Internet Indirection Infrastructure

Ion Stoica
UC Berkeley

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
  - …

2

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

3

Key Observation

- Virtually all previous proposals use indirection, e.g.,
  - Physical indirection point → mobile IP
  - Logical indirection point → IP multicast

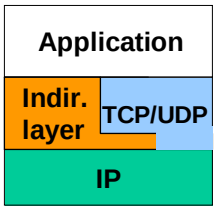
"Any problem in computer science can be solved by adding a layer of indirection"

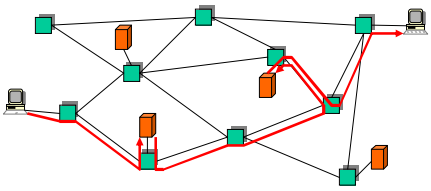
4

Our Solution

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

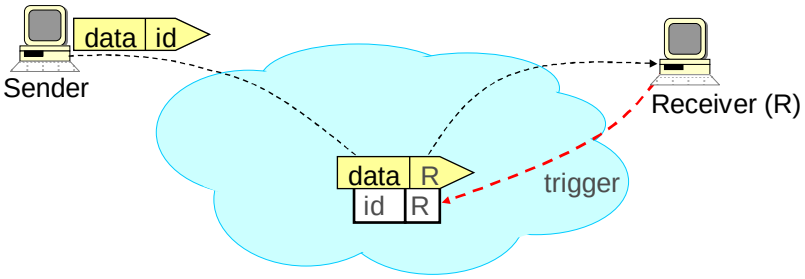




5

Internet Indirection Infrastructure (i3)

- Each packet is associated an identifier ***id***
- To receive a packet with identifier ***id***, receiver R maintains a trigger (***id***, R) into the overlay network



6

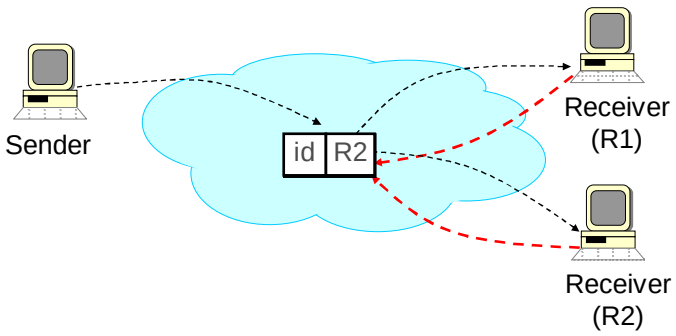
Service Model

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

7

Mobility

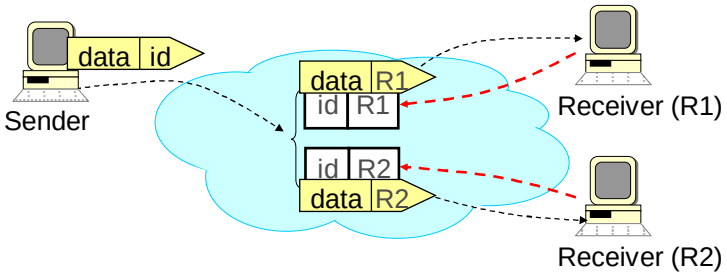
- Host just needs to update its trigger as it moves from one subnet to another



8

Multicast

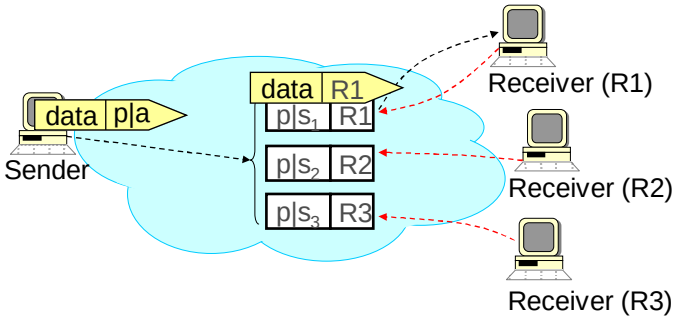
- Receivers insert triggers with same identifier
- Can dynamically switch between multicast and unicast



9

Anycast

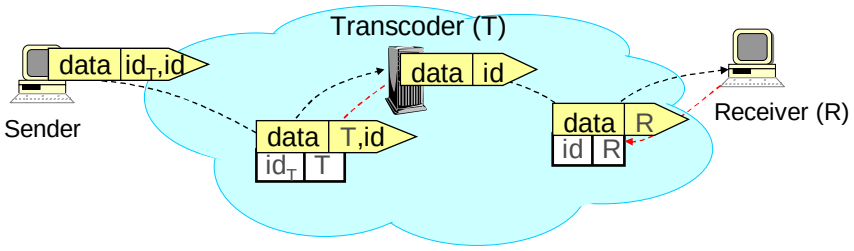
- Use longest prefix matching instead of exact matching
  - Prefix p: anycast group identifier
  - Suffix s_i: encode application semantics, e.g., location



10

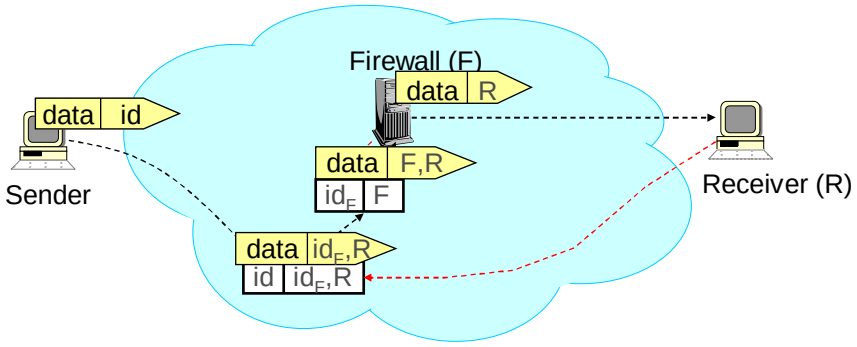
Service Composition: Sender Initiated

- 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



Service Composition: Receiver Initiated

- Receiver can also specify the operations to be performed on data



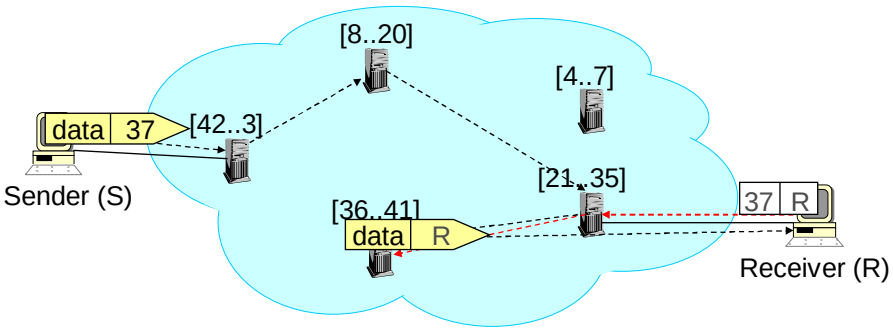
Quick Implementation Overview

- ID space is partitioned across infrastructure nodes
  - Each node responsible for a region of ID space
- Each trigger **(*id, R*)** is stored at the node responsible for ***id***
- Use Chord to route triggers and packets to nodes responsible for their IDs
  - O(log N) hops

13

Example

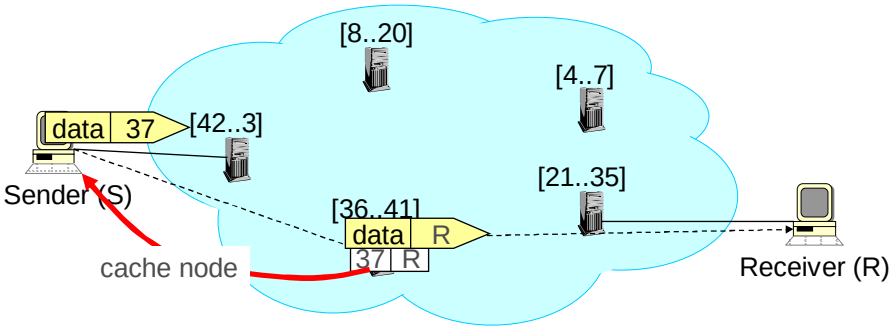
- ID space [0..63] partitioned across five i3 nodes
- Each host knows one i3 node
- **R** inserts trigger **(*37, R*)**; **S** sends packet **(*37, data*)**



14

Optimization: Path Length

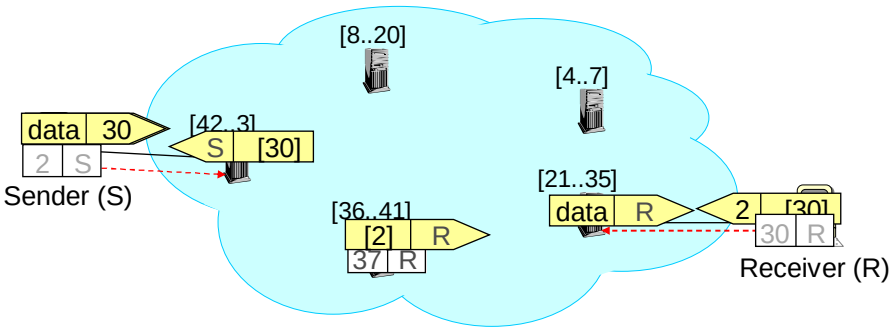
- Sender/receiver caches i3 node mapping a specific ID
- Subsequent packets are sent via one i3 node



15

Optimization: Triangular Routing

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



16

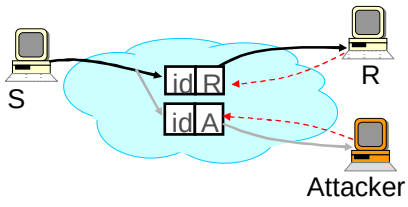
Outline

- Overview
- Security
- Discussion

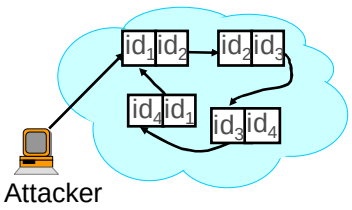
17

Some Attacks

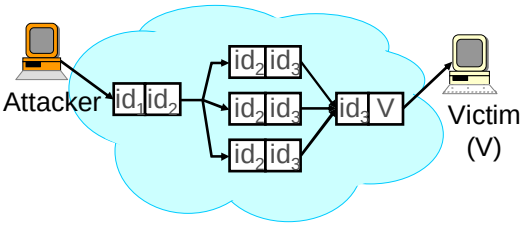
Eavesdropping



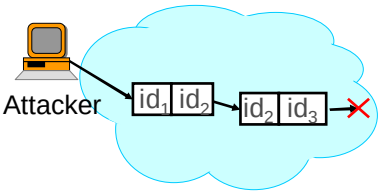
Loop



Confluence



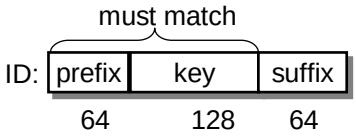
Dead-End

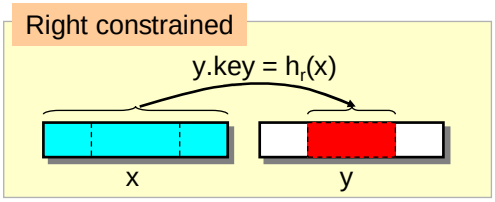


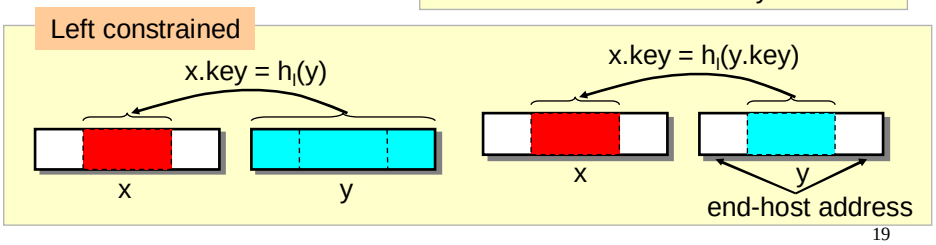
18

Constrained Triggers

- $h_l()$, $h_r()$: well-known one-way hash functions
- Use $h_l()$, $h_r()$ to constrain trigger (x, y)







19

Attacks & Defenses

Defense \ Attack	Trigger constraints	Pushback	Trigger challenges	Public i3 node constraints
Eavesdropping& Impersonation	✓			✓
Loops & Confluences	✓			
Dead-ends		✓		
Reflection & Malicious trigger-removal	✓		✓	
Confluences on i3 public nodes				✓

20

Outline

- Overview
- Security
- Discussion

21

Design Principles

1) Give hosts control on routing
   - A trigger is like an entry in a routing table!
   - Flexibility, customization
   - End-hosts can
     - Source route
     - Set-up acyclic communication graphs
     - Route packets through desired service points
     - Stop flows in infrastructure
     - …
2) Implement data forwarding in infrastructure
   - Efficiency, scalability

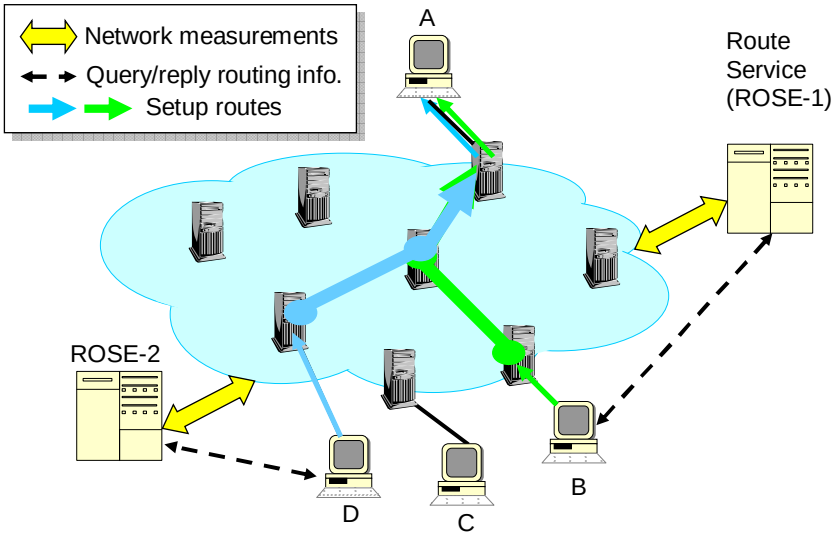
22

Design Principles (cont'd)

	Host	Infrastructure
Internet & Infrastructure overlays		Data plane Control plane
p2p & End-host overlays	Data plane Control plane	
i3	Control plane	Data plane

23

Example: Application Specific Routing



24

Conclusions

- Indirection – key technique to implement basic communication abstractions
  - Multicast, Anycast, Mobility, …
- This research
  - Advocates for building an efficient Indirection Layer on top of IP
  - Explore the implications of changing the communication abstraction; already done in other fields
    - Direct addressable vs. associative memories
    - Point-to-point communication vs. Tuple space (in Distributed systems)

25