

## **Indirection**

*Jasleen Kaur*

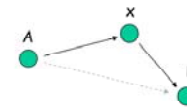
October 28, 2009

COMP 790-088

## **Indirection**

**Indirection:**

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



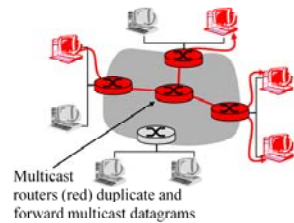
"Every problem in computer science can be solved by adding another level of indirection"  
-- Butler Lampson

3-2

## Multicast: one sender to many receivers

◆ **Multicast:** act of sending datagram to multiple receivers with single “transmit” operation

◆ **Question:** how to achieve multicast?

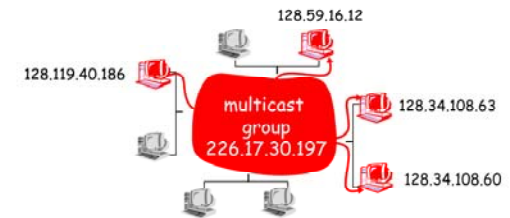


### Network multicast

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

3-3

## Internet Multicast Service Model



multicast group concept: use of *indirection*

- » host addresses IP datagram to multicast group
- » routers forward multicast datagrams to hosts that have “joined” that multicast group

3-4

## Multicast via Indirection: why?

- ◆ Naming and forwarding in IP tailored for point-to-point communication
- ◆ Indirection
  - » Provides flexible naming
  - » Decouples sender from receivers (and their joins and leaves)

3-5

## 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?



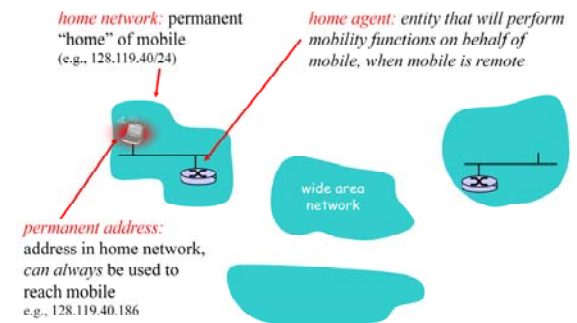
3-6

## Mobility and indirection:

- ◆ Situation:
  - » 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

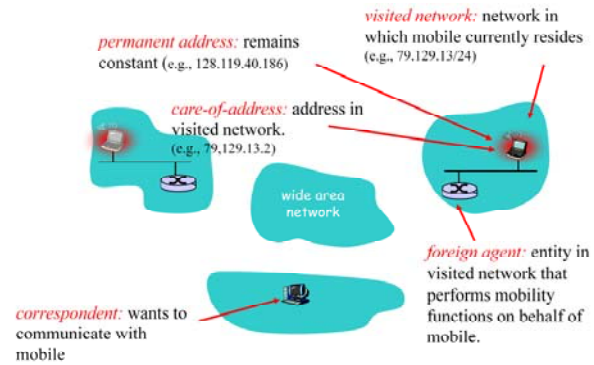
3-7

## Mobility: Vocabulary

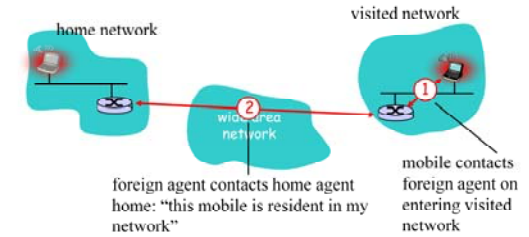


3-8

## Mobility: more vocabulary



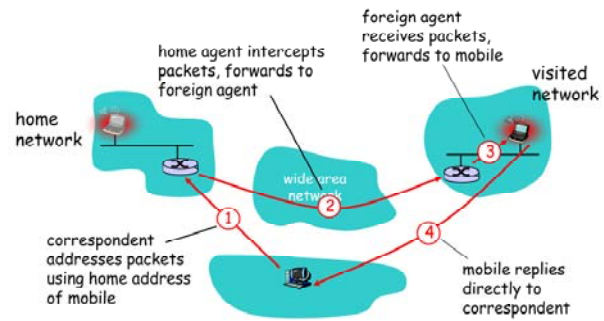
## Mobility: registration



End result:

- ◆ foreign agent knows about mobile
- ◆ home agent knows location of mobile

## Mobility via Indirect Routing



3-11

## 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



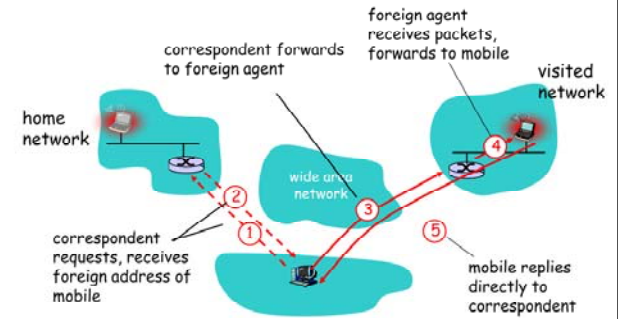
3-12

### 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 updates 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!*

3-13

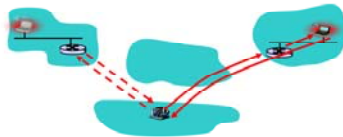
### Mobility via Direct Routing



3-14

## Mobility via Direct Routing: comments

- ◆ overcomes triangle routing problem
- ◆ **non-transparent to correspondent:** correspondent must get care-of-address from home agent
  - » what happens if mobile changes networks?



3-15

## 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)
- ◆ Three components to standard:
  - » agent discovery
  - » registration with home agent
  - » indirect routing of datagrams

3-16



## Mobility via indirection: why indirection?

- ◆ Transparency to correspondent
- ◆ “Mostly” transparent to mobile (except mobile must register with foreign agent)
- ◆ Transparent to routers, rest of infrastructure
  - » practical concern: if egress filtering is in place in foreign networks (since source IP address of mobile is its home address): spoofing?

3-17

## An Internet Indirection Infrastructure

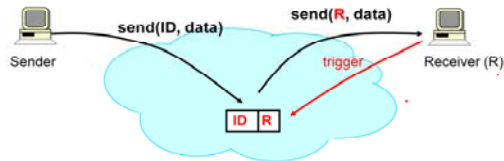
### Motivation:

- ◆ 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”

3-18

## 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) in network
  - » triggers stored in network overlay nodes



3-19

## Service Model

- ◆ API
  - » `sendPacket(p);`
  - » `insertTrigger(t);`
  - » `removeTrigger(t); // optional`
- ◆ Best-effort service model (like IP)
- ◆ Triggers periodically refreshed by end-hosts
  - » Q: what is this approach called?
- ◆ Reliability, congestion control, flow-control implemented at end hosts, and trigger-storing overlay nodes

packet p: (ID,data)  
trigger t: (ID, addr)

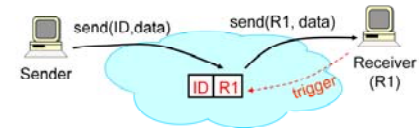
3-20

## Discussion

- ◆ Trigger is similar to routing table entry
- ◆ Application-level overlay infrastructure
  - » Essentially: application layer *publish-subscribe 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

3-21

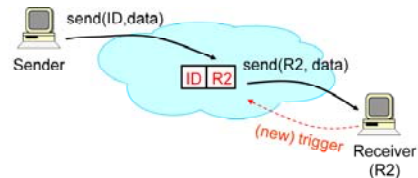
## Mobility



3-22

## Mobility

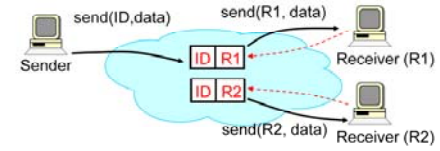
- ◆ Receiver updates its trigger as it moves from one subnet to another
  - » mobility transparent to sender
  - » location privacy



3-23

## Multicast

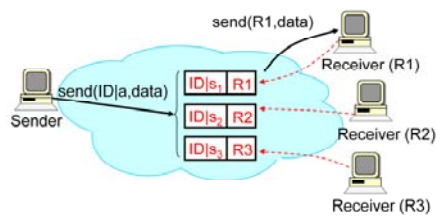
- ◆ Unifies multicast and unicast abstractions
  - » multicast: receivers insert triggers with same ID
- ◆ Application naturally moves between multicast and unicast, as needed
  - » "impossible" in current IP model !



3-24

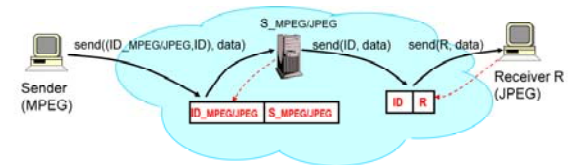
## Anycast

- ◆ Route to any one in set of receivers
- ◆ Receiver  $i$  in anycast group inserts same ID, with anycast qualifications
- ◆ Route to receiver with best match between  $a$  and  $s_i$



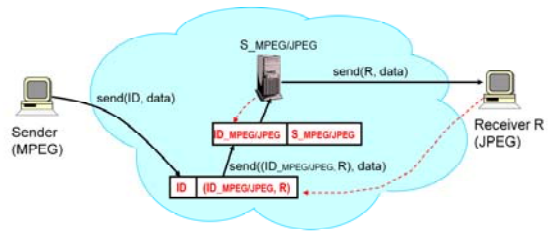
## Composable Services

- ◆ 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)

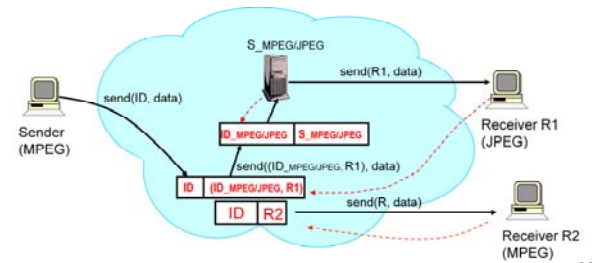
- Both receivers *and* senders can specify operations to be performed on data



3-27

## Heterogeneous Multicast

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



3-28

## 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?
- ◆ In-network storage to handle disconnection?

3-29

## Indirection: Summary

---

- ◆ We've seen indirection used in many ways:
  - » multicast
  - » mobility
  - » Internet indirection
- ◆ Uses of indirection:
  - » sender does not need to know receiver id
    - ◊ do not *want* sender to know intermediary identities
  - » elegant
  - » transparency of indirection is important
  - » performance: is it more efficient?
  - » security: important issue for I3

3-30