Test and Measurement with
the Ninja box (and BART)

David Hayes
dahayes@swin.edu.au
Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Outline
Introduction - Test and Measurement in CAIA
What is the Ninja Box?
4.5G2 DAG cards
   Clock Synchronisation
   Data Stream Management
   Basic Capture
   Basic Traffic Generation
Simple Experiment
   Setup
   Results
Port Mirroring Delays
Conclusions
Thanks
Test and Measurement of various network characteristics is a vital part of CAIA’s research.

The Endace Ninja Box (http://www.endace.com/ninjabox.html) will enhance our traffic measurement and traffic generation capabilities.

This talk will outline the Ninja Box capabilities in conjunction with:

- The Broadband Access Research Testbed (BART) http://caia.swin.edu.au/bart/

What is the Ninja Box?

- Server grade machine, optimised for packet capture
- Our Configuration:
  - 2 × 2 GHz Intel(R) Xeon(R) core 2 duo
  - Linux kernel 2.6.18, Centos OS, 4G ram
  - 2TB disk (8 disk raid)
  - 2×DAG 4.5G2 – Precision packet capture cards
4.5G2 DAG cards

- Data Stream Management
- Inline Forwarding
- Timed Release ERF

DAG Clock Synchronisation

- Geographically separated measurements
  - Synchronise with GPS
DAG Clock Synchronisation

- Geographically separated measurements
  - Synchronise with GPS
- We currently synchronise DAD 0 to the PC clock
  - Dag cards synchronised to each other
  - PC is synchronised by NTP
  - DAG cards synchronised to PC by DUCK

DAG Data Stream Management

- Filter/load balancing
- Packet colourising and dropping
- Packet steering
Packets are received from one or both ports

Either:
- Use inbuilt load balancing classifier
- Or Classify (and drop) based on programmable filters

Colourise (based on above result)

Steer to stream buffer
- 2Rx and 1Tx per DAG

Steering can include duplication

---

**Basic Capture**

- dagsnap — high speed capture (erf)
  - packets from a previous capture session

- Post process if necessary (WAND libtrace (http://research.wand.net.nz/software/libtrace.php))
  - traceanon
  - tracefilter
  - tracesplit
  - tracemerge

- dagconvert — change format (ie to pcap)
Basic Traffic Generation

- **daggen**
  - Configuration file to describe traffic
  - Addresses can be random
  - Payloads can vary, deterministically or randomly
  - Outputs ERF format file (or can go direct to DAG)

- **dagflood**
  - Sends ERF file
  - To flood link: `dagconfig nodelay`
  - For timed replay: `dagconfig relative`

---

Simple Experiment

Objective

- Familiarity with:
  - NinjaBox ([http://www.endace.com/ninjabox.html](http://www.endace.com/ninjabox.html)),

- Compare NinjaBox and PC based tcpdump ([http://www.tcpdump.org/](http://www.tcpdump.org/)) captures
Setup

Measure
- RTT $A \leftrightarrow B$
- for scp $A \rightarrow B$

Results RTT $A \leftrightarrow B$

Observations
- TCP sawtooth
Observations

- 100Mbps burst
- Ack clocked
- Rx TCP Acks while awaiting ssh response

---

Observations

- Pcap timestamp – time kernel saw packet, **Not time sent on the wire**
- Dag timestamp – *mirrored*
Port Mirroring Delays

UP direction

\[ \Delta t_{up} = \tau_{rx} + \tau_{Ain} + \tau_{copy} + \tau_{Mout} \]

Observations
- Local tcpdump overestimates RTT
Port Mirroring Delays

UP direction

\[ \Delta t_{up} = \tau_{rx} + \tau_{Ain} + \tau_{copy} + \tau_{Mout} \]

DOWN direction

\[ \Delta t_{down} = \tau_{copy} + (\tau_{Mout} - \tau_{Aout}) \]

- Perturbation depends on:
  - Packet size
  - Switch load
  - Full duplex utilisation

Conclusions

- Ninjabox will provide increased test and measurement capabilities for CAIA
- SPP is a great tool for RTT calculation
- When very accurate timing is required:
  - Care should be taken with the DAG clock synchronisation
  - Care should be taken with how the packets are captured
Thanks

- Jason
- Amiel
- Lawrence