CIS 700/005
Networking Meets Databases
Boon Thau Loo
Spring 2007
Lecture 15

Announcement
- 2 minute informal spiel on class project on Thursday
- Will miss class on:
  - Apr 5: DARPA meeting

Era of change for the Internet

"in the thirty-odd years since its invention, new uses and abuses, ..., are pushing the Internet into realms that its original design neither anticipated nor easily accommodates...."

Overcoming Barriers to Disruptive Innovation in Networking, NSF Workshop Report '03

Efforts at Internet Innovation

- Evolution: Overlay Networks
  - Commercial (Akamai, VPN, MS Exchange servers)
  - P2P (filesharing, telephony)
  - Research prototypes on testbed (PlanetLab)
- Revolution: Clean-slate design
  - NSF Future Internet Design (FIND) program
  - NSF Global Environment for Network Investigations (GENI) initiative

Overlay

Revolution: Clean-slate design

Internet

Context of ROFL paper

- Up to this point in class:
  - Distributed Hash tables
  - Network Data Independence ⇒ Decouple location from identity
  - Overlay networks
  - Internet Indirection Infrastructure (i3)
  - Active networks
  - Extremer solution
  - Declarative networks / PLAN / ANTs
  - "Safe languages" of active networks
  - Clean-slate radical architectural changes
    - TRAID, IPNL, LFN/DOA, HIP, SNF, ROFL,...
When Addresses rule the Net

- In the ARPANET, addresses were fixed and could actually be used as "long-lived names"
- Address 1 at UCLA-NMC (RFC-597, 1973)
- HOST.TXT file is mapping a static symbol into another static symbol
- Same with the early Internet
- 1.0.0.0/8 is IN-PR (RFC-820, 1982)
- After the introduction of DNS in 1984 addresses are still very long-lived:
  - RFC9 890 (1986) is the last one to contain a list of assigned addresses
  - estimated number of hosts ~ 1,000
  - RFC-990 (1986) is the last one to contain a list of assigned addresses
  - estimated number of hosts < 10,000

Internet Addressing Scheme (Review)

- Class-based addressing schemes:
  - 32 bits divided into 2 parts:
    - Class A
    - 255 network bits
    - 32 network bits
    - 0.0.0.0-126
    - ~16K hosts
  - Class B
    - network bits
    - 8 network bits
    - 254 hosts
    - ~16K addresses
  - Class C
    - network bits
    - 24 network bits
    - 255 hosts
    - ~254 addresses

- Classless Inter-domain Routing (CIDR)
  - Variable prefix – prevents wastage
  - Prefix aggregation – lower overhead of inter-domain routing
  - Routers match to longest prefix

One of the Key Goals of “Clean-slate”:
Evolution of Networks

- How much of an architecture can evolve over time?
- hypothesis: no more "one size fits all" network
- what must remain stable? what is the least common denominator, i.e.
  - the meta-arch?

* Common question: must addresses be explicitly defined/supported by a network?*
  - architecture? (should it be address-centric?)

**Scalability: Routing Table Size**

Without CIDR:
- 232.71.0.0
- 232.71.1.0
- 232.71.2.0
- 232.71.255.0

With CIDR:
- 232.71.0.0/16
- 232.71.1.0/24

---

**What’s wrong with Internet addressing today?**

- Hierarchical addressing allows excellent scaling properties
- But, forces addressing to conform to network topology
- Since topology is not static, addresses can’t persistently identify hosts
- Host can have multiple locations (mobility)
- Host can have multiple identities (multi-homing)

---

**What’s wrong with Internet addressing today?**

- When the Internet was first invented:
  - Nodes are mostly static, so location = identity.
  - Easy to impose hierarchy
- But most network applications today require persistent identity in the presence of changing/multiple locations.
- It’s hard to provide persistent identity in presence of hierarchical addressing
  - Need to decouple identity from addressing
  - Drastically complicates network configuration, mobility, address assignment
- Seems difficult to achieve...

---

**Solutions we have studied…**

- Level of indirection
  - Location-identity split
  - Name resolution
    - Route using a name
    - Name turns into location IP address
    - Location is ephemeral, name is long-term identifier
  - E.g. Mobile IP, Internet Indirection Infrastructure
ROFL: Is there an alternative?

- Why not route on "flat" host identifiers?
  - Assign addresses independently of network topology
  - Get rid of location and hierarchy altogether!!
  - Be a Purist: Architectural uniformity to the extreme.

- Benefits:
  - No separate name resolution system required
  - Simpler network config/allocation/mobility
    - Allocation of identifiers need only ensure uniqueness
  - Need not worry about adherence to network topology
  - Simpler (more meaningful) network-layer access controls
    - Identifier-based instead of IP-based

Is it possible to route on flat identifiers?

- Challenge: flat identifiers break aggregation
- Is it possible to scalably route without aggregation?
  - Paper is more about addressing the feasibility of flat naming without hierarchy, and generating discussions.
  - "Postman delivers using SSN, not address"
- Not a straightforward application of DHTs
  - Assumes point-to-point routing. Cannot address the problem of building "from scratch" a network
  - Doesn’t support routing policies

OSI vs. Internet

- OSI: conceptually define services, interfaces, protocols
- Internet: provide a successful implementation

<table>
<thead>
<tr>
<th>OSI (formal)</th>
<th>Internet (informal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
<td>TCP</td>
</tr>
<tr>
<td>Transport</td>
<td>UDP</td>
</tr>
<tr>
<td>Network</td>
<td>IP</td>
</tr>
<tr>
<td>Datalink</td>
<td>Net access/physical</td>
</tr>
<tr>
<td>Physical</td>
<td>Physical</td>
</tr>
</tbody>
</table>

Basic idea behind ROFL

- Scalable routing on flat identifiers

- Goal #1: Scale to Internet topologies
  - Mechanism: DHT-style routing, maintain source-routes to successors/fingers
  - Provides: Scalable network routing without aggregation

- Goal #2: Support for BGP policies
  - Mechanism: Intelligently choose successors/fingers to conform to ISP relationships
  - Provides: Support for policies, operational model of BGP

How ROFL works

- Goal #1: Scale to Internet topologies
  - Mechanism: DHT-style routing, maintain source-routes to successors/fingers
  - Provides: Scalable network routing without aggregation

- Goal #2: Support for BGP policies
  - Mechanism: Intelligently choose successors/fingers to conform to ISP relationships
  - Provides: Support for policies, operational model of BGP

Basic mechanisms behind ROFL
**Internet policies today**

- Economic relationships: peer, provider/customer
- Isolation: routing contained within hierarchy

**Canon DHT (background)**

- Basis for Inter-domain routing in ROFL
  - Merge rings in a bottom-up fashion with two conditions.
  - For an identifier $i_d$ in ring 1 with external pointer to $i_d$ in ring 2:
    - No identifiers between $[i_d, i_d]$.
    - $i_d$ will be $i_d$'s successor.
    - $O(\log n)$ number of pointers.

**Isolation in ROFL (Canon)**

- Traffic between two hosts traverses no higher than their lowest common provider in the AS hierarchy.

**Policy support in ROFL**

- Peering
- Provider-customer
- Backup

**Scalability in ROFL**

- Two extensions to improve locality:
  - Maintain proximity-based fingers in a policy-safe fashion
  - Pointer caching strategies: prefer nearby, popular pointers

**More Routing Concerns**

- Routing control
  - Inter-domain: endpoint-based negotiation
  - Intra-domain: leverage the inter-domain design (Isolation property)
- Enhanced delivery services
  - Anycast
  - Multicast
- Security
  - Default off (Hotnets '05) – Hosts reachable only from fingers
  - Use capabilities (a capability is a cryptographic token designating that a particular source is allowed to contact the destination)
Evaluation

- Distributed packet-level simulations
  - Deployed on cluster across 62 machines, scaled to 300 million hosts
  - Inferred Internet topology from Routeviews, Rocketfuel, CAIDA skitter traces
- Implementation
  - Ran on Planetlab as overlay, covering 82 ASes
  - Configured inter-ISP policies from Routeviews traces
- Metrics: stretch, control overhead, router memory usage

Enterprise/ISP simulations: Load balance

- No “hot spots” introduced by protocol

Enterprise/ISP simulations: Reduction in churn

- 37 to 181x less control overhead
- 34-1200x less memory requirements

Internet-scale simulations

- Join overhead <300 msgs, stretch < 1.4
- Root-server lookups inflate latency from 54ms to 134ms, Flat IDs has no penalty

<table>
<thead>
<tr>
<th>Metric</th>
<th>ROFL</th>
<th>BGP+DNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join overhead</td>
<td>496 per host</td>
<td>Typically 9 per host</td>
</tr>
<tr>
<td></td>
<td>“Lightweight maps”; 14 per host</td>
<td>40,000 per prefix</td>
</tr>
<tr>
<td>Latency</td>
<td>1.37ms</td>
<td>76ms</td>
</tr>
<tr>
<td></td>
<td>With pointer caches: 1.37ms</td>
<td>No lookup: 54ms</td>
</tr>
<tr>
<td>State</td>
<td>2 million pointer cache entries in core, 150 thousand entries at core and edge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>150 thousand entries at core and edge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>77 million DNS entries at root servers</td>
<td></td>
</tr>
<tr>
<td>Failure</td>
<td>Cashed pointers equivalent of backup paths; failures only affect successors and</td>
<td>Undergoes convergence process; failures have global impact</td>
</tr>
</tbody>
</table>

Conclusion

- Routing On Flat Labels should not leave you Rolling On The Floor Laughing
  - Because performance is tolerable
  - Because it provides several benefits
- ROFL is one point in the design space
  - ROFL as lookup
  - ROFL for content routing
    - Better support for ONYX, INS, PIER, P2, …?
  - “The results are close enough to tempt, but not enough to satisfy”
- Later in the semester:
  - Virtual Ring Routing – DHT on wireless networks
  - Similar to ROFL’s Inter-domain design