



Enabling Grids for E-science

IPv6 Testing

Etienne Dublé - CNRS/UREC EGEE SA2

etienne.duble@urec.cnrs.fr



Mario Reale – GARR

mario.reale@garr.it

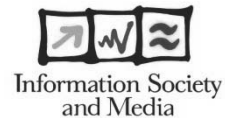
EGEE SA2



Thursday November 6, 2008 – Prague – JRA1 / SA3 All Hands Meeting

This document is available at <https://edms.cern.ch/document/975449>

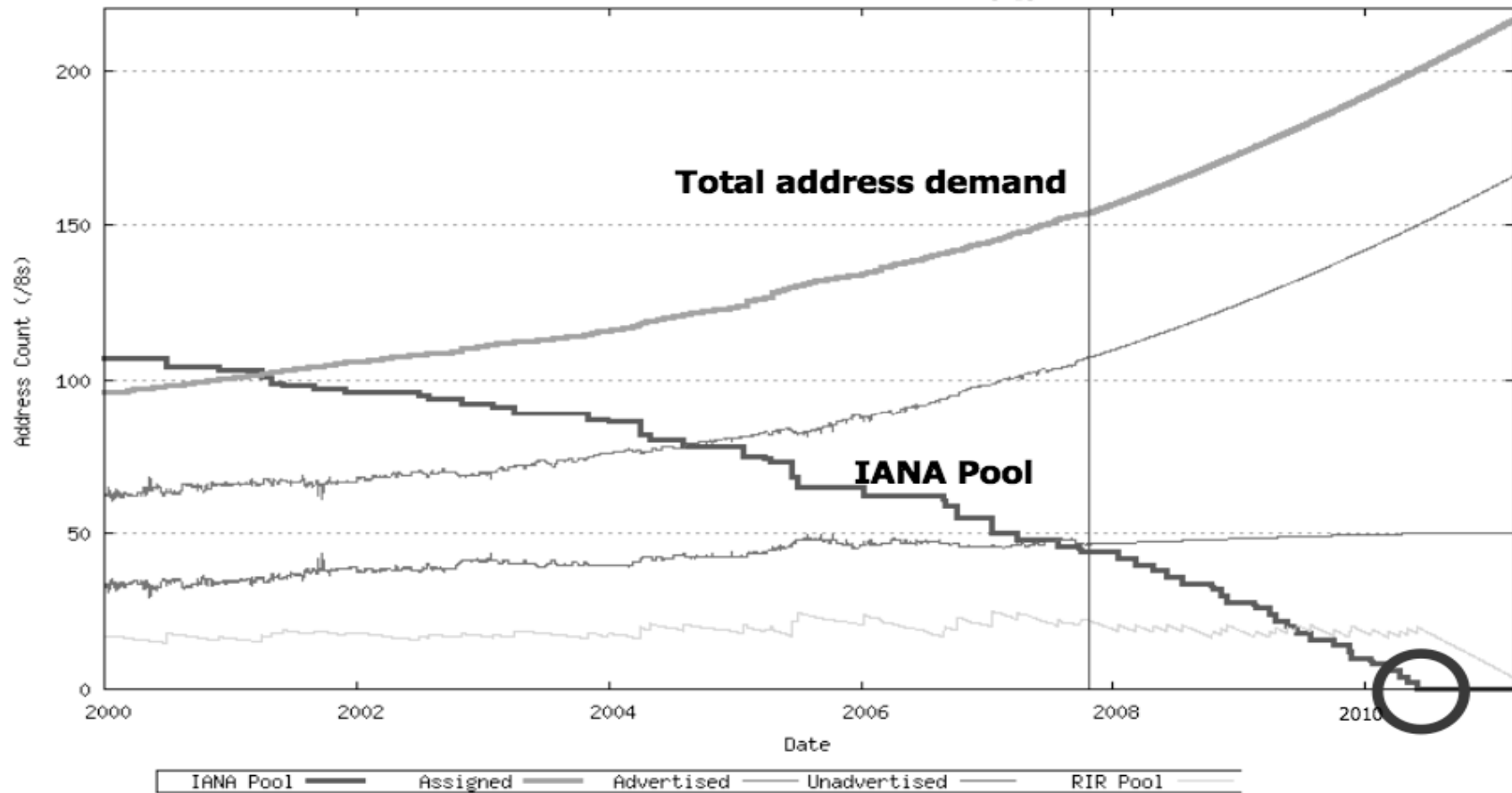
www.eu-egee.org



IPv6 overview in 3 slides

*minimal set of things to know about IPv6
for software developers and testers*

- **Main point: IPv4 addresses are getting more and more scarce**



- **Secondly, some features of IPv6 are not available in IPv4 (extensible header/protocol, embedded security and mobility, host auto-configuration...)**

IPv6 Overview: Addresses

- **128 bits** represented by **8 hexadecimal blocks** separated by colons “:”

- **URL**

```
2001:0660:3003:0001:0000:0000:6543:210F
```

- **Most relevant address types:**
 - **Loopback address** is `::1`
 - **Unspecified address** is `::`
 - **Link-local address:**
 - Starts with `FE80::`
 - Limited to the physical link, not routable, not usable for sockets
 - **Global unicast address:** Unique in the world
- You may have **several addresses** on one interface. Example:
 - `inet6 addr: 2001:760:0:106::21/64 Scope:Global`
 - `inet6 addr: fe80::20c:29ff:fed7:c812/64 Scope:Link`

- **New AAAA record for IPv6 addresses:**

```
myhost      in      A      192.134.0.49
            AAAA  2001:660:3002:7000::1
```

Example DNS query:

```
host -t AAAA ui.dir.garr.it
```

Result: ui.dir.garr.it has AAAA address 2001:760:0:159::212

- **If your DNS is only available through IPv4, dual-stack machines can still get AAAA records from it**
- **Important change in applications to resolve names**
 - The **gethostbyname()** deprecated function returned one IPv4 address OLD
 - The **getaddrinfo()** IPv6 compliant function returns **a list of addresses** (IPv4 and IPv6) NEW

- ***plug and play features:***
 - Auto-configuration of the link local address
 - Auto-configuration of a global IPv6 address as soon as an IPv6 router advertises itself
(another option is to use DHCPv6 → similar to DHCPv4)

IPv6 Linux node configuration

*First step to be able to test IPv6 compliance of
programs on your own machine*

- **The Goal is to setup a dual-stack machine**
 - i.e. to have both IPv6 and IPv4 available

- **IPv6 is supported since Linux kernel 2.4**
 - If IPv6 is supported by my kernel, the file `/proc/net/if_inet6` must be there
 - If not I have to load the IPv6 module:


```
modprobe ipv6
```

- **IPv6 address configuration**
 - Link local address: nothing to do, it is set automatically when IPv6 is enabled
 - Global address: can be
 - set manually
 - provided by autoconfiguration
 - provided by DHCPv6

- **Edit the `/etc/sysconfig/network` file and set there**
 - `NETWORKING_IPV6=yes`
 - `IPV6_DEFAULTGW=<the IPv6 address of your default IPv6 g/w>`
- **Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file and set there**
 - `IPV6INIT=yes`
 - `IPV6ADDR=<my IPv6 address>`
- **Edit the `/etc/resolv.conf` file and set IPv6 address of your IPv6 DNS, if any:**
 - `nameserver <IPv6 address of name server>`
- **Restart the network:**
`service network restart`

- To dynamically add an IPv6 address and g/w:

```
/sbin/ifconfig eth0 inet6 add 2001:760:4004:218::36
```

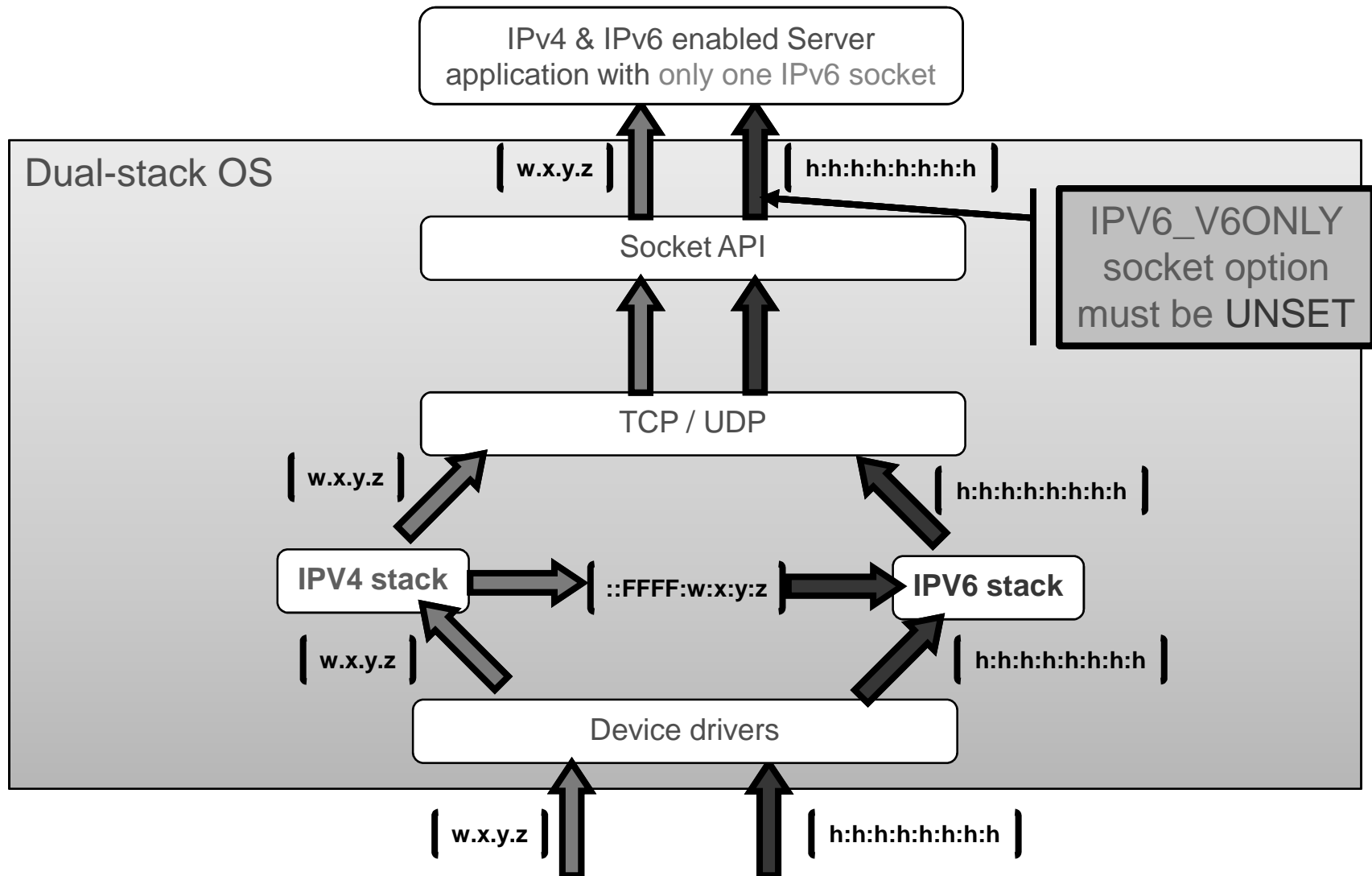
```
route --inet6 add default gw 2001:760:4004:218::1
```
- To print IPv6 routes:

```
/sbin/route -n --inet6
```
- **Am I connected to the IPv6 internet right now on my host?**
 - www.kame.net shows a dancing turtle if IPv6 is being used
- **Tip about IPv6 configuration when you are testing: to avoid modifying your name server, just add ipv6 hosts in /etc/hosts**
- **More info: <http://www.bieringer.de/linux/IPv6/index.html>**



Socket servers on dual-stack systems

Two Socket Servers Types



- **If the `IPV6_V6ONLY` option is not set or unset**, the IPv6 socket may or may not accept IPv4 connections, depending on the **system parameter `/proc/sys/net/ipv6/bindv6only`**.
=> The program will act differently on different systems!

How to check that a program is using IPv6?

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

```
[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) is really connected 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.

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

How to assess IPv6 compliance...

- by using the static
code checker?**

- **What is it?**
 - A bash script seeking for evident non IPv6 compliant patterns in the source code
- **How to use it?**
 - Using ETICS build system:
 - You can check the IPv6 metric on the ETICS UI (see next slides)
 - You can submit an IPv6 check job, for example on the `org.glite.data.transfer-fts` gLite component:


```
etics-submit build -p ipv6check="True" \  
org.glite.data.transfer-fts
```
 - Optionally the code checker can also be used by hand

- **Known limitations of the tool:**

- May detect calls in parts of the code which are actually not executed, like the following:
 - Patterns in comments (`/* gethostname() */`)
 - Pre-compiler instructions ignored (`#ifdef USE_FLAG ...`)
 - Code like `if (ipv6_disabled) { <ipv4-only-code> }`
- Only detects suspect non compliant patterns (there may be other kinds of non-ipv6 compliance)
- **It is not 100% efficient but it provides clear hints to be verified about non-IPv6 compliance.**

- **Reference information available from**

http://www.euchinagrid.org/IPv6/cod_checker.html



Data generated using Salvatore Monforte's 'IPV6'

IPV6 Code Compliance Checker

Project: etics_R_2_0_12_1 (org.etics)
Configuration: etics_R_2_0_12_1 (org.etics)
Date:
Success rate: 100 %
Status: Success

Component name	Configuration name	Result
org.etics.administration.web-application	etics-administration-web-application_R_1_3_0-1	Success
org.etics.build-system.browser	etics-build-system-browser_R_1_1_3_1	Success
org.etics.build-system.client-py	etics-build-system-client-py_R_1_3_10_1	Success
org.etics.build-system.java-utils	etics-build-system-java-utils_R_1_0_1_1	Success
org.etics.build-system.plugin-framework	etics-build-system-plugin-framework_R_1_3_7_1	Success
org.etics.build-system.webservice	etics-build-system-webservice_R_1_3_5_1	Success
org.etics.build-system.webservice-interface	etics-build-system-webservice-interface_R_1_3_5_1	Success
org.etics.build-system.webservice-stub-java	etics-build-system-webservice-stub-java_R_1_3_5_1	Success
org.etics.data-model.data-db-mysql	etics-data-model-data-db-mysql_R_1_3_0_0	Success
org.etics.data-model.schema-db-mysql	etics-data-model-schema-db-mysql_R_1_3_1_0	Success

io.j110/jaxb-api.jar:/opt/jaxb
 /lib/activat
 ion.iar:/oot/commons-

How to assess IPv6 compliance...

- by using the dynamic
(runtime) IPv6 checker?**

B
E
F
O
R
E

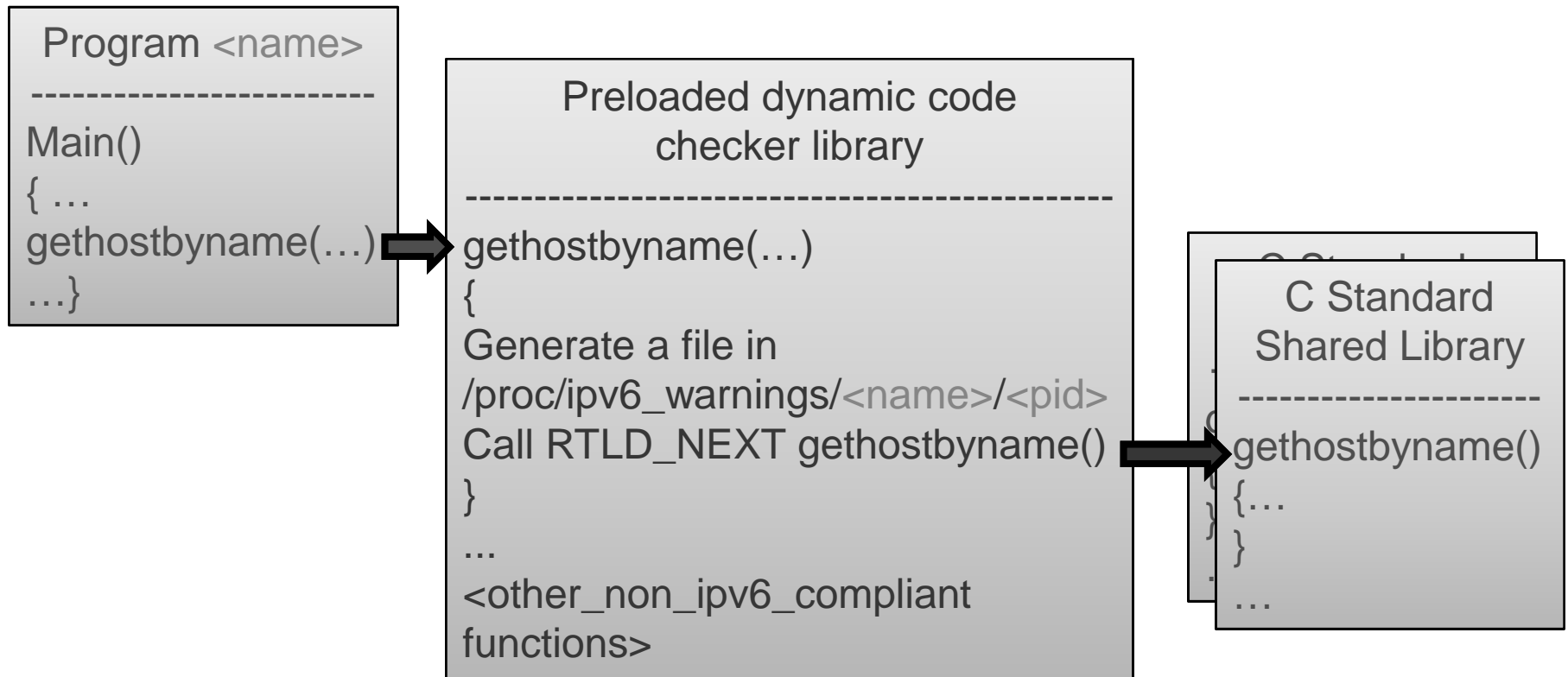


LD_PRELOAD=/path/to/library/A

A
F
T
E
R



LD_PRELOAD=/path/to/dynamic-code-checker-library



- **Usage example:** we want to test a program called test.py
 - test.py is normally run this way:

```
[duble@bdii test]$ ./test.py
Retrieving data from FTP server... done.
[duble@bdii test]$
```

- In order to check IPv6 compliance we run:

```
[duble@bdii test]$ LD PRELOAD=/home/duble/ipv6_checker/libipv6_checker.so ./test.py
Retrieving data from FTP server... done.
[duble@bdii test]$
```

- It seems that the standard behavior is not affected (it seems to work the same).
But, actually, this program is not IPv6 compliant, so a warning has been generated in /tmp/ipv6_warnings/test.py/ (see next slide).

– Let’s check it:

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

➤ The subdirectory correspond to the PID of the process.

```
[duble@bdii test]$ ls /tmp/ipv6_warnings/test.py/18555/
getaddrinfo AF_INET
[duble@bdii test]$
```

➤ There was only one problem detected, represented by the file “getaddrinfo_AF_INET”.

- 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.
```

```
[...]
```

```
SOLUTION:
```

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

- **Advantages:**

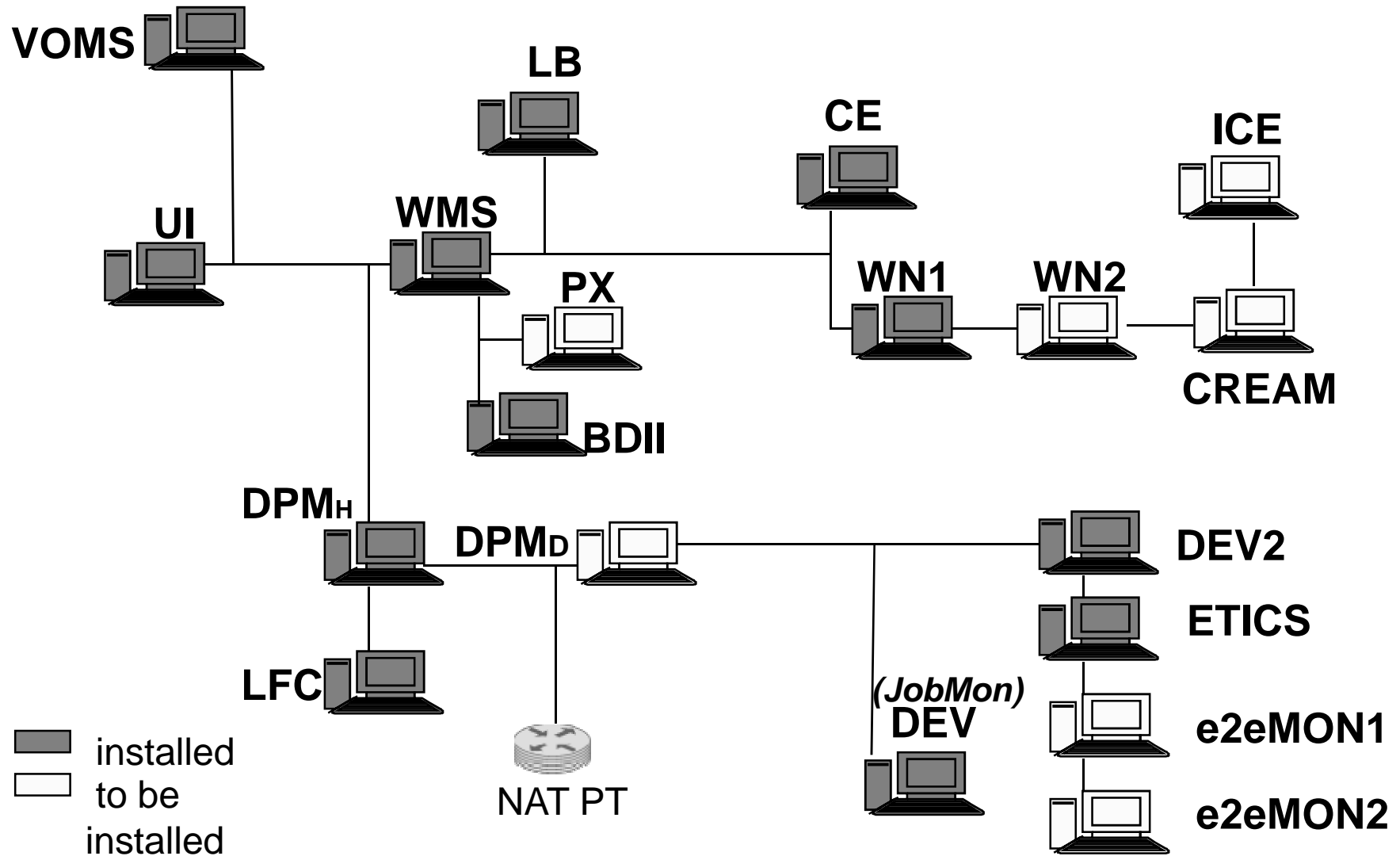
- **It works with all non-static programs, and also with Python, Perl scripts and Java code** because their respective interpreter / virtual machine is dynamically linked with the standard C library
- **It does not affect the standard behavior of the program**
- **It does not warn about parts of code which are not executed**
- **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.

Testbed, tools, and documents provided by SA2 for IPv6 testing



- **Reference documents on IPv6 for gLite developers: (all on SA2 EDMS or Wiki page)**
 - IPv6 Programming methods:
 - Guide to IPv6 compliant programming in C/C++, Java, Python and Perl:
 - *Provides a sample TCP client and server for each programming language*
 - *Explains advantages/drawbacks/limitations of each language regarding IPv6*
 - IPv6 Testing methods:
 - How to make sure the IPv6 behavior of your application is as expected
 - IPv6 Tests reports:
 - Assessment of the current status of the gLite external packages overall
 - Selected IPv6 compliance studies for specific packages: gSOAP, Axis / Axis2, Boost:asio, gridFTP
 - Assessment of the IPv6 compliance of gLite components: DPM, LFC
- **Provisioning of specific IPv6 introductory tutorials for gLite developers**
- **In collaboration with ETICS:**
 - IPv6 resources inside the metronome pool
 - A whole dedicated ETICS project for IPv6 studies

- **FINAL GOAL: General support on IPv6 for the gLite developers**
- **We are fully available and open for discussions on how to move on in the next months.**



Thank You.