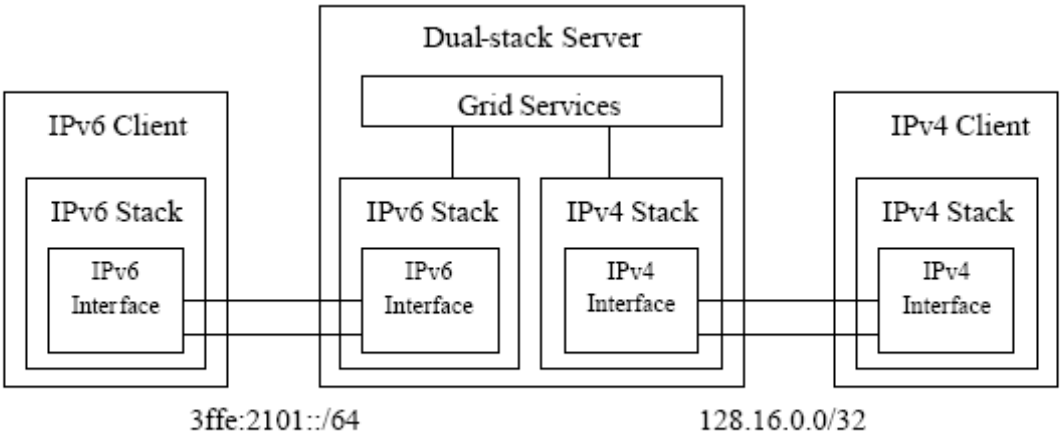
1.1. REVIEW ON IPV6 COMPATIBILITY OF GLOBUS

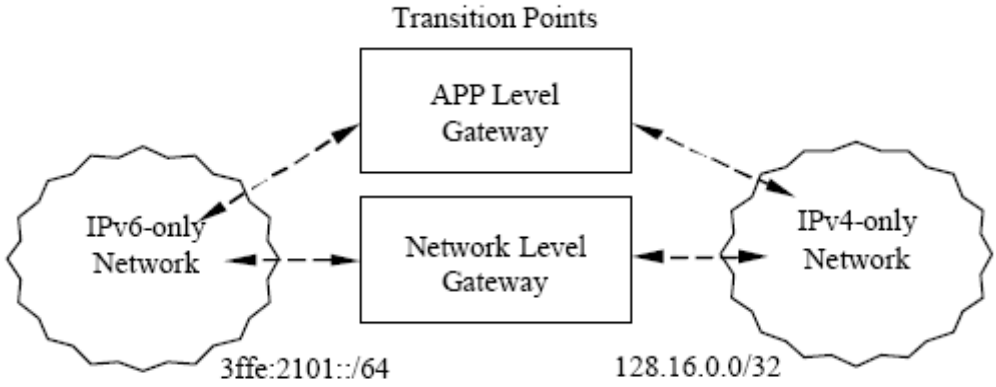
Compatibility with IPv6 has been introduced in Globus Toolkit® from 3.2 Core Release, as stated by the official document (see http://www.globus.org/toolkit/releasenotes/3.2.0/core_notes.html).

Further details on the subject may be found in "Moving Grid Systems Into the IPv6 Era", by S. Jiang, P. O'Hanlon and P. Kirstein we find:

[…] The GT3 is designed to work with IPv4, though many aspects are compatible with IPv6. We discuss our attempts to provide dual-stack, IPv4 and IPv6, facilities in Grid systems in this paper. When the Grid systems are IPv6-enabled, we will be able to experiment with the several features that become possible with IPv6 support like mobility, security and auto-configuration. […] GT3 is only fully working on Linux systems. […] GT3 is mainly written in Java. For the IPv6 support, we use Sun Java SDK 1.4.1 on our IPv6-enabled Grid testbed. […]

The following pictures show the design of the hybrid configuration for the IPv4/IPv6 protocols proposed in the mentioned paper.





]

1.1.1. Generalities on java

When a IPv6-compatible kernel is available and the IPv6 module is loaded, Linux system libraries provide a few IPv6 data structures, such as sockaddr_in6, in6_addr and in6addr_loopback, while IPv6 system functions, such as inet_ntop( ) and inet_pton( ), are available. Nevertheless, this solution are not protocol-independent. Therefore, IP-independent data structures, such as addrinfo and sockaddr_storage, and functions, such as getaddrinfo( ) and getnameinfo( ) should be preferred on dual-stack servers and server applications.

IPv6 support is provided on Solaris and Linux since JDK 1.4 and JDK 1.5 on WindowsXP/2003.. Within Java SDK 1.4, the class java.net.InetAddress has two direct subclasses: java.net.Inet4Address and java.net.Inet6Address, providing support to IPv4 and IPv6 addresses. The InetAddress class uses the Host Name Resolution mechanisms to resolve host names to their appropriate host address type. Additionally there are various system preferences that can influence protocol preferences, such as perferIPv6Addresses and perferIPv4Stack.

The same notes are reported in the “IPv6 Migration guidelines for the EGEE middleware” of EUChinaGRID Project.

1.1.2. Associated applications

It must be highlighted that, in order to ensure IPv6 support, all network-associated applications used within Globus need to be IPv6-enabled:

Java DataBase Connectivity, which is used for Reliable File Transfer, needs an IPv6 patch. As recommended by the Globus Implementation Group, Jakarta Tomcat is used as the web container for the Grid services on a Grid server. The container environment needs to provide IPv6 Web services for Grid services. Tomcat version 5 has been tested

with IPv6 capabilities. Other Web service container environments, such as IBM Websphere and Microsoft .NET, are being investigated in this survey as well.

1.1.3. Networking support for IPv6

In order to run IPv6 tests over a network rather than on local hosts only, IPv6 support for networking is essential. It requires IPv6-enabled routers, which provide forwarding and dynamic routing, and support from IPv6-enabled network services, such as IPv6 DNS, Web services, etc. A number of the major router manufactures provide now basic IPv6 support and are beginning to provide more advanced support such as hardware forwarding. Support for IPv6 in the DNS – provides hostname and IPv6 address resolution which may be provided over IPv4 and/or IPv6 connection. For the communication in heterogeneous IPv4/IPv6 networks, there are many approaches to the provision of transition aids. They need to be considered when building an IPv6 environment within or around current global IPv4 networks.

1.1.4. Integration of IPV6 into Globus

The integration of IPv6 into Grid systems starts with finding IP dependencies in the network protocols. The implementation of network APIs on applications may involve a few IP-dependent functions. A kind of approach will be illustrated in the following sections, using the IPv6 porting of Globus as an example. A number of modifications should be applied for IP-dependent operations. In order to operate in heterogeneous IPv4/IPv6 networks, a few configuration options are needed.

1.1.5. WHAT's in the Globus toolkit InteGration of IPV6 into Globus

The Globus Toolkit isn't a turnkey solution for a specific problem, but a collection of solutions to sub-problems that have proven useful in real projects: it is a genuine "toolkit," or a collection of tools, and specific applications may find different combinations of Globus Toolkit components more useful than others.

- **Two Very Important Software Development Kits**
  The following SDKs are critical for developing new Grid software services within the OGSA framework.
  - Web Services Core (WS-Core) Implementation. Used to develop and run OGSA-compliant Grid Services (available in Java and C/C++)
  - Grid Security Infrastructure (GSI) Implementation. Used to secure communication (e.g., between services and clients)
- **A Variety of Basic Grid Services**
  These services are popular among current Grid application and system builders. They provide uniform interfaces to the most typical system elements; they currently include both OGSA and non-OGSA implementations.

- Computing / Processing Power (GRAM)
- Data Management (GridFTP, DAI, RLS)
- Monitoring/Discovery (MDS)
- Authorization/Security (CAS)
- Currently under development: Telecontrol (NTCP/GTCP), Metadata (MCS), Virtual Data (Chimera, Pegasus)

- **Developer APIs**
  All services and tools mentioned above include C/C++ libraries and Java classes for building Grid-aware applications and tools.
- **Tools and Examples**
  All services and tools mentioned above include tools and examples based on the developer APIs and associated services.

Obviously in this schema we see the main parts concerning our goal to study the IPv6 compatibility of grid middleware.

From the documents concerning next deployments, it appears that, to date, developers are mainly dealing with IPv4 environment, i.e. IPv6 is beginning to be widely deployed. It provides enough address space and enables easy network management. Thus, it solves most problems that NATs try to solve. However, it is still questionable whether IPv6 can replace NATs completely.