

IBL Six Month Report

David Kennedy
dkennedy@swin.edu.au



Introduction



- 6 months experience at CAIA
 - Projects worked on
 - Outcomes achieved
 - Reflection
- Not just a technical presentation, but an overall IBL experience
 - What I learnt
 - Skills I gained/improved



Enemy Territory Server

- Great way to get started
 - Bit of fun
 - Got me interested in my work
 - Put my newly gained FreeBSD knowledge into practice
 - See CAIA tech report "Configuring a 'Wolfenstein: Enemy Territory' Server for Online Play",
<http://caia.swin.edu.au/reports/040831A/CAIA-TR-040831A.pdf>



etctl → qctl

- Etctl is the script we use to start/stop/restart the ET server
- Based on a set of Quake 3 scripts used for the same thing
- Ported etctl to Quake, to function like the original Quake 3 scripts
- Learnt how to write Bourne Shell scripts





Netsniff

- Set up testbed with FreeBSD
- FreeBSD bridge/traffic shaper
- Scripting (Python)
- Functionality and performance testing
- Jitter measurement testing
 - Required NISTNet, which required Linux



Grangenet and the game server

- Game server connected to Grangenet's high-speed network
- Co-hosted web server
- Traffic interaction/interference testing
- Used packet generating/capturing tool called RUDE & CRUDE
- See CAIA tech report "Investigating the Influence of HTTP Traffic on the Accuracy of Packet Time Stamping",
[http://caia.swin.edu.au/reports/041124A/CAIA-TR-](http://caia.swin.edu.au/reports/041124A/CAIA-TR-041124A.pdf)





KUTE

- Following on from previous project
- Kernel based tools written by Sebastian for the Linux kernel
- Used SmartBits
- Compared Tcpdump, Rude & Crude, and Kute
- Kute:
 - Higher packet rates
 - More accurate



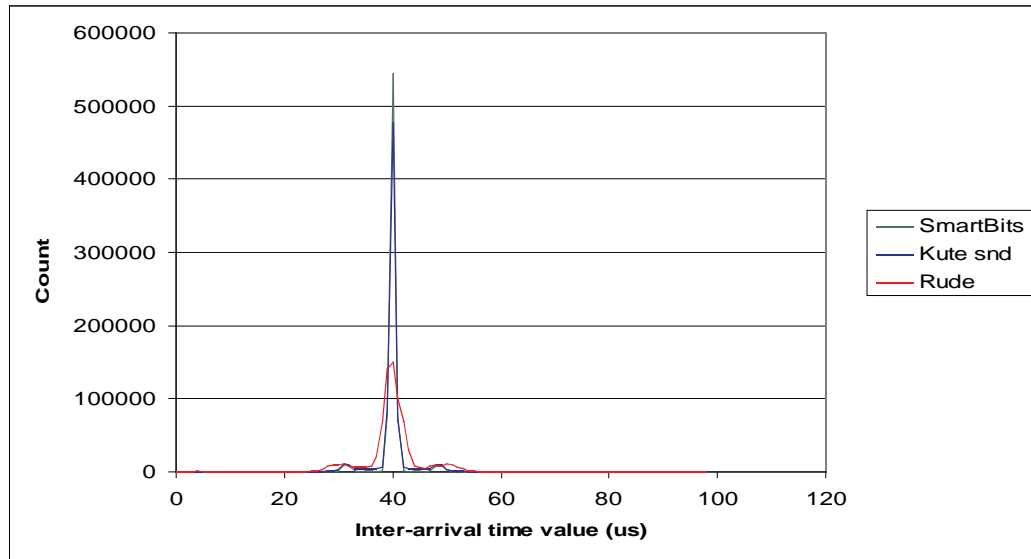
KUTE

- Max sending rates (packets per second x1000)
 - RUDE ~48
 - KUTE ~415
- Max receiving rates
 - CRUDE ~46
 - TCPDump ~50
 - KUTE ~220

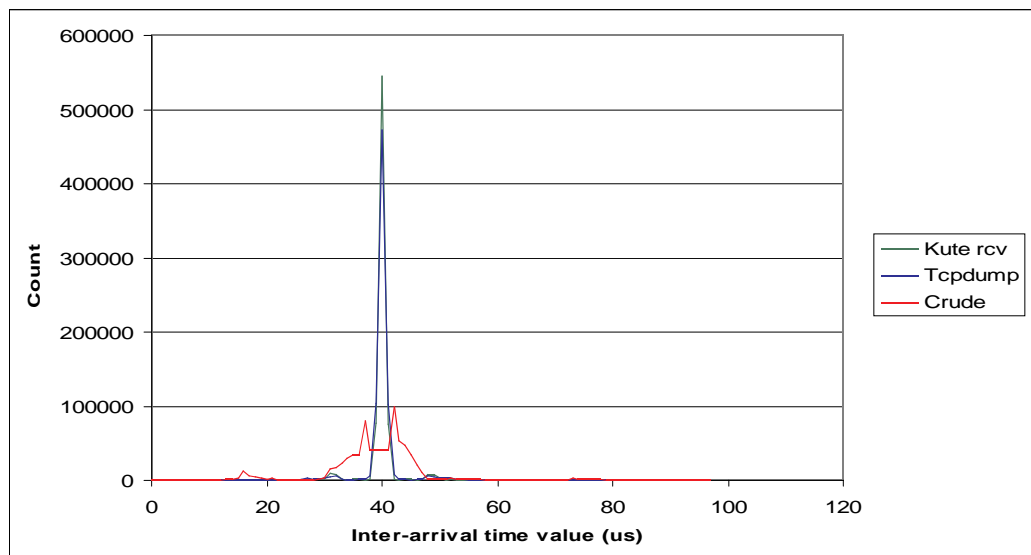




Sender distributions



Receiver distributions



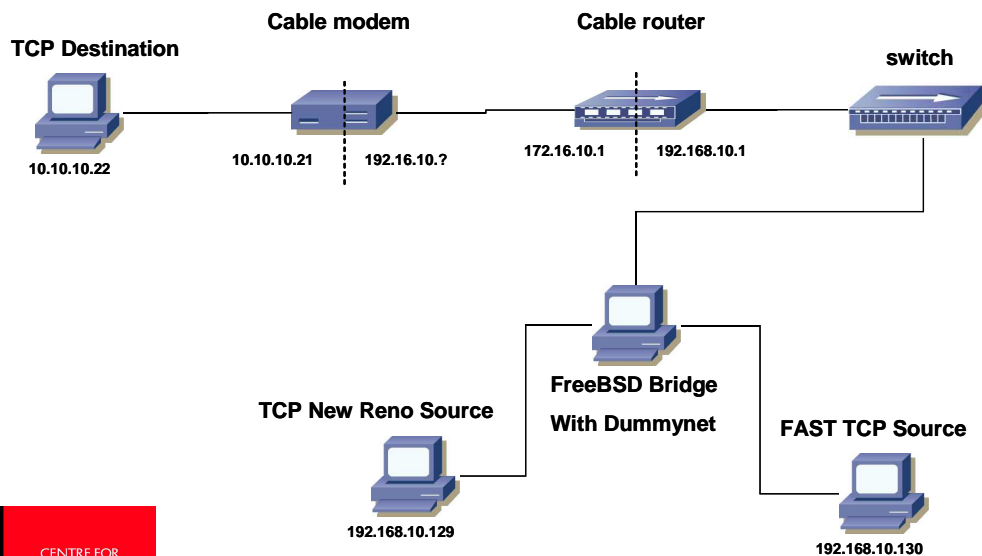


FAST TCP

- Developed by Caltech
 - Designed for and tested only on high speed links
- We are testing it on consumer broadband access networks, using the BART testbed
 - Set up the testbed with both Linux and FreeBSD machines
 - Wrote automated test script
 - Configured BART Cisco network devices



FAST TCP Cable Modem Testbed





FAST testing

- Iperf
- Ping
- GNUPlot
- Test script organises data, produces graphs
- Used various buffer and bandwidth settings



Alpha tuning

- FAST alpha value determines how much queueing delay to sustain
- We were testing to see how this value would affect the delay and the throughput
- All the tests were the same
- Found that “alpha tuning” was overriding the value of alpha
- Now fixed up and we are starting to get meaningful results





Looks like it's ET time

- Unless there are any questions

