EGEE

REPORT ON IMPLICATIONS OF IPV6 USAGE FOR EGEE GRID

EU DELIVERABLE: DJRA4.3

Document identifier:	**EGEE-DJRA4.3-603955-v4.0**
Date:	**07/11/2005**
Activity:	**JRA4: Network Services Development**
Lead Partner:	**CNRS**
Document status:	**FINAL**
Document link:	**https://edms.cern.ch/document/603955**

Abstract: The EGEE-JRA4 activity has, as one of its sub activities, the task of reporting on the potential benefits that IPv6 can bring to the GRID technologies. The document describes the state of the art of IPv4/IPv6 coexistence and integration, gives an overview of already existing architecture deployment project and studies integration of IPv6 in Grid Context, including gLite. This document attempts to give some pointers to the guidelines to be followed in order to have IPv6 capable software programs.

Delivery Slip

	Name	Partner/Activity	Date	Signature
From	EGEE JRA4	CNRS		
Reviewed by	Moderator: Gerben Venekamp Reviewers: Robert Harakaly Geneviève Romier Sam Wilson	NIKHEF/JRA3 CERN/JRA1 CNRS/NA4 CCed		
Approved by	PEB			

Document Log

Issue	Date	Comment	Author
0-0	08/11/04	First draft for IPv6 features	Edoardo Martelli (CERN)
1-0	20/06/05	First draft of the deliverable	Julien Guignard (CNRS), Benjamin Vial (CNRS)
1-1	22/08/05	Review section 1-5, draft for section 6	Julien Guignard (CNRS), Benjamin Vial (CNRS)
1-2	8/09/05	Contribution to sections 1-5	Jean-Paul Gautier (CNRS)
1-4	12/09/05	Minor changes in sections 1-5	Julien Guignard (CNRS), Benjamin Vial (CNRS)
1-5	13/09/05	Chapter 3 organization	Julien Guignard (CNRS), Jean-Paul Gautier (CNRS)

2.0	16/09/05	Contribution to chapter 6	Benjamin Vial (CNRS), Julien Guignard (CNRS), Jean-Paul Gautier (CNRS)
3.0	22/09/05	More details in § 6 and review	CNRS
3.5	26/09/05	Chapter 6 reviewed. English language reviewed	CNRS, Kostas Kavoussanakis (EPCC)
3.8	29/09/05	Minor modifications	Julien Guignard (CNRS), Jean-Paul Gautier (CNRS), Edoardo Martelli (CERN)
3.9	21/10/05	Modifications after first round review	CNRS
4.0	27/10/05	Final document after review	Benjamin Vial (CNRS), Julien Guignard (CNRS)

Document Change Record

Issue	Item	Reason for Change

CONTENT

1. INTRODUCTION

1.1. PURPOSE OF THE DOCUMENT

This document attempts to describe the potential benefits that IPv6 can bring to the GRID technologies. IPv6 is the "next generation" protocol designed by the IETF to replace the current version Internet Protocol, IP Version 4 ("IPv4"). As IPv6 grows in maturity in terms of standards, implementations and deployment, it is time to study the impact of the usage of IPv6 in a Grid context, focussing in particular on the advantages and difficulties which might be offered to EGEE to become an IPv6 enabled Grid.

The document is organized as follows:

- In section 3 gives an overview of the IPv6 features;
- In section 4 studies the coexistence IPv6/IPv4;
- In section 5 shows the importance of IPv6 in some projects and deployments;
- In section 6 deals with IPv6 issues in a Grid context;
- In section 7 gives a general conclusion.

1.2. APPLICATION AREA

This document applies to all people interested in the IPv6 usage in networks and Grids by giving them an overview of the IPv6 protocol on the Grid, mainly oriented towards both deployment and support aspects.

1.3. REFERENCES

[R1] 6NET Deliverable D5.12	IPv6-enabled Globus Toolkit, June 2004 http://www.6net.org/publications/deliverables/D5.12.pdf
[R2] How-to-IPv6 in GT3	http://www.cs.ucl.ac.uk/staff/s.jiang/webpage/how-to-IPv6-Globus.htm
[R3] Routing IPv6 with IS-IS, IETF	http://www.ietf.org/internet-drafts/draft-ietf-isis-IPv6-05.txt
[R4] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", March 1997.	http://standards.ieee.org/regauth/oui/tutorials/EUI64.html
[R5] RFC 791-Internet Protocol	http://www.faqs.org/rfcs/rfc791.html
[R6] RFC 826-An Ethernet Address Resolution Protocol	http://www.faqs.org/rfcs/rfc826.html
[R7] RFC 1305-Network Time Protocol Network Time Protocol (Version 3) Specification, Implementation and Analysis	http://www.faqs.org/rfcs/rfc1305.html
[R8] RFC 1752-The Recommendation for the IP Next Generation Protocol	http://www.faqs.org/rfcs/rfc1752.html
[R9] RFC 1918-Address Allocation for Private Internets	http://www.faqs.org/rfcs/rfc1918.html
[R11] RFC 2428–FTP Extensions for IPv6 and NATs	http://www.faqs.org/rfcs/rfc2428.html

[R12] RFC 2460-Internet Protocol, Version 6 (IPv6) specification	http://www.faqs.org/rfcs/rfc2460.html
[R13] RFC 2461-Neighbour Discovery for IP Version 6	http://www.faqs.org/rfcs/rfc2461.html
[R14] RFC 2462-IPv6 Stateless Address Autoconfiguration	http://www.faqs.org/rfcs/rfc2462.html
[R15] RFC 2663-IP Network Address Translator (NAT)	http://www.faqs.org/rfcs/rfc2663.html
[R16] RFC 2732-Format for Literal IPv6 Addresses in URL's	http://www.faqs.org/rfcs/rfc2732.html
[R17] RFC 2993-Architectural Implications of NAT	http://www.faqs.org/rfcs/rfc2993.html
[R18] RFC 3022-Traditional IP Network Address Translator (Traditional NAT)	http://www.faqs.org/rfcs/rfc3022.html
[R19] RFC 3315-Dynamic Host Configuration Protocol for IPv6 (DHCPv6)	http://www.faqs.org/rfcs/rfc3315.html
[R20] RFC 3031-Multiprotocol Label Switching Architecture	http://www.faqs.org/rfcs/rfc3031.html
[R21] RFC 3493-Basic Socket Interface Extensions for IPv6	http://www.faqs.org/rfcs/rfc3493.html
[R22] RFC 3513-Internet Protocol Version 6 (IPv6) Addressing Architecture	http://www.faqs.org/rfcs/rfc3513.html
[R23] RFC 3542–Advanced Sockets Application Program Interface (API) for IPv6	http://www.faqs.org/rfcs/rfc3542.html
[R24] RFC 3569-An Overview of Source-Specific Multicast	http://www.faqs.org/rfcs/rfc3569.html
[R25] RFC 3697-IPv6 Flow Label Specification	http://www.faqs.org/rfcs/rfc3697.html
[R26] RFC 3736-Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6	http://www.faqs.org/rfcs/rfc3736.html
[R27] RFC 3775-Mobility Support in IPv6	http://www.faqs.org/rfcs/rfc3775.html
[R28] On-demand VPN Support for Grid Applications	http://www.cnaf.infn.it/~ferrari/papers/myarticles/chep2004-vpn-v5.pdf
[R29] RFC 2526-Reserved IPv6 Subnet Anycast Addresses	http://www.faqs.org/rfcs/rfc2526.html
[R30] RFC 3956-Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address	http://www.faqs.org/rfcs/rfc3956.h

[R31] RFC 2080-RIPng for IPv6	http://www.faqs.org/rfcs/rfc2080.html
[R32] RFC 2740-OSPF for IPv6	http://www.faqs.org/rfcs/rfc2740.html
[R33] RFC 2858-Multiprotocol Extensions for BGP-4	http://www.faqs.org/rfcs/rfc2858.html
[R34] RFC 2545-Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing	http://www.faqs.org/rfcs/rfc2545.html
[R35] RFC 2893-Transition Mechanisms for IPv6 Hosts and Routers	http://www.faqs.org/rfcs/rfc2893.html
[R36] RFC 3056-Connection of IPv6 Domains via IPv4 Clouds	http://www.faqs.org/rfcs/rfc3056.html
[R37] RFC 3053- IPv6 Tunnel Broker	http://www.faqs.org/rfcs/rfc3053.html
[R38] RFC 2766-Network Address Translation - Protocol Translation	http://www.faqs.org/rfcs/rfc2766.html
[R39] EGEE Middleware Architecture. DJRA1.1	https://edms.cern.ch/document/476451
[R40] Guidelines for IP version independence in GGF specifications	http://www.ggf.org/documents/GFD.40.pdf
[R41] Survey of IPv4 Dependencies in Global Grid Forum Specifications	http://www.ggf.org/documents/GFD.41.pdf
[R42] RFC 2553- Basic Socket Interface Extension for IPv6	http://www.faqs.org/rfcs/rfc2553.html
[R43] TERENA compendium	http://www.terena.nl/compendium/
[R44] IPv6 Monitoring tools	http://tools.6net.org
[R45] draft-ietf-ngtrans-bgp-tunnel-04	http://www.ietf.org/proceedings/02mar/I-D/draft-ietf-ngtrans-bgp-tunnel-04.txt
[R46] How-to IPv6 in Globus Toolkit 4	http://www.cs.ucl.ac.uk/staff/sjiang/webpage/How-to-IPv6-in-GT4.htm
[47] Web Services Resource Framework	http://www.globus.org/wsrf

1.4. DOCUMENT AMENDMENT PROCEDURE

This document can be amended by the JRA4 Team (http://egee-jra4.web.cern.ch/EGEE%2DJRA4/team.htm). Proposals for amendments can also be sent to JRA4 (project-eu-egee-jra4@cern.ch) with a brief description of the proposed change and its benefits. Minor changes, such as spelling corrections, content formatting or minor text reorganization not affecting the content and meaning of the document can be applied by the authors without review. Other changes must be submitted for review by the JRA4 team.

1.5. TERMINOLOGY

Glossary

6PE	IPv6 Providing Edge
ACL	Access Control List
ADSL	Asymmetrical Digital Subscriber Line

AH	Authentication Header
AIIH	Assignment of IPv4 global addresses to IPv6 Hosts
ALG	Application Level Gateway
ANL	Argonne National Laboratory
API	Application Program Interface
ARP	Address Resolution Protocol [RFC826]
BAR	Bandwidth Allocation and Reservation
BGP	Border Gateway Protocol
BIA	Bump In the API
BIS	Bump In the Stack
BSD	Berkeley Software Distribution
CoS	Class of Service
CPU	Central Processing Unit
DHCP(v6)	Dynamic Host Configuration Protocol (version 6)
DNS	Domain Name Server
DNS-ALG	Domain Name Server – Application Layer Gateway
DSL	Digital Subscriber Line
DSTM	Dual Stack Transition Method
DTI	Dynamic Tunneling Interface
EDG	European DataGrid
ESP	Encapsulating Security Payload
EUI-64	Extended Unique Identifier, 64bits [EUI64]
GÉANT	Trans European network-2.5 to 10 Gbit/s backbone
GGF	Global Grid Forum
gLite	Lightweight Middleware for Grid Computing
GRE	Generic routing encapsulation
GT	Globus Toolkit
GUI	Graphical User Interface
GSI	Grid Security Infrastructure
FTP	File Transfer Protocol
GIX	Global Internet eXchange
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPC	Inter-Process Communication
IPSec	IP security [RFC2401]

IPv4	Internet Protocol version 4 [RFC791]
IPv6	Internet Protocol version 6 [RFC2460]
ISATAP	Intra-Site Automatic Tunnel Addressing Protocol
IS-IS	Intermediate System to Intermediate System
ISO	International Standards Organization
ISP	Internet Service Provider
JDK	Java Development Kit
LAN	Local Area Network
LCG	LHC Computer Grid project
LHC	Large Hadron Collider
MAC address	Media Access Control address
MIB	Management Information Base
MP-BGP	Multi Protocol Border Gateway Protocol
MPLS	Multi Protocol Label Switching [RFC3031]
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NAT-PT	Network Address Translation – Protocol Translation
NFS	Network File System
NREN	National Research and Education Network
NU	Neighbour Discovery
OGSA	Open Grid Services Architecture
OSI	Open System Interconnection
OSPF	Open Shortest Path First
QoS	Quality of Service
RFC	Request For Comments
RIP	Routing Information Protocol
RPC	Remote Procedure Call
SIIT	Stateless IP/ICMP Translation Algorithm
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SSH	Secure SHell
SSL	Secure Socket Layer
SSM	Source Specific Multicast
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TRT	Transport Relay Translator
TSP	Technical Service Provider

UCL	University College of London
UDP	User Datagram Protocol
UoS	University of Southampton
URI	Universal Resource Identifier
URL	Universal Resource Locator
VLAN	Virtual Local Area Network
VoIP	Voice over IP
VPN	Virtual Private Network
WDM	Wavelength Division Multiplexing
WS-RF	Web Services Resource Framework
XIO	eXtensible Input Output library

Definitions

Layer 2	Data Link layer of the ISO OSI model
Layer 3	Network layer of the ISO OSI model
IPv4-only node	A host or router that implements only IPv4. An IPv4-only node does not understand IPv6. The installed base of IPv4 hosts and routers existing before the transition begins are IPv4-only nodes
IPv6/IPv4 node	A host or router that implements both IPv4 and IPv6
IPv6-only node	A host or router that implements IPv6, and does not implement IPv4. The operation of IPv6-only nodes is not yet addressed here
IPv6 node	Any host or router that implements IPv6. IPv6/IPv4 and IPv6-only nodes are both IPv6 nodes
IPv4 node	Any host or router that implements IPv4. IPv6/IPv4 and IPv4-only nodes are both IPv4 nodes
H323	A set of protocols for audio and video

2. EXECUTIVE SUMMARY

The aim of this document is to conduct a study on the implication of the new Internet Protocol (IPv6) in a Grid context and more particularly for EGEE Grid. Since this first recommendation in 1995, the new IP protocol comes to maturity now. IPv6 is expected to gradually replace IPv4, with cohabitation for a number of years during a transition period.

To plan for the adoption of this protocol by the EGEE project, the Technical Annex requests:

- To study the features of IPv6 and seeking those that are interesting for Grids;
- To consider availability of IPv6 in GEANT, the NRENs and Access networks;
- To evaluate the impact of IPv6 on applications and middleware.

This survey on IPv6 support for Grid has been realised in a networking activity with a network-oriented scope and basically not in a development perspective. The studies of the IPv6 features and IPv6 in NRENs are complete. But, the work on IP-dependencies of the Grid components (Middleware, applications, equipments etc.) should be seen only as a first survey.

An overview of the IPv6 features is done, IPv6 advantages which could be interesting to the Grid are emphasized. Address format have changed a lot and extension of addresses length allows restoring end-to-end connectivity. IPSec and IP mobility protocols are foreseen in the core specification of the protocol. The security protocol includes a secure authentication at IP-packet level with IPSec. For the network management, IPv6 introduces a hierarchy in the addressing plans and provides automatic configuration mechanisms.

An overview of the cohabitation mechanisms between IPv4 and IPv6 is provided, this coexistence is necessary. Dual stack, tunnels, and translation mechanisms are briefly described on a theoretical basis. All these mechanisms are operational and currently used in IPv6 deployed networks.

To give some ideas on interest of this protocol in the world, an overview of three major IPv6 projects will be show, pointing out efforts accomplished within their time-frame, a clear understanding of the technical issues they raised and the network infrastructure implemented. Moreover an overview of Internet Service Providers deployments will draw up to understand the impact of IPv6 outside research and educations networks. It is obvious that applications like voice or peer to peer and the needs of new countries, China, India will push IPv6 in the networks.

Because of the crucial need of IPv6 in Grid context in some environment, for instance Asian Grids, efforts have been made to enable IPv6 for Grids. The work from the Grid standardisation authority (Grid Global Forum) and Grid projects is presented; for applications, middleware the Global Grid Forum has produced recommendations to avoid IP-dependencies.

This document introduces the notion of IP-dependencies and their study. A Grid includes applications, middleware, equipments and network infrastructures. Each of these various elements does not interact in same way with the IP protocol. Most of programming languages have been modified allowing the application portability. A first survey on IP-dependencies of gLite middleware is done showing that some modules require a deeper code analysis. As the roll-out of IPv6 is at different stages in different networks across Europe, a survey of the IPv6 deployment is done to assess the match between GEANT/NRENs coverage and the topology of EGEE. The involvement of research networks in IPv6 projects has often pushed NRENs to deploy IPv6.

From a network point of view, including infrastructure deployment, services, programming, the document shows that there is no obstacle to tackle an integration of IPv6 in EGEE.

3. IPV6 FEATURES

IPv6 was designed in order to solve some limitations of IPv4, especially the address space exhaustion. But this was not the only improvement, and some other features have been added. Now, there are a lot of debates about how these new features are really necessary and if they can actually be ported to IPv4. With some workarounds, IPv4 can actually be improved, but on the long term these tricks show some side effects that require new workarounds. A typical example is Network Address Translation (NAT) that successfully solved the address space exhaustion issue, but that in practice introduced end to end unreachability. IPv6 will solve many issues in a cleaner way.

Unfortunately, for the time being, IPv6 has some deficiencies due to its youth. So far, some specifications have not already been standardised and some mandatory functionalities have not been widely implemented yet (like IPSec, or SNMP MIBs). However, it is progressing at a fast pace.

3.1. IPV6 ADDRESSING

The IP address is lengthened from 32 bits in IPv4 to 128 bits in IPv6. Three type of addressing are described in IPv6:

- Unicast, one host to one other host, the classical use;
- Anycast, one host to the nearest of multiple hosts;
- Multicast, one host to multiple hosts.

3.1.1. Extended address space

The IPv6 address is 128 bits long, versus the 32 bits of the IPv4 address. The IPv6 addressing architecture is specified in RFC3513 [R22]. This increases enormously the number of available addresses, dispelling the worries of system administrators that always had to assign IPv4 addresses to end users.

This extended address allows organising addresses into a hierarchy (Figure 1). Aggregation and tree structure organisation for prefixes affect positively the complexity of routing table and make the work of routers more efficient.

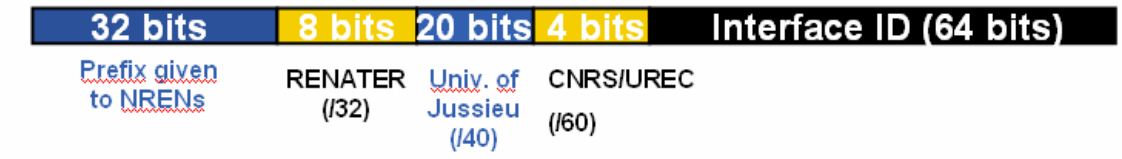

Figure 1. Hierarchical organization of a /64 network prefix

The official recommendation is to assign a /64 network prefix for each Local Area Network (LAN), that means that every segment can host up to 2^{64} end hosts. The purpose of this apparently waste of this address-space is to allow every host to assign to itself a unique address, being quite sure it will not clash with anyone else on the link (more regarding this in the "Automatic configuration" paragraph 3.2.1). Another recommendation suggests to assign at least a /48 network prefix (that means 2^{16} /64 networks) to every customer that needs to create subnets. Although some discussions are on going to reduce this standard allocation to /56, fearing an other waste of space as it happened for Ipv4. Considering these aspects, we can assume that all the problems related with having to manage an almost depletion resource like the IPv4 addresses are over: careful plans for sharing small chunks of IP networks, impossible forecasts about future growing, painful address renumbering, addresses recycling reason of Data Name Server (DNS) or services inconsistencies, everything should be over.

3.1.2. Anycast

In IPv6 networks, a service-oriented address can be assigned to an interface. This address is called anycast address which identifies a set of interfaces. An anycast address [R29] can be assigned to more than one interface typically belonging to different nodes. A packet sent to an anycast address is delivered to the nearest interface having this address. “Nearest” address is chosen by measurement of routing protocol. Anycasting is designed for an efficient updating of router tables.

3.1.3. Enhanced Multicast

In a network designed for multicast, when a host (multicast source) must send an IP packet to multiple receivers, only one IP packet (multicast packet) is send to the receivers (multicast group). By example broadcasting server bandwidth issue is resolved by the multicast, only one flow is sent by the broadcast server toward all the receivers who ask for it.

IPv6 has always considered multicast as a prominent functionality and multicast is now fully integrated in the protocol. Therefore, the IPv4 broadcast function has been replaced with multicast, and all the ambiguities that IPv4 multicast included due to the reduced address space, have been removed.

Most of the multicast routing protocols and discovery mechanisms have been derived from the IPv4 versions, but some of them are still in development. Embedded-RP [R30] is the preferred model. Another model, Source Specific Multicast (SSM) exists but with a few available applications [R24].

Multicast is mainly used for video and sound distribution; it can be used also for file transfers.

3.1.4. End to end addressing

An important consequence of the extended address space is the possibility to restore a clean end-to-end connectivity between each couple of hosts connected to the Internet, which was so far broken by the Network Address Translation mechanism (NAT) [R15]. NAT was implemented to avoid lack of IPv4 addresses by using the IP addresses given in RFC 1918 [9].

NAT is also used to masquerade single IP addresses in some applications behind one or more public IP addresses and to avoid renumbering of IP networks. It is done by network equipment (a router or a firewall) which maps every connection to different ports of the same public IP address and maintains a table of correspondences. This is easily done for outgoing connection, but prevents incoming connections unless specific configurations exist. In any case NAT with a single IP address does not allow having more than one server answering to the same port.

Particular implementations of NAT may be incompatible with applications that map the used IP addresses at upper layers, like it was the case for H323, a set of protocols for audio and video.

Furthermore, NAT has initiated a false feeling of security to system administrators, which were conducted to not adopt more reliable forms of security (like firewalls, regular software upgrades, closing unnecessary services, etc...).

End-to-end connectivity creates new opportunities for the use of multicast utilization. Everyone will be reachable, and every user will be able to address data to an unlimited number of receivers.

With IPv6, all these issues are over. This is very important for Grid applications and services:

- With this extended address space in IPv6, there are more public addresses available on a network;
- Protocols and services do not have to take NAT into account but can rely on a pure Layer 3 connectivity;
- Networks configuration is somewhat easier, because NAT does not have to be implemented nor debugged, and also because servers do not require ad-hoc NAT settings;

- There is no need for powerful NAT gears to deal with high speed data transfers;
- Without NAT, one single point of failure disappears.

3.2. IPV6 AND NETWORK

IPv6 provides an evolutionary set of improvements in the host configuration and in the packet header. IPv6 supports all the routing protocols, Internal Gateway Protocols and External Gateway Protocols. With the service classes, packets can be identified as belonging to a particular flow and prioritized along a data path.

3.2.1. Automatic configuration

There are two main mechanisms for autoconfiguring an IPv6 host:

1. A stateless mechanism which is only available in IPv6. IPv6 stateless address autoconfiguration is described in RFC2462 [R14]. The end host builds an Extended Unique Identifier address [R4] for the end part of address (last 64 bits) and uses the network prefix provided by a router on the LAN;
2. A stateful mechanism using the Dynamic Host Configuration Protocol (DHCP) with a DHCP server derived from IPv4. IPv6 stateful address autoconfiguration (DHCPv6) is described in RFC3315 [R19]. DHCPv6 gives more control on address assignment mechanism, making it possible to assign IP addresses based on the end host MAC address. Also, more information can be provided to the host, like the DNS server address, the domain name of belonging, and others that can be defined later.

Stateless autoconfiguration can also be extended to autoconfiguring more than just the IP address. Stateless DHCPv6 presented in RFC3736 [R26] can also be used for this purpose. It is a service that does not provide the address for the end host, but only additional information like DNS server addresses or Session Initial Protocol (SIP) servers.

IPv6 nodes use Neighbour Discovery [R13] to discover each other's presence, to determine each other's link-layer addresses, to find routers and to maintain reachability information about the paths to active neighbours in a local environment. Neighbour Discovery is the IPv6 successor of the Address Resolution Protocol (ARP) in IPv4. You can retrieve information about the current neighbours; in addition you can set and delete entries. This protocol uses specific unicast addresses for each IPv6 host on the same link (end hosts connected together on the same router on the LAN); they are valid only on that link. In summary an IPv6 host has more than one address:

1. At least one unicast address (network interface, transition mechanisms);
2. A link-local address used by the autoconfiguration mechanism;
3. A localhost address defines the host itself.

3.2.2. Simplified header format with clean extensibility

The IPv6 header has been simplified (Figure 2), compared to the IPv4 header: seldom used fields have been removed, in order to accommodate the impact of the increased space required by the longer address (128 bits versus 32 bits). But functionalities have been preserved and even better extended, thanks to the possibility of adding more nested headers. This feature is called IPv6 Extension Headers and is described in [R12] chapter 4.

A few of these optional headers are already defined (Hop-by-Hop options, Routing, Fragment, Destination options, Authentication, Encapsulating Security Payload), but the mechanism is extensible and allows the definition of new headers and thus the addition of more functionalities into the protocol.

Most extension headers are not processed by any router along a data path until the packet reaches its destination. This improves the router performance for IP packets that contain options; with IPv4 the presence of options requires the router to examine all of them.

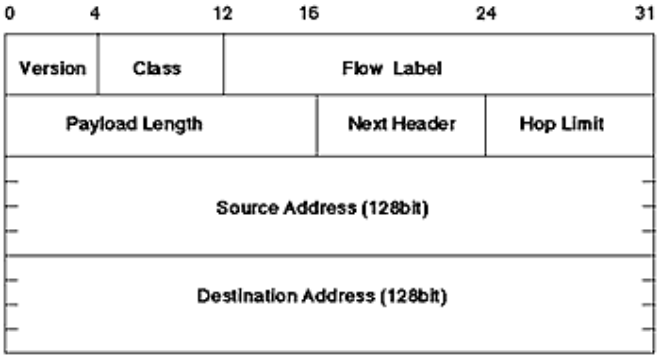

Figure 2. IPv6 header

3.2.3. Quality of service

As it is the case in IPv4, IPv6 packets can be classified and prioritized under the DiffServ model. It allows the use of the same Classes of Service (CoS) as IPv4.

Moreover, the IPv6 header hosts a 20 bit long field called Flow Label, described in [R12] chapter 6 and in [R25]. This label is intended to identify "a sequence of packets sent from a particular source to a particular destination for which the source desires special handling by the intervening routers". This allows the sender to prioritize only well defined subsets of the data that it is generating, even inside a single application; for example for prioritizing control signals rather above bulk data.

3.2.4. Routing

The routing algorithms have not been changed fundamentally with the IPv6 introduction. The organisation into a hierarchy of addresses specific to the IPv6 model aims to aggregate as efficiently as possible the routing tables of the routers. Moreover, the extension header function gives the possibility to force a specific route (only for unicast addresses not for multicast addresses).

Internal Gateway Protocol:

- Routing Information Protocol next generation (RIPng) ([R31] for RIPng v1) is based on (but incompatible with) RIPv2. For this reason, operational procedures, timers and stability functions remain the same. The message format was changed to carry larger IPv6 addresses;
- Open Shortest Path First (OSPF) v.3 [R32] is OSPF for IPv6. It keeps fundamental OSPF mechanisms and algorithms such as basic packet type or Neighbour Discovery, but it runs now directly over IPv6 and distributes IPv6 prefix. In dual environment, both version of OSPF (v2 for IPv4 and v3 for IPv6) must be used;
- Intermediate System to Intermediate System (IS-IS) has passed to IPv6 with only two news TLVs (Type/Length/Value), a reachability TLV and an interface address TLV to distribute the necessary IPv6 information throughout a routing domain [R3]. IS-IS works with both IPv4 and IPv6.

External Gateway Protocol:

- Border Gateway Protocol (BGP) v.4+ [R33] is a new version for IPv6 and multicast IPv4 routes;

- Multi Protocol Border Gateway Protocol (MP-BGP) [R33] is an extension to BGP protocol to allow compatibility with other routing protocols. MP-BGP for IPv6 [R34] manages IPv6 features.

In conclusion, the versions of the routing protocols are updated to adapt to IPv6 and general functionalities are the same.

3.3. NATIVE PROTOCOLS IN IPV6

In the Core Specifications of IPv6 two protocols are mandatory: IPSec for security; and mobility. Together with the address extension they are the most important improvements.

3.3.1. Security

3.3.1.1. Full End-to-End security with IPSec

IPv6 will make network administrators aware of the fact that their hosts are definitely exposed to everybody on the network. Thus, they will be forced to implement effective security policies. IPv6 already integrates the IPSec specifications [R12].

In IPv4, IPSec is optional and is generally implemented between pairs of devices placed at the border of the networks that need to be connected; these devices are usually routers or firewall or Virtual Private Network (VPN) concentrators, and imply that there are parts of the path where the data travel in an unencrypted manner.

With IPv6, IPSec is part of the protocol specifications and the support of the Authentication Header (AH) and the Encapsulating Security Payload (ESP) Header is mandatory.

The main advantages of using IPSec within IPv6 are:

- A complete end-to-end security, with data that travel encrypted from the source to the destination, even for users;
- A spread of the encrypting effort among all the hosts involved, rather than being delegated to few dedicated devices. This reduces single points of failure, and avoids the commission of powerful equipments, dedicated to high throughput encryption.

3.3.1.2. More security considerations

There are more enhancements that IPv6 can provide to network security:

- For link layer to network layer address mapping, IPv6 does not use Address Resolution Protocol (ARP), but the Neighbour Discovery protocol that relies on multicast;
- An IPv6 address has available bits to allow embedding unique identifiers in it.

3.3.2. IP Mobility

IP Mobility allows hosts to move around the Internet keeping their original IP addresses. This can be useful for a simplification of security procedures that allow hosts to use remote facilities.

IPv4 mobility was introduced several years later after the definition of the IPv4 protocol, and further mechanisms for mobility remain to be achieved. IPv6 has been defined taking into account mobility, that results in a more clear and optimal implementation. IPv6 mobility support is described in RFC3775 [R27], and its benefits are described in the second chapter of this RFC.

Even more, since Mobility will be handled at the IP level, applications developers are not required to explicitly consider it.

3.4. IPV6 ADVANTAGES

There are no disadvantages or constraints in comparison of IPv4 protocol other than efforts of transition and deployment, the main problems are due to implementation. In addition to the basic goal of making more addresses available, several other features have been improved and new functionalities were added to IPv6. Some of advantages listed here could benefit the Grids:

- Address space extension implies IP repartition into a hierarchy (Figure 1) and gives better efficiency for routing (better route aggregations allow a huge decrease of the size of routing tables).
- With restoration of end-to-end IP connectivity (temporarily affected by the advent of NAT), a large number of problems were removed. Benefiting several applications. Moreover, end-to-end network can use security functionalities.
- Configuration is simpler with automatic configuration mechanisms. The stateless mechanism could turn to be extremely useful for the Grid, especially with the very fashionable layer 2 connections between remote sites, sometime called LightPath. When network engineers have to set up this type of connections, the addressing problems always rise with IPv4: two remote sites have to agree in using the same network address, and also a non trivial address plan has to be defined, especially when public addresses are used. Stateless autoconfiguration solves this problem by avoiding any manual configuration of the end hosts, avoiding negotiation among remote sites administrators, and requiring a very simple network configuration. Every host connecting to an IPv6 network has to be able to assign a valid and unique IPv6 address to the used interface. Even in the case where no routers on the Layer 2 segment can answer to the router solicitation requests, a simple Link-Local address [R14] will always be defined, allowing communication with the other nodes on the segment. Moreover stateless DHCPv6 can be extended to advertise network services but can be use in the Grid (DNS servers, SIP servers, etc…).
- Security is much improved, but even if security offered by IPv6 does not solve all security issues, it is possible to combine IPSec and Grid Security Infrastructure (GSI) to give better security control. In Global Grid Forum (GGF) specifications, GSI is responsible for establishing the identity of users or services (authentication), protecting communications, and determining who is allowed to perform what actions (authorization), as well as with supporting functions such as managing user credentials and maintaining group membership information. If GSI use end-to-end encryption of IPSec, it is no longer necessary to use SSL over IPSec. Of course after establishing a connection with IPSec, Grid authentication and authorization continue to enable a robust security. There are no clear and definitive answers on these specific security points without experimentations:
    1. Does the fact that IPv6 introduces mandatory IPSEC have any implication for Grid applications?
    2. Does using IPv6 make Grid systems any more or less secure than using IPv4?
    3. Does using IPv6 for Grid traffic rather than IPv4 make it easier or more difficult:
        - to negotiate with a site firewall administrator to allow traffic to and from Grid systems?
        - to configure a firewall protecting Grid systems in a secure manner?
        - to be confident that a Grid system is secured satisfactorily?
- Extension headers offer the possibility to add information available to end-point or hop routers.
- Service classes can be used to differentiate flows; using the same mechanism as with IPv4. The work done by the Network Resource Provision activity (SA2) and in Bandwidth

Allocation and Reservation in JRA4 can be used in an IPv6 context. The Flow Label field and the extended headers could be particularly useful to these activities, as well as applications and middleware.

- The multicast protocol was finalized with the arrival of IPv6 and all ambiguities of the IPv4 multicast model were removed. Moreover this protocol can be useful for file transfer without an overload of the network. One packet sent by a multicast source can be delivered to several destination hosts with a lower traffic on the network.
- Mobile IPv6 is an important feature addressing new needs from mobile users and 3G mobile phones. Thinking of a Grid of computer resources, security policies will be extremely simplified if their access is restricted to some predefined IPv6 prefixes. IP Mobility will allow end-users to keep using the Grid even if they have to move their hosts into another network.

4. IPV6/IPV4 COHABITATION

In a cohabitation context where Ipv6 is meant to progressively integrate the IPv4 world, it is important to study suitable mechanisms capable of facing two different sets of problems:

- The first one is related to IPv6 communications among two or more IPv6 islands or nodes (routers or hosts) isolated in the IPv4 world. We will talk about existing transition mechanisms;
- The second set is related to communications between the IPv4 nodes and IPv6 nodes. We will detail translation methods already deployed.

The IPv6 Operations (v6ops) working group at IETF is in charge of the transition to IPv6. The most well-known and useful mechanisms will be describe in this document. Migration mechanisms problems will be raised for each method.

4.1. TRANSITION MECHANISMS

In this section, we will make a survey of existing transition mechanisms, including:

- Dual IP layer. Providing a complete support for both IPv4 and IPv6 in hosts and routers;
- IPv6 over IPv4 tunneling. Encapsulating IPv6 packets within IPv4 headers to carry them over IPv4 routing infrastructures.

4.1.1. Dual-stack Type mechanisms

This is a technique for providing a complete support for both Internet protocols (IPv4 and IPv6) in hosts and routers by integrating IPv6 itself. There are no real transition mechanisms to use within the dual-stack scenario. Platforms should just be switched to IPv6 to make a node a dual stack one.

Using this method, a host or a router is configured with both IPv4 and IPv6 protocol stacks in the operating system (Figure 3). Those dual stack hosts defined in RFC2893 [R35], need applications, TCP/IP modules and addresses for both IPv4 and IPv6.

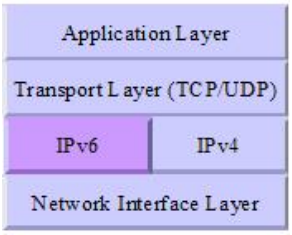

Figure 3. A dual IP layer architecture

For a host, an IPv6-compliant application communicates with IPv4 nodes through the IPv4 stack, and with IPv6 nodes through the IPv6 stack (Figure 4). The selection of the stack can be deduced from the IP destination address (IPv4 or IPv6) given, for example, by the user or as the result of a DNS resolution. For a router, IPv4 packets are forwarded by the IPv4 processes and IPv6 packets are forwarded by the IPv6 processes.

Since IPv6 is expected to become the core protocol for the next generation networks, Grid computing systems must typically track the transition of the lower-layer network protocols. However, the period of transition from IPv4 to IPv6 will not be short as it is quite delicate to prepare such an amount of hardware and software for a hard cut-over in a short time. Another reason for the delay is that the interconnection of various network with distinct administration controls slow down IPv6 migration.

Hence it is important to make Grid systems work with both IPv4 and IPv6, and to be able to communicate in heterogeneous IPv4/IPv6 networks. In this case, the IP version-independent server has to be able to respond to client calls according to the IP version that the client uses. So, the client decides which version of IP is to be used. For instance, an IP version-independent Grid server on a dual-stack machine starts and listens on both its IPv4 and IPv6 interfaces. For example when an IPv4 client connects over IPv4, the Grid server uses its IPv4 interface to call back and only IPv4 communication takes place; the same for IPv6. Fundamental network services should then be dual-stack as well: HTTP, DNS, SSL, routing etc.

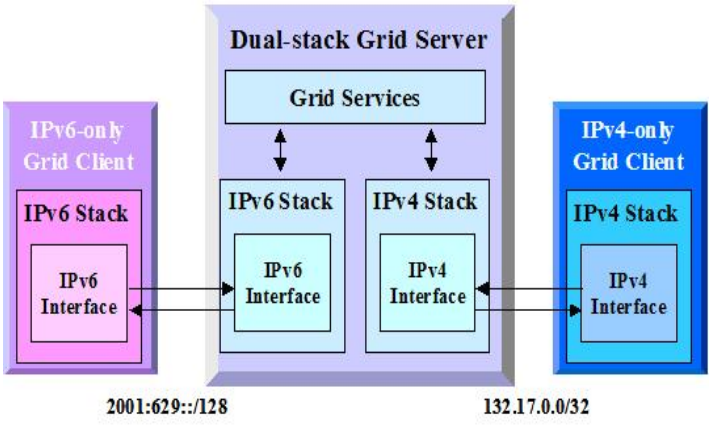

Figure 4. IP communication of Client/Server in Simple Heterogeneous IPv4/IPv6 Networks

This mechanism just needs installation of the IPv6 stack, and many implementations are now available for any kind of hosts or routers. The dual IP layer technique may or may not be used in conjunction with the IPv6-over-IPv4 tunneling techniques (§4.1.2).

An issue with dual stack mechanism is the allocation an IPv4 address for each new IPv6-enabled device. It could be a problem with the lack of IPv4 addresses.

The second drawback of this approach is obviously the overwhelming management load for network administrators who may need to deploy two security policies, one for each protocol.

4.1.2. Tunnel type mechanisms

Tunneling provides a way to utilize an existing IPv4 routing infrastructure (a core network or the Internet) to carry IPv6 traffic by encapsulating IPv6 packets in IPv4 packets. The IPv6 clouds can be interconnected without any IPv6 native interconnection and without upgrading the IPv4 infrastructure.

All tunneling mechanisms require that the endpoints of the tunnel run both IPv4 and IPv6 protocol stacks. The dual-stack routers run both IPv4 and IPv6 protocols simultaneously and thus can interoperate directly with both IPv4 and IPv6 end systems and routers.

For proper operation of the tunnel mechanisms, appropriate entries in a DNS that map between host names and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

The IETF NGTRANS (Next Generation Transition) working group has come up with several tunneling mechanisms, such as configured tunnels, Tunnel broker, 6to4, Generic Routing Encapsulation GRE tunnels and Cisco 6PE which are described in this section.

The following general issues apply to these mechanisms:

- Because they involve encapsulation, decapsulation and tunnel management, more time, bandwidth and other resources will be needed. Therefore the performance and efficiency is affected;
- Because most of them are based on IPv4 addresses at the end points, the IPv6 traffic is encapsulated in IPv4 infrastructure, if there is NAT between the end points of the tunnel, the tunnel will not work. This is the case for configured tunnel, 6to4 and tunnel broker discussed below.
- Adding a tunnel header decreases the payload of the packet. The tunnel MTU represents the maximum size of a tunnel packet payload that can be sent through the tunnel without fragmentation. IPv6 mandates Path MTU Discovery mechanism that allows the tunnel entry-node to adopt a particular MTU for the link and avoid fragmentation issues.
- If a IPv6-in-IPv4 tunnel should be set up to a host behind an IPv4 firewall it is necessary to open that firewall for packets with protocol fields 41 (IPv6) and 58 (ICMPv6) at least for the IPv4 address of the host at the remote end of the tunnel, which will be the source of the incoming IPv4 traffic that contains the IPv6 packets. Ideally it should be possible to restrict this opening to only those IPv4 packets destined for the local tunnel endpoint, limiting the connection to the agreed endpoints. This will leave the site relatively protected concerning IPv4.
- Packet filtering requires either that a firewall look inside the packet payload or that the filtering is done on the tunnel endpoints. In those environments in which this is considered to be a security issue, it may be desirable to terminate the tunnel at the firewall. If a network administrator allows tunnelled traffic at all then:
    - the tunnelling host is no longer protected by the firewall;
    - the tunnel may provide a conduit through the firewall for malicious software or compromising connections.

4.1.2.1. IPv6 manually configured tunnel

A manually configured tunnel [R35] is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The primary use is for stable connections that require regular secure communication (Figure 5).

An IPv6 address is manually configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination.

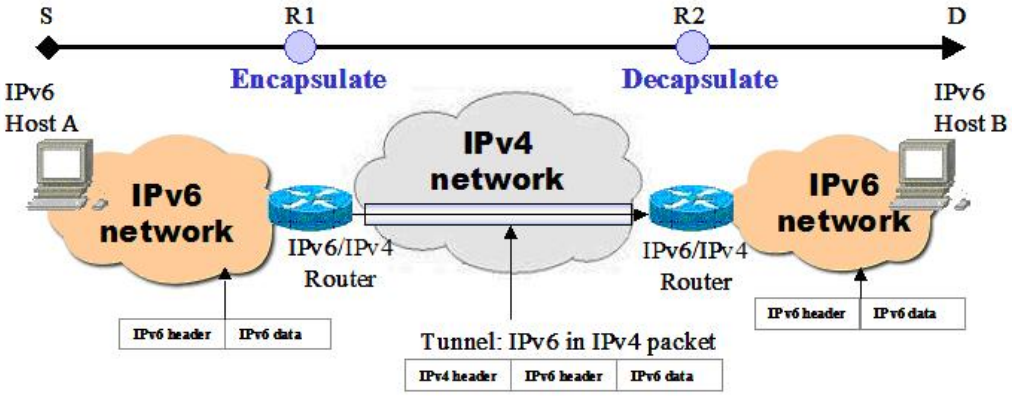

Figure 5. Configured tunneling

Because manually configured tunnels require configuration at both ends of the tunnel, they have a larger management overhead when multiple tunnels are implemented.

4.1.2.2. IPv6 over IPv4 GRE tunnel

IPv6 traffic can be carried over IPv4 GRE (Generic Routing Encapsulation) tunnels using the standard GRE tunneling technique that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme.

GRE tunnels are quite similar to IPv6 manually configured tunnels. In each case, tunnels are links between two points, with a separate tunnel for each link. As it is a point-to-point tunneling protocol, it would be operationally overwhelming to deploy GRE tunneling in a full-mesh configuration, because the number of tunnels would rapidly go out of control. Thus, GRE is ideal in limited, tactical deployments.

Unlike manually configured tunnels, GRE tunnels are not tied to a specific passenger or transport protocol, but in this case carry IPv6 as the passenger protocol over GRE as the carrier protocol. Thus, if the domain is using the Intermediate System-to-Intermediate System (IS-IS) protocol for internal routing, only GRE tunnels can be used. IS-IS runs over a Layer 2 data link, so tunneling techniques other than GRE cannot be used (IPv6-in-IPv4 tunnels are completely Layer 3.) because IS-IS traffic cannot be distinguished from IPv6 traffic.

Security in a network using GRE should be relatively similar to security in a normal IPv4 network, as routing using GRE follows the same routing that IPv4 uses natively.

4.1.2.3. Automatic 6to4 tunnel

The idea is to embed IPv4 tunnel addresses into the IPv6 prefixes so that any domain border router can automatically discover tunnel endpoints for outbound IPv6 traffic (Figure 6).

6to4 is a router-to-router tunneling mechanism which uses prefix 2002::/16, it allows isolated IPv6 domains to be connected over an IPv4 network. An isolated IPv6 site, which wants to communicate with other IPv6 sites via IPv4 infrastructure, will assign itself a prefix of 2002:V4ADDR::/48, where V4ADDR is the global IPv4 address of the IPv6 site's router. An IPv6 over IPv4 tunnel will be established between this IPv6 site's router and another IPv6 site's router. The IPv4 address of tunnel endpoint is determined by the 'V4ADDR' part of the IPv6 destination address contained in the IPv6 packet being transmitted. [R36]

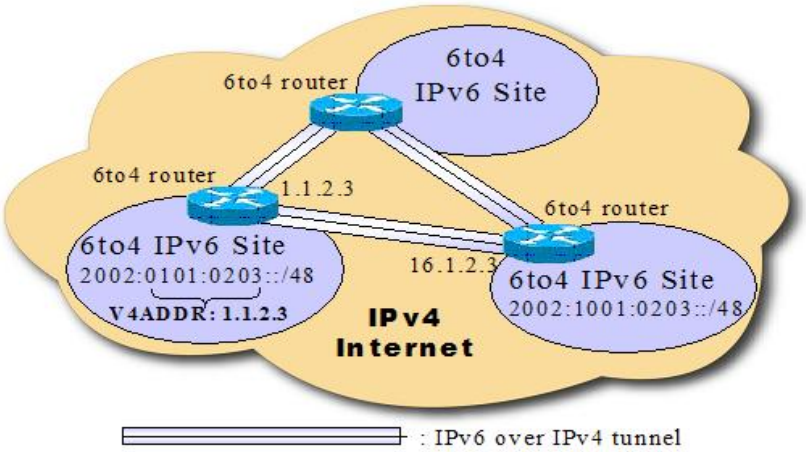

Figure 6. Architecture of 6to4 tunnel mechanism

The mechanism is intended as a start-up transition tool used during the period of co-existence of IPv4 and IPv6. It is not intended as a permanent solution.

It is recommended that each site have only one 6to4 address assigned to the external interface of the router. All sites need to run an IPv6 interior routing protocol, such as Routing Information Protocol next generation (RIPng) for routing IPv6 within the site; exterior routing is handled by the relevant IPv4 exterior routing protocol.

4.1.2.4. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ISATAP is an IPv6 transition mechanism equivalent to the 6to4 mechanisms inside a site. It automatically connects isolated IPv6 hosts or routers (called ISATAP nodes) within an IPv4 site via an automatic IPv6 in IPv4 tunnel.

Each host queries an ISATAP router within the site to obtain address and routing information. Packets sent to the IPv6 Internet are routed via the ISATAP router, and packets destined for other hosts within the same site are tunnelled directly to the destination.

The IPv6 address of ISATAP node has the format as in **Error! Reference source not found.**7. It supports both address autoconfiguration and manual configuration. The IPv4 address of ISATAP link does not need to be globally unique.

64 bits	32 bits	32 bits
Link local, site local or global unicast	0000:5EFE	IPv4 address of ISATAP link

Figure 7. IPv6 address format of ISATAP node [ISATAP]

The typical use of ISATAP is illustrated in Figure 8. The communication between ISATAP hosts and ISATAP routers is done through IPv6-over-IPv4 tunnels.

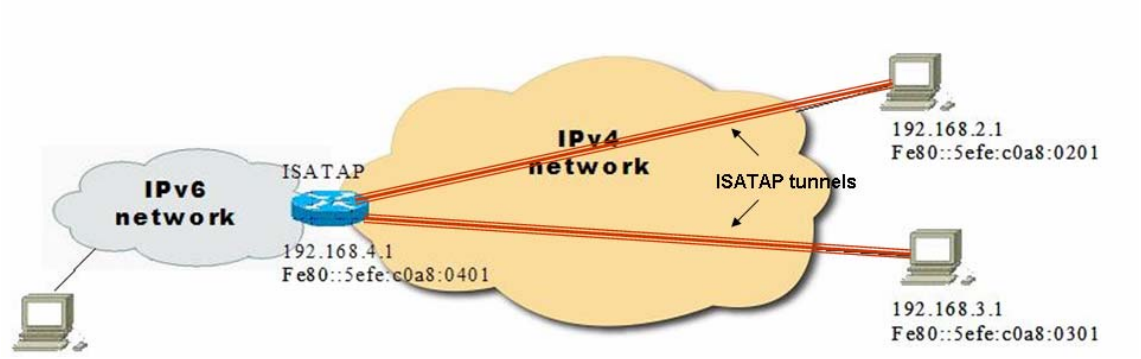

Figure 8. ISATAP scenario

4.1.2.5. IPv6 tunnel brokers

IPv6 Tunnel broker [R37] is an implementation of the Tunnel Setup Protocol (TSP).

The IPv6 tunnel broker fits well with small isolated IPv6 sites, and especially isolated IPv6 hosts on the IPv4 Internet, that want to connect easily to an existing IPv6 network (Figure 9). It can be seen as virtual IPv6 TSPs, providing IPv6 connectivity to users already connected to the IPv4 Internet.

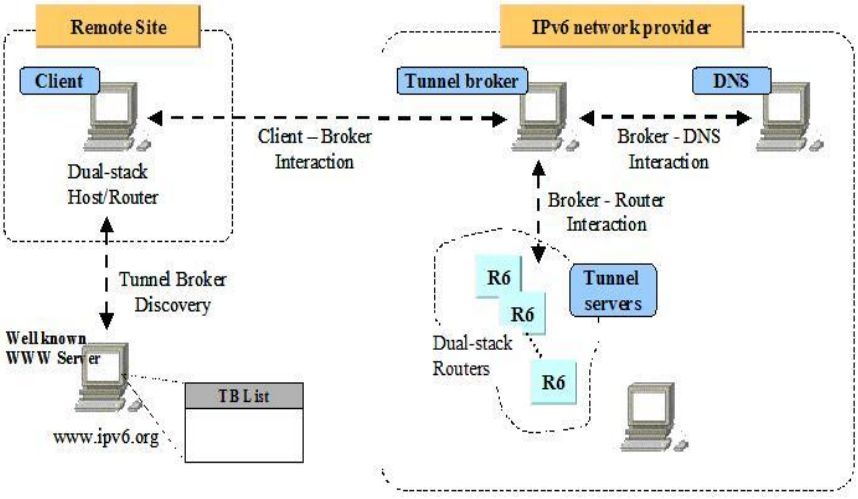

Figure 9. The architecture of Tunnel Broker

The current implementations are web-based tools, which allow interactive setup of an IPv6 over IPv4 tunnel. The created tunnel is between the tunnel client and the tunnel server. Through the tunnel server, the tunnel client can get connected to IPv6 internet and gets assigned IPv6 addresses out of the address space of the tunnel provider, which can be a single address or a network prefix.

The working procedure of tunnel broker is:

1. Dual stack host connects to tunnel broker web server to request tunnel;
2. Tunnel broker web server returns the script to dual stack host, which is used to create tunnel between dual stack host and tunnel broker tunnel server;
3. Dual stack host runs the script, then the tunnel between dual stack host and tunnel broker tunnel server is established;
4. Dual stack host get IPv6 connectivity through tunnel broker to tunnel server.

Tunnel broker does not work if the client uses private IPv4 RFC1918 address [R9] it needs a global unique, routable IPv4 address as client tunnel end point. A client behind a NAT therefore can not use tunnel broker to set up tunnels. It does not matter if the IPv4 address is static or dynamic.

4.1.2.6. 6PE: Deploying IPv6 over MPLS Backbones

IPv6 over MPLS (Multi protocol layer Switching)-enabled backbones allow IPv6 domains to communicate with each other over an MPLS IPv4 core network. IPv6 Provider Edge router (6PE) is Cisco IOS implementation of “BGP Tunnelling” over MPLS [R45]. This implementation requires no backbone infrastructure upgrades and no reconfiguration of core routers, because forwarding is based on labels rather than on the IP header itself. This provides a very cost-effective strategy for IPv6 deployment.

Many service providers have already deployed MPLS in their IPv4 backbone. IPv6 migration does not “need” MPLS, but where MPLS is deployed, it enables attractive approaches for IPv6 migration.

6PE is similar to MPLS VPNs model in terms of technical implementation and complexity:

- Label encapsulation is used for transporting IPv6 packets;
- IPv6 functionality is enabled only at the edge routers. A 6PE router, in simple terms, is an edge router with IPv6 functionality;

- Core routers are IPv6 unaware, a.k.a. support IPv4 only protocols.

4.1.3. Comparison of transition mechanisms

Table 1 summarizes the information on transition mechanisms; it can help to choose a mechanism depending on the use case.

Mechanism Type	Implications	IPv4 address requirements	Scalability	Comments
Dual-stack	Applications have the ability to send and receive both IPv4 & IPv6 packets.	Permanent or Pool of addresses allocated by a DHCP server.	None	Requires an IPv6 compliant Operating System.
Tunnel broker	Applications need to be ported to interface with the IPv6 stack.	One for the dual stack host. At least one for the tunnel broker implementation.	Limitation of the number tunnel supported by the tunnel server. Limitation of the number of IPv6 addresses available to the broker server.	Allows an isolated IPv6 host within an IPv4 only network, to reach an IPv6 wide network.
IPv6 Manually Configured Tunnels	Applications need to be ported to interface with the IPv6 stack.	An IPv6 address is manually configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination.	Because manually configured tunnels require configuration at both ends of the tunnel, they have a larger management overhead when multiple tunnels are implemented.	Tunnels mechanisms generally don't work when traversing a network address translation (NAT). In the case the tunnel is built through a firewall, the latter must be configured ad hoc to permit this kind of traffic.
6to4 (Most commonly deployed automatic tunnel format)	Applications need to be ported to interface with the IPv6 stack.	IPv4 address of border routers.	The scalability and performance of a tunneling mechanism depends on the number of tunnels a device can handle; this metric must be monitored and measured.	Allows to automatically joining IPv6 network separated by an IPv4 only network. Each IPv6 network needs to have a 6to4 border router. No infrastructure change. Has to evolve when many IPv6 clients get connected.
GRE/IPv4 Tunnel Support for IPv6 Traffic	The complicated part is making all of these applications and protocols work together seamlessly.	The edge routers and the end systems must be dual-stack implementations.	The deployment of a GRE tunnel is flexible.	Security in a network using GRE should be relatively similar to security in a normal IPv4 network, as routing using GRE follows the same routing that IPv4 uses

				natively.
6PE	IPv6 packets are transported from 6PE to 6PE inside MPLS that offers different performance levels to applications.	Forwarding is based on labels rather than on the IP header itself.	cost-effective strategy for IPv6 deployment.	No software upgrade or reconfiguration of the MPLS core. It requires MPLS and MP-BGP4. No need to upgrade the Core devices, keep all MPLS features (TE, IPv4-VPN).

Table 1. Transition mechanisms comparison

4.1.4. Conclusion

In this section transition and co-existence scenarios were discussed. There are different paths to adopting IPv6; which makes the most sense often depends on which phase of the transition is in progress and how one expects the future progress to go. Even if for instance 6to4 is seen as the primary connectivity mechanisms, usage scenarios should drive the deployment of IPv6 and the mechanisms adopted and this section was meant to give an overview of the existing solutions by considering and comparing the most basic ones.

What we can say is that when no native IPv6 infrastructure exists between two points but there is IPv4 connectivity, tunneling IPv6 in IPv4 can be used. This is a very common scenario in the early stages of the transition process.

Some mechanisms have not been mentioned here, either because they have not really been implemented (such as 6over4) so that in certain restricted scenarios, the mechanism might have been more practical, or simply because they will not be applicable for some time yet except for some IPv6-only scenarios (as for example Dual-stack Transition Mechanism (DSTM)).

An IPv6/IPv4 node that supports tunneling may support only configured tunneling, or both configured and automatic tunneling. Thus, three modes of tunneling support are possible:

- IPv6/IPv4 node that does not perform tunneling;
- IPv6/IPv4 node that performs configured tunneling only;
- IPv6/IPv4 node that performs configured tunneling and automatic tunneling.

4.2. TRANSLATION TYPE MECHANISMS

When IPv6 islands are installed and connected together using one or several of the previous mechanisms, communication between IPv6 hosts is enabled. The communication between only IPv6 nodes and other nodes located in IPv4-only domains may also need to be established. In this case, a translator must be installed between IPv4 network and IPv6 network. Figure 10 illustrates the basic function and location of translator.

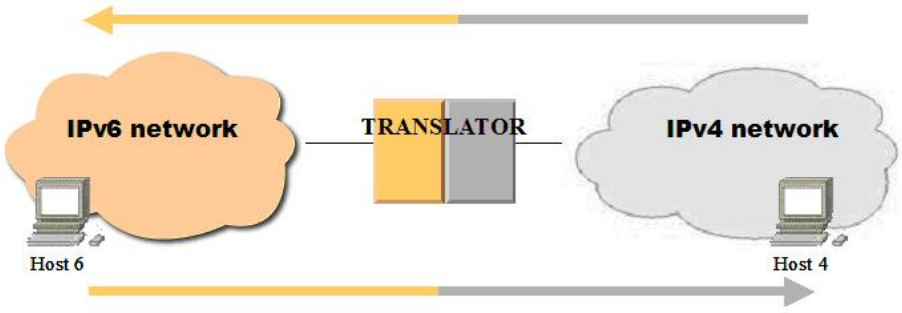

Figure 10. Translation between IPv6 network and IPv4 network

Several translation mechanisms allowing IPv4 and IPv6 devices and networks to establish communication have been proposed:

- SIIT (Stateless IP/ICMP Translation Algorithm) and NAT-PT (Network Address Translation – Protocol Translation) [R38] which work at the network layer.
- Application Layer Gateway (ALG): ALG allows users behind gateways or firewalls to use applications that otherwise are not allowed to traverse gateways and firewalls;
- Bump in the Stack (BIS): BIS is a translation interface between IPv4 applications and IPv6 network infrastructure;
- Bump- in-the-API (BIA): BIA is a translation interface between socket API and TCP/IP modules;
- Socks64: It is a system which accepts enhanced IPv4 socks connections from IPv4 hosts and relays them to IPv4 or IPv6 nodes.

4.2.1. Conclusion

For communication in heterogeneous IPv4/IPv6 networks, there are a number of network transition aids which essentially translate the packet headers between IPv4 and IPv6. They leave the payload untouched and may work in certain circumstances for Grid applications such as when IP addresses are not passed in the content of the payload.

But some problem can arise and the situation becomes complicated, by example when an IPv6-only client requires access to an IPv4-only server. In the University College of London (UCL) and University of Southampton (UoS) activity under the 6NET project, a standard NAT-PT gateway was used to provide network level transition of Grid Traffic. To succeed in the above scenario, the Grid systems should only use hostnames in the content of the payload rather than any IP addresses. If any IP addresses are passed in the packets' content, later failure would appear in case of IP address utilization. In practice, UCL has experienced difficulties when an IPv4-only Grid client submitted jobs to an IPv6-only Grid server, because the specific NAT-PT implementation failed to provide DNS reverse lookups for the temporary IPv4 addresses. This problem was said not to be serious conceptually, but illustrates the problems that must be resolved in this sort of activity.

Moreover, if direct translation between IPv4 and IPv6 packet formats is theoretically possible, it can be consider as an inconvenient because it introduces all the same disadvantages as an IPv4 NAT.

Each mechanism should be considered independently to be able to find the most suitable ones with the required scenario. Whereas SOCKS is usually only an option if there is already experience with SOCKS in the organization, BIA is especially useful for being able to use old IPv4-only, often binary, applications with IPv6. Both ALG and SOCKS require also special configuration in all the clients, so that the scalability is questionable, so that the best approach depends heavily on the circumstances.

5. PROJECTS AND DEPLOYMENT

Many projects were launched to set up IPv6 architectures and prepare a massive IPv6 deployment in international, national and academic networks. The more interesting things to assess are which infrastructures those projects have left in place after being carried out in their terms and which successes and innovations they brought for the IPv6 community.

The presence of IPv6 in network increases each year especially in parts of the world affected by the limited IPv4 addresses space. Continents such as Asia and Europe, where availability of addresses is limited in comparison with their needs has led to effort to introduce IPv6.

Europe initiated most of the projects to coordinate efforts for developing, testing and deploying IPv6. A major IPv6 project named 6NET is ended now. Another one is European IPv6 Internet Exchanges Backbone (Euro6IX).

For Asia, IP address is a critical resource. This continent possesses only 9 percent of the IPv4 addresses while hosting half of the world's population. For this reason, the region substantially invests in IPv6 deployment. Since 2000, the Japanese government has been considering IPv6 as a priority and set 2005 as a potential deadline to upgrade systems in both public and private sectors. In this perspective, IPv6 Promotion Council has been created in Japan to promote the new protocol.

Initially not interested in IPv6, the United States later invested plentiful effort to make up for lost time. The North American IPv6 task force was created in 2001 to promote the use of IPv6 in industries and universities. Since June 2003, the American position on the IPv6 topic has completely changed. Government engages studies and actions to be involved in IPv6 (for instance Department of Defence Transition Office). Moreover important projects such as Moonv6 were launched (§5.3).

We have chosen to present three major projects. Each is focused on a specific continent to underline differences and similarities on the approach of this new protocol by local IPv6 Task Force (http://www.IPv6tf.org/):

- The main European project for IPv6, 6NET, started in January 2002 and ended in June 2005 with six months extended period to carry out a number of additional technical tasks, demonstrations and dissemination activities;
- The Japanese WIDE project, which is important because the global effort of the Japanese government is widely involved in IPv6 tests and deployments;
- The American project Moonv6, this project is an effort led by the North American IPv6 Task Force (NAv6TF). It was the first great involvement in the IPv6 protocol by the US government.

To conclude, an overview of the IPv6 commercial deployment is given.

5.1. 6NET

5.1.1. About 6NET

6NET is a three-year European project (http://www.6net.org/); it aimed to demonstrate that continued growth of the Internet can be met using new IPv6 technology. 6NET has involved thirty-five partners from the commercial, research and academic sectors. The project ran until June 30, 2005.

This project aimed to build and operate an international IPv6 network to promote use of this new protocol in European area (Figure 11). It was the opportunity for European research and industry to play a leading role in defining the next generation network and applications. To reach its objectives, 6NET activities led a lot of tasks:

- Define, implement and operate an international IPv6 network pilot with both static and mobile components in order to gain a better understanding of IPv6 deployment issues;
- Study, implement and validateIPv4 and IPv6 coexistence, migration techniques and transition tools at backbone, regional and campus levels;
- Introduce and test new IPv6 services and applications, as well as legacy services and applications on IPv6infrastructure;
- Evaluate address allocation, routing and DNS operation for IPv6 networks;
- Identify and implement applications and services that support network mobility;
- Trial IPv6-enabled middleware and user applications;
- Develop and test appropriate management tools;
- Collaborate with other IPv6 activities and standardisation bodies;
- Promote IPv6 technology in Europe with participation to conferences, writing of technical documentations and support documents and inform on this protocol with training events.

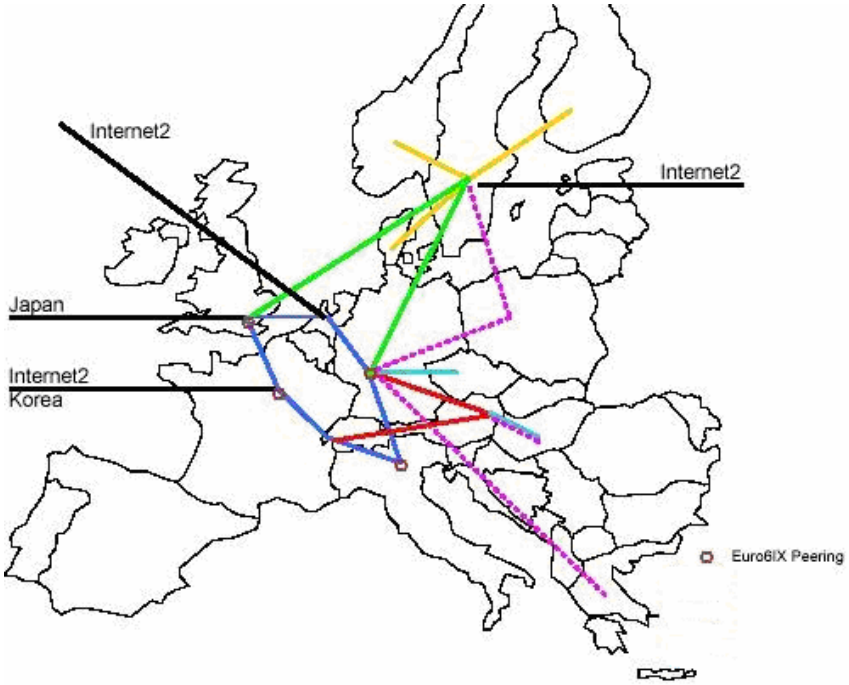

Figure 11. The 6NET Network

5.1.2. Technical overview

The infrastructure (Figure 11) has been deployed and the network became IPv6 QoS enabled and remained operational for about 8 months. Several tests were performed successfully to verify mechanisms in various network conditions and especially under congestion.

In a worldwide perspective, 6NET was connected to Euro6IX and the 6Bone. The 6Bone is an experimental worldwide network for testing interconnectivity of IPv6 implementations and checking if IPv6 really works well in actual situations.

An interesting part of their work is the number of applications developed, deployed and tested in IPv6 context [R1]:

- Conferencing tools such as Videoconference tool (VIC) were tested, for instance the Robust Audio Tool (RAT), the Network Text Editor (NTE) and the Secure Conference Store (SCS);
- They have chosen to test Voice over IP (VoIP), a technology to digitise phone communications with a system based on Session Initiation Protocol (SIP) or H323, such as VOCAL, ISABEL (part of Euro6IX), OpenH323 and GnomeMeeting;
- Streaming applications were also an important point. This is the process of playing a file while it is still downloading. For High Speed Video, VideoLAN deployed by SURFnet, MPEG4IP (video over multicast and unicast IPv6), Digital Video Transport System (DVTS) and Multicast Beacon for IPv6 (MCast6) were chosen. For MP3 Audio Streaming, they tested Trondheim Underground Radio (TUR) and Surge Radio.

The Grid was also a preoccupation in 6Net mainly from the University College of London (UCL) and the University of Southampton (UoS):

- UCL and UoS completed work on Grid Technologies topic with a survey of the Globus Toolkit (GT). This activity was carry out in collaboration with the Globus development team in Argonne National Laboratory. Their studies contributed to make both GT3 and GT4 IPv6-enabled (§6.2.2.1);
- UCL released a set of patches and tools for the IPv6-enabled AccessGrid (http://www-mice.cs.ucl.ac.uk/6net/). The Access Grid is a Globus Alliance project which use Grid resources to support large-scale distributed meetings (high quality videoconferencing).

5.1.3. Results

The 6Net project constituted a large-scale international IPv6 pilot in different domains with some main contributions:

- 6NET has collaborated with other IPv6 activities such as Euro6IX and 6LINK and has contributed to standardisation bodies such as IETF.
- The partners have designed, implemented and tested an IPv6-enabled Network. As a result, a pan-European IPv6 backbone network has been built. Routing (IS-IS and BGP4), tunnelling (IPv6 over IPv4) and DNS support has been used and tested. It has demonstrated that IPv6 is stable and that the new features brought by IPv6, mobility, self-configuration, IPSec and class of services are operational.
- The 6NET project served to launch the IPv6 infrastructure of GÉANT. The two European projects collaborated in a large extent and GÉANT provides operational IPv6 since October 2003.
- The 6NET project offers a Cookbook to inform and give their experience to network designers. It is a guide successful IPv4 to IPv6 migration, available from http://www.6net.org/publications/.
- 6NET has organised training events in Europe (http://www.6net.org/events/) and has been a contributor in the awareness of the IPv6 protocol. Following 6NET, IPv6 DISSemination and exploitation (6DISS) was launched by Europe for two-and-a-half-year to provide IPv6 training and knowledge transfer to research networks in developing regions. 6DISS exploits skills and knowledge gained in 6NET, Euro6IX and GÉANT projects.
- University College of London (UCL), University of Southampton (UoS) and IBM have been involved in studying interactions between Grid technologies and IPv6 [R1].

6NET had taken place in a European Network context with the involvement of GÉANT and European National Research and Education Networks (NRENs). Eleven NRENs involved in the “EGEE network” were also involved in 6NET, see Table 2.

NRENs	Country	EGEE Federation	Connected Since
ACONET	Austria	Central Europe	July 4, 2002
CESNET	Czech Republic	Central Europe	February 14, 2003 (Multicast services since 22 March 2004)
DFN	Germany	Germany and Switzerland	June 27, 2002
FCCN	Portugal	South-West Europe	February 12, 2004
GARR	Italy	Italy	June 21, 2002
GRNET	Greece	South-Est Europe	June 21, 2002
HUNGARnet	Hungary	Central Europe	January 30, 2003
JANET/UKERNA	UK	Ireland and UK	June 26, 2002
RENATER	France	France	August 9, 2002
SURFNET	The Netherlands	Northern Europe	June 19, 2002
SWITCH	Switzerland	Germany and Switzerland	June 12, 2002

Table 2. NREN involved in EGEE and connected to 6NET

5.2. WIDE PROJECT

5.2.1. About WIDE Project

The aim of this project, launched in 1988 and still active, is to establish a Widely Integrated Distributed Environment (WIDE), a new computer environment based on operating systems and communications technology. IPv6 now appears to be a key element in the WIDE infrastructure. This project involves over 670 members, mainly students and researchers from the academic and industrial world.

5.2.2. Technical overview

The WIDE IPv6 backbone includes more than fifty routers and more than sixty IPv6 /48 prefixes (Figure 12). The project works on many aspects of the networks:

- Routing protocols: Two have been tested, Interior Gateway Protocol (IGP) with OSPFv3 and Exterior Gateway Protocol (EGP) with BGP4+;
- Complete works on applications and protocols:
  - DNS compatibility of Berkeley Internet Name Domain (BIND) v.9, a free implementation of DNS;
  - A free HTTP server with Apache v.2;
  - Simple Mail Transfer Protocol: Postfix with IPv6 modifications, Sendmail;
  - Data transfer:File Transfer Protocol (FTP) with BSD's ftpd, wu-ftpd;
  - Multicast (PIM-SM within WIDE).
- Security: firewalls based on operating systems: FreeBSD ipfw, OpenBSD pf; Secure Shell (SSH) with OpenSSH.

Figure 12. WIDE Network

5.2.3. Results

The following sub-projects have undertaken the implementation and validation of IPv6:

- KAME was a joint effort on IPv6 Research and Development by several companies. It provided IPv6 referential implementation for BSD. Three IPv6 RFCs were obtained under KAME project;
- USAGI, with the same objectives as KAME but for Linux;
- TAHI was a project based on a large number of tests to check the conformity of actual network with IETF standards.

WIDE is also a collaborative research foundation and investigates some research topics:

- iCAR (Internet CAR) was the development of mobile communication technologies required to provide a link between automobiles and the Internet;
- Asian Internet Interconnection Initiative Project (AI3) is directed toward satellites. It provides broadband networks to countries lacking cable network infrastructure in Asia where it is impossible to use conventional network deployment;
- School of Internet (SOI) was an experiment where video is the support for teaching;
- Nautilus6 is about practical application of IPv6 in mobile communications. This workgroup performs research and development of mobile communication technologies.

The members of WIDE participate to IPv6Ready, a program to promote the new protocol in the world. WIDE project declared that IPv6 works fine for basic operations (backbone routing, interoperability of various routers, server applications).

5.3. MOONV6

5.3.1. About Moonv6

This is an American IPv6 deployment project led by the North American IPv6 Task Force, Internet 2 (an advanced academic network backbone led by 207 universities) and U.S. Department of Defence

(DREN for Defence Research and Engineering Network). The great majority of partners were from industry.

Two phases were foreseen:

- Phase one (October 2003) was dedicated to building of an IPv6 backbone in the United States, Moonv6 concluded on that the new protocol is stable, resilient and ready for a largest integration;
- The second phase added new partners and tests (February to April 2004).

Objectives of Moonv6 project were to test and to inform on IPv6. Moonv6 tested solutions of IPv6 infrastructure and evaluated interoperability, robustness and stability of the protocol. The project informed also with consultations of the US government, education and firms to deploy IPv6 in North America. The main activities were to establish a broad and solid platform for current and future IPv6 network design and testing, to train the engineers and to pool the resources of vendor equipment for working on implementation of diverse standard protocols.

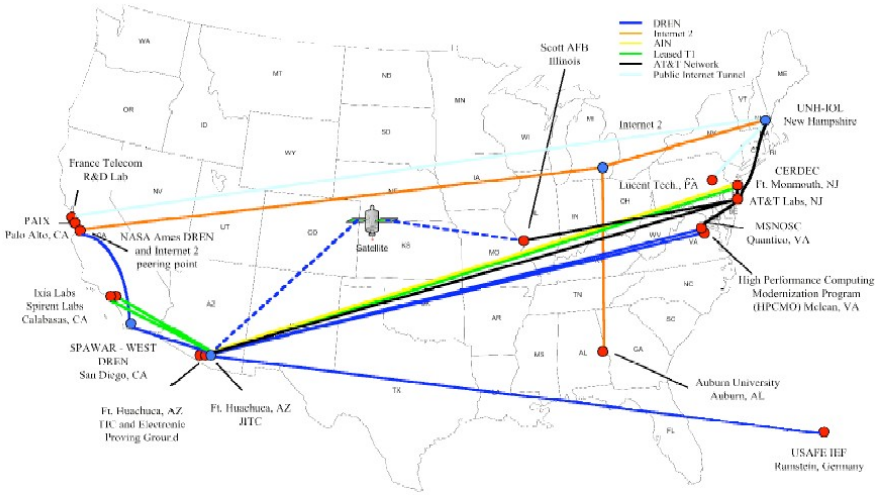

Figure 13. Moonv6 Phase II Network

5.3.2. Technical overview

The backbone of Moonv6 covered more than 3000 miles (across the United States). This project involved 80 servers, switches and routers configured in dual-stack mode (Figure 13).

Like with the 6NET project, many technologies and protocols were tested on IPv6 environment:

- Devices that implement IPv6 were tested for conformance to several base specifications. Interoperability testing was performed by sending data traffic between nodes with ICMP, TCP, and UDP. Core IPv6 functionality items such as address autoconfiguration, duplicate address detection, path MTU and fragmentation, multiple prefixes and network renumbering, and redirect functions were validated;
- Routing interoperability testing was performed on IPv6 capable routers within and between various sites. In addition to the applicable base specification tests, router testing can also include RIPng, OSPFv3 and BGP4+. Another point was coexistence of OSPFv2 and OSPFv3 in IPv4/IPv6 environment. Moreover, transition mechanisms (Dual Stack Routing, Static Tunnel, Additional mechanisms such as tunnel broker) were inserted in a test-bed;

- Mobility tests have validated the ability of home agents, mobile nodes, and corresponding nodes to correctly interoperate in a mobile IPv6 environment (802.11 wireless LANs, Ethernet networks and Applications/Data traffic);
- Primary network services such as DNS, NFS, web servers and general business/personal applications (i.e. SSH, FTP, web browsing, streaming media, video conferencing, and network gaming) were tested for IPv6;
- For security, functionalities of IPv6 firewall technologies like Access Policy, Stateful Firewall Functionality, Network-level testing and deployment, IPSec and Applications between Firewalls were tested.

The end-task was to conduct advanced scenario testing to determine network robustness (via convergence testing); IPv4/IPv6 QoS network level testing; and network security (via hacking).

5.3.3. Results

Before Moonv6, the deployment of IPv6 in United States was less advanced than in Europe and Asia. This project has tested many IPv6 network aspects; for instance, Moonv6 used mobile nodes and IPSec in phase I, Multicast and Quality of Service testing in Phase II.

United States and North American IPv6 Task Force (NAv6TF) were greatly implicated in it. After Moonv6 the US government acted to accelerate the IPv6 integration; two major factors have motivated this attitude, the importance of end-to-end networking model with its impact on the development of IPv6 market, and a true roaming mobility. With the objective of showing that equipments from different vendors can interoperate under realistic operational conditions, Moonv6 takes an important place in an economy strategy, not only for research purpose.

5.4. INTERNET SERVICE PROVIDERS AND IPV6

Nowadays, not only academic and research networks are interested in IPv6. Internet Service Providers (ISP) are concerned and began to offer IPv6 access to their clients.

5.4.1. Consideration of IPv6 protocol in ISP networks

IPv6 transition may be a priority for some ISPs and not for others in the world. The IPv6 deployment can be likened to the chicken and egg situation: i) In Japan and Korea, ISP (for instance NTT) propose directly IPv6 over ADSL to clients and these clients adopt the innovative offer. ii) In Europe, the approach is the opposite, without specific demand from their clients, providers do not invest.

Nobody has today a clear visibility of future services which really need IPv6. Some current applications, for example real-time services (VoIP, videoconference), network games or mobile services must find a great interest in IPv6 with the removal of NAT, but the future applications which can take benefit of IPv6 advantages are not defined yet. Only ISP can lead this evolution to new services allowed by a real end-to-end IP connectivity and by the growing of availability of addresses.

5.4.2. Deployments

Of course, this transition has a cost for providers. This investment is low for equipment, all recent routers run IPv6 natively, but is high for managing both protocols at the same time and for training their technical teams. Despite of these costs, several commercial deployments have been made these last years.

Cooperative Association for Internet Data Analysis (CAIDA), published IPv6 Internet Topology Map in March 2005 on its WEB site (http://www.caida.org/analysis/topology/as_core_network/IPv6.xml). This is a visual representation of IPv6 connectivity around the world, and moreover the webpage

contains a comparative analysis between IPv4 and IPv6 traffics; the company with the richest observed IPv6 peering is NTT/Verio headquartered in Japan, with 141 peers.

Since 2002 the leading Internet Services Providers in Japan (for instance NTT, Japan Telecom, Softbank BB and IIJ) have launched commercial IPv6 services. Different access technologies are available, dedicated line, DSL, Ethernet access, wireless LAN, optical fibre and some connectivity type such as native IPv6,, dual-stack, tunnelling are proposed. An example of new service on IPv6 is free-video on demand.

MCI is providing a direct connexion between its commercial Internet backbone and Moonv6 in United States this year. NTT Verio is the first ISP in the U.S. to offer commercial-grade, nationwide service that supports IPv6. The first deployments of IPv6 in ISP backbone in Europe were in 2002 with investments in example of Global Crossing, Shanova (Telia), NERIM, NTT Europe, Opentransit (France Telecom), Tiscali.

Major points of data exchange on Internet are Global Internet eXchange (GIX) nodes. GIX are used by ISPs to exchange large amount of data across the world. GIX are often managed by NRENs (RENATER for SFINX, SURFnet for AMS-IX), so IPv6 arrived very early in these nodes (in 2000).

The two tables give a list of IPv6 GIX in the world which shows the interest of the ISPs for IPv6.

Name	Location	Web site
6SFINX	Paris, France	http://www.sfinx.tm.fr
FICIX	Helsinki, Finland	http://www.ficix.fi
AMS-IX	Amsterdam, The Netherlands	http://www.ams-ix.net
FNIX6	Paris, France	http://www.fnix6.net
INXS	Munich, Germany	http://www.inxs.de
MCI MAE-Frankfurt	Frankfurt, Germany	http://www.mae.net
NaMeX	Rome, Italy	http://www.namex.it
TREX	Tampere, Finland	http://www.trex.fi
UK6X	London, UK	http://www.uk6x.com
XchangePoint	London, UK	http://www.xchangepoint.net
CIXP	Geneva, Switzerland	http://www.cixp.ch

Table 3. European GIX

Name	Location	Web site
6NGIX	Seoul, South Korea	http://www.ngix.ne.ke
6TAP	Chicago, USA	http://www.6tap.net
JPIX	Tokyo, Japan	http://www.jpix.co.jp
MCI MAE-EAST	Washington DC and New York, USA	http://www.mae.net
MCI MAE-WEST	San Jose, USA	http://www.mae.net
MCI MAE-Central	Dallas and Chicago, USA	http://www.mae.net
NSPIXP6	Tokyo, Japan	http://www.wide.ad.jp/nspixp6
NY6IX	New York, USA	http://www.ny6ix.net
PAIX	Palo Alto, CA	http://www.paix.net

Table 4. Other major GIX in world

6. GRIDS OVER IPV6

The data and control traffic on networks between the components of Grid systems is realized with TCP or UDP protocols over IP, currently IPv4. As it has been seen earlier in the document, IPv6 may become soon a reality and perhaps it is timely to envisage an IPv4/IPv6 cohabitation in Grid infrastructures. The emergence of new countries in Grid Computing (India, China, and Latin America) and high speed networks can accelerate the need for such cohabitation.

This chapter is dedicated to an overview of the major experiences of IPv6 implementation realized within Grid systems, including an analysis of IP dependencies of any Grid components or lower layer network protocols. Authors of this document have conducted this study through a networking approach and can not pretend having suitable developer skills to enter in implementation considerations.

The following are discussed in this section:

1. Grid architecture;
2. Effort to enable IPv6 for Grids;
3. IPv6 impact on applications;
4. gLite and IPv6;
5. Infrastructure readiness for IPv6;
6. IPv6 in the EGEE network infrastructure.

6.1. GRID ARCHITECTURE

Most Grid infrastructure (Figure 14) mainly consist of three basic building blocks: at the lower level the underlying infrastructure providing computing, storage and network resources; at the upper level the users with their applications, wanting to use the resources; and bringing the two together, the so-called "middleware" (collectives services, basic services and system software). The middleware typically is a set of different modules, which acts collectively as an intermediary, hiding the multiple parts and detailed workings of the Grid infrastructure from the user. The Grid thus appears as a single, coherent resource, in which the middleware ensures that the resources are used as efficiently as possible and in a secure and accountable manner.

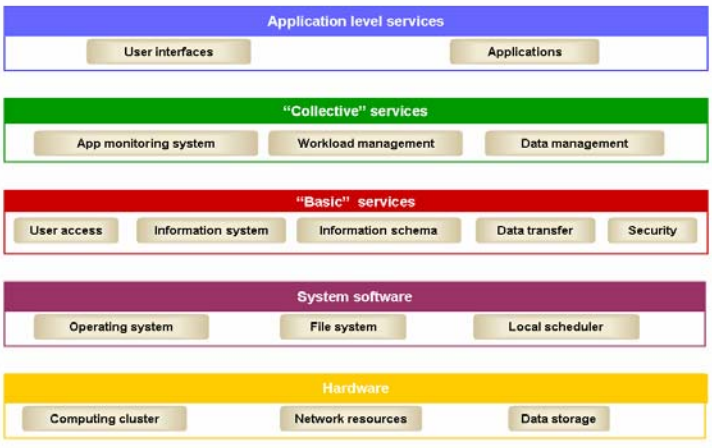

Figure 14. Grid Architecture (source LCG-2)

Most of the modules dedicated to Grid computing deal with network. So, finding the IP-version dependencies of these modules is essential as to provide a complete compatibility with IPv6. Obviously updating the software and testing the new architecture are the two main tasks in order to provide IPv4/IPv6 cohabitation.

6.2. EFFORTS TO ENABLE IPV6 FOR GRIDS

It is important at the dawn of massive arrival of IPv6 protocol in network infrastructures to avoid IPv4 dependencies in applications and core software even if a long period of coexistence is announced.

6.2.1. IPv6 standardisation in GGF (IPv6-WG)

As the scaling up of the internet and thus of Grids will require the additional address space and management features of IPv6, the Global Grid Forum (GGF) has established an IPv6 Working Group (IPv6WG). The purpose of the group is to analyze any GGF specifications in order to provide appropriate guidelines for future specifications, and to communicate any issues discovered with IPv6 to the IETF, the Java community, etc. It has initially been tasked with producing two documents.

The first document [R40], published in December 2004 is a set of guidelines for IP version-independence in future specifications to ensure that current and future Grid technologies and applications can easily become IP-version independent. It describes how to avoid IPv4 dependency in GGF specifications. There is a detailed discussion on how addresses should be parsed and used, on the subject of name resolution functions and mapped IPv4 addresses. IPv6 features have particular impact when one tries to write implementations which are IP-neutral. This document is intended to be used by all GGF Working Groups as a checklist for document approval.

For specifications, there are several suggestions, e.g.:

- If addresses must be included, add an address type code;
- For literal IPv6 addresses use RFC2732 [R16];
- Use Fully Qualified Domain Names (FQDNs) like *lnx1180.cern.ch*, instead of IP address.

For implementations observe the following:

- Code should be modular;
- Code should be IP-independent, including the APIs;
- Care should be taken which of IPv4 or IPv6 is preferred, if both are available;
- Graphical user interfaces must take into account the different lengths and display formats;
- It may be impossible to make implementations IP-independent if some of the unique features of IPv6 are used.

The second document [R41] published in January 2005 is a review of IPv4 dependencies in existing GGF specifications. Any specification that involves the handling of network I/O or IP addresses, or the processing or display of URLs is likely to be affected. The report surveyed some 88 documents for IPv4 dependencies. Possible IPv4 dependencies were surveyed on GGF specifications that are dated earlier than 11th December 2003. Out of the 88 documents analysed, 24, about 30%, had some form of IPv4 dependency.

6.2.2. Enabling IPv6 for Globus Toolkit

The Globus Toolkit (GT) was developed mainly in the Argonne National Laboratory (ANL). The last release of this middleware was produced this year (version 4). Studying the activity to make GT IPv6-compliant is interesting because gLite is Globus-based Grid, even though Globus Toolkit and gLite have taken separate ways after GT2. Two projects have contributed to the support of IPv6 in Globus

Toolkit. The first one is an activity of 6NET project that focused on GT3. The second experiment is Japanese project 6Grid which has introduced IPv6 support in GT2.

6.2.2.1. 6NET Grid activity

The University College of London (UCL) and the University of Southampton (UoS), as mentioned in chapter 5.1, were deeply involved in the evolution of the Globus Toolkit in IPv6-enabled Grids. This workgroup produced documents on this subject between June 2004 and January 2005 and led a successful Test-bed. They wrote deliverable [R1] for 6NET project, and began to work on the Globus Toolkit in the middle of 2002. They gave their conclusions to ANL to produce an alpha IPv6 Globus Toolkit 3 Grid during year 2003.

UCL and UoS analyzed all GT3 components and code to find IP dependencies. They demonstrated full IPv6 functionality within GT3, except for GridFTP. Moreover, they produce a GT3 Test-bed in UCL in May 2003 in both IPv4/IPv6 environments and another one in IPv6-only environment.

This work was included by Globus in the official release and GT3.2 became the first GT version with IPv6 support. Documentation on this topic was produced, the most interesting being: “How to IPv6 in Globus Toolkit 3” [R2]. This document presents issues and solutions concerning GT3. Subjects treated were operating systems, networking support of IPv6, Associated applications (Java SDK, PostgreSQL and Web Containers), configuration of hosts and tests.

Concerning GT4, UCL has continued their studies on GT4 and has produced the document “How-to IPv6 in Globus Toolkit 4” [R46] in January 2005. GT4 is based on the new Web Services Resource Framework (WS-RF) standards [R47]. With the experience on GT3, UCL has successfully demonstrated IPv6 functionalities for WS-RF core, Globus _XIO, GridFTP and WSGRAM on the GT4 alpha version (GT3.9.4). This work made in collaboration with the Globus implementation group in Argonne currently continues.

6.2.2.2. 6Grid Project

Started in 2002 as a sub-project of the BioGrid project, Japanese scientists have participated to 6Grid Project for the development of an IPv6-enabled Grid for Biosciences. The Japanese 6Grid Project wanted to use many benefits of IPv6, firstly to solve the lack of IPv4 addresses, well-known in Japan, and to propose a NAT-free global computing in an end-to-end addressing schema. Secondly they were interested in the IPSec-based data protection. They concluded on three advantages to promote Grid over IPv6:

- Large number of addresses (no NAT). This advantage is very significant in Asia;
- IPSec in IPv6. 6Grid members consider the protocol as a robust secure authentication at IP-Packet level. To approach the security against performance trade-off, they choose to use IPv6 functionality for the realization of a secure Grid environment;
- Dynamic Assignment of IP-addresses. This advantage reduces administration costs.

In the scope of project, three components were developed, an IPv6 Globus Grid Toolkit version 2.2.3 tested with RedHat Linux 7.3 operating system, an IPv6 GridFTP and an IPv6 patch for GSI-SFS (for security component in Globus).

The software is downloadable from the web site http://www.bioGrid.jp/e/research_work/gro1/6Grid/.

6.3. APPLICATIONS AND USER INTERFACES

The next generation of networking applications should be able to communicate over both the IPv4 and IPv6 protocols. In fact, having two different applications (or versions of the same application) to handle networking services, one for IPv4 and the other for IPv6, may cause problems. On the server side there could be inconsistencies that may be very difficult to address, such as dual stack client

applications that connect once to the IPv6-only application and the next time to the IPv4-only application in an unpredictable fashion. On the client side it could be annoying for the users, who must be aware that one application is IPv6-only and the other is IPv4-only, and it may force the system administrators to deploy wrapper applications. In addition, the applications must be designed to work even if the target hosts have IPv4 or IPv6 connectivity (or even support) disabled.

To start any IPv6 deployment, we need the IPv6 support from the platform. First we need an IPv6-enabled networking and also support from IPv6-enabled network services, such as IPv6 DNS, Web services, etc.

The specifications of a few protocols needed to be modified according to previous work and experience gained by several groups. Due to IPv6 changes, certain changes in socket support, and network related libraries should be considered.

6.3.1. Network programming

Applications interact with the network through network interfaces: Sockets, Remote Procedure Call (RPC), Streams. These programming interfaces allow the transport of information over TCP and UDP on IP networks. The sockets programming interface is the more commonly used.

An application which does not use directly a network programming interface is not IP dependant by itself. It is the case when an application calls GridFTP by example; whatever the IP protocol, it will realize its file transfers.

6.3.1.1. Hard-Coded IP addresses and hostnames

A host on a network can be identified by its unique IP address (a hard-coded address) or by its hostname. For instance a Storage Element in CERN has the name *lnx1180.cern.ch* equivalent to hard-coded IPv4 address *137.138.152.209*. Sometimes programmers use the IP address to avoid too many queries to the DNS; because a DNS that receives a huge amount of DNS requests can collapse. Frequent compatibility issues are generated by the use of hard-coded IP addresses, such as loopback addresses (127.0.0.1) or conventional IPv4 addresses. A solution to solve this issue is to convert all hard-coded addresses with the name of the network host. A component that does not use *lnx1180.cern.ch* but the IPv4 address is IPv4-dependant. For the same reason, it is necessary to switch loopback addresses by hostname “localhost” comprehensible by both protocols.

6.3.1.2. Sockets

The concept of sockets was introduced into Unix distributions of Berkeley (Unix BSD) and the term “BSD sockets” (Berkeley Software Distribution) is often used.

The programming interface offers Inter Process Communication (IPC) in order to make it possible for various processes to communicate on the same machine or across an IP network. A socket is completely defined by the source IP address, the destination IP address and the respective port numbers.

Applications use primitives to establish connection and exchange information through sockets. For instance, the *gethostbyname* function retrieves host information corresponding to a hostname, and *gethostbyaddr* function retrieves host information corresponding to an IPv4 address.

RFC 2553 [42] defines the extensions of the sockets programming interface by extending the current interface in two ways:

- A new family of functions, for example *getaddrinfo()* replaces *gethostbyname();*
- IPv6 is represented by the new addresses family AF_INET6.

Through the use of these new interfaces, an application can be IP independent and the code will have to be adapted. However programmers do not commonly employ these new facilities, so the applications become IP-dependant.

6.3.2. Languages

TCP/IP and UDP applications written using the sockets API enjoy a high degree of portability with IPv4 and the same portability is expected to persist with IPv6 applications. But changes are required to the sockets API. The following languages are the most common used in the grid middleware.

6.3.2.1. C/C++ languages

Some changes were needed to adapt the socket API for IPv6 support: a new socket address structure to carry IPv6 addresses, new address conversion functions and several new socket options that are developed in RFC-3493 [R21].

These extensions are designed to provide access to the basic IPv6 features required by TCP and UDP applications, while introducing a minimum of changes into the system and providing complete compatibility for existing IPv4 applications. Access to more advanced features (raw sockets, header configuration, etc.) is addressed in RFC-3542 [R23].

To allow this adaptation, code should be ported to make suitable changes in the client and server components.

So if an application programmer wants a C/C++ program to run in IPv6 mode, he would first need to port it. With Java, it is different.

6.3.2.2. Java

With the release of JDK 1.4 in February 2002, Java began supporting IPv6 on Solaris and Linux. Support for IPv6 on Windows was added with JDK 1.5. While other languages, such as C and C++ can support IPv6, there is some specificity to Java:

- There should be no change in Java application code if everything has been done appropriately. i.e., there are no direct references to IPv4 literal addresses; instead, hostnames are used;
- All the address or socket type information is encapsulated in the Java networking API.
- Through setting system properties, address type and/or socket type preferences can be set.
- For new applications IPv6-specific new classes and APIs can be used.

Thus, the same binary code can run in IPv6mode if both the local host machine and the destination machine are IPv6-enabled. Using IPv6 in Java appears easy as it is transparent and automatic. The advantage is that no porting is necessary and there is no need to even recompile the source files.

When deploying Web Services, JAVA APIs and specific components to the Web Services are most commonly used. A Web Service is an entity that exchanges documents with the outside world. It is self described and has a unique identity. In gLite middleware the Simple Object Access Protocol (SOAP) defines the internal structure of the XML documents consumed and produced by Web Services. It should be noticed that a Web Service carries its own description:

- What kinds of documents it exchanges (the */interface definition/*);
- Where the service lives (the URI-address);
- Which transport protocols it can use for the exchange of documents (the */binding/*).

As Java supports IPv6, there is no reason to consider that the deployment of Web Services in an IPv6 environment is impossible.

6.3.2.3. Perl

Perl provides support for the socket API natively. A module exists and provides functions to deal with IPv4/IPv6 addresses. The module can be used as a class, allowing the user to instantiate IP objects, which can be single IP addresses, prefixes, or ranges of addresses. There is also a way in which most subroutines can take either IPv4 or IPv6 addresses transparently.

6.3.2.4. Python

Python is an interpreted, interactive, object-oriented programming language. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. The standard python interpreter (AKA CPython) natively supports IPv6 and since Python 2.3, the socket module handles IPv6, so that IPv6 supports with Python programs is possible.

6.4. GLITE SERVICES

gLite (Lightweight middleware for Grid Computing) middleware is a Service Oriented Grid middleware suite providing services for managing distributed computing and storage resources and the required security, auditing and information services.

gLite middleware was initially based on the EDG Middleware (LCG), based itself on Globus Toolkit 2 (Figure 15). Studies of 6NET have demonstrated that GT2 was not IPv6-enabled. This is consequently also the case for gLite.

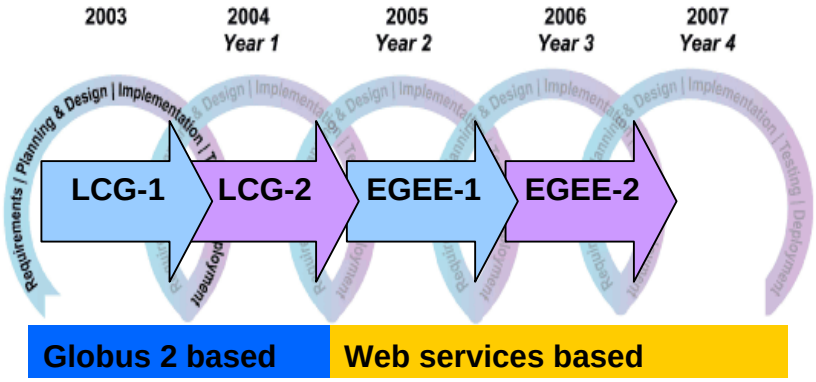

Figure 15. Grid evolution to gLite Middleware

Released in August 2005 the current gLite version is 1.3 (http://glite.web.cern.ch/glite/). The architecture of gLite is presented in EGEE Middleware Architecture [R39]. The modules of gLite are listed below (Table 5).

Module name	Version	Grid Architecture
gLite Computing Element	2.0.2	Computing Element
gLite Workload Management System	2.0.2	Workload Management
gLite Logging and Bookkeeping Server	2.0.0	Logging and Bookkeeping
gLite Worker Node	2.1.0	Workload Management

gLite R-GMA Server	5.0.0	Information and Monitoring
gLite R-GMA Client	5.0.0	Information and Monitoring
gLite R-GMA Service Publisher	5.0.0	Information and Monitoring
gLite R-GMA GIN	5.0.0	Information and Monitoring
gLite I/O Server	2.0.0	Data Transfer
gLite I/O Client	2.0.0	Data Transfer
gLite Single Catalog for MySQL	2.0.0	Data Management
gLite Single Catalog for Oracle	2.0.0	Data Management
gLite File Transfer Service for Oracle	2.1.0	Data Transfer
gLite Data Transfer Agents for Oracle	1.1.0	Data Transfer
gLite File Placement Service for Oracle	2.1.1	Data Management
gLite Stand-Alone Metadata Catalog for MySQL	2.0.0	Data Management
gLite VOMS Server and Administration tools	2.0.0	Security
gLite User Interface	1.1.0	User Interface
gLite Service Discovery APIs	1.0.0	Information and Monitoring
gLite Security Utilities	1.0.3	Security
Torque + Maui Server for the gLite Computing Element Node	2.1.0	Computing Element
Torque Client for the gLite Worker Nodes	2.0.0	Computing Element

Table 5. gLite Modules

In order to find possible IP-dependencies, this chapter attempts to analyze the gLite services that incur network operations such as connection to an IP address and a network port. With EGEE Middleware Architecture [R39] document and other middleware documentations, interactions between compounds were analysed to find obvious IP-dependencies and an external analysis of the middleware has been conducted without survey of the code. Therefore, this study can not include some gLite-related components like gSOAP or Alien-IO.

6.4.1. Workload Management System (WMS)

The WMS client APIs supply the client applications with a set of interfaces over the job description, submission, monitoring and control services. The API-provided methods could be conceptually grouped into two main categories that are Job Description and Job Submission, Monitoring & Control.

Figure 16 shows the overall architecture of the Workload Manager, together with the interactions with external entities mainly through Web Services interfaces (§6.3.2).

The support of IPv6 in WM requires a code analysis of all modules to find the IP-dependencies. This work is out of the scope of this document. An example of suspected dependencies in the Workload Management System: the gLite's package org.glite.wms-util.tls contains a socket implementation for the client/server model; this module may cause problems for IPv6 compatibility if the format of sockets is not IPv6-compliant.

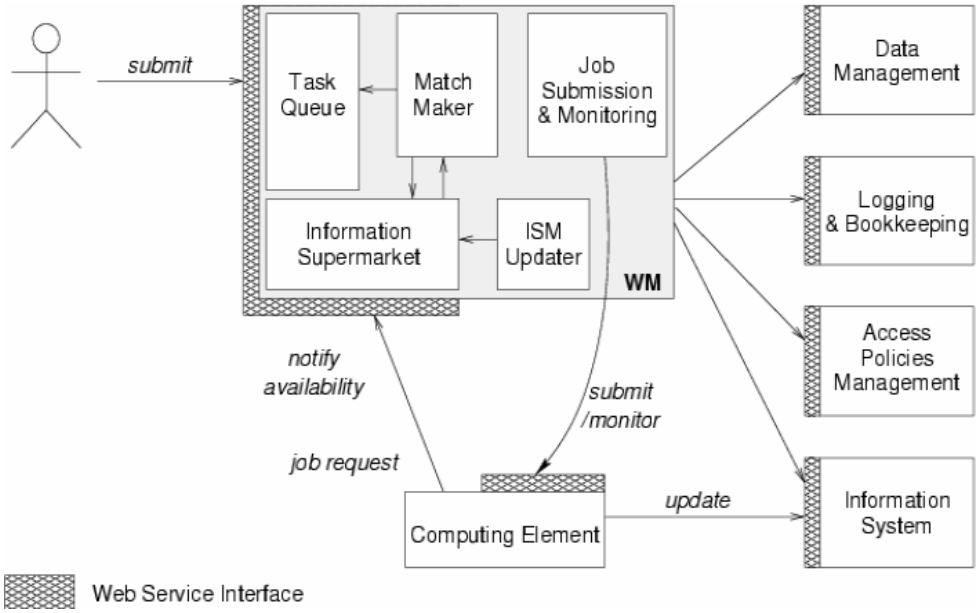

Figure 16. Internal architecture of the Workload Manager.

6.4.2. Data Management

The gLite Data Management component is a set of services, including the following:

- Storage Element (SE): An application should be able to access its data independently of the location of the current running CPU. The gLite middleware provides a lightweight Storage Resource Manager (SRM) as a storage solution. The end-user application only needs to use the gLite-I/O API in order to access its data but the SRM and the security services are used indirectly by the Data Management component. A Storage Element is a host accessible by Grid applications to store data. Each Storage Element is defined by a hostname associated to its physical location. Applications which interact with Grid services on a Storage Element use its hostname and are associated with specific network ports. In principle, this type of interaction does not involve an IP dependency;
- Data Catalogues: Management of catalogues for data. Files may be replicated at many Grid sites if they are often used. To identify data, each file receives a location-independent logical file name (LFN). Upon creation, each file also acquires a Grid Unique ID (GUID). The applications may use either the LFN or GUID to identify their files, the GUID is always mandated by the system, whereas the LFN is assigned by the users;
- Transfer Scheduling: A set of services is defined to schedule and control data movement between Grid sites; there are the Data Scheduler services.

Grid data services identify resources by URIs to manage storage resources; generally, an URI contains a hostname but it is possible to have a hard-coded IP address. In most case there is no IP-dependency.

6.4.3. Computing Element

The Computing Element (CE) is the service abstracting a computing resource. The CE allows access to Web service interface usable by clients. An end-user interacts directly with the Computing Element or the Workload Manager (which interacts itself with the Computing Element). A Computing Element refers to a computing cluster (§6.5.1). It provides the job management functionalities (interaction with

job – cancellation, submission, suspend and other, and assessment of QoS for the job submitted). It can also provide information on itself such as their characteristics and status.

To submit jobs, Computing Element may know the hostname of the used computing resources. The GlueCE database (used by R-GMA) contains this information. Only two pieces of network information are kept in this database: Hostname and GatekeeperPort. GatekeeperPort field is unaffected by IPv6 because the new protocol does not change the network port mechanism, and the Hostname field is understood for both IP protocols. This identification of CE by URI should not have an IP dependency as long as hard-coded addresses were not use in these URI.

6.4.4. Data Access

Access to remote data implies network access, many mechanisms are used by gLite but only the two main systems, GridFTP for file transfer and Network File System (NFS) for data sharing, are considered in this document.

6.4.4.1. GridFTP

GridFTP is the protocol proposed for all data transfers. It is the lowest level of the Globus data management services. It extends the standard FTP protocol that has been updated with "FTP extensions for IPv6 and NATs" [R11]. It also adopts the Globus Grid Security Infrastructure (GSI).

The Globus Toolkit is a widely used set of tools and libraries for grid computing, including certificate-based authentication and data management services.

Files in GridFTP are referenced using a URL of the forms ftp://host/path/file or gsiftp://host/path/file. The former URLs use the traditional FTP protocol, while the latter URLs may also use GridFTP extensions such as GSI authentication, and parallel data transfers.

The specific implementations of these protocols need modifications and improvements to support IPv6. Within the Globus project, GridFTP is currently implemented in standard C. A new IP-independent network module known as globus_XIO (eXtensible Input Output library) is being developed by the Argonne National Laboratory (ANL) for use by GridFTP.

GridFTP in GT4 includes a new server implementation and has IPV6 support included as a new feature. It is totally backwards protocol compatible (with GridFTP 2.4 and higher). The XIO drivers distributed with Globus 4.0 include TCP, UDP, file, HTTP, GSI, GSSAPI_FTP (Generic Security Services Application Programming Interface), telnet and queuing. UCL has tested it working in a proper dual-stack environment.

6.4.4.2. Network File System (NFS)

The Network File System is a distributed file system that enables users to access files and directories located on remote computers and treat those files and directories as if they were local. NFS is independent of machine types, operating systems, and network architectures through the use of Remote Procedure Calls (RPC).

One of the main constraints to support IPv6 in NFS is the support of IPv6 addresses in the RPC call and *getport()* functions to allow the NFS client to communicate with an IPv6 remote server and to get the port on which the remote RPC program is listening.

This can be done by implementing the *getaddr()* (v3) or *getversaddr()* (v4) functions of the *rpcbind* specification.

Different methods of implementations are possible:

- Each time an address (or a socket address) is used, a specific processing is performed according to the family of the address;

- Another way is to use the IPv4-mapping addressing: with this method, recommended to migrate to IPv6, the IPv4 addresses are mapped into IPv6 addresses and then all the processing is done on IPv6 addresses. So it is no more necessary to distinguish IPv4/IPv6 addresses and the IP layer automatically map/unmap the IPv4 addresses when needed;
- A third way is to use sets of functions specific to each protocol written to implement the parts of code dependent on the address type. When a new client or server instance is created, the set of functions corresponding to the transport protocol is stored in the associated structure to be used when needed.

Patches and packages allowing the support of NFS (v2, v3 and v4) over IPv6 already exist.

6.4.5. Information & Monitoring

R-GMA (Relational Grid Monitoring Architecture) is based on the GMA from the GGF (Grid Global Forum), which is a simple Consumer-Producer model. R-GMA is implemented as a set of several types of services running on one or more servers.

R-GMA is currently based on Servlet technology. Each component has the bulk of its implementation in a Servlet. Each service has a well defined set of operations. Each operation of each service is represented by a method (or function) in the API. Multiple APIs in Java, C++, C, Python and Perl are available to communicate with the servlets. Tomcat Servlet is the container used for R-GMA. Tomcat5 would be recommended with its fully IPv6 support. Most of the code is written in Java and is therefore highly portable and APIs are available in various languages (Java, C++, C and Python) for interaction with R-GMA to make it easier for various Grid Services and Applications to interact with the R-GMA services. See 6.3.2 for their IPv6 support.

6.4.6. Logging and bookkeeping

The Logging and Bookkeeping service (L&B) tracks jobs in terms of event (important points of job life, e.g. submission, finding a matching CE, starting execution etc.) gathered from various WMS components as well as CEs (all those have to be instrumented with L&B calls).

The L&B service architecture (Figure 17) features two APIs, a local logger subservice and the servers.

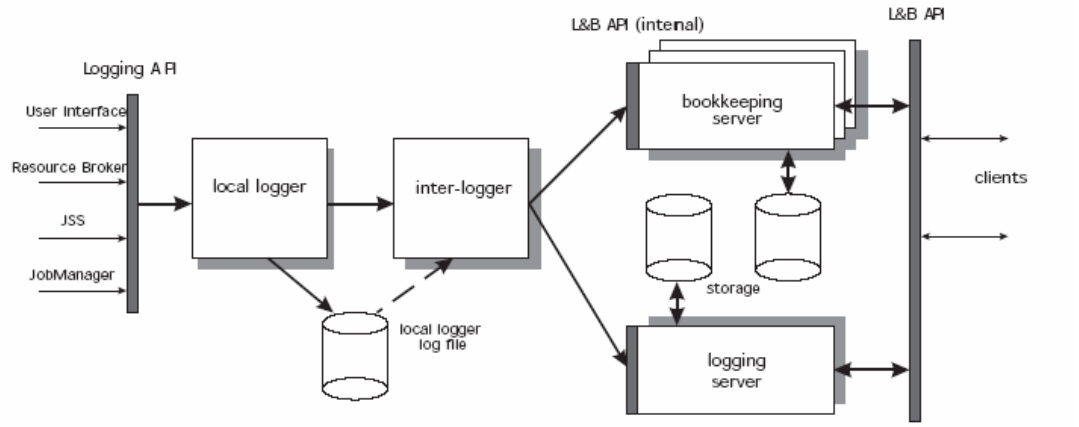

Figure 17. L&B architecture and APIs

As for the Workload Manager, a detailed code survey is necessary to find the IP dependencies.

6.4.7. Accounting

The component gathers information about the usage of Grid resources by the users and by groups of users. This information allows resource usage for individual users to be tracked. DataGrid Accounting System (DGAS) collects data from users, resources and jobs. This component uses URLs to access a specific database and no obvious IPv4-dependency appears.

6.4.8. Analysis summary

The gLite middleware IP-dependencies studied in this chapter are summarized in Table 6.

gLite Services	Observations	Comments
Workload Management System	Suspected dependencies.	Detailed code survey is necessary to find the IP dependencies.
Data Management	Use of URI, there are no obvious IPv4-dependencies.	
Computing Element	Use of URI, there are no obvious IPv4-dependencies.	
Data Access	GridFTP is not IPv6-enabled.	ANL developed a globus_eXtensible Input Output library (XIO), which may be used to implement an IPv6-enabled GridFTP.
	NFS is IPv6-enabled.	
Information & Monitoring	Some libraries are not IPv6-enabled.	
Logging & Bookkeping	Suspected dependencies.	Detailed code survey is necessary to find the IP dependencies.
Accounting	Use of URI, there are no obvious IPv4-dependencies.	

Table 6. gLite middleware analysis summary

6.5. GRID EQUIPMENTS

Two kinds of equipment are concerned:

1. Computing and storage equipments;
2. Network equipments: routers, firewalls.

These equipments are directly connected to the network; so that they must be IPv6 compliant, whether IPv6 only or dual stack (§ 4.1.1).

6.5.1. Clusters

A cluster is a set of resources coupled together, usually so as to tackle one task faster. It can be seen as an association of many processors in a unique entity. A cluster brings together a great variety of architecture; a cluster may be a purpose-built supercomputer or a set of PC servers. The cluster architecture is presented in Figure 18. Usually a resource is defined as master node (a load-balancing unit), and the others as nodes (processing units). The master unit receives request and redirects it to a specific node. Load-balancing can be achieved by software on the master node or by hardware, a router with load-balancing capabilities. Many algorithms have been defined to achieve load-balancing and the most easy to understand is the load-based algorithm which gives the new task to the least busy node.

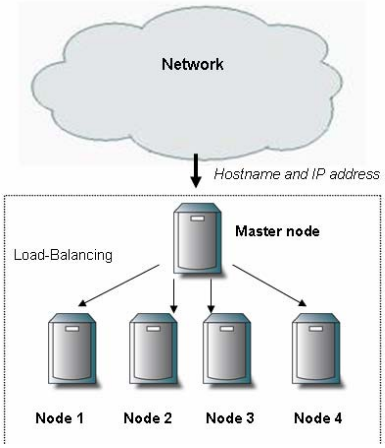

Figure 18. Cluster architecture

A computing cluster is composed by more than one host but has a unique hostname assigned.

Scientific Linux 3 is the recommended Operating System (OS) by the EGEE project. This OS is IPv6-enabled.

On Linux Red Hat 7.1 and later, IPv6 is supported by the kernel, though the IPv6 module is not loaded by default. For Red Hat 8.0 and later, IPv6 is provided and auto-loaded by default.

The best way to make a cluster IPv6-compliant is certainly the dual mechanism. Each node becomes an IPv4/IPv6 host working equally in both worlds.

This IPv4/IPv6 cohabitation implies a competence in the two protocols from system and networks administrators; it will certainly increase the work in their troubleshooting activity. It is important to keep in mind that there is no reason for network administrators to face twice the amount of works.

6.5.2. Network equipment

To enter in the IPv6 world in the same way as in the IPv4 world, routers and firewall must support the new protocol and supply the same network services.

6.5.2.1. Evolution of Network equipment

Recent routers and switch-routers are dual stack with an implementation conforming to the IETF standards. Nevertheless older network equipment which does not support IPv6 must be changed or associated with an IPv6 router if possible. IPv6 network equipments must support the IP features described in chapter 3 and some transition/translation mechanisms. The prerequisites to manage the equipments are supported by the IPv6 routers: SSH, Telnet, TFTP.

Regarding the network security, the filtering functions allow to write access lists (ACL) in the same way as IPv4.

6.5.2.2. Network administration

The deployment of IPv6 networks is relatively easy with mechanisms like automatic configuration (§3.2.1) but a network requires fault, security, topology, configuration and accounting management.

To manage a network, an interesting solution is the Simple Network Management Protocol (SNMP). The SNMP manager collects information from network equipments in a Manager Information Base

(MIB). Network devices can also transmit alerts to SNMP manager. This protocol is independent from IP; its evolution to IPv6 was not a problem.

With SNMP, IPv6 equipments are manage with requests in IPv4 flows, so reports are not available in an IPv6-only network.

The first IPv6-enabled version arrived in May 2002. Furthermore, evolution of MIB has been more complex and some RFCs were affected by these MIB modifications. Internet Engineering Task Force (IETF) has decided to define a MIB-2 (RFC3291, May 2002) to supervise both IPv4 and IPv6 networks but MIB-2 does not yet appear on equipments. Despite this, network administrators generally use IPv4 to supervise all theirs equipments today.

Various monitoring tools are now available for IPv6 networks. Projects like 6Net have widely contributed to test, implement and document them [R44].

6.6. IPV6 NETWORK INFRASTRUCTURE FOR EGEE

The EGEE network is supported by the National Research and Education Networks (NRENs) and GÉANT which provide connectivity between the Resource Centres. In its end-to-end aspect (Figure 19) this connectivity involves also regional, metropolitan and campus networks, collectively known as “last mile”.

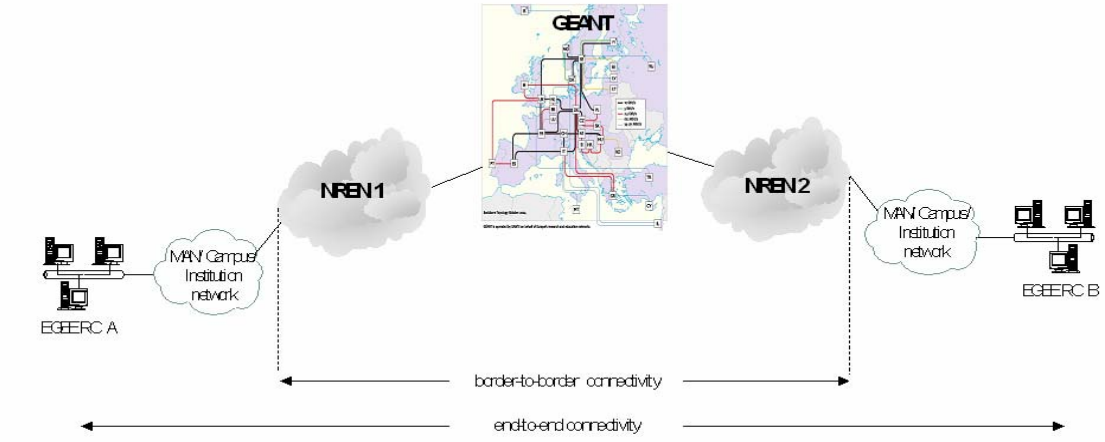

Figure 19. The hierarchical model of network connectivity between EGEE RCs

6.6.1. GÉANT and NRENs IPv6 connectivity

GÉANT is the pan-European network backbone connecting 32 NRENs as partners. An important objective during the third year of the GÉANT project was the test and deployment of IPv6 services. GÉANT offers a dual-stack IPv6 backbone. A GÉANT IPv6 Task Force (http://www.join.uni-muenster.de/geantv6/) was created and an IPv6 service is operational since October 2003 with the same level of support as the IPv4 service.

Most of the NRENs connected to GÉANT (28) now offer IPv6 as a native protocol in their network core. Projects like 6NET have often pushed national networks to deploy IPv6. Their connectivity with GÉANT is either native or tunnel (Table 7).

Name	Country	Website	Connection type	Connection date
ACOnet	Austria	http://www.aco.net/	Tunnel	May 03
ARNES	Slovenia	http://www.arnes.si/	Native	July 03
BELNET	Belgium	http://www.belnet.be/	Native	July 03
CARNet	Croatia	http://www.carnet.hr/	Native	March 03
CESNET	Czech Republic	http://www.ces.net/	Native	July 03
CYNET	Cyprus	http://www.cynet.ac.cy/english/	Native	December 04
DFN	Germany	http://www.dfn.de/	Tunnel	September 03
EENet	Estonia	http://www.eenet.ee/englishEENet/	Native	May 03
FCCN	Portugal	http://www.fccn.pt/	Native	April 03
GARR	Italy	http://www.garr.it/	Native	April 03
GRNET	Greece	http://www.grnet.gr/	Native	July 03
HEAnet	Ireland	http://www.heanet.ie/	Native	April 03
HUNGARNET	Hungary	http://www.hungarnet.hu/	Native	June 03
ISTF	Bulgaria	http://www.ist.bg/	Native	May 05
IUCC	Israel	http://www.iucc.ac.il/	Native	April 03
LITNET	Lituania	http://www.litnet.lt/	Tunnel	May 03
NORDUnet	Nordic countries	http://www.nordu.net/	Native	August 03
PIONER	Poland	http://www.man.poznan.pl/	Native	May 03
Rbnet/RUNnet	Russia	http://www.ripn.net/rbnet/en/	Native	September 03
RedIRIS	Spain	http://www.rediris.es/index.en.html	Native	April 03
RENATER	France	http://www.renater.fr/	Native	April 03
RESTENA	Luxembourg	http://www.restena.lu/	Native	June 003
RoEduNet	Romania	http://www.roedu.net/	Native	May 03
SANET	Solvakia	http://www.sanet.sk/en/index.shtm	Native	October 04
SURFnet	Netherland	http://www.surfnet.nl/	Native	April 03
SWITCH	Switzerland	http://www.switch.ch/	Native	May 03
UKERNA	UK	http://www.ukerna.ac.uk/	Native	June 03
ULAKBIM	Turkey	http://www.ulakbim.gov.tr/	Native	May 03

Table 7. IPv6 Deployment in NRENs and connectivity to GEANT.

Although not an NREN, CERN is directly connected to GÉANT since May 2003. GÉANT has International peering with Abilene, CANARIE, ESnet and SINET since the middle of 2003 with a native connection type.

The IPv6 traffic [R43] is still significantly lesser than IPv4 traffic. The main services offered are DNS, FTP mirroring, websites, video-steaming. IPv6 multicast becomes a new service for some NRENs.

6.6.2. The last mile

This famous last mile is often a source of problems in an end-to-end approach in networks, particularly for high speed connectivity, QoS, monitoring, troubleshooting and Service Level Agreement. This is also true for IPv6 uptake; not all the regional, metropolitan and campus networks today provide an operational IPv6 service with the same level of management as for the IPv4 service. This problem can be solved by the mechanisms described in chapter 4 but the network engineering task will be certainly more complicated. To prove the opportunity to go deeper into IPv6 in EGEE, a testbed involving all the connectivity models in an IPv4/IPv6 cohabitation world could be very

interesting for the Grid by testing deployment, scalability and performance of some transition scenarios (Figure 20).

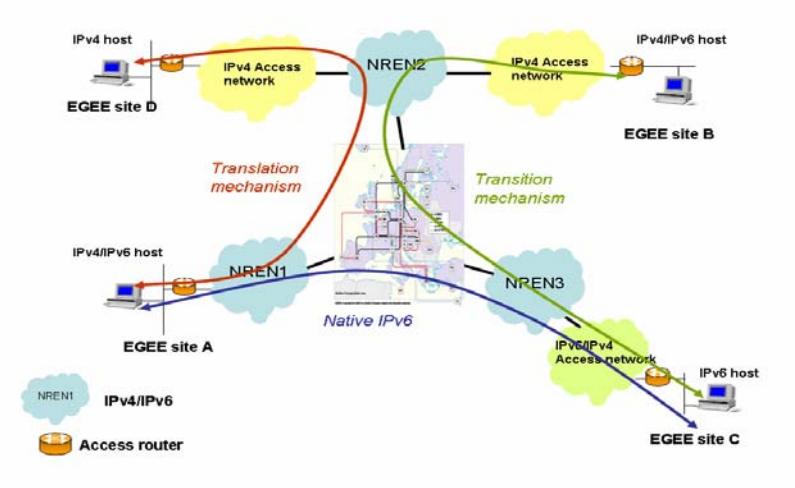

Figure 20. Possible IPv6 tesbed

6.7. EXAMPLE OF IPV6 USE IN A GRID CONTEXT

This example is about one specialised case where IPv6 addressing provides advantages, namely wide area L2 links which happen to be commonly used in the communities which are also currently major Grid users. It has been shown [R28] that Layer 2 connectivity between remote server farms seems to have some positive effects for the Grid middleware.

Layer 2 connectivity between remote sites can be quite easily achieved using several techniques, like MPLS tunnels and WDM equipment. The following drawings depict two possible scenarios (Figure 21):

1. Layer 2 connectivity made with MPLS LSPs;
2. Layer 2 connectivity made with “Lightpath”.

What is not very often considered are the difficulties that arise when it becomes necessary to assign Layer 3 addresses to the equipment connected to the same Layer 2 link that span two different remote organizations. With IPv4 there is not an obvious choice: the public address space of one organization can be used, but this will cause the awkward situation of having to use foreigner addresses in the other(s) organization(s). Also using the RFC1918 [R9] addresses is dicey because they are widely used and there is the risk of overlapping. Even more, the IPv4 addresses being a limited resource, the solution does not scale very well in computer Grids where thousands of CPUs will soon be available. And in any case an agreement and the consequent manual configuration will always be necessary.

IPv6 Link local addresses and the Stateless configuration feature provide a very elegant solution that does not require any Layer 3 manual configuration: as soon as a host is connected to the network, its IPv6 stack will provide to its interface a unique address and it will be able to communicate with any other host connected to the same Layer 2 link. Furthermore, during the standard configuration process, each host will inject to the link some packets that can eventually allow a server to identify who is present on the network.

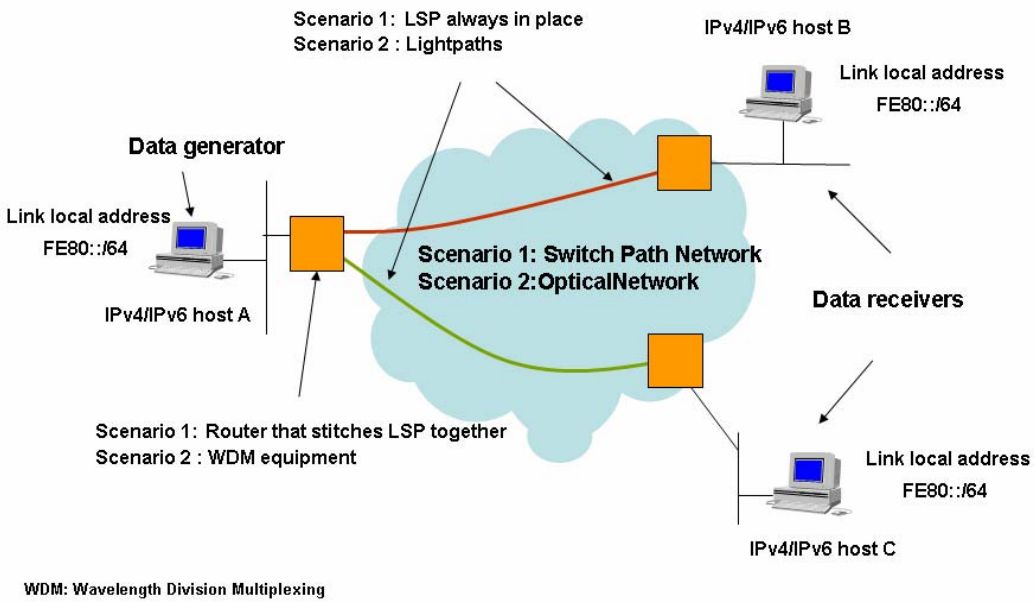

Figure 21. Layer 2 connectivity made with MPLS LSPs or "Lightpath"

7. CONCLUSION

It is expected that in practice IPv4 and IPv6 will coexist for many years, and that applications and middleware will have to live in the resulting "dual stack" network. For example, if one site in a Grid has decided to migrate aggressively to IPv6, and another site in the same Grid has no such plan, the middleware will have to deal with this situation in such a way as to conceal it from the end user applications. It should be noted that although direct translation between IPv4 and IPv6 packet formats is theoretically possible (and has been specified by the IETF), it is inconvenient because it introduces all the same disadvantages as IPv4 NAT. A better approach is for applications to be able to speak either IPv4 or IPv6 according to the destination - if DNS returns an IPv6 address, use it; otherwise use IPv4. This allows the system to exploit islands of IPv6 connectivity in an IPv4 ocean, or conversely when conversion to IPv6 is almost complete. The normal approach is for a site that wishes to convert to IPv6 to run both protocols in parallel, in the knowledge that it may be many years before IPv4 can be switched off. This is well established practice in the sites and NRENs that already offer IPv6 service.

Recommendations have already been addressed to port a middleware package to IPv6 and mechanisms and approaches have already been demonstrated when providing for example Globus Toolkit 3 with IPv6 support. The principal impacts on middleware fall into four categories, and the first approach of the IPv6 impact on gLite in this document confirms this analysis:

1. Use of the new socket API (especially concerns C code) [R21];
2. Use of at least JDK 1.4, preferably JDK 1.5, for Java;
3. Respect of a minor enhancement in URI/URL syntax [R16];
4. Format changes wherever IP addresses appear explicitly (GUIs, ACLs…).

The development and implementation of applications should take into account IP-dependencies, but it does not mean that everything already existing should be automatically ported to IPv6.

Regarding the emergence of new countries in Grid Computing (India, China, and Latin America), the combination of IPv6 and Grid systems will become soon a necessity, so that the network protocol level should be taken into account as soon as possible to avoid any further problems. The best way for software to work over IPv6 would be undoubtedly to adopt as soon as possible IP-neutral considerations as regards development and implementation of standards and protocols.

Applications like voice over IP, peer to peer and certainly Grids will ask for an efficient support of IPv6 in the networks; in this future prospect it is necessary to make gLite IPv6-compliant, though the diversity of the software means this is a major task. In order to succeed, resources and experience from various activities need to team up and existing recommendations from international bodies must be followed.

IPv6 integration requires various skills and involves numerous capabilities and resources from network, development, testing and operations. In the same way that UCL and UoS have led by a parallel task for Globus Toolkit in 6Net project, it can be envisage that a future integration of IPv6 in EGEE should the aim of a related-project.