IPv6 programming: The example of the BDII

- **Xavier Jeannin (CNRS UREC – EGEE-SA2),**

Reviewed by Mario Reale (GARR UREC – EGEE-SA2)

EGEE'07 conference – 2007-10-01, Budapest (HU)

www.eu-egee.org

Information Society and Media

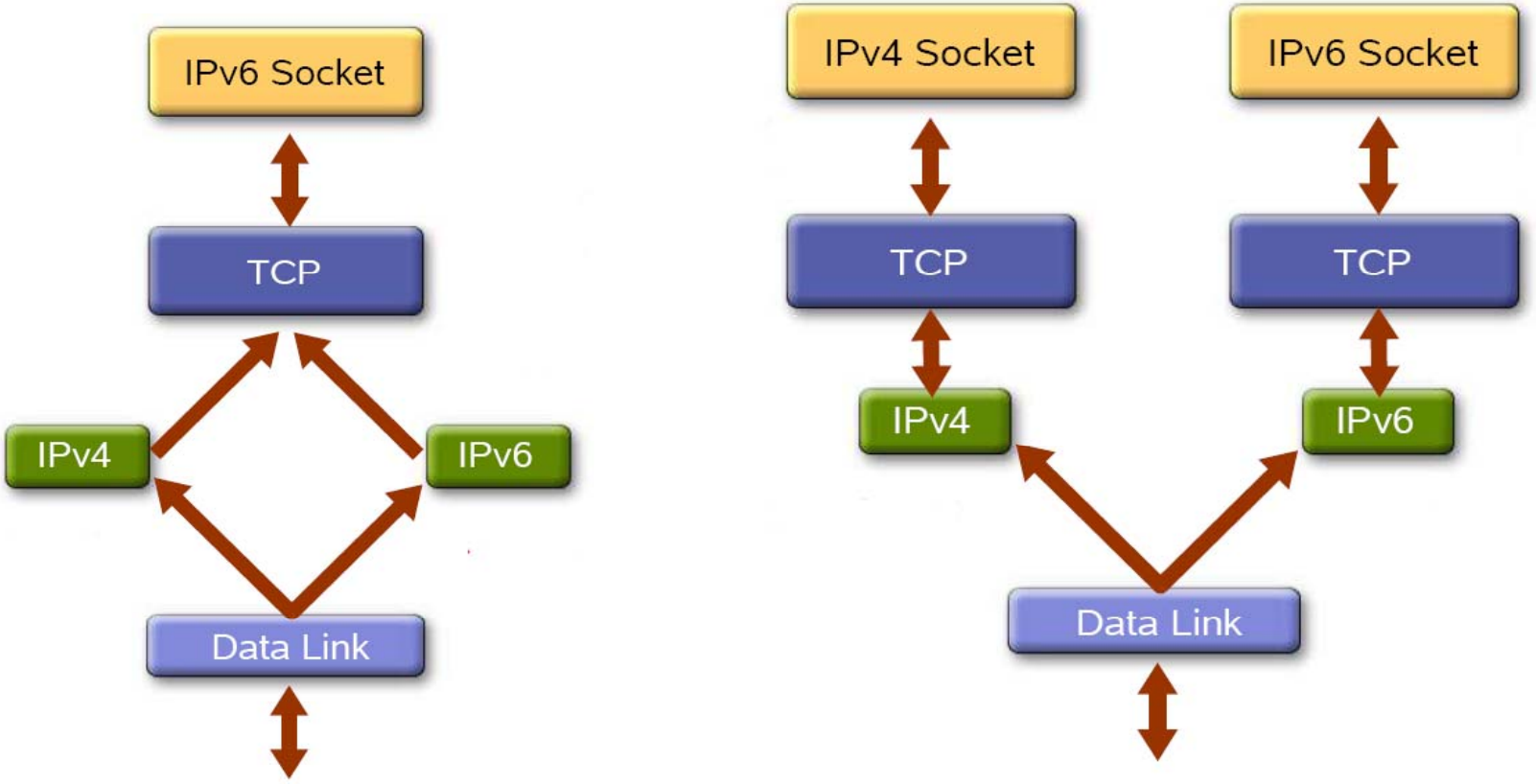
Outline

- **A brief review of different type of IP stack and of IPv4/IPv6 interoperability**
- **Test methodology and testbed**
  - Testbed architecture
  - Description of the test methodology
- **IPv6 programming: 2 different versions of dual stack server BDII**
  - A brief explanation of the different options to port an application on IPv6
- **Conclusion**

Dual stack

Both IPv4 and IPv6 stacks will be available during the transition period

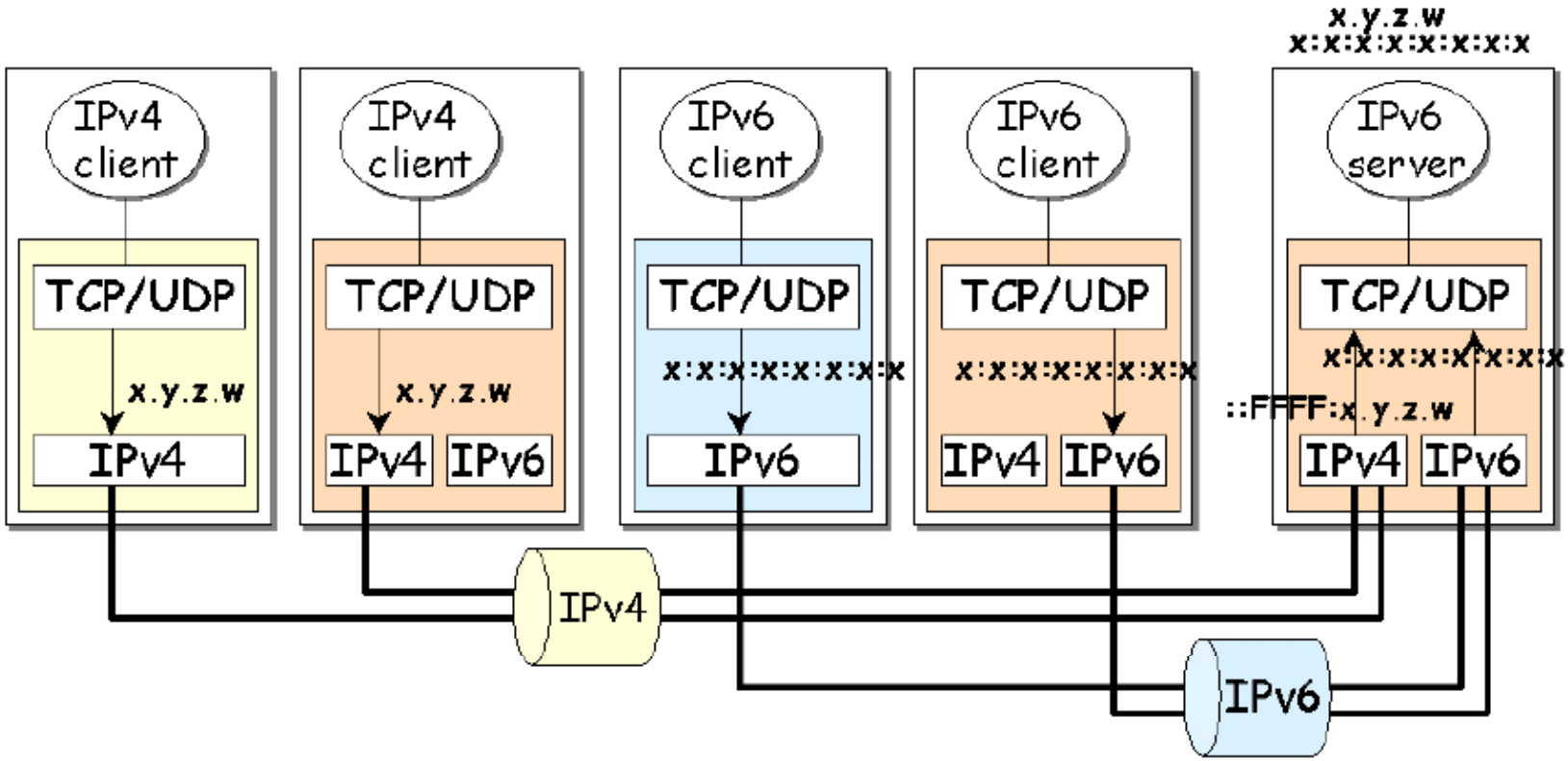
- Dual network stack machine will allow to provide a service both for IPv4 and IPv6
  - 2 different implementations of network stack



Source : Rino Nucara, GARR, EuChinaGRID IPv6 Tutorial

Mapping IPv4 address in IPv6

IPv6/IPv4 Clients connecting to an IPv6 server at dual stack node → 1 socket



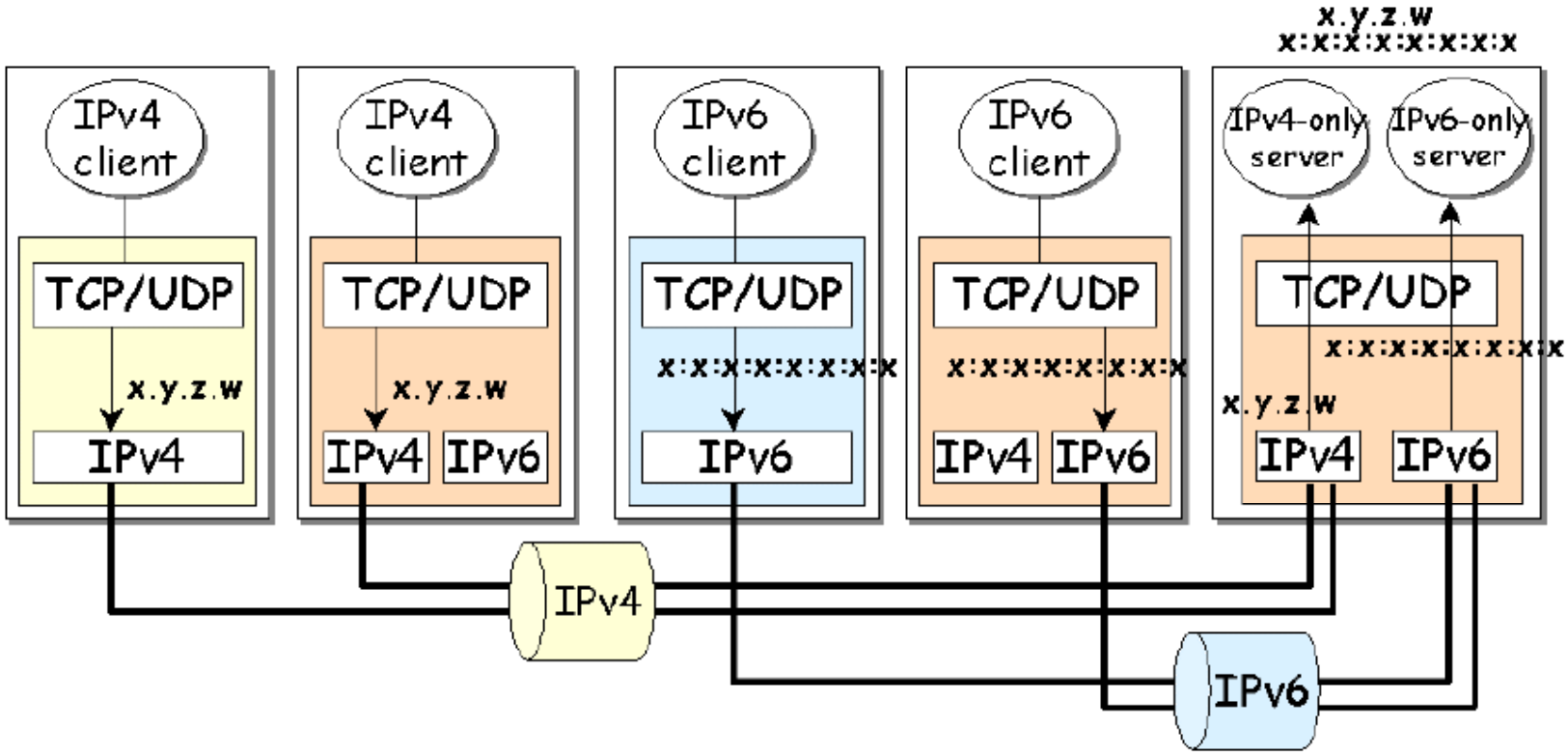
Dual Stack

Single IPv4 or IPv6 stacks

Source : Programming guidelines on transition to IPv6 T. P de Miguel, E. M. Castro

IPv4-only and IPv6-only

IPv6/IPv4 Clients connecting to an IPv4-only server and IPv6 only server at dual stack node → 2 sockets



Source : Programming guidelines on transition to IPv6 T. P de Miguel, E. M. Castro

Porting gLite on IPv6

The porting process on IPv6 of the entire gLite middleware will be long

- We will face a situation in which some gLite meta-packages (aka… gLite profiles, for instance, the gLite-CE or the gLite-BDII) will be already IPv6 compliant and some others won't
- Test a single component under IPv6 and its interplay with other components under IPv4.

Following a methodology to develop and test

- A methodology to develop for IPv6
- SA2 provides a methodology of test
  https://edms.cern.ch/document/810278/1

Porting gLite on IPv6

To develop and test an IPv6 code, developers need

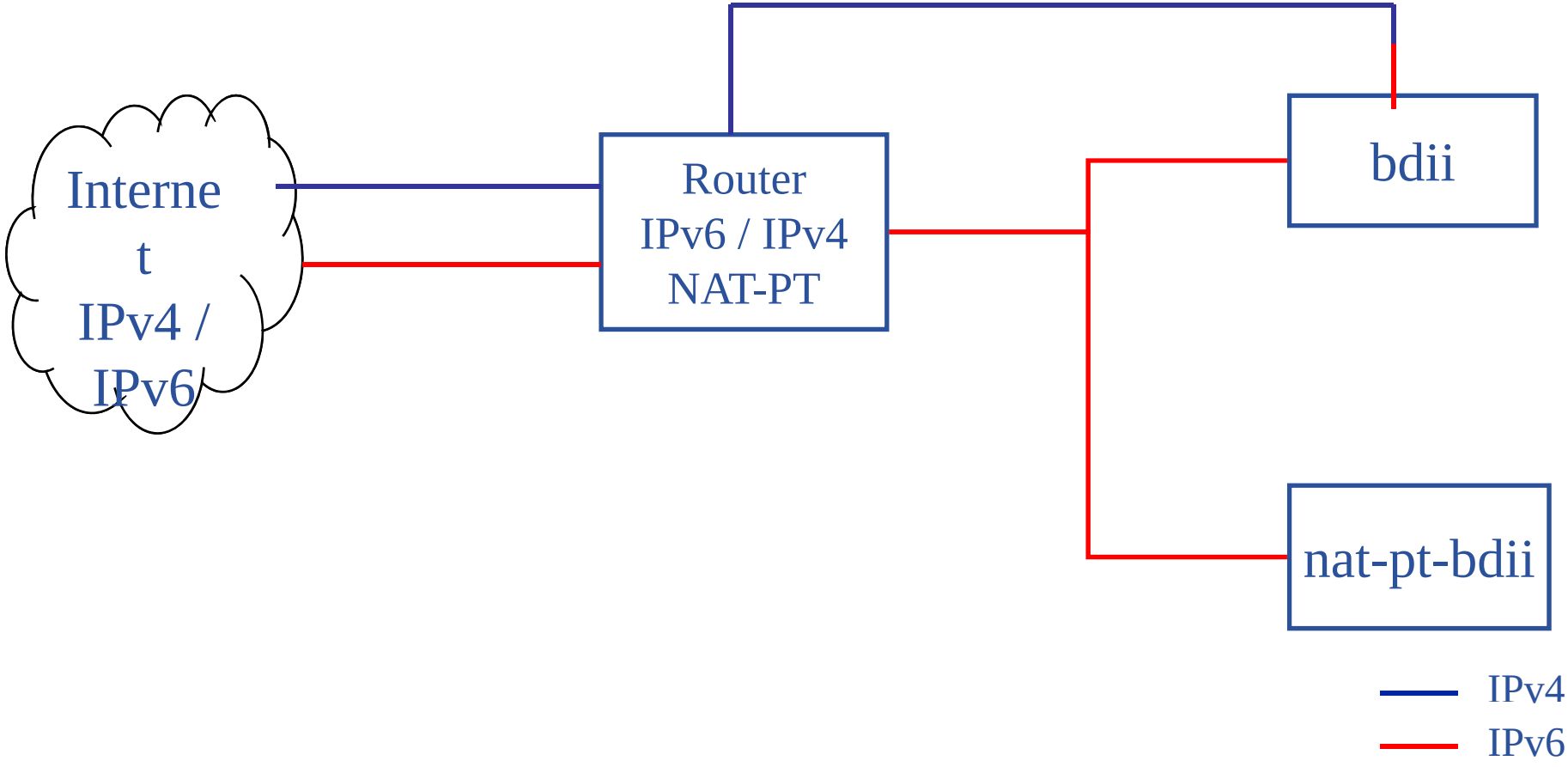
- a testbed (GARR and UREC testbed)
- a tool to install automatically on IPv6 testbed machine their code, test them, and retrieve the result to the developers → ETICS

Testbed must provide

- IPv6 support
- Translation mechanism to test interactions between a gLite meta-packages and the other part of the operational EGEE grid.
  - Using NAT-PT
  - Machines with IPv4 mapped address

Testbed

- IPv6 to IPv6 road
- IPv4 to IPv6 mapped through NAT-PT
- IPv4 to IPv4 road

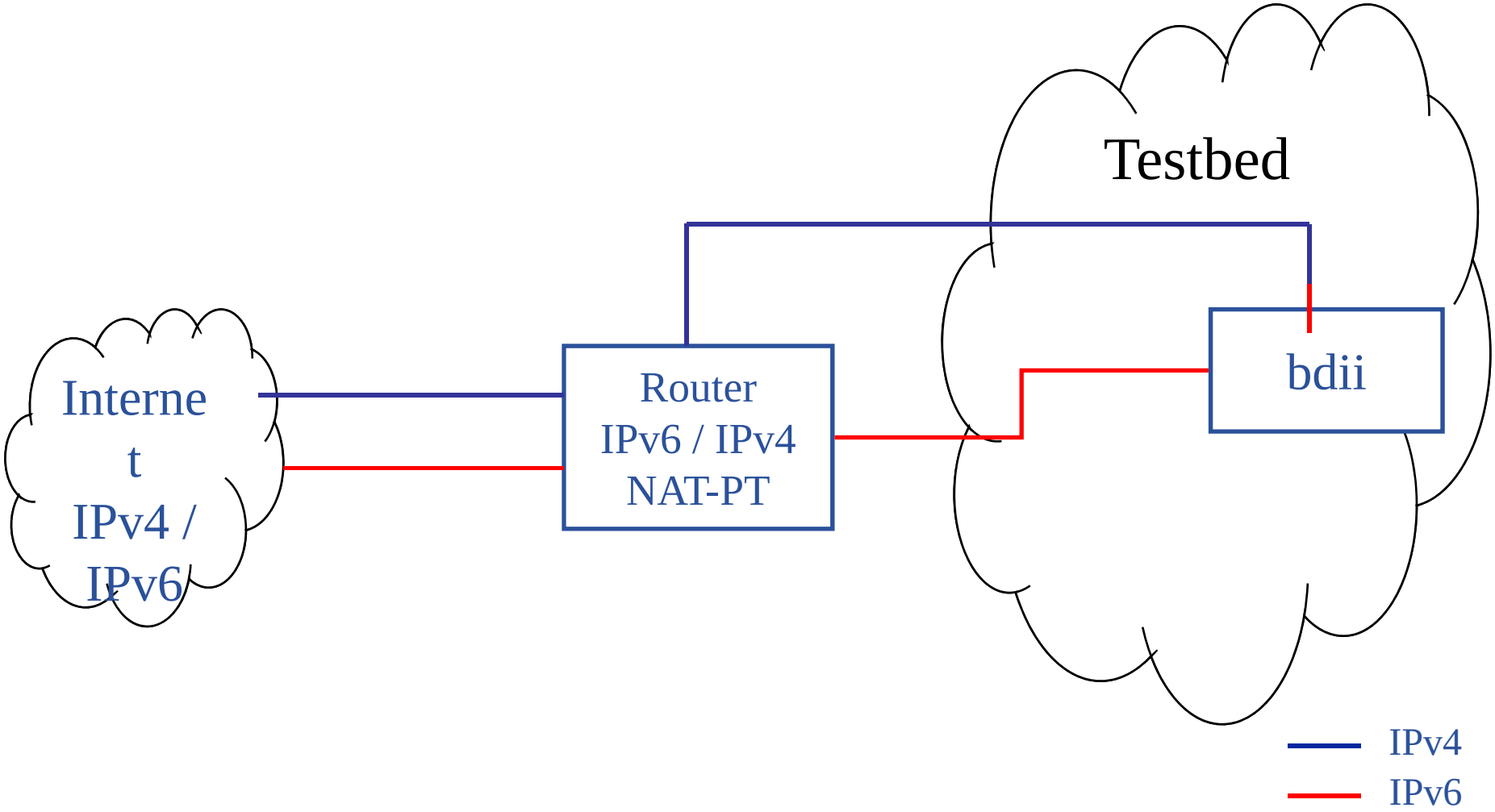


Test Methodology

- **A proposed methodology to test a component:**
  1. Look for IPv6 incompatibilities and modifications of the code;
  2. Installation on IPv4;
  3. Test on IPv4;
  4. Switch on IPv6 and test if the component is still running on IPv4;
  5. Switch off IPv4 and put the component behind the translation mechanism: NAT-PT or IPv4 mapped IPv6 address ;
  6. Test on IPv6 with IPv4 clients, via the translation mechanism;
  7. Test on IPv6 with the IPv6 client if available;
  8. Test the component in a dual stack environment (IPv4 and IPv6 clients without translation).

- **Each successful step grants an IPv6 compliance level.**

Test Methodology



2 different versions of dual stack server BDII

1. The program can open one socket (Sa) that listens on the IPv6 address. To be able to answer to IPv4 client requests, the OS redirect the request to the socket Sa, it is the default behaviour on SC3 (see bindv6only). To do that, the system maps the IPv4 address in the IPv6 address space: IPv4 mapped address

   u.x.y.z → ::FF:u.x.y.z

2. The program can open 2 sockets, one that listens on the IPv4 address and the other one that listens on the IPv6 address

Note: The usage of the IPv4 mapped address depends on the OS, and can be changed by the software

2 different versions of dual stack server BDII

We develop the two versions of the code for the BDII server:

- The first, which corresponds to the previous first case (one socket), is easier to develop → IPv6 server only
- The second one needs to use a lower level library and to modify more the opening section of the server → IPv4/IPv6 server

We provided the first version to BDII developers

Different options to port an application on IPv6

The main point to take in consideration in order to make choice:

- **First, you have to decide what is the OS target (SC4, SC3, Windows) to check what are the features supported by the OS.**
- **The different option provided by the programming language**
- **The different socket options used by the program. You may not find the same option in INET and INET6 socket families and therefore IPv4 mapped address cannot be used.**

There is not a clear advise in this matter even in RFCs

Different options to port an application on IPv6

- The gLite programmers have to take all these options in consideration.
- Considering the knowledge of gLite code, developer have to decide on
  - either a general rule if it is possible
  - or a set of rules that allow developers to make a choice depending on different cases.
- A developer guide would be helpful
  - For instance, If several applications have to be installed on the same node, it would be better to be consistent

Use cases that you can test

Use cases that you can test on UREC tesbed (paris.urec.cnrs.fr)

- **The machine Callisto hosts a BDII server that is patched to work only IPv6**
  - You can use Callisto with IPv6 client directly
  - You can use Callisto with IPv4 client through NAT-PT
- **The machine Quarks hosts a BDII server that is patched to work on dual stack machine**
  - You can use Quarks with IPv6 client directly
  - You can use Quarks with IPv4 client directly
  - You can test the two versions of BDII server on Quarks but one version is available at the same time
    - with one, IPv4 mapped address / IPv6 server
    - With two sockets IPv4/IPv6 server

Conclusions

- **We need to continue our collaboration with ETICS to automate submission of IPv6 tests for developers**
- **The IPv6 BDII work should be handed over to BDII developer team**
- **Developers should port a new gLite component as a next step and test**
  - the ETICS test submission procedure
  - the usage of the testbed
- **Developers should take up a methodology of development**
  - make their choice about using IPv4 mapped addresses
  - write a methodology and a guide for gLite developers
  - validate the test methodology suggested by SA2
- **EGEE should set up an official testbed for IPv6**

Questions?

Reference

References

- RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers
- RFC 3493 basic socket interface extensions for IPv6
- RFC 4038 Application aspects of IPv6 transition Programming guidelines on transition to IPv6, Miguel Castro
- IPv6 network programming jun-ichiro itojun hagino
- IPv4-Mapped Addresses on the Wire Considered Harmful draft-itojun-v6ops-v4mapped-harmful-02.txt jun-ichiro itojun hagino
- Guidelines for IP version independence in GGF specification T Chown
- IPv6 Guide for Windows Sockets Applications [Winsock] http://msdn2.microsoft.com/en-us/library/ms738649.aspx
- IPv6 Théorie et pratique Gisèle Cizault
- IPv6 Transition/Co-existence Security Considerations draft-ietf-v6ops-security-overview-06.txt P Savola
- Status for Java Developers Kit API for IPv6 IPV6_WG J. Bound
- Networking IPv6 User Guide for JDK/JRE 5.0 Sun
- UNIX Network Programming W. Richard Stevens
- Programming guidelines on transition to IPv6 T. P de Miguel, E. M. Castro