



**RIPE**  
**NCC**

## **IPv6 Experiences**

A RIPE Atlas Anchor Project  
Perspective

---

Colin Petrie and Nathalie Trenaman

- What is a RIPE Atlas anchor?
  - Soekris net6501-70
  - Well known target and powerful probe
  - 80 anchors installed
  - 200 probes targeting each anchor with measurements
  - Measurements between anchors



- Capable of 10Mbit bandwidth (less needed today)
- RJ45 network connection
- Static IPv4 and IPv6 address - unfiltered
  - RIPE NCC needs unrestricted access
- IPv6 address needs to be public
- IPv6 connection needs to be native

- The hardware is purchased by the host
- The software is on a USB stick
  - With host-provided network info
- Anchor is configured to boot from USB stick

```
--ip=  
IP address for the interface.  
  
--ipv6=  
IPv6 address for the interface.  
  
--gateway=  
Default gateway, as an IPv4 or IPv6 address.
```

- Silly bug
- Is fixed in CentOS7

- RIPE Atlas anchor provisioning needs to be done over IPv4 and the RIPE NCC remotely configures the IPv6 part of the network later.
- RIPE Atlas anchors will do SLAAC “out-of-the-box”, this needs to be disabled anyway when we do the manual configuration.





# Host IPv6 Installation Challenges

---



- Starting point:
  - SLAAC provided address, gateway via RA
  - Gateway address was link-local
- We pushed the static IPv6 configuration and connectivity was lost...
  - Global unicast gateway address we received from host did not respond to neighbor solicitations:

```
[root@nl-ams-as12345678 ~]# ndisc6 2001:db8:123:a:192:0:2:1 eth0
Soliciting 2001:db8:123:a:192:0:2:1 (2001:db8:123:a:192:0:2:1) on eth0...
Timed out.
Timed out.
Timed out.
No response.
```

- Yet, we are receiving incoming packets on the anchor!
- So the gateway works OK, but is not configured with a global address, or is configured not to respond to NS for that address
- It turned out that the host gave us the wrong gateway address
  - They use two different IPv6 addressing schemes
  - One embeds the IPv4 address
  - One uses ::1/64 as gateway



- The host had working IPv6, ICMPv6 was working fine
- It turned out that TCP/UDP was not allowed

```
bla@loki:~$ mtr -u -nrc 10 nl-bla-as1234567.anchors.atlas.ripe.net
Start: Mon Dec 9 15:57:29 2013
HOST: loki Loss% Snt Last Avg Best Wrst StDev
1.l-- 2001:67c:2e8:13::2 0.0% 10 2.1 3.2 1.9 7.8 1.7
2.l-- 2001:db8:1::b800:2308:1 0.0% 10 2.0 4.2 2.0 15.1 4.0
3.l-- ??? 100.0 10 0.0 0.0 0.0 0.0 0.0
```

- Host forgot to modify edge filters for IPv6

- Host gave us an IPv6 address ending on all zeroes
  - 2001:db8:3bda:666::/64
- The router on their subnet did not respond to NS that originate from the “subnet-zero” address
- 2001:db8:3bda:666::2 worked fine
- RFC4291 section 2.6.1 provided clarity
  - Subnet Router Anycast Address

- Juniper kindly rejected the all-zeroes IPv6 address:

```
[edit interfaces ge-3/0/8 unit 666 family inet6]
'address 2001:db8:3bda:666::0'
  Cannot assign address 0 on subnet
error: configuration check-out failed
```

- Lesson learned: The all-zeroes IPv6 address is not a normal IPv6 address

- This anchor had an invalid router advertisement for an ethernet link

```
[bla@nl-aaa-as2345678 ~]$ sudo rdisc6 eth0
Soliciting ff02::2 (ff02::2) on eth0...
Hop limit           :      64 (  0x40)
Stateful address conf. :      No
Stateful other conf. :      No
Router preference   :      medium
Router lifetime     :      1800 (0x00000708) seconds
Reachable time      :      unspecified (0x00000000)
Retransmit time     :      unspecified (0x00000000)
Source link-layer address: 74:8E:F8:BC:07:89
MTU                 :      1500 bytes (valid)
Prefix              :      2001:db8:102:200::/56
Valid time          :      2592000 (0x00278d00) seconds
Pref. time          :      604800 (0x00093a80) seconds
from fe80::768e:f8ff:febc:0789
```



- [RFC2462] “An IPv6 address prefix used for SLAAC of an Ethernet interface must have a length of 64 bits”
- We saw different variations for the same problem:
  - /32
  - /56
  - /128
- Because we override SLAAC with manual configuration, this problem is easy to solve



- This host had a broken gateway
- Pings and traces did not succeed

```
[bla@nl-aaa-as2345678 ~]$ sudo traceroute6 -I jp-tyo-as2500.anchors.atlas.ripe.net
traceroute to jp-tyo-as2500.anchors.atlas.ripe.net (2001:200:0:6002::a10:1a2), 30 hops max, 80 byte packets
 1 2001:db8:2381:ffe::1 (2001:db8:2381:ffe::1) 1.836 ms !N 2.265 ms !N 2.505 ms !N
```

- Also broken to other destinations within the same provider
- Default route for IPv6 was missing

- This host was using a tunnel instead of native IPv6
- Also, they blocked the ICMPv6 message “packet too big”
- We discovered this because the size of packets we could deliver was 1480 bytes - instead of the 1500 configured on the wire
- When a packet bigger than 1480 bytes was sent, we did not receive the “packet too big” message, with a suggestion for a different MTU size
- Outbound packets of 1500 bytes were dropped



## Conclusions

---



- There are still IPv6 related software bugs in current mainstream server operating systems
- Common mistakes are being made
- IPv6 is still not as well understood as IPv4



