PROCEEDINGS OF SCIENCE

IPv6 deployment in Tier2 site at FZU in Prague

Marek Eliáš*, **Lukáš Fiala, Jiří Chudoba, Tomáš Kouba**
Institute of Physics ASCR, v. v. i. (FZU)
E-mail: elias@fzu.cz, fialal@fzu.cz, jiri.chudoba@cern.ch, koubat@fzu.cz

Deployment of IPv6 is a technological challenge for grid computing, but by progress of time it is turning into a necessity because of IPv4 address exhaustion. Computer Centre of Institute of Physics (FZU) provides computing facilities mainly for high energy physics (HEP). It needs about three times more IP addresses than it possesses. This problem was solved in 2009 by moving computing nodes to private address space, but this solution demands routing between computing elements and services which cannot listen on two different IP addresses like DPM.

In this contribution we present our experiences with running services essential for computer centre management and monitoring in IPv6 environment. We test automatic system installation through PXE and a central configuration management like cfengine. We test tools used in our monitoring framework which consists of tools using SNMPv6, netflow and nagios. Last but not least we consider a solution for accessing remote management interfaces like ILO, IMM or IPMI by remote IPv6 client together with ensuring reasonable level of security.

The International Symposium on Grids and Clouds (ISGC) 2012,
Febraury 26 - March 2, 2012
Academia Sinica, Taipei, Taiwan

*Speaker.

PoS(ISGC 2012)024

1. Introduction

The Computing Centre of the Institute of Physics operates about 400 servers of different types from various vendors. Projects with the highest installed computing capacity are high energy physics experiments ATLAS, ALICE and D0, local groups of solid state physicists and astroparticle project Pierre Auger Observatory. The computing and storage capacity is partially renewed every year. After a steady increase over a number of years in the number of computing worker nodes and support servers we are seeing a small reduction in the number of servers. This is caused by increased number of cores on CPUs and advanced virtualisation techniques. The actual number of virtual services running on different virtual servers is increasing and is expected to continue to grow. This comes with an increased demand for IP addresses.

1.1 Motivation for transition to IPv6

Exhaustion of IPv4 addresses is a common argument for IPv6 transition. In our computing centre it is similar. We possess only one C class subnet and we need at least three times more. This problem was solved by moving all worker nodes to the private address space. But this causes other problems.

For example we use the DPM software for heavy data transfers from DPM pool nodes to worker nodes. But DPM does not support multihoming and worker nodes should access DPM pool nodes through the same IP address as the rest of the world does. This means that the traffic between worker nodes and DPM pool nodes must be routed. Situation is described in the picture 1. Now we have two 10 Gigabit switches and the traffic from DPM pool nodes to worker nodes is in peaks about 30Gbps. Routing of such amount of traffic is not feasible for us.

We deployed a solution suggested by Maarten Litmaath[1]. DPM pool nodes are connected directly to both the private and the public network. Worker nodes from the private network have setup a static route for each DPM pool to access its public IP directly without any routing:

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

This approach works fine, but should be considered as a workaround rather then a systematic solution for this type of problems. It is not very convenient to add a static route to all worker nodes whenever adding a DPM pool node to the production. Also the network setup of the DPM pool nodes is rather complicated.

Deployment of IPv6 would solve this problem. Our institute have 256 IPv6 /64 subnets. Since each of these subnets contains 2^{64} IPv6 addresses, all our worker nodes together with DPM pool nodes would fit into a single subnet and no routing would be needed.

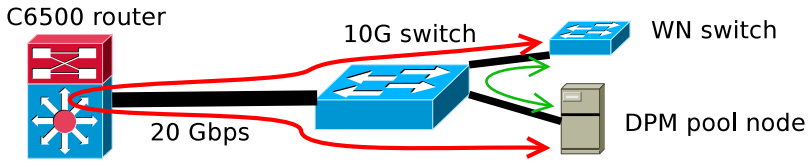

Figure 1: Scheme of routing between worker nodes and DPM pool nodes

[1] email correspondence in the mailing list dpm-users-forum@cern.ch from 27th October 2010

2. IPv6 related hardware problems

Transition to IPv6 involves network devices and also other hardware which is connected to the network and does not run an operating system in addition to production servers. This includes routers, switches, management interfaces of servers (ILO, IMM etc), thermometers and many other types of devices. In this section we consider IPv6 support of our current networking hardware as well as management interfaces of servers.

2.1 Cisco Firewall

We have a central router Cisco C6500 with Firewall Service Module (FWSM). Our FWSM is not able to filter IPv6 packets in a transparent mode. One possible solution is routing both IPv4 and IPv6 in a routed mode or setting up a multicontext mode, where it is possible to switch IPv6 in a routed and IPv4 in a transparent mode. Both of these solutions need downtime of the entire firewall.

However there is a possibility of a temporary workaround. FWSM can filter packets by ethertype. This means that FWSM can pass all traffic with ethertype `86dd` (IPv6) through the firewall also in transparent mode. Nevertheless one should keep in mind the security consequences of this approach.

```
access-list outside_ether_access_in remark IPv6
access-list outside_ether_access_in ethertype permit 86dd
access-list inside_ether_access_in remark IPv6
access-list inside_ether_access_in ethertype permit 86dd
```

2.2 Recent Cisco security bug

In September 2011 a security advisory about a denial of service vulnerablity was released [1]. An attacker could cause the router to reload by sending a malformed IPv6 packet to the right interface of the router. Nearly all IOS versions were vulnerable, but fixes were available within the advisory release. Only possible workaround was to turn off the IPv6 support.

2.3 Switches

Our switches generally support switching IPv6 traffic and we are not aware of any problems or performance issues.

On the other hand only two of our switches as well as the Cisco router are able to configure an IPv6 address on their management interfaces. Surprisingly these are fairly old SMC switches and none of our recently purchased switches supports these functionality. More details can be found in the table 1.

2.4 Management interfaces

Unfortunately none of our machines support IPv6 on a management interfaces. You can find an illustrating list of our hardware in the table 2.

Hardware name	switching	management	SNMPv6
Cisco Catalyst C6500	Yes	Yes	Yes
HP ProCurve J4904A Switch 2848	Yes	No	No
SMC TigerStack II 10/100/1000	Yes	Yes	Yes
HP ProLiant BL p-Class C-GbE2	Yes	No	No
HP GbE2c Switch c-Class BladeSystem	Yes	No	No
Force10 S2410-01-10GE-24P	Yes	No	No
BNT RackSwitch G8124	Yes	No	No
BNT RackSwitch G8000	Yes	No	No

Table 1: IPv6 support in networking hardware

Hardware name	Mgmt	PXE
HP BL 35p	No	No
HP BL 460c	No	No
HP DL360 G3 – G6	No	No
IBM x3650 M2	No	No
IBM iDataPlex dx340	No	No
IBM iDataPlex dx360 M2	No	No
IBM iDataPlex dx360 M3	No	No
SGI Altix XE 310	No	No
SGI Altix XE 340	No	No
SGI C1001-G13	No	No
Supermicro X8DTU	No	No

Table 2: IPv6 support in server hardware

2.4.1 IBM System x3550 M4

We tested a pre-production server x3550 M4 from IBM. Its management interface supports IPv6. Address can be configured manually, using SLAAC and using DHCPv6. In DHCPv6 configuration mode it does not accept routing advertisements, so the address configuration via SLAAC needs to be switched on too, if the routing is needed.

Web interface works through IPv6 without any problems. However we were not able to test the remote console, because it was a testing model and an appropriate license has not been delivered.

3. IPv6 testbed

Before a production deployment of IPv6 could begin some experience was needed to establish best practice. These was our main reasons for establishing an IPv6 testbed. We needed to decide how to configure the network and all possibilities needed to be tested. Also our current processes of administration of the computing centre, monitoring tools and other management services needed to be tested and possibly modified.

Our approach is the following: to try to setup a small "computing site" running IPv6-only with grid services, middleware, workernodes and batch system. Now we are done only with the network, monitoring and management parts; the rest is still to be done.

Testbed infrastructure can possibly be used as an experimental way of connecting a production service to IPv6 world if needed.

In this section we describe our experiments, the decisions we have made and the best practices we learned.

3.1 Testbed description

We have several VLANs dedicated to IPv6 testbed. Our router is dual stack but does not provide any interconnection between IPv6 and IPv4.

We have two dedicated machines intended as hosts for many virtual servers belonging to the testbed. In addition one of our production DNS servers serves as a resolver for nodes in the IPv6 testbed. Almost all servers run Scientific Linux 6.1 (SL 6.1) We run the following services:

- DNS and DHCPv6
- GLite user interface (participating in HEPiX IPv6 testbed, SL 5.7)
- Puppet (configuration management)
- PXE install server
- webserver
- Nagios
- MRTG (Multi Router Traffic Grapher)
- netflow collector (flow-tools package)
- syslog server (syslog-ng on OpenBSD 5.0)
- HTTP proxy (squid)

4. Core network services (DNS, DHCP, NTP)

4.1 DNS

Our production DNS zone farm.particle.cz as well as our zone dedicated to IPv6 testbed (ipv6.farm.particle.cz) and all reverse zones are fully resolvable through IPv4 as well as IPv6. DNS resolvers are available for both IPv4 and IPv6 clients in local network. Our production name servers run bind version 9.3.6 and name server from testbed version 9.7.3. No IPv6 related bugs of bind were encountered in our setup.

To ensure resolvability from IPv6 only host several steps need to be done. Some of the authoritative names servers should have IPv6 address and AAAA record. This probably means they should be dual stack — IPv4 will probably be needed for synchronization of the zone with other

authoritative servers. Secondly an AAAA glue record for this nameserver should be inserted in the parent zone. Finally it is needed to ensure resolvability of all parent zones and eventually perform above steps for them too.

The DNS resolver for IPv6 testbed is a dual stack node. Many sites or resources on the Internet which are accessible through IPv6 cannot be resolved correctly through IPv6. One such example are package repositories of Fedora [5]. Also not everybody who is trying IPv6 has IPv6 enabled authoritative servers. Therefore IPv4 connectivity is often necessary for a DNS resolver.

4.2 DHCP

4.2.1 Means of network configuration in IPv6

IPv6 introduces a new way of address configuration — Stateless Address Autoconfiguration (SLAAC). With SLAAC a device receives IPv6 prefix from a routing advertisement and it choses one address from this prefix. Selection of the address is typically deterministically derived from MAC address using EUI-64. Stateful DHCP is similar to DHCP in IPv4 except that it does not advertise routes. Stateless DHCP can advertise DNS resolvers, NTP servers and similar services on the network and cannot be used to assign an IPv6 address nor to setup routes.

Currently there are three means of network configuration on a host: manual configuration, SLAAC + stateless DHCP (because SLAAC do not advertise DNS resolvers[2]) and stateful DHCP. Manual configuration is not suitable for large computing centres. We were considering stateful DHCP and SLAAC. The problem of SLAAC is the following: when the network adapter in a host is changed, the IPv6 address changes too. We decided that we could not rely on DNS in this manner and that we do not want hosts IPv6 address to change in such a situation. In addition there would be problems when using backup links for example. On the other hand implementation of SLAAC is simpler and we possibly could expect that new IPv6 capable hardware devices like switches and thermometers would implement SLAAC instead of stateful DHCP.

4.2.2 Currently deployed solution

We decided to use stateful DHCP, we use ISC DHCP 4.1 on server and clients. Stateful DHCPv6 server does not assign address to a host according to its MAC address. New identifier of a host was invented: DUID. There are several types of DUID i. e. type LLT which includes a time stamp and changes in time. This one is default in configuration of ISC dhclient and is not suitable when we need to assign a fixed address to the host. We decided to use DUID type LL. Use of this type by dhclient is configured in dhclient.conf by line

```
send dhcp6.client-id = concat(00:03:00, hardware);
```

The actual DUID then looks like `00:03:00:01:<MAC ADDRESS>`. In DHCP server configuration this DUID should be used to assign IPv6 address to the host. But since DHCP 4.2 there is a possibility to match both LL and LLT DUIDs in a single statement `hardware ethernet` similar to DHCPv4 [3].

DHCPv6 does not setup routes, they are advertised using Routing Advertisement (RA). One should ensure, that sysctl `net.ipv6.conf.default.accept_ra` is set to 1.

[2] In the future there should be a possibility to configure DNS resolvers through SLAAC [4]

4.3 Network time protocol (NTP)

NTP server ntpd runs out of a box on IPv6. No special setup is needed. We simply added an IPv6 address to our production NTP server and now it is used for IPv6 testbed too. This means we run NTP server for our network on a dual stack node. It is setup to synchronize the time with stratum 1 through IPv4 and serves clients from both IPv4 and IPv6 networks.

We asked our network provider about possibility to access its time servers through IPv6. It is not currently available, but is planned in future years.

5. Monitoring tools

All monitoring software run on one host. We try to implement the same monitoring functionality as we have in production IPv4 network. This includes monitoring of linux servers as well as physical hardware like switches etc.

5.1 Nagios

Runs out of a box. Possible problem is monitoring of a dual-stack node, where it is not clear when the node should be treated as down.

SNMP monitoring: need to make custom check command: snmpget and snmpwalk need IPv6 address written in a special form

```
snmpget "ipv6:[fec0::dead:beef]"
```

This is needed also when specifying a hostname of an IPv6-only host.

5.2 MRTG

Monitoring of switches through IPv6 runs smoothly out of a box on SL6.1.

6. Configuration management (puppet)

Must be configured manually to listen on IPv6 adresses. Reverse lookup for IPv6 addresses must work properly at least from local network. Nearly whole IPv6 testbed is currently configured with puppet except some hosts that are configured by hand.

7. Automatic installation

In production part of computing centre we use automatic network installation using PXE. Network boot works quite differently in IPv6 however. Instead of a `next-server` option in DHCP there is a `boot-file-url` option which contains the URL of an image that should be loaded by the client. Boot options for network boot are described in RFC 5970 [2] from September 2010. ISC DHCP and Dibbler does not seem to support these options and we are not aware of any DHCPv6 implementations that would implement this RFC.

We do not have any hardware with IPv6-aware implementation of PXE. We tested an open source implementation gPXE. We tried to install it on a piece of old hardware and some network adapters were burn out during the installation so burning gPXE on all network interfaces in a

computing centre does not seem to be a good idea. gPXE is able to configure IPv6 address using SLAAC but can not configure DNS resolvers and it does not seem to support mentioned RFC 5970.

We can say that automatic network installation in IPv6-only network is not available today and that some kind of installation network based on IPv4 is needed.

7.1 Description of deployed solution

There is an unrouted IPv4 network inside an IPv6 VLAN, IPv4 addresses are configured using a DHCPv4 server. The installation proceeds via IPv4, the client downloads required packages through a proxy. After the installation the IPv4 is disabled and after the reboot client gets an IPv6 address and optionally is configured using configuration manager like puppet.

We need the following functional services:

- DHCPv4 and DHCPv6 server
- HTTP proxy with connection to IPv4 internet
- PXE install server accessible through IPv4
- Optionally puppet server

8. Interoperability of IPv6 and IPv4

Sometimes it is necessary to interconnect two IPv6 networks through IPv4-only network or IPv4 networks throgh IPv6-only network. For example when administrators need access to IPv6-only services but are currently in IPv4-only network.

8.1 IPv6 through IPv4 network

First option is 6to4. It uses public 6to4 gateways and there is no possibility to control through which routers and 6to4 gateways will the data flow. This implies possible performance, security and stability problems. We did not setup a 6to4 gateway.

We think that more suitable to our needs is a 6in4 point to point tunnel. A dual stack gateway is needed and a tunnel is setup statically by following command:

```
host-A:~# /sbin/ip tunnel add sit1 mode sit ttl <ttldefault> \
remote <host-B-ipv4-addr> local <host-A-ipv4-addr>
```

IPv6 addresses are then configured on both sides. Moreover forwarding should be enabled on the gateway side and suitable routes configured on the client side. This tunnel works well inside an IPSec tunnel.

8.2 IPv4 through IPv6 network

The first possibility is to setup a tunnel using `ipv6_tunnel` kernel module. We can setup a tunnel interface by following command:

```
host-A:~# /sbin/ip -6 tunnel add mytun mode ipip6 remote \
remote <host-B-ipv6-addr> local <host-A-ipv6-addr>
```

Created tunnel interfaces are then configured with IPv4 addresses on both sides. Forwarding need to be enabled on the gateway side and suitable routes configured on the client side. This can be used to access private IPv4 network (ie network with management interfaces of hardware) by machines from local IPv6 network. If combined with point to point IPSec encryption of IPv6 traffic, this can be used for remote access from IPv6-only network. In addition one could possibly use this solution to establish a routing between two private IPv4 address spaces on sites which are connected together only by IPv6. Maybe there will be more IPv6-only sites in the future. No performance testing of this solution has however been done.

Another possibility is a SSH tunnel. A client connects through IPv6 to a remote gateway using `ssh -w`. Tunnel interfaces will be created automatically on both ends. These interfaces need to be configured with IPv4 addresses, forwarding and routes need to be configured suitably.

9. Conclusion

In this paper we summarized our attempts to run services which are essential for administration of our computing centre. We conclude that we did not find any show-stopper which could make a transition of the computer centre to IPv6 impossible. For major problems we have found a solution or at least a workaround. On the other hand, we have to admit that these solutions need much more testing. In addition the standards are still evolving and new open questions are still emerging.

Acknowledgments

This work was supported by Academy of Sciences of the Czech Republic and by CESNET, z. s. p. o. project number 416R1/2011.

References

[1] Cisco Systems, Inc. Cisco IOS Software IPv6 Denial of Service Vulnerability. http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20110928-ipv6, September 2011.

[2] T. Huth, J. Freimann, V. Zimmer, and D. Thaler. DHCPv6 Options for Network Boot. RFC 5970 (Proposed Standard), Sept. 2010.

[3] Internet Systems Consortium. Internet Systems Consortium DHCP Distribution Version 4.2.0-P1 Release Notes. http://ftp.isc.org/isc/dhcp/dhcp-4.2.0-P1-RELNOTES, October 2010.

[4] J.-H. Jeong, B.-Y. Kim, J.-S. Paru, and H.-J. Kim. IPv6 Router Advertisement based DNS Autoconfiguration. http://tools.ietf.org/html/draft-jeong-ipv6-ra-dns-autoconf-00, April 2003.

[5] Red Hat Bugzilla. The Fedora website is not accessible from IPv6-only machine. https://bugzilla.redhat.com/show_bug.cgi?id=758317, February 2012.