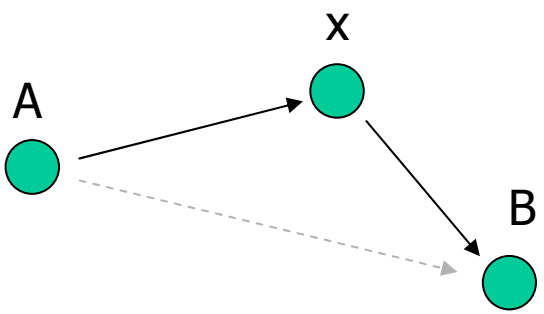
Indirection

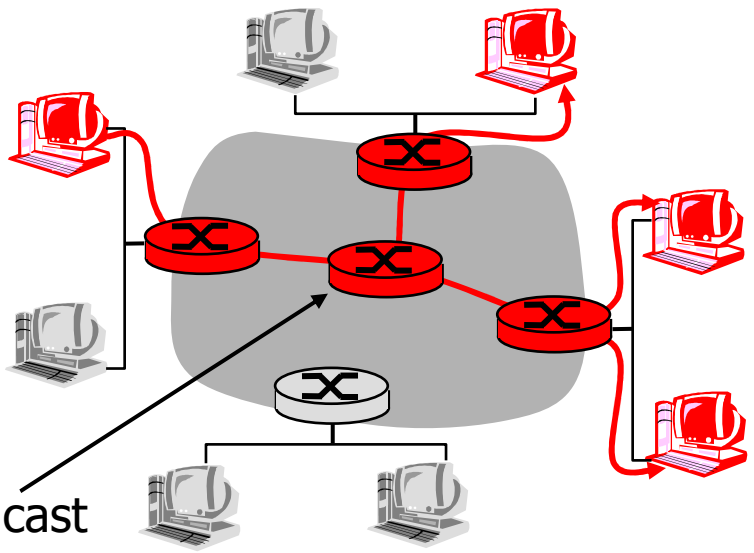
Indirection: rather than reference an entity directly, reference it ("indirectly") via another entity, which in turn can or will access the original entity



"Every problem in computer science can be solved by adding another level of indirection"
-- Butler Lampson

Multicast: one sender to many receivers

- Multicast: Act of sending datagram to multiple receivers with single "transmit" operation
  - Analogy: One teacher to many students
- Question: How to achieve multicast

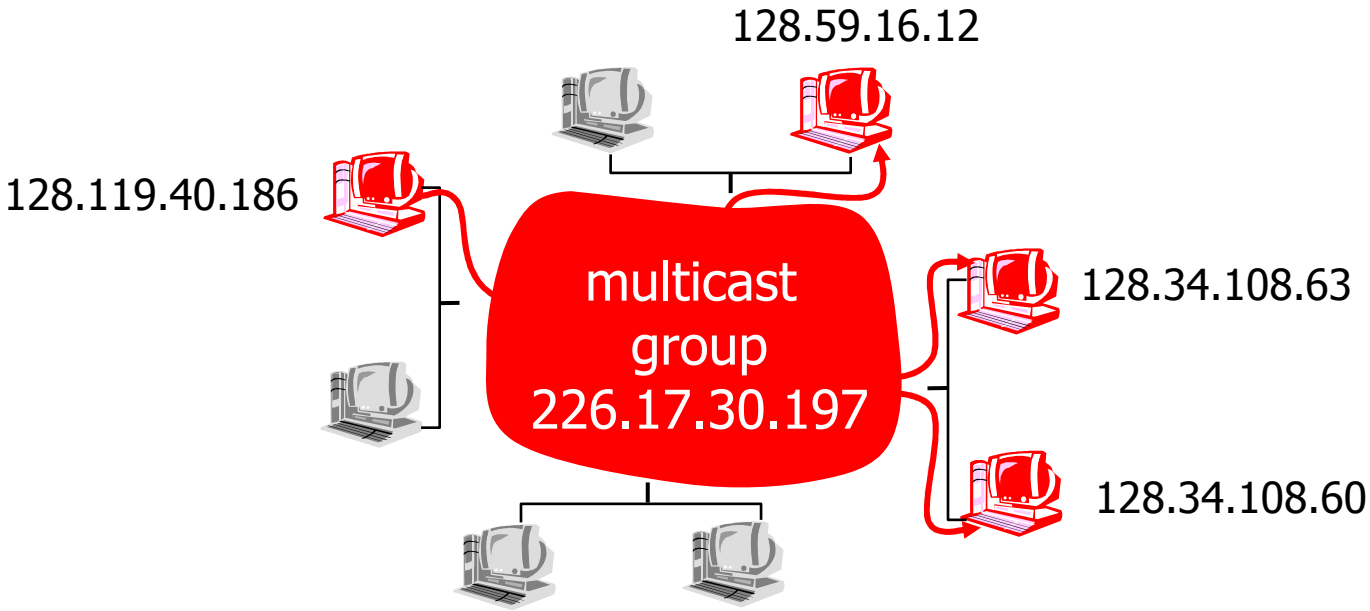


Multicast routers (red) duplicate and forward multicast datagrams

Network multicast

- Router actively participate in multicast, making copies of packets as needed and forwarding towards multicast receivers

Internet Multicast Service Model

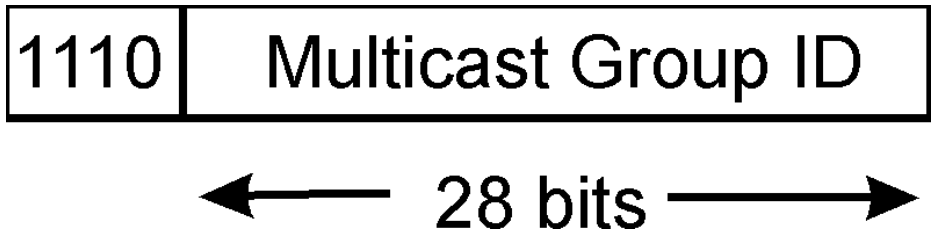


multicast group concept: use of indirection

- hosts addresses IP datagram to multicast group
- routers forward multicast datagrams to hosts that have "joined" that multicast group

Multicast groups

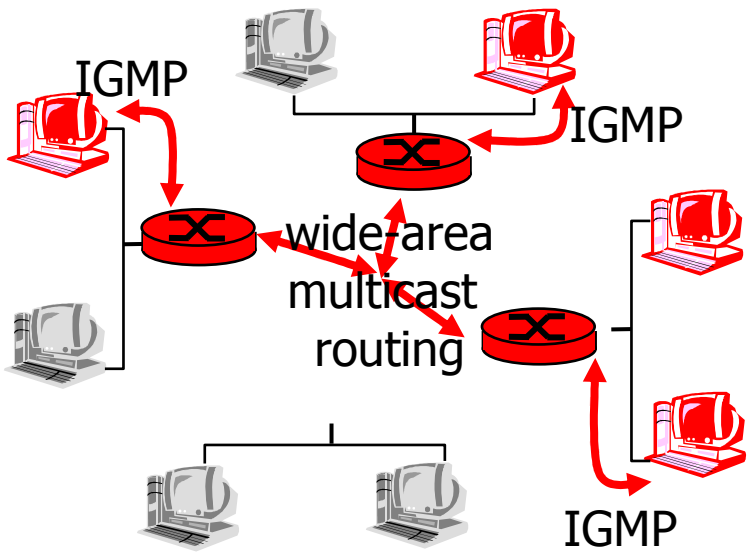
- Class D Internet addresses reserved for multicast:



- Host group semantics:
  - anyone can "join" (receive) multicast group
  - anyone can send to multicast group
  - no network-layer identification to hosts of members
- *Needed:* Infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group

Joining a mcast group: Two-step process

- *Local:* Host informs local mcast router of desire to join group: IGMP (Internet Group Management Protocol)
- *Wide area:* Local router interacts with other routers to receive mcast datagram flow
  - many protocols (e.g., DVMRP, MOSPF, PIM)



Multicast via Indirection: Why?

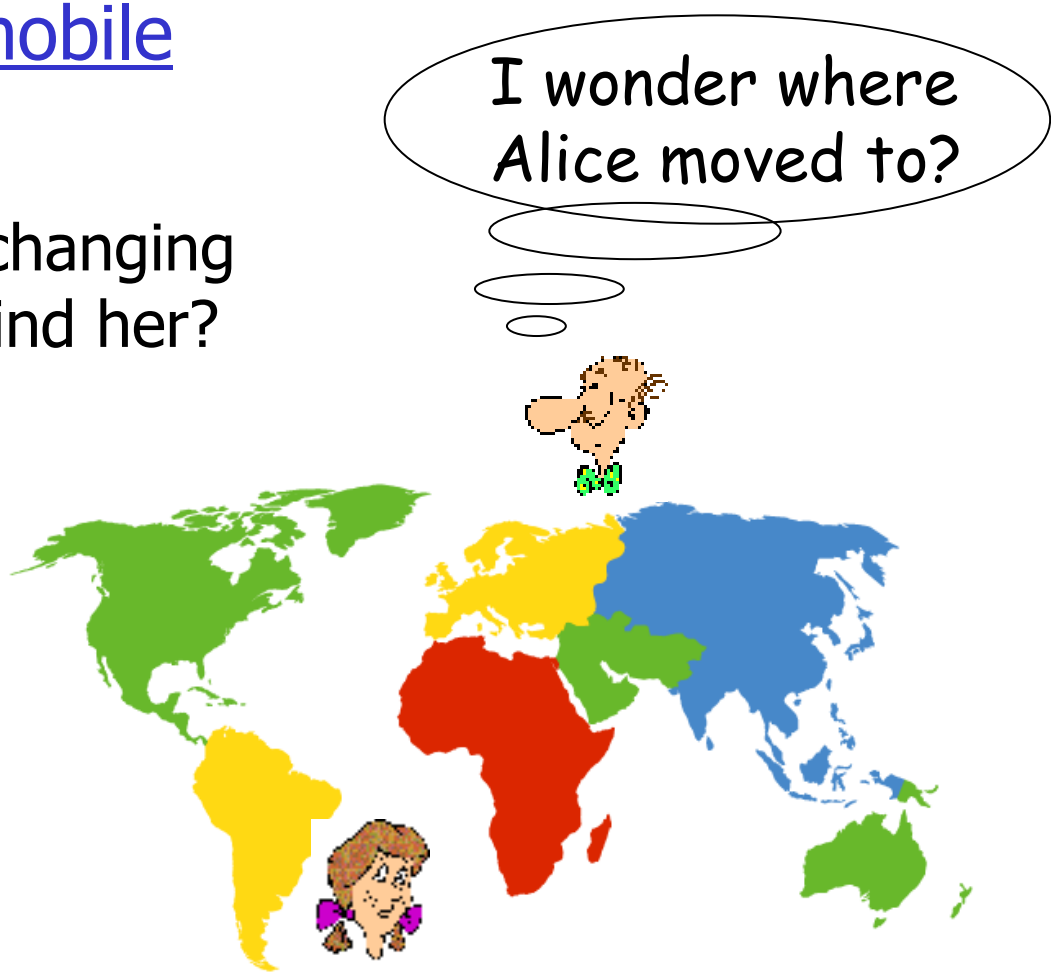
- Don't need to individually address each member in the group: header savings
- Looks like unicast; application interface is simple, single group
- Abstraction, delegating works of implementation to the routers
- More scalable because, sender doesn't manage the group, as receivers are added, new receivers must do the work to add themselves

Mobility and Indirection

How do *you* contact a mobile friend?

Consider friend frequently changing addresses, how do you find her?

- Search all phone books?
- Call her parents?
- Expect her to let you know where he/she is?



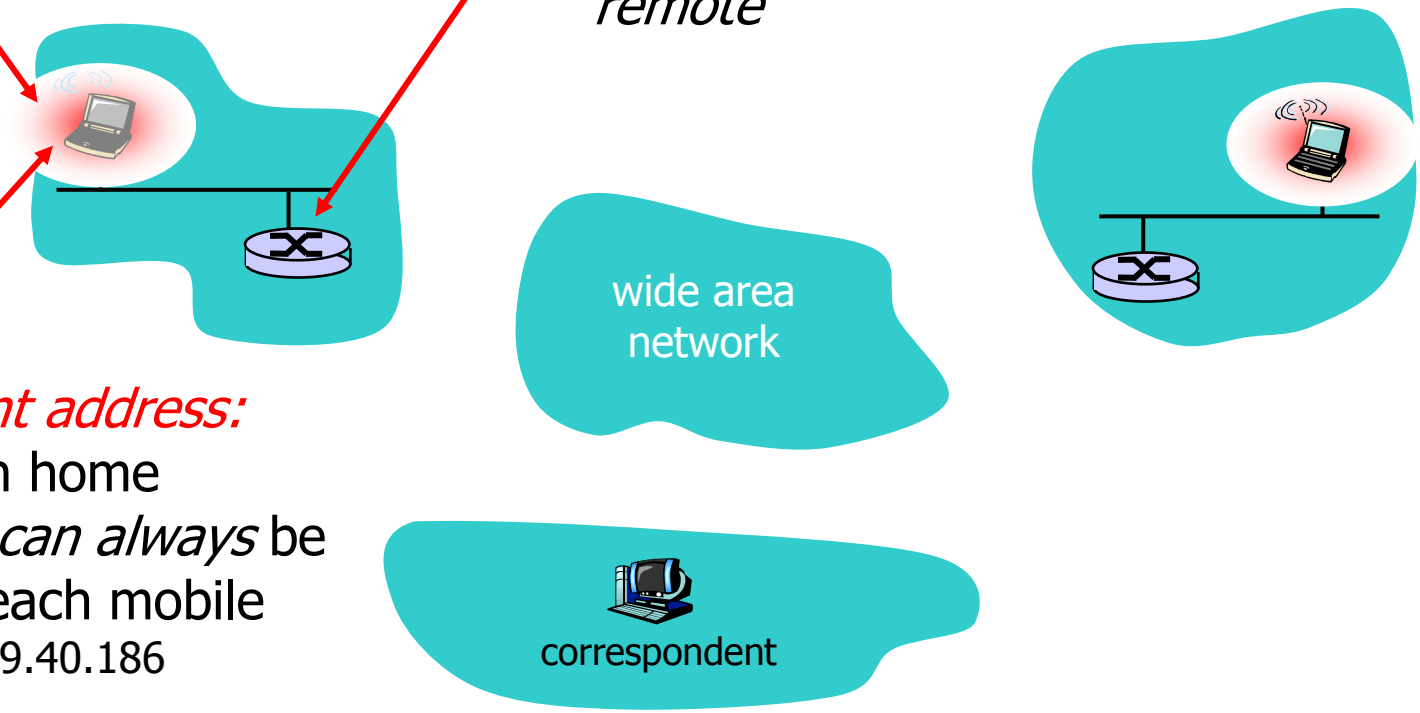
Mobility and indirection:

- Mobile node moves from network to network
- Correspondents want to send packets to mobile node
- Two approaches:
  - *Indirect routing:* Communication from correspondent to mobile goes through home agent, then forwarded to remote
  - *Direct routing:* Correspondent gets foreign address of mobile, sends directly to mobile

Mobility: Vocabulary

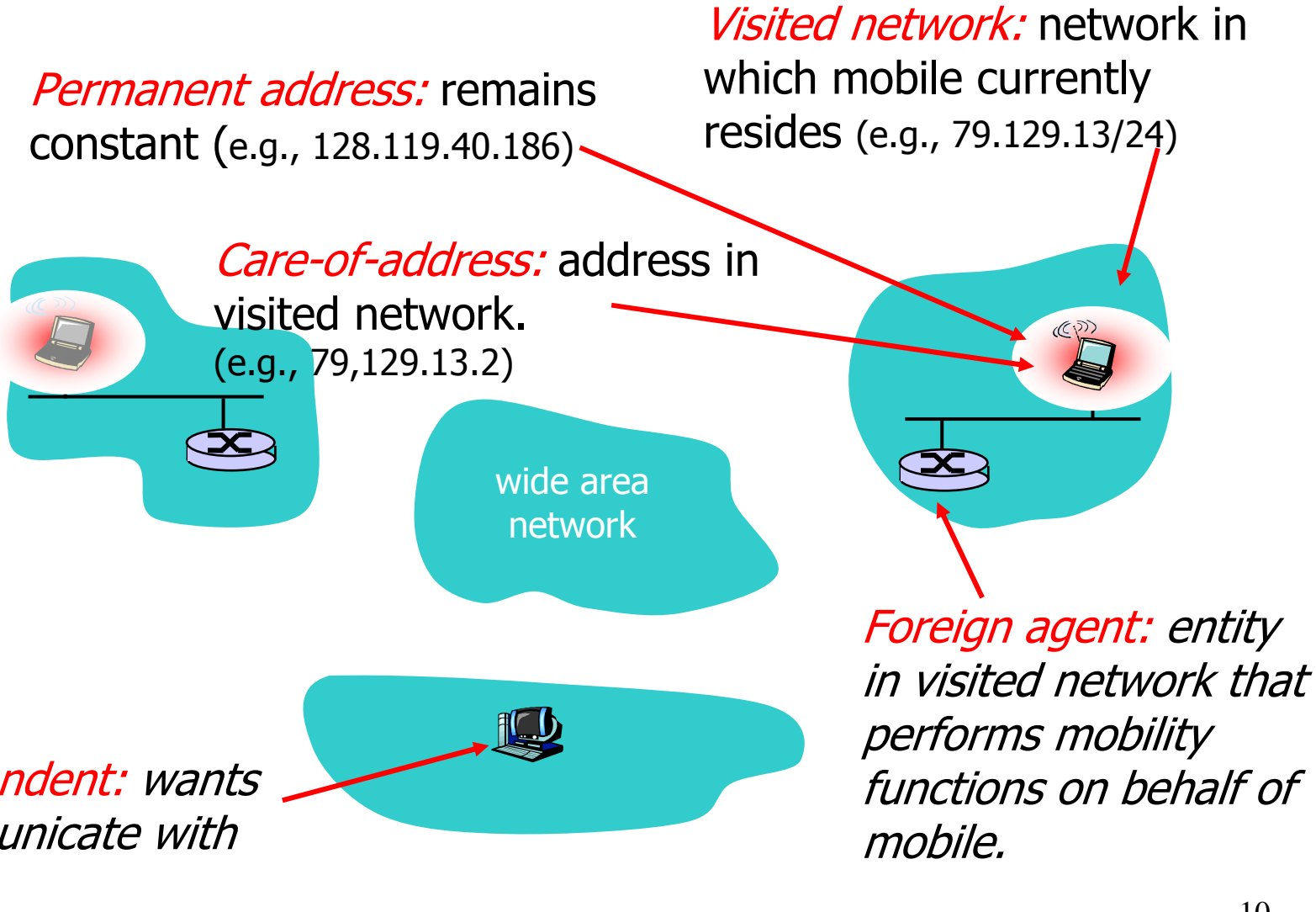
Home network: permanent "home" of mobile (e.g., 128.119.40/24)

Home agent: *entity that will perform mobility functions on behalf of mobile, when mobile is remote*

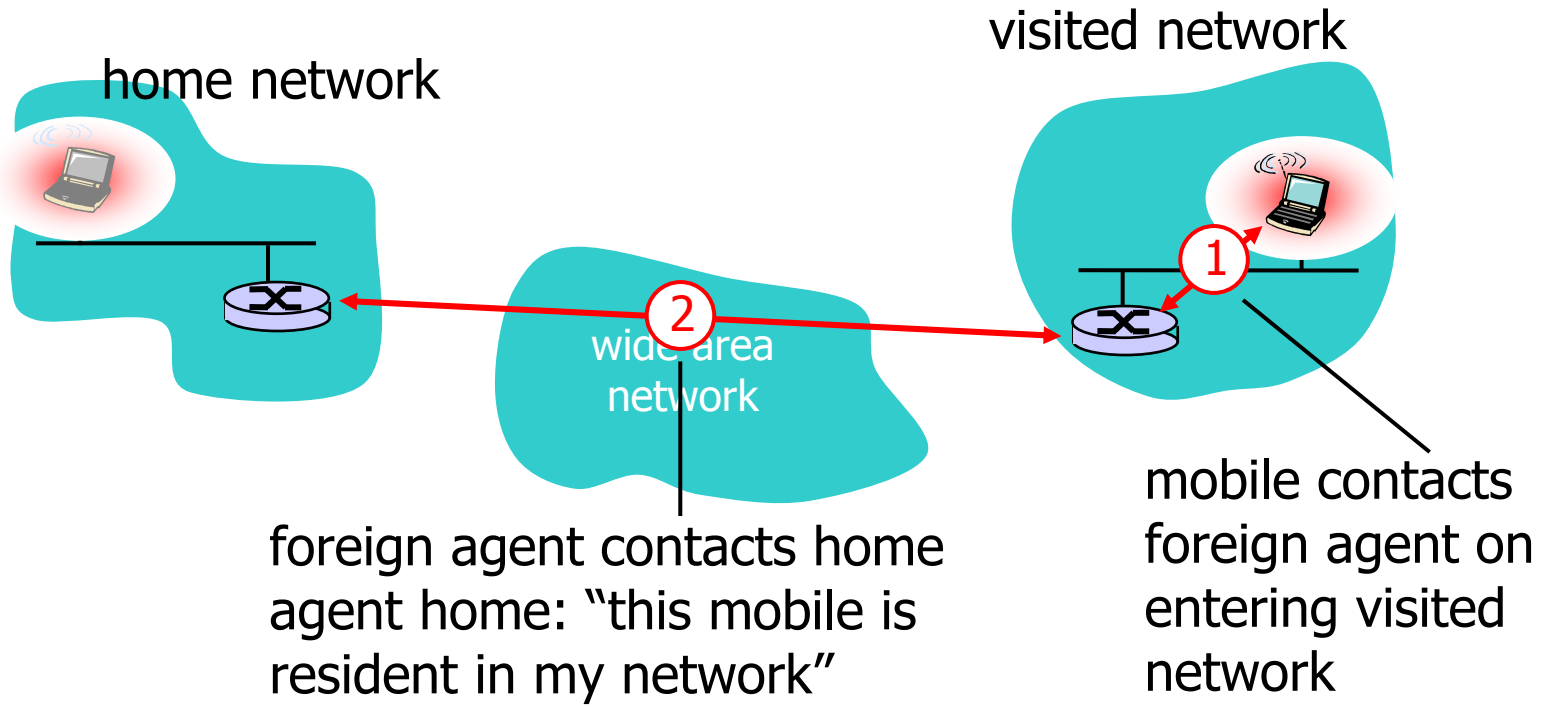


Permanent address: address in home network, *can always* be used to reach mobile e.g., 128.119.40.186

Mobility: more vocabulary



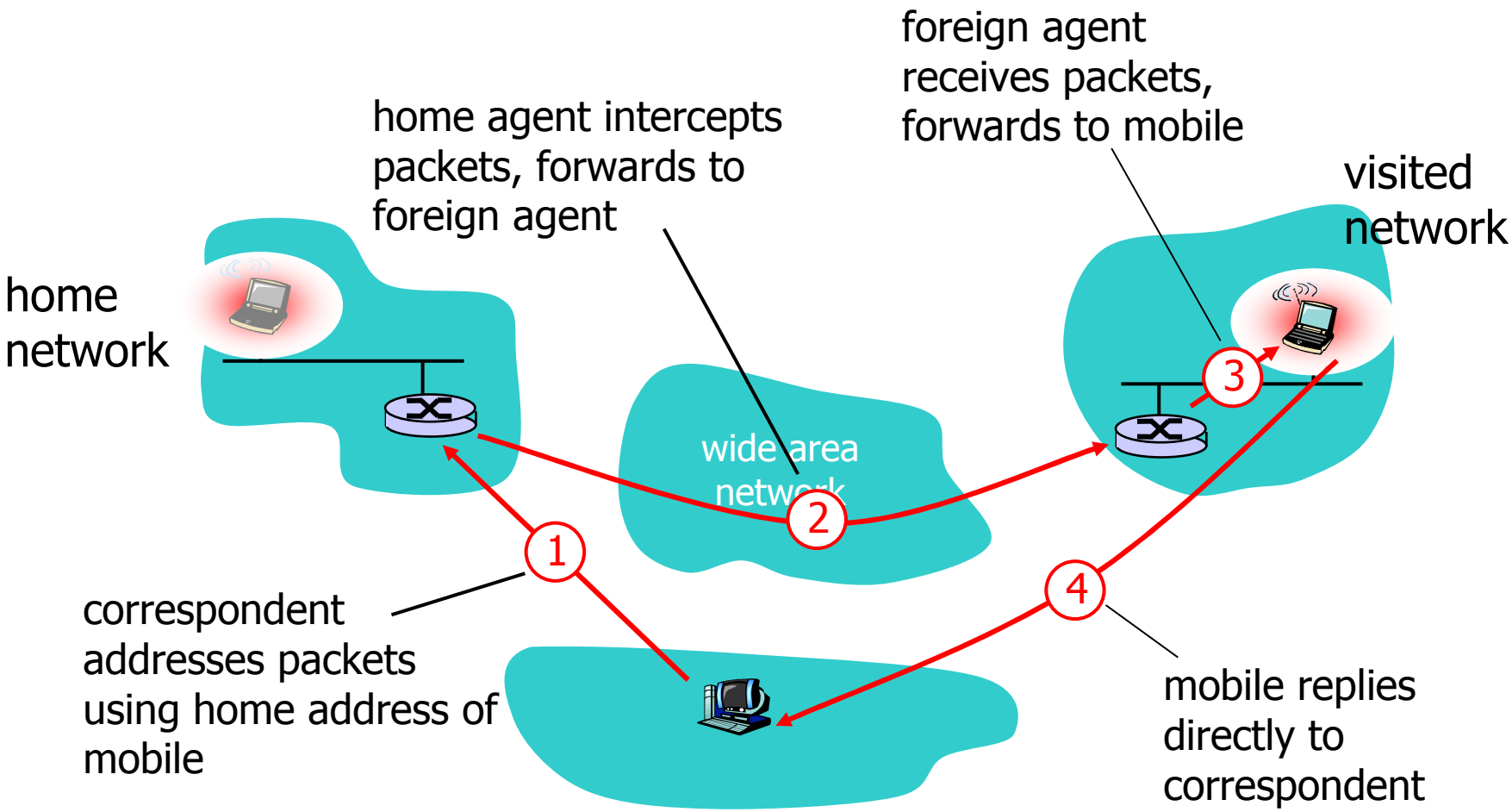
Mobility: registration



End result:

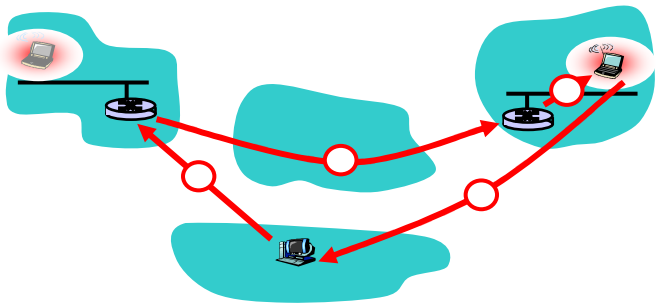
- Foreign agent knows about mobile
- Home agent knows location of mobile

Mobility via Indirect Routing



Indirect Routing: comments

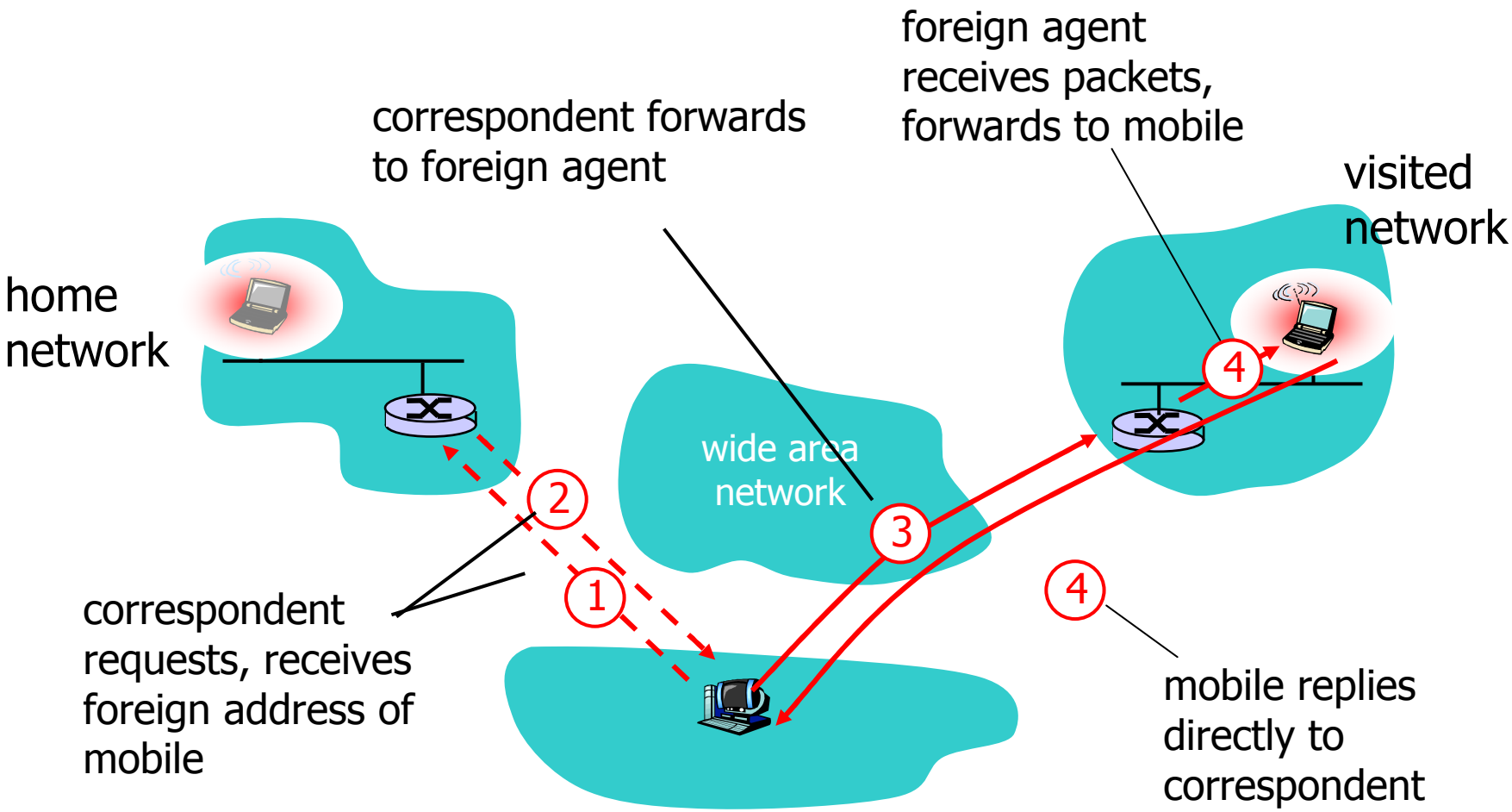
- Mobile uses two addresses:
  - Permanent address: used by correspondent (hence mobile location is *transparent* to correspondent)
  - Care-of-address: used by home agent to forward datagrams to mobile
- Foreign agent functions may be done by mobile itself
- Triangle routing: correspondent-home-network-mobile
  - Inefficient when correspondent, mobile are in same network



Indirect Routing: moving between networks

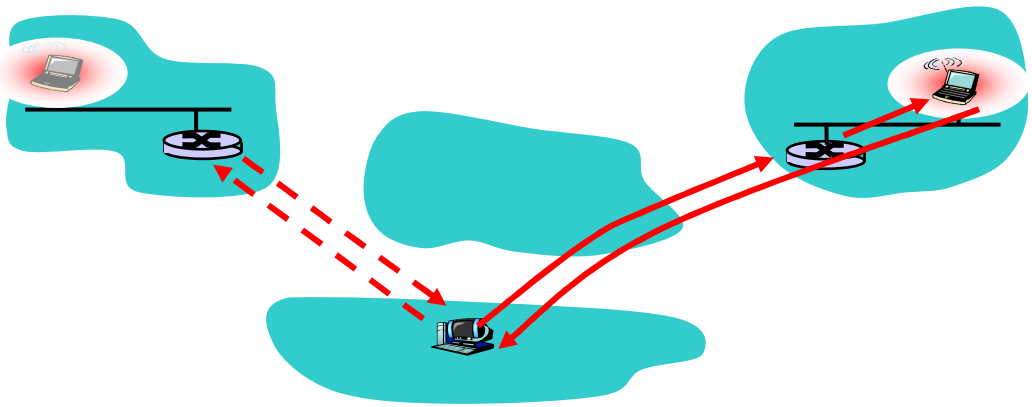
- Suppose mobile user moves to another network
  - Registers with new foreign agent
  - New foreign agent registers with home agent
  - Home agent update care-of-address for mobile
  - Packets continue to be forwarded to mobile (but with new care-of-address)
- Mobility, changing foreign networks transparent: *Ongoing connections can be maintained!*

Mobility via Direct Routing



Mobility via Direct Routing: comments

- Overcome triangle routing problem
- Non-transparent to correspondent: Correspondent must get care-of-address from home agent
  - What happens if mobile changes networks?



Mobile IP

- RFC 3220
- Has many features we've seen:
  - home agents, foreign agents, foreign-agent registration, care-of-addresses, encapsulation (packet-within-a-packet)
- 3 components to standard:
  - agent discovery
  - registration with home agent
  - indirect routing of datagrams

Mobility via indirection: why indirection?

- Transparency to correspondent
- "Mostly" transparent to mobile (except that mobile must register with foreign agent)
  - transparent to routers, rest of infrastructure
  - potential concerns if egress filtering is in place in origin networks (since source IP address of mobile is its home address): spoofing?

Secure Overlay Service

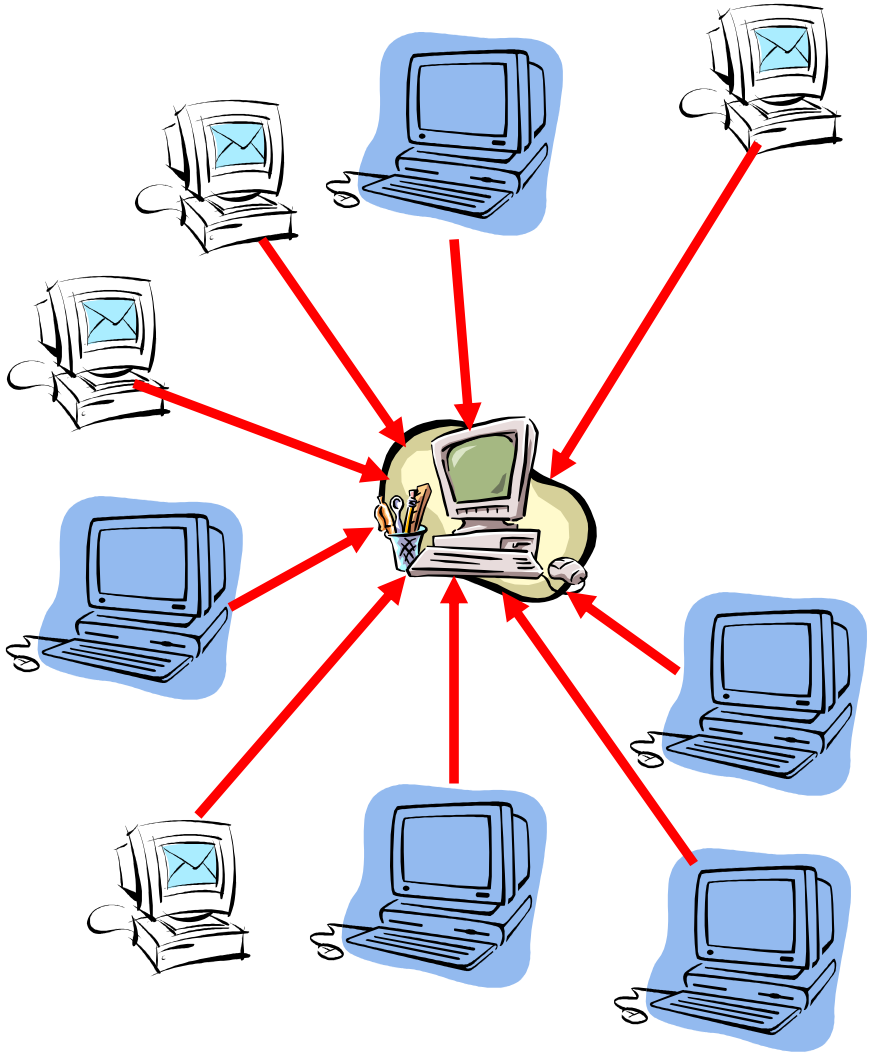
SoS: An overlay network, using indirection and randomization to provide legitimate users (only) with denial-of-service free access to a server.

Overlay network:

- Network or distributed infrastructure with common network services (e.g., routing) built "on top" of other networks
- Example: Distributed application in which application-layer nodes relay messages among themselves, using underlying IP routing to get from one site to another

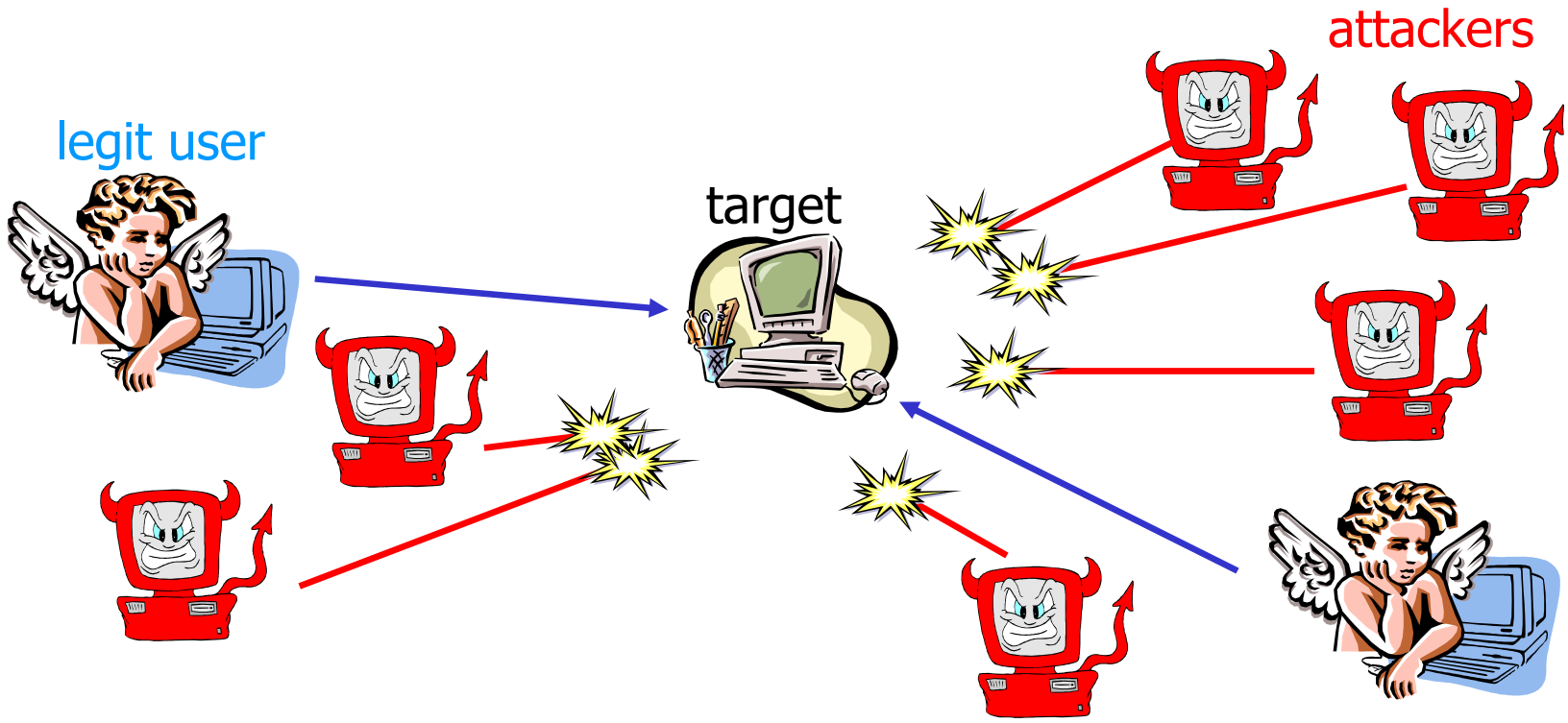
Performing a DoS Attack

1. Select Target to attack
2. Break into accounts around the network
3. Have these accounts send packets toward the target



Goal of Secure Overlay Service (SoS)

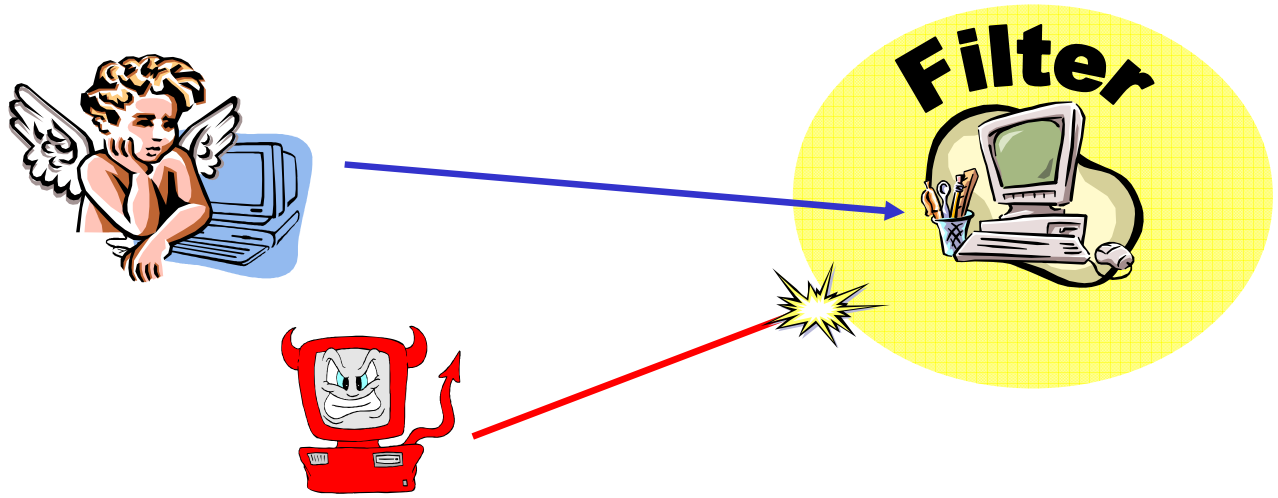
- Pre-approved legitimate users communicate with target
  - legit users may be mobile (IP addresses change)
- Un-approved (attackers') packets don't reach target



Step 1 – Filtering

Routers "near" target filter packets based on IP addr

- IP addresses from legitimate user allowed through
- IP addresses from illegitimate users are not



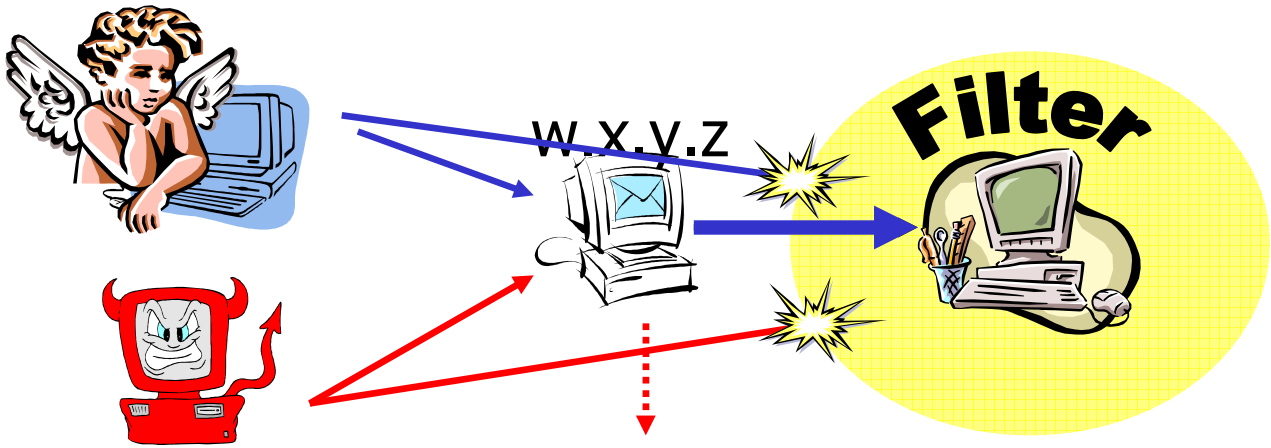
Concerns:

- Bad users have same IP address as good user?
- Bad users know good user's IP address: spoofing?
- Good IP address changes frequently (mobility)?

Step 2 – indirection via a proxy

Use proxy, outside filtered region

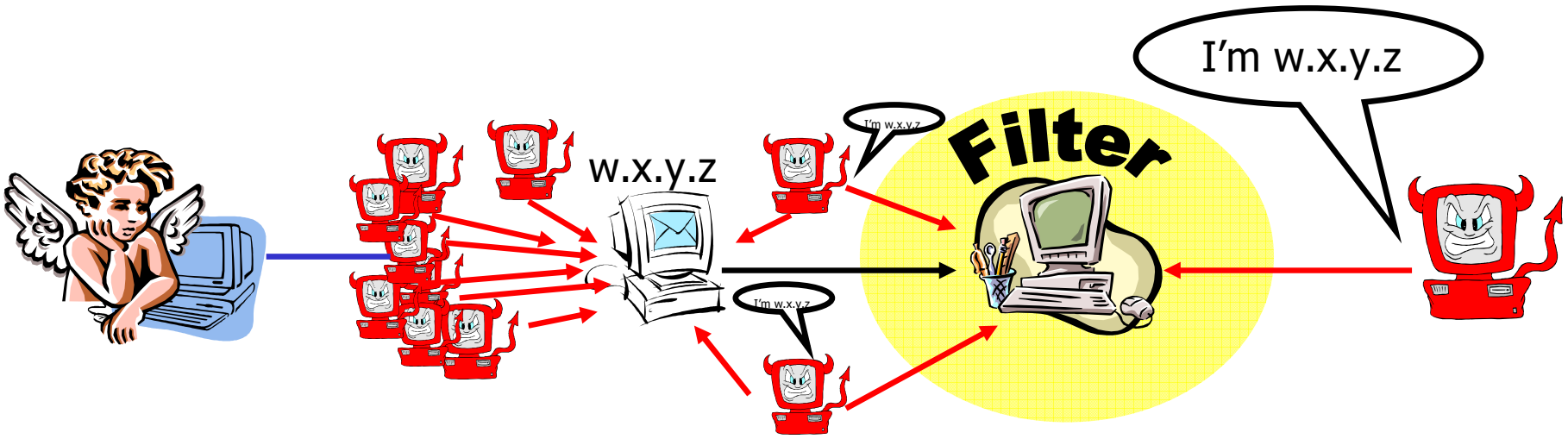
- Proxy, being a computer (rather than router) can perfom heavy-weight authentication, access control
- Only packets from proxy permitted through filter
- Proxy only forwards verified packets from legitimate sources through filter



Problems with a known Proxy

Proxies introduce other problems

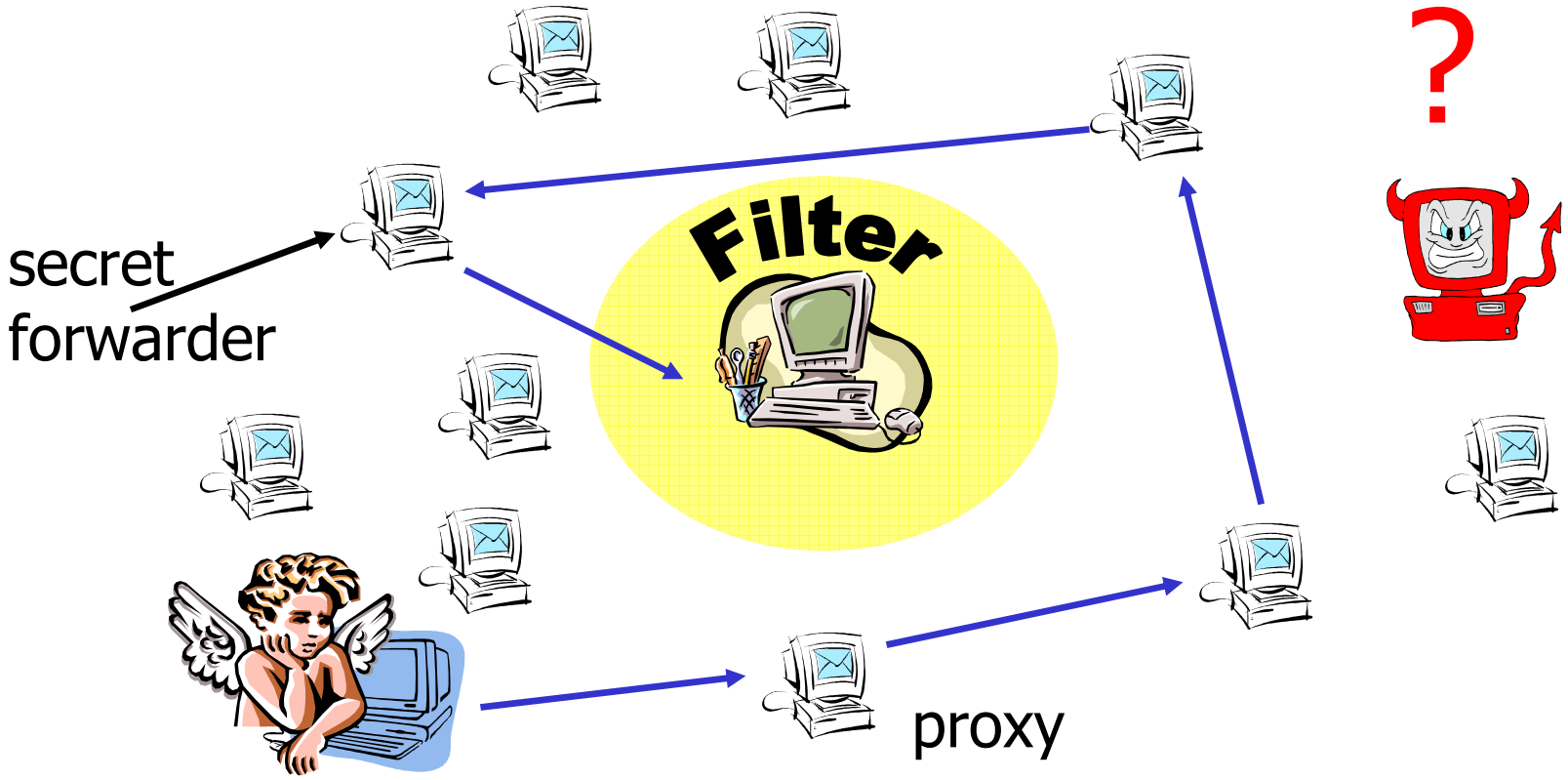
- Attacker can breach filter by attacking with spoofed proxy address
- Attacker can DoS attack proxy, again preventing legitimate user communication



Step 3 – Multiple proxies with secret forwarding

- Create many proxies (too many to attack)
- Target specifies small set of proxies as secret forwarders
  - Only secret-forwarder packets pass through filter
  - Only secret forwarders know they are secret forwarders (other proxies unaware)
- To get host packet to target
  - Host contacts any proxy (which checks legitimacy)
  - Proxy randomly routes packet to another proxy
  - If destination proxy is secret forwarder, packet forwarded to target, otherwise packet randomly routed to another proxy

SOS with "Random" routing



With filters, multiple proxies, and secret forwarder(s), attacker cannot "focus" attack

SoS

Why indirection?

- Ultimate destination address is unknown (hackers can not attack target, only attack proxies (?))
- Address of target only known to small number of secret forwarders, which rotate and can change

Issues:

- Why can't hacker just try all addresses of all proxies to get through?

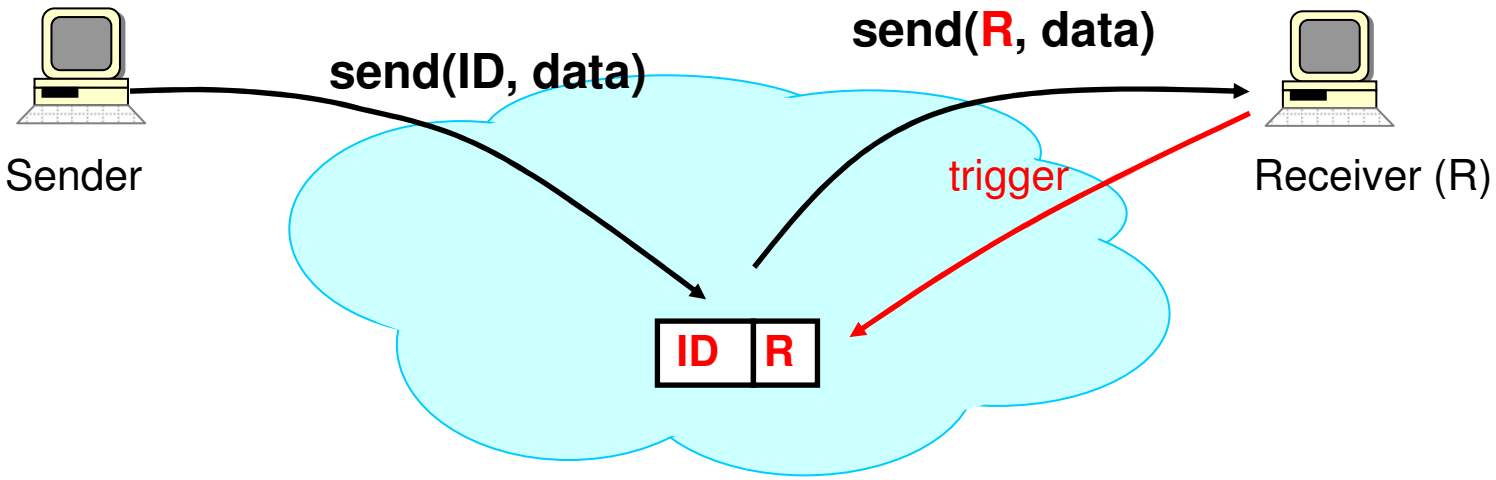
An Internet Indirection Infrastructure

Motivation:

- Today's Internet is built around point-to-point communication abstraction:
  - Send packet "p" from host "A" to host "B"
  - One sender, one receiver, at fixed and well-known locations
- … not appropriate for applications that require other communications primitives:
  - multicast (one to many)
  - mobility (one to anywhere)
  - anycast (one to any)
- We've seen indirection used to provide these services
  - Idea: Make indirection a "first-class object"

Internet Indirection Infrastructure (I3)

- Change communication abstraction: Instead of point-to-point, exchange packets by name
  - each packet has an identifier ID
  - to receive packet with identifier ID, receiver R stores trigger (ID, R) into network
  - triggers stored in network overlay nodes



Service Model

- API
  - sendPacket(p);
  - insertTrigger(t);
  - removeTrigger(t); // optional

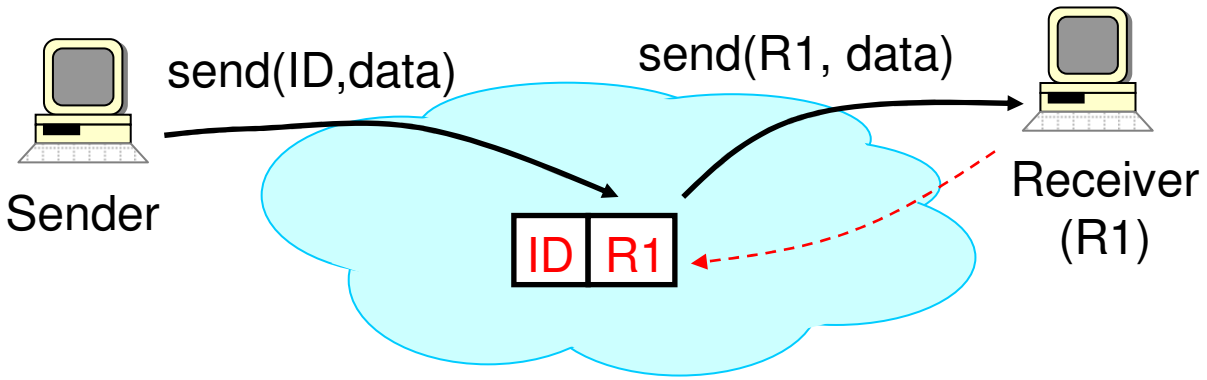
- best-effort service model (like IP)
- triggers periodically refreshed by end-hosts
- reliability, congestion control, flow-control implemented at end hosts, and trigger-storing overlay nodes

Discussion

- Trigger is similar to routing table entry
- Essentially: Application layer publish-subscribe infrastructure
- Application-level overlay infrastructure
- Unlike IP, end hosts control triggers, i.e., end hosts responsible for setting and maintaining "routing tables"
- Provide support for
  - mobility
  - multicast
  - anycast
  - composable services

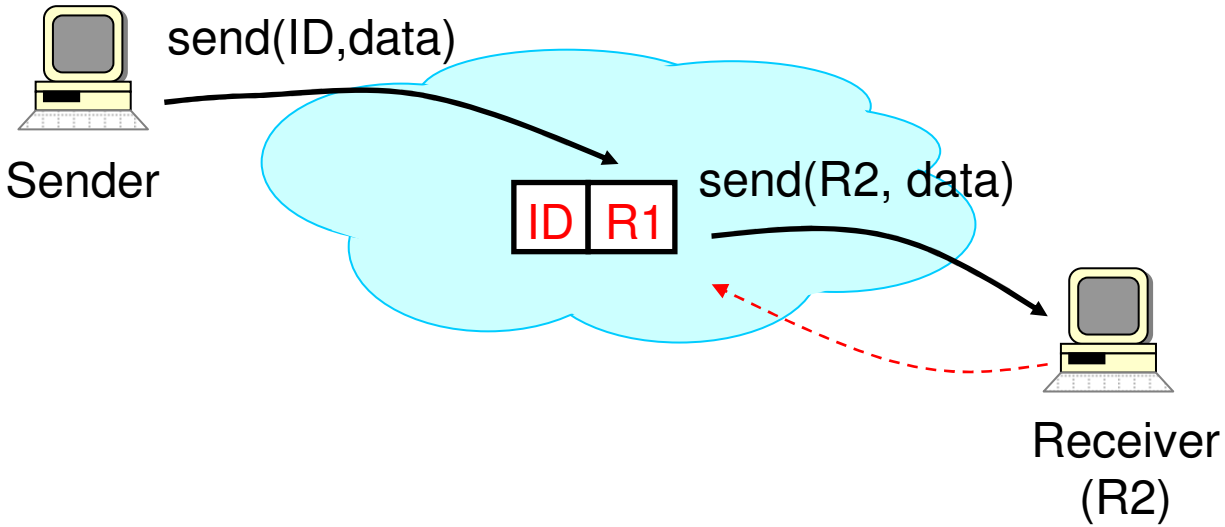
Mobility

- Receiver updates its trigger as it moves from one subnet to another
  - mobility transparent to sender
  - location privacy



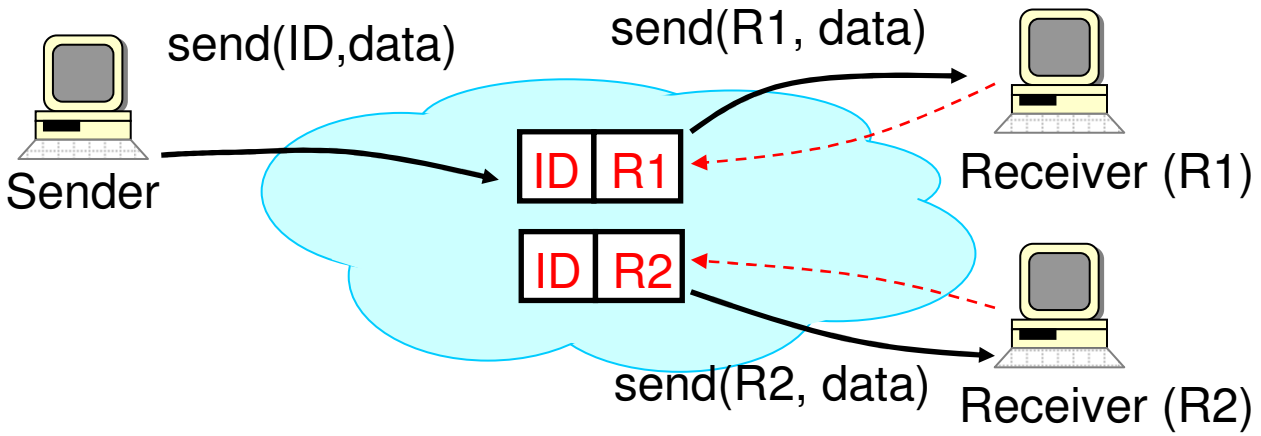
Mobility

- Receiver updates its trigger as it moves from one subnet to another
  - mobility transparent to sender
  - location privacy



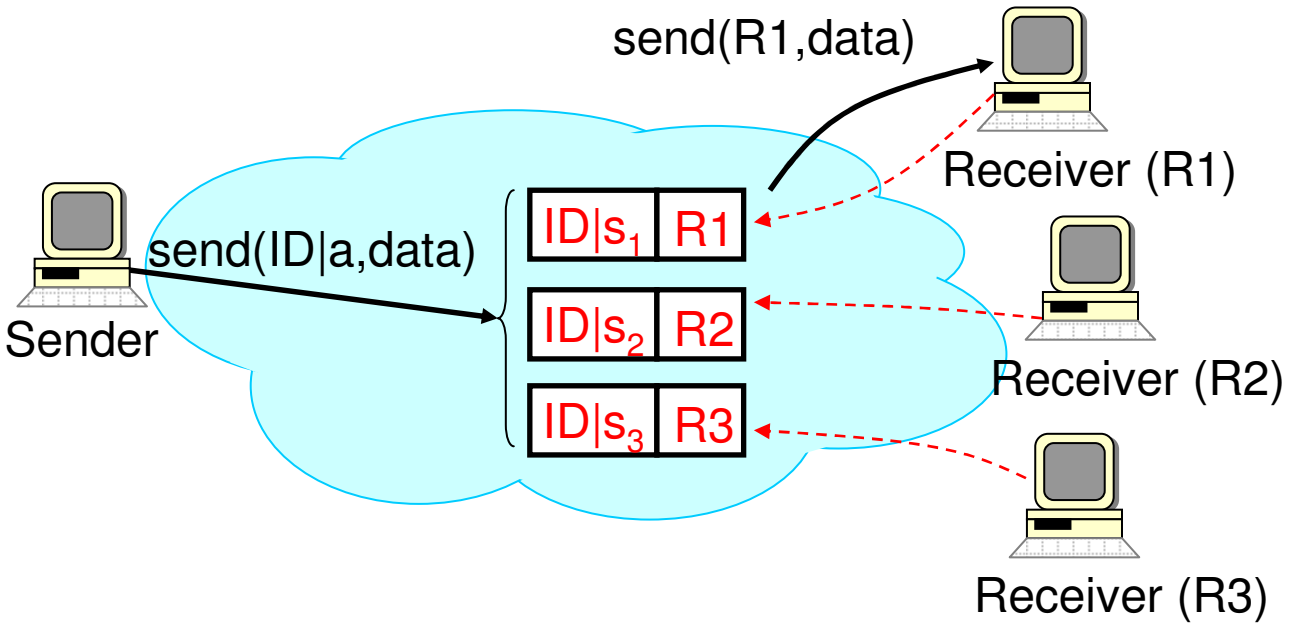
Multicast

- Unifies multicast and unicast abstractions
  - multicast: receivers insert triggers with same ID
- Application naturally moves between multicast and unicast, as needed
  - "impossible" in current IP model



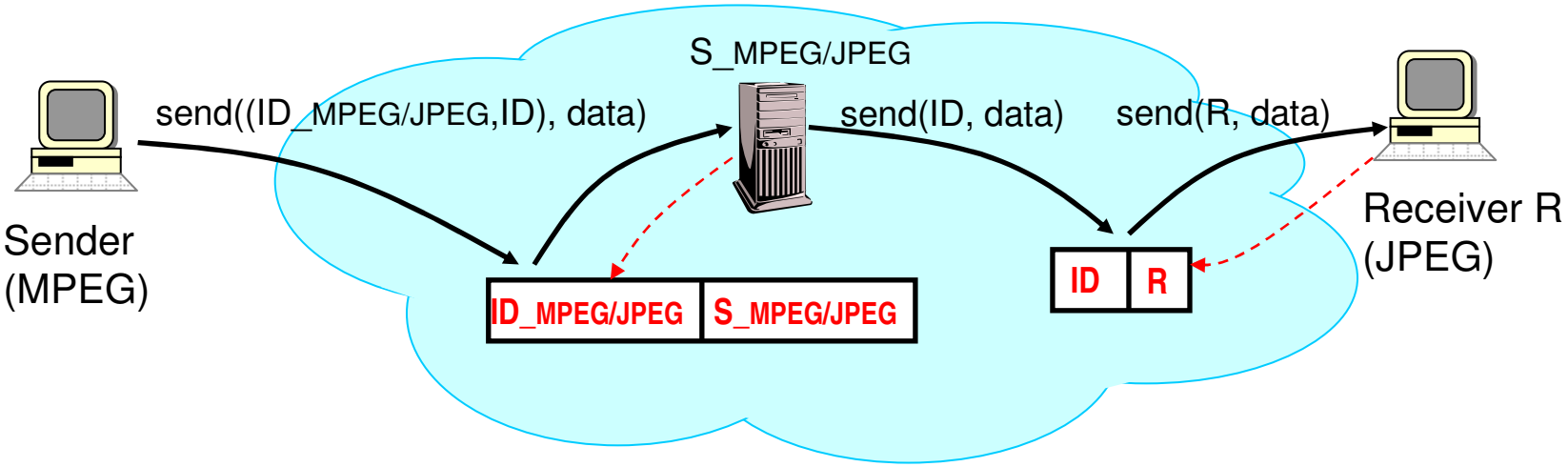
Anycast (cont'd)

- Route to any one in set of receivers
- Receivers i in anycast group inserts same ID, with anycast qualifications



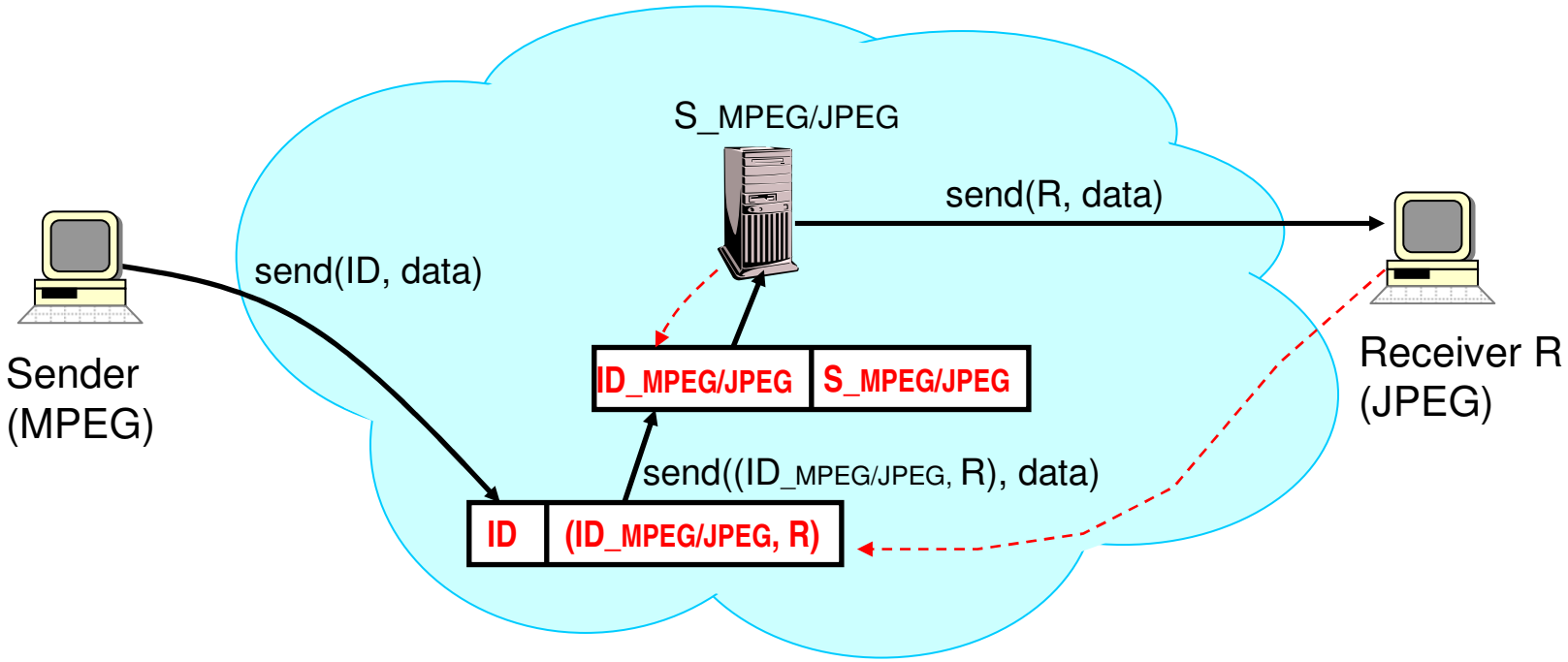
Composable Services

- Use stack of IDs to encode successive operations to be performed on data (e.g., transcoding)
- Don't need to configure path between services



Composable Services (cont'd)

- Both receivers *and* senders can specify operations to be performed on data



Discussion of I3

- How would receiver signal ACK to sender? What is needed?
- Does many-to-one fit well in this paradigm?
- security, snooping, information gathering: what are the issues?

Indirection: Summary

We've seen indirection used in many ways:

- multicast
- mobility
- SoS
- Internet indirection

The uses of indirection:

- Sender does not need to know receiver id – do not *want* sender to know intermediary identities
- Beauty, grace, elegance
- Transparency of indirection is important
- Performance: is it more efficient?
- Security: Important issue for I3