IPv6 Compliance Testing

How to test the IPv6 compliance of software?
Are gLite external dependencies IPv6-compliant?

Etienne Dublé - CNRS/UREC
etienne.duble@urec.cnrs.fr

September 23, 2008 – Istambul – EGEE'08 Conference

This document is available at https://edms.cern.ch/document/961337/1

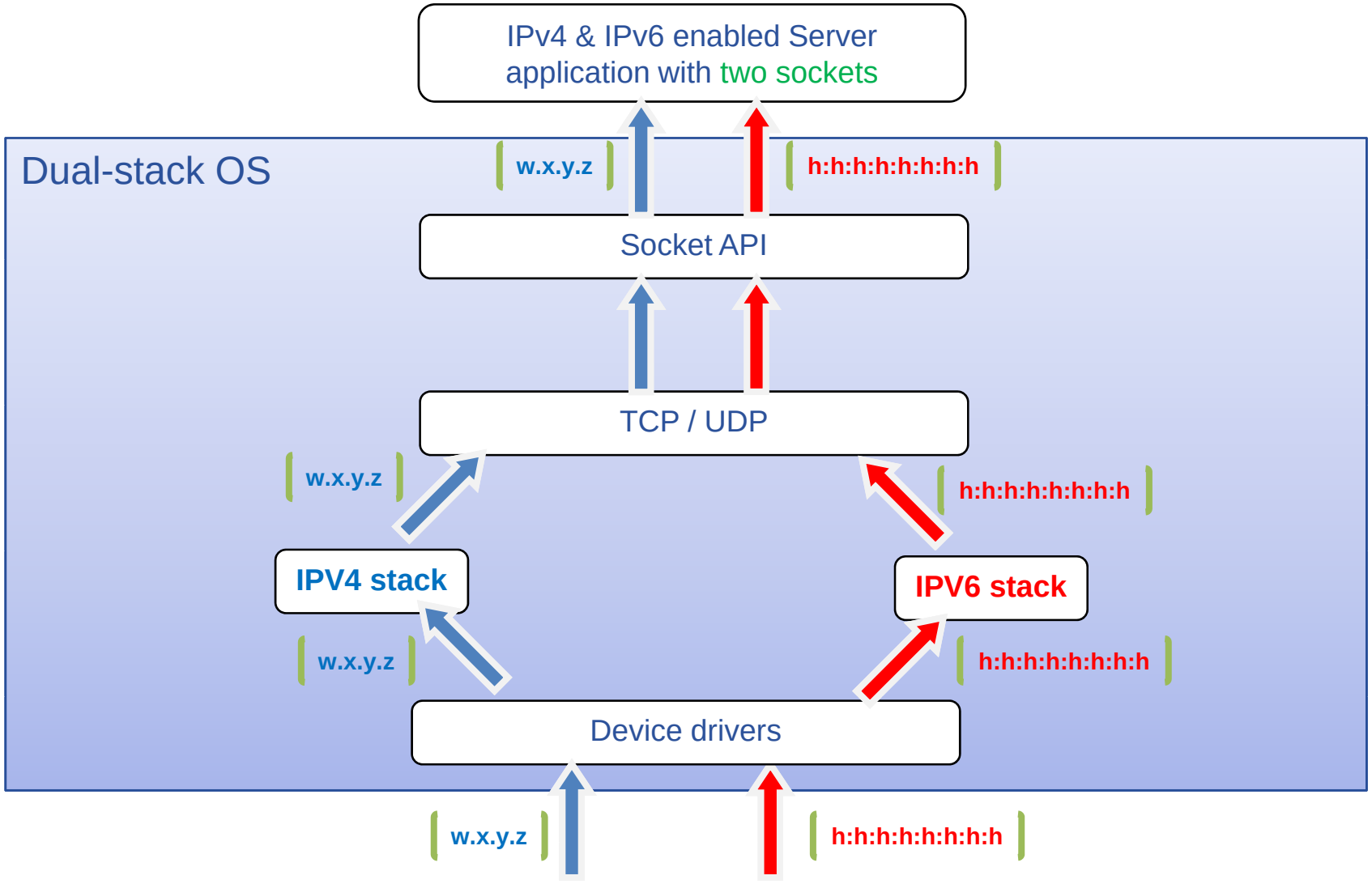
www.eu-egee.org

How to test the IPv6 compliance of software?

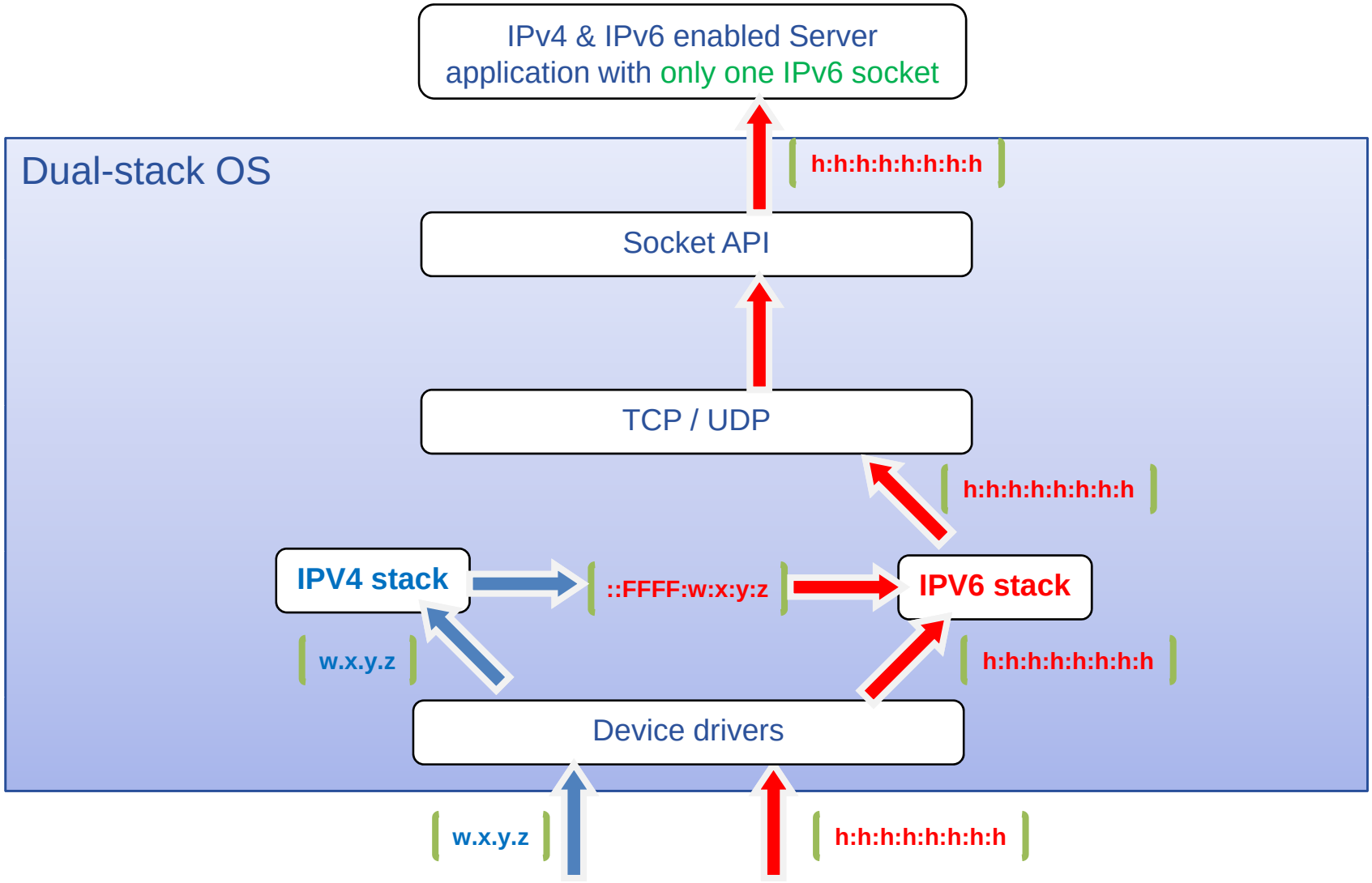
Our expectations

- **IPv4 and IPv6 will be coexisting for years**
- **In order to work in these mixed IPv4/IPv6 environment:**
  - The servers must accept both IPv4 and IPv6 connections
  - The clients must be able to connect to both IPv4 and IPv6 servers

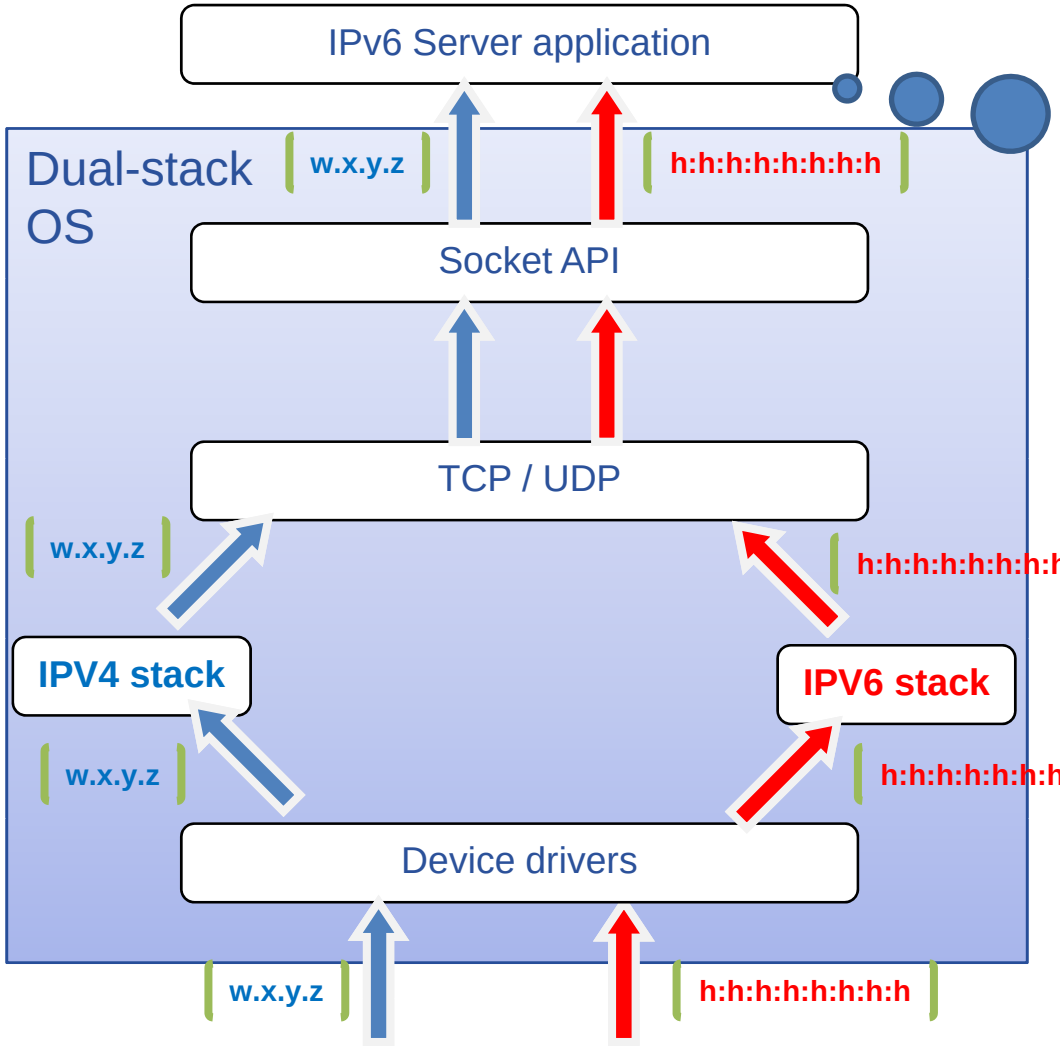
Servers with two sockets



Servers with only one IPv6 socket

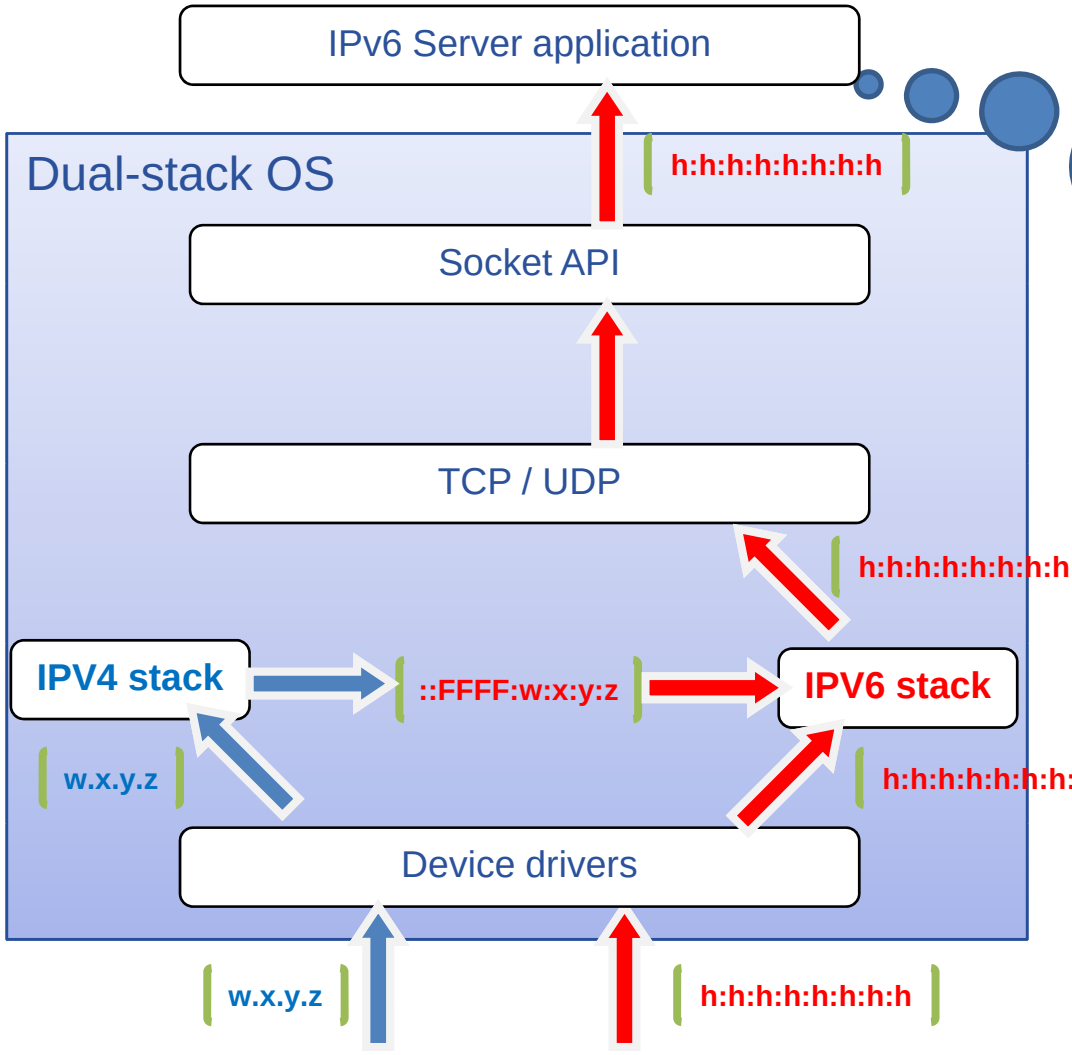


Servers with two sockets



The developer must **SET** the socket option **IPv6_V6ONLY** on the IPv6 socket!!

Servers with only one IPv6 socket



The developer must **UNSET** the socket option **IPv6_V6ONLY**!!

IPv4 connection to an IPv6 socket

- Depending on the kind of server, the IPv6 socket must or must not accept IPv4 connections.

		Value of IPV6_V6ONLY		
		Enabled with setsockopt	**Unspecified (no setsockopt use)**	**Disabled with setsockopt**
Value of /proc/sys/net/ipv6/ bindv6only	0	IPv4 clients **cannot** connect	IPv4 clients **can** connect	IPv4 clients **can** connect
	1	IPv4 clients **cannot** connect	IPv4 clients **cannot** connect	IPv4 clients **can** connect

- Trying to bind an IPv6 socket accepting IPv4 connections and an IPv4 socket on the same network port number will fail.

Programming language limits

- **With some programming languages** (for example: Java) **IPV6_V6ONLY cannot be set or unset. In this case the behavior of the program depends on the system parameter /proc/sys/net/ipv6/bindv6only**.
- For example, **a Java server program run on a dual-stack system will not allow IPv4 clients to connect if /proc/sys/net/ipv6/bindv6only is set to 1** on the operating system.

Listening socket(s) test

- **How to check which kind(s) of listening socket(s) a server opened:**

```
[root@quarks IPv6_test]$ netstat -lnpt | grep 20001
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address  Foreign Address  State  PID/Program name
tcp        0      0 :::20001       :::*            LISTEN 36853/server_one_so
[root@quarks IPv6_test]$ netstat -lnpt | grep 20000
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address  Foreign Address  State  PID/Program name
tcp        0      0 0.0.0.0:20000  0.0.0.0:*       LISTEN 32343/server_two_so
tcp        0      0 :::20000       :::*            LISTEN 32343/server_two_so
[root@quarks IPv6_test]$
```

Established sockets test

- How to check that a supposed IPv6 client (respectively IPv4) really connects through IPv6 (respectively IPv4) (*):

```
[duble@quarks IPv6_test]$ netstat -npt | grep 20001
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address               Foreign Address
State       PID/Program name
tcp        0      0 2001:660:3302:7003::2:20001 2001:660:3302:7003::3:54104
ESTABLISHED 8046/server_one_soc
tcp        0      0 2001:660:3302:7003::3:54104 2001:660:3302:7003::2:20001
ESTABLISHED 8047/client
[duble@quarks IPv6_test]$
```

(*) When applicable: to be done while the client is connected.

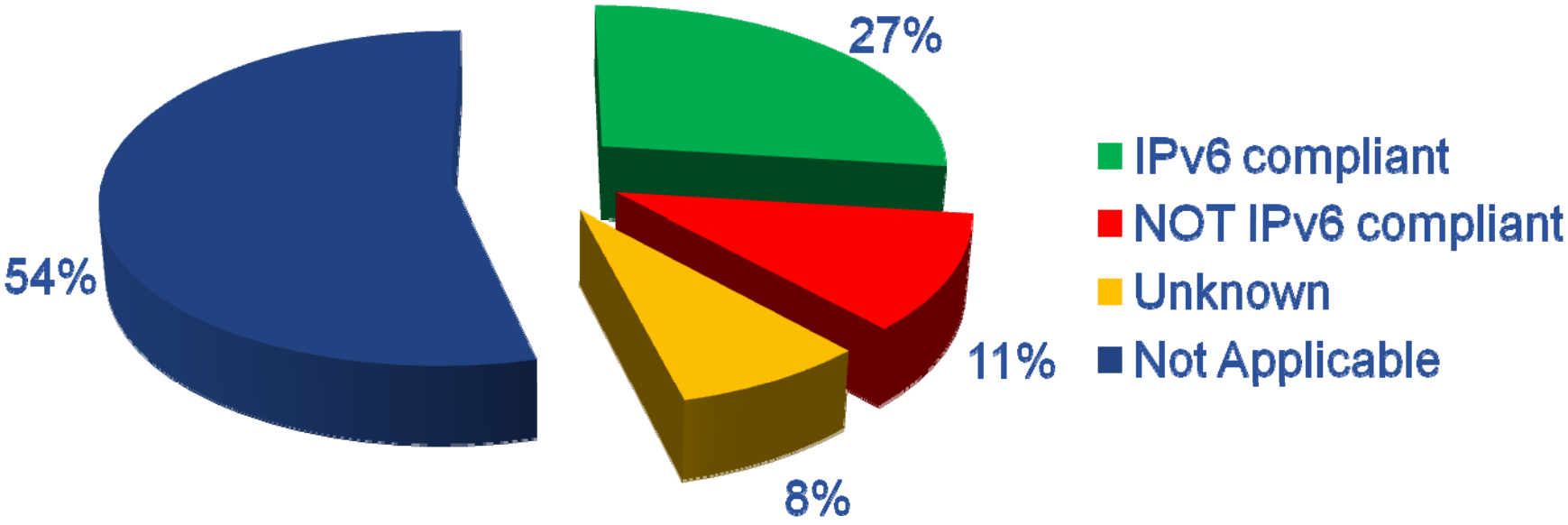
Other possible checks

- `lsof -i -n`
  (returns the same kind of information as: `netstat -lnpt ; netstat -npt`)
- Tests with a packet sniffer (tcpdump, wireshark…)

Are gLite external dependencies IPv6-compliant?

gLite external dependencies

IPv6 compliance on September 05, 2008
Total: 89 packages



Note: 6 packages stated as “not compliant” are mysql sub-packages.

If we consider that mysql is only 1 package, the “not compliant” packages are only 6%.

Packages not IPv6 compliant

- condor
- dcache-client
- mysql-* (6 packages)
- oracle-instantclient
- udpmon

Packages « Unknown »

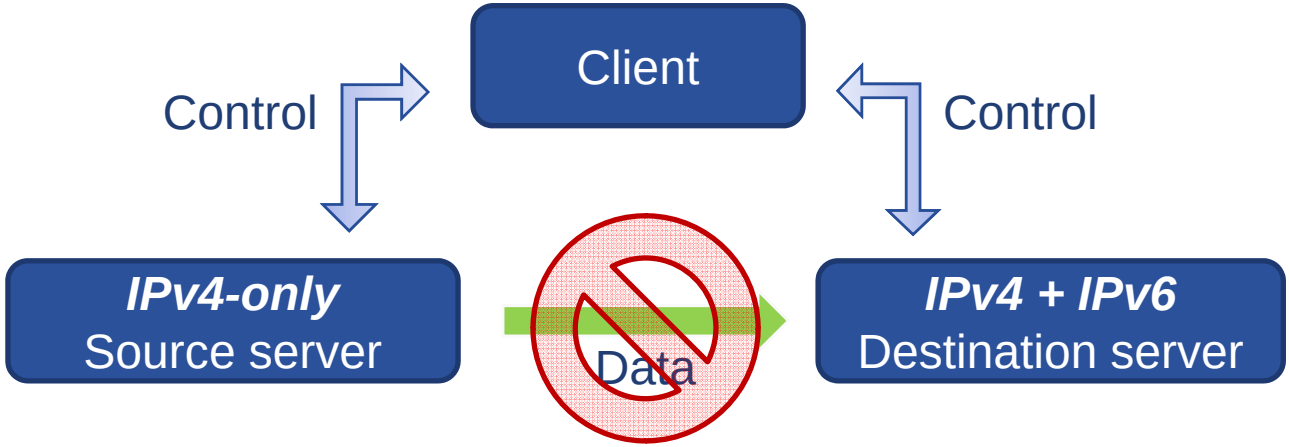
- gsoap 2.6.2 (newer versions have been tested)
- libhj
- pyopenssl
- soappy
- unixodbc
- vdt - Virtual Data Toolkit (The GridFTP part has been tested)
- zsi

Specific tests

- **As a part of the SA2 collaboration with JRA1, some in-depth tests have been done for specific packages. These packages where chosen according to the following criteria:**
  - They are very much used in the middleware, or
  - They are difficult to test, or
  - They may be used in the future
- **The packages chosen are:**
  - Globus GridFTP
  - Boost::asio
  - Web services
    - gSOAP
    - AXIS

Test of GridFTP

- **Methodology:**
  - Installation
  - Transfer tests in various conditions
- **Results:**
  - OK, except that a third-party transfer fails in this case



(Globus implementation does not fully respects the RFC 2428 "FTP extensions for IPv6 and NATs")

  - Report at https://edms.cern.ch/document/942683/1

Test of Boost::asio

- **Note: this may be used by gLite in the future.**
- **Methodology:**
  - Installation
  - Programming of an IPv6 & IPv4 enabled server and client
- **Results:**
  - It is IPv6 compliant
  - How to use it in a fully IPv6 compliant way: https://edms.cern.ch/document/935729/1.1

Test of gSOAP

- **Methodology:**
  - Installation of the package, creation of a web service and of the corresponding client.
  - Test of the Apache-based and standalone servers, test of the client
- **Results:**
  - A standalone gSOAP server will NOT accept IPv4 clients if `/proc/sys/net/ipv6/bindv6only` is set to 1.
  - The maintainer said he will have it corrected in the next version.
  - Except this, it is OK
  - Report at: https://edms.cern.ch/document/942749/1

Test of AXIS and AXIS2 Java

- **Methodology:**
  - Installation of the packages, creation of a web service and of the corresponding client.
  - Test of the Tomcat-based and standalone servers, test of the client
- **Results:**
  - Being Java programs
    - A standalone AXIS[2]/Java server will NOT accept IPv4 clients if `/proc/sys/net/ipv6/bindv6only` is set to 1.
    - Same result for a tomcat-based server (Tomcat is also Java-based)
  - Except this, it is OK
  - Report and usage advices at: https://edms.cern.ch/document/951917/1.2

Test of AXIS2/C

- **Note: this may be used by gLite in the future.**
- **Methodology:**
  - Installation of the package, creation of a web service and of the corresponding client.
  - Test of the Apache-based and standalone servers, test of the client
- **Results:**
  - NOT IPv6 compliant
  - Report at: https://edms.cern.ch/document/961176/1

What's next?

- Write a document summarizing the IPv6-related usage and limitations for the main programming languages
- Discuss with JRA1 for a list of packages which should be deeply tested, ordered by priority. Current list:
  - Test Python/ZSI
  - Test Perl/SOAP::Lite
  - Test PostgreSQL (should be IPv6 compliant)
- Test gLite external packages rated as « Unknown »
- Test the new version of BDII
- Report bugs for not IPv6 compliant packages
- Update the wiki page at: https://twiki.cern.ch/twiki/bin/view/EGEE/IPv6FollowUp
- Finalize the LD_PRELOAD-based IPv6 checker (see Annex)

------- Annex -------
A LD_PRELOAD-based IPv6 checker

Annex: LD_PRELOAD ipv6 checker

- **Mechanism: an LD_PRELOAD-able library**
  - At start, a non-static program will load:
    - The needed shared libraries (type “ldd <prog_file>” to list them)
    - + any library specified in the LD_PRELOAD environment variable
  - The LD_PRELOAD-ed libraries will be first in the chain
    - The functions defined in an LD_PRELOAD-ed library will overload any function with the same name in the following libraries.
  - The LD_PRELOAD-based ipv6 checker is a library which overloads each non-IPv6-compliant function of the C standard library (for example gethostbyname()) by a function with the same name; this new function will:
    - Generate an alert in /tmp/ipv6_warnings/<prog_name>/<pid>
    - Then call the original function
      - *How this is done: the ‘`dlsym(RTLD_NEXT, <name>)`’ call is used to look for a symbol (a function here) in the next libraries of the chain*

Annex: LD_PRELOAD ipv6 checker

- **Usage example:**
  - We would like to check the following program, which uses the IPv4-only function “gethostbyname()”:

```
[duble@bdii test]$ cat ./test.py
#!/usr/bin/python
import socket
print socket.gethostbyname("quarks.paris.urec.cnrs.fr")
```

  - A normal run gives:

```
[duble@bdii test]$ ./test.py
192.168.1.2
```

  - In order to check IPv6 compliance we run:

```
[duble@bdii test]$ LD_PRELOAD=/home/duble/ipv6_checker/libipv6_checker.so ./test.py
192.168.1.2
```

  - We see that the standard behavior is not affected.
    But actually a warning has been generated in /tmp/ipv6_warnings (see next slide).

Annex: LD_PRELOAD ipv6 checker

- Let's check it:

```
[duble@bdii test]$ ls /tmp/ipv6_warnings/
java  server.py  test.pl  test.py
[duble@bdii test]$ ls /tmp/ipv6_warnings/test.py
18552 18555
```

- This shows that "test.py" has been run twice, and the two numbers are the corresponding PIDs.
- The process we just ran is the last directory created:

```
[duble@bdii test]$ ls -1rt /tmp/ipv6_warnings/test.py
18552
18555
[duble@bdii test]$ ls /tmp/ipv6_warnings/test.py/18555/
getaddrinfo_AF_INET
```

- There was only one problem detected, represented by the file "getaddrinfo_AF_INET".

Annex: LD_PRELOAD ipv6 checker

– The content of the file gives a description of the problem, and a solution:

```
[duble@bdii test]$ cat /tmp/ipv6_warnings/test.py/18555/getaddrinfo_AF_INET

PROBLEM DETECTED:
-----------------------------------------
This program uses getaddrinfo() with hints->ai_family set to AF_INET.
Note: Python uses getaddrinfo() to process the gethostbyname() command [...]

SOLUTION:
-----------------------------------------
Use getaddrinfo() with AF_UNSPEC [...] in order to be address-family agnostic.
```

Annex: LD_PRELOAD ipv6 checker

- **Advantages:**
  - It works with all non-static programs, and also with Python, Perl scripts and Java code because their respective interpreter / virtual machine is dynamically linked with the standard C library
  - It does not affect the standard behavior of the program
  - It may easily be run for all programs on a node:

```
[duble@bdii test]$ export LD_PRELOAD=/home/duble/ipv6_checker/libipv6_checker.so
```

- **Drawbacks:**
  - The tool only detect non-IPv6-compliant function calls. There may be other (not common) kinds of non-IPv6 compliance problems which will not be detected.
  - It does not provide any source file name and line number.
    - It is complementary to the static code checker.