NetFPGA in Cambridge

Andrew W. Moore
• Cambridge: not exactly network newcomers
• NetFPGA: right tool / right time
• Teaching
  – Masters course (similar to CS344)
  – Masters dissertation vehicle (6 month piece of work)
  – Undergraduate project vehicle (e.g., TOE implementation)
• Research
  – network emulation elements
  – implementation vehicle for middlebox ideas
  – testing new ideas for a revamped Ethernet
  – new MACs for new networks (SWIFT) and a prototype vehicle
  – target platform for better development toolchains
• Dissemination
  – Tutorials and workshops
Cambridge? never heard of them

- But you may have heard of some of our more successful projects (some have changed name):
  - Autonomy
  - ARM
  - BBC Micro
  - alphamosaic
  - Virata
  - Level5
  - ANSA
  - xenSource

- And some of our not so successful projects:
  - Cambridge Backbone Ring
    - 1 Gb/s LAN/WAN in 1995
  - ATM
    - (we didn’t want 48 byte payloads either – so very silly)
  - Sun’s sunray
NetFPGA Teaching in Cambridge

• Coursework
  – P33 “Building an Internet Router”
    – based upon Stanford cs344

• Graduate Dissertations
  – A new Masters course means 6 month dissertations
    – (think of them as “PhD qualifiers”)

• Undergraduate Projects
  – Smallish “Computer Science complete” projects
    – 2008/9: Peter Ogden implemented a TOE on NetFPGA
P33: “Building an Internet Router”
A Cambridge course from October

- A module in a new single-year Masters degree
  MPhil (Advanced Computer Science)
  - a “pre-PhD” entry programme.
- Lecturer: me
- TAs: Phil Watts and David Miller
- Ideally 3 groups of 3, current expressions of interest is 22(!)... but many
  will fall short of prerequisite requirements.
- Principally a pass-fail subject (with the “project competition reward”), BUT
  the subject is on offer to other Masters has a 0-100 mark scale (60=pass).

This was planned to be a “clone” of cs344
P33: “Building an Internet Router” (how well will we translate?)

Well not a clone, more a translation:
  • *Arnie* becomes *Sean*

- Stanford Terms ≠ Cambridge Terms
  - so not quite enough weeks... solutions include:
    • cut the extension weeks
    • bigger groups (classic Brookes law (Mythical Man-Month) failure)
    • do less (e.g. drop the CLI requirement)
    • start with more:
      (start with CLI and static Ethernet switch)

- A lot more Lecturer contact time (a function of this being a new module and not having as many helpers as Nick, yet...)

- Entry criteria (Stanford and Cambridge have ECAD (Verilog))
  - most of the UK/EU does not (or has VHDL)
    Our solution is to seed with a few Cambridge ECAD backgrounded people
NetFPGA-enabled Research

- network emulation elements
- implementation vehicle for middlebox ideas
- testing new ideas for a revamped Ethernet
- new MACs for new networks (SWIFT) and
  - a prototype vehicle for networks that don’t exist
- target platform for better development toolchains (C# -> kiwi -> (bluespec) -> Verilog)
Middlebox: AtoZ

• AtoZ implements an application-aware traffic manager on NetFPGA

  – Application-detection technology is the “magic in the box” but the implementation was challenging and noteworthy

• NetFPGA allows handcrafting to suite test deployments

Look for our paper in ANCS 2009 in Princeton
MOOSE: Addressing the Scalability of Ethernet

- An approach to Ethernet that blurs the boundary of Layer-2 and Layer-3, through:
  - improved routing
  - mitigating broadcast/multicast data and
  - none of the DHT complexity of SEATTLE

- Currently a software prototype with a NetFPGA implementation in progress.

- (Solves similar problems to the “Floodless in SEATTLE” approach, but in a different/better way...)
Building a new PCI

Multi-\(\lambda\) host test-bed

Electronic control (FPGA-based)

Host sync

Arbiter
Switch control

Logic for path select

Data generators

Control

4x10 Gb/s

Control

 Optical switch fabric

Multi-\(\lambda\) host

Sequential multi-host testbed for multi-wavelength packets with FPGA control
Building a new PCI

• NetFPGA used as a test target in a latency study of PCI (old and new)

Look for our paper in ANCS 2009 in Princeton

• NetFPGA-based prototype network is the basis of a test network for a new (bufferless) PCI approach
NetFPGA 2-Day workshop in Cambridge

- 20 attendees (full house)
- Accommodation for non-locals
- 30% commercial attendees

Next Cambridge workshop: March’10
(tutorial, workshop or camp... to be decided)

Want a tutorial/workshop at your institution? talk to Andrew/John
How might we use NetFPGA?

- Build an accurate, fast, line-rate NetDummy/nistnet element
- A flexible home-grown monitoring card
- Evaluate new packet classifiers
  - (and application classifiers, and other neat network apps....)
- Prototype a full line-rate next-generation Ethernet-type
- Trying any of Jon Crowcroft's ideas (Sourceless IP routing for example)
- Demonstrate the wonders of Metarouting in a different implementation (dedicated hardware)
- Provable hardware (using a C# implementation and kiwi with NetFPGA as target h/w)
- Check that some brave new idea actually works
  - e.g. Rate Control Protocol (RCP), Multipath TCP, toolkit for hardware hashing
- Hardware supporting Virtual Routers

A flexible home-grown monitoring card

- Evaluate new packet classifiers
  - (and application classifiers, and other neat network apps....)
- Prototype a full line-rate next-generation Ethernet-type
- Trying any of Jon Crowcroft's ideas (Sourceless IP routing for example)
- Demonstrate the wonders of Metarouting in a different implementation (dedicated hardware)
- Provable hardware (using a C# implementation and kiwi with NetFPGA as target h/w)
- Check that some brave new idea actually works
  - e.g. Rate Control Protocol (RCP), Multipath TCP, toolkit for hardware hashing
- Hardware supporting Virtual Routers

- Hardware channel bonding reference implementation
- TCP sanitizer
- Other protocol sanitizer (applications... UDP DCCP, etc.)
- Full and complete Crypto NIC
- IPSec endpoint/ VPN appliance
- VLAN reference implementation
- metarouting implementation
- virtual <pick-something>
- Intelligent proxy
  - application embargo-er
  - Layer-4 gateway
  - h/w gateway for VoIP/SIP/skype
  - h/w gateway for video conference spaces
  - security pattern/rules matching
- Anti-spoof traceback implementations (e.g. BBN stuff)
- IPTv multicast controller
- Intelligent IP-enabled device controller (e.g. IP cameras or IP powermeters)
- DES breaker
- platform for flexible NIC API evaluations
- statistics reference implementation
- show (hp) reference implementation
- trajectory sampling (reference implementation)
- Implementation of zeroconf/netconf configuration language for routers
- Openflow and simple POX controller in one
- Network RAID (multicast TCP with redundancy)
- offline compression
- load-balancer
- openflow with (netflow, ACL, ...)
- reference NAT device
- active measurement kit
- passive measurement
- active sender control (e.g. performance feedback fed to endpoints for control)
- Prototype platform for NON-Ethernet or near-Ethernet MACs
  - Optical LAN (no buffers)

- Different driver/buffer interfaces (e.g. PFring)
- or "escalators" (from gridprobe) for faster network monitors
- Firewall reference
- GPS post-timestamp things
- High-Speed Host Bus Adapter reference implementations
  - ISCSI
  - Myranet
  - Fiber Channel
- Smart-Device (presuming a direct connection to FPGA)
- Software Defined Radio (SDR) directly on the FPGA (probably UWB only)
- Routing accelerator
  - Hardware network reflector
  - Internet Exchange route accelerator
- Hardware channel bonding reference implementation
- TCP sanitizer
- Other protocol sanitizer (applications... UDP DCCP, etc.)
- Full and complete Crypto NIC
- IPSec endpoint/ VPN appliance
- VLAN reference implementation
- metarouting implementation
How might YOU use NetFPGA?

- Build an accurate, fast, line-rate NetDummy/nistnet element
- A flexible home-grown monitoring card
- Evaluate new packet classifiers
  - (and application classifiers, and other neat network apps....)
- Prototype a full line-rate next-generation Ethernet-type
- Trying any of Jon Crowcroft’s ideas (Sourceless IP routing for example)
- Demonstrate the wonders of Metarouting in a different implementation (dedicated hardware)
- Provable hardware (using a C# implementation and kiwi with NetFPGA as target h/w)
- Hardware supporting Virtual Routers
- Check that some brave new idea actually works
  - e.g. Rate Control Protocol (RCP), Multipath TCP,
- toolkit for hardware hashing
- MOOSE implementation
- IP address anonymization
- SSL decoding “bump in the wire”
- Xen specialist nic
- computational co-processor
- Distributed computational co-processor
- IPv6 anything
- IPv6 – IPv4 gateway (6in4, 4in6, 6over4, 4over6, ....)
- Netflow v9 reference
- PSAMP reference
- IPFIX reference
- Different driver/buffer interfaces (e.g. PFRING)
- or “escalators” (from gridprobe) for faster network monitors
- Firewall reference
- GPS packet-timestamp things
- High-Speed Host Bus Adapter reference implementations
  - Infiniband
  - iSCSI
  - Myranet
  - Fiber Channel
- Smart Disk adapter (presuming a direct-disk interface)
- Software Defined Radio (SDR) directly on the FPGA (probably UWB only)
- Routing accelerator
  - Hardware route-reflector
  - Internet exchange route accelerator
- Hardware channel bonding reference implementation
- TCP sanitizer
- Other protocol sanitizer (applications... UDP DCCP, etc.)
- Full and complete Crypto NIC
- IPSec endpoint/ VPN appliance
- VLAN reference implementation
- metarouting implementation
- virtual <pick-something>
- intelligent proxy
- application embargo er
- Layer-4 gateway
- h/w gateway for VoIP/SIP/skype
- h/w gateway for video conference spaces
- security pattern/rules matching
- Anti-spoof traceback implementations (e.g. BBN stuff)
- IPTv multicast controller
- Intelligent IP-enabled device controller (e.g. IP cameras or IP powermeters)
- DES breaker
- platform for flexible NIC API evaluations
- snmp statistics reference implementation
- sflow (hp) reference implementation
- trajectory sampling (reference implementation)
- implementation of zeroconf/netconf configuration language for routers
- h/w openflow and (simple) NOX controller in one...
- Network RAID (multicast TCP with redundancy)
- inline compression
- hardware accelerator for TOR
- load-balancer
- openflow with (netflow, ACL, ....)
- reference NAT device
- active measurement kit
- network discovery tool
- passive performance measurement
- active sender control (e.g. performance feedback fed to endpoints for control)
- Prototype platform for NON-Ethernet or near-Ethernet MACs
  - Optical LAN (no buffers)
Next...

• You can do it too....
  (Many of you have done it already!)

  – Research (even the smallest scale)

  – Teaching using the NetFPGA

  – Dissemination of the NetFPGA project...