CS 268: Lecture 24
Designing for the Future

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

Issues for the Future

- Future designs (last lecture)
- Evolving the Internet to incorporate these designs
- Living in the brave new world

We Aren’t in Kansas Anymore...

- Internet was designed in cohesive environment
  - Everyone had the same goals, worked together
- The Internet is now fully commercial
- There are many competing interests
  - Users no longer share same goals
  - Providers now compete with each other

User Cooperation?

- Originally users were all part of a joint endeavor
- Now, each user is engaged in their own task
  - Often completely unaware of other users
  - Care only about own performance
- Why should we assume users will cooperate?
  - Why wouldn’t they act selfishly?

Congestion Control

- Each user’s performance is a function of bandwidth and delay/drops
- Users adjust their sending rate to maximize their performance
- What is the resulting equilibrium?
  - FIFO routers: terrible!
  - FQ routers: very good! (detailed game theory analysis)

Example

- $U_i = r_i/d_i$
- Poisson: $d_i = 1/(1-r_{tot})$
- Social Optimal: $r_{tot} = 1/2, U_{tot} = 1/4$
- Nash Equilibrium w/FIFO: $r_{tot} = n'(n+1), U_{tot} = (n+1)^2$
- Nash Equilibrium with FQ: social optimal
### Designing for Selfishness
- Assume every user (provider) cares only about their own performance (profit)
- Give each user a set of actions
- Design a “mechanism” that maps action vectors into a system-wide outcome
- Choose a mechanism so that user selfishness leads to socially desirable outcome
  - Nash equilibrium, strategy-proof, etc.

### Interdomain Routing
- Assume ISPs incur costs for carrying packets
- Want to decrease total costs by carrying packets over lowest cost paths
- How to get ISPs to declare costs without lying?
- General approach: VCG mechanism
  - generalization of 2nd price auction
- Only slightly more complicated than BGP

### Incentives in Practice
- See UW paper on practical uses of incentives

### Tussle Paper
- Conflicting interests are inevitable, and ever-changing
- Accept the presence of tussles, and design accordingly

### Designing for Tussle
- Design for choice, not for outcome
  - separate mechanism from policy
- Modularize design along tussle boundaries

### Interfaces
- Make value visible: allow parties with compatible interests to reach equilibrium
- Make costs visible: allow parties to make informed choices
- Provide tools to isolate and resolve faults/failures
Competition

- Key to innovation is competition
- Only the fear of competition will lead ISPs to adopt new designs, offer new services
- Competition will only arise when users have “choice”
  - choice in local ISPs
  - choice in downstream ISPs
    - e.g., route control to guide packets to ISPs with better service
- What do you think?

Another Viewpoint

- Basic Internet service not a viable business
- Competition will only lead to everyone going broke
- Need to create public Internet service providers
  - Back to the future....

Yet Another Viewpoint

- New services (QoS, Multicast) weren’t adopted because inter-provider agreements are too primitive
- They don’t make value visible, so that ISPs can’t charge for giving better service
  - they can do so for users, but not between themselves!
- Can’t give users choice without passing on costs appropriately

Discussion

Still Another Viewpoint

- "Overcoming the Internet Impasse through Virtualization"
  - hotnets this year

The Paper’s Fundamental Assumption

- 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
Our Founding (Funding) Fable

- Researchers invent new architectures
- Architectures are validated on a testbed
- IETF, ISPs, and router vendors collaborate to deploy new design

Do Traditional Testbeds Really Test?

- Production-oriented testbeds:
  - Real traffic provides good validation
  - But can test only very incremental changes
- Research-oriented testbeds:
  - 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
- 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

- Focus on areas where we 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

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: (think RON, etc.)
  - No dedicated bandwidth
  - Host proxy directs packets to overlay
  - Proxy must architecturally neutral, and flexible
  - Individuals (anywhere) opt-in by turning on proxy
- Shared infrastructure (think Planetlab)
  - Not everyone is David Andersen…
  - Overlay nodes shared among experiments
  - Slicing on per-packet timescales
- Forget about delivering strict QoS

You’ve Heard This All Before….

- E. g. Xbone, Virtual Internet, etc.
  - But our emphasis is on generality and sharing
  - Implementation and philosophical issue
- Has the desired properties:
  - Low overhead for individual researchers
  - Real traffic provided by “coalition of the willing”
- Goal: revive “applied” architectural research
  - Make testing almost as commonplace as simulation

Deployment: Standard Overlay Story

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

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
- Key to evolving the architecture:
  - Proxy support
- 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
- Key to evolving the architecture:
  - Virtual link establishment and virtual routers
  - Substrate for deploying overlays is new “waist”
  - Planetlab becomes model for Internet
- 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?

Three Discussion Questions

- **Intellectual:** What architecture would we choose?
  - Is it sufficiently better?

- **Procedural:** How can we achieve synergy?
  - Or will we continue along our disparate paths?

- **Philosophical:** Are we purists or pluralists?
  - Do we have a choice?

The Ultimate Pluralist: Active Networking

- Seminal work: D. Tannenhouse and D. Wetherall ’96
  - Routers can download and execute remote code
  - At extreme, allow each user to control its packets

An Active Node Toolkit: ANTS

- Add active nodes to infrastructure

Active Nodes

- Provide environment for running service code
  - Soft-storage, routing, packet manipulation

- Ensure safety
  - Protect state at node; enforce packet invariants

- Manage local resources
  - Bound code runtimes and other resource consumptions

Where Is the Code?

- Packets carry the code
  - Maximum flexibility
  - High overhead

- Packets carry reference to the code
  - Reference is based on the code fingerprint: MD5 (128 bits)
  - Advantages:
    • Efficient: MD5 is quick to compute
    • Prevents code spoofing: verify without trust
Code Distribution

- End-systems pre-load code
- Active nodes load code on demand and then cache it

Applications

- Protocol versions
- Multicast
- Mobility
- Various specialty applications (packet auctions)
- ...

Practical Issues

- Efficiency/Performance: largely solved problem
- Node-level protection: largely solved problem
- Network-level protection: unsolved
  - hard to prevent routing loops, too many generated messages, etc.
  - have to rely on "certified" programs, approved by administrative authorities

Philosophical Issues

- E2E argument
- ISP adoption
- Spectrum of options
- ...

Active Networking and the E2E Principle?

- Is active networking in accord with the E2E principle?

Clark, Saltzer, and Reed

End-to-end arguments address design more than implementation and implementation more than execution--that is, they suggest who should provide the code, not which box it should run on.
...... end-to-end arguments have not one, but two complementary goals:

· Higher-level layers, more specific to an application, are free to (and thus expected to) organize lower-level network resources to achieve application-specific design goals efficiently. (application autonomy)

· Lower-level layers, which support many independent applications, should provide only resources of broad utility across applications, while providing to applications usable means for effective sharing of resources and resolution of resource conflicts. (network transparency)

While making lower layers more active or programmable is likely to enhance application autonomy, the risk is that programmable lower layers may reduce network transparency. The reason is that a key element of transparency is some ability to predict how the network will behave.

Will ISPs Adopt?

· If ISPs aren’t willing to adopt individual services (e.g., multicast), why would they be willing to adopt Active Networking?

· If ISPs aren’t willing, then deploying Active Nodes at the application level is merely another form of an overlay network

· If ISPs are willing to deploy code, then why isn’t this just a case of programmable routers

What is the future of Internet architecture?

· 294 next year on Internet architecture

· Will there be significant changes in the next five years?