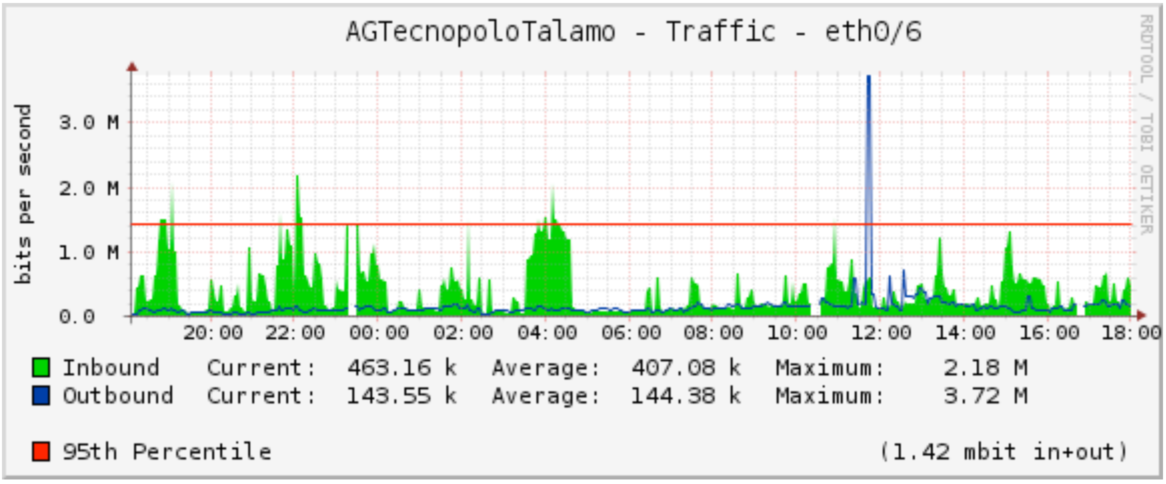
Getting ready to World IPv6 Day 2012

http://www.worldipv6day.org

ninux.org
WIRELESS COMMUNITY NETWORK

Introduction

A few months ago we published a graph on our homepage that shows in real time the last 24 hours of the bandwidth usage of our main IPv6 peering link.



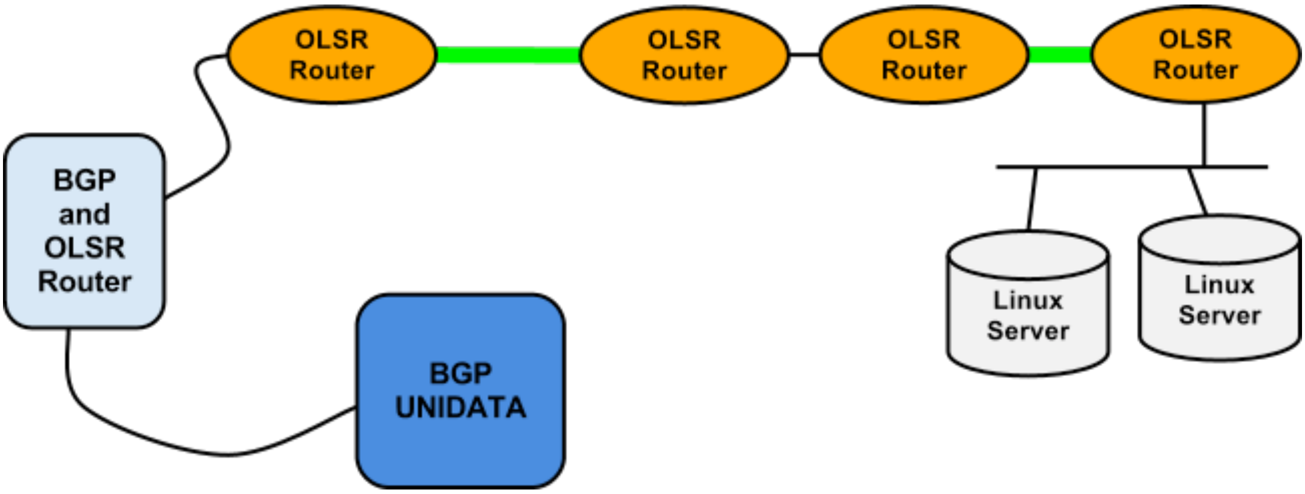
We started our IPv6 deployment in March 2011. Unidata, an ISP operating in the area of Rome (Italy), offered us the chance to make an IPv6 BGP peering providing to us free transit to the public Internet (IPv6 only).
We accepted immediately this chance to experiment with IPv6, as since the beginning of our community network, making experiments with new technology has been a leading factor.
We already had a private peering with Ydea.com, to route a few IPv4 addresses we used for mail servers and other lightweight services. To set up the second peering we started to think of the new network architecture and we faced how to pay the necessary resources.
In the early days of Ninux we started “playing” with wireless equipment together with Iacopo Iacarelli, the owner of Ydea.com. Our stories are different, because Ydea.com is a wireless ISP and we are a community network, but our relationship grew bigger and bigger over time, so he was very happy to help us out by giving us some useful tips to deal with the LIR, a required step to obtain the necessary resources.
We first upgraded our private peering with Ydea.com to a public peering, once our public AS number was registered at RIPE.
The 8th of April 2011 we installed our antenna on the roof of the Unidata PoP located in the eastern suburbs of Rome. We installed our second BGP router, a Linux Server running Quagga (the hardware was kindly offered by Unidata).
The 24th of May 2011 we had our IPv6 address block 2001:4c00:893b::/48 ready to use.

Strategy

To make sure it all worked well we moved by little steps. At Ninux we don't have a vendor that provides hardware and software upgrades, like it happens with large scale ISPs. We have a software team that builds firmware on top of devices. Firmware are open source like in Pisa (OpenWRT firmware) or based on some vendor product that has an available SDK, as in Rome where we use AirOS by Ubiquiti.

Because we were not sure that the new firmware was stable enough, at first we upgraded just a few nodes to connect the Ninux Farm to the BGP router at the edge of the network. The upgrade of this 4 nodes was not critical for the whole network, and it was a perfect test playground to generate enough traffic.

You can see the nodes involved in the first experiment on our mapserver at these links:
http://map.ninux.org/select/ninuxfarm/
http://map.ninux.org/select/tecnopolo/
http://map.ninux.org/select/fishsupernode/



Problems

We started working to provide native IPv6 access to the servers in the Ninux Server Farm, and we faced the first problems.
The hardest problem we faced was to have IPv6 support on our backbone devices. The Ninux.org backbone runs mostly on Ubiquiti devices of the M-Series. Running OpenWRT on these devices would give total IPv6 support, however the open source drivers coming with OpenWRT are not as stable as the proprietary ones embedded in the AirOS firmware, leading to poor performances when using OpenWRT.
Using the AirOS SDK we developed ourselves our own IPv6 solution, but we couldn't get something really stable until the end of september.
We had to face some problems, because Ubiquiti is very fast in releasing new firmware versions when there is a bug, but the SDK release is somehow delayed. So to speed up testing and development, in August we had to reverse engineer their latest firmware.
After a lot of work we have now a stable firmware :)
We published here the patches to apply on top of the Ubiquiti SDK to have a IPv6 OLSR router:
https://github.com/ninuxorg/UbntM5AirOsOLSR/tree/master/patches/AirOS5.3.3

Stateless addresses

When starting with IPv6 you have to give IPv6 addresses to the devices of your network. We had a /48 to be partitioned.
IPv6 gives enough room for messy address management without easily run into a collision of addresses. At first we used a wiki page, and everybody was taking note of the /64 used for his home.
The easier way to deliver an IP address to the end users was using radvd directly on the OLSR routers. However we had a lot of flaps because the routers were changing IP addresses. Soon we realized that we had to disable routing advertisements on routers that had already a manually configured IPv6 address.

```
echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
```

BGP and IGP

The Ninux network in Rome had just IPv4 private addresses. The routing protocol used is OLSR. Connecting to the big Internet we started to identify OLSR as our IGP protocol. We upgraded the network to run in dual stack IPv4 and IPv6, so running two instances of the OLSR protocol. After configuring some stuff without a real plan, we started to write a real routing architecture, that you can find now on our web site.

When you have only one peering point with a single upstream provider, it is enough to redistribute via BGP to your upstream a static route that matches all your address space. But if you have different peering points in your network, then you need to redistribute IGP routers to BGP to understand what is really reachable and should be announced.
This was not trivial, because we use OLSR as IGP routing protocol, and the integration with Quagga is still experimental. We are still testing Quagga version **0.99.21** with this patch https://dev.openwrt.org/browser/packages/net/quagga/patches/120-quagga_manet.patch

Public Services

We are very happy to run a network that is ready for IPv6. Our users are using IPv6 services since several months, and we expect this traffic to increase the 6th of June 2012.
In addition to normal IPv6 Internet access we also set up some IPv6 services for our community members and for the all Internet:

- **Jabber XMPP server**
  A Jabber server running ejabberd is available with both public ipv6 and ipv4 address at **jabber.ninux.org**. The server is federated with the rest of the XMPP network and has proven a useful service for our community.
  More information (in Italian) can be found here: http://wiki.ninux.org/Jabber.

- **Openarena**
  Openarena is a first person shooting game based on the open sourced quake3 engine. An Openarena server is available at **cleopatra.ninux.org:27960**. More information (in Italian) can be found here: http://wiki.ninux.org/openarena_it.

- **Mumble server**
  Mumble is an open source low latency voice chat software that is really useful for quick conferences.
  The server has public IPv6 **2001:4c00:893b:f2::2** and is reachable at port 64738.
  More information (in Italian) can be found here: http://wiki.ninux.org/Mumble.

- **Owncloud**
  Owncloud is an open source web application built in php that offers dropbox-like functionalities, we use it successfully to store documentation of our activities, like photos taken while building nodes or videos recorded at our events. But we don't use it only internally, infact we also allow other people who are not yet connected to our network to send us data: we often ask newcomers to send us photo taken on their roof in order to understand if we can connect them to any of our nodes, when they ask how they can send us the photos we usually give them access to our owncloud server.
  It has proved to be an useful and reliable service for our community. The server is reachable at https://obelix.ninux.org/owncloud/ with public ipv6 **2001:4c00:893b:fa:230:48ff:fe73:53d**.

Conclusion

The Ninux network is now ready for IPv6. Once we fixed the problems described in this document we updated all the nodes of the network. We have now 139 prefixes of size /64 active in our network. To the best of our knowledge we are one of the first Italian networks to deliver native IPv6 access to the home of Internet users.
After 1 year of testing and developing we feel ready for the world IPv6 day, and we are very proud of our experimentation done completely no profit in the spirit of free and open source software.
We hope to see our IPv6 traffic increase on June the 6th !
We want to thank our upstream providers, Unidata, Ydea and E4A. We are very happy to cooperate with experienced companies that have worked for many years in the field of networking, it is a fantastic example of how the open source community can create chances for cooperation and innovation.