EECS 122: Introduction to Computer Networks

The Future Internet

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720-1776

Two General Questions

- Why does the Internet need to change?
  - seems to be doing pretty well so far!

- How can we change the Internet?
  - is it possible to change something this big?
  - what does the research community need to do differently in order for change to happen?

Part 1

Why does the Internet need to change?

The Need for Change

- Old problems have become more critical:
  1. Robust, yet vulnerable
  2. Decentralized control makes it hard to manage
  - .....(and many more)

- New challenges are arising:
  1. New uses
  2. New technologies
  - .....(and many more)

#1 The Internet is Robust

- Robustness was one of the Internet’s original design goals

- Adopted failure-oriented design style:
  - Hosts responsible for error recovery
  - Critical state refreshed periodically
  - Failure assumed to be the common case

- Proof from experience: Internet has withstood major outages with minimal service interruption
  - 9/11, Baltimore tunnel fire, etc.

But the Internet is Very Vulnerable

- Router misconfigurations:
  - single misconfiguration can take down 1/3rd of Internet
  - hundreds of misconfigurations/day

- Denial of service attacks:
  - can overload links and sites
  - attacks are common (and often extortionary)
    - millions of bots ready to attack!

- Congestion control algorithms can be subverted
  - flows can easily get more than their fair share
Not Acceptable Any More!

- These failings were annoying but unimportant when the Internet was mostly for research.
- Now that the Internet supports many critical services (banking, government, etc.), they are no longer acceptable.
- But why did the Internet have these problems in the first place?

General Problem

- Most Internet protocols are designed with (at most) two failure models in mind:
  - Participating nodes: fail-stop
  - Other nodes: malicious (spoofing, etc.)
- Such protocols are usually vulnerable to the misbehavior of participating nodes:
  - Subverted nodes
  - Misconfigured nodes
  - Bug in software

Semantic vs Syntactic Failures

- Syntactic failures:
  - Node doesn’t respond, message ill-formed, etc.
- Semantic failure:
  - Node responds with well-formed message, that is semantically incorrect
- Internet designed for syntactic failures, not semantic ones

Examples

- Protocols can’t tell if routing information is true
- Network can’t tell if incoming traffic is wanted
- Routers can’t tell if congestion signals are being obeyed
- Fundamental issue: network protocols typically disseminate information, not verify it...

Why Don’t Traditional Tools Work?

- Formal verification:
  Verifies that correct protocol operation leads to desired result
- Cryptographic authentication:
  Verifies who is talking, but not what they say
- Fault-tolerance via consensus: (Byzantine, etc.)
  Requires several nodes have information for computation
  Routing: only nodes at end of link know about its existence

Approach: Lightweight Verification

- Checking for semantic correctness
- Trying to detect lying (even if after-the-fact) with some probability
- Mechanism must be lightweight enough to scale
- A few examples to follow......
Example: Denial-of-Service

- Destination gives source a "capability" that proves to the network that the destination wants to receive that traffic
  - assume no, unless explicitly allowed
- Or, destination uses "filters" to express to the network that traffic is definitely not wanted
  - assume yes, unless explicitly told otherwise
- In both, challenge is scalability......

Example: Routing

- Use cryptography to ensure route advertisements came from true destination
  - Requires PKI, or self-certifying AS numbers
- Use consistency checks to ensure most paths agree on the correct destination
  - Higher overhead, but requires less trusted infrastructure

Example: Congestion Control

- To keep congestion signals from being corrupted:
  - send nonces with ECN bit
  - secures signaling channel
- To ensure that flows can't get more than their fair share by ignoring congestion control algorithm:
  - fair queuing
  - enforces fair data channel

#2 Network Management

- Internet was designed for resilience:
  - no central locus of control
  - relies on decentralized control algorithms to remain functioning in the face of failures
  - decentralized control was essential for Internet's growth
- However, ISPs would like to manage their networks
  - Configuring routing policy, access control, etc.
  - Traffic engineering
  - Troubleshooting
  - All of these are very hard today!

Global Policy Defined Implicitly

- No declarative policy mechanisms
- Network behavior is the result of the sum of box configurations (because of decentralized control)
- No way for specification of desired overall behavior to drive box configurations
- Approach: centralize policy, or drive it from ends

Failures are Hidden

- Transparent to success, opaque to failure
  - If packet is delivered, ends know it
  - If packet is dropped, ends have no idea where!
- Byproduct of layering and adaptation
  - can't reveal lower layers
  - silently recovering from failures makes those failures harder to find
- Approach: explicit monitoring infrastructure
  - does not violate layering: info is out-of-band
New Challenges

- New uses
- New technologies

Beyond the PC

Challenges from Edge Devices

- Many are mobile (cell phones!)
  - define identity separate from location (e.g., HIP)
- Physical location often important
  - incorporate location in Internet infrastructure
- Usage is often data-centric
  - name data, not just hosts
- How to interconnect non-IP edge-devices?
  - currently, use physical interconnection (usb, etc.)
  - how to do this remotely?

Wireless Sensor Networks (WSNs)

- Wireless Sensor Networks: a new kind of network
  - resource challenged (energy, memory, etc.)
  - tied to physical world (that’s where the data is)
  - must be self-organizing (unattended)
- Much excitement and progress
  - currently seeking overall architecture
- Not clear how much this impacts Internet’s design
  - use of sensors does, but not clear that WSNs do
  - sensors: data-centric, requires query mechanisms

New Technologies

- Many developments in wireless technologies
  - more bandwidth, longer range, etc.
- Rapid innovation in optical communications
  - more bandwidth, more rapid switching
- Both will impact the future Internet, but will they alter its fundamental design?
Beyond Static Links

- When optical connections can be established and torn down rapidly, then the notion of static links becomes archaic
- Connections in wireless networks don’t have static links either
  - link characteristics change rapidly
  - congestion is non-local

Quantum Communication and Computation

- Distributing keys
- Breaking keys

Part 2

Can we change the Internet?

Changing the Internet

- A decade of frustration and failure....
- Proposed innovations die:
  - QoS
  - Multicast
  - IPv6
  - ....
- Scorned developments succeed:
  - Firewalls, NATs, other middleboxes
  - Make Internet more fragile and rigid

But Many Believe We Must Try Again....

- The Internet’s current evolutionary path will not adequately meet our future needs
  - Security
  - Reliability
  - New application and user requirements
  - ....
- We will need a significant architectural change
  - Perhaps not now, but eventually
  - Perhaps incremental, perhaps all-at-once

To Succeed, We Must Stop Lying

- To others, and ourselves.....
### Our Founding (Funding) Fable
- Researchers invent new architectures
- Architectures are validated on a testbed
- Greatly impressed by our wisdom and foresight, IETF, ISPs, and router vendors collaborate to deploy new design

> yeah, right....

### Do Traditional Testbeds Really Test?
- Production-oriented testbeds: (Internet2)
  - Real traffic provides good validation
  - But can test only very incremental changes
- Research-oriented testbeds: (Deter)
  - Can test radical architectures
  - Lack of real traffic results in poor validation
- Both are expensive (dedicated bandwidth)

### What about Deployment?
- Architectural change requires ISP consensus
  - Hard to agree
  - No competitive advantage from architectural innovation
  - All have huge sunk investment in the status quo
- ISPs are unlikely candidates for architectural change
  - Some disagree, but this is for another talk....
- Architecture isn’t just static, its decaying
  - Ad hoc workarounds muddy the architectural waters

### We Are at an Impasse
- We can’t test new architectures
  - Despite sizable investments in testbeds
- We can’t deploy new architectures
  - And things are getting worse, not better
- Yet there are pressing requirements for which the current architecture is not well suited

### The Community’s Response
- Focused on areas where it can have impact:
  - Empirical studies
  - Incremental changes (subject to current constraints)
- Small stream of architectural proposals
  - Paper designs without hope of deployment
  - More science fiction than engineering
- Have largely abandoned hope of effecting fundamental architectural change
  - Living with, rather than overcoming, the impasse

### The Community’s Shame
- If Cisco asked the research community what the next generation architecture should be, most of us wouldn’t have an answer
- That’s not true of the OS, Dbase, and most other communities
- The inability to build and test what we design has narrowed our scope, reduced our ambition, and made cowards of us all....
**Overcoming the Impasse?**

- Must be able to test new architectures:
  - Wide range of architectures
  - Real traffic from willing individuals
  - Low overhead for individual researchers

- Must have a plausible deployment story
  - Not probable, just plausible
  - Avoid need for ISP action or consensus

**Testing: Virtual Testbed**

- Overlay testbed:
  - No dedicated bandwidth
  - Host proxy directs packets to overlay
  - Individuals (anywhere) opt-in by turning on proxy

- Shared infrastructure (think Planetlab)
  - Overlay nodes shared among experiments
  - Any researcher can easily run an experiment

- Real but volunteer traffic, cheap to run, easy to use
  - but forget about delivering strict QoS

**Deployment via Overlays**

- Next Generation Service Provider (NGSP):
  - Deploys overlay supporting new architecture
  - Distributes proxy
  - Provides support for legacy apps (Msoft?)

- Unilateral deployment, by new entrants
  - No consensus, no sunk investment

- What’s new? *Nothing but the attitude…*
  - Current overlays address narrowly scoped issues
  - We advocate using overlays for more radical changes

**NSF’s Response?**

- NSF is supporting a new initiative in Internet Architecture
  - Current funding for architectural research
  - Planning for large-scale testbed

- Will reinvigorate experimental architectural research

**Pondering Success**

- The deployment story is plausible, but unlikely

- What if it did succeed?

- Two extreme scenarios

**The Purist Scenario**

- Periods of architectural stability followed by disruptive change to new coherent architecture
  - In quiescence, a pure well-defined architecture

- Virtualization is a means
  - For testing and deploying new architecture
## The Pluralist Scenario

- Many different architectures simultaneously available
  - No clear distinction between architecture and services
- Virtualization becomes an *end in itself*

## Purists vs Pluralists

- **Purists:**
  - Architecture specified by universal protocol (e.g., IP)
  - Goal: long-term flexibility and applicability
  - Challenge: foreseeing future needs
- **Pluralists:**
  - IP is just one component of the Internet “system”
  - “Architecture” arises from union of overlays, etc.
  - Goal: meet short-term needs to attract users
  - Challenge: how can apps/users cope with diversity?

## A Very Exciting Time!

- Rapid innovation at lower layers:
  - wireless, optical, ...
- Rapid innovation at higher layers
  - new distributed applications, ...
- NSF’s recommitment to Internet architecture
  - Restoring innovation in the core architecture
  - Revitalizing experimental research
  - Re-engagement with deepest fundamental issues