



## Current Status of IPv6 Management

Isabelle Astic, Olivier Festor

### ► To cite this version:

Isabelle Astic, Olivier Festor. Current Status of IPv6 Management. [Technical Report] RT-0274, INRIA. 2002, pp.32. inria-00069903

**HAL Id: inria-00069903**

**<https://inria.hal.science/inria-00069903>**

Submitted on 19 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

## *Current Status of IPv6 Management*

Isabelle ASTIC — Olivier FESTOR

**N° 0274**

December 2002

THÈME 1

 *apport  
technique*





## Current Status of IPv6 Management

Isabelle ASTIC , Olivier FESTOR

Thème 1 — Réseaux et systèmes  
Projet MADYNES

Rapport technique n° 0274 — December 2002 — 32 pages

**Abstract:** Nowadays, we are at an important time of the development of the IPv6 (Internet Protocol version 6) network. Until now, this protocol was used only between IPv6 “islands”. These “islands” were connected by IPv4 (Internet Protocol version 4) networks. So the transport of IPv6 packets was made through IPv4 ones, or into MPLS (MultiProtocol Label Switching) tunnels. Now, that real IPv6 native equipment appears, it is possible to create real native IPv6 networks.

Because in this new kind of networks, IPv4 does not exist, it defines a new challenge for IPv6 network management, because, until now, most of the IPv6 management architecture were defined upon IPv4, with SNMP (Simple Network Management Protocol) over IPv4 asking for the few MIBs (Management Information Bases) able to manage IPv6 networks. Now that IPv6 networks become IPv6 protocol only, how can we manage them ?

The purpose of this report is to provide a state of the art description of what can be used to manage IPv6 native networks and what is really used.

**Key-words:** IPv6, network management

## Où en sommes-nous de la supervision des réseaux IPv6 ?

**Résumé :** Nous sommes actuellement à un tournant important du développement des réseaux IPv6 (Internet Protocol version 6). Jusqu'à présent, ces réseaux étaient réduits à des "îlots" IPv6 reliés par des réseaux IPv4 (Internet Protocol version 4), ou par des tunnels MPLS (MultiProtocol Label Switching). Maintenant que de plus en plus d'équipements disposent de piles IPv6 natives, il est possible de créer des réseaux IPv6 natifs de larges dimensions.

La disparition du protocole IPv4 dans ce type de réseau conduit à un véritable défi en ce qui concerne leur administration. En effet, jusqu'à présent, les réseaux IPv6 étaient administrés à l'aide du protocole SNMP (Simple Network Management Protocol) au dessus d'IPv4 pour aller chercher les rares MIBs (Management Information Bases) capables de retourner des informations concernant le bon fonctionnement du réseau IPv6. Maintenant que les réseaux ne disposent plus que d'une seule pile, la pile IPv6, comment peut-on les administrer ?

Le but de ce rapport est de faire le point sur ce qui est disponible pour administrer de tels réseaux, mais aussi sur ce qui est réellement utilisé.

**Mots-clés :** IPv6, supervision de réseaux, administration de réseaux

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>I</b>	<b>Porting IPv4 management middlewares upon IPv6</b>	<b>7</b>
<b>2</b>	<b>From IPv4 towards IPv6</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	How to port IPv4 management middlewares toward IPv6 ? . . . . .	7
2.2.1	The standard dependent middlewares . . . . .	7
2.2.2	Standard independent middlewares . . . . .	8
2.2.3	Conclusion . . . . .	9
<b>3</b>	<b>The IPv4 management standards evolution toward IPv6</b>	<b>9</b>
3.1	The SNMP standard evolution . . . . .	9
3.1.1	The textual conventions evolution . . . . .	11
3.1.2	The MIBs evolution . . . . .	11
3.1.3	Conclusion . . . . .	14
3.2	The other standards evolution . . . . .	15
3.2.1	The Policy-based standards . . . . .	15
3.2.2	The Web-based standards . . . . .	17
3.2.3	The Authentication standards . . . . .	18
3.3	Conclusion . . . . .	20
<b>II</b>	<b>How to manage IPv6 networks ?</b>	<b>21</b>
<b>4</b>	<b>IPv6 management middlewares</b>	<b>21</b>
4.1	The CLI-based management middleware . . . . .	21
4.2	The XML-based management middlewares . . . . .	22
4.3	The middlewares using protocols . . . . .	23
<b>5</b>	<b>Some examples</b>	<b>23</b>
5.1	For fault management . . . . .	23
5.2	For services . . . . .	24
5.2.1	MPing . . . . .	24
5.2.2	Analyser and Ethereal . . . . .	24
5.2.3	Multicast beacon . . . . .	24
5.2.4	RRDtool . . . . .	24
5.3	For configuration management . . . . .	25
5.3.1	Configuring an interface . . . . .	25
5.3.2	Archiving the configuration. . . . .	26

5.4	Topology discovery tools . . . . .	26
5.4.1	For backbones . . . . .	27
5.4.2	For LANs . . . . .	27
5.4.3	For multicast networks . . . . .	27
<b>III</b>	<b>Conclusion</b>	<b>28</b>

## 1 Introduction

IPv6 was initially defined to solve the lack of IP (*Internet Protocol*) addresses. The increasing request of Internet access leads to saturation of the IPv4 address space. This saturation was bigger into Asia and Africa where the pool of allowed IPv4 addresses was very small.

If IPv6 was first a research subject, it now becomes really operational, because most of the network equipments are IPv6 enabled. Thus, large IPv6 native networks could be deployed in order to test their deployment phase and to acquire knowledge about their operation.

As government of Japan decided that the year 2005 will be the deadline of the complete switching of its country from IPv4 to IPv6<sup>1</sup>, and as in Europe, the deadline seems to be around 2010<sup>2</sup>, Asian and European governments fund projects to learn a lot about IPv6 native networks.

The 6net<sup>3</sup> project, in which INRIA<sup>4</sup> participates, is one of those European projects. Its goal is to interconnect most of the european NRENs (*National Research and Education Networks*) with an IPv6 native backbone, in order to have a real and practical knowledge of a deployment of such a network at the scale of a continent.

But, to have a complete knowledge of the IPv6 native network behaviour, and to solve them, a network management architecture together with a platform is required. Some could say that as IPv6 is not really recent, so, it should already exist some management solutions. But it should be remembered that until now, IPv6 networks were most of the time IPv6 native islands connected by IPv4 networks. That means that most IPv6 networks could be managed using IPv4 underlying protocols. Now that most of the network infrastructure is IPv6 native, network managers need new architecture and tools in order to manage them. This is the main problem that the Working Package 6 (WP6) of the 6net project, devoted to IPv6 network management and in which the Madynes<sup>5</sup> Research Group of the LORIA<sup>6</sup> participates, has to solve. Its firsts answers were to define an IPv6 network management architecture ([EFTW02]) and to list all middlewares that could be useful ([AAB<sup>+</sup>02]). It also edited a *cookbook*, in order to help the new IPv6 managers to build their IPv6 network management platform ([ACD<sup>+</sup>02]).

This article is a synthesis of the last two reports. In them, the question that the WP6 answered was : what kind of tools could we used or defined to manage IPv6 networks ? This question seems to have a trivial answer: as IPv6 is defined upon the same architecture than IPv4, IPv4 management services or middlewares ported towards IPv6 could certainly be used to manage IPv6 networks. In the first part, we explain why this answer is not so trivial. We first explain the problems encountered to port the IPv4 management middlewares towards IPv6. We, then describe the evolution of the IPv4 management standards towards

---

<sup>1</sup>in article "Slow Road to IPv6", network Computing, february 4, 2002

<sup>2</sup>in "Les Enjeux du déploiement du protocole IPv6", study made by IDATE for the RNRT (*Réseau National de recherche en Telecommunication*), June 2002, [http://www.telecom.gouv.fr/rnrt/index\\_net.htm](http://www.telecom.gouv.fr/rnrt/index_net.htm)

<sup>3</sup><http://www.6net.org>

<sup>4</sup><http://www.inria.fr>

<sup>5</sup><http://www.madynes.org>

<sup>6</sup><http://www.loria.fr>



IPv6. The second part of this document explains how we can currently manage IPv6 native networks and will list the tools or services that are available to manage them. We will then conclude and explain what and what must be the further research items in that domain.

## Part I

# Porting IPv4 management middlewares upon IPv6

## 2 From IPv4 towards IPv6

### 2.1 Introduction

To know what should be changed to move applications from IPv4 towards IPv6, we have to know what is the main difference between IPv4 and IPv6. This difference is an architectural one and concerns the structure of the IP address.

IPv6 addresses are more complex than the IPv4 ones. As written into [HD98], an IPv6 address is defined on 128 bits, split into a prefix and an identifiant. [HOD97] defines that the prefix is 64 bits long as well as the interface Id. It specifies also that an address could be of 3 types : unicast, multicast and anycast. A unicast address defines a unique interface. A multicast address defines a group of interfaces where each member of the group will receive any message sent to this address. An anycast address defines also a group but only one interface of the group will receive a message sent to this address.

Unicast or multicast addresses could be global, i.e. that they could be used to address any interface all over the world, or scoped, that means that this kind of address could only be used to talk to an interface into a specific domain : for example, a link (link-local address), a site (site-local address), etc.. Each scoped address is defined with a specific prefix. For example, [HD98] specifies that the prefix attributed to the unicast link-local address is FE80::/64.

That is why, in order to verify that IPv4 management tools could be used to managed IPv6 networks, we should verify that they are able to manage IPv6 addresses.

### 2.2 How to port IPv4 management middlewares toward IPv6 ?

IPv4 management middlewares could be defined upon management standards or could be independent of them.

#### 2.2.1 The standard dependent middlewares

In a general way, management standards could be represented as a client/server architecture. The client requests management information from the server, using a specific protocol. What is standardized is the protocol itself, the way the management repository is defined, the objects that can be found into it and the way these objects are defined.

For example, the SNMP (*Simple Network Management Protocol*) standard, is defined by the SNMP protocol (RFC (*Request For Comments*) 1157 [CFSD90]), the Structure of Management Information (currently the SMIPv2, RFC 2578 [MPS99a]), the several MIBs already

defined (see section 7, p 10) and its textual conventions (RFC 2579 [MPS99b]).

To be able to manage IPv6 networks, those kind of middlewares should use IPv6 enabled standards and should be themselves IPv6 compliant. So to port them towards IPv6, first should be port the implementation of the standard itself (see figure 1).

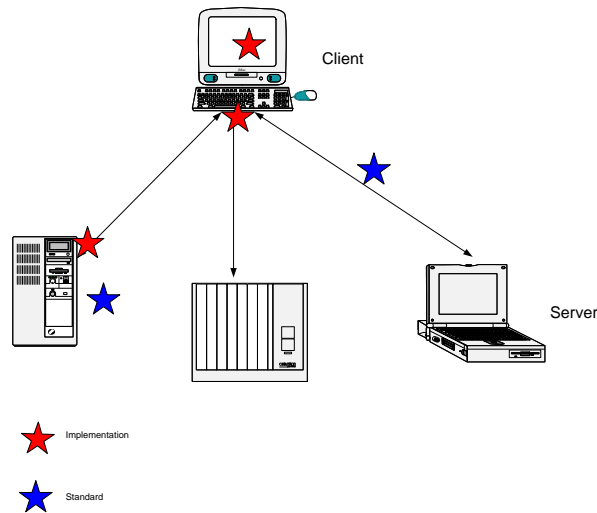


Figure 1: IPv4 management middlewares defined upon standards

As we have seen, network management standards define the protocol and the structure of the information that could be used to manage networks. So they will be available upon IPv6 if both of them are available on IPv6.

The modification that should be made to port the middleware itself are its interaction :

- with the user (i.e., for example, the way the data are displayed to the manager),
- and with the implementation of the standard.

### 2.2.2 Standard independent middlewares

Management standard independent middlewares rely only on standard APIs (*Application Programming Interface*) which connect them to the IP networks.

Here, work seems to be simpler as there is no standard. But it is not always the case. At least, the API must be modified to ask for IPv6 connectivity instead of IPv4 one. But a complete checking of the software is sometimes necessary to verify that there is no implicit

interaction with the IP architecture

In a study made at LORIA, some standard independent middlewares are implicitly dependant of the IP architecture. The topic of this study ([Ast02]) was the IPv4 automatic topology discovery tools. It shew that those services, defined into all the IPv4 main management platforms, are implicitly dependent of the size of the Interface Identifier of the IPv4 addresses: in order to discover all the interfaces used in each subnet of the network, those services make an exhaustive ping to all the possible IPv4 addresses contained into each subnet. Depending of the subnet class, the number of bits devoted to the interface ID is from 8 to 24. That means that the exhaustive search implies a maximum of  $2^{24}$  pings. But in IPv6, the addressing architecture defines an interface ID of 64 bits length. That means that  $2^{64}$  pings are now always needed to verify all the possible addresses of an IPv6 sub-network. So, this solution can not be used further.

That is why the LORIA proposed a new solution defined upon a distributed and hierarchical architecture and relying only upon the neighbor discovery and ICMPv6 protocol.

### 2.2.3 Conclusion

This section shew that each standard independent tool needs to be verified, tool by tool, in order to be sure that they can be used upon IPv6. As there is no general way to port them upon IPv6, we will not discussed about them further in this document. But, it should be noticed that some of theses works will be done by the 6net WP6, for the most interesting tools that was listed into the [AAB<sup>+</sup>02] document.

## 3 The IPv4 management standards evolution toward IPv6

In the following sections, we make a point on which standard is available upon IPv6 and which is not. As SNMP was the main IPv4 management architecture used, we will first study it before seeing where we are with the others.

### 3.1 The SNMP standard evolution

The SNMP set of standards (among them [CFSD90], [RM91], [MR91], [McC96a], [McC96b], [McC96c], [Bak97]) provides a framework for the definition of management information and a protocol for the exchange of this information. The rapid growth of networks and the increasing diversity of the systems over the last decade implied the need for a comprehensive management of this whole infrastructure. SNMP proved to be a good solution adopted by the IETF<sup>7</sup> (*Internet Engineering Task Force*). So, the SNMP protocol and the MIBs became the two key elements of the Internet Protocol Management Framework.

---

<sup>7</sup><http://www.ietf.org>

As the SNMP protocol was independent of the IP protocol architecture, its evolution to IPv6 was not a real problem. The first implementation of the SNMP protocol over IPv6 has been made in the net-snmp OpenSource project and appeared in the net-snmp-5.0.3 version, in May 2002. It seems to be quite late but as it was said, until now, most of the IPv6 networks were IPv4 networks too, so they used to be managed with SNMP over IPv4. Thus, until now, there was not a great need of SNMP over IPv6.

The evolution of the MIBs is more complicate, as they contain information needed to manage network equipments. Several MIBs exist for IPv4 networks, one for each type of equipment that could be connect to the network. For example, a Bridge MIB exist to manage bridges, another one was defined by Cisco<sup>8</sup> to manage its routers. The more important one is the MIB defining the information needed to manage the IP set of protocols, which is called MIB II (cf [McC96a], [McC96b], [McC96c], [Bak97]). Its evolution is explained into the Figure 2.

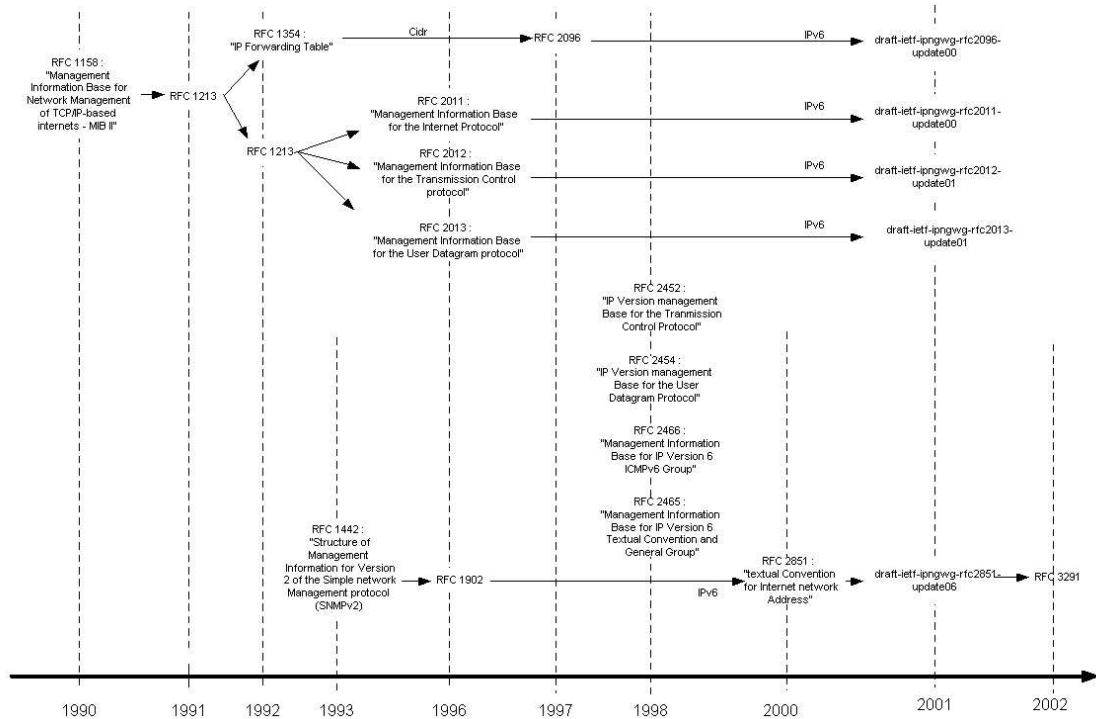


Figure 2: Evolution of the RFC to manage IPv6 networks

<sup>8</sup><http://www.cisco.com>

Since the initial IPv6 protocol specifications, in 1995, the definition of a MIB II able to manage IPv6 networks changed twice, one in 1998, the second in 2000. The main problem was to define the IP address type, that is, its textual convention.

### 3.1.1 The textual conventions evolution

The textual conventions evolution is illustrated into the Figure 3.

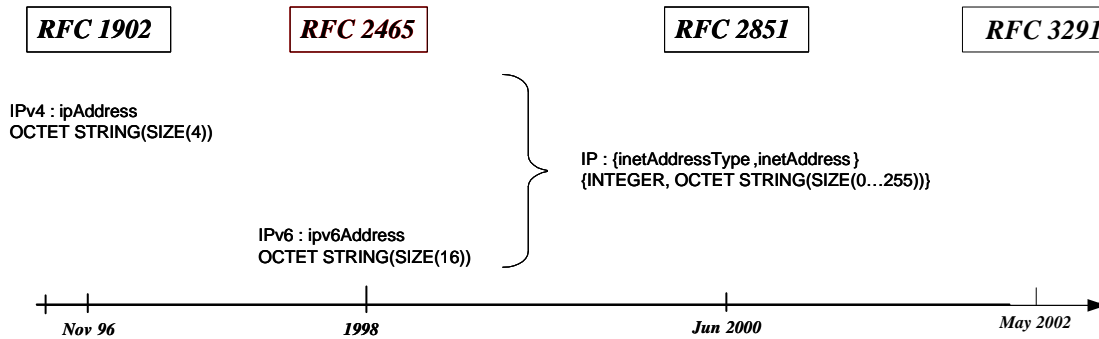


Figure 3: Evolution of the IP address textual convention

The first textual convention defining IP address representation is found into the RFC 1902, in 1996 ([CMRW96]). It defined an ASN.1 (*Abstract Syntax Notation number One*) type, called *IpAddress*, as an OCTET STRING(SIZE(4)). That means that it could be defined only like a string of 4 bytes long.

As IPv6 addresses are 128 bits long, 16 bytes are needed to save them. That is why the first textual convention defining IPv6 addresses was defined as an OCTET STRING(SIZE(16)), called *Ipv6Address* in the RFC 2465, in 1998 ([HO98]). But this approach implies the partition of IPv4 and IPv6 management. So, IETF decided to define an unified MIB II, able to manage both IPv4 and IPv6 networks, which resulted into new textual conventions, defined in 2000, into RFC 2851 ([DHRS00]). This RFC defines an IP address as a structure {*inetAddressType*, *inetAddress*}, where *inetAddressType* is an INTEGER which specifies if the following address is, for example, an IPv4 or IPv6 one. The *inetAddress* is defined as an OCTET STRING(SIZE(0...255)), in order to be able to save the value of an IPv4 or IPv6 address, as well as the value of a DNS name (cf [DHRS00]). The textual conventions for scoped IPv4 and IPv6 addresses are defined into the RFC 3291 ([DHRS02]). In this RFC appeared too the *InetAddressPrefixLength*, the *InetAddressPortNumber*, the *InetAddressAutonomousSystemNumber*.

### 3.1.2 The MIBs evolution

The definition of this textual convention implies associated modifications of the MIB II.

**The several RFC involved in this evolution.** The MIB II was defined in 1990 (RFC 1158, [Ros90]) in order to manage IPv4 networks. It was updated in 1991 with the RFC 1213 ([MR91], and then split in 1992 into two RFCs, the RFC 1354 ([Bak92], which defined the forwarding table, and the RFC 1213. It was split again in 1996 into the RFC 2096, RFC 2011, RFC 2012, RFC 2013. When the first textual convention for IPv6 addresses appeared in 1998, then other RFCs were defined to manage TCP and UDP over IPv6, ICMPv6 and IPv6 itself. They were RFC 2452, RFC 2454, RFC 2465, RFC 2466. (see the Figure 2).

After the definition of this unified textual convention, the “old” RFCs were updated by the drafts draft-ietf-ipngwg-rfc2011-update-00.txt, draft-ietf-ipngwg-rfc2012-update-01.txt, draft-ietf-ipngwg-rfc2013-update-01.txt, draft-ietf-ipngwg-rfc2096-update-00.txt ([FHST01b], [FHKS01], [FHK<sup>+</sup>01] and [FHST01a]), which are the current available drafts.

**The MIBs modifications** If we take the MIB objects and tables point of view, the definition of these multiple MIBs had the following impact:

- In 1996, the RFC 2011, 2012 and 2013 defined 3 groups: *ip*, *tcp* and *udp* (see the Figure 4).

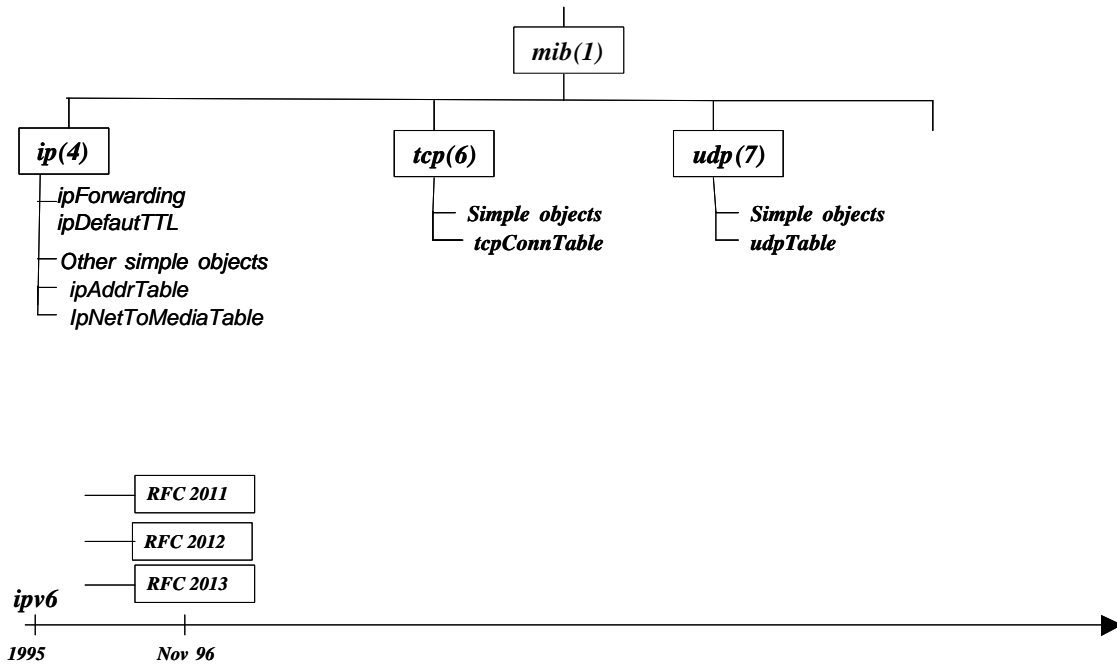


Figure 4: The MIB II in 1996

Each group contains simple objects and tables.

- In 1998, another group was defined, called *ipv6*, containing simple objects and tables, sometimes saving the same information than for IPv4 networks but into an IPv6 context. For examples, the *ipv6IfTable* contains must of the simple objects defined into the *ip(4)* group, the *ipv6AddrTable* or the *ipv6TcpConnTable* had the same purpose than, respectively, the *ipAddrTable* or the *tcpConnTable* (Figure 5).

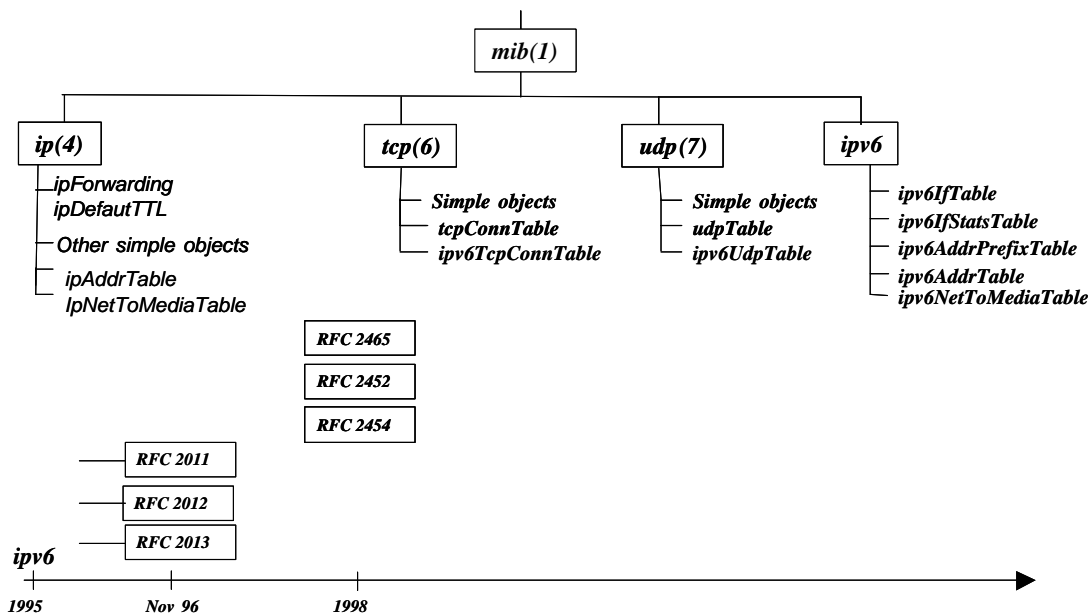


Figure 5: The definition of the separated IPv6 MIB

Further the fact that, by this approach, IPv6 was supposed to be another protocol, there was the risk to save wrong information into the wrong table, like IPv4 data into IPv6 table and conversely.

- The unified approach, synthetized into the Figure 6, unified all the tables : *ipAddrTable* and *ipv6AddrTable* became *ipAddressTable*, *ipNetToMediaTable* and *ipv6NetToMediaTable* became *inetNetToMediaTable*, all the simple objects defined into the IPv4 MIB and the *ipv6IfTable* became the *ipIfStatsTable*. The same with *ipv6TcpConnTable* and *tcpConnTable* which became *tcpConnectionTable*, and with *udpTable* and *ipv6UdpTable*, which became *udpListenerTable*. It must be noticed that, in addition to this table,



issued from the IPv4 management architecture, new tables were defined, specific of strictly IPv6 management, like the *ipv6InterfaceTable*, which help to manage the multiple IPv6 addresses of an interface, and the *ipv6ScopeIdTable*, which help to manage the IPv6 scoped addresses.

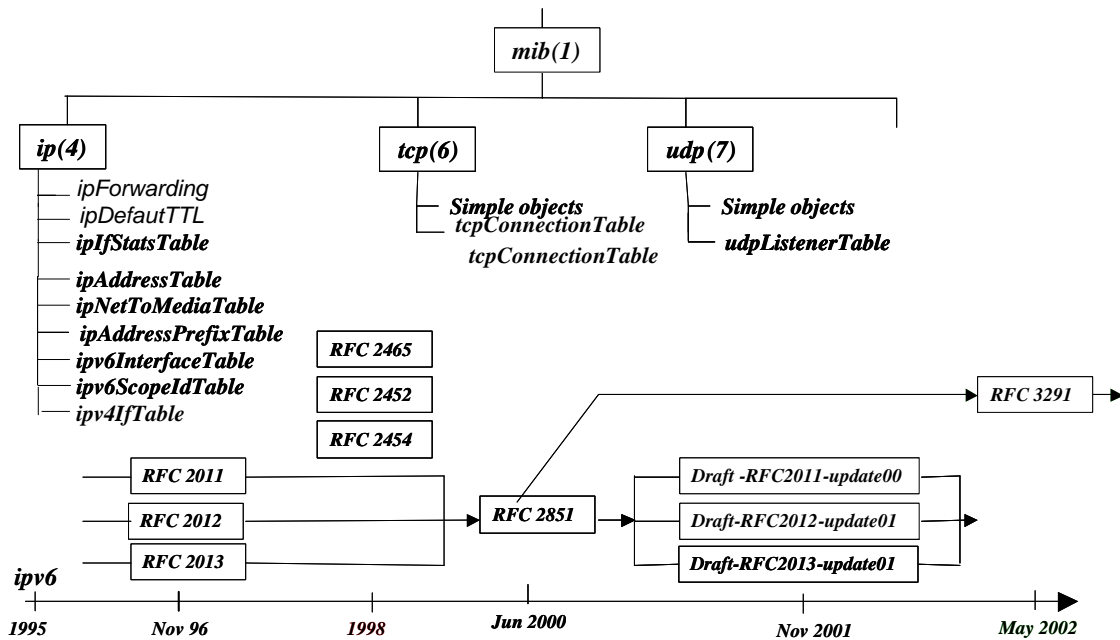


Figure 6: The unified MIB II

### 3.1.3 Conclusion

Despite or because of all these modifications, the SNMP standard is not fully available for the moment. As the MIB II is not fixed yet, very few implementations of it exist. Juniper<sup>9</sup> made the first effort by implementing the 1998's MIB on its routers. Cisco and the LORIA made the effort too, in 2002, Cisco by implementing the unified MIB into its 12000 series of routers, LORIA, by implementing it into the net-snmp package on FreeBSD 4.5 (cf [AJAA02]). A consequence of the actual state of the MIB II, as it is illustrated by the few existing implementations, is that if you want to manage IPv6 equipments using SNMP, you should define 2 interfaces, one to request the 1998 MIB, the second to request

<sup>9</sup><http://www.juniper.net>

the unified one. Another consequence is that the main management platform like InfoVista, HP OpenView or Tivoli are not available upon IPv6.

## 3.2 The other standards evolution

### 3.2.1 The Policy-based standards

**Presentation** If we take the definition exposed into the *IEEE network magazine*<sup>10</sup> dedicated to the *Policy-Based Networking* (cf [CLW02]), a policy is “a persistent specification of an objective to be achieved or a set of actions to be performed in the future or as an on-going regular activity”. The policy-based management is the application of those policies in the domain of the network management, using automated network operations. In a general way, a policy-based management architecture is defined as follows (see Figure 7):

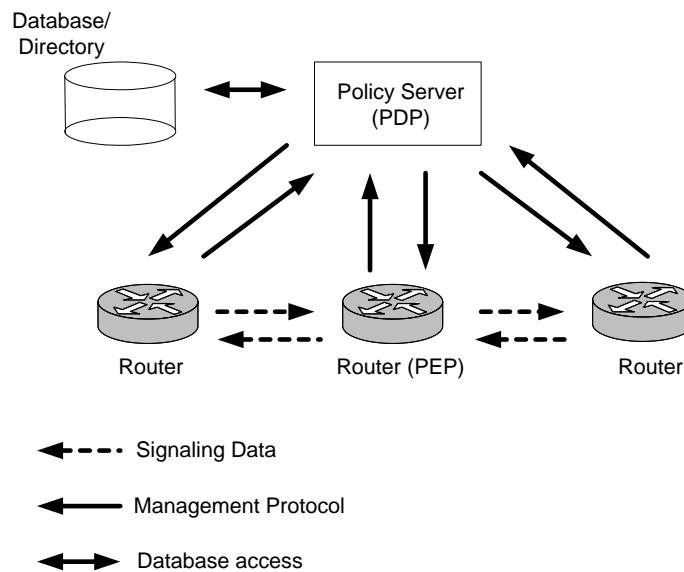


Figure 7: The policy-based management architecture

When the PEP (*Policy Enforced Point*) needs a decision to be taken (because it discovers a lack of bandwidth or a congestion for example), it requests the PDP (*Policy Decision Point*) about it. The PDP reads the request from the PEP, asks the repository for the policy that has to be taken, and sends its reply to the PEP. The PEP then applies this new policy. A

<sup>10</sup> [www.comsoc.org](http://www.comsoc.org)

new policy decision need is appreciated by the PEP in regards of the signaling data send by the other routers or by its own alert flags.

Several approaches already exists. A proposal, made by the *SNMPconf*<sup>11</sup> Working Group of IETF is based on the SNMP set, and uses the SNMP protocol to configure systems. The draft [MPST02] describes the advantages of such an approach and details the modification that should be made into the MIBs design and into the agents implementations. It shows that some care should be taken with the transaction integrity for example, and that a new notion, called template, should be defined to be able to configure several instances of a MIB object using a unique SNMP SET PDU. The draft [WSH02] describes the new objects that should be defined to SNMP policy-based configuration. All these objects are related to new notions that have not be used yet into MIBs, that are *roles*, *capabilities* and *time*. *Roles* are managed object characteristics and allow to know if a politic could be apply to the system or not. The *capabilities* give the system possibilities and determinate if a policy could be download on it or not. The third draft defined by this Working Group is “The differentiated Services Configuration MIB” [HP01]). It is a middle level MIB that make the link between the high level politics and the Differentiated Services MIB already defined, used to manage DiffServ and which details very precisely all the instances that should be managed. The second approach defines the COPS (*Common Open Policy Service*) protocol (RFC 2748 [DBC<sup>+</sup>00b]), as the Configuration protocol. Within this approach, two architectures exist:

- One based on RSVP which outsources each decision about the router configuration (the PEP asks the PDP for every change of its policy). In that case, the management protocol is COPS for RSVP, RFC 2749 ([DBC<sup>+</sup>00a]);
- The second based on the provisionning : a Policy Information Base (PIB) is defined within the PEP. It stores the policy available on this PEP. The PEP most of the time takes its own decision. It only asks the PDP about new policies if the current ones seem inefficient in the context of the current networking. In the second case, the management protocol is COPR-PR, RFC 3084 ([CSD<sup>+</sup>01]).

As reading the few articles devoted to the policy-based management on IPv4 networks, it seems that COPS is mostly used to configure the network for mobility, QoS or the security (IPsec).

**Its IPv6 availability** Since we have already seen the SNMP standard, we will study here the second approach concerning the COPS architecture.

Its protocol is defined upon TCP and the several objects included into its messages, and which could contain IP addresses, are able to contain both IPv4 and IPv6 addresses (see RFC 2748 [DBC<sup>+</sup>00b]).

The Policies themselves are IPv6 enabled and the field able to contain IP addresses are defined to be able to contain both type of them (see RFC 3084 [CSD<sup>+</sup>01], for example).

---

<sup>11</sup><http://www.ietf.org>

So COPS can be implemented over IPv6, without any major changes. But, currently, it seems that nothing was done.

### 3.2.2 The Web-based standards

The main Web-based standard is WBEM.

**Presentation** WBEM is defined by the *Distributed Management Task Force* (DMTF<sup>12</sup>). Its purpose is to:

- deliver an homogeneous view of the managed ressources, whatever they are and whatever the protocol to access to them is,
- integrate the already defined management approaches,
- enable the management information exchange between multiple management applications.

WBEM is defined by:

- a data model, called *Common Information Model* (CIM) standard,
- an encoding specification *xmlCIM encoding specification*,
- and a transport mechanism, *CIM operations* over HTTP.

The Information Base is called *CIM Object Repository*, and the entity that manages it is called the *CIM object manager*. The data made visible into the CIM Object repository are collected by providers. Those providers request the managed entity with the specific standard protocol used by this entity (SNMP for example) (see Figure 8).

---

<sup>12</sup><http://www.dmtf.org>

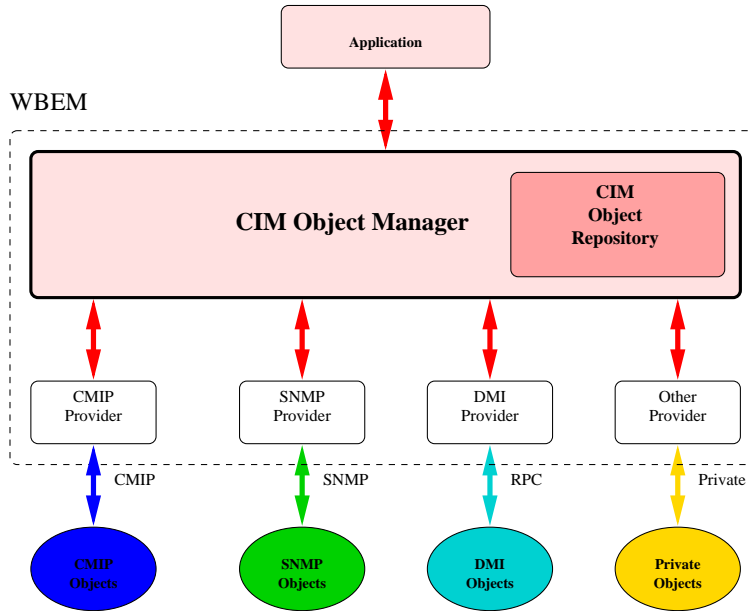


Figure 8: WBEM architecture entities

Each WBEM entity could be server one time and client at another time. It allows some possible and non permanent hierarchisation, if needed.

**Its IPv6 availability** As could be observed onto the WBEM site, the Common Information Models are already IPv6 enabled. HTTP also exists upon IPv6. So, WBEM is available over IPv6 but, as far as we could search, we could not find any implementation of it.

### 3.2.3 The Authentication standards

For the AAA (*Authentication Authorisation and Accounting*) functionality, the main standards used for IPv4 networks are RADIUS (*Remote Authentication Dial In User Service*), and KERBEROS V.

#### RADIUS and DIAMETER

**Presentation.** RADIUS was first defined by the IETF into the RFC 2865 ([RWRS00]). It is a protocol to carry authentication, authorisation and configuration information between

a *Network Access Server* (NAS) and a shared authentication server. This authentication server could be seen as a database which contains authentication, authorisation and configuration information for each user (like for example the services enabled for each user).

The architecture is a client/server architecture where the RADIUS client is the NAS and the RADIUS server is the authentication server. A *link* is open when a user or service wants to access the network. When a RADIUS client wants to authenticate one of its links, it sends a request to its dedicated RADIUS server, including first users information (like a login and a password that the user has to give). The RADIUS server verifies the validity of those information and sends its answer to the client. Depending on this answer, the RADIUS client authorizes the user to connect itself or not. If the connection is allowed, it is configured with the information received from the RADIUS server.

All the transactions between a client and a server are authenticated with a shared secret and the confidential information, like a password, is encrypted.

**Its IPv6 availability.** It was updated, to be able to authenticate IPv6 networks, in april 2001, by the RFC 3162 ([AZM01]). So everything is ready but, here again, there is currently no implementation of it.

This is certainly because RADIUS has some scalability problems. As described into the RFC 2865, RADIUS could “suffer degraded performance and lost of data when used in large scale systems”. That is why a new protocol, called Diameter, was defined by the AAA group of the Operation and Management Working Group of the IETF<sup>13</sup>. Since the beginning, this new protocol was specified to be able to authenticate IPv6 networks. And, while it is currently only a simple draft (draft-ietf-aaa-diameter-15.txt [CLG<sup>+</sup>02]) and not a RFC, there is already one implementation of it, due to Sun, based upon the 7th version of the draft. A new one will be available in 2003, due to the MobyDick<sup>14</sup> IST project, defined upon the 10th version of its draft. As Diameter offers more fonctionnalities than RADIUS and solves its scalability problem, this is certainly this AAA standard that will be used to authenticate or account IPv6 networks.

## KERBEROS V

**Presentation.** This standard was developped by the Massachussets Institut of Technology (MIT<sup>15</sup>). It only authenticates users or services, that is,

- it verifies that they are allowed to connect themselves to the network,
- and that the request and the traffic is coming from the source that it was supposed to.

The interest of Kerberos, against most of authentication protocols, is that it never sends a password on the network without having it encrypted before. But this supposes that every

<sup>13</sup><http://www.ietf.org/html.charters/aaa-charter.html>

<sup>14</sup><http://www.ist-mobydick.org>

<sup>15</sup><http://web.mit.edu/kerberos/www/>

application or service connected to the network should use Kerberos.

The idea is to define a mapping between the user password and a special Kerberos Key. This mapping is made locally on the system and it is the Kerberos Key that transits through the network. This key allows a Kerberos server to authenticate the requester and send back a *ticket* that authorizes the user to connect itself during a temporary period. This period is often of 8 hours.

**Its IPv6 availability.** In the case of Kerberos, the data are strings. So, IPv6 does not affect them. The problem could be encountered with the protocol, because the IP addresses are included into the ticket. But the MIT itself said that Kerberos V was partly implemented over IPv6 since its `version`<sup>16</sup> 1.2.

### 3.3 Conclusion

Except for the AAA management functionality, for which IPv6 standards exist and are implemented, there is no standards available yet for the other management functionalities. The reason is either because the standards themselves are not available (like SNMP), or either because there is no implementation of them (like COPS or WBEM).

So, how can we manage IPv6 networks today ?

---

<sup>16</sup><http://web.mit.edu/kerberos/www/krb5-1.2/>

## Part II

# How to manage IPv6 networks ?

Today, all IPv6 network management middlewares that exist are all standard independent. They are sometimes the result of the study and porting of an IPv4 network management middlewares or are new ones, defined explicitly for IPv6 native networks.

## 4 IPv6 management middlewares

A first kind of tools mixes CLI (*Command Line Interface*) commands, telnet, rsh or RPC calls and a Web interface. A second one replaces the CLI commands with a XML file. Some others are simply relying on basic protocols like ICMPv6. We will describe each of them into the following sections.

### 4.1 The CLI-based management middleware

Those middlewares are most of the time defined with CGI and Perl files. They can be accessed by a URL (like w6.loria.fr). Then, an HTML page, like the one in the Figure 9, is sent to the user.

It gives a list of CLI commands that could be send to one or more routers already known. As illustrated into Figure 10, it then connects itself to the remote router. When it is done, it sends the chosen CLI command and waits for the response. The result is sent back to the requester and could be either display or parse in order to save some particular data into a repository.

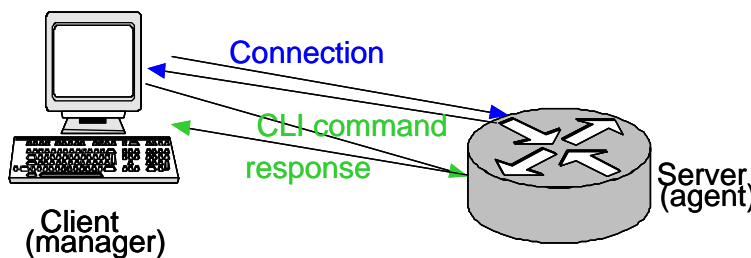


Figure 10: Schema of a CLI-based middleware

One inconvenient of such middleware is that it needs a login and a password to access any equipment it manages.



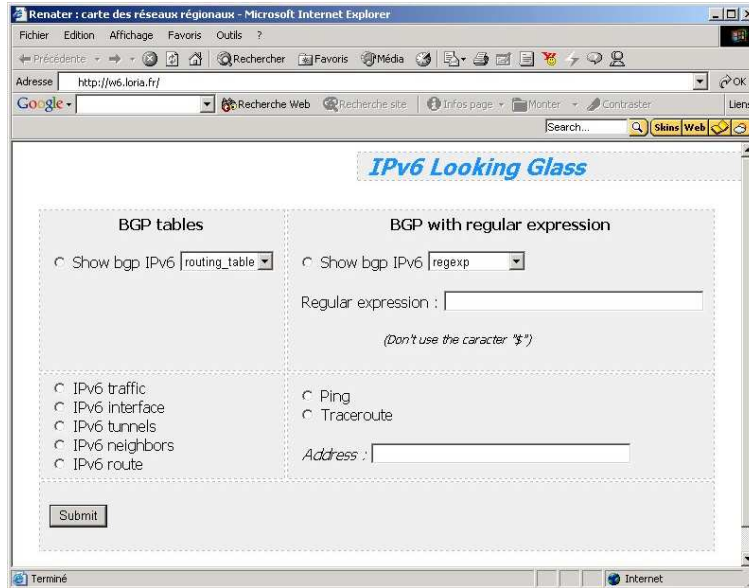


Figure 9: Example of HTML interface for a CLI-based middleware

Another inconvenient is that it is CLI dependent. That means that any update of the CLI commands implies an update of the middleware. Further, currently, most of those tools are defined to address Cisco routers only. But some labs, like the LORIA, plan to update them to be able to connect other equipments like Juniper or 6wind routers.

## 4.2 The XML-based management middlewares

They are defined in the same way than the CLI-based middlewares (see Figure 11), except that it is XML files, or pieces of XML files (case of a XML-RPC), that are sent or received, to or from the remote server, instead of CLI commands.

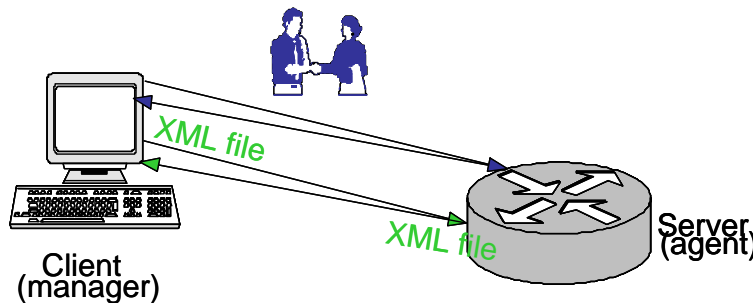


Figure 11: Schema of a XML-based middleware

### 4.3 The middlewares using protocols

Currently, the main protocol used is ICMPv6 (Internet Control Message Protocol). The tools that rely on this protocol most of the time are defined upon the ping6 function or the traceroute6.

## 5 Some examples

We will present here some examples of the most important middlewares that can be used to manage IPv6 networks. Some others could be found into the [AAB<sup>+</sup>02] report.

### 5.1 For fault management

One of the main used tools for fault management is the *Looking Glass*. For the moment, it is defined only with the Cisco CLI but will be available for Juniper ones at the beginning of 2003. The commands available are those which are enabled for the login used by the Looking Glass to log itself to the Cisco (see Figure 9) With that tool, you could verify the configuration of the router, see some traffic statistics or the routing table. They are very useful also to try to explain the problems that are encountered in the traffic transmission. Further, the parsing of the returned information could allow to have a more comprehensive display.

They are IPv6 native. Two versions exist: a perl<sup>17</sup> one and a PHP<sup>18</sup> one.

---

<sup>17</sup><http://w6.loria.fr>

<sup>18</sup>[www.6net.garr.it/lg6.php](http://www.6net.garr.it/lg6.php)

## 5.2 For services

Looking Glass could be used here again, associated to some other tools like *MPing*, *Analyser*, *Ethereal* or the *RRDtool* (Round-Robin Database tool).

### 5.2.1 MPing

*Mping*<sup>19</sup> is a tool relying on the ping6 fonctionnality. It helps to verify the IPv6 connectivity between multiple IPv6 interfaces and could realize some performance testing between them. It will be IPv6 native at the end of 2002.

### 5.2.2 Analyser and Ethereal

*Analyser*<sup>20</sup> and *Ethereal*<sup>21</sup> are traffic analysers. They catch the packet coming through the interface, analyse them and display them in a convenient way. They could help to verify that the traffic between two IPv6 interfaces is exactly what it should be and could help to verify that an interface is well-configured or behave as it should. They could be both used on IPv6 native networks.

### 5.2.3 Multicast beacon

*Multicastbeacon*<sup>22</sup> is a client/server application which gives some statistics on the multicast traffic. Multicast is one of the new functionality of IPv6. It already existed in IPv4 networks but was integrated into the new IP protocol because it was of a great importance for the transmission of streaming flows, like video or audio.

Each client owns a beacon daemon. This daemon periodically send message to the other multicast group members to measure the loss, the delay, the jitter, the number of messages that arrived into the wrong order and the number of messages that are duplicate. All those informations are sent to a beacon server which displays a synthesis table. This application is defined in Java and is available for IPv6 networks.

### 5.2.4 RRDtool

This tool displays the traffic that you want to monitor, into graphs like the one showed in Figure 12. It is defined on CLI as the Looking Glass. It regularly pools the remote router and parses the information received from the Cisco to get only the data expected. They are saved into a round-robin database. By this way, any information can be displayed later on into smart graphics. They are defined with Perl and are available on IPv6<sup>23</sup>.

---

<sup>19</sup><http://drift.uninett.no/mping/index.html>

<sup>20</sup><http://analyser.polito.it>

<sup>21</sup><http://www.ethereal.com>

<sup>22</sup><http://dast.nlanr.net>

<sup>23</sup><http://w6.loria.fr>

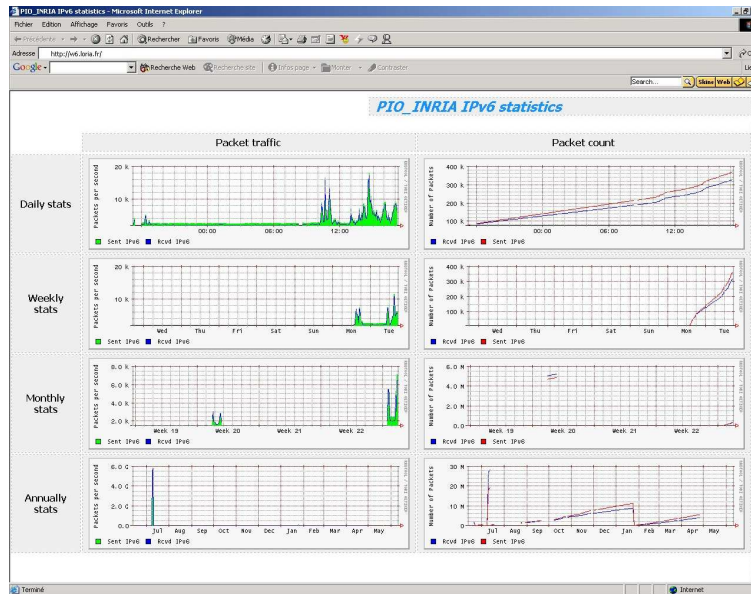


Figure 12: RRDtool display

### 5.3 For configuration management

To manage the configuration of an IPv6 networks, few tools are available.

#### 5.3.1 Configuring an interface

The IPv6 autoconfiguration fonctionnality could be used. IPv6 autoconfiguration was defined because:

- the configuration of a network was often long and difficult,
- further to the modification of the format of the IP address, a lifetime was added to each address. So an interface should have to change its address if its associated lifetime is over. That is why a new fonctionnality was needed to help to the regular configuration of the IPv6 interfaces.

Currently, by autoconfiguration, the IETF means the definition of an IPv6 link-local address, the verification of its uniqueness, the definition of a global unicast address. There are two types of autoconfiguration:

- the stateful autoconfiguration (DHCPv6, see [DPV<sup>+</sup>02]). A server sends its configuration to each interface requesting it.
- the stateless autoconfiguration which allows an interface to configure itself, without any external help like for a “Plug and Play” driver.

We will describe here the stateless autoconfiguration.

We have seen (cf section 2.1, p 7) that the link-local address owns a specific prefix. To define the whole address, an interface Id should be defined. That is the role of the several algorithms defined, depending on which type of card or network you have (see [HD98] or [Ciz02] for more details about them).

When the link-local address is defined, the interface should verify its uniqueness before using it. That is the DAD (*Duplicate Address Detection*) role: it sends a message to its neighbors, using its link-local interface. If no interface answers that this address is already in use, then the address is defined as valid.

The last step is the construction of the IPv6 global unicast address. Depending on the Router Advertisement received from the default router, the interface knows that it has to use a stateful autoconfiguration or a stateless autoconfiguration. The stateless autoconfiguration is defined on the same principle than the definition of the link-local address. Mostly, its interface Id is the same than the one used into the link-local address. The prefix is received into the Router Advertisement. So, by concatenating this two pieces of information, the interface is able to create its new IPv6 global unicast address.

But this autoconfiguration could only be used for simple nodes and not for routers. Some proposals exist to solve the problem of routers autoconfiguration, like the draft-chelius-autoconf-zero.txt (see [CFT02]), which describe a way to configure OSPFv3 routers. But they are just drafts and no standards currently exist. So, for the moment, the only solution to configure routers is to manually configure them, that is use CLI commands and scripts, like for IPv4 networks. It should be noticed that Juniper has recently try to help the network managers by adding a XML interface to its routers, called JUNOScript<sup>24</sup>.

### 5.3.2 Archiving the configuration.

Rancid<sup>25</sup> is available for IPv6 networks.

## 5.4 Topology discovery tools

Several tools already exist, depending on what kind of network is to be managed.

---

<sup>24</sup><http://www.juniper.net>

<sup>25</sup><http://www.shrubbery.net/rancid>

#### 5.4.1 For backbones

The ASPath-tree is a service defined for the 6bone (the IPv6 backbone) to display its BGP4+ topology. It is based on CLI, using rsh functions and displaying the BGP4+ routes between Cisco routers. Available on IPv4 for the moment, it will be ported on IPv6 in the context of the 6net project.

#### 5.4.2 For LANs

The LAN automatic topology discovery tool presented at the section 2.2.2, p 9 will be available at the beginning of 2003, on the LORIA IPv6 site<sup>26</sup>. It relies on the ICMPv6 protocol and on the Neighbor Discovery Protocol. With its hierarchical architecture, it could find out and display all the interfaces connected into an addressing domain defined with an IPv6 prefix, and discovered all their IPv6 link local addresses, physical addresses, most of their IPv6 global addresses if the interface are stateless autoconfigured. In some cases explained in [Ast02], it could also associates the multiple interfaces of a node.

#### 5.4.3 For multicast networks

To find out the topology of a multicast network, a tool called Mtrace6 made by the kame<sup>27</sup> project can be used. This tool is the implementation over IPv6 of the Mtrace function, the implementation of the traceroute for multicast networks. A request is propagated along the reverse path depending of the multicast group and the source address that should be tested. Data about the path are collected into the packet and sent back to the requester.

To be IPv6 enabled, this tool has to change:

- the requests and the responses which are now ICMPv6 packets,
- the interface identifier. As IPv6 addresses, especially link local ones, are not unique on a network, even on the same node, IP addresses could not be used to identify the incoming and outgoing interfaces. The interface index on the node is used instead,
- the packet format:
  - as the interface index is context dependent, new fields have to be inserted to give a global dimension to this index (most of the time it is the associated node IPv6 global outgoing interface address that is sent),
  - to be able to carry IPv6 addresses.

---

<sup>26</sup><http://w6.loria.fr>

<sup>27</sup><http://www.kame.net>

## Part III

# Conclusion

IPv6 will be in a very close future the underlying protocol of the Internet, at least into some part of the world. The management problem, as it was not really taken into account until now, is becoming one of the crucial point of the next years, because IPv6 native networks emerge and because it is impossible to manage a network without appropriate tools.

As we saw, IPv6 is a big evolution over IPv4 but does not mean that IPv6 management will be just the following of IPv4 management. The modification of the IP addressing architecture has a lot of consequences on the management applications.

We have seen that because of this modification, the main IPv4 management framework, SNMP, is not yet available for IPv6 networks and most of the tools should be analysed in depth, one after another.

The consequence is that the main management platforms that exist for IPv4 networks are not available for IPv6 ones, and that very few tools are available today. Those that are currently used are “light tools”: they are not defined upon standards. But all those tools do not satisfy the scalability factor and most of them are not secure: see the Looking Glass, which is the main middleware currently used for IPv6 networks, that needs to store a login and a password for each equipment it manages.

Only standards can solve those two problems. But which standards ? The definition of the IPv6 management architecture gave a good opportunity to make a point on the IPv4 management. This was done at the IAB<sup>28</sup> (*Internet Architecture Board*) Network Management Architecture Workshop in Reston, VA, US, in June 4-6 2002. The main conclusion of this workshop is that SNMP was a great architecture but with a lot of disadvantages. SNMP is too big and too complex now. It is « heavy » and not obvious to use or to implement, because a SNMP architecture means a lot of files, on both side, agent and manager. Not obvious to use also because the way that SNMP is requesting the MIB is not the way the manager needs the information. The study that Jürgen Schönwälder did about the « Evolution of the OpenSource SNMP tools » [Sch02] shows as an evidence that there is a great need to simplify and to automate the interface between the manager and the agent. That is why the SNMP tools, like scripts, and the WEB-based architecture tools are taking importance. People prefer to work with simpler procedure (CLI commands and scripts). Further, he says that the first users of the management platform needed to have a real knowledge of the semantics of the MIB objects. Because it was not often the case, new tools try to simplify the access of the user to the information by taking itself into account the semantic of the object.

Another problem of SNMP is the number of MIBs that are available for IPv4 networks now. When you see the time that was needed to port the MIB II upon IPv6, one could be afraid about the amount of time that will be needed to port the hundred of other MIBs, and their

---

<sup>28</sup><http://www.iab.org>

thousand implementations. That is why some researchers think that this work will never be done and that SNMP will hardly survive of the change of IPv6.

But as we said, SNMP, despite all these critics, remains a good architecture to monitor the fault of the network and, in fact, this is the only one that we have for the moment. So, certainly, SNMP is not dead. But, it will certainly have not the same importance that it had to manage IPv4 networks. The time that will be necessary to port the entire SNMP architecture upon IPv6 will be used by other standards to take off. Other standards already existing, like COPS or WBEM may be, or other standards to be defined.

This document shows also that if there is some solution to monitor IPv6 network, to manage its fault or its authorisation, authentication or accounting functions, there are very few things for the configuration. And at the last IAB workshop, referenced above, the operators and managers have noticed that there is, in that domain, great needs: great needs of a common high level data models, great needs of a configuration transfert protocol and great needs of a generalized configuration language. And here, it seems that XML has a great role to play.

So, in fact, there is a lot of work to do to use IPv4 management tools to manage IPv6 networks. But managing IPv6 networks is more complex than managing IPv4 ones. When it was decided to define a new IP protocol to solve the problem of address space, researchers take this opportunity to integrate new functionalities like autoconfiguration and multicast, but also mobility and security (with IPsec).

Managing IPv6 networks means also, during a period that could be long, to manage the transition. here again, it is a hard problem, as there is not one transition architecture but several ones, that could be used at the same time, on the same network. Some researchers proposed solutions like Tunneltrace<sup>29</sup> or the SNMP proxy<sup>30</sup> defined in the context of the 6net project. But they are the first solutions proposed and are not enough to fully solve this problem.

In conclusion, currently, we can manage IPv6 networks with a subset of effective tools but a lot of work remains to fully manage them.

---

<sup>29</sup><http://www.dia.uniroma3.it/compunet>

<sup>30</sup><http://www.ip6.man.poznan.pl>



## References

- [AAB<sup>+</sup>02] François-Xavier Andreu, Isabelle Astic, Gabriela Barbagalo, Tim Chown, Rob Evans, Olivier Festor, Bartosz Gajda, Ioannis Kappas, Georgios Koutepas, Simon Leinen, Paolini Gabriella, Maurizio Patrignani, Fulvio Risso, Trond Skjesol, Robert Szuman, and Bernard Tuy. 6NET Management Tools Requirements. report 32603/LORIA/DS/6.2.1/A1, European Commission, June 2002.
- [ACD<sup>+</sup>02] Isabelle Astic, Tim Chown, Jérôme Durand, Robert Evans, Fulvio Risso, Duncan Rogerson, and Bernard Tuy. 6net IPv6 Network Management Cookbook. report 32603/Partner/DS/6.3.1, A02-R-187, European Commission, Oct 2002.
- [AJAA02] Mongi Abdelmoula, Neha Jha, Ashok Anand, and Isabelle Astic. Implementation of IP-MIB modules for IPv4 and IPv6 protocol. Technical Report RT-0271, A02-R-186, INRIA, Oct 2002.
- [Ast02] Isabelle Astic. Recherche dynamique de topologie pour les réseaux IPv6. report, LORIA, Nov 2002.
- [AZM01] B. Aboba, G. Zorn, and D. Mitton. RADIUS and IPv6, August 2001. RFC 3162.
- [Bak92] F. Baker. Ip forwarding table, July 1992. RFC 1354.
- [Bak97] F. Baker. IP Forwarding Table, January 1997. RFC 2096.
- [CFSD90] J.D. Case, M. Fedor, M.L. Schoffstall, and C. Davin. Simple network management protocol (snmp), May 1990. RFC 1157.
- [CFT02] G. Chelius, E. Fleury, and L. Toutain. Using OSPFv3 for IPv6 router autoconfiguration, June 2002.
- [Ciz02] Gisèle Cizault. *IPv6, Théorie et pratique*. O'Reilly, 2002.
- [CLG<sup>+</sup>02] P.R. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter Base Protocol, draft-ietf-aaa-diameter-15.txt, October 2002.
- [CLW02] Ritu Chadha, George Lapiotis, and Steven Wright. Policy-based Networking. volume 16, March/April 2002.
- [CMRW96] J. Case, K. McCloghrie, M.T. Rose, and S. Waldbusser. Structure of Management Information for Version 2 of the Simple Network Management protocol (SNMPv2), January 1996. RFC 1902.
- [CSD<sup>+</sup>01] K. Chan, J; Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith. COPS usage for Policy Provisioning (COPS-PR), march 2001. RFC 3084.

- [DBC<sup>+</sup>00a] D Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. COPS Usage for RSVP, January 2000. RFC 2749.
- [DBC<sup>+</sup>00b] D Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol, January 2000. RFC 2748.
- [DHRS00] M. Daniele, B. Haberman, S. Routhier, and J. Schoenwaelder. Textual conventions for internet network addresses, June 2000. RFC 2851.
- [DHRS02] M. Daniele, B. Haberman, S. Routhier, and J. Schoenwaelder. Textual Conventions for Internet Network Addresses, 2002. RFC 3291.
- [DPV<sup>+</sup>02] R. Droms, C. Perkins, B. Volz, M. Carney, and T. Lemon. Dynamic Host Configuration Protocol for IPv6 (DHCPv6), draft-ietf-dhc-dhcpv6-28.txt, November 2002.
- [EFTW02] Rob Evans, Olivier Festor, Bernard Tuy, and Ralf Wolter. 6NET Network Management Initial Architecture. report 32603/LORIA/DS/6.1.1/A1, European Commission, June 2002.
- [FHK<sup>+</sup>01] B. Fenner, B. Haberman, McCloghrie K., J. Schoenwaelder, and D. Thaler. Management Information Base for the User datagram Protocol (UDP), draft-ietf-ipngwg-RFC2013-update-01.txt, 2001.
- [FHKS01] B. Fenner, B. Haberman, McCloghrie K., and J. Schoenwaelder. Management Information Base for the Transmission Control Protocol (TCP), draft-ietf-ipngwg-RFC2012-update-01.txt, 2001.
- [FHST01a] B. Fenner, B. Haberman, J. Schoenwaelder, and D. Thaler. IP Forwarding Table MIB, draft-ietf-ipngwg-RFC2096-update-00.txt, 2001.
- [FHST01b] B. Fenner, B. Haberman, J. Schoenwaelder, and D. Thaler. Management Information Base for the Internet Protocol (IP), draft-ietf-ipngwg-RFC2011-update-00.txt, 2001.
- [HD98] R. Hinden and S. Deering. IP Version 6 Addressing Architecture, July 1998. RFC 2373.
- [HO98] D. Haskin and S. Onishi. Management Information Base for IP Version 6 : Textual Conventions and General Group, December 1998. RFC 2465.
- [HOD97] R. Hinden, M. O'Dell, and S. Deering. An IPv6 Aggregatable Global Unicast address Format, January 1997. RFC 2374.
- [HP01] H. Hazewinkel and D. Partain. The Differentiated Services Configuration MIB, June 2001.

- [McC96a] K. McCloghrie. Management Information Base for the Internet Protocol (IP), November 1996. RFC 2011.
- [McC96b] K. McCloghrie. Management Information Base for the Transmission Control Protocol (TCP), November 1996. RFC 2012.
- [McC96c] K. McCloghrie. Management Information Base for the User Datagram Protocol (UDP), November 1996. RFC 2013.
- [MPS99a] K. McCloghrie, D. Perkins, and J. Schoenwaelder. Structure of Management Information v2 (SMIv2), April 1999. RFC 2578, STD 58.
- [MPS99b] K. McCloghrie, D. Perkins, and J. Schoenwaelder. Textual conventions for SMIv2, April 1999. RFC 2579, STD 58.
- [MPST02] M. MacFaden, D. Partain, J. Saperia, and W. Tackabury. Configuring Networks and Devices With SNMP, October 2002.
- [MR91] K. McCloghrie and M.T. Rose. Management information base for the network management of tcp/ipbased internets - mib ii, March 1991. RFC 1213, STD 17.
- [RM91] M.T. Rose and K. McCloghrie. Concise mib definitions, March 1991. RFC 1212, STD 16.
- [Ros90] M.T. Rose. Management information base for network management of tcp/ip-based internets : Mib ii. Technical report, IETF, May 1990. RFC 1158, STD 15.
- [RWRS00] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS), June 2000. RFC 2865.
- [Sch02] Jürgen Schönwälder. Evolution of Open Source SNMP Tools. April 2002. URL::[www.ibr.cs.tu-bs.de/users/schoenw/papers/sane-2002.pdf](http://www.ibr.cs.tu-bs.de/users/schoenw/papers/sane-2002.pdf).
- [WSH02] S. Waldbusser, J. Saperia, and T. Hongat. Policy-Based Management MIB, June 2002.



---

Unité de recherche INRIA Lorraine  
LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)  
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)  
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)  
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-0803