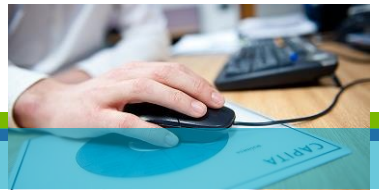
CAPITA
Networking Solutions

updata infrastructure
Part of Capita plc

Open Source Network Performance and Validation Testing

James Bensley
Platform Architect

UKNOF 39, 17 January 2018



Disclaimer

Many of the views and opinions expressed in this talk are my own and do not represent the views of my employer (CNS). If you are unclear which views are my own and which are the views of my employer, please contact me.

We're part of Capita Networking Solutions…

updata infrastructure
Part of Capita plc

SWAN | SCOTLAND NETWORKING TOGETHER

ELECTRANET

pervasive
Part of Capita plc

CAPITA
Networking Solutions

CAPITA
Collaboration Technologies

CAPITA
Cisco Solutions

End to End Network Services

Cabling | LAN | WAN | Managed | Voice

About us

Company

- Founded in 2003
- Acquired by Capita in 2014
- 500 x Staff
- £90M revenues in 2016
  £101M in 2017
- PSN accreditation
  HSCN Compliance (2017)
  [ISO27001, CAS(T)]

Proposition

- Managed Wide Area Networks – long term contracts
- Internet Service Provider – Tier 2
- Reigate and Glasgow Network Operations Centres – 24 x 7
- 1,035 points of presence in UK (inc BT exchanges)
- Network Integrator e.g. BT, Virgin, Talk Talk, Dark Fibre, Wireless

Customers

- <50 direct customers serving 10,000+ customer sites
- 90% Public sector focused – Schools and Local Govt – 12% of UK schools
- Selling to and through Capita
- Focus markets for 2017 – NHS, Central Govt. and Private Sector

CESG Certified Professional

bsi. ISO/IEC 27001 Information Security Management

PRINCE2

MSP

ITIL

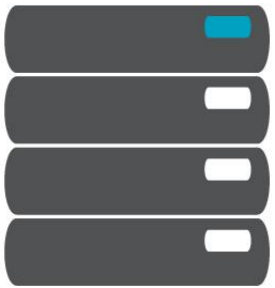
PSN Public Services Network

Crown Commercial Service Supplier

HSCN HEALTH & SOCIAL CARE NETWORK COMPLIANT

Problem definition

- Nothing is what it seems with vendors *and* operators; advertisement vs. reality
- Vendors sell black box devices with undocumented behaviour
  - ➢ Not knowing how a device works is unsafe/insecure/unacceptable
- Troubleshooting can *require* TAC assistance (undocumented commands)
  - ➢ Not knowing how to debug/troubleshoot an issue is unacceptable

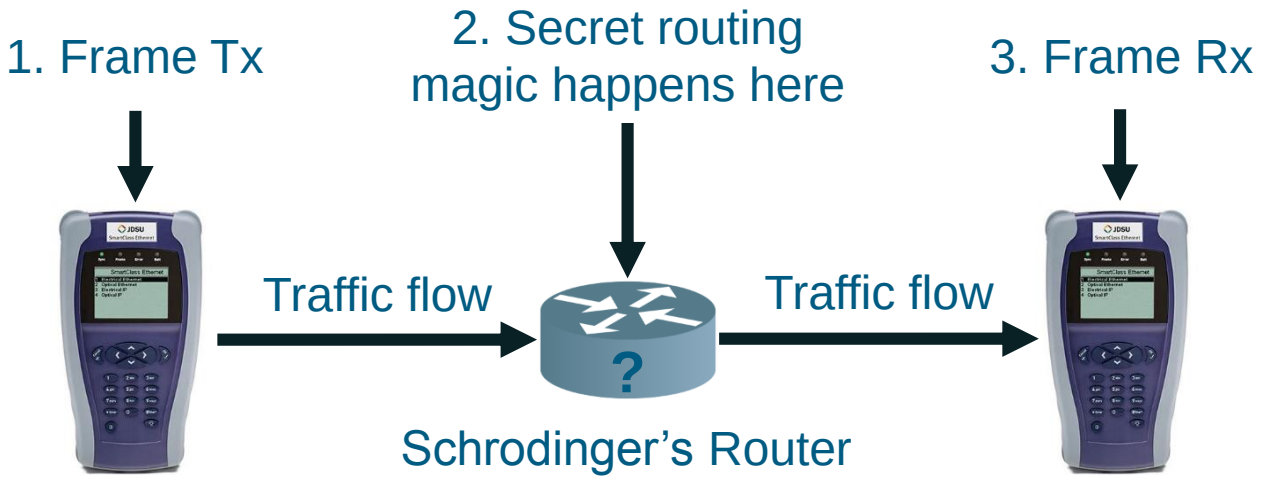


Problem definition

- Fail safe: “If you haven’t tested it, it doesn’t work!”
- However, operators use black box testers to verify that magic is happening!
  - Not knowing the tester validation process is unsafe/insecure/unacceptable
- “We don’t have time to test all of our customer requirements” and/or

  “We can’t test every possibility in the lab!”

Problem definition



Was the *exact* same frame received that was sent?

http://www.testequipmentdepot.com/viavi/images/csc-ethernet.jpg

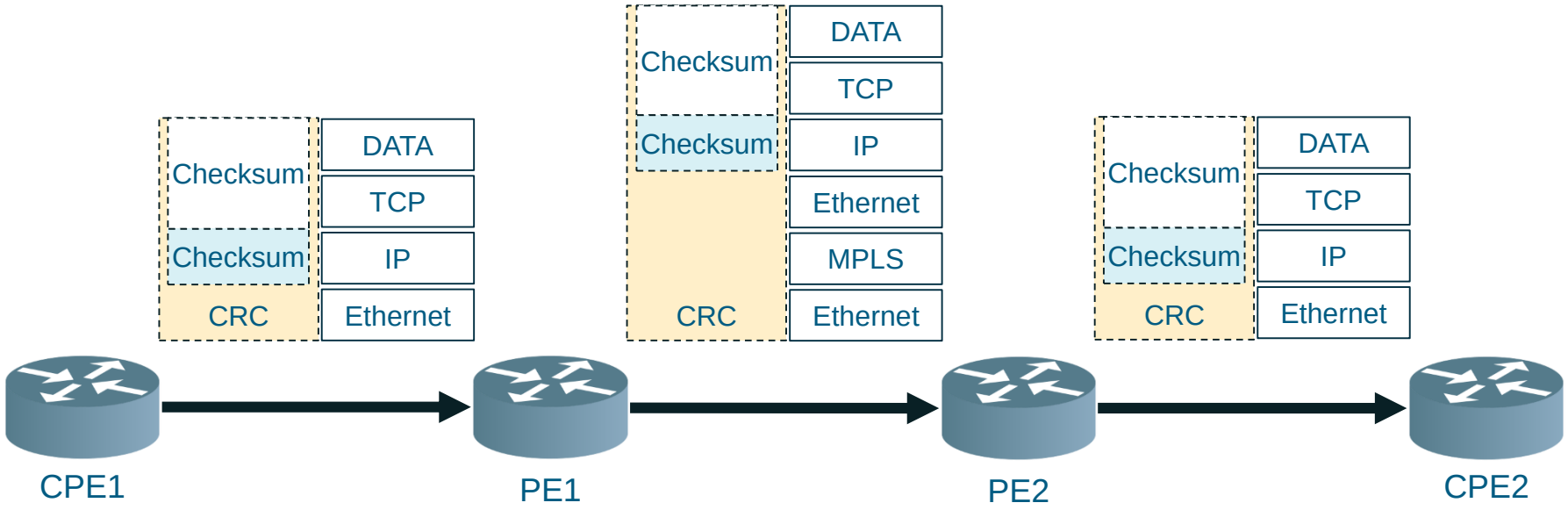
Problem definition

Example vendor issues:

- Does the hardware/software even work? Hashing on Broken Assumptions
- Is it being tested? NPU cache misses causes 33% performance drop (CSCvf44769)
- Are known features bugs being fixed? ECMP with L2 and L3 MPLS VPNs is inherently flawed due to a heuristic methodology. Old problem BCP128 (2007), still an issue draft-ietf-pals-ethernet-cw-00.txt (2017!)
- Are known security bugs being fixed? CVE-2018-0014 Juniper ScreenOS Etherleak vulnerability is back! (origins in CVE-2003-001!)

Problem definition

Example operator issue: MPLS is a transport protocol, not an encapsulation protocol



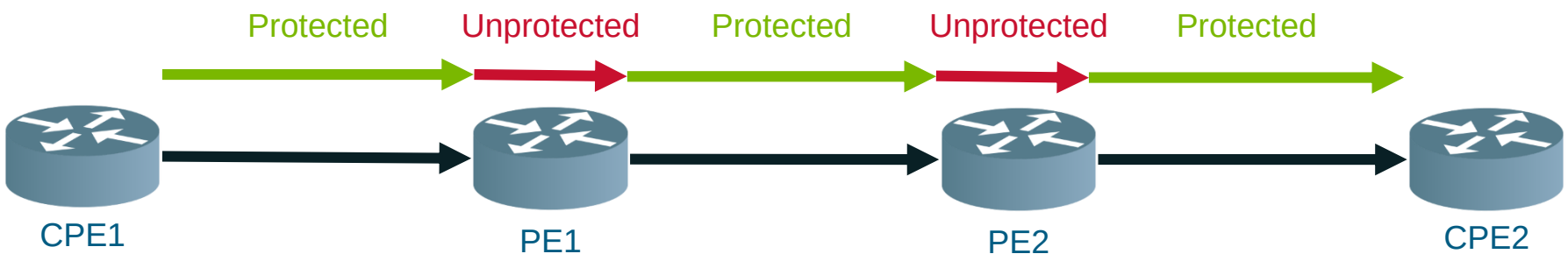
https://commons.wikimedia.org/wiki/File:Router.svg

Problem definition

Some vendors have NPU counters (which are exposed via SNMP):

- E.g. Cisco ASR9K: PARSE_DROP_IPV4_CHECKSUM_ERROR
- E.g. Juniper: bad-IPv4-hdr

No known tooling to test this end to end!



https://commons.wikimedia.org/wiki/File:Router.svg

Problem conclusion

- At the lowest layers of our protocol stack things are (still) quite broken
- Many problems arise from the secrecy of hardware/software operations
- Accessible tooling to self test for these issues isn't widely available

Solution scoping

I have been focusing my research on Ethernet and MPLS only so far:

- Essential:
  - Only network devices are in scope, end user/hosts are out of scope
  - Standards based testing (RFC2544 and ITU-T Y.1564 compliant)
  - Open source/open design
- Nice to have:
  - Low cost
  - Vendor neutral

Solution scoping

- RFC2544 (1999) isn't a strict definition, ITU-T Y.1564 is "better" but not perfect (however, any standardisation is better than none!)
- FYI:
  - RFC5180 - IPv6 Benchmarking Methodology for Network Interconnect Devices
  - RFC5695 - MPLS Forwarding Benchmarking Methodology for IP Flows
  - Benchmarking Methodology Working Group

FOSS tooling comparisons

FOSS tooling comparisons

Etherate - Raw socket based Ethernet and MPLS packet generator:

- Any layer 2/2.5 header value (load packet-as-hex fall back)
- Constraint based testing (time/speed/volume)
- Easy CLI usage (no API / not scriptable)
- Hardware agnostic
- Lowest performance
- Stateless

https://commons.wikimedia.org/wiki/File:Green_tick_pointed.svg
https://commons.wikimedia.org/wiki/File:Red_X.svg

FOSS tooling comparisons

Pktgen - DPDK based packet generator using LuaJIT:

- Most layer 2-4 header options (load PCAP as fall back)
- “Range” and “sequence” native features
- All options in CLI and Lua API (scriptable)
- Highest performance
- Requires DPDK supported NIC
- Stateless

FOSS tooling comparisons

MoonGen - DPDK based packet generator using LuaJIT:

- Any layer 2-4 header options
- No CLI options, scripted tests only (Lua API)
- Partially stateful
- High(er) performance
- Requires DPDK support NIC
- DPDK EAL settings are hidden

FOSS tooling comparisons

- Etherate assumes two difference devices are being used.

  MoonGen & Pktgen assume the Tx and Rx hosts are the same device.
- Etherate can be used to test a physical device or link at layers 2/2.5.

  Pktgen & MoonGen can be used to test a physical device or link at layers 2- 4 for high performing metrics (high throughput or low latency)
- Etherate can also test the raw socket path within the Kernel networking stack.

  PktGgen & MoonGen can also provide some low level NIC stats.

Examples: BUM filter accuracy

NIC: Intel I350 1G, DUT: Cisco 2960, Test: Etherate broadcast test

2960#show storm-control fa0/15

Interface	Filter State	Upper	Lower	Current
Fa0/15	Forwarding	0.25%	0.25%	0.24%

$ sudo ./etherate -i eno2 -g -G -d FF:FF:FF:FF:FF:FF -M 250000

Seconds	Mbps Tx	MBs Tx	FrmTx/s	Frames Tx
1	0.24	0	20	20
2	0.24	0	20	40

Examples: every Ethertype value

NIC: Intel I350 1G, DUT: Cisco 2960, Test: MoonGen “setType” ethertype

Rx NIC drops ~1400 frames, from etype 0x2F to 0x5DC, 0x8100, and 0x888e

Random missing Ethertype chosen and retested, 0x2F == 100% lost

Random working Ethertype chosen and retested, 0x2E == 100% received

0x2E-0x5DC are length values for 802.3 Ethernet + LLC/SNAP (802.2) framing

0x8100 (802.1q VLAN tag): 0 packets input, 65536 runts

0x888e (802.1X EAP): 65536 packets input, 0 errors/drops/runs/discards

“switchport mode access” / no 802.1X configured

Examples: every Ethertype value

NIC: Intel X710 10G, DUT: ASR9001, Test: MoonGen “setType” ethertype

Rx NIC drops ~8500 Ethertypes

0x8808 (802.3x “pause”) 0 packets input, no NP counters

0x88a8 (802.1ad QinQ/PB), 0x9100 and 0x9200 (802.1q QinQ), NP Counter: PARSE_DROP_IN_UIDB_TCAM_MISS

Intel X710’s use secret squirrel firmware. 3C34: *Demystifying Network Cards*

Examples: every Ethertype value

NIC: Intel X710 10G, DUT: ASR9001, Test: Pktgen performance/size distribution

Pktgen:/> set 0 size 247

Pktgen:/> start 0

NP Counters:

PARSE_TOP_LOOP_RECEIVE_CNT	5987587286	6742449
MDF_PIPE_LPBK	5987589832	6742451
MDF_PIPE_LPBK_BUFFER_PREFETCH	2963853909	3371225

Solution: Recap of work so far

Initial research has been presented:

- Researched the existing hardware/software/protocol problems
- Evaluated the existing toolset (currently using ADE 651 equivalent)
- Evaluated tests to detect known issues

https://en.wikipedia.org/wiki/ADE_651#/media/File:ADE_651_at_QEDcon_2016_01.jpg

Solution: recap of work so far

Vendors are important! I'm happy to pay someone else to:

- ✓ Develop new hardware & software
- ✓ Provide support when there is an issue
- ✓ Provide training to staff
- ✓ etc...

But:

- I don't want to be completely dependant on them and a victim of their failures...

Solution: next steps

- New Etherate features; frame pacing, bit fiddling, frame checksums
- EtherateMT; coming soon for faster host/kernel sourced testing
- MoonGen & Pktgen: Fix RFC2544 test scripts and implement ITU-T Y.1564
- Add RFC5180 and RFC5695 test scripts
- GitHub repo with MoonGen Lua scripts
- Working towards “we can test every possibility in the lab”

Solution: next steps (long term)

Run RFC2544/ITU-T Y.1564 tests…

- *directly* to/from the network device: Cumulus Linux ?
- *along side* the network device:

  Hosting Applications on IOS-XR / ASR9K Virtualized Services Module / Configuring Virtual Services ?
- *in* the network forwarding software: VPP / FD.io ?

Or using open source FPGAs ?

Questions?

Contact me using these details:

- Email: jwbensley@gmail.com / james.bensley@updata.net
- Slack: http://networktocode.slack.com/
- Skype: jameswbensley

Thank you