

# **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
  - Mobility
  - Anycast
  - Service composition

# 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

# Key Observation

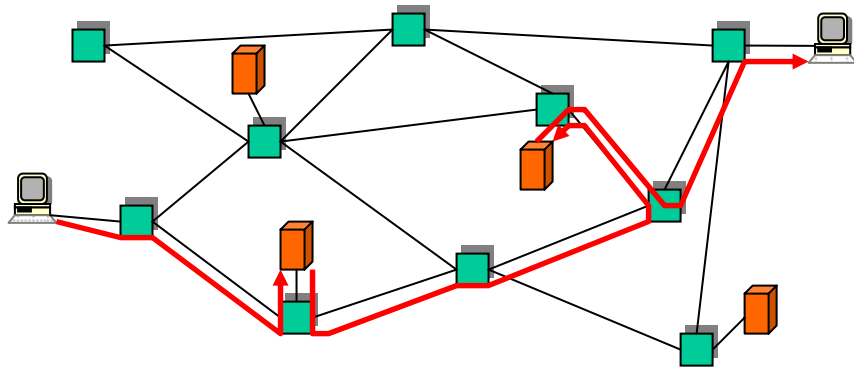
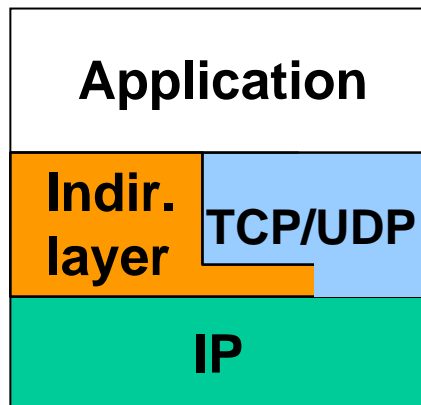
- Virtually all previous proposals use indirection, e.g.,
  - Physical indirection point → mobile IP
  - Logical indirection point → IP multicast

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

# Our Solution

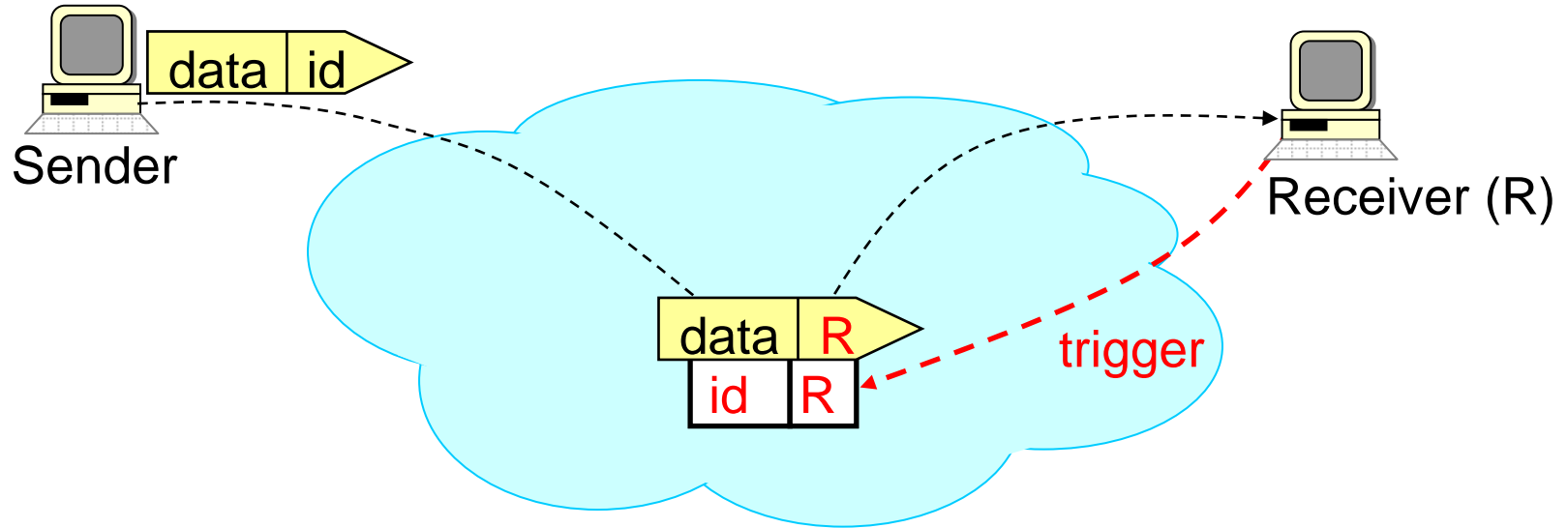
Build an efficient indirection layer  
on top of IP

- Use an **overlay** network to implement this layer
  - Incrementally deployable; don't need to change IP



# 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

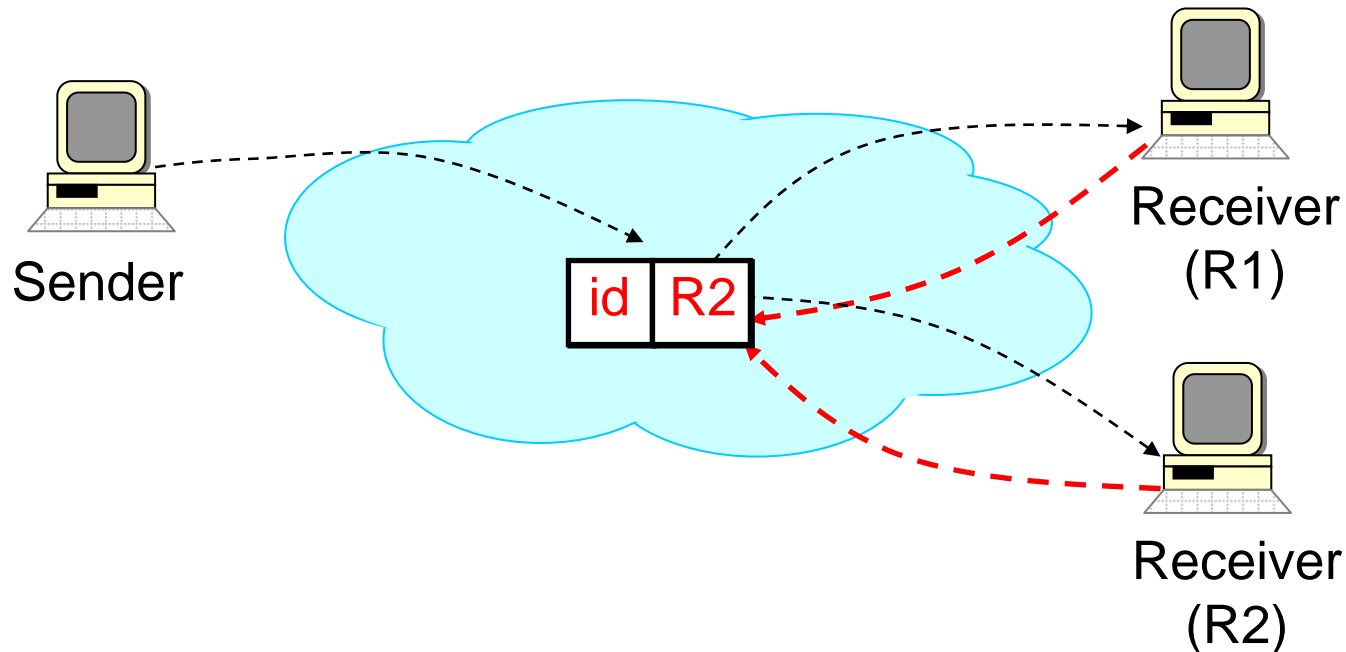


# Service Model

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

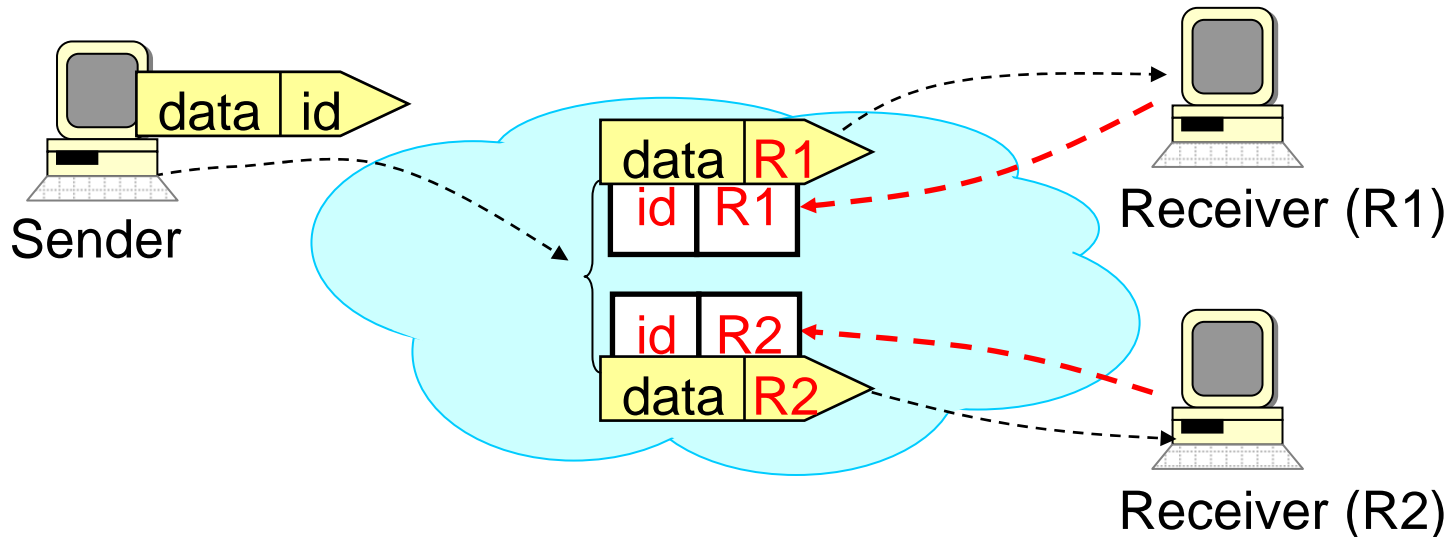
# Mobility

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



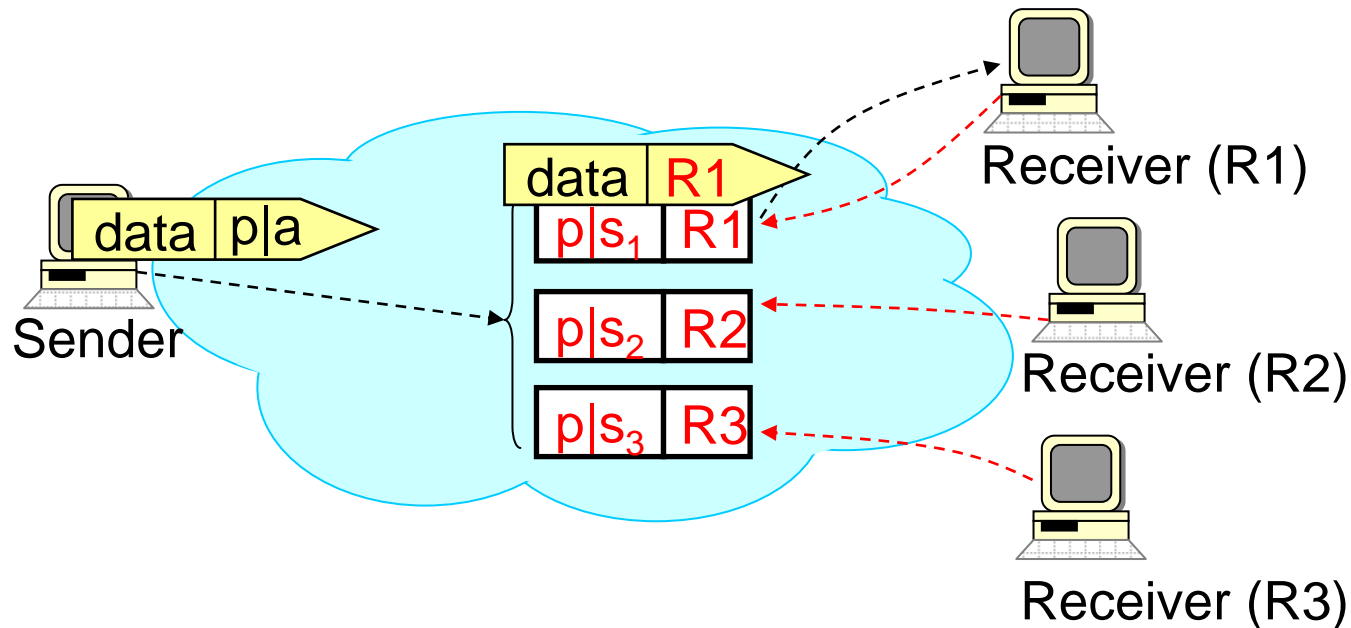
# Multicast

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



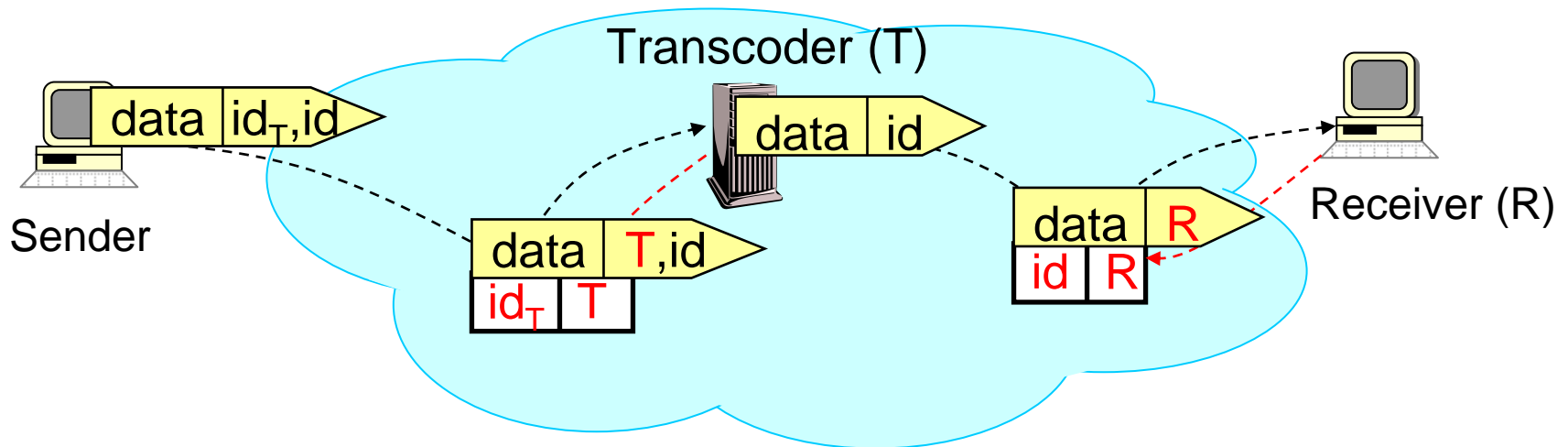
# Anycast

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



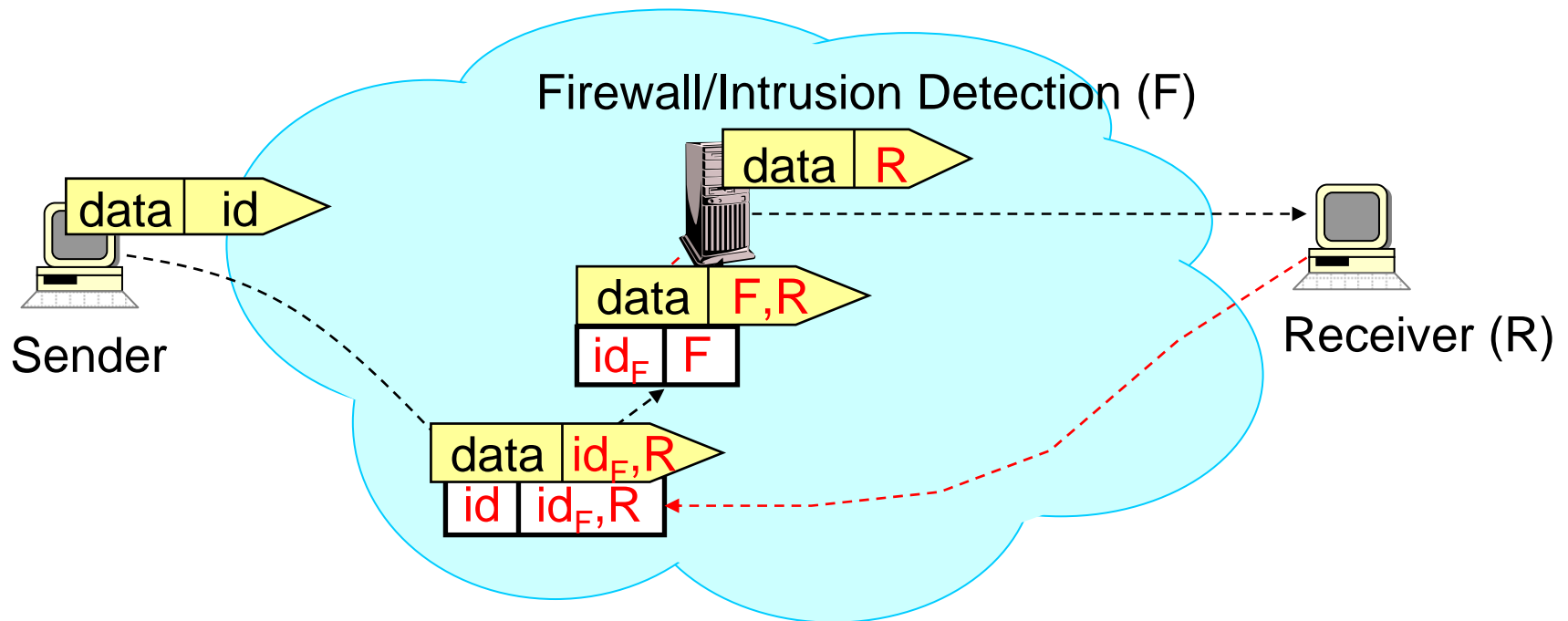
# Service Composition: Sender Initiated

- 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



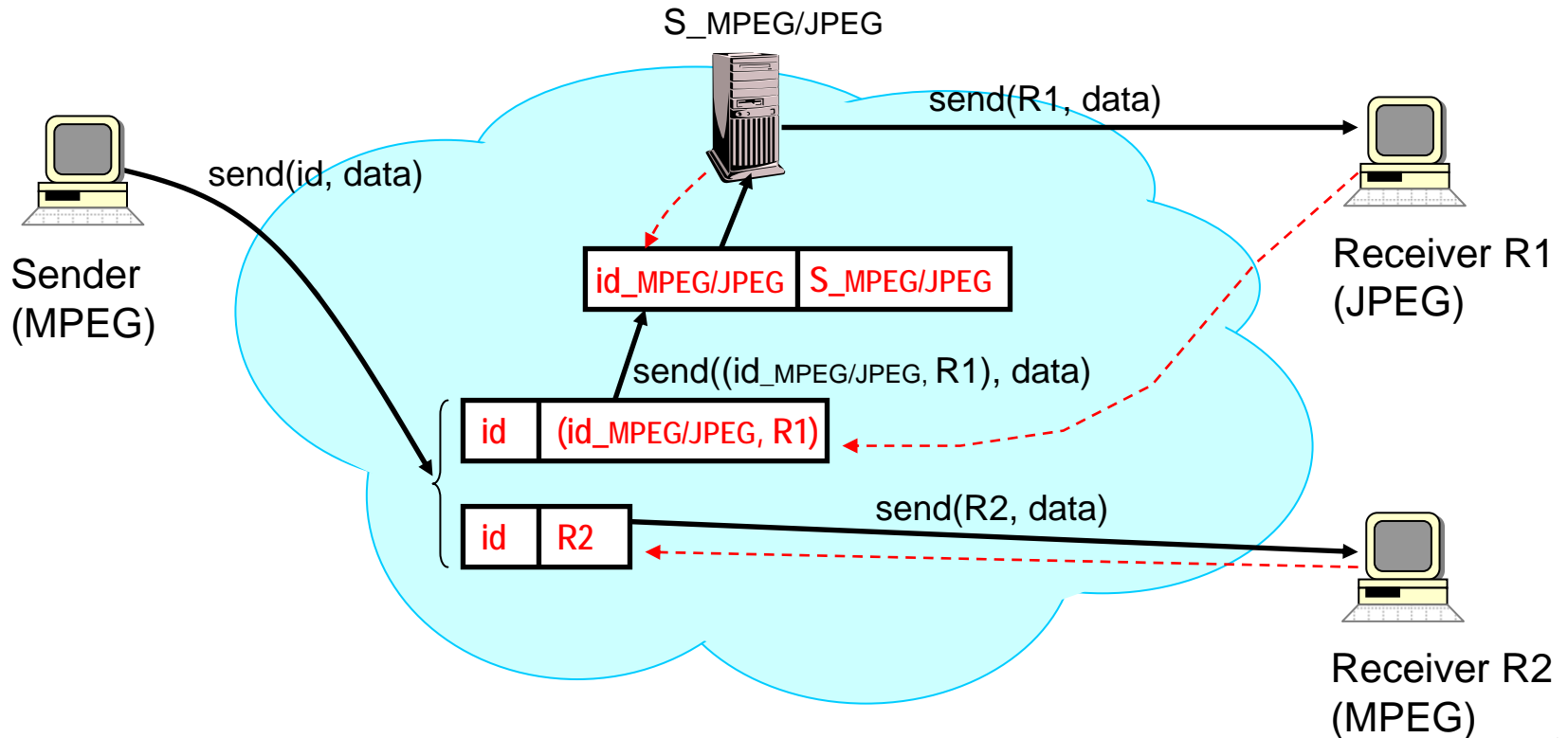
# Service Composition: Receiver Initiated

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



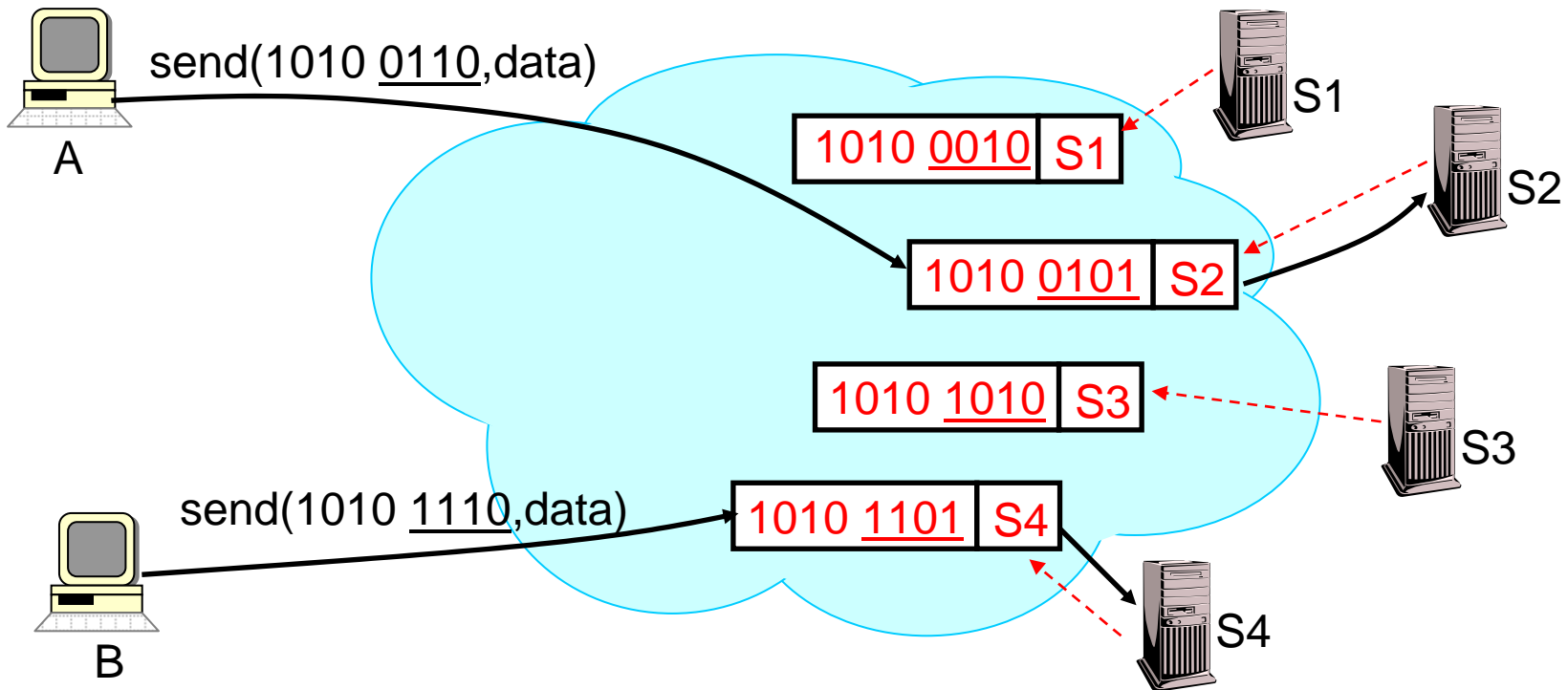
# Example 1: Heterogeneous Multicast

- Sender not aware of transformations



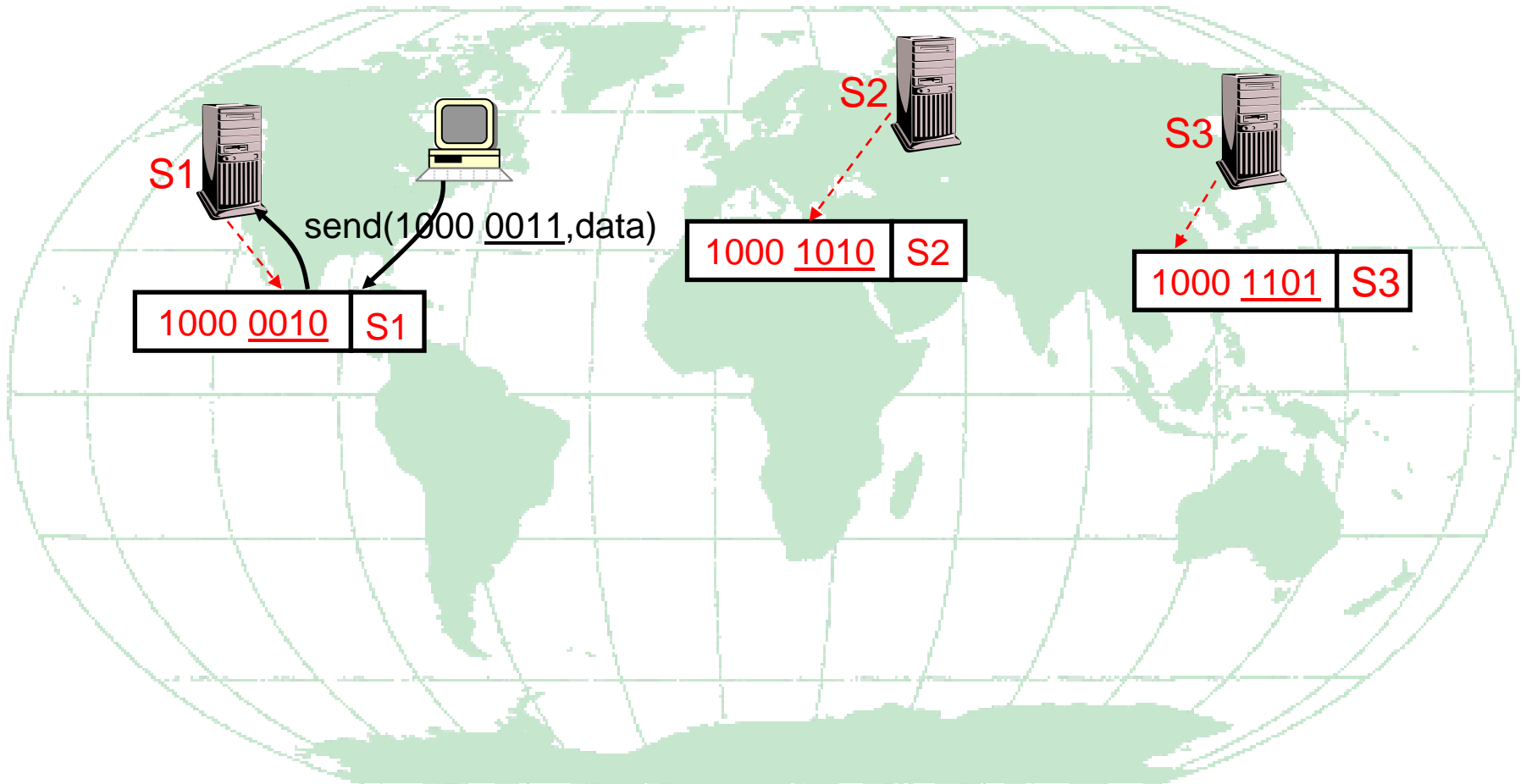
# Example 2: Load Balancing

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

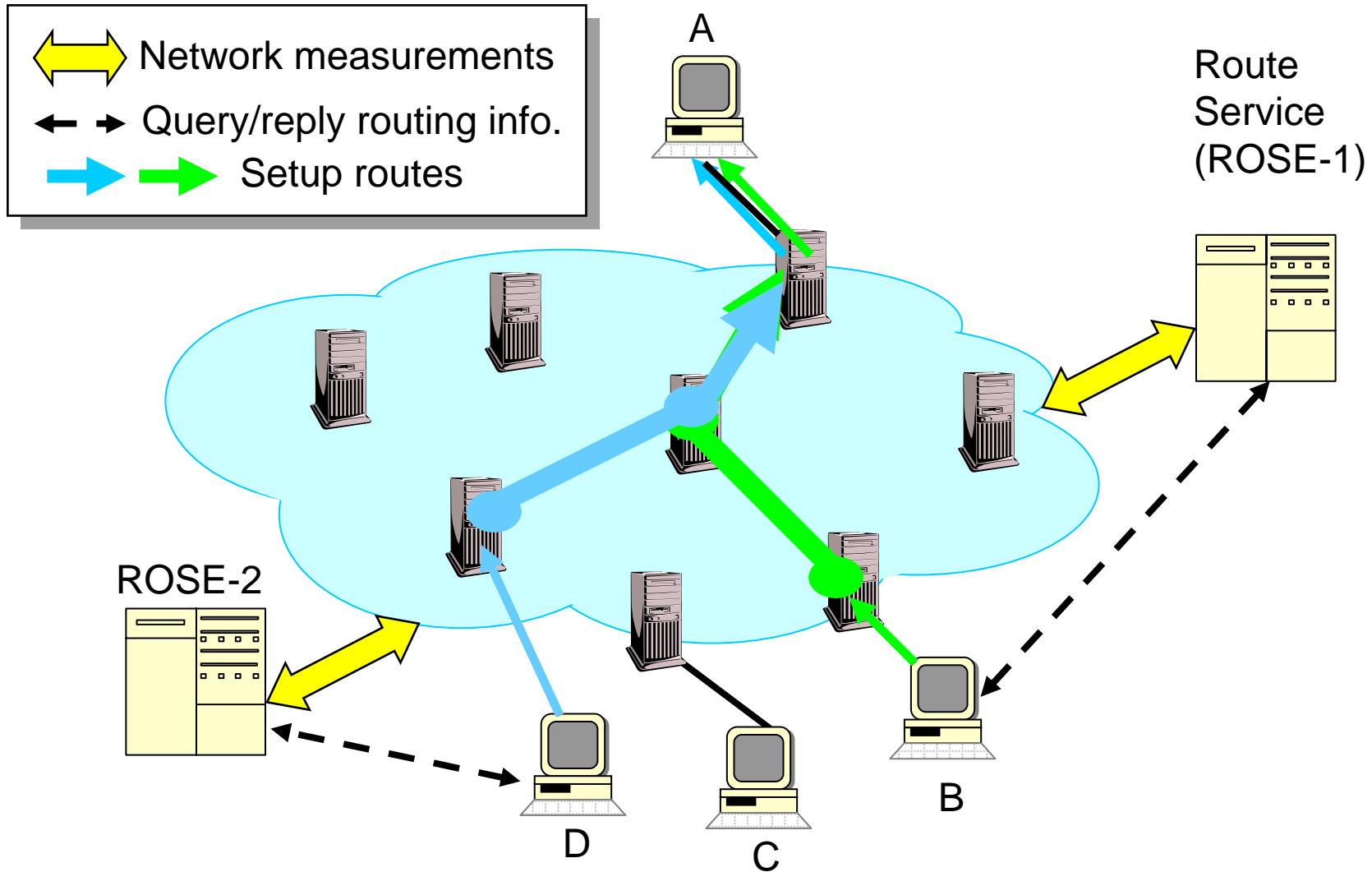


# Example 3: Proximity

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



# Example 4: Application Specific Routing



# What Can i3 Do For VoIP?

- Provide native support for
  - Mobility
  - Multicast
- Use service composition to allow third-party services to provide
  - Improved security, e.g., compression, anonymity
  - QoS, e.g., aggressive error correction and coding
  - Language translation
- Can reach nodes behind NATs/Firewalls

# Conclusions

- Indirection – key technique to implement basic communication abstractions
  - Multicast, Anycast, Mobility, ...
- Internet Indirection Infrastructure
  - Platform to deploy new services in the Internet
  - Highly flexible **forwarding** infrastructure: allows end-hosts and third-parties to choose their own routes!

# Status

- i3 available as a service on Planetlab
- Support for legacy applications in Linux and Windows XP
- Current applications
  - Mobility
  - Transparent access to machines behind NATs
  - Secure and transparent access to services behind firewalls

Available: <http://i3.cs.berkeley.edu>