



Network Assurance on a Budget.

Tim Warnock - AusNOG 2026

Who am I?

Principal Network Engineer at Superloop

Been in the networking industry more than 20 years

This page is intentionally blank

What is this talk about?

Covering how Superloop uses its lab environment to:

- Validate vendor claims
- Verify supplier specification conformance (IE: NBN TC4)
- Replicate complex faults
- Software upgrade validation and regression
- Some of the pitfalls and lessons learned along the way

How we achieved this?

Use of:

- Low cost hardware
 - MikroTik CCR1036 -> MikroTik CCR2216
- Ethernet Testers
- Open-source software on x86 compute
- Problem register

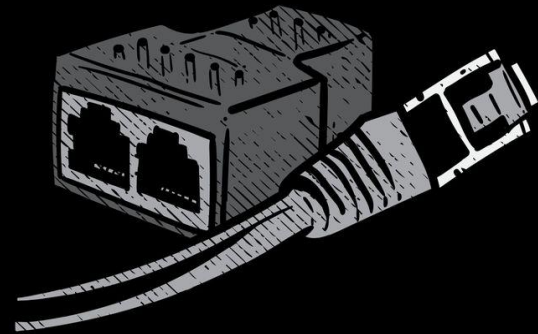
Prerequisite Information.

- What is a PCAP?
- What is Wireshark?
- What is tcpump?

What is a PCAP?

A standardised file format that contains the timestamp, frame length and contents of one or more captured ethernet frames

PCAP
OR IT DIDN'T
HAPPEN



Yes - I'm actually explaining what a PCAP is to a room full of network engineers 😊

Why are PCAPs important?

Packet inspection is **incredibly powerful**

- Eliminates guess work/see the problem

PCAPs can often be decoded with wireshark

- Two examples are MPLS and SIP traffic

Can be replayed with tcpreplay/mikrotik PCAP injector

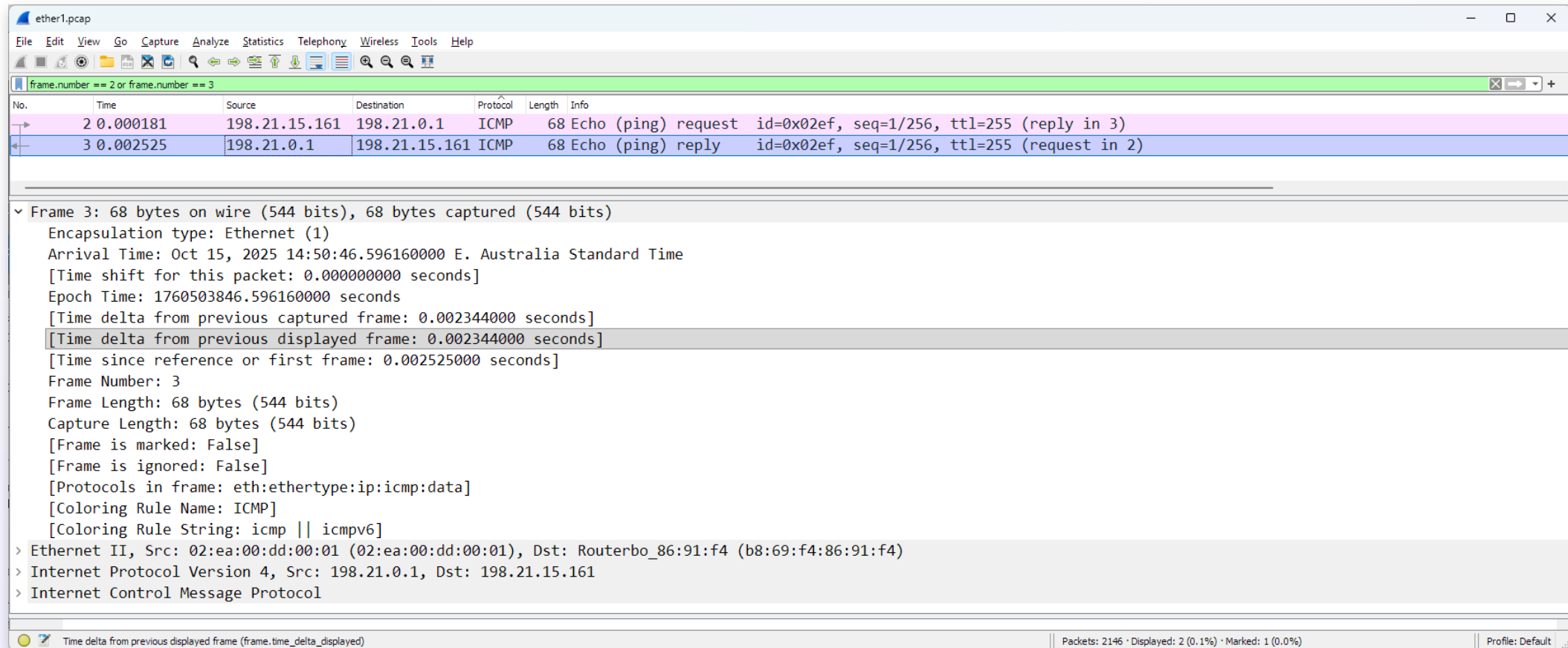
Can be given to your vendor

PCAP generation.

How do generate PCAPs?

- Mirror/span port to PC with software like wireshark or tcpdump
- On device capture (eg: mikrotik, cisco, arista and more)
- Packet captures in your virtual labs? proxmox + tcpdump

What is Wireshark?



A graphical utility (Win/Mac/Linux) that lets you create and view pcap files.

What is tcpdump?

```
user@lab:~$ sudo tcpdump -i ens18 -nnXXs 1500 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens18, link-type EN10MB (Ethernet), snapshot length 1500 bytes
```

```
04:55:34.112874 IP 192.168.252.120 > 192.168.252.121: ICMP echo request, id 10378, seq 1, length 64
0x0000: bc24 113d 8bb9 bc24 11c7 96d9 0800 4500  .$.=...$......E.
0x0010: 0054 e418 4000 4001 dc4c c0a8 fc78 c0a8  .T...@...L...x..
0x0020: fc79 0800 b755 288a 0001 c641 646a 0000  .y...U(....Adj..
0x0030: 0000 46bc 0100 0000 0000 656c 6c6f 4175  ..F.....elloAu
0x0040: 734e 4f47 3230 3236 4865 6c6c 6f41 7573  sNOG2026HelloAus
0x0050: 4e4f 4732 3032 3648 656c 6c6f 4175 734e  NOG2026HelloAusN
0x0060: 4f47                                     OG
```

```
04:55:34.112903 IP 192.168.252.121 > 192.168.252.120: ICMP echo reply, id 10378, seq 1, length 64
0x0000: bc24 11c7 96d9 bc24 113d 8bb9 0800 4500  .$......$.=....E.
0x0010: 0054 8fbb 0000 4001 70aa c0a8 fc79 c0a8  .T....@.p....y..
0x0020: fc78 0000 bf55 288a 0001 c641 646a 0000  .x...U(....Adj..
0x0030: 0000 46bc 0100 0000 0000 656c 6c6f 4175  ..F.....elloAu
0x0040: 734e 4f47 3230 3236 4865 6c6c 6f41 7573  sNOG2026HelloAus
0x0050: 4e4f 4732 3032 3648 656c 6c6f 4175 734e  NOG2026HelloAusN
0x0060: 4f47                                     OG
```

A CLI utility for *nix platforms that lets you create and view pcap files.

Generating bulk traffic.

Having the ability to generate bulk traffic in the lab is very useful

- Allows you to verify traffic is making it from A to B
- Validate device capacity and capability



This is important as vendors often tell porkies about capabilities or quote capabilities with only that feature enabled

Hardware ethernet tester.

Dedicated hardware based ethernet testers are useful but relatively expensive.

Pros:

- Can generate reports
- Accurate packet loss/bit error detection

Cons:

- High cost per bps
- Will struggle to do any kind of complex testing



MikroTik traffic generator.

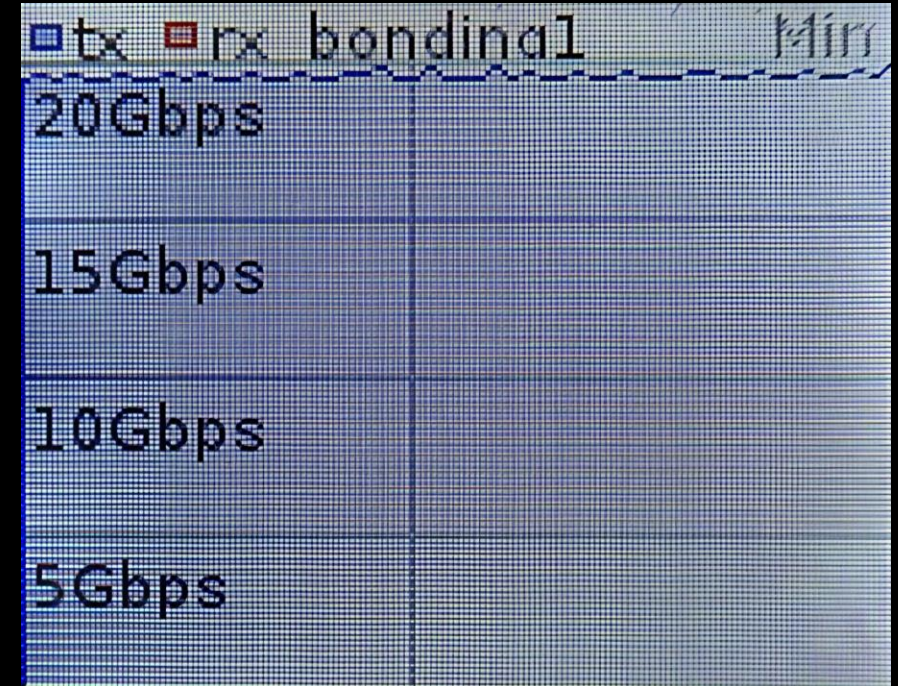
```
[admin@MikroTik] > /system/device-mode/update traffic-gen=yes  
update: turn off power or reboot by pressing reset or mode  
button in 4m46s to activate changes
```

```
/tool traffic-generator packet-template  
add name=packet-template1  
mac-dst=FF:FF:FF:FF:FF:FF/FF:FF:FF:FF:FF:FF  
mac-src=FF:FF:FF:FF:FF:FF/FF:FF:FF:FF:FF:FF
```

```
/tool traffic-generator port  
add interface=sfp-sfpplus1 name=port1  
add interface=sfp-sfpplus2 name=port2
```

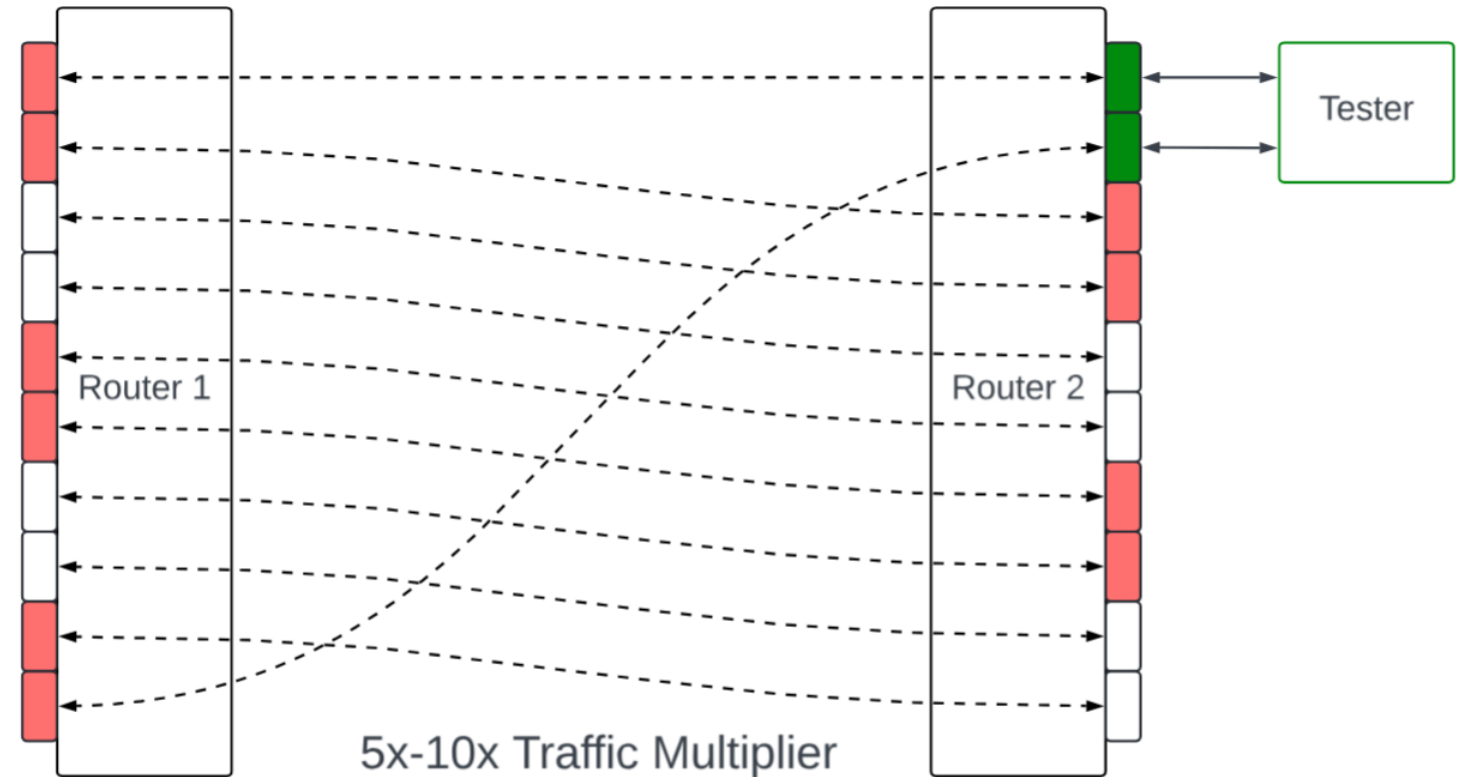
```
/tool traffic-generator stream  
add id=1 name=str1 packet-size=256 port=port1  
tx-template=packet-template1
```

```
add id=2 name=str2 packet-size=256 port=port2  
tx-template=packet-template1
```



Snake test.

Simple way to amplify traffic
Loop adjacent ports in router
Build circuits between routers
Traffic multiplied uplink
router <-> router



Bulk traffic pitfalls.

Rate adaptation

- 100G to 10G or 1G is hard on buffers

MAC addresses

- Using invalid MACs can be troublesome.
- Test known troublesome MACs such as 4x:x..x:xx and 6x:x..x:xx

Generated traffic will likely have invalid CRCs.

1	0.000000000	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
2	0.000023611	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
3	0.004121742	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
4	0.004131442	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
5	0.008240172	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
6	0.008249587	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
7	0.012362824	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
8	0.012372186	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214
9	0.016489622	127.0.0.1	0.0.0.0	UDP	256	100 → 200	Len=214

> Packet comments

> Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits) on interface sfp-sfpplus1 rx, id 0

> Ethernet II, Src: Broadcast (ff:ff:ff:ff:ff:ff), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 0.0.0.0

> User Datagram Protocol, Src Port: 100, Dst Port: 200

Source Port: 100

Destination Port: 200

Length: 222

[Checksum: [missing]]

[Checksum Status: Not present]

[Stream index: 0]

> [Timestamps]

UDP payload (214 bytes)

> Data (214 bytes)

RtBrick BNG Blaster.

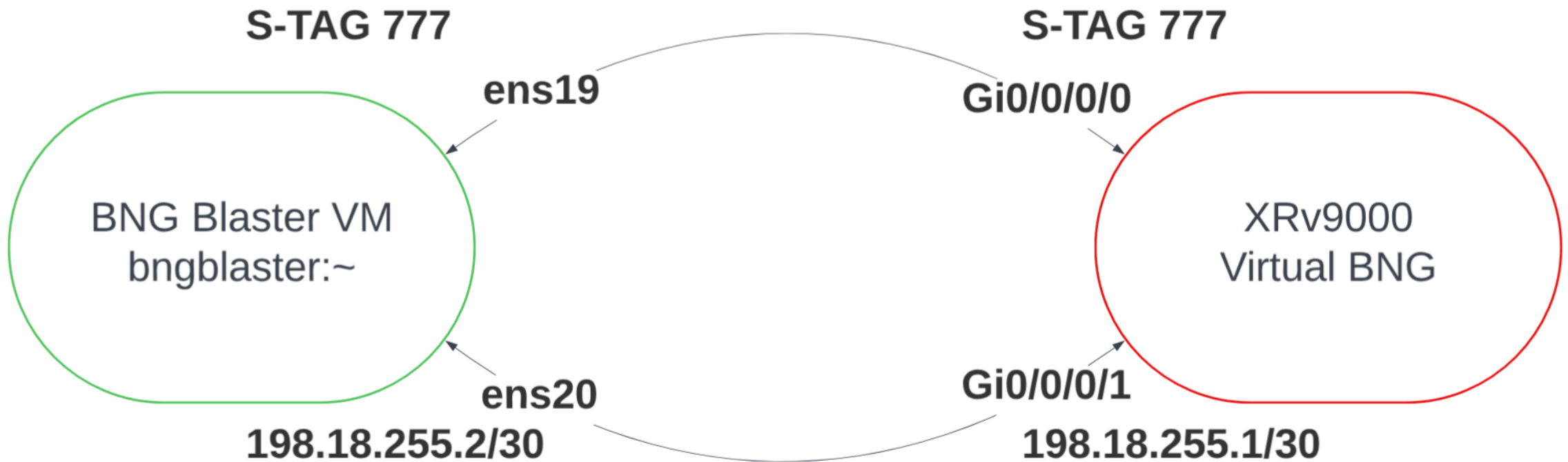


Open source software for generating PPPoE and IPoE subscribers at massive scale

- We've tested to 32k sessions on a 10-core server
- Can do full duplex traffic verification
- Beats a "rack full of modems"



BNG Blaster diagram.



BNG Blaster demo config.

Network

- ens20 { "address": "198.18.255.2/30", "gateway": "198.18.255.1",
"address-ipv6": "fc00::100:2/64", "gateway-ipv6": "fc00::100:1" }

Access

- ens19 { "type": "ipoe", "ipv6": true, "outer-vlan-min": 777, "outer-vlan-max": 777,
"inner-vlan-min": 1, "inner-vlan-max": 4000 }

DHCPv4 + DHCPv6 enabled

{ "agent-circuit-id": "AVC0000{padded-outer-vlan}{padded-inner-vlan}" }

(eg AVC000007770001 via custom source code patch)

BNG Blaster demo verification.

Traffic streams for traffic verification

3 streams:

- Best Effort IPv4 "access prefix": duplex, 2 pps, 256 bytes
- Best Effort IPv6 "access prefix": duplex, 2 pps, 256 bytes
- Best Effort IPv6 "delegated prefix": duplex, 2 pps, 256 bytes
- Testing 4000 sessions at low PPS values due to compute at home

BNG Blaster demo results.

Multi-hour test:

- 4000 sessions
- 48k pps (FDX) traffic (99.998%)
- No sessions flapped

All on a HP T740 thin client running proxmox

F1: Select View F7/F8: Start/Stop Traffic F9: Terminate Sessions
F2: Network Interface Left/Right: Access Interface



Test Duration: 13130s All Sessions Established: 13009s

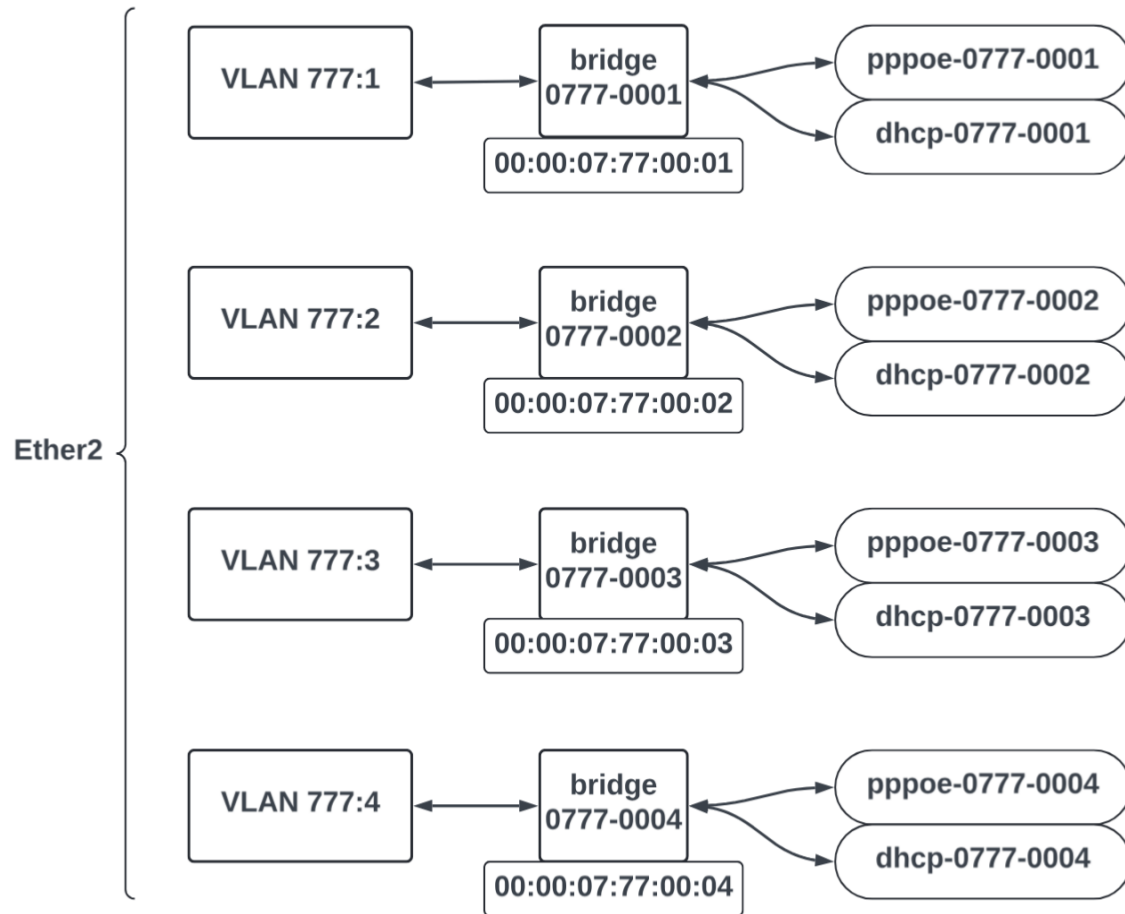
```
Sessions          4000 (0 PPPoE / 4000 IPoE)
Established       4000 [#####]
Outstanding       0 [ ]
Terminated       0 [ ]
DHCPv4           4000/4000 [#####]
DHCPv6           4000/4000 [#####]
Setup Time       120746 ms
Setup Rate       33.13 CPS (MIN: 0.19 AVG: 29.49 MAX: 34.08)
Flapped          0
```

```
Traffic Flows Verified
Stream           24000/24000 [#####]
```

```
Network Interface ( ens20 )
TX Packets        310711946 | 24252 PPS      52384 Kbps 0.052 Gbps
RX Packets        310706246 | 24251 PPS      52382 Kbps 0.052 Gbps
TX Stream Packets 310708966 | 24252 PPS
RX Stream Packets 310704155 | 24251 PPS      4858 Loss (0.002%)
TX Multicast Packets 0 | 0 PPS
```

```
Access Interface ( ens19 )
TX Packets        311913074 | 24256 PPS      53938 Kbps 0.054 Gbps
RX Packets        312505535 | 24304 PPS      53197 Kbps 0.053 Gbps
TX Stream Packets 310709030 | 24251 PPS
RX Stream Packets 310703923 | 24250 PPS      5031 Loss (0.002%)
RX Multicast Packets 0 | 0 PPS      0 Loss
```

MikroTik vCPE.



- Simulates 4 q-in-q subs (DHCP or PPP) on CHR
- Set DHCP option 82 manually per sub for auth.

• `/ip dhcp-client option add code=82`

`name=AVC000007770001`

`value=0x01`

`0F 41 56 43 30 30 30 30`

`30 37 37 37 30 30 30 31`

Subscriber Testing Pitfalls.

Watch your COS markings (for nbn)

- DHCP (FSOL and renews)
- PPP (PADO/PADS/PADT)

Test other functions like RADIUS
Auth and CoA

RIB/FIB stress test.

Understanding route scale is important

Not all routes are the same/Not all TCAM is the same

LEM vs LPM vs eTCAM

IPv4 vs IPv6 key size vs prefix size



MikroTik fantasy routing (ROS7).

```
/routing/fantasy
```

```
add disabled=no  
dst-address=20.0.0.0/8 gateway=10.100.1.2 name=fantasy4-000  
count=100 prefix-length=24
```

```
add disabled=no  
dst-address=4000::/16 gateway=2101::2 name=fantasy6-000  
count=100 prefix-length=48
```

```
BGP sessions: output.redistribute=fantasy
```

Routes are installed on the 'Tik: requires a valid gateway IP (but could be a bridge to nowhere)

Route testing pitfalls.

Use a mix of IPv4 **and** IPv6 and mix prefix lengths.

Fantasy routing:

- Prefix of /8 of IPv4 and a length of /24, count = 100 = 100 x /24 routes within specified /8.
- Uses a random seed. Can end up with duplicate routes/ECMP.

Generating complex traffic.

Thought exercise:

How would you generate 16,000 ICMP echo requests in your lab when each request must have a:

- Unique source IP
- Unique source MAC
- Unique S-TAG + C-TAG
- Valid CRC



Create a file icmp_echo_request.pcap

Write out the standard PCAP header

S = 900 to 915, C = 1000 to 1999 (inclusive)

IP = 10.0.0.1

Loop start

SRC MAC maps S + C

VLAN S + C = literally S + C

Build IPv4 header + ICMP code 0, TTL=64, seq = 0

Calculate CRC, Write out packet to disk

Increment timestamp by 0.1ms

Loop end

**Code to
the rescue.**

|| Packets: 16000 · Displayed: 16000 (100.0%)

(python) PCAPs or Lies.

1	0.000000	10.0.0.1	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
2	0.000100	10.0.0.2	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
3	0.000200	10.0.0.3	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
4	0.000300	10.0.0.4	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
5	0.000400	10.0.0.5	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)

```
> Frame 1: 50 bytes on wire (400 bits), 50 bytes captured (400 bits)
> Ethernet II, Src: HDMILice_84:03:e8 (00:0c:03:84:03:e8), Dst: XeroxXer_02:02:00 (00:00:aa:02:02:00)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 900
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1000
> Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.2.1
> Internet Control Message Protocol
```

15997	0.1599600	10.0.62.125	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
15998	0.1599700	10.0.62.126	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
15999	0.1599800	10.0.62.127	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)
16000	0.1599900	10.0.62.128	192.168.2.1	ICMP	50 Echo (ping) request	id=0x0000, seq=0/0, ttl=64 (no response found!)

```
> Frame 16000: 50 bytes on wire (400 bits), 50 bytes captured (400 bits)
> Ethernet II, Src: HDMILice_93:07:cf (00:0c:03:93:07:cf), Dst: XeroxXer_02:02:00 (00:00:aa:02:02:00)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 915
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 1999
> Internet Protocol Version 4, Src: 10.0.62.128, Dst: 192.168.2.1
> Internet Control Message Protocol
```

tcpreplay/MikroTik inject PCAP.

tcpreplay (tcpreplay) - Replay network traffic
stored in pcap files

MikroTik has this same "Inject PCAP" feature

Manipulating the replay speed can set the desired pps rate

Can be looped to run indefinitely

What else?

I have generated PCAPs of many thousands of unique RFC compliant DHCP requests to test CoPP

PCAPs are a standard format – the limit is your imagination

Closing Thoughts.

- Have a problem register so you know your defects
- Have a suite of tests you run and know should work
 - (Bonus points if you have PCAPs for comparison)
- Test for regressions after changes



Thanks!

Questions?

*'Our curiosity goes with you on your journey.
You walk in the footsteps of those who came before you,
and your path guides those who will follow later.'*