

# Moving Grid Systems into the IPv6 Era

Sheng JIANG, Piers O'Hanlon, Peter Kirstein

Department of Computer Science  
University College London  
Gower Street, WC1E 6BT  
London, United Kingdom  
{S.Jiang, P.Ohanlon, P.Kirstein}@cs.ucl.ac.uk

**Abstract.** This paper focuses on integrating IPv6 functionality into Grid systems. We outline the advantages of IPv6 and the benefit to Grid systems. We then introduce our methodology and our efforts to provide IPv6 support on Grid systems using the Globus Toolkit Version 3 as our concrete working example. The status of global Grid IPv6 activities is introduced. We conclude by summarising how to bring IPv6 into Grid systems.

## 1. Overview – Grid System Working With IP Networks

During the last few years, Grid systems [6, 11, 12, 13, 14, 21] have emerged to perform large-scale computation and data storage over IP-enabled data communication networks. They use distributed, potentially remote, resources to optimise computation and storage resources.

Grid systems are normally considered as network middleware [1], since they lie between applications and network resources. The data of Grid systems is transported over TCP/IP [2, 8] – currently using Internet Protocol IPv4 [18], now twenty years old. The next generation IP - IPv6 [10, 24, 28], is expected to replace IPv4 with a number of improvements.

Since IPv6 is expected to become the core protocol for next generation networks, Grid computing systems must track the migration of the lower layer network protocols to IPv6. The period of transition from IPv4 to IPv6 will be long. Hence, it is important that Grid systems allow both IPv4 and IPv6 networks to be used.

The Globus Toolkit [15, 36], developed mainly in the Argonne National Laboratory (ANL), provides the libraries and services for and Grid computing. The current edition of Globus Toolkit – Version 3 (GT3) is based on the latest Grid standards – the Open Grid Services Architecture (OGSA) [11, 26] and integrates the Grid services with the Web services.

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.

We have structured the paper in the following way. First we consider the potential advantages of IPv6, from which Grids could benefit. Then we discuss the general IPv6 host and environment. During the discussion, we survey the general steps needed to build up an IPv6 experimental environment. In Section 4 and 5, we discuss our experience of implementation of IPv6 within Grid systems using the Globus Toolkit as example. We keep the consideration as general as we can, so that others could use our approach in other Grid systems. In Section 6, we introduce the current status of global activities for getting IPv6 in Grids. Finally, we end this paper by briefly summarising how to bring IPv6 into Grid systems.

## **2. IPv6 Advantage for Grid Systems**

As network middleware, the current release of Globus Toolkit uses IPv4-based network resources to serve the upper layer applications and users. Like all widespread applications, the Globus Toolkit should be prepared to move into the IPv6 era.

The bulk of the IPv6 standards (e.g. [10]) were ratified in the Internet Engineering Task Force (IETF) in 1998. IPv6 fulfils the future demands on address space, and also addresses other features such as multicast, encryption, Quality of Service, and better support for mobile computing. In comparison to the current IPv4 protocol family, IPv6 offers a number of significant advantages. Most of these advantages will also be very useful for Grid purposes. The IPv6 data format does not really provide most of these advantages by itself. However, the design of the IPv6 protocol suite has taken the opportunity to re-design the relevant protocols with a better and more logical system; for example the IPv6 renumbering mechanism could simplify dynamic mergers and acquisitions of Virtual Organisation in Grid systems.

We address here three major advantages: bigger address space, mobility support and security support – ignoring the many other advantages and potential benefits of IPv6, such as auto-configuration [25], hooks for QoS, etc. While we take advantages of IPv6 features, we also give consideration to communication in heterogeneous IPv4/IPv6 networks.

### **2.1. Bigger Address Space**

With its 128-bit address space and much better address aggregation properties, IPv6 potentially makes massive scaling of Grid networking possible; this is important in view of the aims to deploy Grid computation globally.

With the enlarged address space, workarounds like NATs (Network Address Translators) [23] are no longer needed. This allows full, global IP connectivity for IP-based machines as well as upcoming mobile devices like PDAs and cell phones – all can benefit from full IP access through end-to-end services. There can be multiple addresses for a single interface; where the addresses can be used for different functions. The large address space allows for simpler end-to-end security, IPv6 renumbering mechanism, separated addressing and routing, etc.

## **2.2. Mobility Support**

Until recently, most Grid research has focused only on fixed systems. However, the mobility support within Grid systems will be needed as mobility takes an ever more important role in modern life. In our research, the Mobile-Grid-specific auto-configuration mechanisms are proposed to allow a Grid Mobile node to use the Grid resources available locally. As regards next generation mobile networks, IPv6 is mandated by the 3rd Generation Partnership Project [35]. Mobile IPv6 [16, 19] is accessible using general IPv6 APIs, appearing transparent to the application layer.

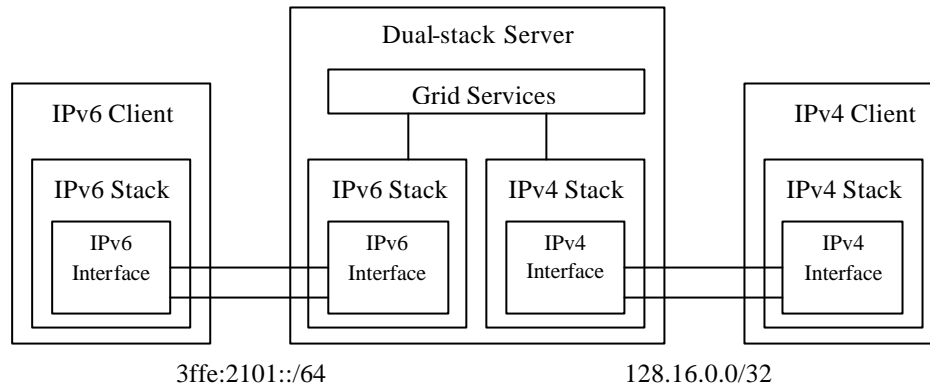
Thus, in an IPv6 implementation, there is potential support for roaming between different networks, with global notification when you leave one network and enter another. Support for roaming is possible with IPv4 too, but it is generally less efficient.

## **2.3. Built-in Security**

While scalability, performance and heterogeneity are desirable goals for any distributed systems, including Grid systems, the characteristics of computational Grids lead to security issues. Though the security improvement from IPv6 does not solve all the security problems, Grid systems can benefit from IPv6's security features. The IPv6 security and Grid Security Infrastructures are running at different levels. They can be employed together to provide better security granularity.

Besides support for mobility, security was another requirement for IPv6. The IPv6 protocol stacks are required to include IPsec [20], which allows authentication and encryption of IP traffic. With IPsec all IP traffic between two nodes can be handled without adjusting any applications. Alternatively application-level security can be employed per service if required. However, using IPsec all applications on a machine can benefit from encryption and authentication, and policies can be set on a per-host (or even per-network) basis instead of per application/service. Full IPsec security operates over IPv4 today – when there is a full end-to-end connection. If NATs are used, as often occurs in IPv4 networks, but are not needed in IPv6 ones, it is not possible to use full IPsec on the end-to-end communications.

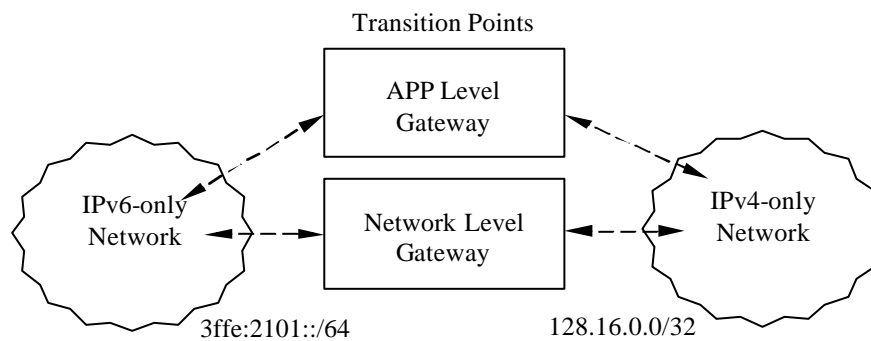
## 2.4. Communication in Heterogeneous IPv4/IPv6 Networks



**Fig. 1.** IP Communication in heterogeneous IPv4/IPv6 networks

Since there will be a period of IP transition, consideration must be given to an interim coexistence of IPv4 and IPv6 [9]. Our effort to integrate IPv6 into Grid systems takes an IP-protocol independent [27] approach, i.e. it supports both IPv4 and IPv6. The IP-independent server, shown in Fig. 1, has to be able to respond to client calls according to the IP family that the client uses. Since IPv4-only machines will exist for many more years, while IPv6-only machines are starting to appear, it is necessary to provide support to both IPv4-only and IPv6-only environments.

The client decides which version of IP is to be used. The Grid server responds to the client calls according to which IP family the client uses. For instance, in Fig. 1, an IP-independent Grid server starts and listens on both its IPv4 and IPv6 interfaces. When an IPv4 client connects over IPv4, the Grid server uses IPv4 interface to callback; only IPv4 communication takes place – similarly with IPv6. With dual-stack servers, the client can choose which IP family is the default or preferred (see Section 4.4).



**Fig. 2.** IP Transition between heterogeneous IPv4-only and IPv6-only networks

For communication in heterogeneous IPv4/IPv6 networks, there are a number of network transition aids [7, 22], which essentially translate the packet headers between IPv4 and IPv6, leaving the payload untouched. These approaches may work in certain circumstances for Grid applications. A higher level approach, which is employed by other services for transition, is application level gatewaying. This operates in a dual-stack node and actually does an application level translation of the payload of the packets between the two communicating nodes. (See Fig2.)

### **3. IPv6 Environment for Grid**

To start any IPv6 experiments, the host must be IPv6-enabled. The IPv6-capable application API libraries are required in order to run the IPv6-enabled applications or IP-independent applications over IPv6. All network-associated applications, such as network-sharing database applications and web containers, need to be IPv6-enabled to run IPv6 tests. In order to run tests over a network rather than only on local hosts, IPv6 support on the network is essential. We discuss how to build up an IPv6 environment step-by-step, using our IPv6-enabled Grid testbed as an example.

#### **3.1. Operating System Support on Hosts**

The IPv6 support on hosts depends on the operating system and its kernel. For the time being, we restrict ourselves to the LINUX/PC platform since GT3 is only fully working on Linux systems. UCL has set up an IPv6-enabled Grid testbed, which includes 5 nodes running Linux Red Hat 8 and 3 nodes running Linux Red Hat 7.3 with recompiled kernel, for IPv6 support.

From Linux Red Hat 8 (kernel 2.4.18), the IPv6 module is provided with auto-load as default. On the earlier distributions of Linux, users had to re-compile the kernel to get the IPv6 support [29]. On Windows 2000, an IPv6 preview package (free download [30]) is available with limited functionality. The IPv6 package in Windows XP, which provides a better IPv6 support, is distributed with the edition, but requests individual installation [31].

#### **3.2. IPv6-capable Application API Libraries**

IPv6-capable application API libraries need to provide support for upper-layer applications. GT3 is mainly written in Java. For the IPv6 support, we use Sun Java SDK 1.4.1 on our IPv6-enabled Grid testbed.

With the IPv6-compatible kernel and the IPv6 module loaded, Linux system libraries provide a few IPv6 data structures, such as `sockaddr_in6`, `in6_addr` and `in6addr_loopback`, and IPv6 system functions, such as `inet_ntop()` and `inet_pton()`, are available to be used. But they are not IP-version-independent. To be IP-independent, IP-independent data structures, such as `addrinfo` and `sockaddr_storage`, and functions, such as `getaddrinfo()` and `getnameinfo()` should be used on dual-stack servers and server applications.

As a platform-independent runtime environment, JDK 1.4 provides the IPv6 support on Solaris and Linux. JDK 1.5 is planned to provide IPv6 for WinXP. Within Java SDK 1.4, the class `java.net.InetAddress` has two direct subclasses: `java.net.Inet4Address` and `java.net.Inet6Address`. They provide the support for 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 `preferIPv6Addresses` and `preferIPv4Stack`.

### **3.3. Associated Applications**

The Globus system also utilises external applications. All network associated applications need to be IPv6-enabled as well. In GT3, the Java run-time environment needs to be IPv6-enabled as mentioned earlier. 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 our Grid IPv6 project as well.

### **3.4. Networking Support for IPv6**

In order to run IPv6 tests over a network rather than only on local hosts, 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 (see Section 2.4). They need to be considered when building an IPv6 environment within or around current global IPv4 networks.

## **4. Integration of IPv6 into Globus**

The integration of IPv6 into the Grid systems starts with finding IP-version-dependencies in the network protocols. The implementation of network APIs within applications may involve a few IP-dependent functions. We introduce our methodologies in later sections using Globus IPv6 porting as an example. A number of modifications need to be made for IP-dependent operation. In order to operate in heterogeneous IPv4/IPv6 networks, a few configuration options are needed.

#### 4.1. Methods of Finding IP Dependencies

To find out exactly which lower-layer protocols and APIs are being used, two approaches are taken – firstly the ‘top down’ approach where we execute some upper layer applications. Secondly the ‘bottom up’ approach where we monitor all the data traffic between nodes and on the Loopback interface. The following have been identified as relevant; they have been modified to be IP-independent:

- ? Which network protocols are involved and whether they are IP-dependent
- ? Where to get or generate IP addresses
- ? How to generate URLs [4] and URIs [5] with IP addresses
- ? How to create sockets and network connections
- ? Hard-coded IPv4 addresses

#### 4.2. GT3 Protocols Modification for IPv6

The specifications of a few protocols have needed to be modified to suit IPv6. In the Globus Toolkit, Grid FTP is being modified in a way similar to FTP (RFC2428 [3] – “FTP extend for IPv6 & NATs”). Correspondingly, the specific implementations of these protocols need modification as well. Within the Globus project, GridFTP is currently implemented in standard C. A new IP-independent network module known as globus\_XIO is being developed for use by for GridFTP.

In our porting, all protocols involved must be examined, we are in contact with the Globus implementation group on this, and have examined most. The GGF IPv6-WG is surveying all GGF specifications for IP version dependence (see Section 6). This is also happening with the Internet Engineering Task Force.

#### 4.3. IPv6 Modification in GT3 Implementation

While modifying the IP-independent protocols and their implementation, the implementation of the Globus Toolkit has needed to be modified as well. Corresponding to the IP network functions found using the method we mentioned in Section 4.1, the following modifications have been made to various modules to realise IPv6 functionality in GT3 while Java SDK has provided the IP-independent data structures and functions:

- ? Where to get or generate IP addresses
  - “Localhost” or particular hostname are used in both the Globus configuration file and Globus initial functions. Then IP-independent functions (InetAddress.getByName in Java, getaddrinfo in standard C) are used everywhere need to translate hostname into IP address.
- ? How to generate URLs and URIs with IP addresses
  - All URLs and URIs generating functions have been modified in order to handle the particular format of the literal IPv6 addresses in URLs [17]. It ensures the literal IPv6 addresses in URLs are included by square brackets.
- ? Hard-coded IPv4 addresses

All hard-coded IPv4 addresses have been replaced by “localhost” or particular hostnames. Then use IP-independent functions look up the IP addresses when need to translate hostname into IP address.

#### **4.4. Configuration for IP Operations**

In GT3, a few configuration options are available to allow the user to choose the startup IP bindings. To use hostname instead of IP address, the user needs to set up the configuration option “publishHostName” be “true” in the globalConfiguration section of server-config.wsdd. If there are other IP addresses that are associated with a host, such as a particular IPv6 hostname, the user needs to set up which the hostname is used with the configuration option “logicalHost”. By Java default options, IPv6 address has higher priority on dual-stack machines. To operate in IPv4-only network or set IPv4 be higher priority, user needs to set Java system properties “preferIPv6Addresses” be “false” and “preferIPv4Stack” be “true”.

### **5. IPv6 Grid Tests**

Having realised the IPv6-enabled Globus system, a few experiments and tests have been run successfully in different scenarios. These include ones where some of the system is IPv4-only, whereas some are IPv6-only, and most dual stack. These tests were to allow for the scenario of a transition to IPv6 since most current Grid usage and systems are mainly in IPv4 now. A few upper-layer services have been run successfully over IPv6-enabled Grid systems in order to validate the adaptation between IPv6 and Grid applications.

#### **5.1. GT3 Version Dependency**

The GT3 alpha version became available at the beginning of 2003. Since then, we have mainly worked on the GT3 alpha version because it provided the most verbose debug information. We have demonstrated successfully IPv6 functionality and run our tests on the GT3 alpha version. We started to move to release GT 3.0 when it came out at the end of June 2003. More components are involved in GT 3.0; some of GT 3.0 components involved more IP-dependent issues; we are still surveying them. Globus Resource Allocation Manager (GRAM) have been identified as not working in GT 3.0, which worked in GT3 alpha, due to Java IPv6 reverse looking up bug.

#### **5.2. IPv6 Test Scenarios and Porting Stages**

The IPv6-enabled tests and porting start from an IPv4-only test in the heterogeneous IPv4/IPv6 environment. This may differ from the IPv4-only environment. Then, IPv6-only porting has been done with the minimal modification. During this stage, as much as possible, IP-independent functions and data structures



have been used instead of IPv4-only or IPv6-only one. The situation becomes much more complicated for IPv4/IPv6 dual stacks. In the dual-stack environment, parallel independent support for both IPv4 and IPv6 must be provided. The Grid server starts with the IP-independent hostname and respond the client calls according to the IP family user used. After modification has been made, we do run IPv4-only tests as well, since most Globus users are obviously still IPv4.

### **5.3. Test Services**

A few upper-layer services have been run successfully over IPv6-enabled Grid systems in order to confirm the adaptation between IPv6 and Grid applications.

The services or applications that are distributed with GT3 are used as general initial test services in our test scenarios. We managed to access Grid Services through IPv6 interfaces by using the OGSA graphic user interface service browser. We managed to submit remote GRAM jobs through IPv6 interfaces. These tests were also successfully run with IPv4 in our heterogeneous IP Grid testbed.

We also tested systems with externally developed GT3 services as well. The E Protein project [32] had developed a remote executing service based on GT3 GRAM. It was successfully transplanted to the IPv6-enabled Globus infrastructure. We have also begun working with a bio-chemical simulation project, Materials Simulation [33], to test their OGSA-compatible simulation services on IPv6.

## **6. Grid IPv6 Standardisation Status**

Since February 2003, when the Global Grid Forum (GGF) and the IPv6 Forum announced a liaison relationship to drive New Generation Applications deployment worldwide, IPv6 has become relevant to Grid activities.

An IPv6-Grid Working Group has been set up in GGF. The GGF IPv6 WG [34] is currently working on two drafts: "IP version dependencies in GGF specifications" and "Guidelines for IP independence in GGF specifications", which have been presented in GGF9 (October 2003). UCL makes contributions for both of drafts. UCL is currently involved with a number of activities that bring IPv6 into other Grid systems, such as Sun Grid Engine.

## **7. Conclusion and Future Work**

Our main contribution is to ensure that we have a version of GT3 at UCL giving IPv6 support, which has been tested with a number of Grid applications. We have had reasonable success. The mechanisms and approach for integrating IPv6 into Globus introduced in this paper could benefit other research to integrate IPv6 into other Grid systems.

While keeping modification in GT3 to a minimum during the tests and experiments, we have identified a few further modifications still needed, which will

make IPv6 configuration and operation easier and smoother. They will be implemented on the stable Globus Toolkit distribution.

We plan to provide a variety of services over IPv6 based on UCL's own Grid services. In later research, we will look at issues of mobility and security in Grids. Mobility support in Grid computing systems will be achieved based on using Mobile IPv6. Our proposed Mobile-Grid-specific reconfiguration mechanisms will be developed to meet the particular requests for transparent, dynamic and automatic network services in Grid computing systems.

## Acknowledgement

The authors wish to thank Soren-Aksel Sørensen, University College London and to Tim Chown and David Mills, of the University of Southampton, who made substantial contributions to the work of this paper. This work has done under the aegis of the IST, 6NET project. The support of the European Communities is gratefully acknowledged.

## Reference

1. Aiken, B. 2000. "Network Policy and Services: A Report of a Workshop on Middleware," RFC 2768.
2. Albitz, P. 2001. "DNS and Bind, 4th Edition", Published by: O'Reilly.
3. Allman M. 1998. "FTP Extensions for IPv6 and NATs", RFC 2428.
4. Berners-Lee T. 1994. "Uniform Resource Locators (URL)", RFC 1738.
5. Berners-Lee T. 1998. "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396.
6. Chervenak, A. 2001. "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets," Journal of Network and Computer Applications 23: 187-200.
7. Chown T. 2003. "Advanced Aids to Deployment", Deliverable 17, the 6WINIT Project, UCL.
8. Comer D. 2000. "Internetworking with TCP/IP, Volume 1, 4th edition," Published by Prentice Hall.
9. Davies, J. 2002. "IPv6/IPv4 Coexistence and Migration," White paper of Microsoft Corporation.
10. Deering, S. 1998. "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460.
11. Foster I. 2002. "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Global Grid Forum.
12. Foster, I. 2002. "The Grid: A New Infrastructure of 21st Century Science," Physics Today Volume 55: 42-52.
13. Foster, I. 2001. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International J. Supercomputer Applications Volume 15(3).
14. Foster, I. 1998. "The Grid: Blueprint for a New Computing Infrastructure," Published by: Morgan Kaufmann.
15. Foster, I. 1997. "Globus: A Metacomputing Infrastructure Toolkit," Intl J. Supercomputer Applications Volume 11(2): 115-128
16. Fritsche, W. 2000. "Mobile IPv6 - the Mobility Support for Next Generation Internet," IPv6 forum.

17. Hinden R. 1999. "Format for Literal IPv6 Addresses in URL's", RFC 2732.
18. ISI. 1981 "Internet Protocol DARPA Internet Program Protocol Specification", RFC 791.
19. Johnson, D. 2003. "Mobility Support in IPv6", IETF Internet Draft.
20. Kent, S. 1998. "Security Architecture for the Internet Protocol", RFC 2401.
21. Laszewski, von G. 1999. "Grid Infrastructure to Support Science Portals for Large Scale Instruments," Proc. of the Workshop Distributed Computing on the Web (DCW).
22. R. Gilligan. 2000 "Transition Mechanisms for IPv6 Hosts and Routers", RFC 2893.
23. Srisuresh P. 2001. "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022.
24. Thomson S. 2003. "Basic Socket Interface Extensions for IPv6", RFC 3493.
25. Thomson, S. 1998. "IPv6 Stateless Address Autoconfiguration," Request For Comments 2462.
26. Tuecke, S. 2003. "Open Grid Service Infrastructure (OGSI) – Version 1.0 (draft)", Global Grid Forum.
27. Shin, M. 2003. "Application Aspects of IPv6 Transition", IETF Internet Draft.
28. Stevens, W. 2003. "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542.
29. <http://www.bieringer.de/linux/IPv6/>
30. <http://msdn.microsoft.com/downloads/sdks/platform/tpipv6.asp>.
31. <http://www.microsoft.com/windowsxp/pro/techinfo/administration/ipv6/>
32. [http://grid.ucl.ac.uk/proj\\_epro.html](http://grid.ucl.ac.uk/proj_epro.html)
33. [http://grid.ucl.ac.uk/proj\\_mats.html](http://grid.ucl.ac.uk/proj_mats.html)
34. <http://forge.gridforum.org/projects/ipv6-wg/>
35. <http://www.3gpp.org>
36. <http://www.globus.org>