An End to the Middle

Colin Dixon
Tom Anderson, Arvind Krishnamurthy
University of Washington

Affiliates Day, October 2008
The Internet

- Rapidly approaching 40 years old
- Scaled by ~9 orders of magnitude (2 hosts to 1 billion)
- We must have really gotten it right (pat pat)

The reality:
- First 20 years were progress from IETF & original design
- Last 20 years have been almost entirely middleboxes
- Improving performance, security, scalability, etc.
- We’re hitting bigger and bigger problems
- Where do the next 20 years come from?
Middleboxes

- “A middlebox is defined as any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host.” – RFC 3234

- Stuff you can’t control which can help but can also get in the way
Today’s Reality (Big)

The idea

Internet

VPN Gateway

Firewall

PacketShaper

LAN

The reality
Today’s Reality (Small)

- Home, Small Business and Developing World networks
- Lack large (or any) IT department
- Lack expensive, well configured middleboxes
- Work, but with lots of effort and performance issues
Why the End?

- End-hosts are evolvable (software over hardware)
- They are uniquely equipped to solve problems
- Cheap (we already have them)
- Reliable (distributed)
- Virtual Machines and trusted computing mean we can actually trust (some) of what they say
- Centralize policy, distribute implementation
Why Now?

- Multicore
- Complex protocols
- Virtual Machines
- Trusted Computing
- iPlane
Applications
End-host NAT

• NATs are the quintessential middlebox violating the end-to-end principle and "architectural purity"

• Also probably the most prominently used

• Can we do it on end-hosts alone?
  • Assign all hosts on a LAN the same IP/MAC address
  • Run a consensus algorithm to determine who "owns" which ports

• NAT becomes visible to the host, while invisible to apps,
Application Firewall

- Current firewalls are mostly port-based
  - Fails to block things it should (everything on port 80)
  - Blocks things it shouldn’t (Second Life)
- Really want to block/allow applications
- If we’re running software on the end-host
  - Can make claims about which applications own which connections
  - Can even write protocol descriptions for applications to tell between good and bad runs
Application QoS

● Performance can really, really suck
  ● Go home and try to browse the web while sending a large e-mail attachment

● Current solution (if any) is mostly port-based

● Bending over backwards to get information the end-host already has
  ● Have end-hosts tag flows
  ● Can differentiate between different port 80s
  ● Can do fancy things like prefer flows belonging to the active window