CS 268: Lecture 24
Internet Architectures: i3, DOA, HIP,...

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

Outline
- Internet Indirection Infrastructure (i3)
  - Design comparison
    - Host Identity Protocol
    - i3
    - Semantic Free References (SFR)
    - Delegation Oriented Architecture (DOA)
  - Another view of indirection/delegation

Motivations
- Today’s Internet is built around a unicast point-to-point communication abstraction:
  - Send packet “p” from host “A” to host “B”
- This abstraction allows Internet to be highly scalable and efficient, but...
- … not appropriate for applications that require other communications primitives:
  - Multicast
  - Anycast
  - Mobility
  - Service composition (Middleboxes)
  - …

Why?
- Point-to-point communication → implicitly assumes there is one sender and one receiver, and that they are placed at fixed and well-known locations
  - E.g., a host identified by the IP address 128.32.xxx.xxx is located in Berkeley

IP Solutions
- Extend IP to support new communication primitives, e.g.,
  - Mobile IP
  - IP multicast
  - IP anycast
- Disadvantages:
  - Difficult to implement while maintaining Internet’s scalability (e.g., multicast)
  - Require community wide consensus -- hard to achieve in practice

Application Level Solutions
- Implement the required functionality at the application level, e.g.,
  - Application level multicast (e.g., Fastforward, Narada, Overcast, Scattercast…)
  - Application level mobility
- Disadvantages:
  - Efficiency hard to achieve
  - Redundancy: each application implements the same functionality over and over again
  - No synergy: each application implements usually only one service; services hard to combine
**Key Observation**

- Virtually all previous proposals use indirection!

> “Any problem in computer science can be solved by adding a layer of indirection”

**Internet Indirection Infrastructure (i3)**

- Build an efficient indirection layer on top of IP
- Use an overlay network to implement this layer
  - Incrementally deployable; don’t need to change IP

**Service Model**

- API
  - sendPacket(r);
  - insertTrigger(i);
  - removeTrigger(i) // optional
- Best-effort service model (like IP)
- Triggers periodically refreshed by end-hosts
- ID length: 256 bits

**Mobility**

- Host just needs to update its trigger as it moves from one subnet to another
Multicast

- Receivers insert triggers with same identifier
- Can dynamically switch between multicast and unicast

Service Composition: Sender Initiated

- Use a stack of IDs to encode sequence of operations to be performed on data path
- Advantages
  - Don’t need to configure path
  - Load balancing and robustness easy to achieve

Outline

- Implementation
  - Examples
  - Security
  - Architecture Optimizations
  - Applications

Quick Implementation Overview

- i3 is implemented on top of Chord
- Each trigger \( t = (id, R) \) is stored on the node responsible for \( id \)
- Use Chord recursive routing to find best matching trigger for packet \( p = (id, data) \)
Routing Example

- R inserts trigger \( t = (37, R) \); S sends packet \( p = (37, \text{data}) \)
- An end-host needs to know only one i3 node to use i3
  - E.g., S knows node 3, R knows node 35

Optimization #1: Path Length

- Sender/receiver caches i3 node mapping a specific ID
- Subsequent packets are sent via one i3 node

Optimization #2: Triangular Routing

- Use well-known trigger for initial rendezvous
- Exchange a pair of (private) triggers well-located
- Use private triggers to send data traffic

Outline

- Implementation
  - Examples
    - Heterogeneous multicast
    - Scalable Multicast
    - Load balancing
    - Proximity

Example 1: Heterogeneous Multicast

- Sender not aware of transformations

Example 2: Scalable Multicast

- i3 doesn’t provide direct support for scalable multicast
  - Triggers with same identifier are mapped onto the same i3 node
- Solution: have end-hosts build an hierarchy of trigger of bounded degree
Example 2: Scalable Multicast (Discussion)

Unlike IP multicast, i3
1. Implement only small scale replication → allow infrastructure to remain simple, robust, and scalable
2. Gives end-hosts control on routing → enable end-hosts to
   - Achieve scalability, and
   - Optimize tree construction to match their needs, e.g., delay, bandwidth

Example 3: Load Balancing

• Servers insert triggers with IDs that have random suffixes
• Clients send packets with IDs that have random suffixes

Example 4: Proximity

• Suffixes of trigger and packet IDs encode the server and client locations

Example 5: Protection against DOS Attacks

• Problem scenario: attacker floods the incoming link of the victim
• Solution: stop attacking traffic before it arrives at the incoming link
  – Today: call the ISP to stop the traffic, and hope for the best!
• Approach: give end-host control on what packets they receive
  – Enable end-hosts to stop the attacks in the network

Example 5: Why End-Hosts (and not Network)?

• End-hosts can better react to an attack
  – Aware of semantics of traffic they receive
  – Know what traffic they want to protect
• End-hosts may be in a better position to detect an attack
  – Flash-crowd vs. DoS

Example 5: Some Useful Defenses

1. White-listing: avoid receiving packets on arbitrary ports
2. Traffic isolation:
  – Contain the traffic of an application under attack
  – Protect the traffic of established connections
3. Throttling new connections: control the rate at which new connections are opened (per sender)
Example 5: (1) White-listing
• Packets not addressed to open ports are dropped in the network
  – Create a public trigger for each port in the white list
  – Allocate a private trigger for each new connection

Example 5: (2) Traffic Isolation
• Drop triggers being flooded without affecting other triggers
  – Protect ongoing connections from new connection requests
  – Protect a service from an attack on another services

Example 5: (3) Throttling New Connections
• Redirect new connection requests to a gatekeeper
  – Gatekeeper has more resources than victim
  – Can be provided as a 3rd party service

Outline
• Internet Indirection Infrastructure (I3)
  ➢ Design comparison
    ➢ Host Identity Protocol
    – I3
    – Semantic Free References (SFR)
    – Delegation Oriented Architecture (DOA)
• Another view of indirection/delegation

Host Identity Protocol (HIP)
• Provides:
  – Fast mobility
  – Multi-homing
  – Support for different addressing schemes
    • Transparent IPv4 to IPv6 migration
  – Security
    • Anonymity
    • Secure and authenticate datagrams
**HIP**

- A public key used to identify an end-host
- A 128-bit host identity tag (HIT) used for system calls
  - HIT is a hash on public key
  - Global scope
- A 32-bit local scope identifier (LSI) for IPv4 compatibility

HIT replaces IP address as a name of a system

**Protocol Stack**

**How It Works?**

**Outline**

- Internet Indirection Infrastructure (i3)
  - Design comparison
    - Host Identity Protocol
    - i3
    - Semantic Free References (SFR)
    - Delegation Oriented Architecture (DOA)
- Another view of indirection/delegation

**i3 Identifiers**

- 256-bit IDs
- ID maps to another ID or to an (IPaddr:Port)
  - (IPaddr:Port) points to an application layer demultiplexer
- ID can represent
  - A host, flow, service, etc

ID can identify any entity
Outline

- Internet Indirection Infrastructure (i3)
  - Design comparison
    - Host Identity Protocol
    - i3
    - Semantic Free References (SFR)
  - Delegation Oriented Architecture (DOA)
- Another view of indirection/delegation

Goal: Address DNS Limitations

- DNS names identify machines and organizations, not data
  - Data cannot be easily moved
  - Data cannot be easily replicated
- DNS names are brand names
  - Political fighting

SFR Solution

- Use IDs instead of DNS name
- ID space is flat and IDs have no semantics
- A generalization of DNS
  - Returns metadata instead of an IP address
- How to implement it?
  - Use distributed hash-tables (DHTs)

Delegation Oriented Architecture (DOA)

- Supports:
  - Mobility
  - Multi-homing
- Integrate middle-boxes
- Security (through middle-boxes)
  - Anonymity
  - DoS
  - …
An Old Naming Taxonomy

• Four kinds of network entities (Saltzer):
  – Services (and data)
  – Hosts (endpoints)
  – Network attachment points
  – Paths

• Should name each individually:
  – Ignore paths (router involvement)
  – IP addresses name attachment points
  – Endpoint identifiers (EIDs) name hosts
  – Service identifiers (SIDs) name services/data

Protocol Stack

How It Works?

Principles

• Don’t bind to lower-level IDs prematurely
  – Host mobility and renumbering (HIP)
  – Service and data migration
• Resolution of name need not point to object itself, but can point to its delegate
  – Resolution can point to intermediaries who process packets on behalf of the named target

Naming (Indirection) Architecture Requirements

1) There should be a layer in the protocol stack that uses IDs not IP addresses
   • Mobility, multi-homing, replications, …
2) IDs should be able to name arbitrary objects
3) IDs should encode as little semantics as possible
4) End-points should be able to use indirection at the ID level
   • Integrate middle boxes

How Many ID Layers?

• HIP: one layer; IDs identify machines
• SFR: one layer; IDs identify data
• i3: one layer; IDs identify arbitrary objects
• DOA: two layers
  – EIDs identify machines
  – SIDs identify everything else
Where is the Resolution ID→IP Done?

- SFR: above transport
- HIP: below transport, at HIP layer
- i3: in the infrastructure
- DOA: above & below transport
  - But IP address can be an intermediate point

Security Support?

- HIP:
  - Authentication, data integrity
  - Anonymity at transport layer
  - Transport layer resistance to DoS attacks
- i3
  - Anonymity at IP layer
  - Some DoS defense at IP layer
  - Everything else can be done though middle-boxes
- DOA
  - Everything can be done through middle-boxes

Outline

- Internet Indirection Infrastructure (i3)
- Design comparison
  - Host Identity Protocol
  - i3
  - Semantic Free References (SFR)
  - Delegation Oriented Architecture (DOA)
  - Another view of indirection/delegation

Another View of Indirection/Delegation

- Indirection point → point where control is transferred from sender to receiver
  - Translation/forwarding entry usually controlled by receiver

End-host Empowerment

- Both the sender and receiver are able to explicitly control the service-path
  - Note: multiple indirection points possible

Realization: i3 vs DOA