Sheng Jiang, Piers O'Hanlon, Peter Kirstein (2004) *Moving Grid Systems into the IPv6 Era*, Grid and Cooperative Computing: Second International Workshop, GCC 2003, Shanghai, China, December 7-10, 2003, Revised Papers, Part II, Volume 3033, pp 490-499. http://dx.doi.org/10.1007/978-3-540-24680-0_83

PROCEEDINGS PAPER

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 Grid could get from IPv6. 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 summary how to bring IPv6 into Grid systems.

Key words: Grid computing, Globus, IPv6.

1. Overview – Grid System Working With IP Networks

During the last few years, Grid systems [ ] 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 called a network middleware [ ], since they lie between applications and network resources. The data of Grid systems is transported over TCP/IP to ensure reliable transport. The current version of the Internet Protocol - IPv4, is now twenty years old. The next generation IP - IPv6, is expected to replace IPv4. A large number of deficiencies of IPv4 have been remedied by IPv6. For example the 32-bit address space of IPv4 will run out fairly soon; IPv6 has a 128-bit address space.

So far all the existing Grid systems have supported only IPv4. IPv6 is considered as the core protocol for next generation networks. Grid computing systems must track the migration of the lower layer network protocols [ ] to IPv6. While we are confident that the Internet will eventually transition to an all-IPv6 one, the period of transition will be long. For this reason, we consider it important to ensure that Grid systems allow both IPv4 and IPv6 networks to be used.

In this paper, the Globus Toolkit is used as a concrete example of a Grid system. Globus is a large active academic project to develop Grid middleware. It is being developed mainly in the Argonne National Laboratory (ANL). The Globus Toolkit [ ], which is the concrete result of the Globus project, provides the libraries and services for computational Grid and Grid applications. The current edition of Globus Toolkit Version 3 (GT3) is designed based on the latest Grid standards – the Open Grid Services Architecture (OGSA) and integrates the Grid services with the Web services. The current Globus Toolkit is designed to work with IPv4, though many aspects are designed to be 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. University College London (UCL) and the University of Southampton (UoS) started porting Globus Toolkit Version 2 (GT2) to an IPv6-enabled form during the middle of 2002. This had almost reached the state of a working system when in January 2003 an alpha release of GT3 was made. GT3 is so different, that UCL decided to terminate its work on GT2 forthwith, even without a fully working system, to concentrate on GT3. This activity is being carried out in collaboration with the Globus development team in ANL. A Japanese project 6Grid continues to complete the GT2 porting.

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 introduce the general steps needed to build up an IPv6 experimental environment. In Section 4 and 5, we discuss our experience on implementation of IPv6 within Grid systems using the Globus Toolkit as example. We keep the consideration as general as we can, so that user could use our consideration in other Grid systems rather than only Globus. 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, Globus Toolkit should be prepared to move into the IPv6 era. The bulk of the IPv6 standards (e.g. [ ]) were ratified in the Internet Engineering Task Force (IETF) in 1999. IPv6 fulfils the future demands on address space, and addresses also 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. There are many other advantages and potential benefits of IPv6, such as auto-configuration [ ], hooks for QoS, etc. We do not address these in this paper. While we take advantages of IPv6 features, we also give consideration to communication in heterogeneous IPv4/IPv6 networks as well.

2.1. Bigger address space

IPv4 will run out its 32-bit address space shortly. With 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.

The bigger address space of IPv6 is the most obvious enhancement of IPv6 over IPv4. While today's Internet architecture is based on 32-bit wide addresses, the new version has 128-bit space available for addressing. With the enlarged address space, workarounds like NATs (Network Address Translators) [ ] do not have to be used anymore. This allows full, unconstrained, IP connectivity for today's IP-based machines as well as upcoming mobile devices like PDAs and cell phones – all can benefit from full IP access through end-end services. There can be multiple addresses for a single interface; where its addresses can be used for different functions. The large address space makes a few advanced networking features and end-end encryption be possible, such as an IPv6 renumbering mechanism, separate addressing and routing,

2.2. Mobility support

Up to now, most Grid research has focused only on fixed systems. However, the mobility support within Grid systems is needed sooner or later as mobility takes an ever more important role in modern life. In an our research, the Mobile-Grid-specific auto-configuration mechanisms are proposed to allow a Grid Mobile node to use the geographical local resource as the Grid resources. Mobile IPv6 [ ] is implemented based on general IPv6 APIs and appears transparent to the application layer.

In the context of mobile devices and IP, it is important to note that a special protocol is needed to support mobility; an implementation of this protocol – called “Mobile IP” [ ] – is one of the requirements for every IPv6 stack. 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 system, the characteristics of computational Grids lead to security problems. 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 the successor to today's Internet Protocol version. As a result, IPv6 protocol stacks are required to include IPSec [ ]. IPSec security works with IPv4 today – when there is an end-end transmission path with constant source and destination addresses. If NATs are used as often occurs in IPv4 networks, it is not possible to use IPSec on an end-end communication. The biggest advantage of IPv6 for security is that, with its addressing scheme, no NATs are needed; this removes some obstacles to the use of IPSec and therefore provides greater security. IPSec allows authentication and encryption of IP traffic. All IP traffic between two nodes can be handled without adjusting any applications, while the application-level security approach could be employed over the IP layer in addition if designed. Thus 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.

2.4. Communication in heterogeneous IPv4/IPv6 networks

Since there will be a period of IP transition, considerations must be given to an interim coexistence of IPv4 and IPv6 [ ]. Our effort to integrate an API within the Grid systems is IP-protocol independent, 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 give supports to both IPv4-only and IPv6-only environments. For the communication in heterogeneous IPv4/IPv6 networks, there are many approaches to the provision of transition aids [ ]; examples are the tunnelling approach, which links IPv6 islands established over IPv6 networks together, and protocol translator approach. They need to be considered when building an IPv6 environment within or around current global IPv4 networks.

Page --> 493

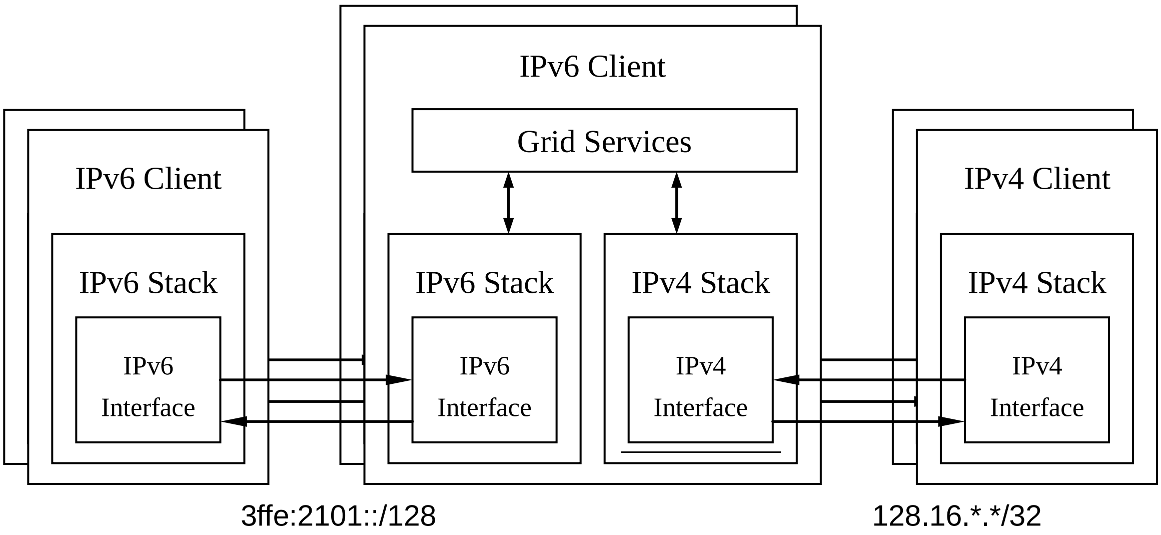

Figure 1: IP Communication in heterogeneous IPv4/IPv6 networks

On dual-stack machines, the Grid server starts with the hostnames, which are always IP-independent. It is down to the client to determine which version of IP is used. The Grid servers respond to the client calls according to which IP family client uses. For instance, in Fig. 1, an IP-independent Grid server starts with its hostnames. There is no particular IP address binding to the server, and both IPv4 and IPv6 interfaces are accessible. When an IPv4 client calls with IPv4 addresses, the Grid server uses IPv4 interface to callback; only IPv4 communication takes place. Similarly, when an IPv6 client calls with IPv6 addresses, the Grid server uses IPv6 interface to callback; only IPv6 communication takes place. For the client from the dual stacks machine, client can choose which IP family is the default or priority.

3. IPv6 Environment for Grid

To start any IPv6 experiments, we need the host to be IPv6-enabled first. Then, IPv6 capable application API libraries are required in order to run the IP-enabled or IP-independent applications. All network-associated applications, such as network-sharing database applications, 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 from networking is essential. We discuss how to build up an IPv6 environment step-by-step, using our IPv6-enabled Grid testbed as an example.

Page --> 494

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 under Linux Red Hat 8 and 3 nodes under Linux Red Hat 7.3 with recompiled kernel.

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 [ ]. On Windows 2000, an IPv6 preview package (free download [ ]) 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 [ ].

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.

A few system libraries are available on the IPv6-enabled hosts. With the IPv6 module loaded, 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 IPv6-only. To be IP-independent, both IPv4-only and IPv6-only data structures and functions should be avoided. Instead, IP-independent data structure, such as 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, Unix 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 perferIPv6Addresses and perferIPv4Stack.

3.3. Associated Applications

The Globus system also utilises extended 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. JDBC, which is used for Reliable File Transfer, need 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. For the IPv6 functionality, the “full” edition of Tomcat, which has libraries duplicated conflict with Java SDK 1.4, cannot work correctly. The “lightweight edition” of Tomcat has to be applied. Other Web service container environments, such as IBM Websphere and Microsoft .NET, are investigated in our Grid IPv6 project as well.

Page --> 495

3.4. Networking Support for IPv6

In order to run IPv6 tests over a network rather than only on local hosts, the 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. For the communication in heterogeneous IPv4/IPv6 networks, there are many approaches to the provision of transition aids [ ], such as the tunnelling approach, which links IPv6 islands established over IPv6 networks together, and protocol translator approach. 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 Grid starts from finding IP-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 section using Globus IPv6 porting as an example. Then a few modifications need to be made on these IP-dependency implementations.

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 [ethereal/tcpdump] all the data traffic between nodes and on the Loopback interface. The following network functions 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 with IP addresses
- How to create sockets and network connections
- How Callback functions work
- Hard-coded IPv4 addresses

4.2. GT3 Protocols Modification For IPv6

The definitions of a few protocols need to be modified to suit IPv6. In the Globus Toolkit, Grid FTP has already been flagged; it is similar to the IPv6 modification needed for FTP (RFC2428 – “FTP extend for IPv6 & NATs”). Correspondingly, the specific implementation of these protocols needs modification as well. The Globus Implementation Group has written a new module globus_XIO for GridFTP instead of globus_io.

In our porting, all protocols involved are examined, we are in contact with the Globus implementation group on this. Actually, IETF is surveying all their RFCs for IPv6 dependencies.

GGF IPv6-WG is surveying all GGF specifications as well.

4.3. IPv6 Modification on GT3 Implementation

While modifying the IP-independent protocols and their implementation, the implementation of the Globus Toolkit need 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 realise IPv6 functionality in GT3.

- 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 with IP addresses

An additional URL check and handle function has been embedded in order to handle the particular format of the literal IPv6 addresses in URLs [RFC2732]. It ensures the literal IPv6 addresses in URLs are included by square brackets.

- How to create network connections and how Callback functions work

On both sever side and client side, network connection are built with IP addresses. Most of binding and socket functions are IP-independent. However, we still need to make sure that they use the right data structures, which support both IPv4 and IPv6 data structures.

- Hard-coded IPv4 addresses

All hard-coded IPv4 addresses have been replace 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 set up in order that the user can choose the IP version, which is adapted to the particular networks or hosts. To use hostname instead of IP address, set up the configuration option “publishHostName” be “true” in the globalConfiguration section of server-config.wsdd. If there are other hostnames that link to the localhost, such as a particular IPv6 hostname, the user can set up which hostname is used with the configuration option “logicalHost”. At the client side, user can start client functions with particular IP version by set up Java system properties “perferIPv6Addresses” and “perferIPv4Stack” be “true” or “false”.

5. IPv6 Grid Tests

Having realised the IPv6-enabled Globus system, a few experiments and tests have been run successfully under 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.

Page --> 497

5.1. GT3 Version Dependency

The GT3 alpha version became available at the beginning of 2003. Since then, we 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; we are still surveying them. Some of GT 3.0 components, such as the Globus Resource Allocation Manager (GRAM), have been identified including more IP-dependent coding.

5.2. IPv6 test scenarios and porting stages

The IPv6-enabled tests and porting start from an IPv4-only test in the IPv6-enabled 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, the 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 had developed a remote executing Services based on GT3 GRAM services. It had been successfully transplanted to the IPv6-enabled Globus infrastructure. A bio-chemical simulation project is involving in our tests with their OGSA-compatible simulation services.

6. Grid IPv6 Activities Status

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

An IPv6-Grid Working Group has been set up in GGF. The GGF IPv6 WG is currently surveying IPv4 dependencies in the GGF Specifications. 17 approved draft specifications, 3 public draft specifications and 39 current draft specifications are being reviewed now. Two GGF IPv6 WG drafts are being written: “IP version dependencies in GGF specifications” and “Guidelines for IP independence in GGF specifications”, which will be submitted for GGF9 (Chicago, October 2003).

Page --> 498

UCL makes contributions for both of drafts. UCL is currently involved with a few activities that bring IPv6 into other Grid systems, such as Sun Grid Engine.

7. Conclusion and Future Work

In order to realise IPv6 functionality within Grid systems, the IPv6 environment is needed as the lower-layer infrastructure. Then all IP dependencies need to be modified to be IP-independent. For Java implementation, most network functions are IP-independent. The main action is to use the right version of JDK (1.4 or later). For C/C++ implementation, the system IP-independent functions must be used. In order to operation in heterogeneous IPv4/IPv6 networks, a few configuration options are needed.

While keeping the minimum modification with in GT3 during the tests and experiments, we have identified a few further modifications, which will make IPv6 configuration and operation earlier 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 by using Mobile IPv6, which is based on the same basic API as the normal 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.

Reference

[] Aiken; B. 2000. "Network Policy and Services: A Report of a Workshop on Middleware," Request for Comments 2768.

[] Albitz, P. 2001. "DNS and Bind, 4th Edition", Published by: O'Reilly.

[] Bresnahan, J. 1999. "Communication Services for Advanced Network Applications," the International Conference on Parallel and Distributed Processing Techniques and Applications Volume 4: 1861-1867.

[] 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.

[] Chown T. 2003, "Advanced Aids to Deployment", Deliverable 17, the 6WINIT Project, UCL.

[] Comer D. 2000. "Internetworking with TCP/IP, Volume 1, 4th edition," Published by Prentice Hall.

[] Davies, J. 2002. "IPv6/IPv4 Coexistence and Migration," White paper of Microsoft Corporation.

[] Deering; S. 1998. "Internet Protocol, Version 6 (IPv6) Specification," Request For Comments 2460.

[] Foster I. 2002, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Global Grid Forum.

[] Foster, I. 2002. "The Grid: A New Infrastructure of 21st Century Science," Physics Today Volume 55: 42-52.

[] Foster, I. 2001. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International J. Supercomputer Applications Volume 15(3).

[] Foster, I. 1998. "The Grid: Blueprint for a New Computing Infrastructure," Published by: Morgan Kaufmann.

[] Foster, I. 1997. "Globus: A Metacomputing Infrastructure Toolkit," Intl J.

Supercomputer Applications Volume 11(2): 115-128

[] Fritsche, W. 2000. “Mobile IPv6 - the Mobility Support for Next Generation Internet,” IPv6 forum.

[] Korva. J. 2003, “Advanced Non-Clinical Applications”, Deliverable 16, the 6WINIT Project, UCL.

[] Laszewski, von G. 1999 “Grid Infrastructure to Support Science Portals for Large Scale Instruments,” Proc. of the Workshop Distributed Computing on the Web (DCW).

[] Thomson, S. 1998. “IPv6 Stateless Address Autoconfiguration,” Request For Comments 2462.

[RFC 959] J. Postel, and J. Reynolds, “File Transfer Protocol (FTP)”, October 1985.

[RFC1900] B. Carpenter, Y. Rekhter, “Renumbering Needs Work”, February 1996.

[RFC 2068] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1”, January, 1997.

[RFC2396] T. Berners-Lee, R. Fielding, L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax”, August 1998.

[RFC2460] S. Deering, R. Hinden. “Internet Protocol, Version 6 (IPv6) Specification”, December 1998.

[RFC2428] M. Allman, S. Ostermann, C. Metz, “FTP Extensions for IPv6 and NATs”, September, 1998.

[RFC2732] R. Hinden, B. Carpenter, L. Masinter, “Format for Literal IPv6 Addresses in URL’s”, December 1999.

[RFC2743] J. Linn, “Generic Security Service Application Program Interface”, January 2000.

[RFC3542] W. Stevens, M. Thomas, E. Nordmark, T. Jinmei, “Advanced Sockets Application Program Interface (API) for IPv6”, May 2003.

[] I. Foster, C. Kesselman, J. Nick, S. Tuecke, “The Physiology of the Grid – An Open Grid Services Architecture for Distributed Systems Integration” http://www.Globus.org/research/papers/OGSA.pdf

[] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S.Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, P. Vanderbilt, “Open Grid Service Infrastructure (OGSI) – Version 1.0 (draft)”,

[] I. Foster, “The Grid: A New Infrastructure of 21st Century Science”, Physics Today Volume 55: 42-52, 2002.

[] I. Foster, C. Kesselman, “Globus: A Metacomputing Infrastructure Toolkit”, Intl J. Supercomputer Applications Volume 11(2): 115-128, 1997.

[] mobile IPv6 IETF draft

[] Application Aspects of IPv6 Transition, Internet Draft, June 2003

http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=27b1e6a6-bbdd-43c9-af57-dae19795a088

http://www.bieringer.de/linux/IPv6/

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxppro/Plan/FAQIPV6.asp

http://www.cs.ucl.ac.uk/staff/sjiang/webpage/how-to-IPv6-Globus.htm