

Internet Protocol version 6 Overview^{*}

Albert Cabellos-Aparicio, Jordi Domingo-Pascual

Departament d'Arquitectura de Computadors,

Universitat Politècnica de Catalunya, Spain

{acabello,jordid}@ac.upc.edu

Abstract: IP stands for “Internet Protocol” and it was designed during the ‘70s with the purpose of interconnecting heterogeneous network technologies. IP was a huge success, and made it possible to create today’s Internet. Currently, the Internet is predominantly using the fourth version of IP (IPv4), however the huge success of the Internet is pushing IPv4 to its limits. The Internet Engineering Task Force (IETF) designed IPv6 to become the replacement for IPv4. IPv6 solves most of the major problems of IPv4 and has several new features. This paper presents an overview of the IPv6 protocol, discussing its operation, its addressing architecture, its header format, the neighbor discovery protocol and one of the major issues of IPv6: how to transition from IPv4; presenting a set of transition mechanisms that provide communication between IPv4 and IPv6.

Keywords: *IPv6, IPv6 Header Format, Extension Headers, Neighbor Discovery, IPv6 Addressing, Transition Mechanisms*

1.- Introduction

IP stands for “Internet Protocol” and it was designed during the ‘70s with the purpose of interconnecting heterogeneous network technologies. IP was a huge success, and made it possible to create today’s Internet. Currently, the Internet is predominantly using the fourth version of IP (IPv4) [1] however the huge success of the Internet is pushing IPv4 to its limits.

During the ‘90s several problems related with IPv4 were raised at the Internet Engineering Task Force (IETF) [2]. IPv4 uses a 32-bit field to identify host interfaces (commonly known as Internet Addresses).

^{*} This work was partially funded by the MCyT (Spanish Ministry of Science and Technology) under contract FEDER-TIC2002-04531-C04-02 and the CIRIT (Catalan Research Council) under contract 2001-SGR00226

When IPv4 was designed 32 bits were enough and in fact, it was never designed to support such a big network as the Internet. This 32-bit field is becoming restrictive nowadays, an Internet Address is scarce; this has caused the appearance of Network Address Translation (NAT) and other mechanisms that, while they permit communication using a single IPv4 address, they break the principles of the Internet. The Internet architecture is based on the end-to-end principle [3] where all the end hosts can communicate between each other without any impediment. The imposition of NAT slows down Internet growth and the creation of new services and applications. That's why the IETF began to design a successor to IPv4: IPv6.

IPv6 [4] is the new version of the Internet Protocol and it has several improvements. It has extended addressing capabilities; the address field is 128-bits in length. With IPv6 we have a huge address space (3.4×10^{28} addresses), we can connect more devices to the Internet without breaking the end-to-end principle, create a complex address hierarchy and benefit from simpler configuration. IPv6 also provides an improved header format and routers are able to process the IPv6 header in a more efficient way. Options in the IPv4 header are a patch (like mobility and security) but, in IPv6, options are part of the protocol (using the new extension header format). In summary, the Internet will be even more scalable with IPv6 than with IPv4.

The Internet is still using IPv4, but IPv6 is now being widely deployed in research networks, and this deployment is a critical issue. In the future, it is possible that the Internet will be IPv6 only, but, until that moment, IPv4 and IPv6 must coexist. IPv6 deployment must not disrupt the current Internet, and, somehow, IPv4 and IPv6 must coexist. This is accomplished by special mechanisms, named transition mechanisms which allow communication between the IPv4 and the IPv6 world. A lot of transition mechanisms have been designed and implemented but they provide less forwarding speed than a native communication (IPv4 with IPv4 or IPv6 with IPv6) and some of them are difficult to deploy.

2.- The IPv6 Protocol

2.1.- The IPv6 Header

The new IPv6 format header has a fixed size of 40 bytes, IPv4 has a header size of 20 bytes (plus the options, if necessary). Figure 1 shows the IPv6 header format.

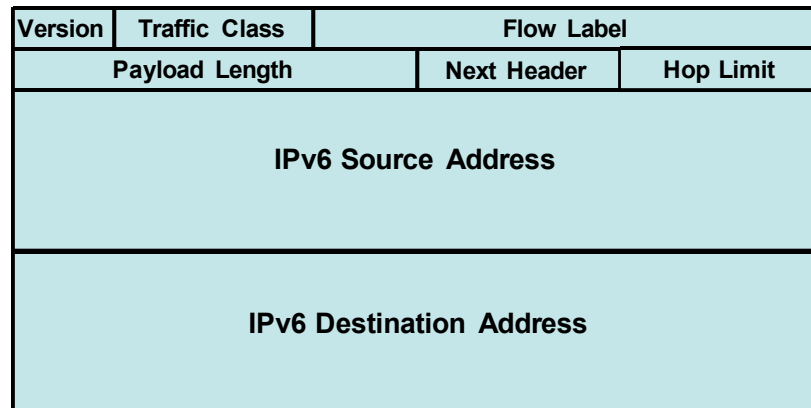


Figure 1 – The IPv6 Header Format

Version (4 bits) → The IP version of the header, in IPv6 this is of course “6”. This field is exactly the same as in IPv4.

Traffic Class (8 bits) → A label used for Quality of Service, similar to the Type of Service field in IPv4.

Flow Label (20 bits) → This is a new field and it is used to label flows which routers will forward according to different policies.

Payload Length (16 bits) → The length of the payload; the rest of the packet following the IPv6 header in bytes. Up to 65,536 bytes.

Next Header (8 bits) → The type of header following the IPv6 header. This field is similar to the IPv4 “Protocol” field, in IPv6 it also carries the next header identification.

Hop Limit (8 bits) → Decrement at each router, when it reaches zero the packet is discarded. This field is similar to the IPv4 “Time To Live” field.

Source Address (128 bits) → Address of the originator of the packet.

Destination Address (128 bits) → Destination address of the packet.

The IPv6 header has fewer fields than the IPv4 header. The checksum is no longer necessary because the probability of a transmit error that forces a packet loss is very low. The IPv4 fragmentation fields (Identification and Fragment Offset) are also removed, it was decided that for best performance, an IPv6 packet cannot be fragmented en route (only by source hosts and using a special header). When a host decides to communicate with another one, it has to detect the Maximum Transfer Unit (MTU) in the path [5] and send the packets with the correct size, or use the generalized minimum transfer unit for IPv6 (1280 bytes). As mentioned above, the IPv6 header format has a fixed size and the header fields are 64-bit-aligned, which allows faster hardware processing by the newest 64-bit network processors.

2.2.- IPv6 Extension Headers

In IPv6, optional internet-layer information is encoded in separate headers that are placed between the IPv6 header and the upper-layer header in a packet.

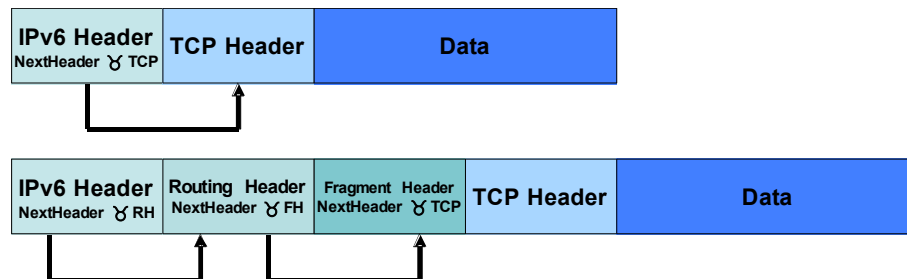


Figure 2 – IPv6 Extension Headers

Figure 2 shows how IPv6 extension headers are encoded into an IPv6 packet. IPv6 extension headers provide great flexibility to the IPv6 protocol. Using these special headers, the protocol can be extended to incorporate new functionality in an easy and efficient way. The IPv6 header has a fixed size (40 bytes) and a Next Header pointer that points to the following header, which can be the transport layer header (TCP, UDP...) or another Extension Header.

The IPv6 standard defines six types of Extension Headers that, with one exception (Hop-by-Hop) are not examined or processed by any node along the packet's delivery path. IPv4 options are limited by its header size, while in IPv6 there is not a virtual limitation. In general, IPv6 tries to optimize the performance not processing Extension Headers until they are used. In IPv4 options are rarely used [6], however, a lot of new IPv6 functionality relies on Extension Headers, such as Mobility or Security.

3.- IPv6 operation

3.1.- IPv6 Addressing

IPv6 addresses are 128-bit fields, which uniquely identify a single interface or a group of network interfaces. In addition, we can assign one or more IPv6 addresses to a single network interface. IPv6 has three different types of addresses:

- Unicast → This type of addresses identify uniquely only one network interface. They are the most common type of addresses.
- Multicast → They identify a set of different network interfaces, usually belonging to different nodes. If a packet is sent to a multicast destination address, the packet will be delivered to all the network interfaces identified by this special address.
- Anycast → This type of address is assigned to more than one network interface, and if a packet is sent to an anycast address, it is routed to the “nearest” interface (in terms of routing distance) having that address.

IPv6 addresses are represented as text strings using the “xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx” format, where ‘xxxx’ is a 16-bit hexadecimal value; figure 3 shows several examples.

IPv6 Multicast Address

FEDC:BA98:7654:3210:FEDC:BA98:5678:3321

IPv6 Unicast Address

2001:0145:4556:0000:0000:0000:0000:0002

Figure 3 – *Textual representation of IPv6 addresses.*

Some IPv6 addresses may contain long strings of zero bits, in this case, we can use “::” which indicates one or more groups of 16 bits of zeros. For example, the address “2001:0145:4556:0000:0000:0000:0000:0002” could easily be rewritten as “2001:145:4556::2”.

Figures 4 shows the Unicast address basic structure currently used in IPv6:

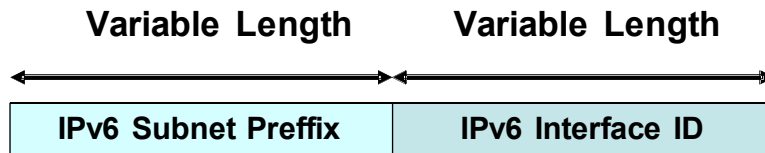


Figure 4 – Unicast address structure

The text representation of IPv6 address prefixes is similar to the way IPv4 addresses prefixes are written in CIDR (Class-less interdomain routing) notation; the format used is “IPv6 Address/Prefix-Length”. Prefix-Length is a decimal value specifying how many of the leftmost contiguous bits of the address comprise the prefix (in a given routing level). For example, “2001:145:4556::2/48” represents an address with the 48-bit prefix “2001:145:4556”.

Depending on the scope of the unicast address they belong to a different subtype; if their scope is the entire Internet they are Global Unicast Addresses and if their scope is only a single link, they are called Link-Local IPv6 Unicast Addresses. The first type of unicast addresses have the “subnet prefix” subdivided into “Global Routing Prefix” where its value is assigned to a site and “subnet ID” where its value is an identifier of a link within the site. The link-local unicast addresses are designed to be used for addressing on a single link for purposes such as automatic address configuration (see next section).

The IPv6 standard defines some special addresses; table I shows the most important ones:

Addresses in CIDR notation	Special Meaning
::/128	The unspecified address, used when the interface has not an address assigned.
::1/128	The loopback address, corresponding to 127.0.0.1 in IPv4
::/96	The IPv4-compatible address used for SIIT, see below
::FFFF:0:0/96	The IPv4-mapped address is used for SIIT, see below.
FE80::/10	The prefix for link-local addresses
FF00::/8	Multicast addresses

Table I – IPv6 Special Addresses

3.2.- IPv6 Neighbor Discovery

This protocol [7] solves a set of problems related to the interaction between nodes attached to the same link, defining mechanisms for discovering routers, autoconfiguring nodes and resolving link layer addresses among others. The IPv6 Neighbor Discovery (ND) protocol is similar to the ARP protocol for IPv4, however in IPv6 its functionalities have been extended. Moreover, ND relies on the ICMPv6 [8] protocol that uses network layer multicast. In contrast, ARP depends on different implementations of the link layer.

When two nodes in the same link want to exchange packets, they must know the link-layer address of each other, typically the Ethernet MAC address. In this case, the node that starts the communication sends a special ICMPv6 “Neighbor Solicitation” message to a multicast address asking for the given link-layer address of the destination host. The end-host, upon receipt of this message, replies with a “Neighbor Advertisement” announcing its link-layer address. This procedure operates in a similar way to ARP for IPv4.

In IPv6, nodes are usually configured automatically. IPv6 routers have manually configured prefixes on their interfaces, and send “Router Advertisement” messages periodically, including this prefix and other important parameters. When a node starts up, it first configures all its interfaces with a link-local address, this address is created automatically based on the link-layer address, and its purpose is to communicate within the link. Then, it must configure a global unicast address in order to be reachable for nodes outside its link. The node listens for Router Advertisement messages that have the subnet prefix embedded in them. Prepending the 64-bit subnet prefix to the automatically generated 64-bit interface ID creates a Global Unicast Address. This type of configuration is called “Stateless Autoconfiguration”. IPv6 also supports “Stateful Autoconfiguration” where nodes are configured using DHCPv6 [10] or manually.

3.3.- IPv6 Address architecture

IPv6 is now being widely deployed in research networks, and IANA (Internet Assigned Numbers Authority) through the Regional Internet Registries (RIR) is assigning IPv6 addresses. RIPE (Réseaux IP Européens) is the RIR in charge of Europe and it is assigning address to ISPs and other entities. The IPv6 addressing assignment policy states [11,12,13] that the 2000::/3 address block will be initially assigned. RIRs will assign /32 prefixes to ISPs, ISPs will assign /48 prefixes to end-sites (such as universities), /64

prefixes will be given to end-users (such as DSL users) and finally if a user needs only one address (and will never be any need for further addresses) a /128 prefix will be assigned. IANA and RIRs will assign adjacent address blocks when possible. At February 2005 near 1500 millions of /48 prefixes have been assigned (to educational entities, commercial ISPs...) where more than 50% belong to RIPE [14].

4.- Transition Mechanisms

It is clear that IPv6 will not be deployed on a “flag day”. A huge IPv4 infrastructure is currently deployed and working and a lot of services rely on it, and thus IPv6 deployment must be as smooth as possible. This means, that although IPv4 and IPv6 are not compatible they must coexist for an undetermined period of time. Transition mechanisms provide communication between IPv4 and IPv6 nodes. A lot of transition mechanisms have been defined but IETF is now focusing on a set of them.

Transition mechanisms can be divided into three different classes:

- Dual Stack → Both protocols (IPv4 and IPv6) are installed on nodes and routers. If IPv4 communication is desired, the IPv4 infrastructure will be used and the same applies for IPv6.
- Tunneling → IP packets are encapsulated into IP packets creating a “virtual link”. We can encapsulate IPv6 packets into IPv4 and travel through an IPv4 network, or vice versa.
- Translation Mechanisms → IPv4 header is translated to an IPv6 header and vice versa following a set of defined rules.

4.1.- Dual Stack

Dual Stack is the simplest transition mechanism. It provides communication between all the nodes without encapsulation or translation, which are in general costly processes. However, it has several drawbacks, the system administrators must maintain two infrastructures, and it does not reduce the demand for IPv4 addresses.

4.2.- Tunneling

Several types of tunneling transition mechanisms have been defined: Configured Tunnels [15], Automatic Tunnels and 6to4 [16]. In general, their main objective is to allow communication between isolated IPv6 hosts or networks through IPv4 networks. Figure 5 shows a tunnel transition mechanism example.

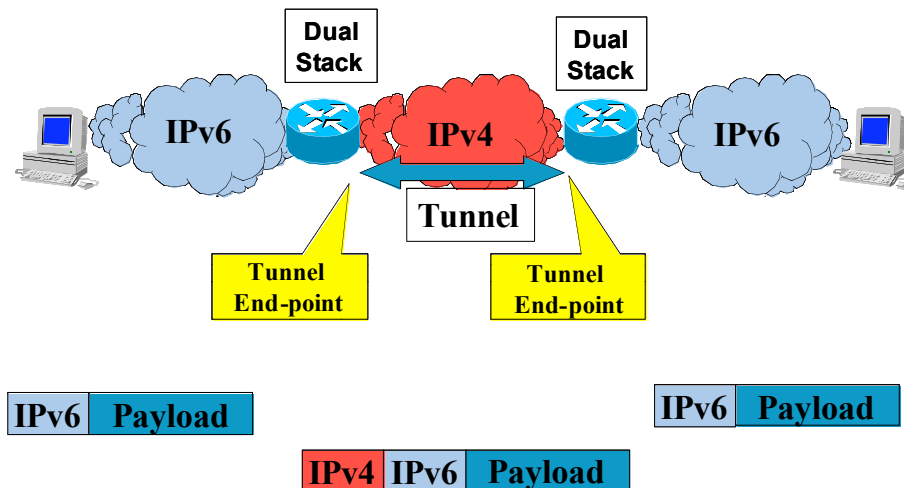


Figure 5 – Two isolated IPv6 networks communicating through an IPv4 infrastructure

Depending on which type of tunnels are used, configuration is done manually (by an administrator) or automatically (through a special IPv6 address, by the user on demand or using pre-defined bindings between IPv4 and IPv6 addresses). In general, tunneling transition mechanisms are easy to deploy and they are transparent for the IPv6 node, however they must use two IP headers (overhead) and, in some cases, they must be manually configured providing less scalability, or they use IPv4 addresses (not reducing the IPv4 address demand).

This approach has been successfully used in the 6BONE network [17] and the LONG project [18] among others. Tunneling transition mechanisms have been defined for the early stage of the IPv6 transition, where few IPv6 infrastructures have been deployed.

4.3.- Translation Mechanisms

Translation mechanisms provide communication between IPv4 and IPv6 nodes by translating the IP/ICMP packet headers. Stateless IP/ICMP Translation Algorithm (SIIT) [19] specifies how this translation must be made, these rules are depicted in table II.

IPv6 header Field	IPv4 →	IPv6	IPv4 header Field	IPv6 →	IPv4
Version	4	6	Version	6	4
Traffic Class (Class of Service)	Xxxxxxxx	XXXXXXXX (TOS and preference bits copied)	IP Header Length	N/A	5 (no options)
Flow Label	N/A	0	TOS and	XXXXXXXX	XXXXXXXX

			precedence		
Payload Length	X	X-size(IPv4 header)- size(IPv4 options)	Total Length	X	X + size(IPv4 header)
Next Header	X	X	Identification	N/A	0
Hop Limit	X	X-1	Flags	N/A	MF=0 DF=1
Source Address	A.B.C.D	::FFFF:A.B.C.D	Fragment Offset	N/A	0
Destination Address	E.F.G.H	::FFFF:0:E.F.G.H	TTL	X	X-1
			Protocol	Next Header=X	X
			Header Checksum	-	(generated)
			Source Address	::FFFF:0:A.B.C.D	A.B.C.D
			Destination Address	::FFFF:E.F.G.H	E.F.G.H

Table II – *SIIT rules for translating IPv4 headers to IPv6 and vice versa.*

NAT-PT (Network Address Translation- Protocol Translation) [20] is based on NAT and follows the SIIT translation rules, with some differences on how IP addresses are translated. While SIIT requires a complex temporal routing mechanism to route IPv4 addresses to IPv6 nodes, NAT-PT can easily assign address from a given set. It can also provide bi-directional communication between IPv4 and IPv6 nodes without changing anything on the nodes moreover, a great deal of experience exists on configuring NAT-based devices. However, translation is a costly process, it inherits the same problems that NAT has: it can not provide end-to-end security and if the protocol transports the IP address into the payload and it does not support IPv6 extension headers, and therefore, it does not support mobility or other new IPv6 features. Despite the fact NAT-PT has been extremely used in experimental networks IETF is currently working on deprecating the current specification because of its low applicability.

5.- Summary

IPv6 is version 6 of the Internet Protocol and it has been designed by the IETF as a successor to IPv4. IPv4 was never designed to support a huge network such as the Internet. Although it has a great scalability, it is now being pushed to its limits and it is limiting Internet growth.

This paper presents an overview of the IPv6 protocol, showing its header format and its basic operation. It also discusses the Neighbor Discovery protocol and the IPv6 addressing architecture. One of the major issues of IPv6 is the transition from IPv4 to IPv6. The paper explains several transition mechanisms that have been defined by the IETF.

IPv6 is now ready for world-wide deployment, near 1500 of /48 addresses blocks have been assigned by IANA and the IETF has finalized its standardization. Only minor aspects of the protocol remain experimental. IT vendors have commercial implementations of IPv6 and major Operating System vendors have included IPv6 stacks into their systems.

References

- [1] Postel J. "Internet Protocol", RFC 791, September 1981
- [2] The Internet Engineering Task Force (IETF) <http://www.ietf.org>
- [3] David D. Clark, "Rethinking the design of the Internet: The end to end arguments vs. the brave new world", ACM Transactions on Internet Technology, Vol 1, No 1, August 2001
- [4] S. Deering, R. Hinden "Internet Protocol Version 6 (IPv6) Specification" RFC 2460, December 1998
- [5] J. McCann, S. Deering, J. Mogul "Path MTU Discovery for IP version 6" RFC 1981
- [6] Fransson P, Jonsson A "End-to-End measurements on performance penalties of IPv4 options", Global Telecommunications Conference, December 2004
- [7] T. Narten, E. Nordmark, W. Simpson "Neighbor Discovery for IP Version 6 (IPv6)" RFC 2461
- [8] A. Conta, S. Deering "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification" RFC 2463
- [9] S. Thomson, T. Narten "IPv6 Stateless Address Autoconfiguration" RFC 2462
- [10] D. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" RFC 3315, July 2003
- [11] R. Hinden, S. Deering, R. Fink, T. Hain, "Initial IPv6 Sub-TLA ID Assignments", RFC 2928, September 2000
- [12] APNIC, ARIN, RIPE NCC, "IPv6 Address Allocation and Assignment Policy", ripe-267, January 2003

- [13] R. Hinden, S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003
- [14] RIPE IPv6 Allocation, Statistics <http://www.ripe.net/rs/ipv6/stats/>
- [15] R. Gilligan, E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, April 1996
- [16] B. Carpenter, K. Moore "Connection of IPv6 Domains via IPv4 clouds", RFC 3056, February 2001
- [17] 6BONE, testbed for deployment of IPv6, <http://www.6bone.net>
- [18] LONG, "Laboratories over Next Generation Networks" IST-1999-20393. <http://long.ccaba.upc.edu>
- [19] E. Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT)" RFC 2765, February 2000
- [20] G. Tsirtsis, P. Srisuresh, "Network Address Translation – Protocol Translation (NAT-PT)", RFC 2766, February 2000

Albert Cabellos-Aparicio is a PhD student in the Computer Architecture Department at the Universitat Politècnica de Catalunya (UPC). There, he received the engineering degree in Computer Science (2001), his main research topics are Mobility, QoS management and provision and IPv6 transition. Currently, he is working with several mobility protocols, meanwhile starting the PhD he is also developing research work at the Advanced Broadband Communications Centre (CCABA). Besides, he has also taken part in IST projects like LONG (<http://long.ccaba.upc.edu>) and in Spanish research projects like SABA and SAM. Currently he is participating in the E-NEXT NoE and in the EuQoS project. More detailed information can be found at <http://www.ccaba.upc.edu>. <acabello@ac.upc.edu>

Jordi Domingo-Pascual is Full Professor of Computer Science and Communications at the Universitat Politècnica de Catalunya (UPC) in Barcelona. There, he received the engineering degree in Telecommunication (1982) and the Ph.D. Degree in Computer Science (1987). In 1983 he joined the Computer Architecture Department. He is co-founder and researcher of the Advanced Broadband Communications Centre of the University (CCABA) that participated in the Spanish National Host and in the PLANBA demonstrator (1994). He was visiting researcher at the International Computer Science Institute in Berkeley (California) for six months. His research topics are Broadband Communications and Applications, IP/ATM integration, QoS management and provision, traffic engineering, IP traffic analysis and characterisation, group communications and multicast. Since 1988 he has participated in the following projects: RACE projects Technology for ATD (R1022) and EXPLOIT; Spanish Broadband projects (PLANBA) AFTER, TR1 and IRMEM; ACTS projects INFOWIN, MICC, and IMMP; IST projects LONG, ENET and EuQoS; EU VI FP NoE E-NEXT; Spanish research projects CASTBA, MEHARI, SABA, MIRA, SABA2, CARISMA and SAM; and in the research project of an experimental next

generation network in Catalunya i2CAT. More detailed information may be found in:
<http://www.ac.upc.edu/homes/jordid/> and <http://www.ccaba.upc.edu>. <jordi.domingo@ac.upc.edu>