as Ethernet. Extensions to the ‘interfaces’ group for media-specific management
can be defined based on these managed objects. Experience with media-specific MIB modules has shown that the model defined by MIB-II is too simplistic and static for some types of media-specific management. The Interfaces Group MIB incorporates the interfaces group extensions documented in MIB II and standardizes an evolution to this model as well as fills in the detected gaps.

An RMON (Remote Network Monitoring) monitor [RFC2819] can be configured to recognize conditions, most notably error conditions, and continuously to check for them. When one of these conditions occurs, the event may be logged, and management stations may be notified in a number of ways (for further discussion on RMON see Section 4.4).

Proposed Standards:

DISMAN-EVENT-MIB in [RFC2981] and DISMAN-EXPRESSION-MIB in [RFC2982] provide a superset of the capabilities of the RMON alarm and event groups. These modules provide mechanisms for thresholding and reporting anomalous events to management applications.

The ALARM MIB in [RFC3877] and the Alarm Reporting Control MIB in [RFC3878] specify mechanisms for expressing state transition models for persistent problem states. ALARM MIB defines:
- a mechanism for expressing state transition models for persistent problem states,
- a mechanism to correlate a notification with subsequent state transition notifications about the same entity/object, and
- a generic alarm reporting mechanism (extends ITU-T work on X.733 [ITU-X733]).

[RFC3877] in particular defines objects for controlling the reporting of alarm conditions and extends ITU-T work M.3100 Amendment 3 [ITU-M3100].

Other MIB modules that may be applied to fault management with SNMP include:

- NOTIFICATION-LOG-MIB [RFC3014] describes managed objects used for logging SNMP Notifications.

- ENTITY-STATE-MIB [RFC4268] describes extensions to the Entity MIB to provide information about the state of physical entities.

- ENTITY-SENSOR-MIB [RFC3433] describes managed objects for extending the Entity MIB to provide generalized access to information related to physical sensors, which are often found in networking equipment (such as chassis temperature, fan RPM, power
supply voltage).

The SYSLOG protocol document [RFC5424] defines an initial set of Structured Data Elements (SDEs) that relate to content time quality, content origin, and meta-information about the message, such as language. Proprietary SDEs can be used to supplement the IETF-defined SDEs.

The IETF has standardized MIB Textual-Conventions for facility and severity labels and codes to encourage consistency between SYSLOG and MIB representations of these event properties [RFC5427]. The intent is that these textual conventions will be imported and used in MIB modules that would otherwise define their own representations.

An IPFIX MIB module [RFC5815] has been defined for monitoring IPFIX meters, exporters and collectors (see Section 2.3). The ongoing work on PSAMP MIB module extends the IPFIX MIB modules by managed objects for monitoring PSAMP implementations [I-D.ietf-ipfix-psamp-mib].

The NETCONF working group defined the necessary data model to monitor the NETCONF protocol with the modeling language YANG [RFC6022]. The monitoring data model includes information about NETCONF datastores, sessions, locks, and statistics, which facilitate the management of a NETCONF server. NETCONF monitoring RFC also defines methods for NETCONF clients to discover the data models supported by a NETCONF server and defines the operation <get-schema> to retrieve them.

4.2. Configuration Management

MIB modules for monitoring of network configuration (e.g. for physical and logical network topologies) already exist and provide some of the desired capabilities. New MIB modules might be developed for the target functionality to allow operators to monitor and modify the operational parameters, such as timer granularity, event reporting thresholds, target addresses, etc.

Draft standards:

[RFC3418] contains objects in the system group useful e.g. for identifying the type of device, the location of the device, the person responsible for the device. [RFC3413], part of STD 62 SNMPv3, includes objects designed for configuring notification destinations, and for configuring proxy- forwarding SNMP agents, which can be used to forward messages through firewalls and Network Address Translation (NAT) devices.

The Interfaces Group MIB [RFC2863] is used for the configuration and monitoring of network interface parameters. [RFC2863] includes the
'interfaces' group of MIB-II and discusses the experience gained from the definition of numerous media-specific MIB modules for use in conjunction with the 'interfaces' group for managing various sub-layers beneath the internetwork-layer.

Proposed standards:

The Entity MIB [RFC4133] is used for managing multiple logical and physical entities managed by a single SNMP agent. This module provides a useful mechanism for identifying the entities comprising a system. There are also event notifications defined for configuration changes that may be useful to management applications.

[RFC3165] supports the use of user-written scripts to delegate management functionality.

Policy Based Management MIB [RFC4011] defines objects that enable policy-based monitoring using SNMP, using a scripting language, and a script execution environment.

Few vendors have implemented MIB modules that support scripting. Some vendors consider running user-developed scripts within the managed device as a violation of support agreements.

For configuring IPFIX and PSMAP devices, the IPFIX working group is currently developing an XML-based configuration data model [I-D.ietf-ipfix-configuration-model], in close collaboration with the NETMOD working group. IPFIX configuration data model uses YANG as modeling language (see Section 2.4.2). The model specifies the necessary data for configuring and monitoring selection processes, caches, exporting processes, and collecting processes of IPFIX and PSAMP compliant monitoring devices.

At the time of this writing the NETMOD working group is developing core system and interface models in YANG.

Non-standard data models:

The CAPWAP protocol exchanges Type Length Values (TLV). The base TLVs are specified in [RFC5415], while the TLVs for IEEE 802.11 are specified in [RFC5416]. CAPWAP Base MIB [RFC5833] specifies managed objects for modeling the CAPWAP Protocol and provides configuration and WTP status-monitoring aspects of CAPWAP, where CAPWAP Binding MIB [RFC5834] defines managed objects for modeling of CAPWAP protocol for IEEE 802.11 wireless binding. Note: RFC 5833 and RFC 5834 have been published as Informational RFCs to provide the basis for future work on a SNMP management of the CAPWAP protocol.
4.3. Accounting Management

Non-standard data models:

[RFC4670] ‘RADIUS Accounting Client MIB for IPv6’ defines RADIUS Accounting Client MIB objects that support version-neutral IP addressing formats.


IPFIX/PSAMP Information Elements:

As expressed in Section 2.3, the IPFIX architecture [RFC5470] defines components involved in IP flow measurement and reporting of information on IP flows. As such IPFIX records provide fine-grained measurement data for flexible and detailed usage reporting and enable usage-based accounting.

The IPFIX Information Elements (IE) have been initially defined in the IPFIX Information Model [RFC5102] and registered at the IANA [IANA-IPFIX]. The IPFIX IEs are composed of two types: IEs related to identification of IP flows and IEs related to counter and timestamps.

Following are examples of IEs related to identification of IP flows:

- Identifiers for line cards, ports, interfaces, etc...
- IP header fields such as source and destination IP addresses
- Transport header fields such as UDP and TCP ports
- Sub-IP header fields such as source and destination MAC address, MPLS label stack entries
- Derived packet properties such as IGP and BGP next hop IP address, BGP AS, etc.
- Min/max flow properties such as the minimum and maximum IP total length and Time To Live (TTL)

Below are examples of IEs related to counter and timestamps:

- Flow timestamps such as flow start times, flow end times, and flow duration,
o Per flow counters such as octets count, packets count,
o Miscellaneous flow properties such as flow duration.

The Information Elements specified in the IPFIX information model [RFC5102] are used by the PSAMP protocol where applicable. Packet Sampling (PSAMP) Parameters defined in the PSAMP protocol specification are registered at [IANA-PSAMP]. An additional set of PSAMP Information Elements for reporting packet information with the IPFIX/PSAMP protocol such as Sampling-related IEs are specified in the PSAMP Information Model [RFC5477]. These IEs fulfill the requirements on reporting of different sampling and filtering techniques specified in [RFC5475].

4.4. Performance Management

Full Standards:

RMON (Remote Network Monitoring) MIB [RFC2819] has the Full Standard status [STD59] and defines objects for managing remote network devices and collecting data related to network performance and traffic. An organization may employ many remote management probes, one per network segment, to manage its internet. These devices may be used by a network service provider to access a client network, often geographically remote. Most of the objects in the RMON MIB module are suitable for the management of any type of network, where some of them are specific to management of Ethernet networks.

RMON allows a probe to be configured to perform diagnostics and to collect network statistics continuously, even when communication with the management station may not be possible or efficient. The alarm group periodically takes statistical samples from variables in the probe and compares them to previously configured thresholds. If the monitored variable crosses a threshold, an event is generated.

The RMON host group discovers hosts on the network by keeping a list of source and destination MAC Addresses seen in good packets promiscuously received from the network, and contains statistics associated with each host. The hostTopN group is used to prepare reports that describe the hosts that top a list ordered by one of their statistics. The available statistics are samples of one of their base statistics over an interval specified by the management station. Thus, these statistics are rate based. The management station also selects how many such hosts are reported.

The RMON matrix group stores statistics for conversations between sets of two addresses. The filter group allows packets to be matched by a filter equation. These matched packets form a data stream that
may be captured or may generate events. The Packet Capture group allows packets to be captured after they flow through a channel. The event group controls the generation and notification of events from this device.

Draft standards:

The RMON-2 MIB [RFC4502] extends RMON by providing RMON analysis up to the application layer and defines performance data to monitor. The SMON MIB [RFC2613] extends RMON by providing RMON analysis for switched networks.

Proposed standards:

RMON MIB Extensions for High Capacity Alarms [RFC3434] describes managed objects for extending the alarm thresholding capabilities found in the RMON MIB and provides similar threshold monitoring of objects based on the Counter64 data type.


RMON MIB Extensions for Interface Parameters Monitoring [RFC3144] describes an extension to the RMON MIB with a method of sorting the interfaces of a monitored device according to values of parameters specific to this interface.

[RFC4710] describes Real-Time Application Quality of Service Monitoring. RAQMON is part of the RMON protocol family, and supports end-2-end QoS monitoring for multiple concurrent applications and does not relate to a specific application transport. RAQMON is scalable and works well with encrypted payload and signaling. RAQMON uses TCP to transport RAQMON PDUs.

[RFC4711] proposes an extension to the Remote Monitoring MIB [RFC2819] and describes managed objects used for real-time application Quality of Service (QoS) monitoring. [RFC4712] specifies two transport mappings for the RAQMON information model using TCP as a native transport and SNMP to carry the RAQMON information from a RAQMON Data Source (RDS) to a RAQMON Report Collector (RRC).

Application Performance Measurement MIB [RFC3729] uses the architecture created in the RMON MIB and defines objects by providing measurement and analysis of the application performance as experienced by end-users. Application performance measurement measures the quality of service delivered to end-users by applications.
Transport Performance Metrics MIB [RFC4150] describes managed objects used for monitoring selectable performance metrics and statistics derived from the monitoring of network packets and sub-application level transactions. The metrics can be defined through reference to existing IETF, ITU, and other standards organizations’ documents.

The IPPM working group defined an Information Model and XML Data Model for Traceroute Measurements [RFC5388], which defines a common information model dividing the information elements into two semantically separated groups (configuration elements and results elements) with an additional element to relate configuration elements and results elements by means of a common unique identifier. Based on the information model, an XML data model is provided to store the results of traceroute measurements.

The IPPM working group has defined [BCP108][RFC4148] "IP Performance Metrics (IPPM) Metrics Registry". The IANA-assigned registry contains an initial set of OBJECT IDENTITIES to currently defined metrics in the IETF as well as defines the rules for adding IP Performance Metrics that are defined in the future. However, the current registry structure has been found to be insufficiently detailed to uniquely identify IPPM metrics. Due to the ambiguities between the current metrics registrations and the metrics used, and the apparent non-adoption of the registry in practice, it has been proposed to reclassify [RFC4148] as Obsolete.

Note: With the publication of [RFC6248] the latest IANA registry for IPPM metrics and [RFC4148] have been declared Obsolete and IANA prevents registering new metrics. Actual users can continue using the current registry and its contents.

SIP Package for Voice Quality Reporting [RFC6035] defines a SIP event package that enables the collection and reporting of metrics that measure the quality for Voice over Internet Protocol (VoIP) sessions.

4.5. Security Management

There are numerous MIB modules defined for multiple purposes to use with RADIUS:

- [RFC4668] 'RADIUS Authentication Client MIB for IPv6’ defines RADIUS Authentication Client MIB objects that support version-neutral IP addressing formats and defines a set of extensions for RADIUS authentication client functions.

- [RFC4669] 'RADIUS Authentication Server MIB for IPv6’ defines RADIUS Authentication Server MIB objects that support version-neutral IP addressing formats and defines a set of extensions for
RADIUS authentication server functions.

- [RFC4670] 'RADIUS Accounting Client MIB for IPv6' defines RADIUS Accounting Client MIB that objects that support version-neutral IP addressing formats.

- [RFC4671] 'RADIUS Accounting Server MIB for IPv6' defines RADIUS Accounting Server MIB that objects that support version-neutral IP addressing formats.

- [RFC4672] 'RADIUS Dynamic Authorization Client MIB' defines the MIB module for entities implementing the client side of the Dynamic Authorization Extensions to RADIUS [RFC5176].


The MIB Module definitions in [RFC4668], [RFC4669], [RFC4670], [RFC4671], [RFC4672], and [RFC4673] are intended to be used only for RADIUS over UDP and therefore do not support RADIUS/TCP. There is also a recommendation that RADIUS clients and servers implementing RADIUS/TCP should not re-use earlier listed MIB modules to perform statistics counting for RADIUS/TCP connections.

Currently there are no standardized MIB modules for DIAMETER applications, which can be considered as a lack on the management side of DIAMETER nodes. There are ongoing efforts to produce standard MIBs for the 'Diameter Base Protocol' [I-D.ietf-dime-diameter-base-protocol-mib] and the 'Diameter Credit-Control Application' [I-D.ietf-dime-diameter-cc-appl-mib].

5. IANA Considerations

This document does not introduce any new code-points or namespaces for registration with IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

This document introduces no new security concerns.

Note to RFC Editor: this section may be removed on publication as an RFC.
7. Contributors

Following persons made significant contributions to this document:

- Ralph Droms (Cisco) - revised the section on IP address management and DHCP.
- Jouni Korhonen (Nokia Siemens Networks) - contributed the sections on RADIUS and DIAMETER.
- Al Morton (AT&T) - contributed to the section on IP Performance Metrics.
- Juergen Quittek (NEC) - contributed the section on IPFIX/PSAMP.
- Juergen Schoenwaelder (Jacobs University Bremen) - contributed the section on YANG.

8. Acknowledgements

The editor would like to thank Tom Petch, Dan Romascanu, Henk Uijterwaal, Alex Clemm, and Randy Presuhn for their valuable suggestions, comments in the OPSAWG sessions and mailing list.

The editor would like to especially thank Dave Harrington, who created the document "Survey of IETF Network Management Standards" a few years ago. While this draft expired, the editor used it as a starting point and enhanced it with a special focus on the description of the IETF network management standards and management data models.

9. Informative References


[BCP27] D. O'Dell, M., "Advancement of MIB specifications on
the IETF Standards Track", October 1998.


[RFC2458] Lu, H., Krishnaswamy, M., Conroy, L., Bellovin, S.,
Burg, F., DeSimone, A., Tewani, K., Davidson, P.,
Schulzrinne, H., and K. Vishwanathan, "Toward the PSTN/
Internet Inter-Networking --Pre-PINT Implementations",
RFC 2458, November 1998.

Information Version 2 (SMIv2)", STD 58, RFC 2578,
April 1999.

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.

[RFC2610] Perkins, C. and E. Guttman, "DHCP Options for Service

Waldbusser, "Remote Network Monitoring MIB Extensions
for Switched Networks Version 1.0", RFC 2613,
June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence,
S., Leach, P., Luotonen, A., and L. Stewart, "HTTP
Authentication: Basic and Digest Access

Connectivity", RFC 2678, September 1999.

Delay Metric for IPPM", RFC 2679, September 1999.

Packet Loss Metric for IPPM", RFC 2680, September 1999.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-
trip Delay Metric for IPPM", RFC 2681, September 1999.

[RFC2748] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan,
R., and A. Sastry, "The COPS (Common Open Policy


RFC 3411, December 2002.


[RFC5181] Shin, M-K., Han, Y-H., Kim, S-E., and D. Premec, "IPv6 Deployment Scenarios in 802.16 Networks", RFC 5181,
May 2008.


[RFC5717] Lengyel, B. and M. Bjorklund, "Partial Lock Remote
Procedure Call (RPC) for NETCONF", RFC 5717, December 2009.


BCP 158, RFC 6158, March 2011.


Appendix A.  High Level Classification of Management Protocols and Data Models

The following subsections aim to guide the reader for the fast selection of the management standard in interest and can be used as a dispatcher to forward to the appropriate chapter.  The subsections below classify the protocols on one hand according to high level criteria such as push versus pull mechanism, and passive versus active monitoring.  On the other hand the protocols are categorized concerning the network management task they address or the data model extensibility they provide.  Based on the reader’s requirements a reduced set of standard protocols and associated data models can be selected for further reading.

As an example, someone outside of IETF typically would look for the TWAMP protocol in the Operations and Management Area working groups as it addresses performance management.  However, the protocol TWAMP has been developed by the IPFM working group in the Transport Area.

Note that not all protocols have been listed in all classification sections.  Some of the protocols, especially the protocols with specific focus in Section 3 cannot be clearly classified.  Note also that COPS and COPS-PR are not listed in the tables, as COPS-PR is not recommended to use (see Section 3.3).
A.1. Protocols classified by the Standard Maturity at IETF

This section classifies the management protocols according their standard maturity at the IETF. The IETF standard maturity levels Proposed, Draft or Full Standard, are defined in [RFC2026]. IETF specifications must have "multiple, independent, and interoperable implementations" before they can be advanced from Proposed to Draft Standard status. An Internet or Full Standard (also referred as Standard) is characterized by a high degree of technical maturity and by a generally held belief that the specified protocol or service provides significant benefit to the Internet community.

The table below covers the standard maturity of the different protocols listed in this document. Note that only the main protocols (and not their extensions) are noted. An RFC search tool listing the current document status is available at [RFCSEARCH].

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Maturity Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP [STD62][RFC3411] (Section 2.1)</td>
<td>Full Standard</td>
</tr>
<tr>
<td>SYSLOG [RFC5424] (Section 2.2)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>IPFIX [RFC5101] (Section 2.3)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>PSAMP [RFC5476] (Section 2.3)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>NETCONF [RFC4741] (Section 2.4.1)</td>
<td>Full Standard</td>
</tr>
<tr>
<td>DHCP for IPv4 [RFC2131] (Section 3.1.1)</td>
<td>Draft Standard</td>
</tr>
<tr>
<td>DHCP for IPv6 [RFC3315] (Section 3.1.1)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>OWAMP [RFC4656] (Section 3.4)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>TWAMP [RFC5357] (Section 3.4)</td>
<td>Full Standard</td>
</tr>
<tr>
<td>RADIUS [RFC2865] (Section 3.5)</td>
<td>Draft Standard</td>
</tr>
<tr>
<td>DIAMETER [RFC3588] (Section 3.6)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>CAPWAP [RFC5416] (Section 3.7)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>ANCP [RFC6320] (Section 3.8)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>Ad-hoc network configuration [RFC5889] (Section 3.1.2)</td>
<td>Informational</td>
</tr>
<tr>
<td>ACAP [RFC2244] (Section 3.9)</td>
<td>Proposed Standard</td>
</tr>
<tr>
<td>XCAP [RFC4825] (Section 3.10)</td>
<td>Proposed Standard</td>
</tr>
</tbody>
</table>
This subsection classifies the management protocols matching to the management tasks for fault, configuration, accounting, performance, and security management.

<table>
<thead>
<tr>
<th>Fault Mgmt</th>
<th>Configuratio nMgmt</th>
<th>Accounting Mgmt</th>
<th>Performance Mgmt</th>
<th>Security Mgmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP</td>
<td>SNMP</td>
<td>SNMP</td>
<td>SNMP</td>
<td></td>
</tr>
<tr>
<td>notificatio n with trap operation (S. 2.1.1)</td>
<td>with configuration (S. 2.1.1)</td>
<td>with monitoring (S. 2.1.1)</td>
<td>with get operation (S. 2.1.1)</td>
<td></td>
</tr>
<tr>
<td>IPFIX (S. 2.3)</td>
<td>CAPWAP (S. 3.7)</td>
<td>NETCONF (S. 2.4)</td>
<td>RADIUS Accounting (S. 3.5)</td>
<td></td>
</tr>
<tr>
<td>PSAMP (S. 2.3)</td>
<td>ANCP (S. 3.8)</td>
<td>AUTOCONF (S. 3.1.2)</td>
<td>DIAMETER Accounting (S. 3.6)</td>
<td></td>
</tr>
<tr>
<td>SYSLOG (S. 2.2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RADIUS Authent.&amp; Authoriz. (S. 3.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DIAMETER Authent.&amp; Authoriz. (S. 3.6)</td>
</tr>
</tbody>
</table>

Table 2: Protocols Matched to Management Tasks

Note: Corresponding section numbers are given in parenthesis.

A.3. Push versus Pull Mechanism

A pull mechanism is characterized by the Network Management System (NMS) pulling the management information out of network elements, when needed. A push mechanism is characterized by the network elements pushing the management information to the NMS, either when the information is available, or on a regular basis.
Client/Server protocols, such as DHCP, ANCP, ACAP, and XCAP are not listed in Table 3.

<table>
<thead>
<tr>
<th>Protocols supporting the Pull mechanism</th>
<th>Protocols supporting the Push mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP (except notifications) (Section 2.1)</td>
<td>SNMP notifications (Section 2.1)</td>
</tr>
<tr>
<td>NETCONF (except notifications) (Section 2.4.1)</td>
<td>NETCONF notifications (Section 2.4.1)</td>
</tr>
<tr>
<td>CAPWAP (Section 3.7)</td>
<td>SYSLOG (Section 2.2)</td>
</tr>
<tr>
<td></td>
<td>IPFIX (Section 2.3)</td>
</tr>
<tr>
<td></td>
<td>PSAMP (Section 2.3)</td>
</tr>
<tr>
<td></td>
<td>RADIUS accounting (Section 3.5)</td>
</tr>
<tr>
<td></td>
<td>DIAMETER accounting (Section 3.6)</td>
</tr>
</tbody>
</table>

Table 3: Protocol classification by Push versus Pull Mechanism

A.4. Passive versus Active Monitoring

Monitoring can be divided into two categories, passive and active monitoring. Passive monitoring can perform the network traffic monitoring, monitoring of a device or the accounting of network resource consumption by users. Active monitoring, as used in this document, focuses mainly on active network monitoring and relies on the injection of specific traffic (also called "synthetic traffic"), which is then monitored. The monitoring focus is indicated in the table below as "network", "device" or "accounting".

This classification excludes non-monitoring protocols, such as configuration protocols: Ad-hoc network autoconfiguration, ANCP, and XCAP.
The application of SNMP to passive traffic monitoring (e.g. with RMON-MIB) or active monitoring (with IPPM MIB) depends on the MIB modules used. However, SNMP protocol itself does not have operations, which support active monitoring. NETCONF can be used for passive monitoring, e.g. with the NETCONF Monitoring YANG module [RFC6022] for the monitoring of the NETCONF protocol. CAPWAP monitors the status of a Wireless Termination Point.

RADIUS and DIAMETER are considered as passive monitoring protocols as they perform accounting, i.e. counting the number of packets/bytes for a specific user.

A.5. Supported Data Model Types and their Extensibility

The following table matches the protocols to the associated data model types. Furthermore, the table indicates how the data model can be extended based on the available content today and whether the protocol contains a built-in mechanism for proprietary extensions of the data model.
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Data Modeling</th>
<th>Approach to extend the Data Model</th>
<th>Proprietary Data Modeling Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP (Section 2.1)</td>
<td>MIB modules defined with SMI (Section 2.1.3)</td>
<td>New MIB modules specified in new RFCs</td>
<td>Enterprise specific MIB modules</td>
</tr>
<tr>
<td>SYSLOG (Section 2.2)</td>
<td>Structured Data Elements (SDE) (Section 4.1)</td>
<td>With the procedure to add Structured Data ID in [RFC5424]</td>
<td>Enterprise specific SDEs</td>
</tr>
<tr>
<td>IPFIX (Section 2.3)</td>
<td>IPFIX Information Elements, IPFIX IANA registry at [IANA-IPFIX] (Section 2.3)</td>
<td>Information Elements specified in [RFC5102]</td>
<td>Enterprise specific Information Elements</td>
</tr>
<tr>
<td>PSAMP (Section 2.3)</td>
<td>PSAMP Information Elements, PSAMP IANA registry at [IANA-PSAMP] (Section 2.3)</td>
<td>With the procedure to add Information Elements specified in [RFC5102]</td>
<td>Enterprise specific Information Elements</td>
</tr>
<tr>
<td>NETCONF (Section 2.4.1)</td>
<td>YANG modules (Section 2.4.2)</td>
<td>New YANG modules specified in new RFCs following the guideline in [RFC6087]</td>
<td>Enterprise specific YANG modules</td>
</tr>
<tr>
<td>IPPM OWAMP/TWAMP (Section 3.4)</td>
<td>IPPM metrics (*) (Section 3.4)</td>
<td>New IPPM metrics (Section 3.4)</td>
<td>Not applicable</td>
</tr>
<tr>
<td>RADIUS (Section 3.5)</td>
<td>Type-Length-Values (TLV)</td>
<td>RADIUS related registries at [IANA-AAA] and [IANA-PROT]</td>
<td>Vendor Specific Attributes (VSA)</td>
</tr>
</tbody>
</table>
Table 5: Data Models and their Extensibility

(*): With the publication of [RFC6248] the latest IANA registry for IPFIX metrics has been declared Obsolete.

Appendix B. New Work related to IETF Management Standards

B.1. Energy Management (EMAN)

Energy management is becoming an additional requirement for network management systems due to several factors including the rising and fluctuating energy costs, the increased awareness of the ecological impact of operating networks and devices, and the regulation of governments on energy consumption and production.

The basic objective of energy management is operating communication networks and other equipments with a minimal amount of energy while still providing sufficient performance to meet service level objectives. Today, most networking and network-attached devices neither monitor nor allow control energy usage as they are mainly instrumented for functions such as fault, configuration, accounting, performance, and security management. These devices are not instrumented to be aware of energy consumption. There are very few means specified in IETF documents for energy management, which includes the areas of power monitoring, energy monitoring, and power state control.

A particular difference between energy management and other management tasks is that in some cases energy consumption of a device is not measured at the device itself but reported by a different place. For example, at a Power over Ethernet (PoE) sourcing device or at a smart power strip, in which cases one device is effectively metering another remote device. This requires a clear definition of the relationship between the reporting devices and identification of remote devices for which monitoring information is provided. Similar considerations will apply to power state control of remote devices, for example, at a PoE sourcing device that switches on and off power at its ports. Another example scenario for energy management is a gateway to low resourced and lossy network devices in wireless a...
building network. Here the energy management system talks directly to the gateway but not necessarily to other devices in the building network.

At the time of this writing the EMAN working group works on the management of energy-aware devices, covered by the following items:

- **Requirements for energy management**, specifying energy management properties that will allow networks and devices to become energy aware. In addition to energy awareness requirements, the need for control functions will be discussed. Specifically the need to monitor and control properties of devices that are remote to the reporting device should be discussed.

- **Energy management framework**, which will describe extensions to current management framework, required for energy management. This includes: power and energy monitoring, power states, power state control, and potential power state transitions. The framework will focus on energy management for IP-based network equipment (routers, switches, PCs, IP cameras, phones and the like). Particularly, the relationships between reporting devices, remote devices, and monitoring probes (such as might be used in low-power and lossy networks) need to be elaborated. For the case of a device reporting on behalf of other devices and controlling those devices, the framework will address the issues of discovery and identification of remote devices.

- **Energy-aware Networks and Devices MIB document**, for monitoring energy-aware networks and devices, will address devices identification, context information, and potential relationship between reporting devices, remote devices, and monitoring probes.

- **Power and Energy Monitoring MIB document** will document defining managed objects for monitoring of power states and energy consumption/production. The monitoring of power states includes: retrieving power states, properties of power states, current power state, power state transitions, and power state statistics. The managed objects will provide means for reporting detailed properties of the actual energy rate (power) and of accumulated energy. Further, it will provide information on electrical power quality.

- **Battery MIB document** will define managed objects for battery monitoring, which will provide means for reporting detailed properties of the actual charge, age, and state of a battery and of battery statistics.
Applicability statement will describe the variety of applications that can use the energy framework and associated MIB modules. Potential examples are building networks, home energy gateway, etc. Finally, the document will also discuss relationships of the framework to other architectures and frameworks (such as Smart Grid). The applicability statement will explain the relationship between the work in this WG and the other existing standards such as those from the IEC, ANSI, DMTF, and others. Note that the EMAN WG will be looking into existing standards such as those from the IEC, ANSI, DMTF and others, and reuse existing work as much as possible.

Appendix C. Open issues

- Add a section or appendix for the high-level overview of IETF MIB modules in contrast to the overview of data models following the FCAPS-based view for management applications

Appendix D. Change Log

RFC EDITOR: Please remove this appendix before publication.

D.1. 01-02

- Resolved bugs, nits and open issues
- Reduced subsections RADIUS and DIAMETER with text on expired drafts.
- Extended dispatcher tables in Appendix A
- Added a note indicating that IETF has not developed so far specific technologies for the management of sensor networks.
- Added a note that IETF has not used the FCAPS view as an organizing principle for its data models.
- Added [I-D.weil-shared-transition-space-request] assuming that it'll get published pretty fast
- Added RFC references
- Removed text on expired drafts
D.2. 00-01

- Reduced text for the Security Requirements on SNMP and referenced to RFC 3411
- Reduced subsection on VACM
- Merged subsection on "RADIUS Authentication and Authorization with SNMP Transport Models" into the section "SNMP Transport Security Model"
- Section on Dynamic Host Configuration Protocol (DHCP) revised by Ralph Droms
- Subsections on DHCP and Autoconf assembled in section "IP Address Management"
- Removed subsection on "Extensible Provision Protocol (EPP)"
- Introduced new Appendix on "High Level Classification of Management Protocols and Data Models"
- Deleted detailed positive comments
- Resolved some of the I-D references with the correct reference to the published RFC number
- Added RFC references
- Removed text on expired drafts
- Resolved bugs, nits and open issues

D.3. draft-ersue-opsawg-management-fw-03-00

- Diverse bug fixing
- Incorporated comments from Juergen Schoenwaelder
- Reduced detailed text on pro and contra on management technologies
- Extended Terminology section with terms and abbreviations
- Explained the structure based on the management application view
- Definition of ‘MIB module’ aligned in different sections
o Text on SNMP security reduced

o All protocol sections discuss now security and AAA as far as relevant

o Added IPFIX IEs, SYSLOG SDEs and YANG modules to the data model definition

o Added text on YANG data modules to section 4.2.

o Added text on IPFIX IEs to section 4.3.

o Added numerous references


D.4.1. 02-03

o Rearranged the document structure using a flat structure putting all protocols onto the same level.

o Incorporated contributions for RADIUS/DIAMETER, IPFIX/PSAMP, YANG, and EMAN.

o Added diverse references.

o Added Contributors and Acknowledgements sections.

o Bug fixing and issue solving.

D.4.2. 01-02

o Added terminology section.

o Changed the language for neutral standard description addressing diverse SDOs.

o Extended NETCONF and NETMOD related text.

o Extended section for 'IPv6 Network Operations'.

o Bug fixing.

D.4.3. 00-01

o Extended text for SNMP
o Extended RADIUS and DIAMETER sections.

o Added references.

o Bug fixing.

Authors’ Addresses

Mehmet Ersue (editor)
Nokia Siemens Networks
St.-Martin-Strasse 53
Munich  81541
Germany

EMail: mehmet.ersue@nsn.com

Benoit Claise
Cisco Systems, Inc.
De Kleetlaan 6a b1
Diegem  1831
Belgium

EMail: bclaise@cisco.com
Load balancer is deployed widely in datacenter nowadays. There is a requirement to build a unique LB network management system where two or more vendors’ LB devices are used. We propose the standard MIBs for unique NMS.

Load balancer description is introduced at "http://en.wikipedia.org/wiki/Load_balancing_(computing)".

This memo defines an portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes a MIB module for load balance device.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 May 16, 2012.

Copyright Notice

Copyright (c) 2011 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
Table of Contents

1. Introduction .................................................. 3
2. The Internet-Standard Management Framework ................. 3
3. Conventions ..................................................... 3
4. Structure of Load-Balance MIB objects ........................ 3
5. Loadbalance-MIB Module Definitions ........................... 4
6. Security Considerations ........................................ 13
7. IANA Considerations ............................................ 14
8. Normative References .......................................... 14
Authors’ Addresses ................................................. 14
1. Introduction

Load balancer is deployed widely in datacenter nowadays. There is a requirement to build a unique LB network management system where two or more vendors’ LB devices are used. We propose the standard MIBs for unique NMS.

This document defines 5 MIB Modules which together support the configuration and monitoring of Load Balance device.

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, [RFC2578] STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. 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 RFC 2119 [RFC2119].

4. Structure of Load-Balance MIB objects

The following subsections describe the purpose of each of the objects contained in the loadbalance-MIB.

4.1. Load balance Virtual Service Table

Services provided by LB devices are virtual services. Configured on an LB device, a virtual service is uniquely identified by virtual service IP address, service protocol, service mode, and service port number. Access requests of users are sent to the LB device through a public or private network. If matching the virtual service, the requests are distributed to real services by the LB device.

4.2. Load balance Real Service Table
Services provided by real servers are real services. A real service can be a traditional FTP or HTTP service, and can also be a forwarding service in a generic sense. For example, a real service in firewall load balancing is the packet forwarding path.

4.3. Load balance Real Service Group Table

Server group----a real service group is a logical concept. Servers can be classified into different groups according to the common attributes of these servers. For example, servers can be classified into static storage server group and dynamic switching server group according to their functions; or they can be classified into music server group, video server group and picture server group according to the services they provide.

4.4. Load balance health checking Table

Health monitoring allows an LB device to check the statuses of real servers or links, collect the corresponding information, and quarantine the servers or links that work abnormally. Health monitoring can not only mark whether servers or links can work normally, but also can collect statistics of the response time of the servers or links for selecting servers or links.

4.5. Load balance Statistic Table

The statistic for Virtual Service or Real Service session, transmission rate.

5. Loadbalance-MIB Module Definitions

LOAD-BALANCER-MIB DEFINITIONS ::= BEGIN

IMPORTS
  MODULE-IDENTITY, OBJECT-TYPE, mib-2,
  Unsigned32, Integer32
  FROM SNMPv2-SMI                         -- RFC2578
  MODULE-COMPLIANCE, OBJECT-GROUP
  FROM SNMPv2-CONF                        -- RFC2580
;

lbMIB MODULE-IDENTITY
  LAST-UPDATED "201111310000Z"
  ORGANIZATION
    "IETF Operations and Management Area Working Group
    http://datatracker.ietf.org/wg/opsawg/
    CONTACT-INFO
"email: Li Chen (lichenyj@chinamobile.com) China Mobile"

DESCRIPTION
"MIB objects for load-balancing devices.

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

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

REVISION "201111310000Z"

::= { mib-2 XXX }

lbMIBNotifications OBJECT IDENTIFIER ::= { lbMIB 0 }
lbMIBObjects OBJECT IDENTIFIER ::= { lbMIB 1 }
lbMIBConformance OBJECT IDENTIFIER ::= { lbMIB 2 }

lbMIBCompliances OBJECT IDENTIFIER ::= { lbMIBConformance 1 }
lbMIBGroups OBJECT IDENTIFIER ::= { lbMIBConformance 2 }

-- Load-balancer Virtual Service table
--

lbVSTable OBJECT-TYPE
SYNTAX SEQUENCE OF LbVSEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Configured on an LB device, a virtual service is uniquely identified by virtual service IP address, service protocol, service mode, and service port number. Access requests of users are sent to the LB device through a public or private network. If matching the virtual service, the requests are distributed to real services by the LB device."

::= { lbMIBObjects 1 }

lbVSEntry OBJECT-TYPE
SYNTAX LbVSEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A row describing LB virtual service."
INDEX  { lbVSId }
::= { lbVSTable 1 }

LbVSEntry ::= SEQUENCE {
  lbVSId          Unsigned32,
  lbVSAddr        IpAddress,
  lbVSPort        INTEGER,
  lbVSmode        INTEGER,
  lbVSproto       INTEGER,
}

LbVSId          OBJECT-TYPE
SYNTAX      Unsigned32 (1..'ffffffff'H)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION   "LB virtual service identifier."
 ::= { lbVSEntry 1 }

lbVSAddr      OBJECT-TYPE
SYNTAX      IpAddress
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION   "Virtual service IP address of cluster/LB, used for users
to request services."
 ::= { lbVSEntry 2 }

lbVSPort        OBJECT-TYPE
SYNTAX      INTEGER (0..65535)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION   "The LB distributes the requests with the same source IP
address and source port
to a specific server."
 ::= { lbVSEntry 3 }

lbVSmode        OBJECT-TYPE
SYNTAX      INTEGER (NAT(0),DR(1))
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION   "Layer 4 server load balancing can be classified into
Network Address Translation (NAT)-mode server load
balancing and Direct routing (DR)-mode server load balancing."
 ::= { lbVSEntry 4 }

lbVSproto OBJECT-TYPE
SYNTAX INTEGER (TCP(0),UDP(1))
MAX-ACCESS read-write
STATUS current
DESCRIPTION "LB can support protocol for user."
 ::= { lbVSEntry 5 }

--
-- Load-balancer Real Service table
--

lbRSTable OBJECT-TYPE
SYNTAX SEQUENCE OF LbRSEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Services provided by real servers are real services. A real service can be a traditional FTP or HTTP service, and can also be a forwarding service in a generic sense. For example, a real service in firewall load balancing is the packet forwarding path."
 ::= { lbMIBObjects 2 }

LbRSEntry OBJECT-TYPE
SYNTAX LbRSEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A row describing LB real service."
INDEX { lbRSId }
 ::= { lbRSTable 1 }

LbRSEntry ::= SEQUENCE {
    lbRSId Unsigned32,
    lbRSGId Unsigned32,
    lbRSAAddr IpAddress,
    lbRSPort INTEGER,
}

lbRSId OBJECT-TYPE
SYNTAX Unsigned32 (1..'ffffff'f'H)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"LB real service identifier."
::= { lbRSEntry 1 }

1bRSGId OBJECT-TYPE
SYNTAX  Unsigned32 (1.'ffffffff'H)
MAX-ACCESS read-only
STATUS   current
DESCRIPTION
"a real service group is a logical concept. Servers
  can be classified into different groups according
to the common attributes of these servers."
::= { lbRSEntry 2 }

1bRSAddr OBJECT-TYPE
SYNTAX   IpAddress
MAX-ACCESS read-write
STATUS   current
DESCRIPTION
"IP address of a server, used by the LB device to
distribute requests."
::= { lbRSEntry 3 }

1bRSPort OBJECT-TYPE
SYNTAX   INTEGER (0..65535)
MAX-ACCESS read-write
STATUS   current
DESCRIPTION
"The LB uses the port for communication with server."
::= { lbRSEntry 4 }

--
-- Load-balancer Real Service Group table
--

1bRSGTable OBJECT-TYPE
SYNTAX   SEQUENCE OF LbRSGEntry
MAX-ACCESS not-accessible
STATUS   current
DESCRIPTION
"Real Server group is a logical concept. Servers can
  be classified into different groups according to the
  common attributes of these servers."
::= { lbMIBObjects 3 }

1bRSGEntry OBJECT-TYPE
SYNTAX   LbRSGEntry
MAX-ACCESS not-accessible
A row describing LB real service group.

INDEX { lbRSGId } ::= { lbRSGTable 1 }

LbRSGEntry ::= SEQUENCE {
    lbRSGId             Unsigned32,
    lbRSID              Unsigned32,
    lbRSGschdalgorithm  INTEGER,
    lbRSGhealth         INTEGER
}

lbRSGId OBJECT-TYPE
SYNTAX    Unsigned32 (1..'ffffffff'H)
MAX-ACCESS read-write
STATUS    current
DESCRIPTION "LB real service group identifier."
::= { lbRSGEntry 1 }

lbRSID OBJECT-TYPE
SYNTAX    Unsigned32 (1..'ffffffff'H)
MAX-ACCESS read only
STATUS    current
DESCRIPTION "LB real service identifier."
::= { lbRSGEntry 2 }

lbRSGschdalgorithm OBJECT-TYPE
SYNTAX    INTEGER(
    Round Robin(0),
    Weighted Round Robin(1),
    Random(2),
    Weighted Random(3),
    Source IP Hashing(4),
    Source IP and Source Port Hashing(5),
    Destination IP Hashing(6),
    UDP Packet Load Hashing(7),
    Least Connection(8),
    Weighted Least Connection(9),
    Bandwidth(10)
)
MAX-ACCESS read only
STATUS    current
DESCRIPTION "An LB needs to distribute service traffic to different
real services according to a load balancing scheduling algorithm."
::= { lbRSGEntry 3 }

lbRSGhealth OBJECT-TYPE
SYNTAX INTEGER(
      DNS(0),
      ICMP(1),
      HTTP(2)
    )
MAX-ACCESS read-write
STATUS  current
DESCRIPTION "The health monitoring method of RSG. It allows an LB device to detect whether real servers can provide services. The common method includes DNS\ICMP\HTTP, etc."
::= { lbRSGEntry 4 }

-- Load-balancer health monitoring table
--

lbHealthchkTable OBJECT-TYPE
SYNTAX SEQUENCE OF LbHealthchkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table contains information about the health check parameters, which include IP address, prot, health check type, health check interval, retry times."
::= { lbMIBObjects 4 }

LbHealthchkEntry OBJECT-TYPE
SYNTAX LbHealthchkEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "A row describing LB health check."
INDEX { lbHealthchkId }
::= { lbHealthchkTable 1 }

LbHealthchkEntry ::= SEQUENCE {
   lbHealthchkId             Unsigned32,
   lbHealthchkAddr           IpAddress,
   lbHealthchkPort           INTEGER,
   lbHealthchktype           INTEGER,
   lbHealthchkintvl          Integer32,

lbHealthchkRetrytimes  Integer32
}

lbHealthchkId  OBJECT-TYPE
SYNTAX       Unsigned32 (1..'ffffffff'H)
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION   "LB health check identifier."
::= { lbHealthchkEntry 1 }

lbHealthchkAddr  OBJECT-TYPE
SYNTAX        IpAddress
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION   "The remote IP address of server."
::= { lbHealthchkEntry 2 }

lbHealthchkPort  OBJECT-TYPE
SYNTAX        INTEGER (0..65535)
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION   "The remote port of server supporting service."
::= { lbHealthchkEntry 3 }

lbHealthchkType  OBJECT-TYPE
SYNTAX        INTEGER(ICMP(0),DNS(1),HTTP(2))
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION   "The set of health check method that include ICMP\DNS\HTTP, etc."
::= { lbHealthchkEntry 4 }

lbHealthchkIntvl  OBJECT-TYPE
SYNTAX        Integer32
MAX-ACCESS    read-write
STATUS        current
DESCRIPTION   "The definite length of between two packets. the packet can be ICMP\DNS\HTTP message."
::= { lbHealthchkEntry 5 }

lbHealthchkRetrytimes  OBJECT-TYPE
SYNTAX        Integer32
MAX-ACCESS    read-write
STATUS current
DESCRIPTION
"the LB will retry the defined times when server doesn’t reply
health check packet in time."
::= { lbHealthchkEntry 6 }

--
-- Statistic table
--

lbStaTable OBJECT-TYPE
SYNTAX SEQUENCE OF LbStaEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The statistic for Virtual Service or Real Service session,
transmission rate."
::= { lbMIBObjects 5 }

lbStaEntry OBJECT-TYPE
SYNTAX LbStaEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A row describing LB Statistic."
INDEX { lbStaId }
::= { lbStaTable 1 }

LbStaEntry ::= SEQUENCE {
    lbStaId             Unsigned32,
    lbStasession        INTEGER,
    lbStarate           INTEGER
}

lbStaId OBJECT-TYPE
SYNTAX Unsigned32 (1.'ffffffff'H)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"LB statistic table identifier."
::= { lbStaEntry 1 }

lbStasession OBJECT-TYPE
SYNTAX INTEGER32
MAX-ACCESS read only
lbStarate OBJECT-TYPE
SYNTAX INTEGER32
MAX-ACCESS read only
STATUS current
DESCRIPTION "the max or min flow rate of a RS or RSG."
::= { lbStaEntry 3 }

-- Conformance statements
--

lbMIBCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION "The compliance statement for SNMP engines that support
the LOAD-BALANCER-MIB."
MODULE
MANDATORY-GROUPS { lbMIBGroup }
::= { lbMIBCompliances 1 }

lbMIBGroup OBJECT-GROUP
OBJECTS {
   lbVSmode,
   lbRSGsichdalgorith,m,
   lbHealthchktype,
   lbStasession,
}
STATUS current
DESCRIPTION "A collection of objects for managing load-balancer."
::= { lbMIBGroups 1 }

END

6. Security Considerations

[TBD]
7. IANA Considerations

IANA is requested to assign a value for "XXX" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

8. Normative References


Authors' Addresses

Chen Li
China Mobile
Unit2, Dacheng Plaza, No. 28 Xuanwumenxi Ave, Xuanwu District
Beijing 100053
P.R. China

Email: lichenyj@chinamobile.com
Lianyuan Li
China Mobile
Unit2, Dacheng Plaza, No. 28 Xuanwumenxi Ave, Xuanwu District
Beijing 100053
P.R. China
Email: lilianyuan@chinamobile.com

Tina TSOU
Huawei
Email: Tina.Tsou.Zouting@huawei.com
Definitions of Managed Objects for Network Address Translators (NAT)
draft-perreault-opsawg-natmib-bis-00

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 configuration as well as monitoring of a device capable of NAT function. This memo is a revision of the previous NAT-MIB [RFC4008] to take into account new types of NAT.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 8, 2012.

Copyright Notice

Copyright (c) 2011 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
Table of Contents

1. Introduction ................................................. 3
2. Changes from RFC4008 ........................................ 3
3. The Internet-Standard Management Framework ...................... 4
4. Terminology ................................................... 4
5. Overview ..................................................... 5
   5.1. natInterfaceTable .................................... 6
   5.2. natAddrMapTable ....................................... 6
   5.3. Default Timeouts, Protocol Table, and Other Scalars ....... 7
   5.4. natAddrBindTable and natAddrPortBindTable ................ 7
   5.5. natSessionTable ....................................... 8
   5.6. RFC 3489 NAPT Variations, NAT Session and Bind Tables ...... 8
   5.7. Notifications ........................................... 9
   5.8. Notifications ........................................... 9
   5.9. Configuration via the MIB ................................ 10
   5.10. Relationship to Interface MIB ............................ 10
6. Definitions .................................................. 11
7. Acknowledgements ............................................. 68
8. Security Considerations ....................................... 68
9. IANA Considerations .......................................... 70
10. References ................................................ 70
    10.1. Normative References ................................ 70
    10.2. Informative References ............................... 71
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 configuration and monitoring of a device capable of NAT function. 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 3 provides references to the SNMP management framework, which was used as the basis for the MIB module definition. Section 4 describes the terms used throughout the document. Section 5 provides an overview of the key objects, their inter-relationship, and how the MIB module may be used to configure and monitor a NAT device. Lastly, Section 6 has the complete NAT MIB definition.

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. Changes from RFC4008

TODO: Move this section to an appendix after initial reviews.

- Address pools can now be shared between multiple interfaces. This change makes this MIB applicable to DS-Lite’s AFTR [RFC6333]. See [draft-schoenw-behave-nat-mib-bis-00] for rationale.

- TODO: Merge CGN stuff from draft-jpdionne-behave-cgn-mib.

- TODO: Merge NAT64 stuff from draft-jpdionne-behave-nat64-mib.

- TODO: Update to RFC 4787 terminology for describing NAT behavior.

- TODO: Support protocols other than UDP and TCP.

- TODO: Add support to limit and/or throttle binding allocations.

- TODO: Clarify existing notifications (e.g., natPacketDiscard) and add any additional notifications that may be needed for binding limits / binding throttling.

- TODO: Are we missing anything for PCP support? (time-limited static entries)

- TODO: Include (for example in an appendix) a description plus examples how the revised NAT-MIB can be used by NAT64 implementations, CGNs, and DS- Lite implementations.
3. 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 [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].

4. Terminology

[To be Reviewed]

Definitions for a majority of the terms used throughout the document may be found in [RFC2663]. Additional terms that further classify NAPT implementations are defined in [RFC3489]. Listed below are terms used in this document.

Address realm - An address realm is a realm of unique network addresses that are routable within the realm. For example, an enterprise address realm could be constituted of private IP addresses in the ranges specified in [RFC1918], which are routable within the enterprise, but not across the Internet. A public realm is constituted of globally unique network addresses.

Symmetric NAT - Symmetric NAT, as defined in [RFC3489], is a variation of Network Address Port Translator (NAPT). Symmetric NAT does not use port bind for translation across all sessions originating from the same private host. Instead, it assigns a new public port to each new session, irrespective of whether the new session used the same private end-point as before.

Bind or Binding - Several variations of the term 'Bind' (or 'Binding') are used throughout the document. Address Bind (or Address Binding) is a tuple of (Private IP address, Public IP Address) used for translating an IP address end-point in IP packets. Port Bind (or, Port Binding, or Address Port Bind, or Address Port Binding) is a tuple of (transport protocol, Private IP address, Private port, Public IP Address, Public port) used for translating a port end-point tuple of (transport protocol, IP address, port). Bind is used to refer to either Address Bind or Port Bind. Bind Mode identifies whether a bind is Address Bind or Port Bind.
NAT Session - A NAT session is an association between a session as seen in the private realm and a session as seen in the public realm, by virtue of NAT translation. If a session in the private realm were to be represented as (PrivateSrcAddr, PrivateDstAddr, TransportProtocol, PrivateSrcPort, PrivateDstPort) and the same session in the public realm were to be represented as (PublicSrcAddr, PublicDstAddr, TransportProtocol, PublicSrcPort, PublicDstPort), the NAT session will provide the translation glue between the two session representations. NAT sessions in the document are restricted to sessions based on TCP and UDP only. In the future, NAT sessions may be extended to be based on other transport protocols such as SCTP, UDP-lite and DCCP.

The terms 'local' and 'private' are used interchangeably throughout the document when referring to private networks, IP addresses, and ports. Likewise, the terms 'global' and 'public' are used interchangeably when referring to public networks, IP addresses, and ports.

5. Overview

NAT MIB is configurable on a per-interface basis and depends in several parts on the IF-MIB [RFC2863].

NAT MIB requires that an interface for which NAT is configured be connected to either a private or a public realm. The realm association of the interface plays an important role in the definition of address maps for the interface. An address map entry identifies the orientation of the session (inbound or outbound to the interface) for which the entry may be used for NAT translation. The address map entry also identifies the end-point of the session that must be subject to translation. An SNMP Textual-Convention 'NatTranslationEntity' is defined to capture this important characteristic that combines session orientation and applicable session endpoint for translation.

An address map may consist of static or dynamic entries. NAT creates static binds from a static address map entry. Each static bind has a direct one-to-one relationship with a static address map entry. NAT creates dynamic binds from a dynamic address map entry upon seeing the first packet of a new session.

The following subsections define the key objects used in NAT MIB, their inter-relationship, and how to configure a NAT device using the MIB module.
5.1. natInterfaceTable

[To be reviewed]

natInterfaceTable is defined in the MIB module to configure interface specific realm type and the NAT services enabled for the interface. natInterfaceTable is indexed by ifIndex and also includes interface specific NAT statistics.

The first step for an operator in configuring a NAT device is determining the interface over which NAT service is to be configured. When NAT service is operational, translated packets traverse the NAT device by ingressing on a private interface and egressing on a public interface or vice versa. An operator may configure the NAT service on either the public interface or the private interface in the traversal path.

As the next step, the operator must identify the NAT service(s) desired for the interface. The operator may configure one or more NAT services on the same interface. The MIB module identifies four types of NAT services: Basic NAT, NAPT, twice NAT and bidirectional NAT. These are NAT varieties as defined in [RFC2663]. Note that [RFC3489] further classifies NAPT implementations based on the behavior exhibited by the NAPT devices from different vendors. However, the MIB module does not explicitly distinguish between the NAPT implementations. NAPT implementations may be distinguished between one another by monitoring the BIND and NAT Session objects generated by the NAT device as described in section Section 5.6.

5.2. natAddrMapTable

[To be reviewed]

natAddrMapTable is defined in the MIB module to configure address maps on a per-interface basis. natAddrMapTable is indexed by the tuple of (ifIndex, natAddrMapIndex). The same table is also used to collect Statistics for the address map entries. Address maps are key to NAT configuration. An operator may configure one or more address map entries per interface. NAT looks up address map entries in the order in which they are defined to determine the translation function at the start of each new session traversing the interface. An address map may consist of static or dynamic entries. A static address map entry has a direct one-to-one relationship with binds. NAT will dynamically create binds from a dynamic address map entry.

The operator must be careful in selecting address map entries for an interface based on the interface realm-type and the type of NAT service desired. The operator can be amiss in the selection of
address map entries when not paying attention to the associated interface characteristics defined in natInterfaceTable (described in section 4.1). For example, say the operator wishes to configure a NAPT map entry on an interface of a NAT device. If the operator chooses to configure the NAPT map entry on a public interface (i.e., interface realm-type is public), the operator should set the TranslationEntity of the NAPT address map entry to be outboundSrcEndPoint. On the other hand, if the operator chooses to configure the NAPT map entry on a private interface (i.e., interface realm-type is private), the operator should set the TranslationEntity of the NAPT address map entry to be InboundSrcEndPoint.

5.3. Default Timeouts, Protocol Table, and Other Scalars

[To be reviewed]

DefTimeouts is defined in the MIB module to configure idle Bind timeout and IP protocol specific idle NAT session timeouts. The timeouts defined are global to the system and are not interface specific.

Protocol specific statistics are maintained in natProtocolTable, which is indexed by the protocol type.

The scalars natAddrBindNumberOfEntries and natAddrPortBindNumberOfEntries hold the number of entries that currently exist in the Address Bind and the Address Port Bind tables, respectively.

The generation of natPacketDiscard notifications can be configured by using the natNotifThrottlingInterval scalar MIB object.

5.4. natAddrBindTable and natAddrPortBindTable

[To be reviewed]

Two Bind tables, natAddrBindTable and natAddrPortBindTable, are defined to hold the bind entries. Entries are derived from the address map table and are not configurable. natAddrBindTable contains Address Binds, and natAddrPortBindTable contains Address Port Binds. natAddrBindTable is indexed by the tuple of (ifIndex, LocalAddrType, LocalAddr). natAddrPortBindTable is indexed by the tuple of (ifIndex, LocalAddrType, LocalAddr, LocalPort, Protocol). These tables also maintain bind specific statistics. A Symmetric NAT will have no entries in the Bind tables.
5.5.  natSessionTable

natSessionTable is defined to hold NAT session entries. NAT session entries are derived from NAT Binds (except in the case of Symmetric NAT) and are not configurable.

The NAT session provides the necessary translation glue between two session representations of the same end-to-end session; that is, a session as seen in the private realm and in the public realm. Session orientation (inbound or outbound) is determined from the orientation of the first packet traversing the NAT interface. Address map entries and bind entries on the interface determine whether a session is subject to NAT translation. One or both endpoints of a session may be subject to translation.

With the exception of symmetric NAT, all other NAT functions use endpoint specific bind to perform individual end-point translations. Multiple NAT sessions would use the same bind as long as they share the same endpoint. Symmetric NAT does not retain a consistent port bind across multiple sessions using the same endpoint. For this reason, the bind identifier for a NAT session in symmetric NAT is set to zero. natSessionTable is indexed by the tuple of (ifIndex, natSessionIndex). Statistics for NAT sessions are also maintained in the same table.

5.6.  RFC 3489 NAPT Variations, NAT Session and Bind Tables

[RFC3489] defines four variations of NAPT - Full Cone, Restricted Cone, Port Restricted Cone, and Symmetric NAT. These can be differentiated in the NAT MIB based on different values for the objects in the session and the bind tables, as indicated below.

In a Port Restricted Cone NAT, NAT Session objects will contain a non-zero PrivateSrcEPBindId object. Further, all address and port objects within a NAT session will have non-zero values (i.e., no wildcard matches).

An Address Restricted Cone NAT may have been implemented in the same way as a Port Restricted Cone NAT, except that the UDP NAT Sessions may use ANY match on PrivateDstPort and PublicDstPort objects; i.e., PrivateDstPort and PublicDstPort objects within a NAT session may be set to zero.

A Full Cone NAT may have also been implemented in the same way as a
Port Restricted Cone NAT, except that the UDP NAT Sessions may use
ANY match on PrivateDstAddr, PrivateDstPort, PublicDstAddr, and
PublicDstPort objects. Within a NAT Session, all four of these
objects may be set to zero. Alternately, all address and port
objects within a NAT Session may have non-zero values, yet the
TranslationEntity of the PrivateSrcEPBindId for the NAT Sessions may
be set bi-directionally, i.e., as a bit mask of (outboundSrcEndPoint
and inboundDstEndPoint) or (inboundSrcEndPoint and
outboundDstEndPoint), depending on the interface realm type. Lastly,
a Symmetric NAT does not maintain Port Bindings. As such, the NAT
Session objects will have the PrivateSrcEPBindId set to zero.

5.7. Notifications

[To be reviewed]

natPacketDiscard notifies the end user/manager of packets being
discarded due to lack of address mappings.

[Port exhaustion, CGN-MIB?]

5.8. Notifications

[To be reviewed]

The association between the various NAT tables can be represented as
follows:

```
          Interface
                         |    Address map
                         |    ------------------------------
                         |    |                              |
                         |    |                              |
                         |    |                              |
                         |    |                              |
                         |    | Address Bind                Port Bind
                         |    ------------------------------
                         |    |                              |
                         |    |    --------------------------
                         |    |                              |
                         |    |                              |
                         |    |    --------------------------
```

Perreault & Tsou          Expires March 8, 2012                 [Page 9]
NAT Session

All NAT functions, with the exception of Symmetric NAT, use Bind(s) to provide the glue necessary for a NAT Session. 
natSessionPrivateSrcEPBindId and natSessionPrivateDstEPBindId objects represent the endpoint Binds used by NAT Sessions.

5.9. Configuration via the MIB

[To be reviewed]

Section 5.1, and Section 5.2 and part of Section 5.3 refer to objects that are configurable on a NAT device. NAT derives Address Bind and Address Port Bind entries from the Address Map table. Hence, an Address Bind or an Address Port Bind entry must not exist without an associated entry in the Address Map table.

Further, NAT derives NAT session entries from NAT Binds, except in the case of symmetric NAT, which derives translation parameters for a NAT session directly from an address map entry. Hence, with the exception of Symmetric NAT, a NAT session entry must not exist in the NAT Session table without a corresponding bind.

A Management station may use the following steps to configure entries in the NAT-MIB:

- Create an entry in the natInterfaceTable specifying the value of ifIndex as the interface index of the interface on which NAT is being configured. Specify appropriate values, as applicable, for the other objects (e.g., natInterfaceRealm, natInterfaceServiceType) in the table (refer to Section 5.1).

- Create one or more address map entries sequentially in reduced order of priority in the natAddrMapTable, specifying the value of ifIndex to be the same for all entries. The ifIndex specified would be the same as that specified for natInterfaceTable (refer to Section 5.2).

- Configure the maximum permitted idle time duration for BINDs and TCP, UDP, and ICMP protocol sessions by setting the relevant scalars in natDefTimeouts object (refer to Section 5.3).

5.10. Relationship to Interface MIB

[To be reviewed, relationship to other MIB?]

The natInterfaceTable specifies the NAT configuration attributes on each interface. The concept of "interface" is as defined by
6. Definitions

This MIB module IMPORTs objects from [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC3411], and [RFC4001]. It also refers to information in [RFC0792], [RFC2463], and [RFC3413].

NAT-MIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY,
OBJECT-TYPE,
Integer32,
Unsigned32,
Gauge32,
Counter64,
TimeTicks,
mib-2,
NOTIFICATION-TYPE
FROM SNMPv2-SMI
TEXTUAL-CONVENTION,
StorageType,
RowStatus
FROM SNMPv2-TC
MODULE-COMPLIANCE,
NOTIFICATION-GROUP,
OBJECT-GROUP
FROM SNMPv2-CONF
ifIndex,
ifCounterDiscontinuityGroup
FROM IF-MIB
SnmpAdminString
FROM SNMP-FRAMEWORK-MIB
InetAddressType,
InetAddress,
InetPortNumber
FROM INET-ADDRESS-MIB;

natMIB MODULE-IDENTITY
LAST-UPDATED "YYYYMDDhmmmZ"
ORGANIZATION "IETF Transport Area"
CONTACT-INFO
"Simon Perreault
Viagenie
2875 boul. Laurier, suite D2-630
Quebec
Canada"
DESCRIPTION
"This MIB module defines the generic managed objects for NAT.

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

REVISION "200503210000Z" -- 21th March 2005

DESCRIPTION
"Initial version, published as RFC 4008."

REVISION "YYYYMMDDhhmmZ"

DESCRIPTION
"Second version, published as RFC XXXX."

::= { mib-2 123 }

natMIBObjects OBJECT IDENTIFIER ::= { natMIB 1 }

NatProtocolType ::= TEXTUAL-CONVENTION
STATUS current
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 current
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 current
DESCRIPTION "A unique id that is assigned to each address map by a NAT enabled device."
SYNTAX Unsigned32 (1..4294967295)

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

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

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

NatTranslationEntity ::= TEXTUAL-CONVENTION
 STATUS current
 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 current
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 current
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 current
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       current
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       current
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 { 60 }
 ::= { natDefTimeouts 4 }
system."
DEFVAL { 86400 }
::= { natDefTimeouts 5 }
natTcpDefNegTimeout OBJECT-TYPE
SYNTAX     Unsigned32  (1..4294967295)
UNITS      "seconds"
MAX-ACCESS read-write
STATUS     current
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      current
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      current
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      current
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,  

Perreault & Tsou          Expires March 8, 2012                [Page 18]
natInterfaceOutTranslates Counter64,
natInterfaceDiscards Counter64,
natInterfaceStorageType StorageType,
natInterfaceRowStatus RowStatus,
natInterfaceSharedAddrMapIndex NatSharedAddrMapId
}

natInterfaceRealm OBJECT-TYPE
SYNTAX INTEGER {
  private (1),
  public (2)
}
MAX-ACCESS read-create
STATUS current
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 current
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 current
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 current
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 current
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 current
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 current
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 }

natInterfaceSharedAddrMapIndex OBJECT-TYPE
SYNTAX NatSharedAddrMapId
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Link to a NatSharedAddrMapEntry. If NULL, it is expected that there exist at least one NatAddrMapEntry pointing to this interface entry."
::= { natInterfaceEntry 8 }

--
-- The Address Map Table
--

natAddrMapTable OBJECT-TYPE
SYNTAX SEQUENCE OF NatAddrMapEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table lists address map parameters for NAT."
::= { natMIBObjects 4 }

natAddrMapEntry OBJECT-TYPE
SYNTAX NatAddrMapEntry
MAX-ACCESS not-accessible
STATUS current
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                 NatAddrMapId,
  natAddrMapName                  SnmpAdminString,
  natAddrMapEntryType             NatAssociationType,
  natAddrMapTranslationEntity     NatTranslationEntity,
  natAddrMapLocalAddrType         InetAddressType,
  natAddrMapLocalAddrFrom         InetAddress,
  natAddrMapLocalAddrTo           InetAddress,
  natAddrMapLocalPortFrom         InetPortNumber,
  natAddrMapLocalPortTo           InetPortNumber,
  natAddrMapGlobalAddrType        InetAddressType,
  natAddrMapGlobalAddrFrom        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      current
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      current
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      current
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      current
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 NAPT 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 NAPT 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      current
DESCRIPTION  
  "This object specifies the address type used for
  natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo."
::= { natAddrMapEntry 5 }

natAddrMapLocalAddrFrom OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-create
STATUS      current
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      current
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      current
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      current
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      current
DESCRIPTION
"This object specifies the address type used for natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo."
::= { natAddrMapEntry 10 }

natAddrMapGlobalAddrFrom OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-create
STATUS      current
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 current
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 current
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 current
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."

Perreault & Tsou Expires March 8, 2012 [Page 26]
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      current
DESCRIPTION
"This object specifies a bitmap of protocol identifiers."
 ::= { natAddrMapEntry 15 }

natAddrMapInTranslates OBJECT-TYPE
SYNTAX     Counter64
MAX-ACCESS read-only
STATUS      current
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      current
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      current
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       current
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       current
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       current
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’.\n
Perreault & Tsou Expires March 8, 2012 [Page 28]
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 current
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 current
DESCRIPTION
"This table holds information about the currently active NAT BINDs."
::= { natMIBObjects 6 }

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

natAddrBindLocalAddr OBJECT-TYPE
SYNTAX        InetAddress
MAX-ACCESS    not-accessible
STATUS        current
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        current
DESCRIPTION   "This object specifies the address type used for
  natAddrBindGlobalAddr."
Internet-Draft                 NAT-MIB-bis                September 2011

::= { natAddrBindEntry 3 }

natAddrBindGlobalAddr OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-only
STATUS     current
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     current
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     current
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 }

Perreault & Tsou          Expires March 8, 2012                [Page 31]
natAddrBindType OBJECT-TYPE
SYNTAX NatAssociationType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates whether the bind is static or dynamic."
::= { natAddrBindEntry 7 }

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

natAddrBindMaxIdleTime OBJECT-TYPE
SYNTAX TimeTicks
MAX-ACCESS read-only
STATUS current
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 current
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     current
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     current
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     current
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     current
DESCRIPTION
 "This table holds information about the currently
 active NAPT BINDs."
 ::= { natMIBObjects 8 }

natAddrPortBindEntry OBJECT-TYPE
SYNTAX     NatAddrPortBindEntry
MAX-ACCESS not-accessible
STATUS     current
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,
natAddrPortBindLocalAddrType OBJECT-TYPE
SYNTAX      InetAddressType
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "This object specifies the address type used for
natAddrPortBindLocalAddr."
::= { natAddrPortBindEntry 1 }

natAddrPortBindLocalAddr OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS not-accessible
STATUS      current
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      current
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      current
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      current
DESCRIPTION
  "This object specifies the address type used for natAddrPortBindGlobalAddr." 
::= { natAddrPortBindEntry 5 }

natAddrPortBindGlobalAddr OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-only
STATUS      current
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      current
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 current
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 current
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 current
DESCRIPTION
"This object indicates whether the bind is static or
dynamic."
 ::= { natAddrPortBindEntry 10 }

natAddrPortBindMapIndex OBJECT-TYPE
SYNTAX NatAddrMapId
MAX-ACCESS read-only
STATUS current
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 current
DESCRIPTION
"Number of sessions currently using this BIND."
 ::= { natAddrPortBindEntry 12 }

natAddrPortBindMaxIdleTime OBJECT-TYPE
SYNTAX TimeTicks
MAX-ACCESS read-only
STATUS current
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     current
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     current
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     current
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
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** current

**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** current

**DESCRIPTION**
"The session ID for this NAT session."
::= { natSessionEntry 1 }

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

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

natSessionPrivateAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object specifies the address type used for
natSessionPrivateSrcAddr and natSessionPrivateDstAddr."
::= { natSessionEntry 10 }
natSessionPrivateSrcAddr OBJECT-TYPE
SYNTAX     InetAddress
MAX-ACCESS read-only
STATUS     current
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     current
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     current
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 current
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 current
DESCRIPTION
"This object specifies the address type used for
natSessionPublicSrcAddr and natSessionPublicDstAddr."
::= { natSessionEntry 15 }

natSessionPublicSrcAddr OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
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     current
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     current
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.

```plaintext
 ::= { natSessionEntry 18 }

natSessionPublicDstPort OBJECT-TYPE
SYNTAX     InetPortNumber
MAX-ACCESS read-only
STATUS     current
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     current
DESCRIPTION

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

 ::= { natSessionEntry 20 }

natSessionCurrentIdleTime OBJECT-TYPE
SYNTAX     TimeTicks
```
Internet-Draft                   NAT-MIB-bis            September 2011

MAX-ACCESS read-only  
STATUS     current  
DESCRIPTION    
"The time since a packet belonging to this session was last detected."  
::= { natSessionEntry 21 }

natSessionInTranslate OBJECT-TYPE  
SYNTAX     Counter64  
MAX-ACCESS read-only  
STATUS     current  
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 }

natSessionOutTranslate OBJECT-TYPE  
SYNTAX     Counter64  
MAX-ACCESS read-only  
STATUS     current  
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     current  
DESCRIPTION    
"The (conceptual) table containing per protocol NAT statistics."  
::= { natMIBObjects 10 }
natProtocolEntry OBJECT-TYPE
    SYNTAX     NatProtocolEntry
    MAX-ACCESS not-accessible
    STATUS     current
    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     current
    DESCRIPTION
        "This object represents the protocol pertaining to which
         parameters are reported."
 ::= { natProtocolEntry 1 }

natProtocolInTranslates OBJECT-TYPE
    SYNTAX     Counter64
    MAX-ACCESS read-only
    STATUS     current
    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     current
    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 current
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 }

--
-- The Shared Address Map Table
--

natSharedAddrMapTable OBJECT-TYPE
SYNTAX SEQUENCE OF NatSharedAddrMapEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table lists address map parameters for NAT."

::= { natMIBObjects 11 }

natSharedAddrMapEntry OBJECT-TYPE
SYNTAX NatSharedAddrMapEntry
MAX-ACCESS not-accessible
STATUS current
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 { natSharedAddrMapIndex }

::= { natSharedAddrMapTable 1 }

NatSharedAddrMapEntry ::= SEQUENCE {
natSharedAddrMapIndex OBJECT-TYPE
SYNTAX   NatSharedAddrMapId
MAX-ACCESS not-accessible
STATUS   current
DESCRIPTION
 Along with ifIndex, this object uniquely identifies an entry in the natAddrMapTable. Address map entries are applied in the order specified by natAddrMapIndex.
 ::= { natSharedAddrMapEntry 1 }

natSharedAddrMapName OBJECT-TYPE
SYNTAX   SnmpAdminString (SIZE(1..32))
MAX-ACCESS read-create
STATUS   current
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."
 ::= { natSharedAddrMapEntry 2 }

natSharedAddrMapEntryType OBJECT-TYPE
SYNTAX NatAssociationType
MAX-ACCESS read-create
STATUS   current
DESCRIPTION

"This parameter can be used to set up static or dynamic address maps."
::= { natSharedAddrMapEntry 3 }

natSharedAddrMapTranslatEntity OBJECT-TYPE
SYNTAX NatTranslationEntity
MAX-ACCESS read-create
STATUS current
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 NAPT 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 NAPT 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."
::= { natSharedAddrMapEntry 4 }

natSharedAddrMapLocalAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"This object specifies the address type used for natAddrMapLocalAddrFrom and natAddrMapLocalAddrTo."
::= { natSharedAddrMapEntry 5 }

natSharedAddrMapLocalAddrFrom OBJECT-TYPE
SYNTAX      InetAddress  
MAX-ACCESS  read-create  
STATUS      current  
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."
 ::= { natSharedAddrMapEntry 6 }

natSharedAddrMapLocalAddrTo OBJECT-TYPE
SYNTAX      InetAddress  
MAX-ACCESS  read-create  
STATUS      current  
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."
 ::= { natSharedAddrMapEntry 7 }

natSharedAddrMapLocalPortFrom OBJECT-TYPE
SYNTAX      InetPortNumber  
MAX-ACCESS  read-create  
STATUS      current  
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."
natSharedAddrMapLocalPortTo OBJECT-TYPE
SYNTAX      InetPortNumber
MAX-ACCESS  read-create
STATUS      current
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 }
::= { natSharedAddrMapEntry 9 }

natSharedAddrMapGlobalAddrType OBJECT-TYPE
SYNTAX      InetAddressType
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
  "This object specifies the address type used for
  natAddrMapGlobalAddrFrom and natAddrMapGlobalAddrTo."
::= { natSharedAddrMapEntry 10 }

natSharedAddrMapGlobalAddrFrom OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-create
STATUS      current
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."
::= { natSharedAddrMapEntry 11 }

natSharedAddrMapGlobalAddrTo OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-create
STATUS      current
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.

::= { natSharedAddrMapEntry 12 }

natSharedAddrMapGlobalPortFrom OBJECT-TYPE
SYNTAX      InetPortNumber
MAX-ACCESS  read-create
STATUS      current
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 }
::= { natSharedAddrMapEntry 13 }

natSharedAddrMapGlobalPortTo OBJECT-TYPE
SYNTAX      InetPortNumber
MAX-ACCESS  read-create
STATUS      current
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 }
::= { natSharedAddrMapEntry 14 }

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

natSharedAddrMapInTranslates OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
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."
::= { natSharedAddrMapEntry 16 }

natSharedAddrMapOutTranslates OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
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."
::= { natSharedAddrMapEntry 17 }

natSharedAddrMapDiscards OBJECT-TYPE
SYNTAX Counter64
MAX-ACCESS read-only
STATUS current
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.

```plaintext
::= { natSharedAddrMapEntry 18 }
```

**natSharedAddrMapAddrUsed OBJECT-TYPE**

- **SYNTAX**: Gauge32
- **MAX-ACCESS**: read-only
- **STATUS**: current
- **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.

```plaintext
::= { natSharedAddrMapEntry 19 }
```

**natSharedAddrMapStorageType OBJECT-TYPE**

- **SYNTAX**: StorageType
- **MAX-ACCESS**: read-create
- **STATUS**: current
- **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 }

```plaintext
::= { natSharedAddrMapEntry 20 }
```

**natSharedAddrMapRowStatus OBJECT-TYPE**

- **SYNTAX**: RowStatus
- **MAX-ACCESS**: read-create
- **STATUS**: current
- **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
Internet-Draft                  NAT-MIB-bis                September 2011

while the value of this object is active(1)."
REFERENCE
"Textual Conventions for SMIv2, Section 2."
::= { natSharedAddrMapEntry 21 }

--
-- Notifications section
--

natMIBNotifications OBJECT IDENTIFIER ::= { natMIB 0 }

--
-- Notifications
--

natPacketDiscard NOTIFICATION-TYPE
  OBJECTS { ifIndex }
  STATUS current
  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,

Perreault & Tsou          Expires March 8, 2012                [Page 57]
natAddrMapEntryType,
natAddrMapTranslationEntity,
natAddrMapLocalAddrType,
natAddrMapLocalAddrFrom,
natAddrMapLocalAddrTo,
natAddrMapLocalPortFrom,
natAddrMapLocalPortTo,
natAddrMapGlobalAddrType,
natAddrMapGlobalAddrFrom,
natAddrMapGlobalAddrTo,
natAddrMapGlobalPortFrom,
natAddrMapGlobalPortTo,
natAddrMapProtocol,
natAddrMapStorageType,
natAddrMapRowStatus,
natSharedAddrMapName,
natSharedAddrMapEntryType,
natSharedAddrMapTranslationEntity,
natSharedAddrMapLocalAddrType,
natSharedAddrMapLocalAddrFrom,
natSharedAddrMapLocalAddrTo,
natSharedAddrMapLocalPortFrom,
natSharedAddrMapLocalPortTo,
natSharedAddrMapGlobalAddrType,
natSharedAddrMapGlobalAddrFrom,
natSharedAddrMapGlobalAddrTo,
natSharedAddrMapGlobalPortFrom,
natSharedAddrMapGlobalPortTo,
natSharedAddrMapProtocol,
natSharedAddrMapStorageType,
natSharedAddrMapRowStatus,
natBindDefIdleTimeout,
natUdpDefIdleTimeout,
natIcmpDefIdleTimeout,
natOtherDefIdleTimeout,
natTcpDefIdleTimeout,
natTcpDefNegTimeout,
natNotifThrottlingInterval }

STATUS current
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  current

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 current

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 current

DESCRIPTION
"A collection of protocol specific NAT statistics, to aid troubleshooting/monitoring of NAT operation."

::= { natMIBGroups 4 }

natStatsAddrMapGroup OBJECT-GROUP


STATUS current

DESCRIPTION
"A collection of address map specific NAT statistics, to aid troubleshooting/monitoring of NAT operation."

::= { natMIBGroups 5 }

natMIBNotificationGroup NOTIFICATION-GROUP

NOTIFICATIONS { natPacketDiscard }

STATUS current

DESCRIPTION
"A collection of notifications generated by devices supporting this MIB."

::= { natMIBGroups 6 }
-- Compliance statements
--

natMIBFullCompliance MODULE-COMPLIANCE
  STATUS current
  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 current
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

7. Acknowledgements

The authors would like to thank R. Rohit, P. Srisuresh, Rajiv Raghunarayan, Nalinksh Pai, and Cliff Wang, the original authors of [RFC4008], as well as the following individuals who have participated in the drafting, review, and discussion of this memo:

Cathy Zhou, Juergen Schoenwaelder, Marc Blanchet, and Yu Fu.

8. Security Considerations

[To be reviewed, note about large number of mappings/bindings]

It is clear that this MIB can potentially be useful for configuration. Unauthorized access to the write-able objects could cause a denial of service and/or widespread network disturbance.
Hence, the support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

At this writing, no security holes have been identified beyond those that SNMP Security is itself intended to address. These relate primarily to controlled access to sensitive information and the ability to configure a device — or which might result from operator error, which is beyond the scope of any security architecture.

There are a number of managed objects in this MIB that may contain information that may be sensitive from a business perspective, in that they may represent NAT bind and session information. The NAT bind and session objects reveal the identity of private hosts that are engaged in a session with external end nodes. A curious outsider could monitor these two objects to assess the number of private hosts being supported by the NAT device. Further, a disgruntled former employee of an enterprise could use the NAT bind and session information to break into specific private hosts by intercepting the existing sessions or originating new sessions into the host. There are no objects that are sensitive in their own right, such as passwords or monetary amounts. It may even be important to control GET access to these objects and possibly to encrypt the values of these objects when they are sent over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

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

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

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.
9. IANA Considerations

TBD

10. References

10.1. Normative References


10.2. Informative References


Authors' Addresses

Simon Perreault
Viagenie
2875 boul. Laurier, suite D2-630
Quebec
Canada

Phone: +1-418-656-9254
EMail: simon.perreault@viagenie.ca
Tina Tsou
Huawei Technologies
2330 Central Expressway
Santa Clara
USA

Phone: +1-408-330-4424
EMail: tena@huawei.com
Virtual Subnet: A Scalable Data Center Interconnection Solution

draft-xu-virtual-subnet-06

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 27, 2011.

Copyright Notice

Copyright (c) 2009 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.
Abstract

This document proposes a host route based IP-only L2VPN solution called Virtual Subnet, which reuses BGP/MPLS IP VPN [RFC4364] and ARP proxy [RFC925][RFC1027] technologies. Virtual Subnet provides a much scalable approach for interconnecting geographically dispersed data centers.

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 RFC-2119 [RFC2119].

Table of Contents

1. Introduction ................................................ 3
2. Terminology ................................................. 3
3. Solution Description......................................... 4
   3.1. Unicast ................................................ 4
      3.1.1. Intra-subnet Unicast............................... 4
      3.1.2. Inter-subnet Unicast............................... 5
   3.2. Multicast/Broadcast..................................... 6
   3.3. CE Host Discovery....................................... 7
   3.4. CE Multi-homing......................................... 7
   3.5. CE Host Mobility........................................ 8
   3.6. ARP Proxy .............................................. 8
4. Comparison with VPLS......................................... 8
5. Future work ................................................ 10
6. Security Considerations..................................... 10
7. IANA Considerations ........................................ 10
8. Acknowledgements ........................................... 10
9. References ................................................ 10
   9.1. Normative References................................... 10
   9.2. Informative References................................ 10
Authors’ Addresses ............................................ 11
1. Introduction

To achieve service agility to the full extent of current virtual machine (VM) technology, cloud data center operators are demanding solutions for VM mobility across data centers of geographically dispersed locations. In this challenging environment, a solution that enables fast, reliable, high-capacity and highly scalable data center interconnection is essential. Virtual Private LAN Service (VPLS) [RFC4761, RFC4762] seems as an available technology for such demand. However, those scaling issues (e.g., ARP broadcast storm, unknown unicast flooding, etc.) that exit within the large Layer2 Ethernet bridge network would badly impact the network performance when such a flat Layer2 network is extended across multiple data centers.

This document describes a host route based IP-only L2VPN solution called Virtual Subnet (VS), which reuses BGP/MPLS IP VPN [RFC4364] and ARP proxy [RFC925][RFC1027] technologies. VS provides a much scalable approach for interconnecting geographically dispersed data centers. In contrast with existing VPLS solutions, VS alleviates the broadcast storm impact on the network performance to a great extent by partitioning the otherwise whole ARP broadcast and unknown unicast flooding domain associated with an IP subnet that has been extended across the MPLS/IP backbone, into multiple isolated parts per data center location. Besides, VS could provide many other desirable benefits that VPLS could never support. For example, the MAC table capacity pressure that the large amount of CE switches within data centers would have to face is greatly reduced. In addition, active-active data center exit capability could be achieved easily even in the case where path symmetry is required. Finally, the ARP table pressure on data center exit gateways could be reduced by several orders of magnitude.

Note that non-IP traffic would not be supported in VS since VS just provides an IP-only L2VPN service.

2. Terminology

This memo makes use of the terms defined in [RFC4364], [MVPN], [RFC2236] and [RFC2131].
3. Solution Description

3.1. Unicast

3.1.1. Intra-subnet Unicast

As shown in Figure 1, CE hosts dispersed across different VPN sites of a given IP-only L2VPN instance are actually within a single IP subnet (e.g., 10.0.0.0/8). PE routers automatically discover their locally connected CE hosts by some approaches such as ARP learning or ICMP PING and accordingly create host routes for their locally connected CE hosts. These host routes are distributed across PE routers with the existing BGP/MPLS IP VPN signaling. In addition, to avoid forwarding those packets destined for nonexistent hosts within the scope of their configured VPN subnet mistakenly according to the default route, PE routers each are configured with a null route for that VPN subnet. Meanwhile, APR proxy is enabled on the VRF interfaces of each PE router, thus, upon receiving from a local CE host an ARP request for a known remote CE host, the ingress PE router would return its own MAC address as a response.

![Figure 1: Intra-subnet Unicast]
Assume host A sends an ARP request for host B before communicating with host B, upon the receipt of this ARP request, ingress PE, PE-1, lookups the associated VRF table to find the corresponding host route for host B. If found and the route is learnt from a remote PE router, PE-1 acting as an ARP proxy, returns its own MAC address as a response to the above ARP request. Otherwise, PE-1 doesn’t need to respond to that ARP request. Once receiving the above ARP reply from PE-1, host A would send out an IP packet destined for B with the destination MAC address of PE-1’s MAC address which has been learnt through the above ARP resolution. One this packet arrives at PE-1, PE-1 would tunnel it towards the egress PE router (i.e., PE-2), which in turn forwards the packet to the destination CE host (i.e., host B).

### 3.1.2. Inter-subnet Unicast

As shown in Figure 2, for a CE host (e.g., host A) to communicate with other hosts outside its own subnet, a PE router (e.g., PE-2) which is connected to a CE gateway router (e.g., GW) would be configured with a default route with the next-hop pointing to that CE gateway router, and this default route would be distributed to other PE routers.
Now host A sends an ARP request for its default gateway (i.e., GW) before communicating with a destination host outside its subnet. Upon receiving this ARP request, PE-1 acting as an ARP proxy returns its own MAC address as a response in accordance with the rules described in the above section. Host A then sends out an IP packet for that destination host with destination MAC address of PE-1’s MAC. Upon receiving the above packet, PE-1 tunnels it towards PE-2 according to the default route that is learnt from PE-2. PE-2 in turn forwards the packet to GW according to the configured default route.

For the CE gateway router redundancy purpose, more than one CE gateway router could be connected to a given VPN subnet. In this case, Virtual Router Redundancy Protocol (VRRP) [RFC2338] could be optionally enabled among these CE gateway routers, in this way, only the PE router which is connected to the VRRP master is entitled to announce a default route. To achieve that goal, the next-hop of the default route SHOULD be set to the corresponding Virtual Router IP address, and the default route SHOULD not be deemed as valid unless there is a directly connected host route for the next-hop address. Due to the fact that only the VRRP master is entitled to respond to ARP requests for the corresponding Virtual Router IP address and broadcast gratuitous ARP requests or replies on behalf of the Virtual Router, only the PE router which is connected to the VRRP master could have an ARP entry corresponding to the Virtual Router IP address and therefore could have a directly connected host route for the Virtual Router IP address. In this way, packets destined for the outside of a given VPN subnet would be exactly sent to the corresponding VRRP master. Alternatively, PE routers could intercept the VRRP messages received from their locally connected CE routers and prevent them from flooding across the MPLS/IP backbone. As a result, each CE router will act as a VRRP master and therefore each PE router connected to the CE routers would announce a default route. In this way, inbound and outbound traffic of the VPN subnet would be load-balanced across multiple CE gateway routers and route optimization for the above traffic is achieved simultaneously.

3.2. Multicast/Broadcast

The MVPN technology [MVPN], in particular, the Protocol-Independent-Multicast (PIM) tree option with some extensions, could be reused here to support IP multicast and broadcast between CE hosts of the same VPN instance. For example, PE routers attached to a given VPN join a default provider multicast distribution tree which is dedicated for that VPN. Ingress PE routers, upon receiving customer
multicast or broadcast traffic from their local CE hosts, tunnel such customer traffic towards remote PE routers of the same VPN over the corresponding default provider multicast distribution tree. When receiving customer multicast or broadcast traffic over a provider multicast distribution tree, egress PE routers forward such customer traffic via the corresponding VRF interfaces.

More details about how to support multicast and broadcast in VS will be explored in a later version of this document.

3.3. CE Host Discovery

When receiving an ARP request or reply from a local CE host, PE router SHOULD cache or update the corresponding ARP entry for that CE host. In addition, PE router SHOULD periodically send ARP requests to those discovered local CE hosts (better in unicast) so as to keep the ARP entries fresh.

To ensure a PE router to discover all of its locally connected CE hosts in time, this PE router SHOULD perform the IP or ARP scan on its attached VPN site at least once when rebooting up. One possible option is to use the ICMP echo approach for host discovery. For example, a PE router could send out an ICMP echo request to an IP broadcast address (e.g., 10.255.255.255), every CE host receiving that ICMP echo request would respond with an ICMP echo reply which contains its IP and MAC addresses. Thus the PE router could discover all of its local CE hosts by inspecting the received ICMP echo replies. If the PE router couldn’t be able to process so many replies in a short period of time, the otherwise whole subnet could be partitioned into multiple segments and the corresponding host discovery for each segment could be performed in turn.

3.4. CE Multi-homing

For PE router redundancy purpose, a VPN site could be connected to more than one PE router. In this case, VRRP SHOULD run among these PE routers and only the PE router which is the VRRP master could respond to the ARP requests from local CE hosts and it MUST use the Virtual Router MAC address in any ARP packet it sends. To achieve active-active multi-homing for inbound traffic to a given multi-homed VPN site, those PE routers being VRRP slave could also perform the host discovery function and accordingly advertise host routes for local CE hosts. Note that there is no any contravention to the VRRP specification [RFC2338].
3.5. CE Host Mobility

Once a CE host moves from one VPN site to another, it will usually send out a gratuitous ARP request or reply when attaching to a new VPN site. The PE router attached to the new VPN site will create a CE host route upon receiving that gratuitous ARP message and then advertise it to remote PE routers.

When the PE router attached to the old VPN site receives a host route announcement for one of its local CE hosts from a remote PE router, it SHOULD immediately send an ARP request or ICMP echo for that CE host to determine whether or not that CE host is still locally connected to it. If no corresponding reply is returned in a given period of time, the PE router would delete the ARP entry of that CE host and accordingly withdraw the corresponding host route. Meanwhile, the PE router would broadcast a gratuitous ARP on behalf of that CE host, with the sender hardware address field being filled with its own MAC addresses. As a result, the ARP entry for that CE host that is cached on other local CE hosts of that old VPN site would be refreshed timely.

3.6. ARP Proxy

A PE router, acting as an ARP proxy, SHOULD only respond to ARP requests for those CE hosts which are exactly attached to other PE routers. In other words, the PE router SHOULD not respond to ARP requests for its local CE hosts or those nonexistent CE hosts.

When VRRP is configured on multiple PE routers which are attached to a given VPN site for redundancy purpose, only the PE router which is the VRRP master is entitled to perform the ARP proxy function.

4. Comparison with VPLS

Since VPLS simply extends a LAN across multiple sites and it operates as an Ethernet bridge, most scaling issues (e.g., ARP broadcast storm, unknown unicast flooding, etc.) that exist within a large Ethernet bridge network are not addressed by VPLS. In VS, by partitioning the otherwise whole ARP broadcast and unknown unicast flooding domain associated with a given subnet, which has been extended across the MPLS/IP backbone, into multiple isolated parts, the broadcast storm impact on network performance is alleviated to a great extent. For example, ARP broadcast traffic is limited within the scope of a VPN site. Similarly, unknown unicast traffic would not be flooded across the MPLS/IP backbone as well.
As for the MAC table capacity requirement on CE switches, CE switches in VPLS would have to learn MAC addresses of both local CE hosts and remote CE hosts. In contrast, CE switches in VS only needs to learn MAC addresses of local CE hosts and local PE routers due to the usage of ARP proxy.

Active-active DC exit is a much desirable capability when considering route/path optimization for traffic routing to/from the outside of geographically dispersed data centers (e.g., the Internet). In normal cases, each DC site will be connected to a default gateway (i.e., DC exit router) which is responsible for forwarding traffic routing to/from the outside. However, since these default gateways are within a single subnet due to the layer2 DCI usage, normally there is only one default gateway router (acting as VRRP master) is allowed to forward traffic routing to/from the outside. This is obviously not optimal from the perspective of WAN bandwidth utilization. Active-active VRRP approach has been proposed in the above case so that the traffic destined for the outside could be forwarded by the local DC exit gateways. This is workable when path symmetry is not required. However, in most cases where firewall or NAT devices are deployed at the DC exits, path symmetry is a must. As a result, active-active VRRP is not available anymore in such cases. In contrast, if VS is used as a DCI solution, when incoming traffic from the Internet enters a DC, source IP addresses of the traffic could be NATed on the DC exit gateway. Notes that DC exit gateways of geographically dispersed DCs are configured with different IP address pools without any overlapping for source NAT. In addition, the corresponding routes for the above NAT address pools are advertised by the DC exit gateways to their own connected PE routers of the VS respectively. Thus, when the outgoing traffic destined for the Internet arrives at its local PE router, that PE router would forward the traffic according to the matching routes for the above address pools. In this way, active-active DC exit can be achieved easily even in the case where path symmetry is required.

Another obvious advantage of VS over VPLS, as a DCI solution, is to reduce the ARP table size on DC gateways by several orders of magnitude. Assume there are millions of CE hosts within a single VLAN/subnet, if VPLS is used as a DCI solution, DC exit gateways would have to know millions of ARP entries corresponding to these CE hosts. In contrast, with VS as a DCI solution, DC exit gateways are directly connected to the PE routers of the VS which act as ARP proxies, MAC addresses of those ARP entries for CE hosts on DC gateways are identical (i.e., the PE router’s MAC). Thus these millions of ARP entries can be aggregated into one entry (e.g., 10.0.0.0/8->the PE router’s MAC). That’s to say, the exact-matching algorithm for ARP cache lookup is changed to the longest-matching
algorithm. Of course, there is no free lunch. The side-effect of this change is that DC exit gateways could send out packets destined for non-existing CE hosts to their connected PE routers of the VS. Fortunately, once those packets arrive at the PE router, that PE router in turn will drop those packets directly since there is no matching route for them.

5. Future work

How to support IPv6 CE hosts in VS is for future study.

6. Security Considerations

TBD.

7. IANA Considerations

There is no requirement for IANA.

8. Acknowledgements

Thanks to Dino Farinacci, Himanshu Shah, Nabil Bitar and Giles Heron for their valuable comments on this document.

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Xiaohu Xu
Huawei Technologies,
No.3 Xinxi Rd., Shang-Di Information Industry Base,
Hai-Dian District, Beijing 100085, P.R. China
Phone: +86 10 82882573
Email: xuxh@huawei.com