IPv6 testing and deployment at Prague Tier 2

Tomáš Kouba, Jiří Chudoba, Marek Eliáš, Lukáš Fiala
Fyzikální ústav AV ČR, Na Slovance 2, Prague 182 21, Czech Republic
E-mail: koubat@fzu.cz

Abstract. Computing Center of the Institute of Physics in Prague provides computing and storage resources for various HEP experiments (D0, Atlas, Alice, Auger) and currently operates more than 300 worker nodes with more than 2500 cores and provides more than 2PB of disk space. Our site is limited to one C-sized block of IPv4 addresses, and hence we had to move most of our worker nodes behind the NAT. However this solution demands more difficult routing setup. We see the IPv6 deployment as a solution that provides less routing, more switching and therefore promises higher network throughput.

The administrators of the Computing Center strive to configure and install all provided services automatically. For installation tasks we use PXE and kickstart, for network configuration we use DHCP and for software configuration we use CFEngine. Many hardware boxes are configured via specific web pages or telnet/ssh protocol provided by the box itself. All our services are monitored with several tools e.g. Nagios, Munin, Ganglia. We rely heavily on the SNMP protocol for hardware health monitoring.

All these installation, configuration and monitoring tools must be tested before we can switch completely to IPv6 network stack. In this contribution we present the tests we have made, limitations we have faced and configuration decisions that we have made during IPv6 testing. We also present testbed built on virtual machines that was used for all the testing and evaluation.

1. Introduction

1.1. Motivation for IPv6 testing

There are several reasons why our site should be prepared for upcoming IPv6 era of the Internet:

(i) Lack of IPv4 addresses for our growth
(ii) Ability to talk to new HEP sites that may be IPv6-only
(iii) Utilize the fact that big Internet players switched to IPv6 after World IPv6 Launch on 6th June

1.2. Participation in community testbeds

Moving to IPv6 is a big step for established production sites, because many aspects of networking change in IPv6. The hardware is not as tested by time as in case of IPv4. Some consequences of IPv6 design are not yet fully understood in context of local scientific network (e.g. privacy extensions in automatic configuration can cause firewalling issues).

To understand challenges in IPv6 and to share experience in setting things up, we joined two IPv6 testbed projects. The first one was Hepix testbed[1]. It focuses on testing basic network setup and testing data transfers. The second testbed was established by EMI project[2] to test middleware installation and setup in dual-stack and IPv6-only environment.

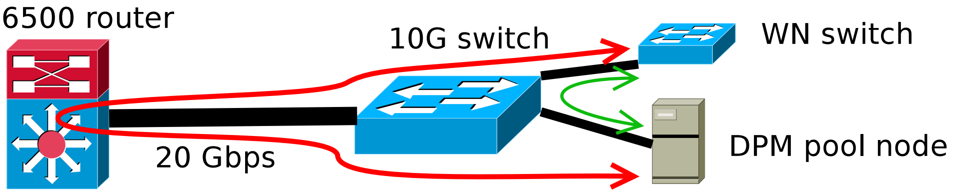

Figure 1. Routing schema of DPM traffic between WNs and disk arrays

2. Current IPv4 situation

Our computing center is assigned with only one C class IPv4 range of addresses. With about 300 worker nodes and about 400 servers in total this situation forced us to put the worker nodes behind NAT in a private address space.

However there is one major issue when using NAT in our environment. The DPM[3] nodes need to be accessible from the public address space of the Internet and also from our worker nodes in the private address space. If we separate the private and public network completely and only connect these networks by our main router we would face serious performance problems (our computing center is equipped with a 10Gb backbone and the traffic between worker nodes and DPM disk nodes is about 30Gb/s in peaks).

The second option - multihoming - is not available either. If we set the DPM to have one public address and one private address, the network communication would work fine, but we would break SRM protocol. Some SRM commands request the endpoint to always answer from the same IP address. So if the srmcp would start on our worker node and would try to copy from our DPM to an SRM endpoint out of our network, the transfer would fail because the remote SRM point would try to connect to the private address of our DPM.

Currently we use a workaround suggested by Maarten Litmaath in dpm-users mailing list. It uses the fact that the connection is initialized by the worker node. So we add the following static route on every worker node for every DPM disk node we have:

```
ip route add <IP-addr-of-DPM-pool> dev eth0
```

This asks the worker node not to use the default gateway (router doing NAT), but to communicate directly with the DPM even though it is not in the same network. The green line on the figure 1 shows how data flow with this workaround. The red line shows how the data between worker node and DPM node would flow otherwise.

In the IPv6 world we would have no such problem. All our nodes would fit into one /64 subnet and no routing or NAT would be necessary.

3. Adapting our workflow

We use IPv4 only network in current production infrastructure. We have established the following workflow when installing and configuring new hardware in our network:

(i) Obtain basic HW information and put it in our HW database application (type, disk size, performance, power consumption, size in Us, weight, MAC addresses)
(ii) Assign IP addresses to the machine. There can be several addresses - system, management, RAID array management etc.
(iii) Manually configure management interfaces (if present)
(iv) Add IP to DHCP for systems supporting DHCP
(v) Install operating system via PXE + RHEL kickstart
(vi) Set the roles of the new system in fabric management (CFEngine2 and puppet in the future)

(vii) Run CFEngine[4] and check the result configuration

(viii) Add system to monitoring tools (Nagios[5], Munin[6], MRTG[7], Ganglia[8])

If we wanted to extend these steps to support IPv6 the following had to be done:

(i) Adapt HW DB application for IPv6 addresses

(ii) Design the rules for IPv6 address assigning

(iii) Try to configure management interfaces with IPv6 addresses

(iv) Test PXE + kickstart installation over IPv6

(v) Convert the CFEngine configuration to Puppet[9] (not exactly IPv6-related)

(vi) Test Puppet over IPv6

(vii) Install monitoring tools with IPv6 support

Adapting our HW database for IPv6 addresses was not difficult although it is worth mentioning that routines for evaluating correct IPv6 address are a bit more complex than in IPv4 case.

Designing the rules for assigning addresses brought a discussion what type of network configuration would be appropriate for smooth transition from our IPv4 environment where we assign fixed IP addresses by DHCP according to the MAC address of the interface. In IPv6 there are practically three options how to configure network on a host: manual configuration, SLAAC + stateless DHCP (DHCP needed to obtain DNS resolvers), stateful DHCP. Manual configuration during host installation would be feasible for a small network (and we used it during installation of the first nodes of our IPv6 testbed). On a large network with complex network setup it would be very difficult to perform changes or to keep info about used addresses. The main problem with SLAAC is that the generated IP address depends on the MAC address of the network adapter which means it changes every time the adapter is replaced. We decided to use stateful DHCP that gives us the full control of the format and look of the IP addresses so we can for example have similar IP addresses for system and management interface of a given node.

3.1. Automatic installation

The rest of the steps performs automatic installation. We faced several problems when trying to install nodes with Scientific Linux version 5 or 6 automatically via network.

First the PXE over IPv6 is poorly supported by HW vendors. Even if firmware of a network adapter declares it supports IPv6 it is difficult to set up the address manually and we did not see any card supporting DHCPv6[12]. Open source implementation gPXE[13] contains support but most of the time it freezes during boot so it is practically unusable.

When the network is configured by PXE and the installation image is downloaded over tFTP the Anaconda installer is launched in Scientific Linux (SL). Unfortunately there are big differences between Anaconda in SL5 and SL6. Anaconda in SL5 does not support specification of DNS server. Anaconda generally does not support using proxy servers during installation process etc. We also encountered a strange timeout problem – Anaconda ends with a timeout error when downloading data from IPv6-only package repository, but a second attempt right after the first one works.

To workaround these problems we have set up a private IPv4 network which is not routed and serves only for installation purposes. The final statement in Anaconda configuration (so called kickstart) disables this installation network on the node.

4. Schema of Prague Tier2 IPv6 testbed

We have installed several nodes to form our IPv6 testbed. The schema and roles of the nodes are depicted on the figure 2.

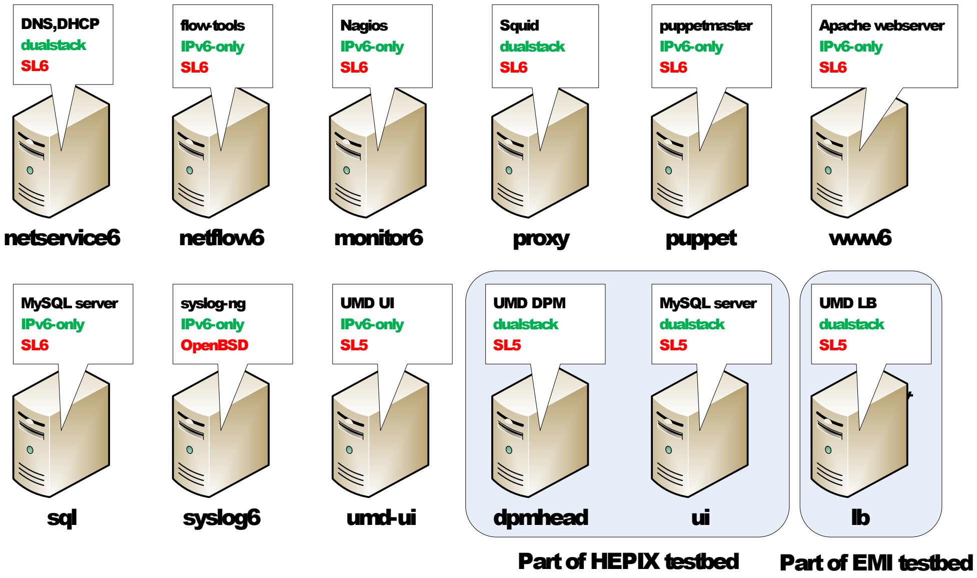

Figure 2. Nodes of IPv6 testbed at Prague Tier 2

5. Running EMI middleware in IPv6 testbed

The final goal of our testbed was to run EMI middleware so we tried to install it an configure it with the general configuration utility YAIM[14]. During these steps we encountered the following problems:

(i) YAIM depends on command `hostname -f` which is not working correctly on SL5 on IPv6-only host. This problem was solved with a wrapper script.

(ii) Fetching CRLs – only few certificate authorities (CA) publish their certificate revocation lists (CRLs) in a IPv6 friendly way. The workaround for this issue was setting up a proxy server and patching `fetch-crl` script to use this proxy.

6. Future work

In future work we would like to test more the middleware stack we run for HEP experiments. Especially batch system Torque and GRID gateway built upon this system – CREAM[15] computing element – should be carefully tested. Currently we are aware of the fact that CREAM CE depends on uberftp[16] utility that is not IPv6 ready (although there are patches developed by Francesco Prelz in Hepix IPv6 working group).

7. Conclusion

This paper summarizes steps needed to move Prague Tier 2 nodes and services to the new Internet protocol version 6. We do not see any crucial problem that could not be solved or at least mitigated with a workaround. There is much effort put into IPv6 these days: IPv6 launch day, RFC describing experience of running IPv6 only network - RFC6586[10], IANA[17] defining new private IPv4 networks for transition mechanisms - RFC6598[11]. From our experience we still feel it will be lot of work before IPv6 could be adopted without problems on all our network services.

7.1. Acknowledgement

This work is partially supported by CESNET, z. s. p. o. project number 416R1/2011.

References

[1] Mario Reale, "HEPiX IPv6 distributed testbed," https://w3.hepix.org/ipv6-bis/doku.php?id=ipv6:testbed.

[2] EMI, "European Middleware Initiative," http://www.eu-emi.eu/.

[3] DPM development team, "Disk Pool Manager," https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm.

[4] CFEngine AS, "CFEngine," http://cfengine.com/.

[5] Nagios Enterprises, "Nagios," http://nagios.org/.

[6] Munin development team, "Munin," http://munin-monitoring.org/.

[7] Tobi Oetiker, "MRTG," http://oss.oetiker.ch/mrtg/.

[8] Ganglia development team, "Ganglia," http://ganglia.sourceforge.net/.

[9] PuppetLabs, "Puppet," http://puppetlabs.com/.

[10] J. Arkko and A. Keranen, "Experiences from an IPv6-Only Network," http://tools.ietf.org/html/rfc6586.

[11] J. Weil and V. Kuarsingh and C. Donley and C. Liljenstolpe and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space," http://tools.ietf.org/html/rfc6598.

[12] The Internet Society, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," http://tools.ietf.org/html/rfc3315.

[13] Etherboot project, "Etherboot/gPXE Wiki," http://etherboot.org/wiki/start.

[14] European Middleware Initiative, "YAIM," https://twiki.cern.ch/twiki/bin/view/EGEE/YAIM.

[15] European Middleware Initiative, "CREAM Computing Element," http://grid.pd.infn.it/cream/.

[16] Jason Alt, "UberFTP," http://dims.ncsa.illinois.edu/set/uberftp/.

[17] IANA, "Internet Assigned Numbers Authority," http://www.iana.org/.