

NAT64

Demonstration Deployment

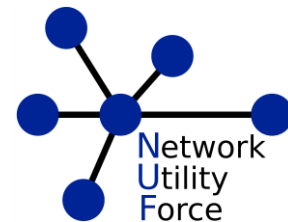
RMv6TF 2013

Demo network was available during the live presentation

Notes have been added to slides 30-32 to clarify FTP issues.

Agenda

- Introduction
- Problem Statement
- NAT64 Concepts
- Demo Setup
- NAT64 Experience
- Conclusion

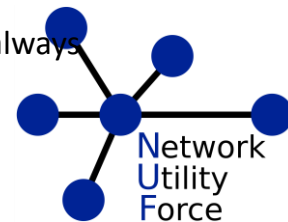


Core Values and Beliefs

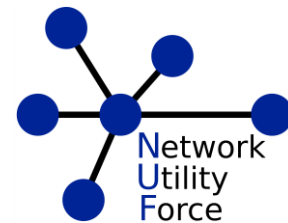
Network Utility Force believes that, above and beyond our experience, it's our values that drive our success in both business and life. As such, NUF has adopted a system of Core Values & Beliefs that we live by:

- We respect the individual, and believe that individuals who are treated with respect and given responsibility respond by giving their best.
- We require complete honesty and integrity in everything we do.
- We make commitments with care, and then live up to them. In all things, we do what we say we are going to do.
- Work is an important part of life, and it should be fun. Being a good business person does not mean being stuffy and boring.
- We are frugal. We guard and conserve the company's resources with at least the same vigilance that we would use to guard and conserve our own personal resources.
- We insist on giving our best effort in everything we undertake.
- Furthermore, we see a huge difference between "good mistakes" (best effort, bad result) and "bad mistakes" (sloppiness or lack of effort).
- Clarity in understanding our mission, our goals, and what we expect from each other is critical to our success.
- We are believers in the Golden Rule. In all our dealings we will strive to be friendly and courteous, as well as fair and compassionate.
- We feel a sense of urgency on any matters related to our customers. We own problems and we are always responsive. We are customer driven.

Permission to use these values granted by their creator, Charles Brewer of MindSpring.

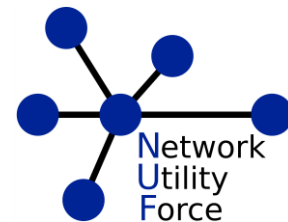


INTRODUCTION



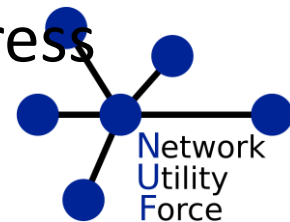
NUF Tenets

- Our reputation is **EVERYTHING**, without it, we are worthless
 - We will ALWAYS advise our clients on what we believe is the right answer for them without exception
- Vendor neutral – The right tool for the right job
 - Cisco, Juniper, Brocade, HP, Huawei, A10, Dell (Force10), Extreme, Vyatta, ADVA, Arista, Alcatel, etc., etc
- No hardware sales, 100% professional services
- No geographical boundaries, we go where we are needed

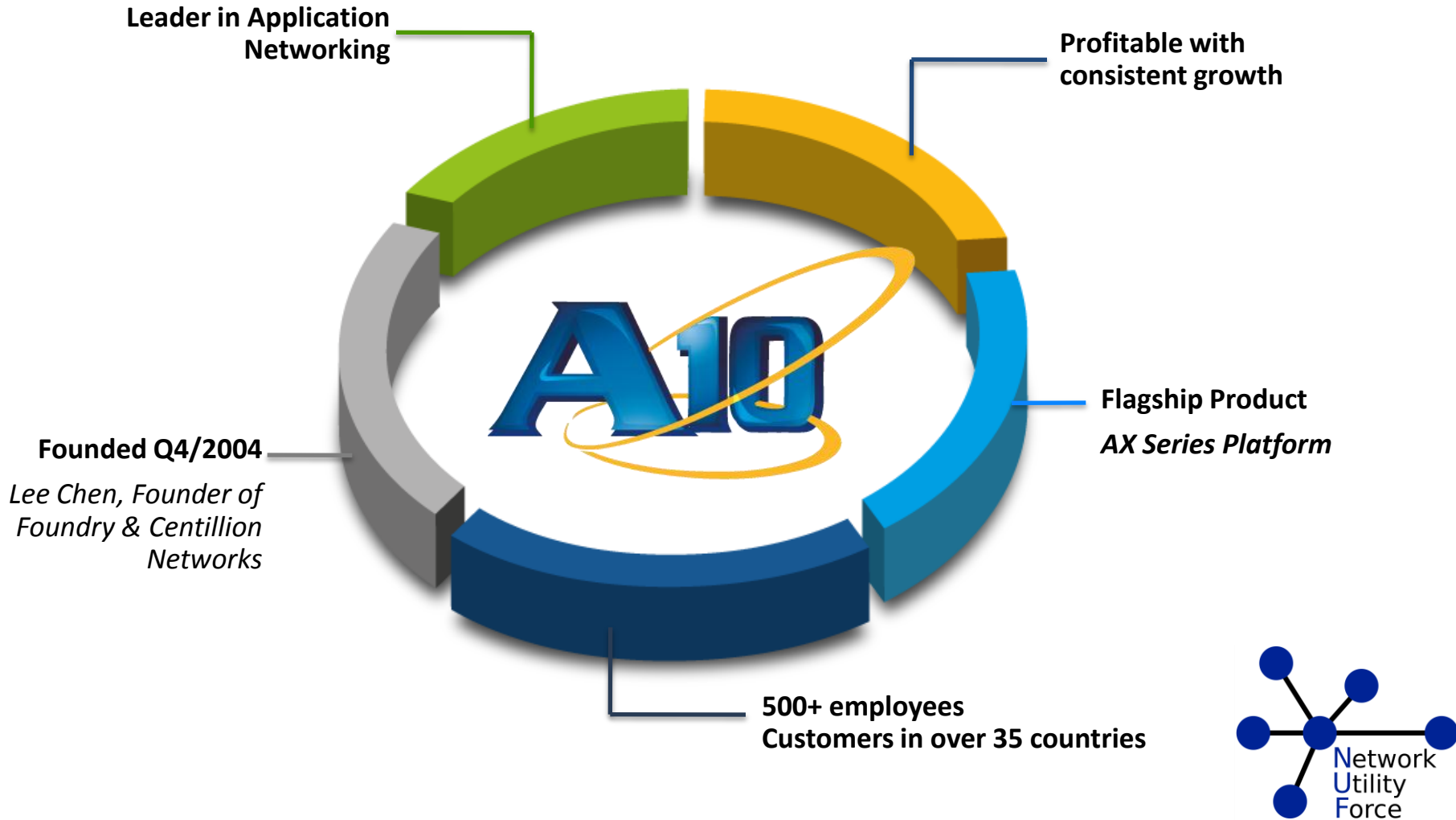


Who is NUF?

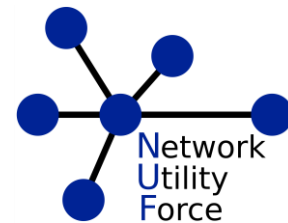
- Founded in December of 2011
- 6 principal consultants/owners
- Numerous specialist contractors
- Example experience
 - Recently highlighted by local news for deployment of community wifi network sponsored by Google
 - Consulting for Ethiopia TLD
 - Netrail, MindSpring, Comcast, Internap, numerous small service providers
 - ARIN, NANOG, IETF participation
- IPv6 training, architecture, deployment and address management



Overview

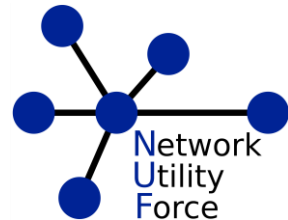


PROBLEM STATEMENT



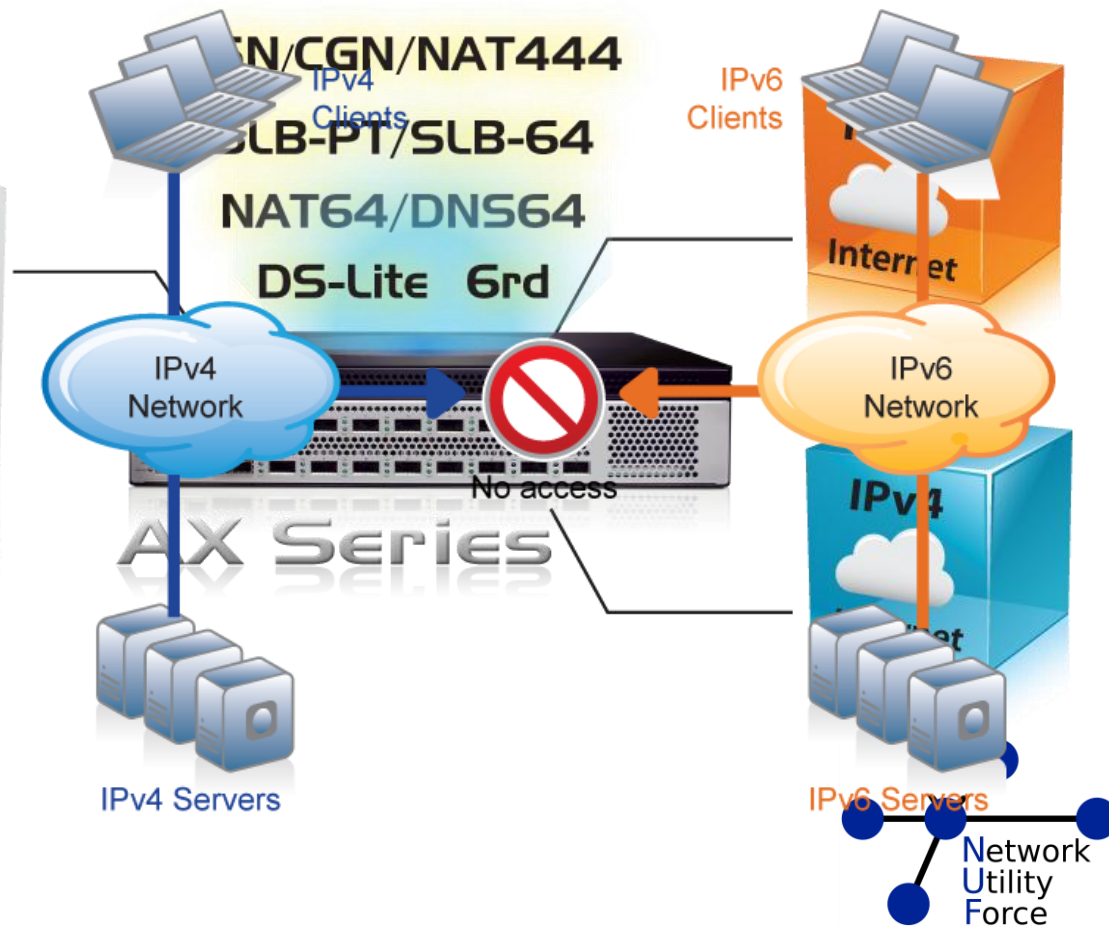
Problem Statement

I have run out of IPv4 addresses and need to find a way to provide Internet access to my clients. My core network supports IPv6, but there are many IPv4-only resources my clients need to reach. I want to go with IPv6 because I want to future-proof my internal network, however I understand I need connectivity to IPv4 resources for quite some time.



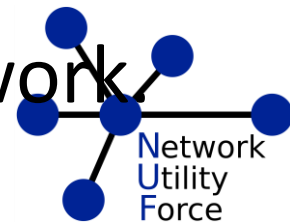
IPv4 Exhaustion and IPv6 Migration Solutions

- No standard compatibility
- Different requirements
 - Home
 - Enterprise
 - Service Provider
- “IPv4 Legacy Networks”



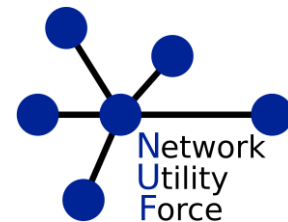
Transition Mechanisms

- Dual Stack – All network links and hosts have both IPv4 and IPv6 addressing, all traffic is native to its protocol
- Dual Stack – Lite (DS-Lite) – Allow distribution network (including CPE) to have ONLY IPv6 addressing.
- NAT64 – Translate IPv6 into IPv4 and vice versa.
- 6rd – Carry IPv6 across an IPv4-only network

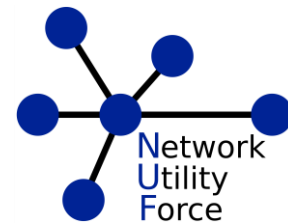


Protocol Translations

- May prefer to use IPv6-IPv4 protocol translation for:
 - new kinds of Internet devices (e.g., cell phones, cars, appliances)
 - benefits of shedding IPv4 stack (e.g., serverless autoconfig)
- Simple extension to NAT techniques, to translate header format as well as addresses
 - IPv6 nodes behind a translator get full IPv6 functionality when talking to other IPv6 nodes located anywhere
 - Get the normal (i.e., degraded) NAT functionality when talking to IPv4 devices
 - Drawback : minimal gain over IPv4/IPv4 NAT approach
 - Drawback : no support for legacy IPv4-only devices

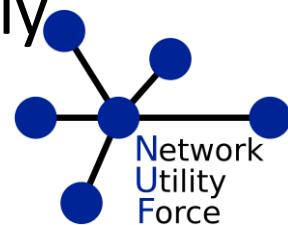


NAT64 CONCEPTS



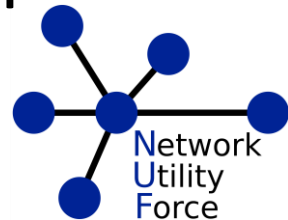
NAT64 and DNS64

- Provides stateful translation between IPv6 and IPv4 traffic when that traffic is initiated by an IPv6-only node
 - NAT64 translates IPv6 and IPv4 traffic
 - DNS64 maps IPv6-only address record (AAAA) DNS queries to IPv4 address record (A) queries
- Makes it possible for IPv6-only nodes to initiate communications with IPv4-only nodes with no changes to the IPv6-only node and the IPv4-only node



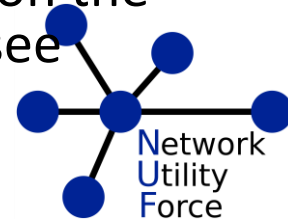
NAT64

- No host or CPE support necessary
- No IPv4 address at all required on the CPE or host to access IPv4 resources (compare with DS-Lite which requires RFC1918 addresses)
- Host gets only an IPv6 address
- NAT44 issues for IPv4 traffic still apply including session start from behind the NAT to establish state and ALGs are necessary for many protocols



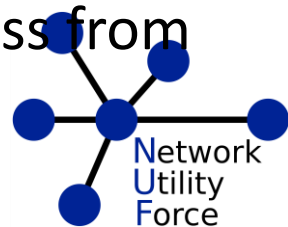
NAT64 cont'd

- Workstation (which has only IPv6) requests communication with a IPv4-only site (www.example.com)
 - It asks for a AAAA record since it is IPv6 only
- DNS server (DNS64) first tries a AAAA query. If one exists, it is passed to the client and the client communicates with the site via IPv6. If no AAAA record exists, the DNS64 functionality will translate an A record into a AAAA.
 - To translate, the DNS64 server must know the prefix in use in the network for NAT64
 - Can be assigned by administrator
 - Can use well-known prefix 64:ff9b::/96
 - When using a /96, simply concatenate the IPv4 address on the end (more complex rules available for shorter prefixes, see RFC6052)

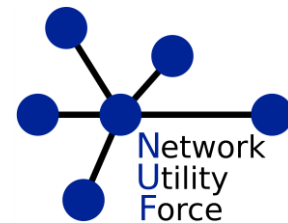


NAT64 cont'd

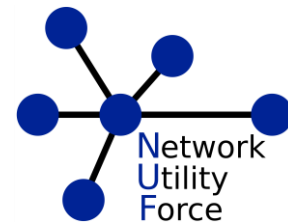
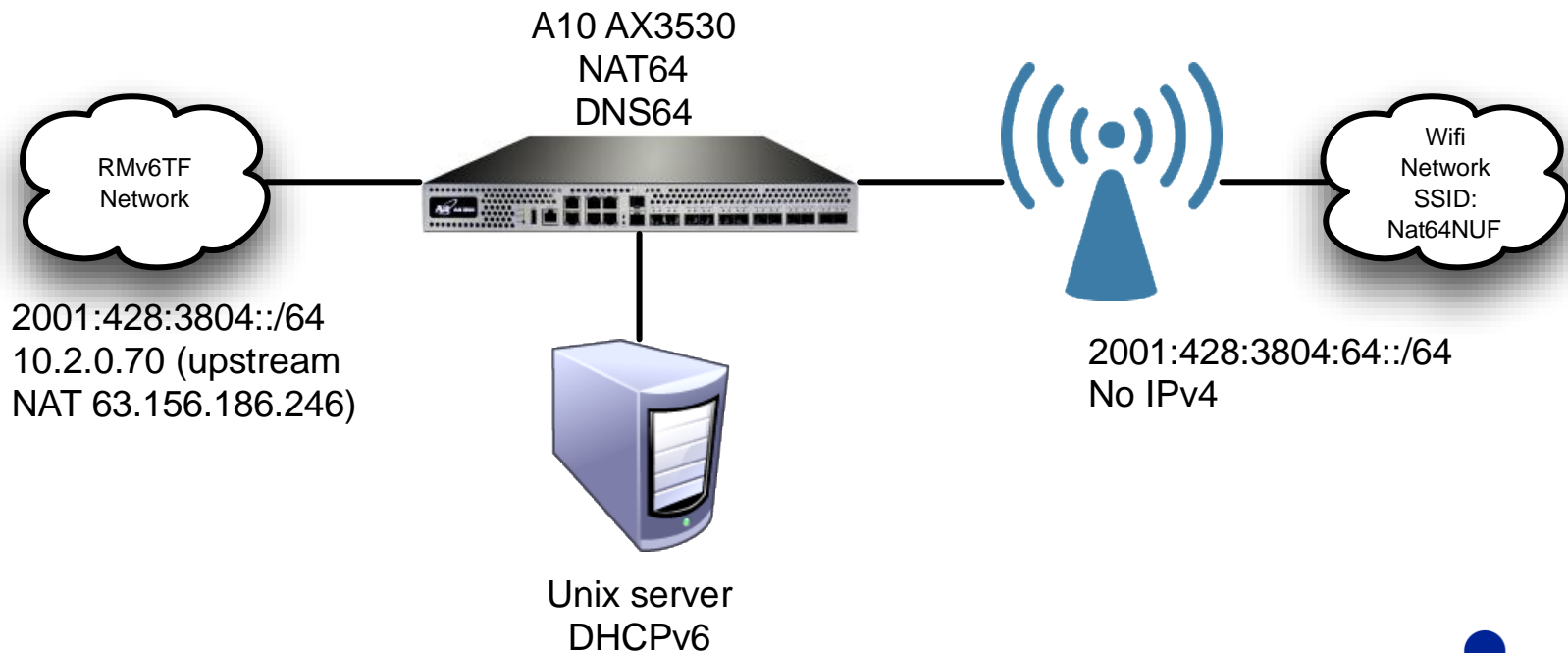
- The workstation, having received a AAAA for www.example.com initiates a TCP session with that IPv6 address
- The NAT64 device has the IPv6 prefix just discussed routed to it (could be OSPF, BGP, etc.)
- The NAT64 device recognizes that this address is an encapsulated IPv4 address
 - Adds NAT state if necessary
 - Strips the prefix to find the IPv4 destination address
 - Translates other parts of the header
 - Sends packet to IPv4 host using a source IPv4 address from a local pool



DEMO SETUP



Demo Topology



A10 Config Bits

Inside config

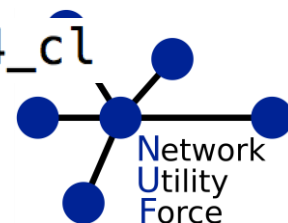
```
interface ve 64
  ipv6 address 2001:428:3804:64::1/64
  ipv6 nat inside
  ipv6 ndisc router-advertisement other-configuration-flag enable
  ipv6 ndisc router-advertisement prefix 2001:428:3804:64::/64
  ipv6 ndisc router-advertisement enable
```

Configure NAT pools

```
ip nat pool dns64pool 10.2.0.71 10.2.0.71 netmask /24 gateway 10.2.0.152
ip nat pool nat64pool2 10.2.0.72 10.2.0.72 netmask /24 lsu
```

Configure NAT using the well-known prefix

```
nat64 prefix well-known inside source class-list nat64_cl
```



A10 Config Bits

```
lb server localdns-rs 8.8.8.8
  port 53  udp
lb server localdns-rs1 10.2.0.10
  weight 100
  port 53  udp

lb service-group dns53 udp
  member localdns-rs1:53 priority 10
  member localdns-rs:53

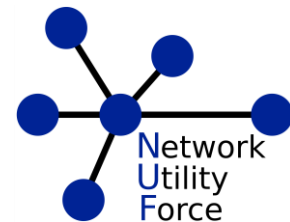
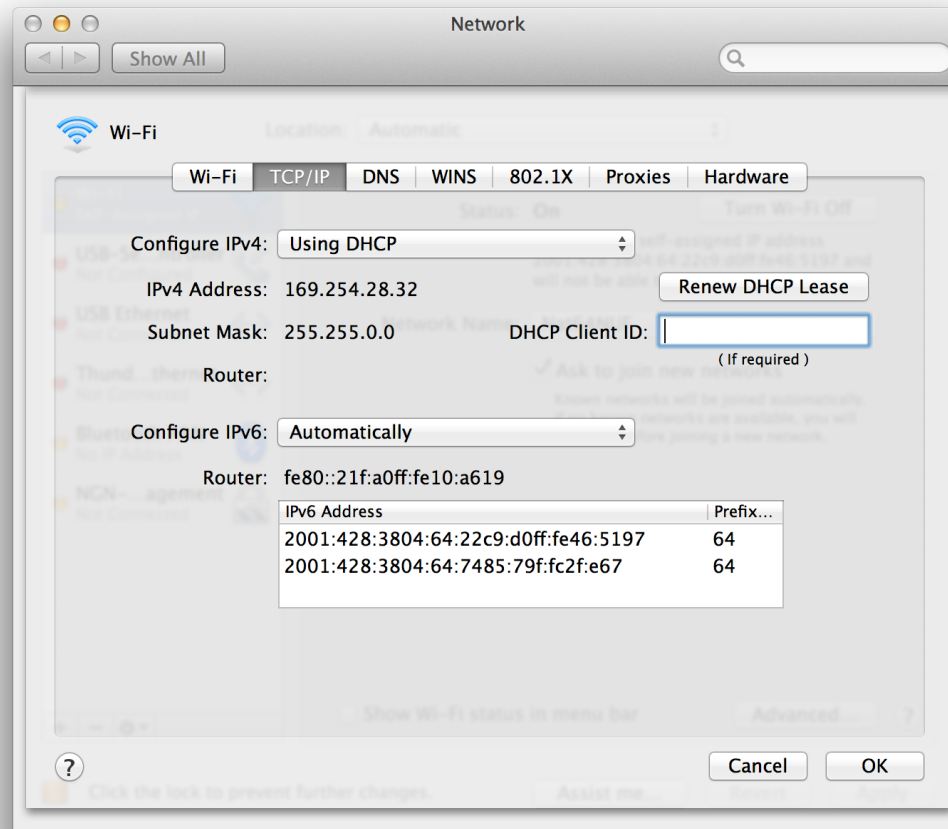
lb template dns dns64-temp
  dns64

lb virtual-server dns_vs 2001:428:3804:64::64
  port 53  dns-udp
  source-nat pool dns64pool
  service-group dns53
  template dns dns64-temp
```

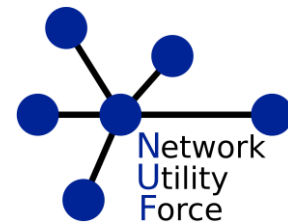


Demo Setup

Join the network using SSID: Nat64NUF

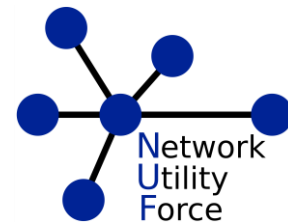


NAT64 EXPERIENCE



What works?

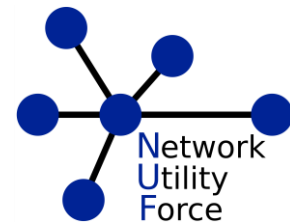
- Nearly everything
 - A10's implementation was solid, no signs of bugs or performance problems
 - Operating systems: Windows, MacOS, Linux
 - Nearly all classic IP protocols: POP, IMAP, SSH, Telnet, Standard web stuff, SMTP
 - Many chat protocols: Google talk, FaceBook chat
 - Entertainment: Pandora web client, Netflix, YouTube, Hulu



IOS is Weird



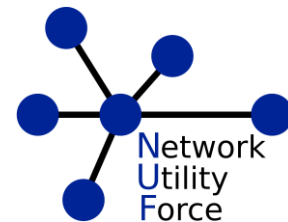
We appear to be connected



IOS is Weird



Seems to be stuck in the process of connecting...



IOS is Weird

AT&T 4G 13:18 69%

Test your IPv6.

ipv6.test-ipv6.com/ Google

Test IPv6 FAQ Mirrors

Test your IPv6 connectivity.

Summary Tests Run Share Results / Contact Other IPv6 Sites

- Your IPv4 address on the public Internet appears to be 63.156.1...
- Your IPv6 address on the public Internet appears to be 2001:428:3804:64:e466:ace:9af1:145e
- Your Internet Service Provider (ISP) appears to be ASN-QWEST NOVARTIS-DMZ-US
- Since you have IPv6, we are including a tab that shows how well reach other IPv6 sites. [\[more info\]](#)
- Good news!** Your current configuration will continue to work as enable IPv6. [\[more info\]](#)
- Your DNS server (possibly run by your ISP) appears to have IPv access.

Your readiness score

10/10 for your IPv6 stability and readiness, when publishers are IPv6 only

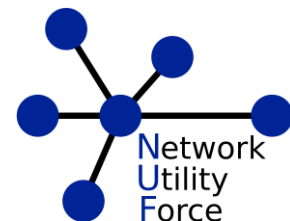
Click to see [test data](#)

(Updated server side IPv6 readiness stats)

Like 10,822 people like this. Tweet 5

Copyright (C) 2010, 2012 Jason Fesler. All rights reserved. -- r897
[Mirrors](#) | [Mission](#) | [Source](#) | [Email](#) - [Attributions](#) | [Debug](#)
This is a mirror of test-ipv6.com. The views expressed here may or may not reflect the views of the

But look, it worked!

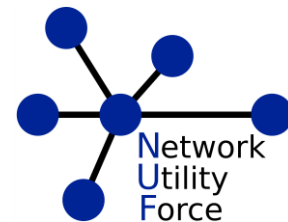


Whois broken??

| Filter: | | | | | | Expression... | Clear | Apply | Save |
|--|-------------------|---|---|----------|---|--|-------|-------|------|
| No. | Time | Source | Destination | Protocol | Length | Info | | | |
| 97 | 7.834513000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 199.212.0.46? Tell 169.254.198.17 | | | |
| 98 | 7.853140000 | 2001:428:3804:64:22c9:d0ff:2001:428:3804:64::64 | DNS | 94 | Standard query 0x683c A whois.arin.net | | | | |
| 99 | 7.853189000 | 2001:428:3804:64:22c9:d0ff:2001:428:3804:64::64 | DNS | 94 | Standard query 0x683c AAAA whois.arin.net | | | | |
| 100 | 7.865513000 | 2001:428:3804:64::64 | 2001:428:3804:64:22c9:d0ff:2001:428:3804:64::64 | DNS | 262 Standard query response 0xc40 AAAA 2001:500:31::46 AAAA 2001:500:31::47 AAAA 2001:500:31::48 AAAA 2001:500:31::49 | | | | |
| 101 | 7.892392000 | 2001:428:3804:64::64 | 2001:428:3804:64:22c9:d0ff:2001:428:3804:64::64 | DNS | 190 Standard query response 0x683c A 199.212.0.47 A 199.212.0.48 A 199.71.0.46 A 199.71.0.47 | | | | |
| 102 | 7.893080000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 199.212.0.46? Tell 169.254.198.17 | | | |
| 103 | 7.905681000 | Apple_74:bc:a3 | Broadcast | ARP | 42 | Who has 108.160.163.42? Tell 169.254.167.30 | | | |
| 104 | 8.006691000 | 169.254.198.17 | 169.254.255.255 | BJNP | 58 | Scanner Command: Unknown code (2) | | | |
| 105 | 8.006746000 | 169.254.198.17 | 224.0.0.1 | BJNP | 58 | Scanner Command: Unknown code (2) | | | |
| 106 | 8.109779000 | fe80::21f:a0ff:fe10:a619 | ff02::1 | ICMPv6 | 110 | Router Advertisement from 00:1f:a0:10:a6:19 | | | |
| 107 | 8.375217000 | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | HTTP | 396 | POST http://65.54.61.209/gateway/gateway.dll?Action=poll&SessionID=924096825.741617309 | | | |
| 108 | 8.412027000 | 64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | HTTP | 355 | HTTP/1.1 200 OK | | | |
| 109 | 8.412120000 | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | TCP | 86 | 65066 > http [ACK] Seq=1552 Ack=1343 Win=65535 Len=0 TSval=941107121 TSecr=8241692 | | | |
| 110 | 8.739428000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 199.212.0.46? Tell 169.254.198.17 | | | |
| 111 | 8.941478000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 199.212.0.46? Tell 169.254.198.17 | | | |
| 112 | 8.941479000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 199.212.0.46? Tell 169.254.198.17 | | | |
| 113 | 9.134114000 | Apple_74:bc:a3 | Broadcast | ARP | 42 | Who has 108.160.163.42? Tell 169.254.167.30 | | | |
| 114 | 9.338584000 | fe80::21f:a0ff:fe10:a619 | ff02::1 | ICMPv6 | 110 | Router Advertisement from 00:1f:a0:10:a6:19 | | | |
| 115 | 9.344155000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 64.12.24.161? Tell 169.254.198.17 | | | |
| 116 | 9.344156000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 64.12.24.27? Tell 169.254.198.17 | | | |
| 117 | 9.344157000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 64.12.104.25? Tell 169.254.198.17 | | | |
| 118 | 9.948587000 | Apple_46:51:97 | Broadcast | ARP | 42 | Who has 199.212.0.46? Tell 169.254.198.17 | | | |
| 119 | 10.098981000 | Apple_74:bc:a3 | Broadcast | ARP | 42 | Who has 108.160.163.42? Tell 169.254.167.30 | | | |
| 120 | 10.375736000 | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | HTTP | 397 | POST http://65.54.61.209/gateway/gateway.dll?Action=poll&SessionID=924096825.1623318622 | | | |
| 121 | 10.412538000 | 64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | HTTP | 354 | HTTP/1.1 200 OK | | | |
| 122 | 10.412636000 | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | TCP | 86 | 65066 > http [ACK] Seq=1863 Ack=1611 Win=65535 Len=0 TSval=941109111 TSecr=8241892 | | | |
| 123 | 10.736363000 | fe80::22c9:d0ff:fe46:5197 | fe80::21f:a0ff:fe10:a619 | ICMPv6 | 86 | Neighbor Solicitation for fe80::21f:a0ff:fe10:a619 from 20:c9:d0:46:51:97 | | | |
| 124 | 10.739688000 | fe80::21f:a0ff:fe10:a619 | fe80::22c9:d0ff:fe46:5197 | ICMPv6 | 86 | Neighbor Advertisement fe80::21f:a0ff:fe10:a619 (rtr, sol, ovr) is at 00:1f:a0:10:a6:19 | | | |
| 125 | 10.772237000 | fe80::21f:a0ff:fe10:a619 | ff02::1 | ICMPv6 | 110 | Router Advertisement from 00:1f:a0:10:a6:19 | | | |
| 126 | 11.386696000 | Apple_74:bc:a3 | Broadcast | ARP | 42 | Who has 108.160.163.42? Tell 169.254.167.30 | | | |
| 127 | 12.375691000 | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | HTTP | 396 | POST http://65.54.61.209/gateway/gateway.dll?Action=poll&SessionID=924096825.675580934 | | | |
| 128 | 12.412366000 | 64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | HTTP | 354 | HTTP/1.1 200 OK | | | |
| 129 | 12.412495000 | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | 2001:428:3804:64:35ed:bed3:64:ff9b::4137:40fe | TCP | 86 | 65066 > http [ACK] Seq=2173 Ack=1879 Win=65535 Len=0 TSval=941111098 TSecr=8242092 | | | |
| 130 | 12.534010000 | 2001:428:3804:64:35ed:bed3:64:ff9b::42dc:9763 | 2001:428:3804:64:35ed:bed3:64:ff9b::42dc:9763 | TCP | 187 | [TCP segment of a reassembled PDU] | | | |
| 131 | 12.564952000 | 64:ff9b::42dc:9763 | 2001:428:3804:64:35ed:bed3:64:ff9b::42dc:9763 | TCP | 432 | [TCP segment of a reassembled PDU] | | | |
| 132 | 12.565097000 | 2001:428:3804:64:35ed:bed3:64:ff9b::42dc:9763 | 2001:428:3804:64:35ed:bed3:64:ff9b::42dc:9763 | TCP | 86 | 65066 > http client [ACK] Seq=102 Ack=247 Win=8170 Len=0 TSval=941111250 TSecr=824219111 | | | |
| ▸ Frame 97: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0 | | | | | | | | | |
| ▸ Ethernet II, Src: Apple_46:51:97 (20:c9:d0:46:51:97), Dst: Broadcast (ff:ff:ff:ff:ff:ff) | | | | | | | | | |
| ▸ Address Resolution Protocol (request) | | | | | | | | | |
| | | | | | | | | | |
| 0000 | ff ff ff ff ff ff | 20 c9 d0 46 51 97 08 06 00 01 |FQ..... | | | | | | |
| 0010 | 08 00 06 04 00 01 | 20 c9 d0 46 51 97 a9 fe c6 11 |FQ..... | | | | | | |

Whois

- A few whois attempts later, everything was working
- Does it depend on if the AAAA or A record comes back first?



FTP before ALG

Wireshark
erikm@acer-lappy: ~

Capturing from eth0 [Wireshark 1.8.2]

Filter: `tcp.port==20 or tcp.port==21` Expression... Clear Apply Save Filter

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-----------------------|-----------------------|----------|--------|---|
| 30 | 15.38574100 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38951 > ftp [ACK] Seq=23 Ack=964 Win=16768 Len= |
| 31 | 15.38608400 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | FTP | 352 | Response: 230- If you need to link to a publi |
| 32 | 15.38614100 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38951 > ftp [ACK] Seq=23 Ack=1230 Win=18048 Len |
| 33 | 15.38624000 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 92 | Request: SYST |
| 34 | 15.44289800 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | FTP | 105 | Response: 215 UNIX Type: L8 |
| 35 | 15.48218300 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38951 > ftp [ACK] Seq=29 Ack=1249 Win=18048 Len |
| 38 | 18.57449500 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | Request: EPRT 2 2001:470:e05b:64:45aa:baca:c0 |
| 39 | 18.83017300 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | [TCP Retransmission] Request: EPRT 2 2001:470 |
| 40 | 19.34218200 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | [TCP Retransmission] Request: EPRT 2 2001:470 |
| 42 | 20.37039300 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | [TCP Retransmission] Request: EPRT 2 2001:470 |
| 44 | 22.42639400 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | [TCP Retransmission] Request: EPRT 2 2001:470 |
| 48 | 26.53839900 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | [TCP Retransmission] Request: EPRT 2 2001:470 |
| 58 | 34.74621900 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | [TCP Retransmission] Request: EPRT 2 2001:470 |

Frame 38: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
Ethernet II, Src: QuantaCo 5d:4f:b9 (00:26:9e:5d:4f:b9), Dst: A10Netwo 10:a6:19 (00:1f:a0:10:a6:19)
Internet Protocol Version 6, Src: 2001:470:e05b:64:45aa:baca:c07:3342 (2001:470:e05b:64:45aa:baca:c07:3342), Dst: 64:ff9b::3
Transmission Control Protocol, Src Port: 38951 (38951), Dst Port: ftp (21), Seq: 29, Ack: 1249, Len: 52
File Transfer Protocol (FTP)
EPRT |2|2001:470:e05b:64:45aa:baca:c07:3342|53408|\r\n
Request command: EPRT
Request arg: |2|2001:470:e05b:64:45aa:baca:c07:3342|53408|

0000 00 1f a0 10 a6 19 00 26 9e 5d 4f b9 86 dd 60 00&.]0...
0010 00 00 00 54 06 ff 20 01 04 70 e0 5b 00 64 45 aa ...T...p.[.dE.
0020 ba ca 0c 07 33 42 00 64 ff 9b 00 00 00 00 00 003B.d
0030 00 00 3f f5 d7 38 98 27 00 15 fb 75 12 d6 c0 8a ...?.8.'...u...
0040 7e 15 80 18 00 8d 5c 77 00 00 01 01 08 0a 00 20 ~.....\w
0050 42 5d a8 11 fa 12 45 50 52 54 20 7c 32 7c 32 30 R1 FP RT |2|20

FTP with ALG, Outside

The image shows a Wireshark 1.8.2 capture of an FTP session. The left pane displays the packet list, and the right pane shows the packet details and raw data. The filter is set to 'tcp.port==20 or tcp.port==21'. The session includes a login, directory listing, and a file transfer.

Packet List:

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-------------|-----------------------|-----------------------|----------|--------|---|
| 3189 | 1628.542263 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38964 > ftp [ACK] Seq=24 Ack=1230 Win=18048 Len=0 |
| 3190 | 1628.542448 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 92 | Request: SYST |
| 3191 | 1628.598476 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | TCP | 86 | ftp > 38964 [ACK] Seq=1230 Ack=30 Win=15360 Len=0 |
| 3192 | 1628.599336 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | FTP | 105 | Response: 215 UNIX Type: L8 |
| 3193 | 1628.638185 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38964 > ftp [ACK] Seq=30 Ack=1249 Win=18048 Len=0 |
| 3196 | 1631.445816 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 95 | Request: CWD pub |
| 3197 | 1631.503063 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | FTP | 123 | Response: 250 Directory successfully changed. |
| 3198 | 1631.503187 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38964 > ftp [ACK] Seq=39 Ack=1286 Win=18048 Len=0 |
| 3200 | 1632.415227 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 138 | Request: EPRT 2 2001:470:e05b:64:45aa:baca:c07 |
| 3201 | 1632.472917 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | FTP | 137 | Response: 200 PORT command successful. Consider |
| 3202 | 1632.473028 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | TCP | 86 | 38964 > ftp [ACK] Seq=91 Ack=1337 Win=18048 Len=0 |
| 3203 | 1632.473144 | 2001:470:e05b:64:45aa | 64:ff9b::3ff5:d738 | FTP | 92 | Request: LIST |
| 3204 | 1632.531160 | 64:ff9b::3ff5:d738 | 2001:470:e05b:64:45aa | FTP | 125 | Response: 150 Here comes the directory listing. |

Packet Details:

- Frame 3200: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
- Ethernet II, Src: QuantaCo 5d:4f:b9 (00:26:9e:5d:4f:b9), Dst: A10Netwo 10:a6:19 (00:1f:a0:10:a6:19)
- Internet Protocol Version 6, Src: 2001:470:e05b:64:45aa:baca:c07:3342 (2001:470:e05b:64:45aa:baca:c07:3342), Dst: 64:ff9b::3ff5:d738
- Transmission Control Protocol, Src Port: 38964 (38964), Dst Port: ftp (21), Seq: 39, Ack: 1286, Len: 52
- File Transfer Protocol (FTP)
 - EPRT |2|2001:470:e05b:64:45aa:baca:c07:3342|35010|\r\n
 - Request command: EPRT
 - Request arg: |2|2001:470:e05b:64:45aa:baca:c07:3342|35010|

Raw Data:

```
0000 00 1f a0 10 a6 19 00 26 9e 5d 4f b9 86 dd 60 00 .....&.]0...
0010 00 00 00 54 06 ff 20 01 04 70 e0 5b 00 64 45 aa ...T...p.[.dE.
0020 ba ca 0c 07 33 42 00 64 ff 9b 00 00 00 00 00 00 ....3B.d .....
0030 00 00 3f f5 d7 38 98 34 00 15 a8 e1 53 eb 9f 5f ...?.8.4 ....S.
0040 77 14 80 18 00 8d 5c 77 00 00 01 01 08 0a 00 26 w.....w .....&
0050 6a 61 a8 2a a2 d0 45 50 52 54 20 7c 32 7c 32 30 ia * EP RT 12120
```


FTP with ALG, Inside

| Filter: tcp.port==20 tcp.port==21 | | Expression... | | Clear | Apply | Save |
|---|--------------|---------------|---------------|----------|--------|--|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 98 | 25.413183000 | 192.168.100.9 | 63.245.215.56 | TCP | 66 | 38964 > ftp [ACK] Seq=24 Ack=1230 Win=18048 Len=0 TSval=2516633 TSecr=2821363519 |
| 99 | 25.413361000 | 192.168.100.9 | 63.245.215.56 | FTP | 72 | Request: SYST |
| 100 | 25.469344000 | 63.245.215.56 | 192.168.100.9 | TCP | 66 | ftp > 38964 [ACK] Seq=1230 Ack=30 Win=15360 Len=0 TSval=2821363575 TSecr=2516633 |
| 101 | 25.470243000 | 63.245.215.56 | 192.168.100.9 | FTP | 85 | Response: 215 UNIX Type: L8 |
| 102 | 25.509151000 | 192.168.100.9 | 63.245.215.56 | TCP | 66 | 38964 > ftp [ACK] Seq=30 Ack=1249 Win=18048 Len=0 TSval=2516657 TSecr=2821363576 |
| 109 | 28.316876000 | 192.168.100.9 | 63.245.215.56 | FTP | 75 | Request: CWD pub |
| 110 | 28.373840000 | 63.245.215.56 | 192.168.100.9 | FTP | 103 | Response: 250 Directory successfully changed. |
| 111 | 28.374200000 | 192.168.100.9 | 63.245.215.56 | TCP | 66 | 38964 > ftp [ACK] Seq=39 Ack=1286 Win=18048 Len=0 TSval=2517373 TSecr=2821366480 |
| 117 | 29.286307000 | 192.168.100.9 | 63.245.215.56 | FTP | 94 | Request: PORT 192,168,100,9,136,194 |
| 118 | 29.343745000 | 63.245.215.56 | 192.168.100.9 | FTP | 117 | Response: 200 PORT command successful. Consider using PASV. |
| 119 | 29.343993000 | 192.168.100.9 | 63.245.215.56 | TCP | 66 | 38964 > ftp [ACK] Seq=67 Ack=1337 Win=18048 Len=0 TSval=2517615 TSecr=2821367449 |
| 120 | 29.344096000 | 192.168.100.9 | 63.245.215.56 | FTP | 72 | Request: LIST |
| 122 | 29.402048000 | 63.245.215.56 | 192.168.100.9 | FTP | 105 | Response: 150 Here comes the directory listing. |
| 124 | 29.403843000 | 63.245.215.56 | 192.168.100.9 | FTP | 90 | Response: 226 Directory send OK. |
| ▶ Frame 117: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0 | | | | | | |
| ▶ Ethernet II, Src: A10Netwo_10:a6:19 (00:1f:a0:10:a6:19), Dst: D-Link_ff:3d:4b (00:26:5a:ff:3d:4b) | | | | | | |
| ▶ Internet Protocol Version 4, Src: 192.168.100.9 (192.168.100.9), Dst: 63.245.215.56 (63.245.215.56) | | | | | | |
| ▶ Transmission Control Protocol, Src Port: 38964 (38964), Dst Port: ftp (21), Seq: 39, Ack: 1286, Len: 28 | | | | | | |
| ▼ File Transfer Protocol (FTP) | | | | | | |
| ▼ PORT 192,168,100,9,136,194\r\n | | | | | | |
| Request command: PORT | | | | | | |
| Request arg: 192,168,100,9,136,194 | | | | | | |
| Active IP address: 192.168.100.9 (192.168.100.9) | | | | | | |
| Active port: 35010 | | | | | | |

0020

d7 38 98 34 00 15 a8 e1 53 eb 9f 5f 77 14 80 18

.8.4.... S...w...

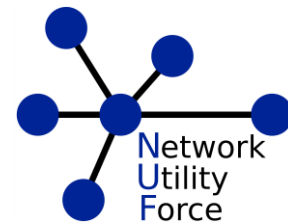
0030

00 8d 20 3e 00 00 01 01 08 0a 00 26 6a 61 a8 2a

..>....&ia.*

What doesn't work?

- All Android based devices we tried did not function
 - We suspect that Android performs an IPv4 connectivity specific test and reject networks that doesn't have IPv4
 - We also suspect that Android will not consider DNS over IPv6 transport



AOL Instant Messenger

Won't connect to the server, why?

The image shows a Wireshark network traffic capture. The top pane displays a list of packets filtered by 'tcp.port==5190'. The middle pane shows the details of the selected packet (No. 117), which is an AIM Signon, Logon Reply. The bottom pane shows the raw packet data in hexadecimal and ASCII. A status window at the bottom left indicates 'erikmuller disconnected' and 'Unable to connect to BOS server: Network is unreachable'.

Wireshark
erikm@acer-lappy: ~
Capturing from eth0 [Wireshark 1.8.2]

Filter: **tcp.port==5190**

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-----------------------|-----------------------|------------|--------|--|
| 105 | 73.43477900 | 2001:470:e05b:64:45aa | 64:ff9b::400c:ac61 | AIM | 84 | New Connection |
| 106 | 73.44989200 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | AIM | 84 | New Connection |
| 107 | 73.44994700 | 2001:470:e05b:64:45aa | 64:ff9b::400c:ac61 | TCP | 74 | 36168 > aol [ACK] Seq=11 Ack=11 Win=14400 Len=0 |
| 108 | 73.44996400 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | TCP | 74 | aol > 36168 [ACK] Seq=11 Ack=11 Win=16384 Len=0 |
| 109 | 73.44998600 | 2001:470:e05b:64:45aa | 64:ff9b::400c:ac61 | AIM Signon | 113 | AIM Signon, Sign-on Username: erikmuller |
| 110 | 73.46510700 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | TCP | 74 | aol > 36168 [ACK] Seq=11 Ack=50 Win=16384 Len=0 |
| 111 | 73.47346700 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | AIM Signon | 102 | AIM Signon, Sign-on Reply |
| 112 | 73.47385700 | 2001:470:e05b:64:45aa | 64:ff9b::400c:ac61 | AIM Signon | 201 | AIM Signon, Logon |
| 113 | 73.48907400 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | TCP | 74 | aol > 36168 [ACK] Seq=39 Ack=177 Win=16384 Len=0 |
| 114 | 73.48986100 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | AIM | 512 | Close Connection |
| 115 | 73.50628700 | 2001:470:e05b:64:45aa | 64:ff9b::400c:ac61 | TCP | 74 | 36168 > aol [FIN, ACK] Seq=177 Ack=477 Win=15008 Len=0 |
| 116 | 73.52118300 | 64:ff9b::400c:ac61 | 2001:470:e05b:64:45aa | TCP | 74 | aol > 36168 [FIN, ACK] Seq=477 Ack=178 Win=16384 Len=0 |
| 117 | 73.52122500 | 2001:470:e05b:64:45aa | 64:ff9b::400c:ac61 | TCP | 74 | 36168 > aol [ACK] Seq=178 Ack=478 Win=15008 Len=0 |

Internet Protocol Version 6, Src: 64:ff9b::400c:c248 (64:ff9b::400c:c248), Dst: 2001:470:e05b:64:45aa:baca:c07:3342 (2001:470:e05b:64:45aa:baca:c07:3342)

Transmission Control Protocol, Src Port: aol (5190), Dst Port: 60644 (60644), Seq: 39, Ack: 177, Len: 438

AOL Instant Messenger

AIM Signon, Logon Reply

- TLV: Unknown
- TLV: Screen name
- TLV: BOS server string
 - Value ID: BOS server string (0x0005)
 - Length: 17
 - Value: 64.12.24.166:5190
- TLV: Authorization cookie

erikmuller disconnected
Unable to connect to BOS server:
Network is unreachable

Modify Account Reconnect

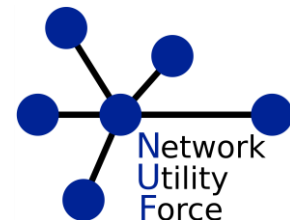
Available

erikm@acer-lappy:~\$

Text item (text), 17 bytes

Packets: 185 Displayed: 48 Marked: 0

Profile: Default



Skype

Also won't connect to the Skype network

The image shows a terminal window on the left with DHCP logs for 'erikm@acer-lappy: ~'. The logs indicate the system is bound to *:546, listening on Socket/eth0, and sending on Socket/eth0. It shows a PRC: Requesting information (INIT), XMT: Forming Info-Request, 0 ms elapsed, XMT: Info-Request on eth0, interval 1030ms, RCV: Reply message on eth0 from fe80::cad7:19ff:fe66:3ee6, and PRC: Done. Below the logs is the Skype 4.1 for Linux login screen, which displays 'Skype can't connect.' and fields for Skype Name, Password, and a Microsoft account link. The right side of the image shows a Wireshark 1.8.2 packet capture from eth0, filtered for 'Intp and icmpv6'. The packet list shows several DNS queries and responses. The packet details pane shows a query for 'dsn7.d.skype.net: type A, class IN' and the corresponding answer with IP addresses. The packet bytes pane shows the raw data for the selected packet.

Wireshark 1.8.2

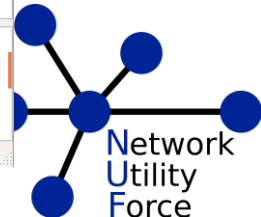
Filter: **Intp and icmpv6**

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|-----------------------|-----------------------|----------|--------|-------------------------|
| 13 | 16.16691600 | 2001:470:e05b:64:45aa | 2001:470:e05b:64::64 | DNS | 96 | Standard query 0x47fe |
| 14 | 16.16902100 | 2001:470:e05b:64::64 | 2001:470:e05b:64:45aa | DNS | 260 | Standard query response |
| 87 | 94.34948200 | 2001:470:e05b:64:45aa | 2001:470:e05b:64::64 | DNS | 97 | Standard query 0x5fa5 |
| 88 | 94.35184100 | 2001:470:e05b:64::64 | 2001:470:e05b:64:45aa | DNS | 262 | Standard query response |
| 170 | 174.56777400 | 2001:470:e05b:64:45aa | 2001:470:e05b:64::64 | DNS | 97 | Standard query 0xd945 |
| 171 | 174.56925800 | 2001:470:e05b:64::64 | 2001:470:e05b:64:45aa | DNS | 559 | Standard query response |

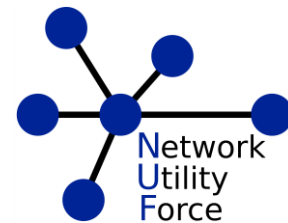
Authority RRs: 0
Additional RRs: 0
▼ Queries
▶ dsn7.d.skype.net: type A, class IN
▼ Answers
▶ dsn7.d.skype.net: type CNAME, class IN, cname dsn7.skype-dsn.akadns.net
▶ dsn7.skype-dsn.akadns.net: type A, class IN, addr 213.199.179.166
▶ dsn7.skype-dsn.akadns.net: type A, class IN, addr 65.55.223.12
▶ dsn7.skype-dsn.akadns.net: type A, class IN, addr 157.56.52.27
▶ dsn7.skype-dsn.akadns.net: type A, class IN, addr 64.4.23.161
▶ dsn7.skype-dsn.akadns.net: type A, class IN, addr 157.55.235.150

0040 81 80 00 01 00 09 00 00 00 00 04 64 73 6e 37 01dsn7.
0050 64 05 73 6b 79 70 65 03 6e 65 74 00 00 01 00 01 d.skype.net....
0060 c0 0c 00 05 00 01 00 00 02 48 00 18 04 64 73 6eH...dsn
0070 37 09 73 6b 79 70 65 2d 64 73 6e 06 61 6b 61 64 7.skype-dsn.akad
0080 6e 73 c0 19 c0 2e 00 01 00 01 00 00 00 0e 00 04 ns.....
0090 d5 c7 b3 a6 c0 2e 00 01 00 01 00 00 00 0e 00 04

Text item (text), 22 bytes Packets: 483 Displayed: 6 Marke... Profile: Default

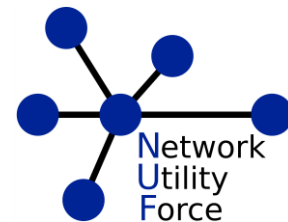


CONCLUSION

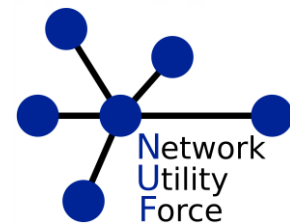


NAT64 is good

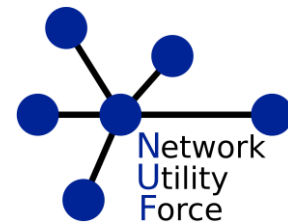
- Works for most apps
- Most of the non-working apps seem reasonably fixable
- Very good vendor support from A10



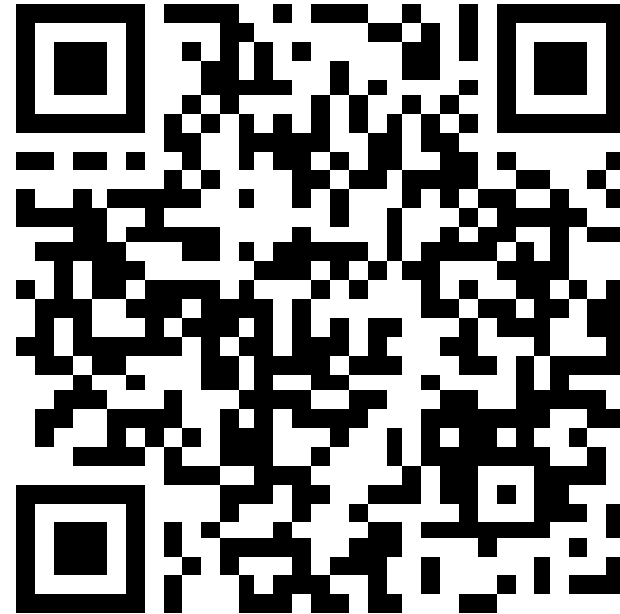
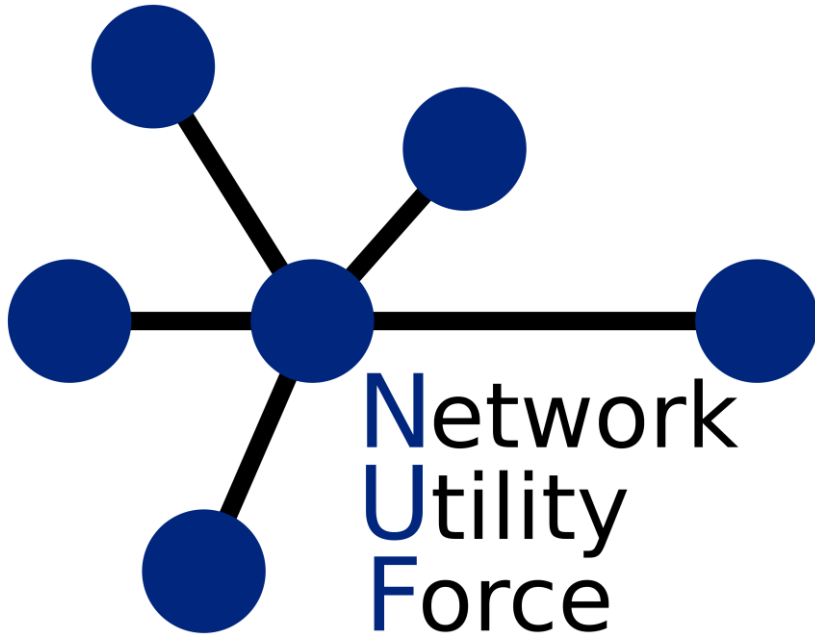
Thanks!!



Q and A



Download the presentation here:



Brandon Ross – Chief Network Architect – Network Utility Force – bross@netuf.net

Erik Muller – Network Architect – Network Utility Force – em@netuf.net

René Paap - Technical Marketing Engineer – A10 network – rpaap@a10networks.com

