A Hands-On Environment for Teaching Networks

Stanford High Performance Network Group

Presented By:  Martin Casado
Nick McKeown
Gregory Watson
Martin Casado
Dan Wendlandt

February, 2005
Presentation Outline

- The problem
- Our approach
- Application in the classroom
- Experiences in the classroom
Problem Overview:

Motivation

\[ \text{App} \quad \{ \text{TCP/IP} \quad \text{Phy} \} \quad \text{OS + hardware} \]
Problem Overview:

Motivation

“How can you teach router implementation to a large undergraduate class?”
Problem Overview:

Provide Router per Student?
(dedicated hardware)

- Obvious resource constraints
- Difficult to set up/manage
- Complicated development environment
Problem Overview: Use Network Simulator?

- No access to real Internet traffic/Synthetic
- Specialized development environment
  - Learning curve
  - Non-standard scripting interfaces (we want C!)
Presentation Outline

- The problem
- Our approach
- Application in the classroom
- Experiences in the classroom
Our Approach:
Virtualize the Network
Virtual Network System:

Why Don't We ....

- Start with a machine between the Internet and some servers
- .. have it emulate a network topology
- .. make that multiple network topologies
Our Approach:

Virtual Network System

Student Computer

Internet

VNS Client

VNS Server
Virtual Network System:

Client Library

- Reserve a virtual host on a given topology
- Read a packet from a specified interface
- Write a packet to a specified interface
- Bindings in C/C++/Java/Python
Virtual Network System: Result?

- Each student has their own topology
- Students can develop from anywhere
- Access to live traffic
- Can interact with hosts on Internet
- Can support 1000's of students
Virtual Network System: Packet Forwarding

1. Packet arrives destined for web server ‘C’.

2. VNS server forwards packet to students VNS client

3. Student’s client tells the server, which interface to route packet to

Internet

VNS Client

VNS Server

PC

Linux

Web/ftp servers

PC

Linux

Web/ftp servers

PC

Linux

PC

Linux

PC

Linux

PC

Linux

VNS Server
Presentation Outline

• The problem
• Our approach
• Application in the classroom
• Experiences in the classroom
Virtual Network System:

VNS in the Classroom

- Projects
  - Building a functional router
  - Building the full networking stack
  - Building a hardware/software router
- Demonstrating networking concepts
Example Assignment: Software Router

- Each student gets their own topology
- Single router attached to Internet and application servers
VNS in the Classroom

Software Router

- Handle ARP
  - request/response
  - cache/queue of packets pending responses
- Support various ICMP types
  - ECHO (ping)
  - Time Exceeded (traceroute)
  - Port Unreach
  - Host Unreach
- Forward correctly from static routing table
- IP header checksum
VNS in the Classroom

Software Router: Code Complexity

- Typically around 1,000 lines of code
- No-threads required
- Project takes 2 weeks – 1 month
- Roughly 30 hours of work
VNS in Practice:
Building a Full Network Stack

- All components of the networking stack
- Combination of three projects
  - Ftp client
  - Software router
  - TCP compatible transport layer (STCP)
VNS in Practice:
Full Network Stack
VNS in Practice:
Building an Integrated Software/Hardware Router

- Full router in hardware and software
- Managed via CLI
- Hardware forwarding path
- Router – router protocol
- Must inter-operate with other students
VNS in Practice:

Hardware, NetFPGA

- Network programmable FPGA
- Basically a programmable, 8 port network device
- Provides facilities for handling communication paths to/from network
- Students use Verilog
- Works with VNS
VNS in Practice: Software

- Software runs as “CPU” for hardware through VNS
- Integrated TCP stack
- User applications
- Simplified OSPF variant
- Controls hardware via 'control packets'

[Diagram showing layers: CLI, “User Apps”, TCP, IP/ICMP/OSPF, ARP, IP Forwarding, Layer 2 forwarding]
VNS in Practice:
Other Projects

- Simplified link-state routing protocol
- Open ended router design
- Security (Conceptual)
  - ssh man in the middle
  - tcp session hijack
  - Traffic poisoning/redirection
  - etc.
VNS in Practice:
Classroom Demonstrations

- Want to visualize network concepts
  - TCP congestion window over time
  - Router buffer occupancies
- “View” inside operating router
- Real-time inspection of protocol stack
VNS in Practice:

Clack

- GUI Router (Java applet)
- Build router out of visual components
- Allows real time visualization
  - tcp flows
  - tcp sawtooth
  - queue sizes etc
- Useful for teaching/demoing
- Work in progress ...
- [http://yuba.stanford.edu/vns/clack](http://yuba.stanford.edu/vns/clack)
VNS in Practice: Clack

TCP Flow Data Outstanding

Data Outstanding (Bytes)

Time (milliseconds)

0 10,000 20,000 30,000 40,000 50,000 60,000 70,000

ftp server

http server
Presentation Outline

• The problem
• Our approach
• Application in the classroom
• Experience in the classroom
Experiences in the Classroom

- **Stanford**
  - Introductory Networking Course (3 years)
    (router, stcp, full network stack)
  - Graduate course in router design (2 years)
    (Integrated hardware router)
- **Johns Hopkins**
  - upper level networking protocols course (2 years)
    (router, OSPF)
- **WSU**
  - introductory networking course (1 year)
    (router)
- Over 1500 students
- Create remote teaching lab
Student Feedback

- Overwhelmingly positive (anonymous formal evaluations)
- Hard but very rewarding
- Many requests for access after projects are finished
- Many students experiment with more complex functionality (NAT, VPN, firewalls, web-servers, QOS etc.)
Teacher Feedback

- Very positive
- Both pilot schools will continue using
- Have been excited to contribute project ideas/extensions
Plug

- Are looking to host more remote projects
- Provide full curriculum and support
- Collaboration with other projects?
- If you are interested, please contact us directly!
  - casado@cs.stanford.edu
  - http://yuba.stanford.edu/vns
  - http://yuba.stanford.edu/vns/clack/
Thank You!

[Diagram showing network protocols and APIs]
Virtual Network System:
Physical Setup

![Diagram showing network setup with Internet, firewall, hub, VNS Servers, and Application Servers.]}
Problem Overview:

Requirements

- Route live Internet traffic
- Isolated network topology per student
- Systems programming expertise not required
- Efficient use of hardware
- Support 1000s of students
- Support remote projects
- Safe
VNS in Practice:
Building a full router

Software

Hardware

CLI
“User Apps”
TCP
IP/ICMP/OSPF
ARP
IP Forwarding
Layer 2 forwarding
Problem Overview:

Virtual Overlay Network?

- No access to link layer (generally intended for network layer on up)
- Designed for dedicated research environment
- Often virtualize components of OS
- Normally have large scale deployment (X-bone, Planetlab)
Virtual Network System

- Many possible solutions
- None met our particular requirements
- Developed our own within the context of our introductory networking course
Virtual Network System: Transparency

- A challenge! more always better
- Provide tcpdump-compatible files for all packets seen by the router
  - Encourage students to use ethereal/tcpdump for debugging
  - Helps students find most errors
- Run mini-web-server showing which topologies/routers are reserved, number of packets services etc.
- Active area of exploration ... (Especially for complex setups with integrated hardware)
Virtual Network System: Providing Student Access

- Students write user-level program (using provided api)
- Allow user-level program to send/receive packets as if it were on the topology
- Student's can then send real traffic to their server's on the topology
VNS in Practice:

Topology

Application servers

VNS provides cpu

NetFPGA

Firewall

Internet
VNS in Practice:
Software

Application

Socket API

Router API

Transport Subsystem

ICMP Input

IP Forwarding

IP Input

ARP Input

Interface Input

low-level input

ICMP Output

IP Output

IP Routing

ARP Output

Interface State

low-level output

IP Routing State

Dropped

ARP State

ARP Thread

OSPF