IPv6 Applications

Location, country

Date

Speaker name (or email address)

Copy …Rights

- *Looking for a contact ?*
  - *Mail to : martin.potts@martel-consulting.ch*
  - *Or bernard.tuy@renater.fr*

Contributions

– Jim Bound, HP
– Brian Carpenter, IBM, Switzerland
– Tim Chown, UoS, UK
– Johann Fiedler, FhG, Germany
– Ian Foster, Argonne National Labs
– Tony Hain, Cisco, USA
– Sheng Jiang, Peter Kirstein, Piers O'Hanlon, Socrates Varakliotis, UCL, UK
– R. Ruppelt, FhG, Germany
– Jacques Saint Blancart, IBM, France
– Laurent Toutain, ENST-Bretagne – IRISA, France
– Bernard Tuy, Renater, France

Table of Contents

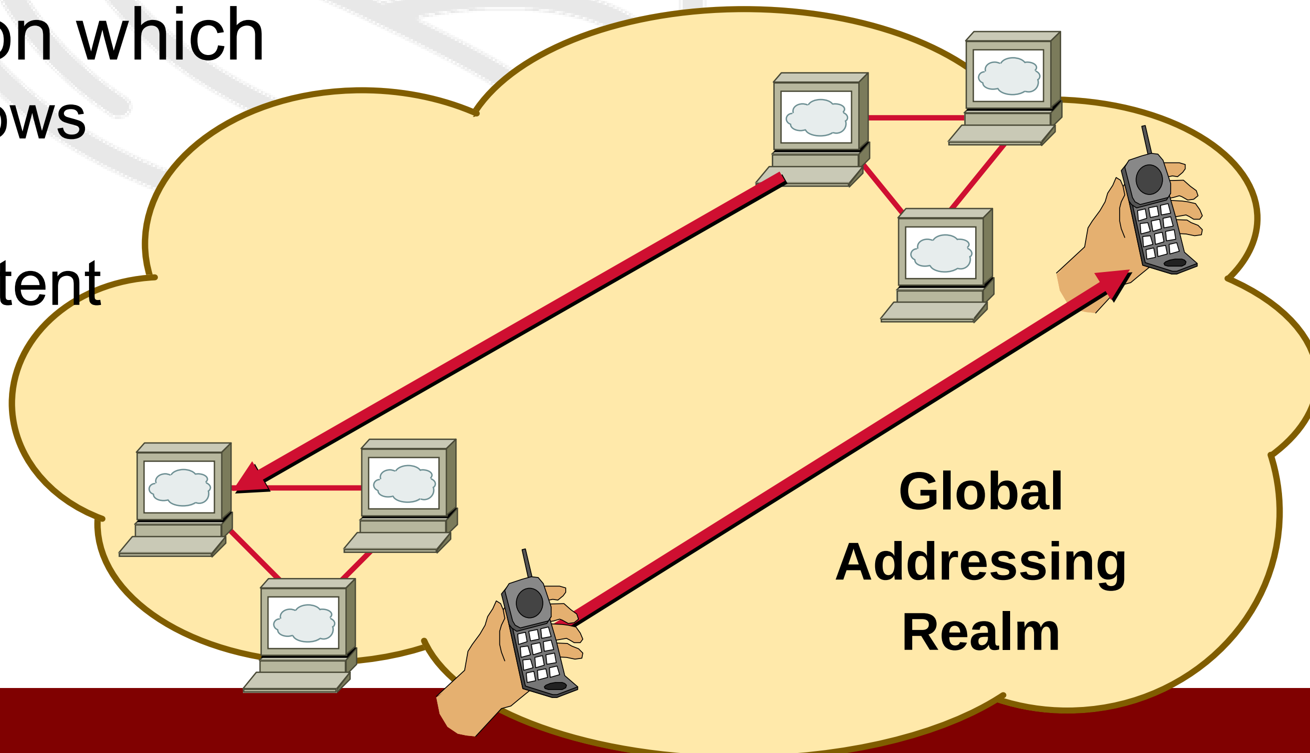
Introduction and Applications

Overview of Applications Environment

- peer-to-peer
- current Internet environment
- games
- collaboration / real-time interpersonal communication
- personal file server
- summary

Peer-to-Peer

- Virtually all nodes host a service
  - –The only required middle-box - dns / rendezvous service
- No restriction on which
  - –end initiates flows
- All participants
  - –share a consistent
  - –network view

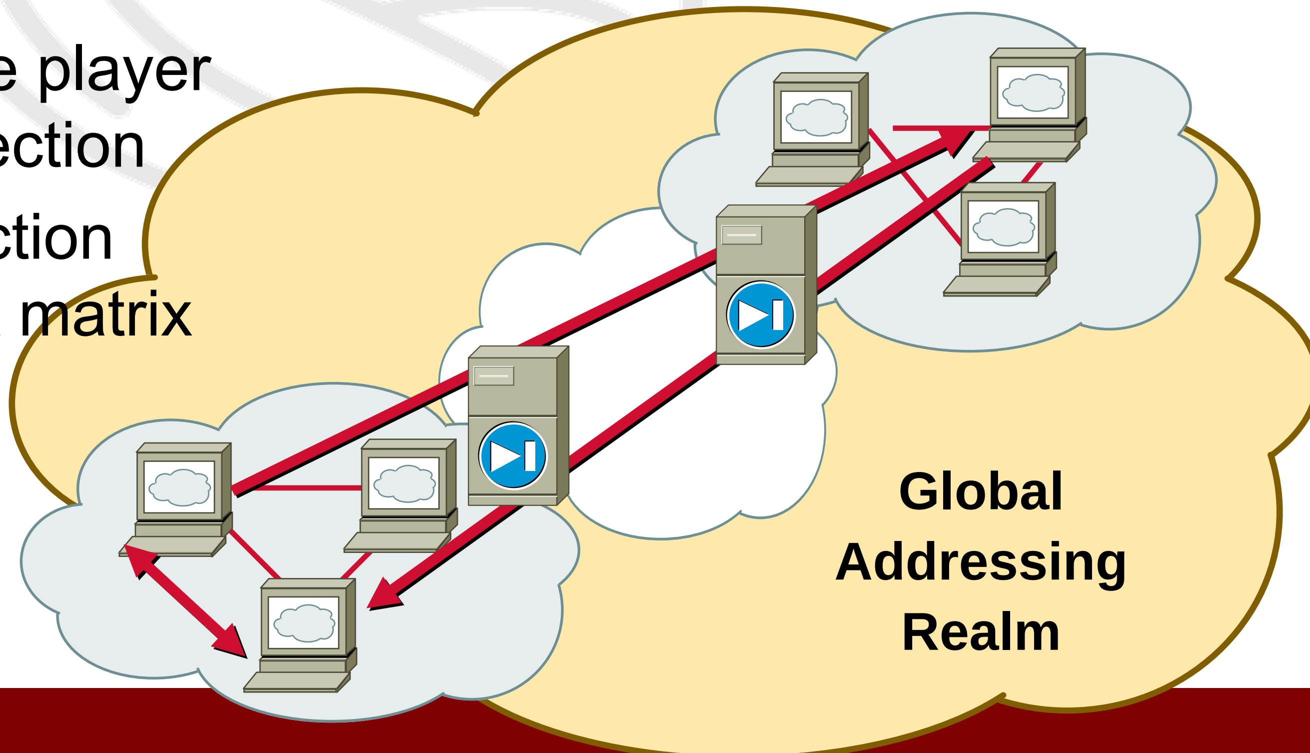


Current Internet environment

- IPv4 with NAT
  - deployment synchronization
  - scaling limitations
  - restricted topologies
  - single point of failure

Games

- Multi-player game requires consistent network view
  - More than one player per ISP connection
  - N-way connection establishment matrix

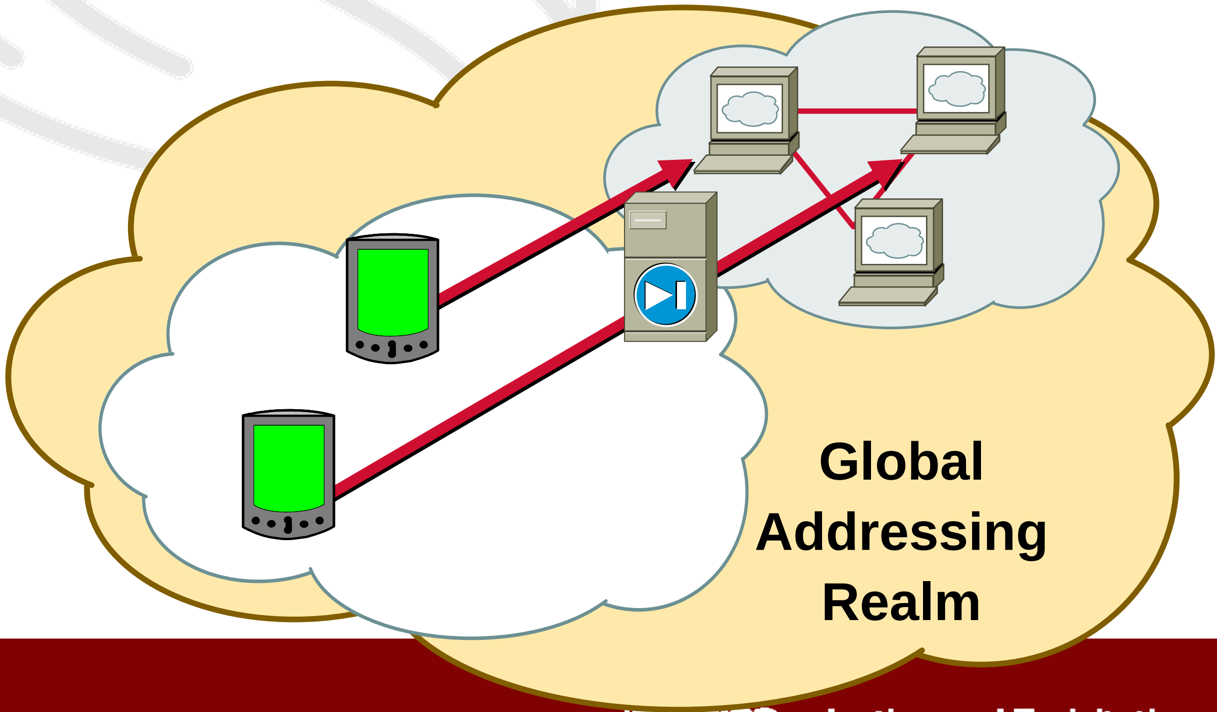


Collaboration / real-time interpersonal communication

- Asymmetric characteristics of NAT
  - im/voice : netmeeting
- Multiple streams - originating from opposite ends
  - mm training
- Always-on services
  - IP phone

Personal file server

- Always-on services
  - personal file servers



Rationale for IPv6 Applications

Rationale for Using IPv6

- Moving to IPv6 provides
  - Application development simplicity
  - Application deployment simplicity
  - Infrastructure diagnostic simplicity
  - Extensible networking simplicity

Application Development Simplicity

Application Deployment Simplicity

Infrastructure Diagnostic Simplicity

Extensible Networking Simplicity

The Stages of Application Development and Deployment

Stages of Deployment

- Legacy Applications ported to run over IPv6
  - Usable also where there is IPv6 infrastructure
- New Applications developed for use over IPv4, IPv6 or coupled IPv4/IPv6 infrastructure
  - Requires transition tools of course
- New Applications developed for use over IPv4, IPv6 or coupled; uses potential of IPv6, runs over IPv4
  - Requires transition tools of course
- New Applications developed specifically for IPv6 networks, no backward compatibility needed

Current Stage of Development

- Currently only in first two stages
  - Must ensure all needs applications can run over IPv6
- New Applications must run in IPv4, IPv6 and normally coupled
  - Cannot rely on IPv6 infrastructure being there
- Starting to consider how to use potential of IPv6 underlying services but must run on IPv4
- Need common view on the availability of underlying services before we reach fourth stage

Issues in Extending Applications to include the IPv6 Environment

General Considerations

- Most IPv4 Applications can be IPv6 enabled
  - If certain precautions are taken
  - Good Programming discipline is applied
- If there are IPv4 and IPv6 versions, most can be made dual stack
- Benefiting from IPv6 is much more difficult
  - Requires assumptions on underlying stacks and underlying network infrastructure
- Particularly satisfactory if written in a language that allows for IPv6
  - Java is good example

Effects on higher layers

- Changes TCP/UDP checksum “pseudo-header”
- Affects anything that reads/writes/stores/passes IP addresses (just about every higher protocol)
- Packet lifetime no longer limited by IP layer (it never was, anyway!)
- Bigger IP header must be taken into account when computing max payload sizes
- New DNS record type: AAAA

Ways of Dealing with IPv6 address change in the applications

Sockets API Changes

- Name to Address Translation Functions
- Address Conversion Functions
- Address Data Structures
- Wildcard Addresses
- Constant Additions
- Core Sockets Functions
- Socket Options
- New Macros

Functions of Changed Socket APIs

- Core APIs
  - Use IPv6 Family and Address Structures
  - socket() Uses PF_INET6
- Functions that pass addresses
  - bind()
  - connect()
  - sendmsg()
  - sendto()
- Functions that return addresses
  - accept()
  - recvfrom()
  - recvmsg()
  - getpeername()
  - getsockname()

Name to Address Translation

- getaddrinfo()
  - Pass in nodename and/or servicename string
    - Can Be Address and/or Port
  - Optional Hints for Family, Type and Protocol
    - Flags – AI_PASSIVE, AI_CANNONNAME, AI_NUMERICHOST, AI_NUMERICSERV, AI_V4MAPPED, AI_ALL, AI_ADDRCONFIG
  - Pointer to Linked List of addrinfo structures Returned
    - Multiple Addresses to Choose From
- freeaddrinfo()

```
int getaddrinfo(
    IN const char FAR * nodename,
    IN const char FAR * servname,
    IN const struct addrinfo FAR * hints,
    OUT struct addrinfo FAR * FAR * res
    );
```

```
struct addrinfo {
    int ai_flags;
    int ai_family;
    int ai_socktype;
    int ai_protocol;
    size_t ai_addrlen;
    char *ai_canonname;
    struct sockaddr *ai_addr;
    struct addrinfo *ai_next;
    };
```

Address to Name Translation

- getnameinfo()
  - Pass in address (v4 or v6) and port
    - Size Indicated by salen
    - Also Size for Name and Service buffers (NI_MAXHOST, NI_MAXSERV)
  - Flags
    - NI_NOFQDN
    - NI_NUMERICHOST
    - NI_NAMEREQD
    - NI_NUMERICSERV
    - NI_DGRAM

```
int getnameinfo(
    IN const struct sockaddr FAR * sa,
    IN socklen_t salen,
    OUT char FAR * host,
    IN size_t hostlen,
    OUT char FAR * serv,
    IN size_t servlen,
    IN int flags
    );
```

Porting Considerations

Porting Environments

- Node Types
  - IPv4-only
  - IPv6-only
  - IPv6/IPv4
- Application Types
  - IPv6-unaware
  - IPv6-capable
  - IPv6-required
- IPv4 Mapped Addresses

Porting Issues

- Running on ANY System
  - Including IPv4-only
- Address Size Issues
- New IPv6 APIs for IPv4/IPv6
- Ordering of API Calls
- User Interface Issues
- Higher Layer Protocol Changes

Specific things to look for

- Storing IP address in 4 bytes of an array.
- Use of explicit dotted decimal format in UI.
- Obsolete / New:
  - AF_INET replaced by AF_INET6
  - SOCKADDR_IN replaced by SOCKADDR_STORAGE
  - IPPROTO_IP replaced by IPPROTO_IPV6
  - IP_MULTICAST_LOOP replaced by SIO_MULTIPOINT_LOOPBACK
  - gethostbyname replaced by getaddrinfo
  - gethostbyaddr replaced by getnameinfo

IPv6 literal addresses in URL’s

- **From RFC 2732**

Literal IPv6 Address Format in URL's Syntax To use a literal IPv6 address in a URL, the literal address should be enclosed in "[" and "]" characters. For example the following literal IPv6 addresses: FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

3ffe:2a00:100:7031::1

::192.9.5.5

2010:836B:4179::836B:4179

would be represented as in the following example URLs:

http://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:80/index.html

http://[3ffe:2a00:100:7031::1]

http://[::192.9.5.5]/ipng

http://[2010:836B:4179::836B:4179]

Other Issues

- Renumbering & Mobility routinely result in changing IP Addresses –
  - Use Names and Resolve, Don't Cache
- Multi-homed Servers
  - More Common with IPv6
  - Try All Addresses Returned
- Using New IPv6 Functionality

Porting Steps -Summary

- Use IPv4/IPv6 Protocol/Address Family
- Fix Address Structures
  - •in6_addr
  - •sockaddr_in6
  - •sockaddr_storage to allocate storage
- Fix Wildcard Address Use
  - •in6addr_any, IN6ADDR_ANY_INIT
  - •in6addr_loopback, IN6ADDR_LOOPBACK_INIT
- Use IPv6 Socket Options
  - •IPPROTO_IPV6, Options as Needed
- Use getaddrinfo()
  - •For Address Resolution

Heterogeneous Environments

Precautions for Dual Stack

- **Avoid any explicit use of IP addresses**
  - **Normally do Call by Name**
- **Ensure that calls to network utilities are concentrated in one subroutine**
- **Ensure that libraries and utilities used support both stacks**
- **Do not request utilities that would not exist in both stacks**
  - **E.g. IPsec, MIP, Neighbour Discovery may vary**

New Applications

- For new Apps, some can use high-level language
  - JAVA fully supports dual stack
- Examples of utilities that must so support
  - DNS, SSH, FTP, Web server, Resource Location
- Examples of libraries and applications that must so support
  - RTP library, NTP time protocol, Web browser, IPsec library

Legacy Applications

- If most parts are written in say Java, and small parts in say C, try to rewrite C part to be in Java or at least make sure that I/O is concentrated in certain regions
- Potentially re-arrange code so that it fits needs of earlier slide
- Adjust I/f to code to fit dual-stack specs
  - Or do all networking via a utility which is IPv6-enabled
  - VIC, RAT using RTP are good example

Heterogeneous IPv4/IPv6 Environments

- May require dual-stack client/server, accessible by both stacks
  - Often used, for example, with Web services and with SIP signalling
- May require transition gateway
  - As for example with IPv4 telephones accessing other IPv6 ones
- May be very difficult, as when encrypted IPv4 messages are passed into the IPv6 networks with packet header encrypted, or certificate cryptographically bound to IP4 address

Available Applications

Available IPv6 Enabled Applications

- Many such applications exist. An Up-to-date database exists on: http://apps.6diss.org
- Many have been tested under 6NET, Description given in http://Ipv6.niif.hu/ipv6_apps
- Most currently useful utilities exist, e.g.
  - SIP, WWW, RTP, SSH, MIP, IPsec, NTP
- 6NET Deliverables discuss their use
  - Particularly those of WP5

Example of Application DB

WebSphere 6net Application database - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://193.55.253.34/WP5Apps/do?command=WP5APPSMasterView&order=ready&ascdesc=desc

Home Bookmarks

Applications summary

These are the application being ported, tested or developed by 6NET. Our aim is to perform trials on the suitability and robustness of IPv6 applications with a view to wide-scale deployment. Click on the column headers to change sorting order.

name	category	class	summary	status	responsible	modified	passed test
TUR	Streaming Radio	A	Trondheim Underground Radio	Running. Publicly available. Multicast support planned by mid 2003.	UNINETT	2004-03-11	✔
VideoLAN	Streaming	A	Streaming video server and player	Works. A multicast demonstrator. A first implementation of RTSP is available for better stream control.	SURFnet	2004-02-27	✔
Quake	Gaming	B	Multiplayer FPS action game	Works.	GARR	2004-02-27	✔
Kphone	Conferencing	A	SIP based Voice-over-IPv6 telephony application.	Demo version released	FhG Fokus	2004-03-11	✔
WMA through ftunnel	Streaming	A	Streaming of Windows Media using ftunnel	working	SURFnet bv	2004-03-11	✔
SER	Conferencing Support	A	SIP server	Operational	FhG Fokus	2004-03-11	✔
VIC	Conferencing	A	Video Conferencing Tool	VIC is currently fairly stable, and provides good performance. Further work is required on use of direct video display and integration of more codecs.	UCL	2004-03-17	✔
MCast6	Streaming	A	Tool for multimedia streaming in a computer network	testing phase	PSNC	2004-05-13	✔

Three Case Histories

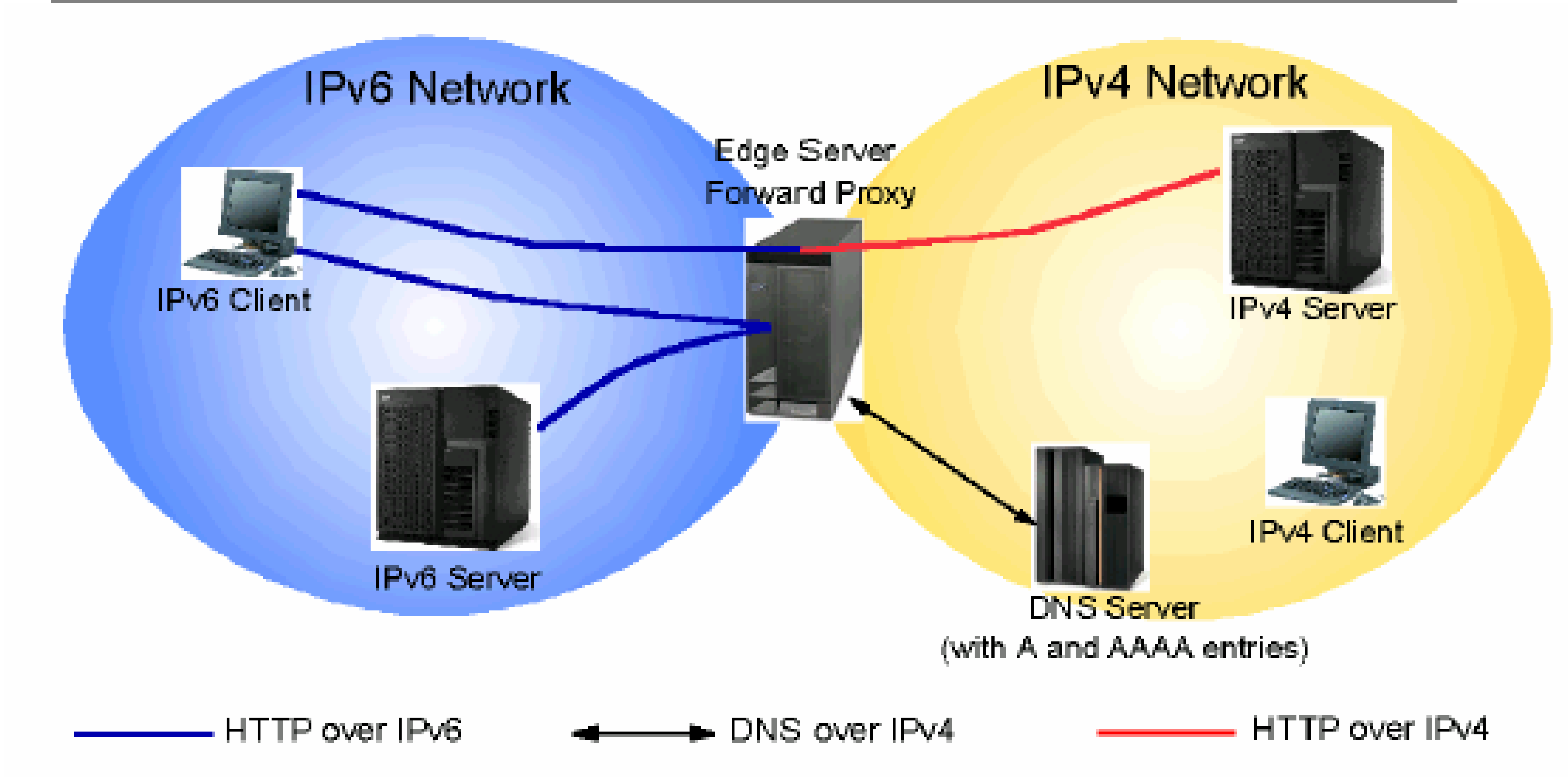
Three Case Histories

- Will consider three application services developed under 6NET to operate in a heterogeneous IPv4/IPv6 environment
  - Accessing servers
  - Voice over IP
  - Grid Computing through Globus

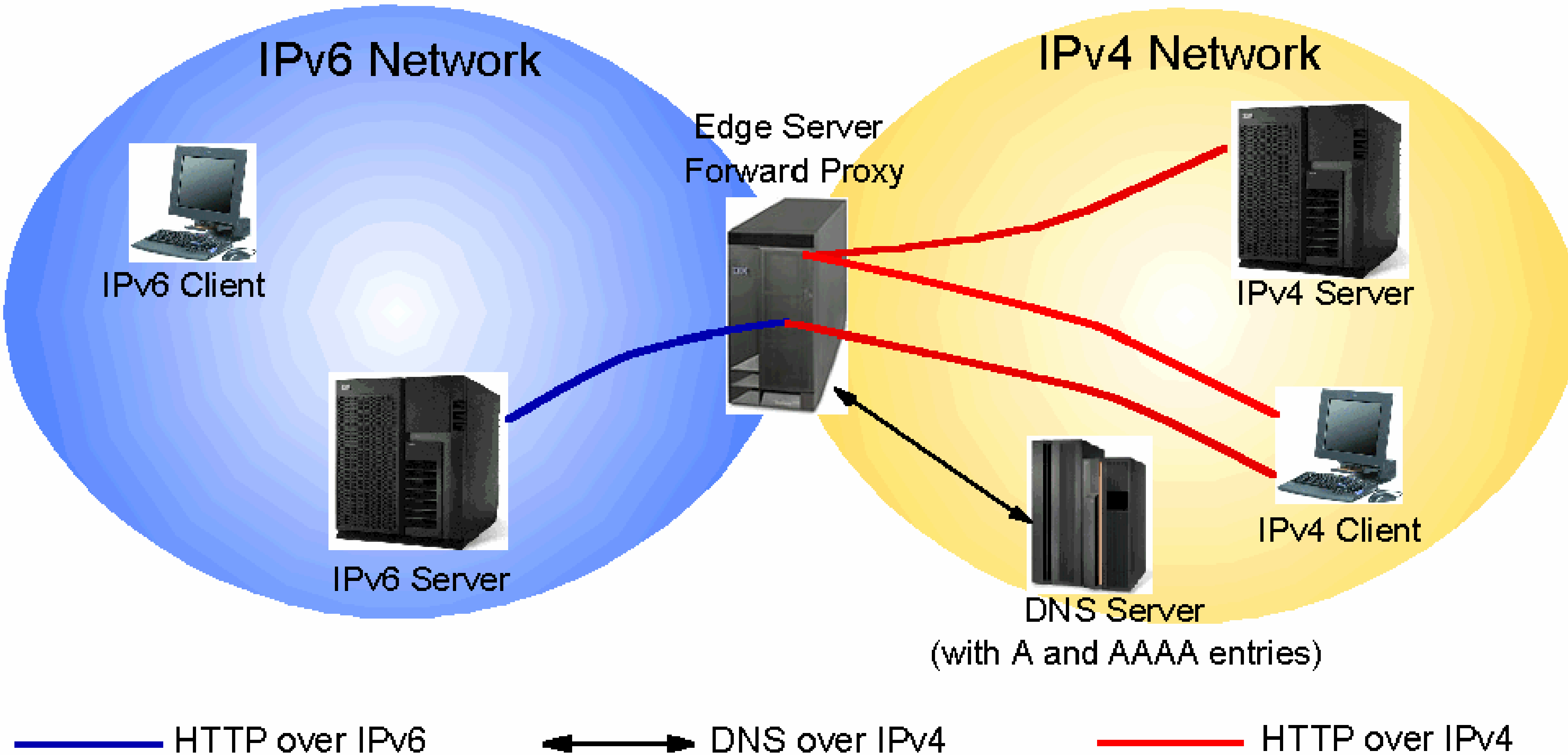
Accessing Servers

- Requires edge server proxies
- DNS in one of two regions
- Both sets of clients and servers

IPv6 Client Accessing IPv4 and IPv6 Servers



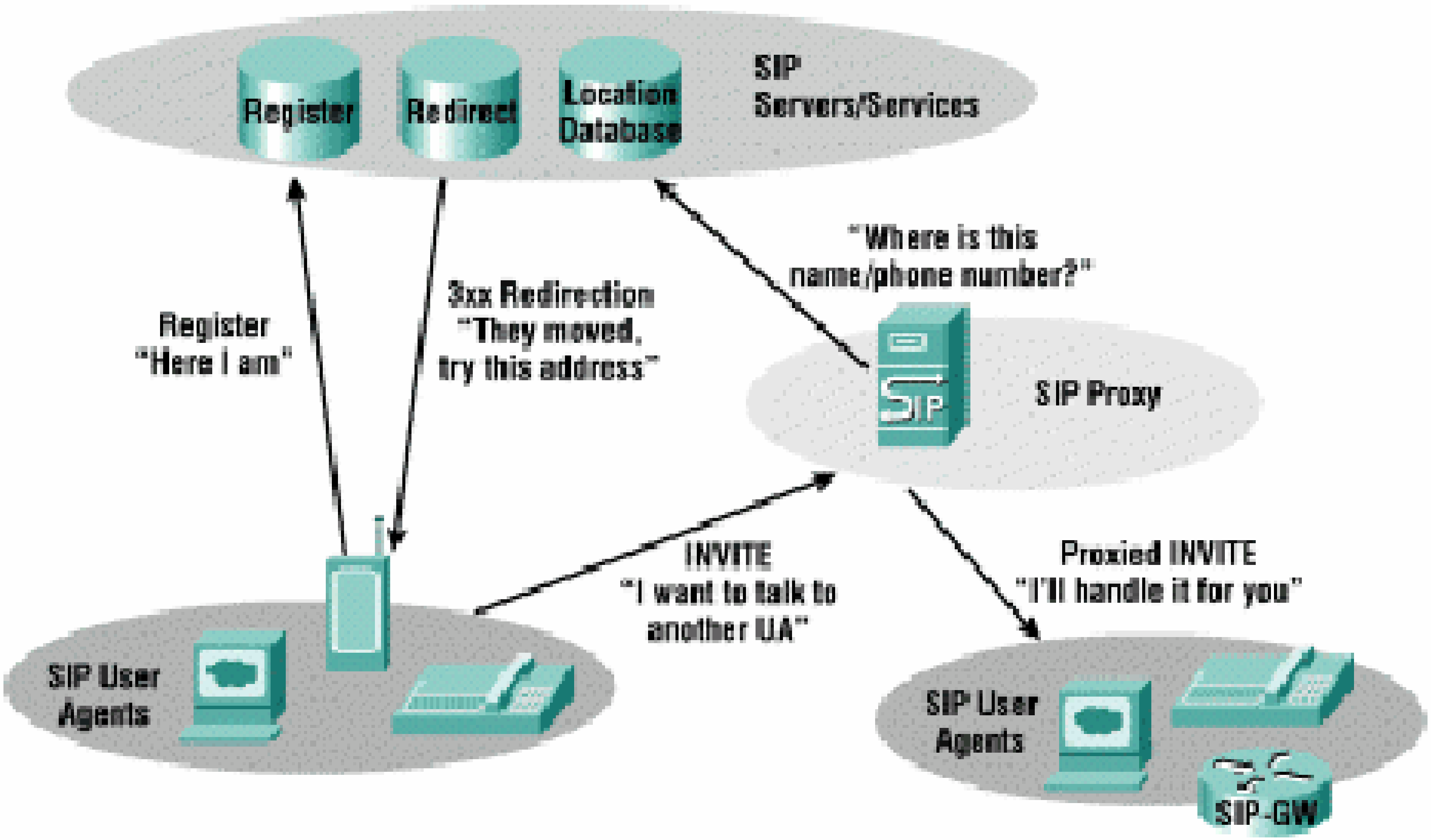
IPv4 Client Accessing IPv4 and IPv6 Servers



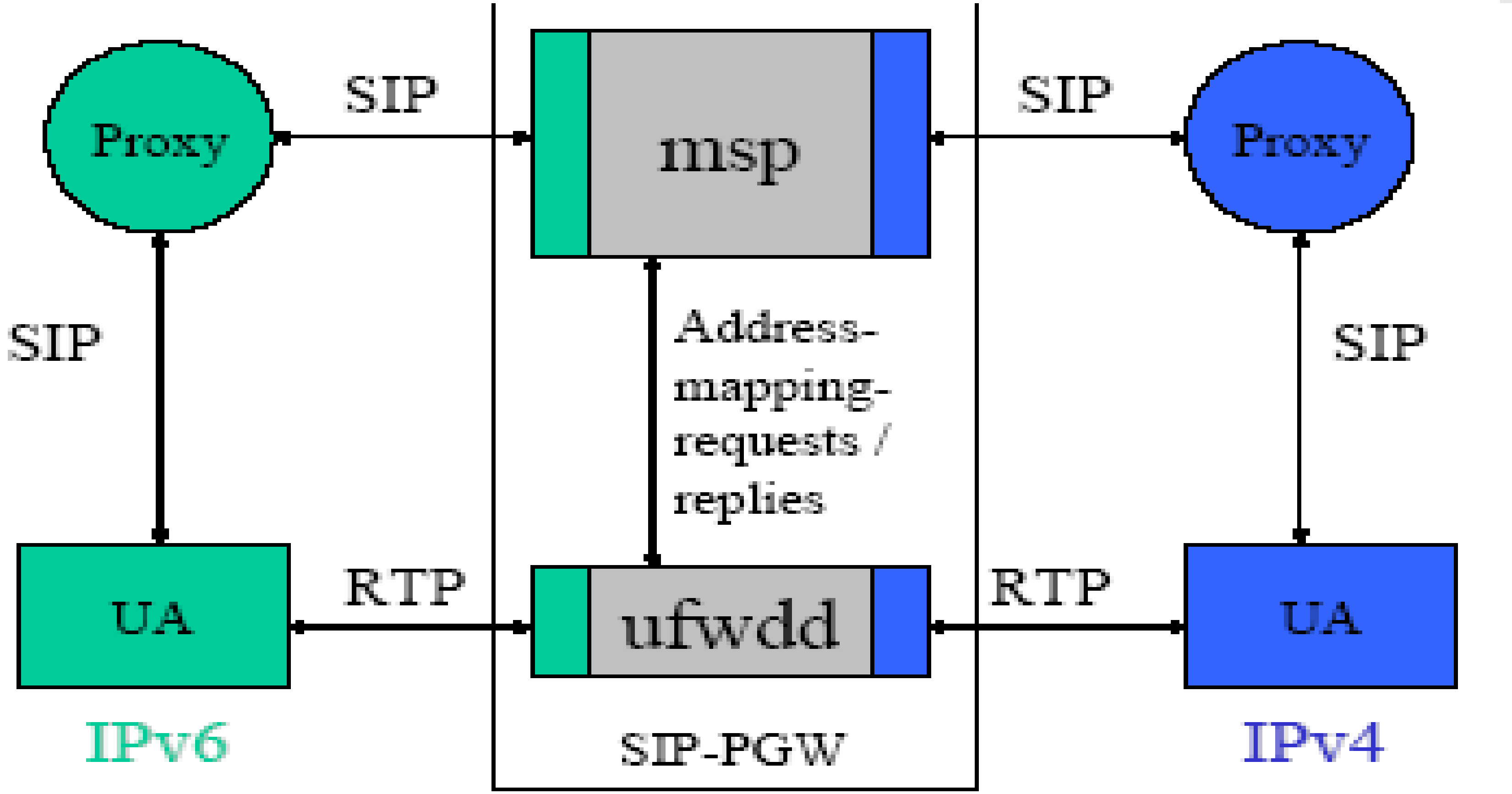
Voice Over IP App

- Requires Voice Servers to be dual stack
  - Use SER Servers
- Needs Session Initiation Protocol – SIP
  - SIP operation must be dual stack
  - SIP messages must be translated between IPv4 and IPv6
- Voice transport must pass through transition gateway
- Voice Multiplexing must be achieved

SIP Distributed Architecture



SIP Translation Gw



VoIP Integrated Scenario

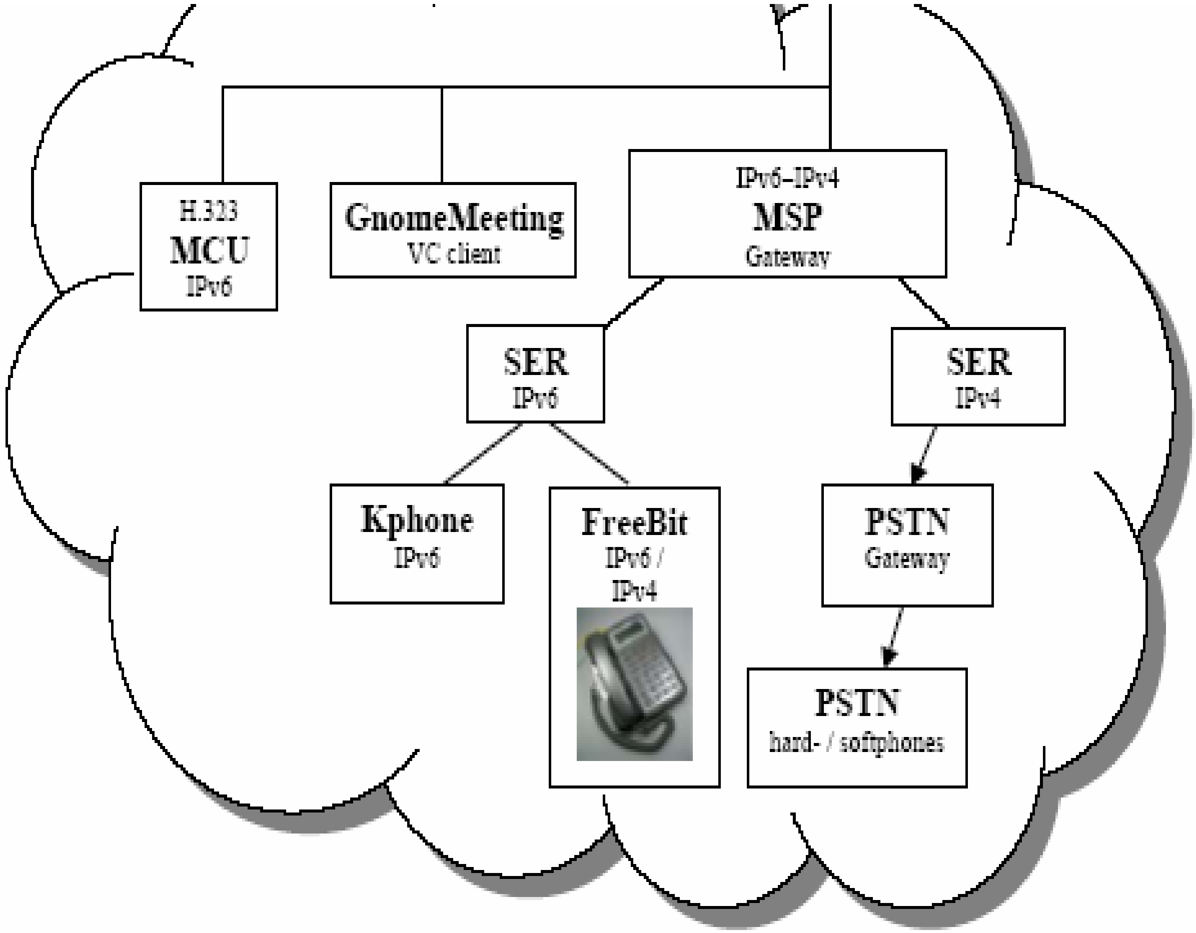


Figure 5 FhG VoIP system architecture

IPv6 and Grid

Outline

- Why Grids and IPv6?
- What is the Grid
- Benefits of IPv6 to Grid
- Work on IPv6 Grids
- Standardisation
- Future

Why Grids and IPv6?

- Grid computing represents a fundamental shift in how we approach distributed computing, like the fundamental shift in information access introduced by the Web
- IPv6 represents a major step function in the Internet's ability to scale, like the introduction of IPv4 twenty one years ago
- Inevitably there is synergy between these two game changers
- Let's share a common goal of reaching 10 billion Internet nodes

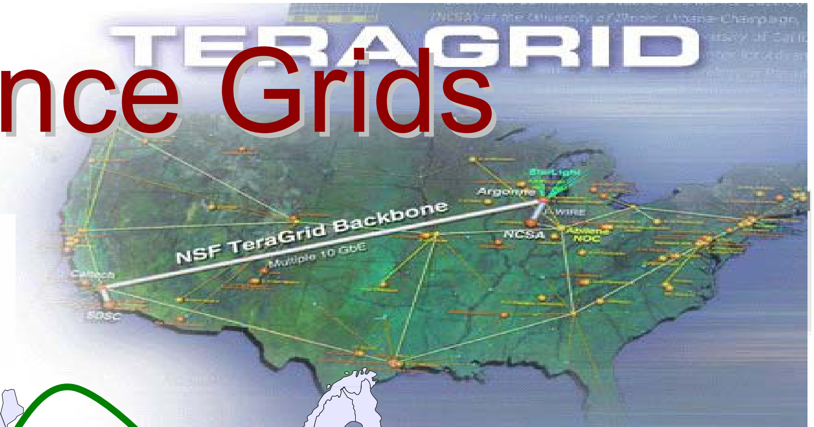
The Grid Is …

- A collaboration & resource sharing infrastructure with origins in the sciences
- A distributed service integration and management technology
- A disruptive technology that enables a virtualized, collaborative, distributed world
- An open source technology & community
- An analogy with the Power Grid
- A marketing slogan
- All of the above

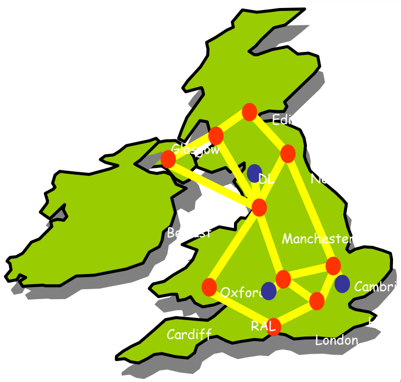
Not quite like the Power Grid!

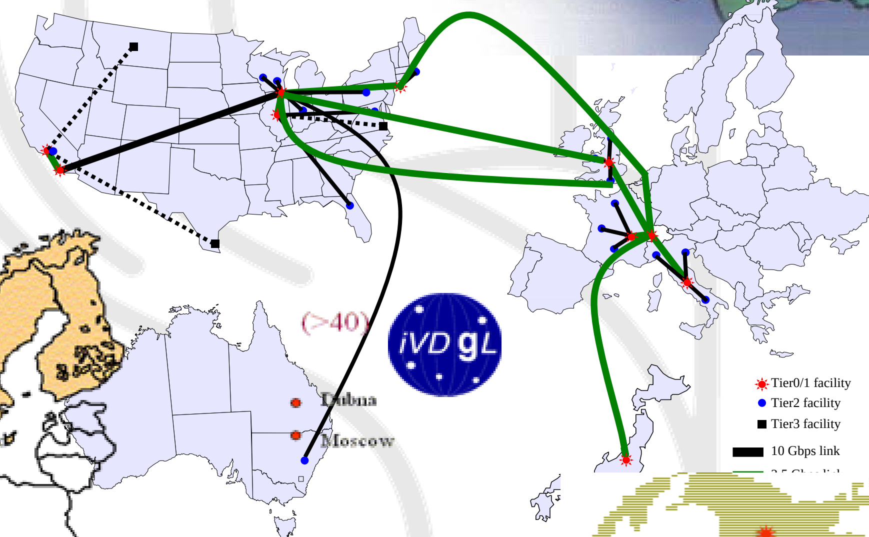
- I import electricity but must export data
- "Computing" is not interchangeable but highly heterogeneous
  - Computers, data, sensors, services, …
- But more significantly, the sum can be greater than the parts
  - Real opportunity: Construct new capabilities dynamically from distributed services
  - Virtualization & distributed service mgmt

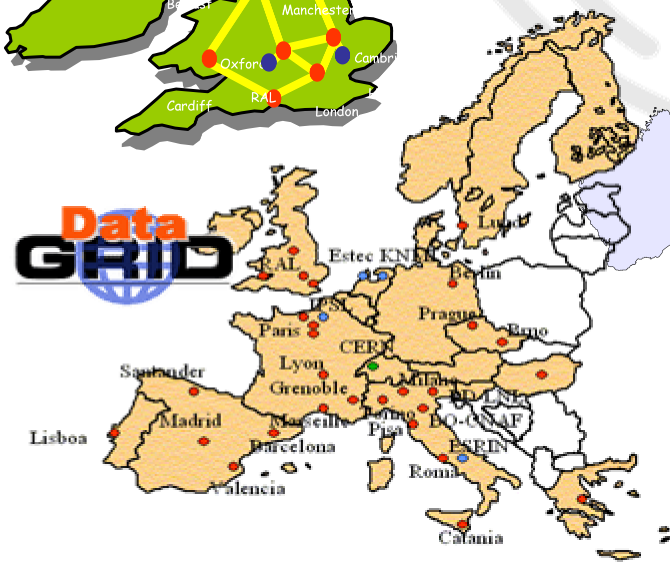
Example Science Grids

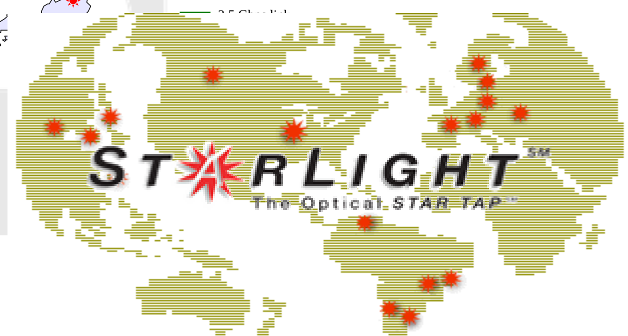


Building the National Virtual Collaboratory for Earthquake Engineering Research
NEESgrid







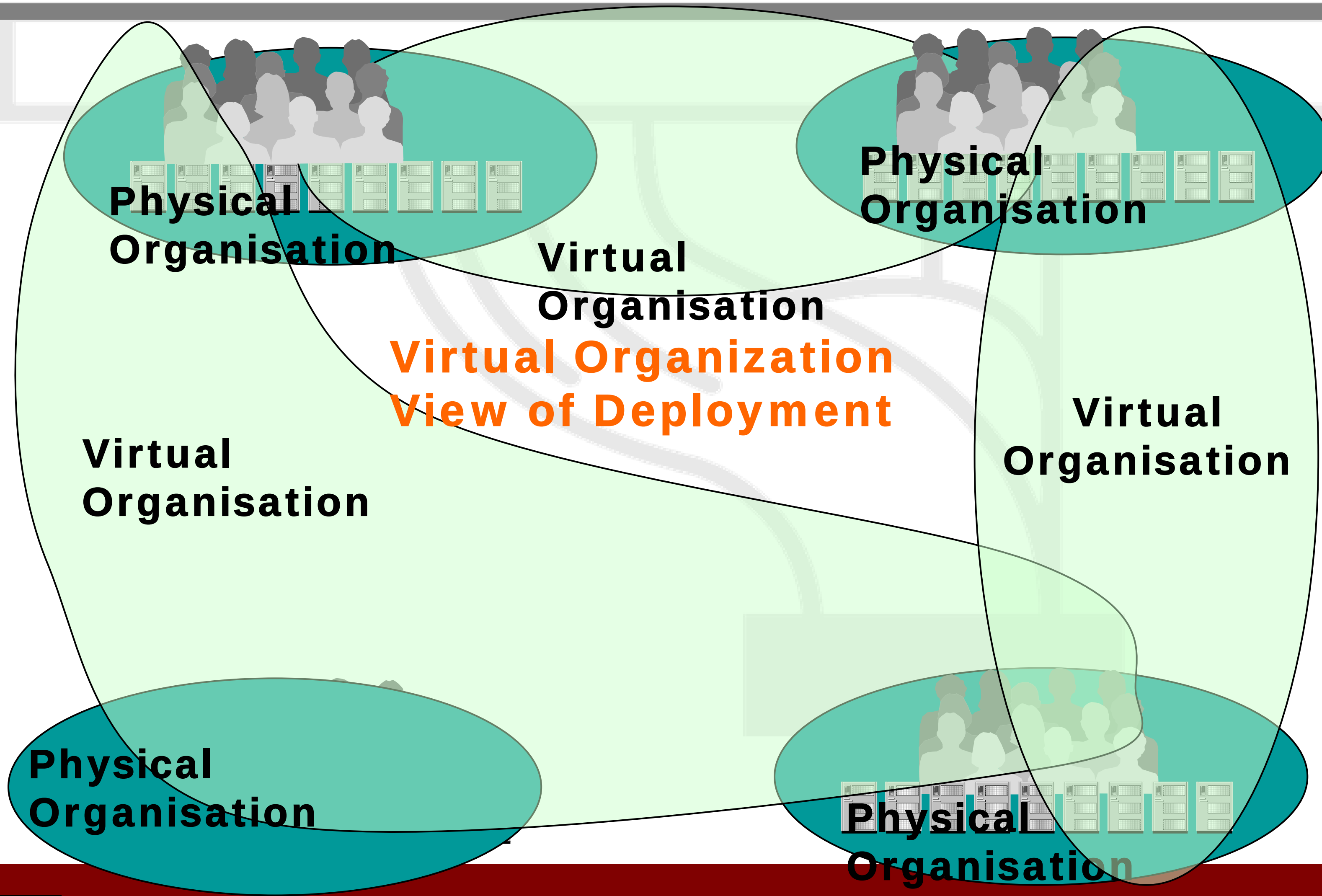


Abstract Computing Grids

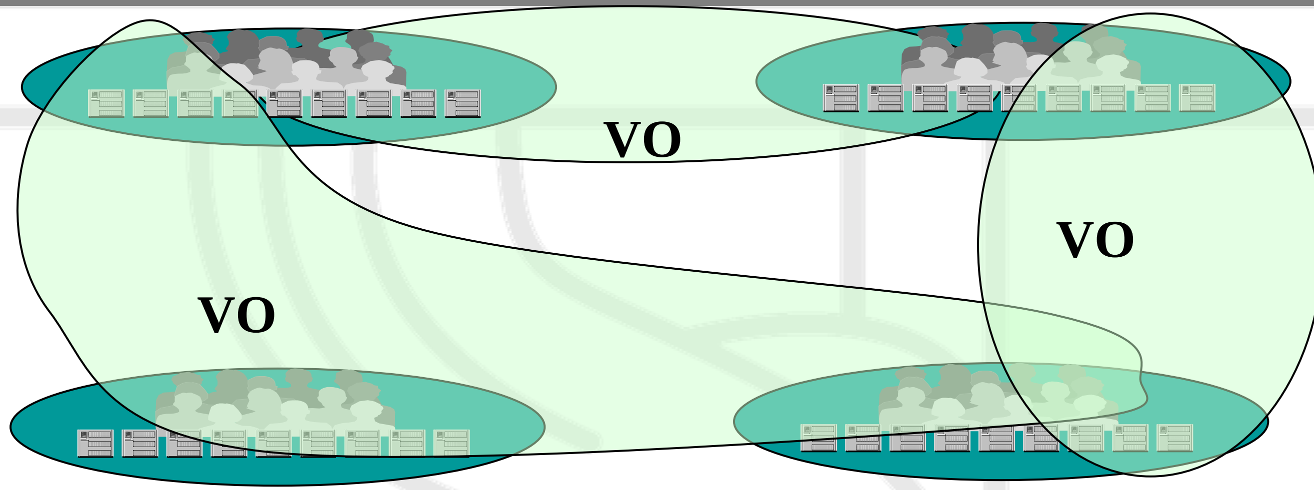
- Like public utilities
  - Shared
  - Reliable
  - Someone else runs it for you
- Computing Grid is a mechanism to

  "coordinate resource sharing and problem solving in or between physically dispersed virtual organisations (VOs)"
- Assigning resources, users and applications to VOs is fundamental to Grid

Virtual Organisations



Overlapping Virtual Organizations

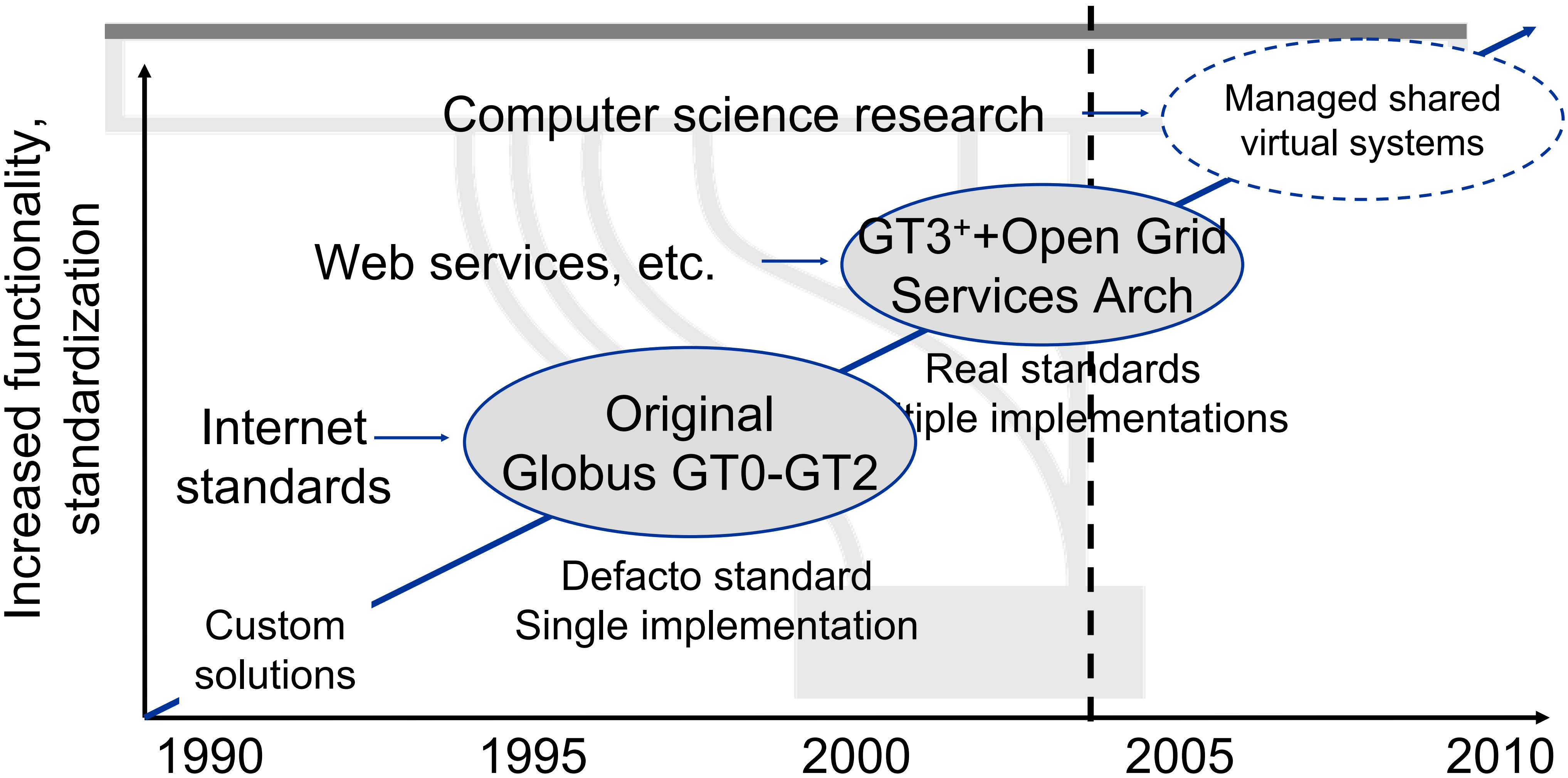


- Any system can be in any number of VOs with any number of other systems
  - Needs uniform address space to avoid proxies & allow end-to-end security (e.g. IPSec)
  - Addressing ambiguities unacceptable
  - Security boundaries ≠ organization boundaries
  - Not achievable at massive scale with IPv4

Virtual Organizations Look Like Dynamic Mergers & Acquisitions

- The effect of a Grid VO on networks is like a temporary partial merger of the organizations
- Merging two networks is painful today
  - “Private” IPv4 address space becomes ambiguous
  - Worst case: forced to renumber both networks
- Temporary partial mergers of an arbitrary number of IPv4 networks is unthinkable
- IPv4-based Grids are forced to rely on HTTP proxying between organisations: inefficient, and cannot exploit network-level security

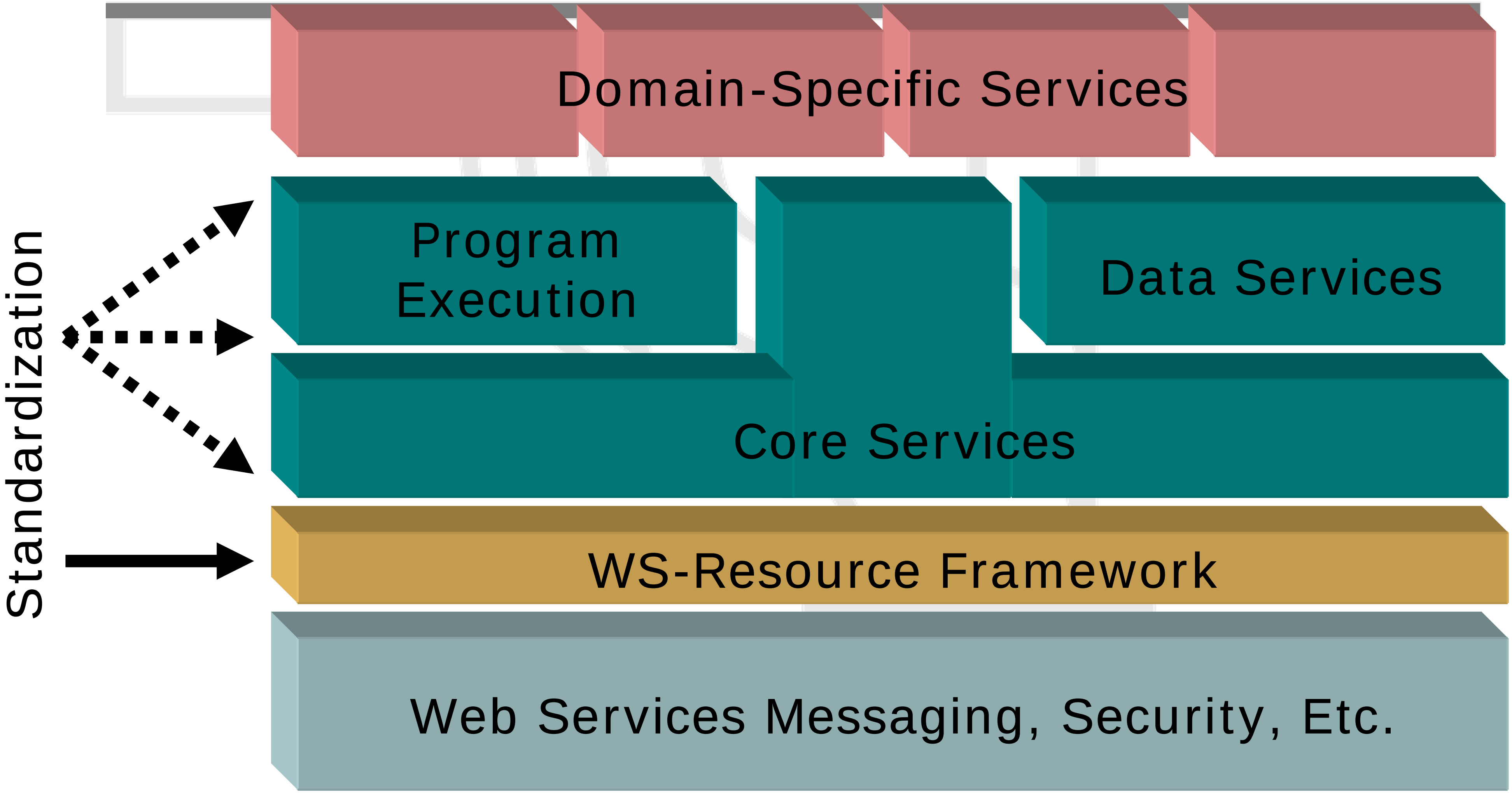
Emergence of Open Grid Standards



Open Grid Services Architecture

- Service-oriented architecture
  - Key to virtualization, discovery, composition etc
- Addresses vital “Grid” requirements
  - AKA utility, on-demand, system management, collaborative computing
- Web Services based framework
  - Distributed services based on XML/SOAP/WSDL
- Open Grid Services Infrastructure (OGSI)
  - Specifies ‘Grid Services’ mechanisms
  - New version WS-Resource Framework (WSRF)
- Standardised in Global Grid Forum (GGF) and Organization for the Advancement of Structured Information Standards (OASIS)

Open Grid Services Architecture



Benefits of IPv6 to Grid

- Bigger Address Space
  - Massive scaling potential >> 4 Billion(IPv4) nodes
- End-to-end addressing
  - Reduce need for NATs, Proxies etc
  - Enables full network level security (IPsec)
- Auto-configuration, renumbering
  - Simplifies network (re)configuration
- Complete Mobility Solution
- Modular design with clean extensibility
  - Streamlined processing, effective header compression etc
- Additional hooks for QoS – Flow Label

GGF IPv6-Working Group

- Setup & co-chaired by 6NET:IBM and UCL
- Global Grid Forum (IPv6-Working Group) http://forge.gridforum.org/projects/ipv6-wg/
  - IP version dependencies in GGF specifications
  - Guidelines for IP independence in GGF specifications
  - Status for Java Developers Kit API for IPv6

Current IPv6-WG documents

- Guidelines for IP independence in GGF specs
- Out of 88 documents surveyed 24 had some form of dependency
  - 60% failed to reference IPv6 URL RFC2732
    - **e.g. http://[2001:0DB8::CAFE]/sofia/**
  - 24% IP dependent textual material
  - The rest contained other dependencies
  - IP independence in specifications, Implementation
  - Implications for new features
- Status for Java Developers Kit API for IPv6
  - Add support for Flow Label and IPv4-mapped

Globus.org Toolkit

- Open source Grid Toolkit (GT)
  - From ANL, USC, UofC, EPCC, KTH
  - Corporate support IBM, MS, etc
- Java based Implementation of OGSI
  - Cross-platform interfaces & hosting
- Worked mainly with older GT3 release
- GT4 provides for new WSRF
- Ported GT3/GT4 to IPv6 under the auspices of the IST 6NET project

6NET, Grid and IPv6

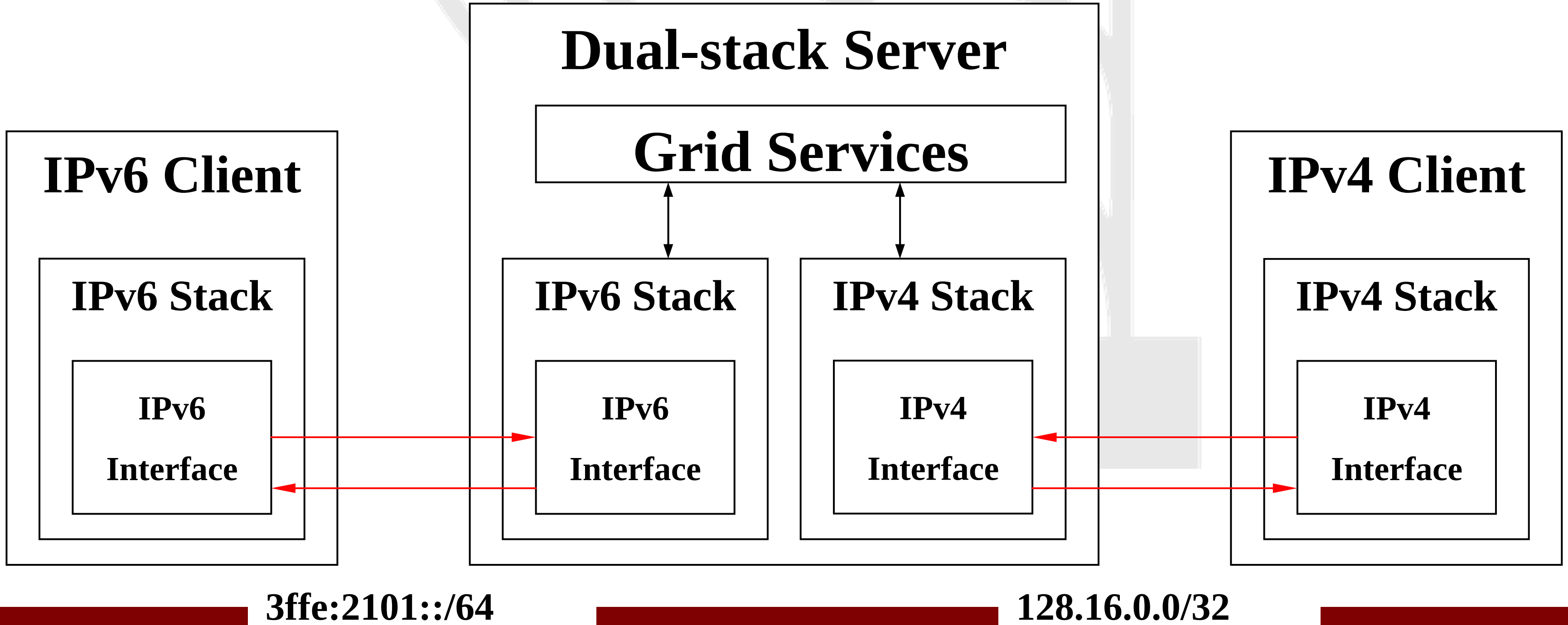
- Deploy IPv6 Grid services
  - Trials on 6NET test beds
- Transition considerations
  - IPv6 only
  - IPv6 and IPv4 coexistence
  - Devise appropriate policy and configuration
- Investigation of mobility and Grid
- Promote IPv6 compliance thru IPv6-WG

Globus Changes for IPv6 Support

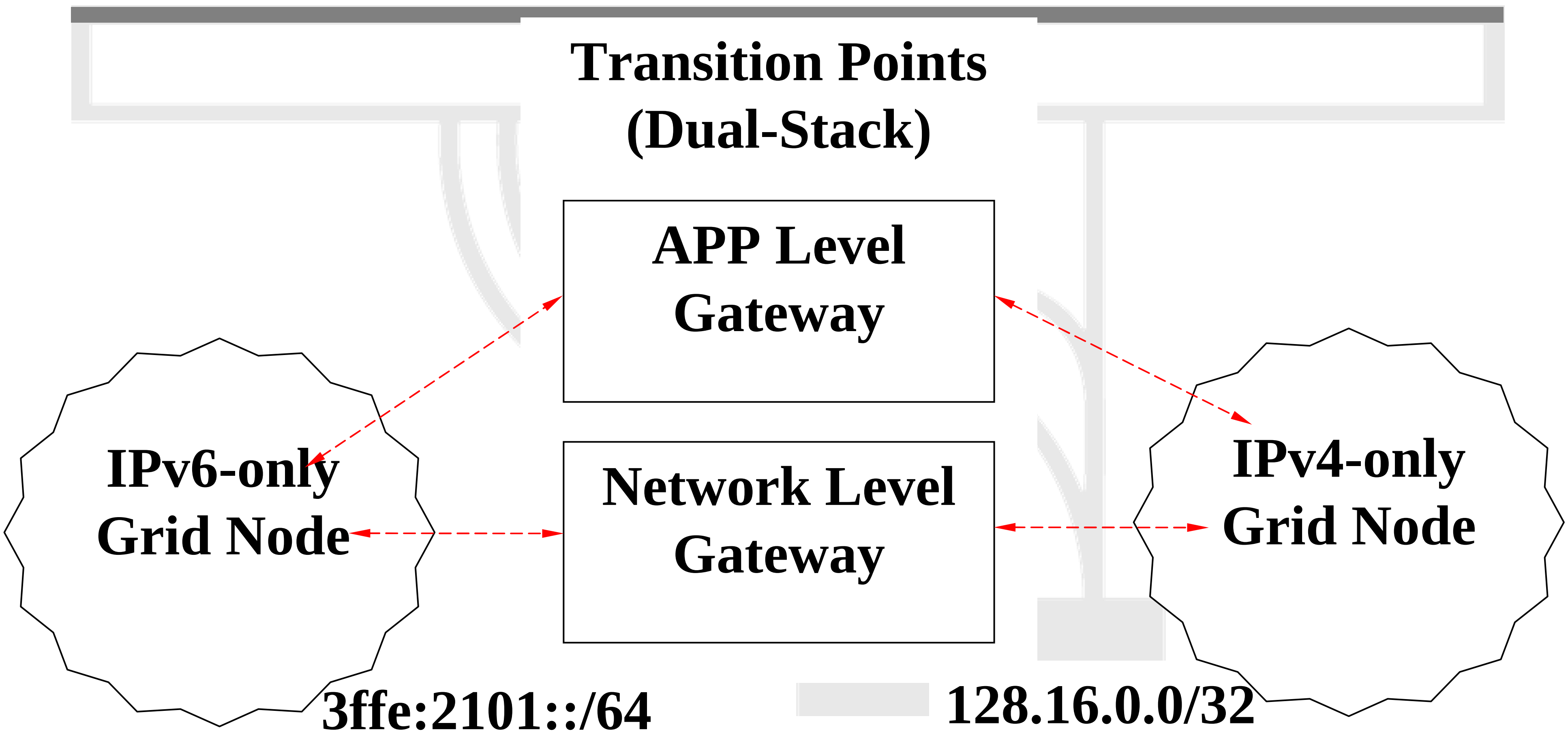
- A few protocols needed to be modified to suit IPv6 protocols
  - For example, Grid-FTP
- Correspondingly, the specific implementation needed modification
  - UCL has contributed to code changes in Globus core for IPv6
  - ANL developed XIO architecture for GridFTP with IPv6 capability
  - Still problems with security and resource location in mixed IPv4/IPv6 system

Transition between IPv4/IPv6

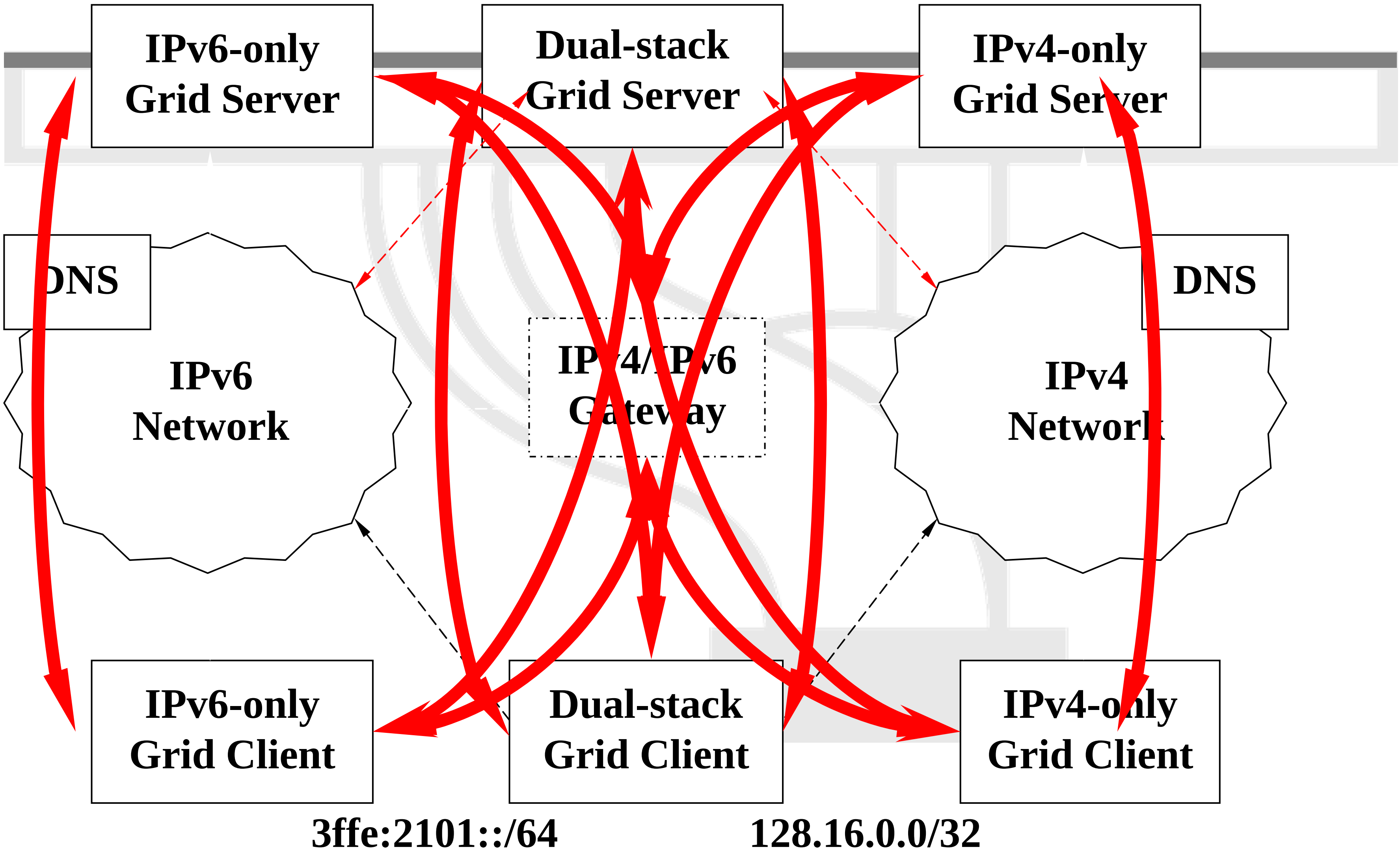
- A long transition period from IPv4 to IPv6 is expected
- Most Grid users are in IPv4 still
- Run Grid services on Dual-stack server
  - Be able to serve both IPv4 and IPv6 Grid clients



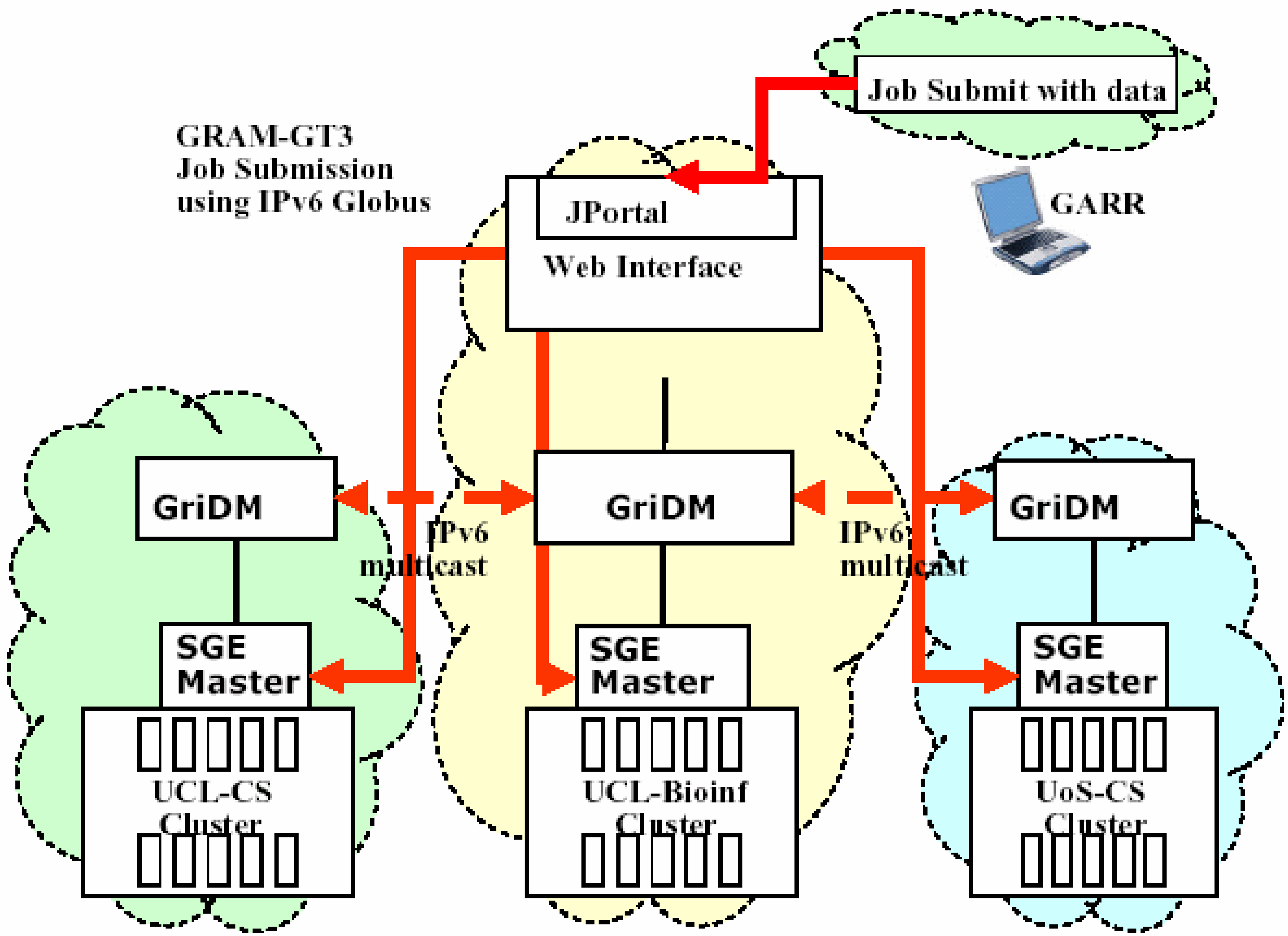
Transition Scenario



Globus in Mixed IPv6/v4 networks



UCL IPv6 Grid Test Scenario



Related work

- Other projects
  - EGEE : Large FW6 EU Grid project
  - SEINIT: FW6 EU security project
  - 6Grid : Japanese project working on IPv6 and Grid
  - Moonv6 : US IPv6 project
- Other Grid systems (such as Sun Grid Engine) are moving to IPv6

Some Links

- www.ggf.org
- forge.gridforum.org/projects/ipv6-wg
- www.globus.org
- www.6net.org
- www.cs.ucl.ac.uk/staff/s.jiang