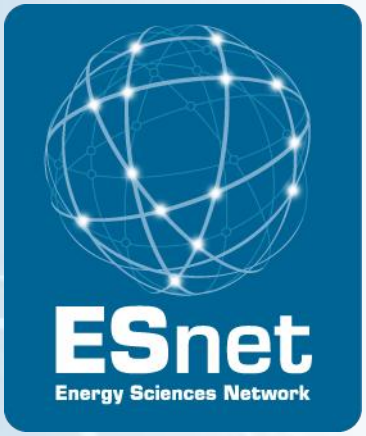

IPv6 perfSONAR Support in JET

Joe Metzger, Michael Sinatra, Network Engineers

ESnet Network Engineering Group

Large-Scale Networking

$Date$

Overview

- Update on IPv6 support in perfSONAR
- Status of IPv6 perfSONAR deployment among JET participants
- Lessons learned from the current round of testing
- Avenues for further research
- Recommendations
- Conclusions

Current IPv6 Status: perfSONAR Suite

- IPv6 support in perfSONAR has become robust and complete in most of the tools.
- BWCTL and OWAMP will prefer IPv6 if they see both A and AAAA records in DNS for a particular node. (This has implications for deployment.)
- Some still need some work, notably NPAD, which currently doesn't support IPv6.
- pingER and Traceroute MA/MP have IPv6 support in release-candidate (as of August 2011) code.

Current IPv6 Status: perfSONAR Suite

Tool	IPV6 Development Completed	Limited Deployment	Production Deployment Across Multiple Domains
BWCTL	✔	✔	✔
OWAMP	✔	✔	✔
pSB MA	✔	✔	
Lookup Services	✔		
Topology Service	✔		
SNMP MA	✔	✔	
PinGER	✔	✔	
NDT	✔		
NPAD	✔		
Toolkit Configuration Tools & GUI	✔		

IPv6 perfSONAR Deployment Status

- Internet2
  - IPv6 support in Internet2's perfSONAR nodes has been in place for some time.
  - Ongoing tests between nodes.
  - Tests occurring with other organizations and Internet2 members (e.g. U of Utah and UEN).

IPv6 perfSONAR Deployment Status

- ESnet
  - IPv6 support added to ESnet's ~60 perfSONAR nodes during the spring and summer of 2011.
  - Ongoing tests between nodes.
  - Currently using different hostnames to differentiate between IPv4 and IPv6. This is due to BWCTL and OWAMP preferring IPv6 over IPv4, and the need to troubleshoot both IPv6 and IPv4 with near-simultaneous tests.
  - Unclear how this is affected by the OMB IPv6 mandates. perfSONAR is a "public service."
    - Will likely maintain a canonical hostname with both A and AAAA records and then –v4 and –v6 hostnames for specific protocol testing.
    - Other ways to deal with this issue?

IPv6 perfSONAR Deployment Status

- NASA (EOS)
  - Dual stack measurement infrastructure based on Ensight.
    - Separate system from perfSONAR.
    - Uses many of the same tools as perfSONAR (e.g. BWCTL).
    - Different user interface.
  - Interdomain testing with IPv6 (where supported):
    - Internet2
    - DOE
    - NASA
    - NOAA
    - others

IPv6 perfSONAR Deployment Status

- TransLight/Pacific Wave (TLPW)
  - Currently planning and coordinating perfSONAR deployments among TLPW members.
  - University of Hawaii has begun deploying perfSONAR nodes.
  - IPv6 status uncertain at the present time, but there are plans to support it.

IPv6 perfSONAR Deployment Status

- University of Utah/Utah Education Network
  - Three perfSONAR nodes that are dual stack and doing interdomain testing.
  - Currently testing with Internet2's perfSONAR nodes.
  - Ongoing tests with other entities as well.

IPv6 perfSONAR Deployment Status

- TransPAC3 (TP3)
  - TP3 has been supporting IPv6 for some time and has been running dual-stack perfSONAR nodes where appropriate.
  - Ongoing tests between TP3 and JGN2 will begin in the near future (as of August 2011)
  - TP3 closely following perfSONAR development and is deploying IPv6 features and supported tools as they become available.

Lessons from the Deployment

- Parallel IPv4 and IPv6 tests are useful for diagnosing a variety of problems:
  - OS performance issues.
  - Router forwarding plane issues and punt-to-cpu problems in hardware-based switch-routers.
  - Routing issues (i.e. routing protocol and topology issues, topology asymmetries, etc.).
  - NIC hardware issues.
- This is important as more science data flows begin to use IPv6 for large-data transfer.

Lessons from the Deployment

- Inability to control BWCTL's and OWAMP's use of IPv6 vs. IPv4 can lead to unanticipated results.
  - Remote testing node's hostname updated in DNS to include both A and AAAA records.
  - Other hosts testing with this remote node will begin using IPv6 transport for their tests without notice.
  - Can lead to interesting results if there are performance differences between IPv4 and IPv6.
  - Observed by UofU/UEN nodes when Internet2 switched DNS to include AAAA records.
  - Led to ESnet's decision to maintain separate hostnames for now.
  - Does this require a change to perfSONAR or can we keep using separate hostnames? Impact from OMB IPv6 mandates?

Lessons from the Deployment

- OS performance can differ between IPv4 and IPv6
  - ESnet dual-stack nodes now running parallel IPv4 and IPv6 tests.
  - These nodes have 10GE interfaces.
  - Run the FreeBSD operating system, plus a small number of Linux-based hosts.
  - → Typical results: Chicago to Washington, DC (TCP throughput):
    - IPv4: ~8.2 gbps
    - IPv6: ~2.4 gbps
  - Linux doesn't exhibit a significant difference between IPv4 and IPv6, but doesn't perform as well as FreeBSD (8.x) in IPv4.

Lessons from the Deployment

- → Typical results: Chicago to Washington, DC (TCP throughput):
  - No significant difference in one-way delay between IPv4 and IPv6.
  - No noticeable packet loss with either protocol.
  - UDP performance very similar between both protocols.
- In examining TCP connections, I noticed a case were SACK holes followed by fast retransmits seemed to be ignored and the entire window segment eventually had to be retransmitted.
- → This turned out to be a different issue.
- Basically, in FreeBSD 8.x, TCP performs extremely well over IPv4 (better than Linux in many cases), but performs much worse over IPv6.

Lessons from the Deployment

- Investigation of this problem is still ongoing. Why might it be happening?
  - FreeBSD uses the KAME stack for IPv6. This stack is 13 years old, and much of the code was imported as-is and hasn't seen a lot of maintenance until now.
  - FreeBSD 9.0 will have a revamp of IPv6 kernel code.
  - The KAME stack appears in a number of other OSes.
- We're dealing with some old code that simply hasn't been exercised.
- Issue has been discussed with FreeBSD developers. It turns out there are many issues that need to be cleaned up and work is on going.
- → perfSONAR is actually helping to make OS stacks perform better in IPv6.

Lessons from the Deployment

- NASA/EOS discovered hardware NIC issue
  - TCP segmentation offloading (TSO)
    - Allows for very large frames to be sent from OS stack to NIC.
    - NIC will re-segment the frames into sizes appropriate for the MTU, while still maximizing throughput for any given MTU.
    - Frame segmentation done in hardware, leaving the OS free to pump out as much data as it can. Sounds great, right?
    - Unfortunately TSO can be buggy.
    - NASA noticed packet loss associated with TSO and bursty traffic. TSO disabled, packet loss went away.
    - Note that some NICs only support TSO for IPv4, not IPv6. Some support IPv6, but IPv6 TSO may be buggy (or buggier than IPv4).

Avenues for Further Work: Background

- Modern routers use a combination of a main CPU for the *control plane* (routing protocols, management, etc.) and specialized hardware, such as ASICs and FPGAs for the actual *forwarding plane.*
- As much of the stuff that needs to happen at wire-speed, such as the forwarding of IPv4 and IPv6 packets, needs to be done “in hardware.”
- IPv4 moved to forwarding in hardware many years ago, although there are still some CPU-based “software” routers in use.
- The result is two-fold: separation of the control plane and forwarding plane, and hardware-based forwarding, both of which improve performance and reliability.

Avenues for Further Work: Background

- When initially implementing IPv6 years ago, router vendors initially implemented IPv6 in software-only (CPU-based) routers. Hardware-based switch/routers, such as the Cisco 6500/7600 series, couldn't even support IPv6 at first.
- Eventually hardware-based routers could support IPv6, but they did all of the forwarding entirely in software!
- Slowly, hardware routers gained the capability of forwarding IPv6 traffic in hardware.
- However, there are still cases where enabling certain features or using certain configurations will cause IPv6 traffic to be forwarded in the main CPU.

Avenues for Further Work: Forwarding Performance

- Example: Cisco 6500 Sup 720:
  - Very popular backbone router among US universities.
  - Some even use it as a border router.
  - Most of the “work” is done in hardware.
  - Has a rather under-powered CPU, since most forwarding done in ASICs.
  - IPv6 now routed in ASICs, not CPU, *but* if you enable unicast reverse path forwarding in IPv6, *all IPv6 traffic gets sent to the CPU!*
  - *This doesn’t happen with IPv4!*
  - Unicast RPF is a very important feature for universities and DOE labs, as it is an easy way to prevent IP spoofing.

Avenues for Further Work: Forwarding Performance

- Example: Cisco 6500 Sup 720:
  - This is an example where there is *feature parity* between IPv4 and IPv6, but not *performance parity.*
  - Thanks to Jimmy Kyriannis of New York University for pointing this out to the US IPv6 community.
- Performance parity is still a problem, but so is feature parity, and it has end-to-end implications.
- → Using perfSONAR to better understand how changed to router configurations can change end-to-end relative performance of IPv4 and IPv6 will be a big win.

Avenues for Further Work: Relative Performance

- ESnet's experience with differences between IPv4 and IPv6 in FreeBSD is probably not unique to ESnet or FreeBSD.
- A quick look at NASA/EOS's Ensight site shows some cases were IPv6 performance is worse than IPv4.
- There are likely to be other cases out there where there are significant differences between IPv4 and IPv6 performance.
  - Old code that hasn't been exercised.
  - Too many cases of "punt-to-cpu" in hardware routers.
  - Buggy implementations (e.g. TSO)
- perfSONAR can really help to identify and isolate these issues. Dual-stack perfSONAR deployments will continue to become more important.

Recommendations

- It's important to understand whether a performance test is taking place over IPv4 or IPv6.
  - BWCTL and OWAMP don't allow specification of protocol.
  - But hostnames can help. Providing separate hostnames for IPv4 and IPv6 testing (while keeping a dual A and AAAA canonical hostname for compliance purposes) may be the best way to go.
  - Sites and networks deploying dual-stack perfSONAR nodes should choose a consistent naming scheme that clearly identifies the protocol represented by that hostname, e.g. psnode1-v4 and psnode1-v6.
  - In addition, testing metadata should include the protocol used so that this can be analyzed with the data itself.

Conclusions

- perfSONAR support for IPv6 has become sufficiently robust that it is now providing very useful information in understanding issues surrounding the relative performance of IPv6.
- We could improve the current situation by creating more ways to properly identify the protocol being tested, either via naming or command-line options (or perhaps both). Metadata could also include this information for fruitful results.
- There are still issues with IPv6 performance. More interdomain testing will be necessary to continue to identify and isolate these problems.
- Dual-stack deployment of perfSONAR needs to continue to expand, and this JET project has helped to build a community of perfSONAR users to enable better understanding of these issues and build momentum for continued and expanded perfSONAR deployment.