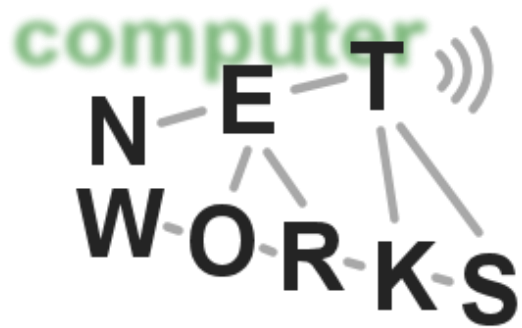


Network Layer – Part III

Multicast and Mobility

Computer Networks, Winter 2010/2011

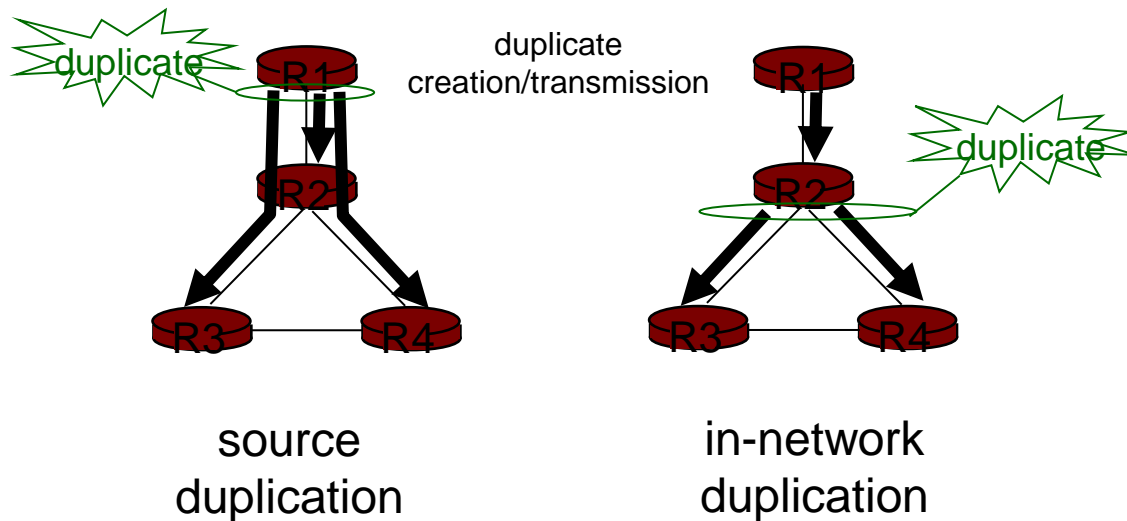


Network Layer III

- 4.6 Multicast
 - Broadcast routing
 - Multicast routing
 - Multicast routing protocols
- 4.7 Mobility
 - What is Mobility?
 - Network layer mobility concepts and principles
 - Mobile IP

Broadcast Routing

- Deliver packets from source to all other nodes
- Source duplication is inefficient:



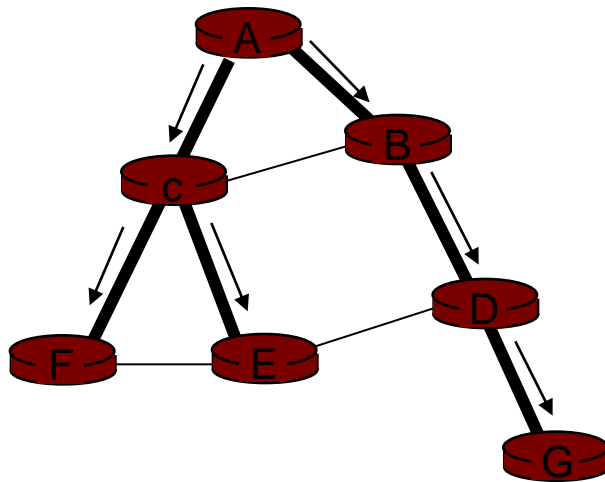
- Source duplication: how does source determine recipient addresses?

In-network duplication

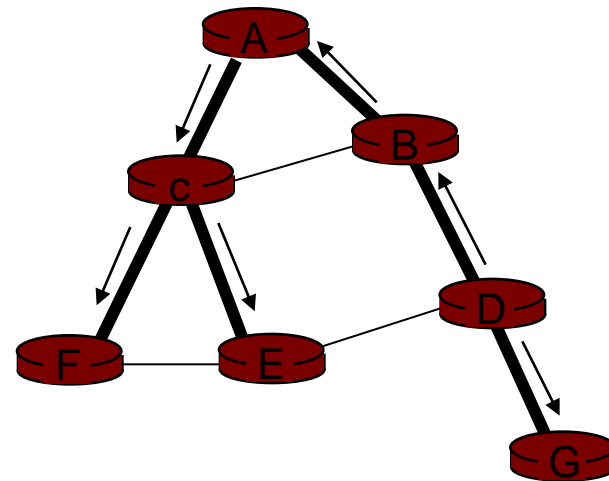
- **Flooding:** when node receives broadcast packets, sends copy to all neighbors
 - Problems: cycles & broadcast storm
- **Controlled flooding:** node only broadcast pkt if it hasn't broadcasted same pkt before
 - Node keeps track of pkt ids already broadcasted
 - Reverse path forwarding (RPF): only forward pkt if it arrived on shortest path between node and source
- **Spanning tree**
 - No redundant packets received by any node

Spanning Tree

- First construct a spanning tree
- Nodes forward copies only along spanning tree



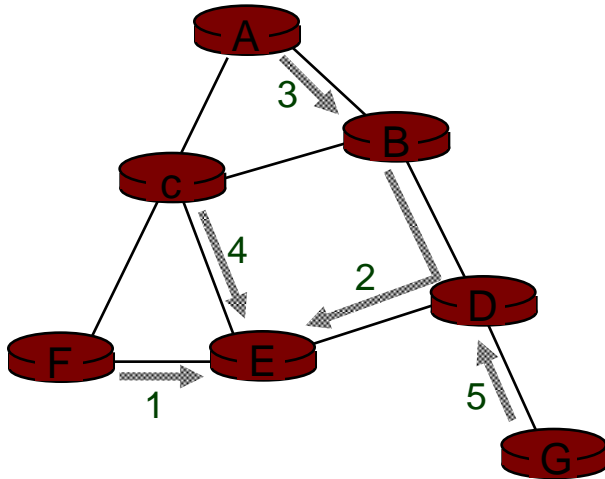
(a) Broadcast initiated at A



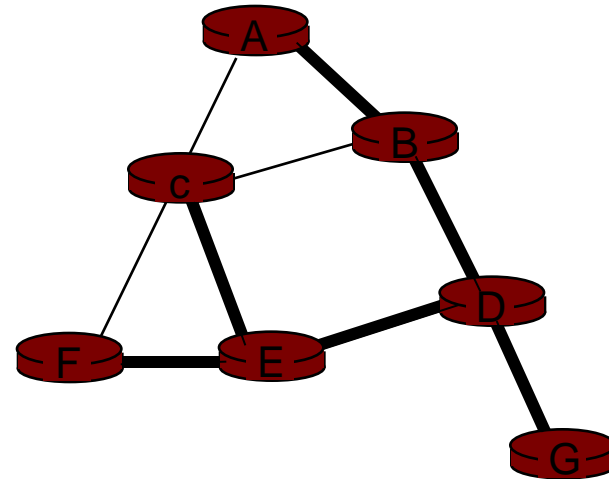
(b) Broadcast initiated at D

Spanning Tree: Creation

- Center node
- Each node sends unicast join message to center node 'E'
 - Message forwarded until it arrives at a node already belonging to spanning tree



(a) Stepwise construction of spanning tree



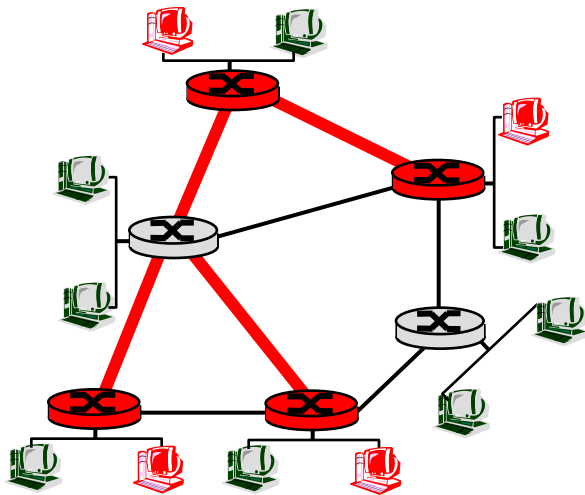
(b) Constructed spanning tree

Network Layer II

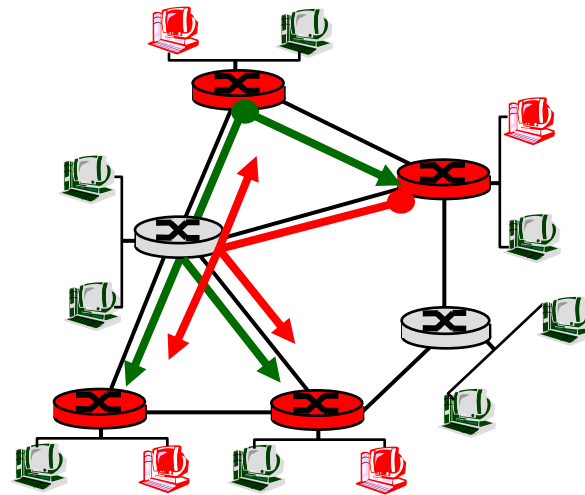
- 4.6 Multicast
 - Broadcast routing
 - Multicast routing
 - Multicast routing protocols
- 4.7 Mobility
 - What is Mobility?
 - Network layer mobility concepts and principles
 - Mobile IP

Multicast Routing: Problem Statement

- **Goal:** find a tree (or trees) connecting routers that have local multicast group members
 - Tree: not all paths between routers used
 - Source-based: different tree from each sender to receiver
 - Shared-tree: same tree used by all group members



Shared tree



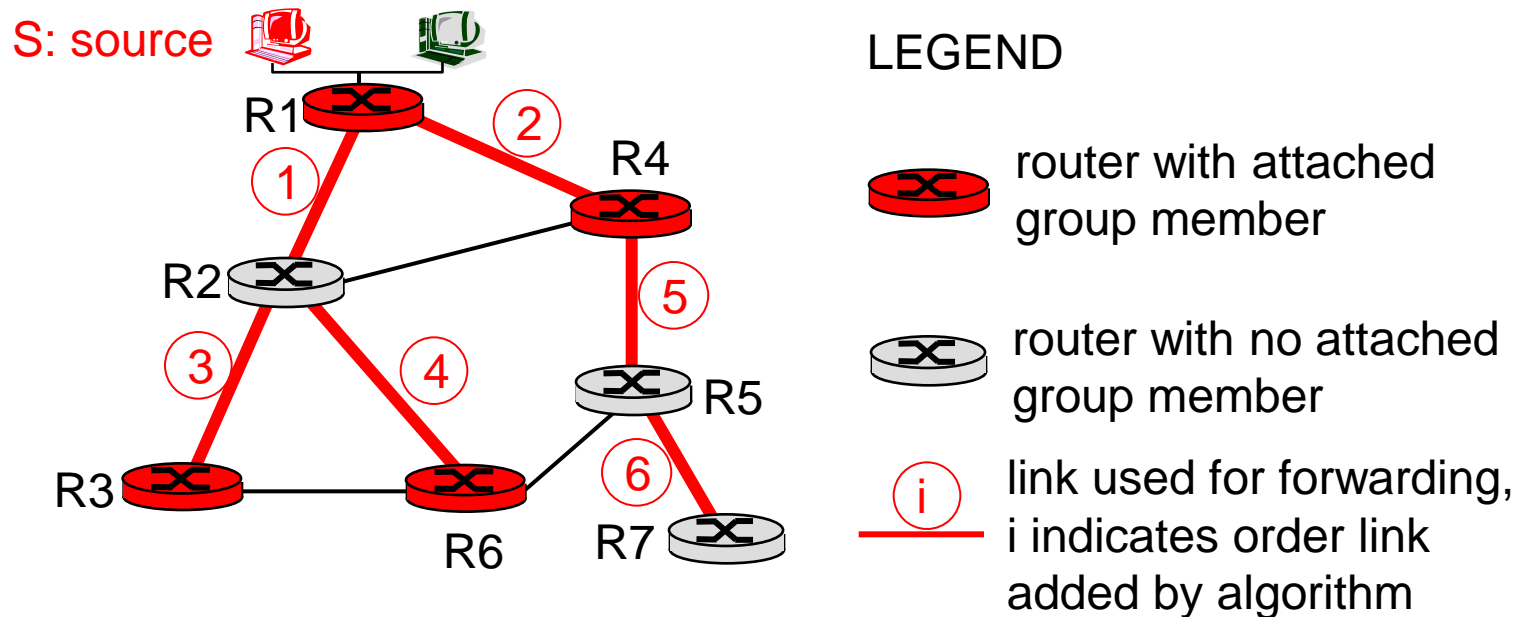
Source-based trees

Approaches for building mcast trees

- Source-based tree: one tree per source
 - shortest path trees
 - reverse path forwarding
- Group-shared tree: group uses one tree
 - minimal spanning (Steiner)
 - center-based trees

Shortest Path Tree

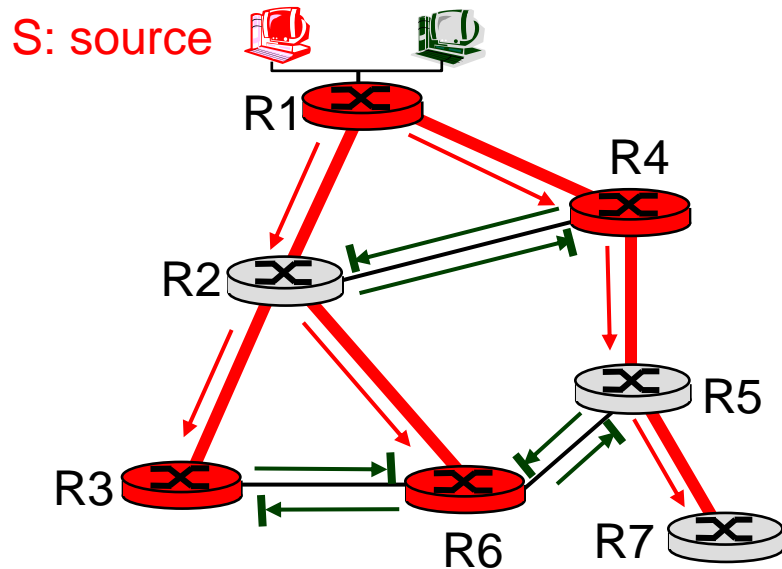
- Multicast forwarding tree: tree of shortest path routes from source to all receivers
 - Dijkstra's algorithm



Reverse Path Forwarding

- Relies on router's knowledge of unicast shortest path from it to sender
- Each router has simple forwarding behavior:
 - if (mcast datagram received on incoming link on shortest path back to center)
 - then flood datagram onto all outgoing links
 - else ignore datagram

Reverse Path Forwarding: example



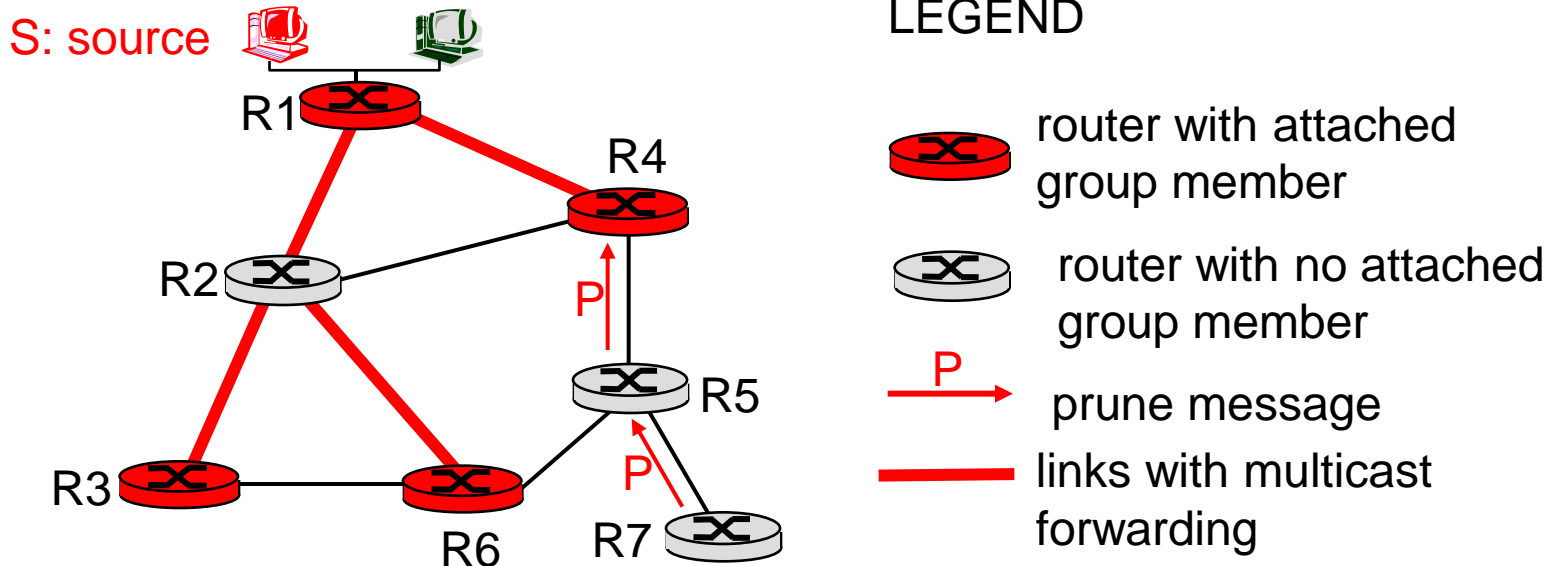
LEGEND

- router with attached group member
- router with no attached group member
- datagram will be forwarded
- datagram will not be forwarded

- result is a source-specific *reverse* SPT
 - may be a bad choice with asymmetric links

Reverse Path Forwarding: pruning

- forwarding tree contains subtrees with no multicast group members
 - no need to forward datagrams down subtree
 - “prune” msgs sent upstream by router with no downstream group members

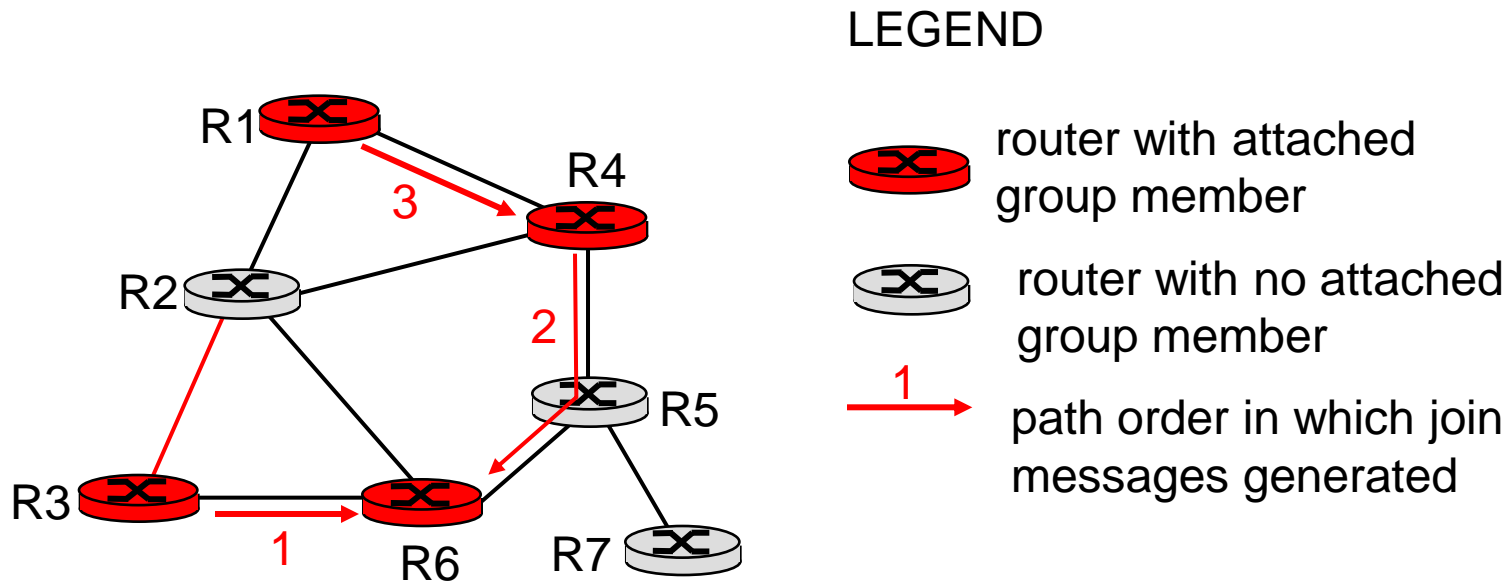


Center-based trees

- Single delivery tree shared by all
- One router identified as “center” of tree
- To join:
 - edge router sends unicast join-msg addressed to center router
 - join-msg “processed” by intermediate routers and forwarded towards center
 - join-msg either hits existing tree branch for this center, or arrives at center
 - path taken by join-msg becomes new branch of tree for this router

Center-based trees: an example

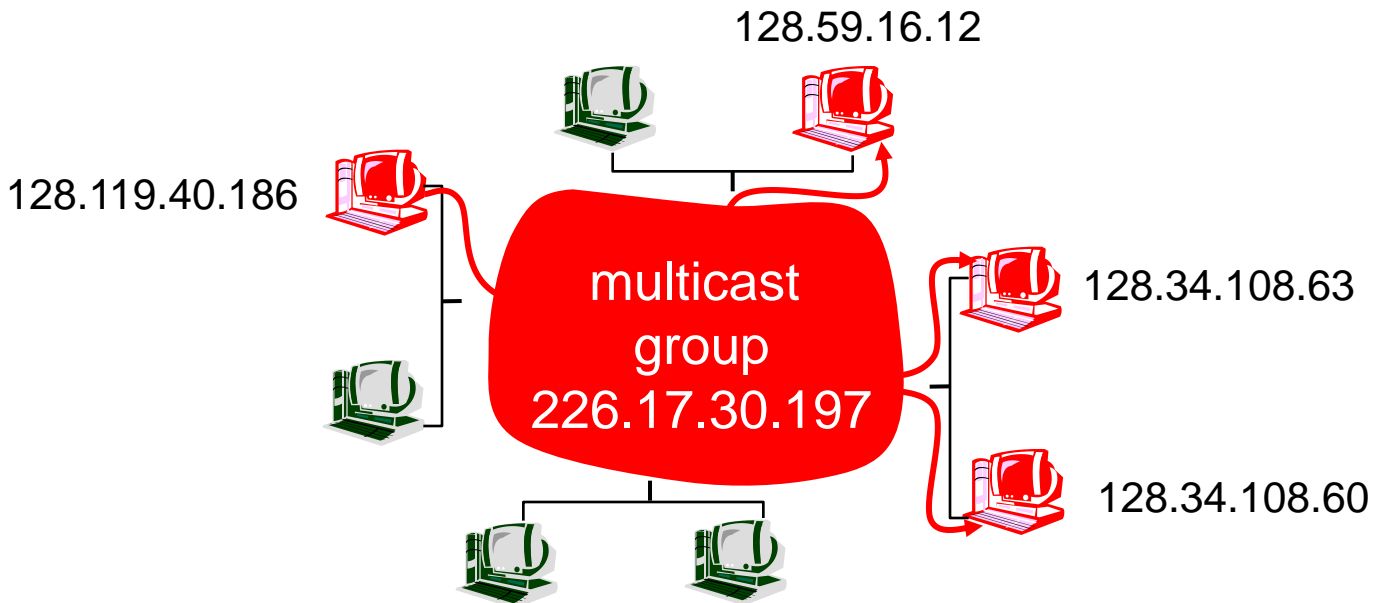
Suppose R6 chosen as center:



Network Layer II

- 4.6 Multicast
 - Broadcast routing
 - Multicast routing
 - Multicast routing protocols
- 4.7 Mobility
 - What is Mobility?
 - Network layer mobility concepts and principles
 - Mobile IP

Internet Multicast Service Model

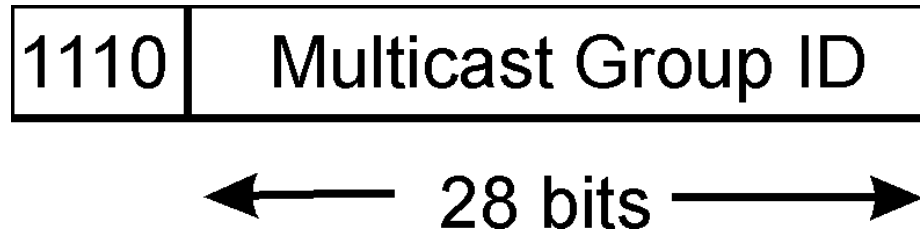


Multicast group concept: use of **indirection**

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

Multicast Groups

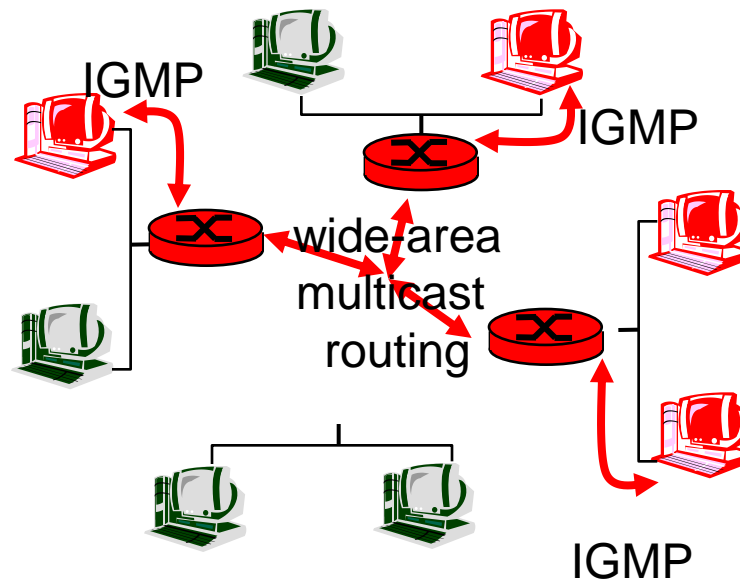
- Class D Internet addresses reserved for multicast:



- Host group semantics:
 - anyone can “join” (receive pkts) multicast group
 - anyone can send pkts to multicast group
 - no network-layer identification to hosts of the members
- Needed: infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group

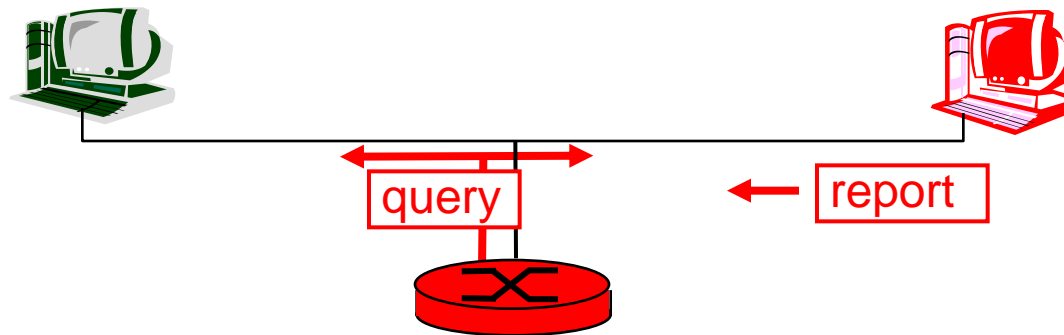
Joining a mcast group: two-step process

- **Local:** host informs local mcast router of a 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)



IGMP: Internet Group Management Protocol

- **Host:** sends IGMP report when application joins mcast group
 - IP_ADD_MEMBERSHIP socket option
 - host needs not explicitly “disjoin” group when leaving
- **Router:** sends IGMP query at regular intervals
 - host belonging to a mcast group must reply to query



Internet Multicasting Routing: DVMRP

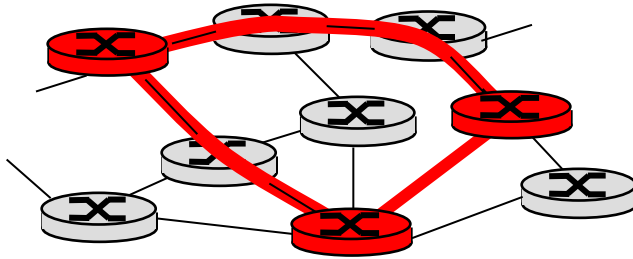
- DVMRP: distance vector multicast routing protocol, RFC1075
- flood and prune: reverse path forwarding, source-based tree
 - RPF tree based on DVMRP's own routing tables constructed by communicating DVMRP routers
 - no assumptions about underlying unicast
 - initial datagram to mcast group flooded everywhere via RPF
 - routers not wanting group: send upstream prune msgs

DVMRP: continued...

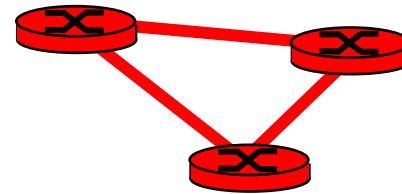
- soft state: DVMRP router periodically (1 min.) “forgets” branches are pruned:
 - mcast data again flows down unpruned branch
 - downstream router: re prune or else continue to receive data
- routers can quickly regraft to tree
 - following IGMP join at leaf
- odds and ends
 - commonly implemented in commercial routers
 - Mbone routing done using DVMRP

Tunneling

- Q: How to connect “islands” of multicast routers in a “sea” of unicast routers?



physical topology



logical topology

- mcast datagram encapsulated inside “normal” (non-multicast-addressed) datagram
- normal IP datagram sent through “tunnel” via regular IP unicast to receiving mcast router
- receiving mcast router de-capsulates pkt to get mcast datagram

PIM: Protocol Independent Multicast

- not dependent on any specific underlying unicast routing algorithm (works with all)
- two different multicast distribution scenarios :
 - Dense:
 - group members densely packed, in “close” proximity.
 - bandwidth more plentiful
 - Sparse:
 - # networks with group members small wrt # interconnected networks
 - group members “widely dispersed”
 - bandwidth not plentiful

Consequences of Sparse-Dense Dichotomy

○ Dense

- group membership by routers assumed until routers explicitly prune
- data-driven construction on mcast tree (e.g., RPF)
- bandwidth and non-group-router processing profligate

○ Sparse

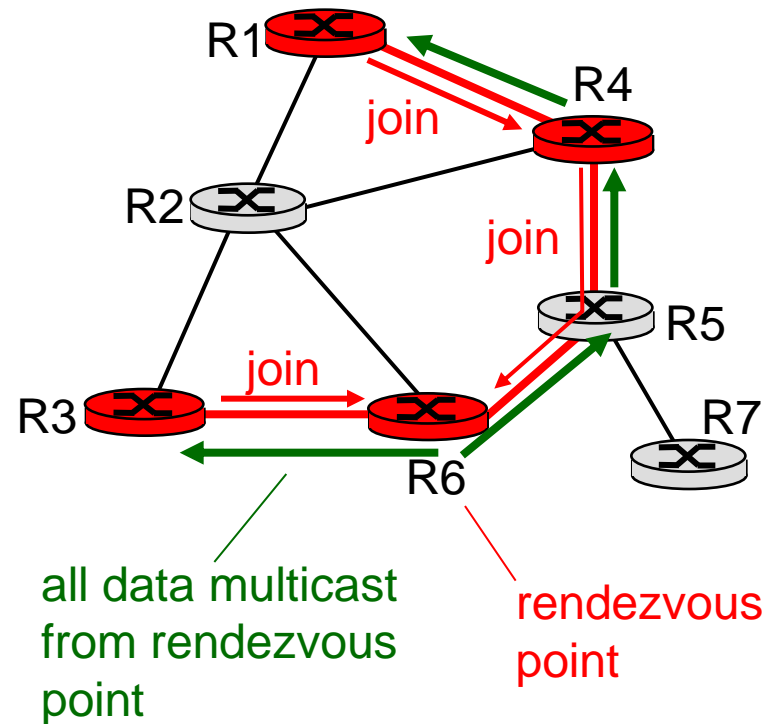
- no membership until routers explicitly join
- receiver-driven construction of mcast tree (e.g., center-based)
- bandwidth and non-group-router processing conservative

PIM- Dense Mode

- Flood-and-prune RPF, similar to DVMRP but
 - underlying unicast protocol provides RPF info for incoming datagram
 - less complicated (less efficient) downstream flood than DVMRP reduces reliance on underlying routing algorithm
 - has protocol mechanism for router to detect it is a leaf-node router

PIM - Sparse Mode

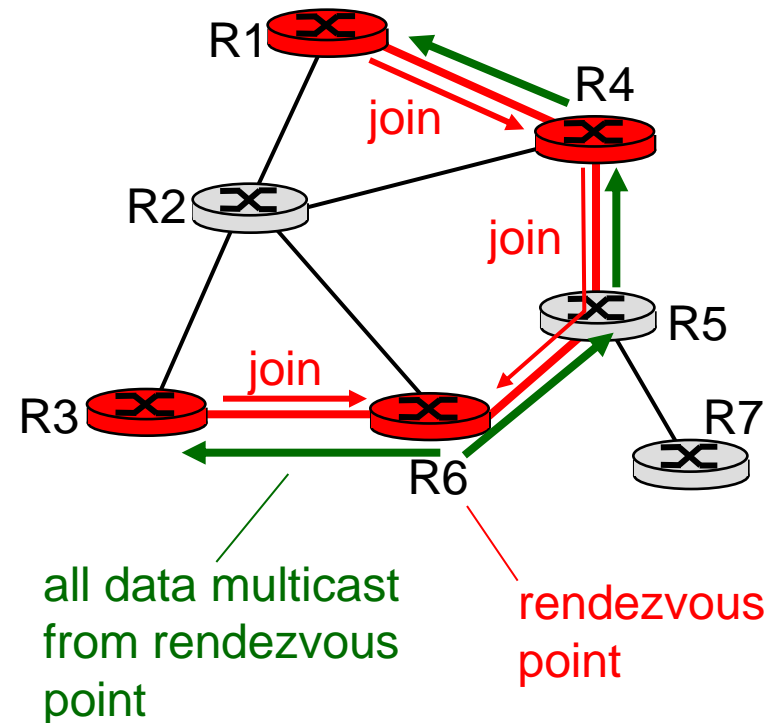
- center-based approach
- router sends *join* msg to rendezvous point (RP)
 - intermediate routers update state and forward *join*
- after joining via RP, router can switch to source-specific tree
 - increased performance: less concentration, shorter paths



PIM - Sparse Mode

sender(s):

- unicast data to RP, which distributes down RP-rooted tree
- RP can extend mcast tree upstream to source
- RP can send *stop* msg if no attached receivers
 - “no one is listening!”

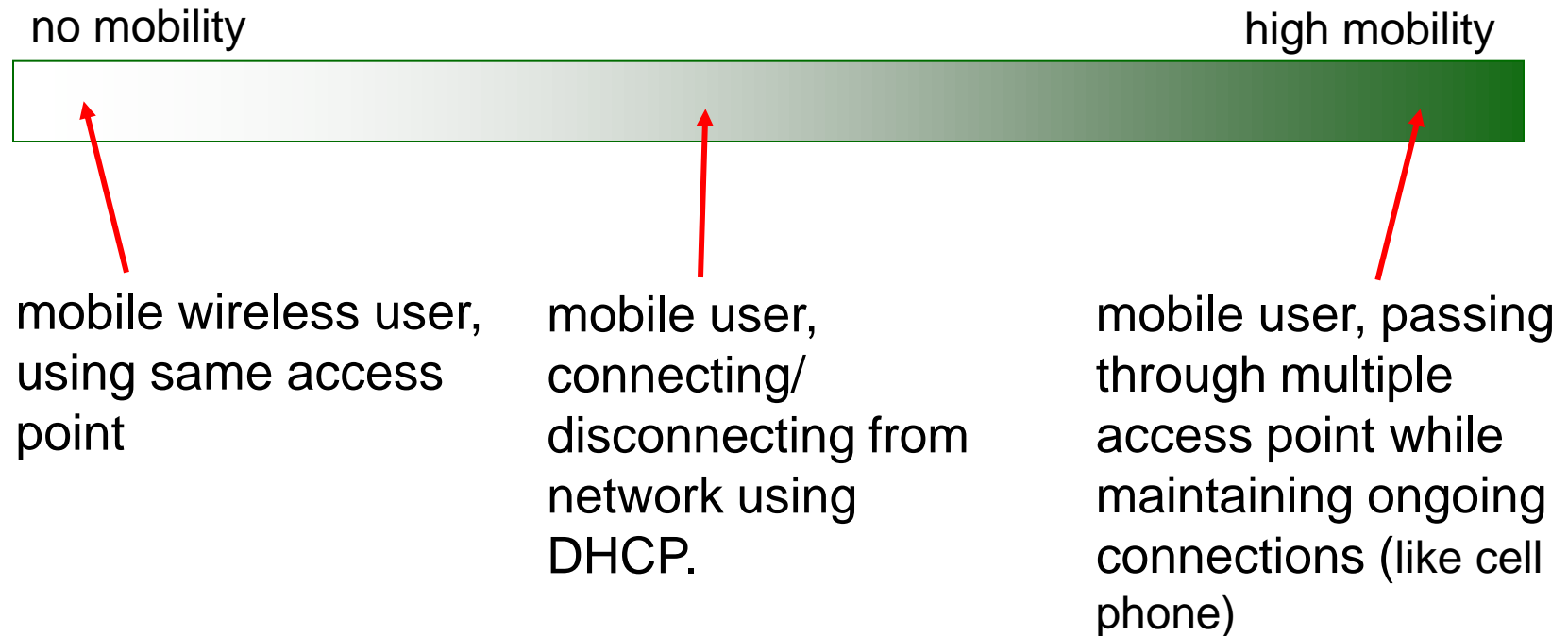


Network Layer II

- 4.6 Multicast
 - Broadcast routing
 - Multicast routing
 - Multicast routing protocols
- 4.7 Mobility
 - What is Mobility?
 - Network layer mobility concepts and principles
 - Mobile IP

What is mobility?

- spectrum of mobility, from the *network* perspective:

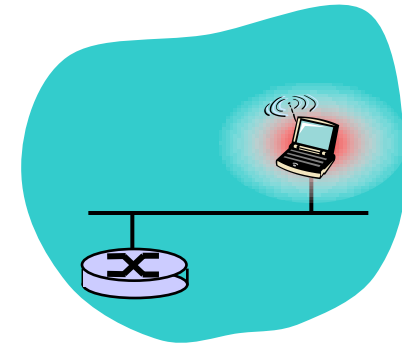
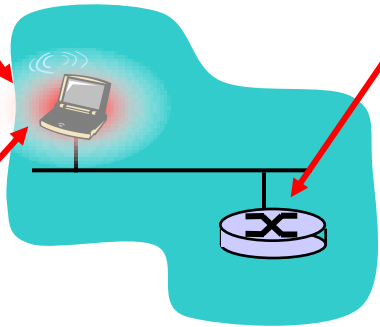


Mobility: Vocabulary

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

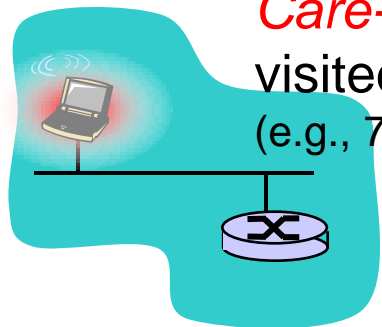


Mobility: more vocabulary

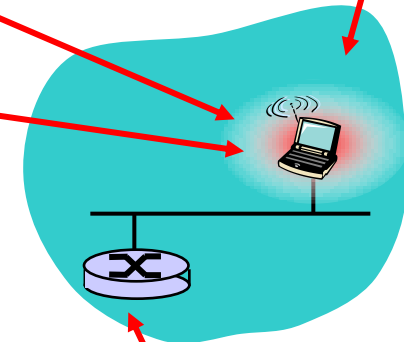
Permanent address: remains constant (e.g., 128.119.40.186)

visited network: network in which mobile currently resides (e.g., 79.129.13/24)

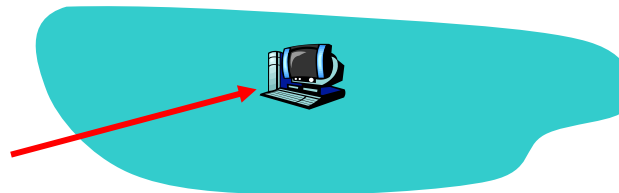
Care-of-address: address in visited network. (e.g., 79.129.13.2)



wide area network



correspondent: wants to communicate with

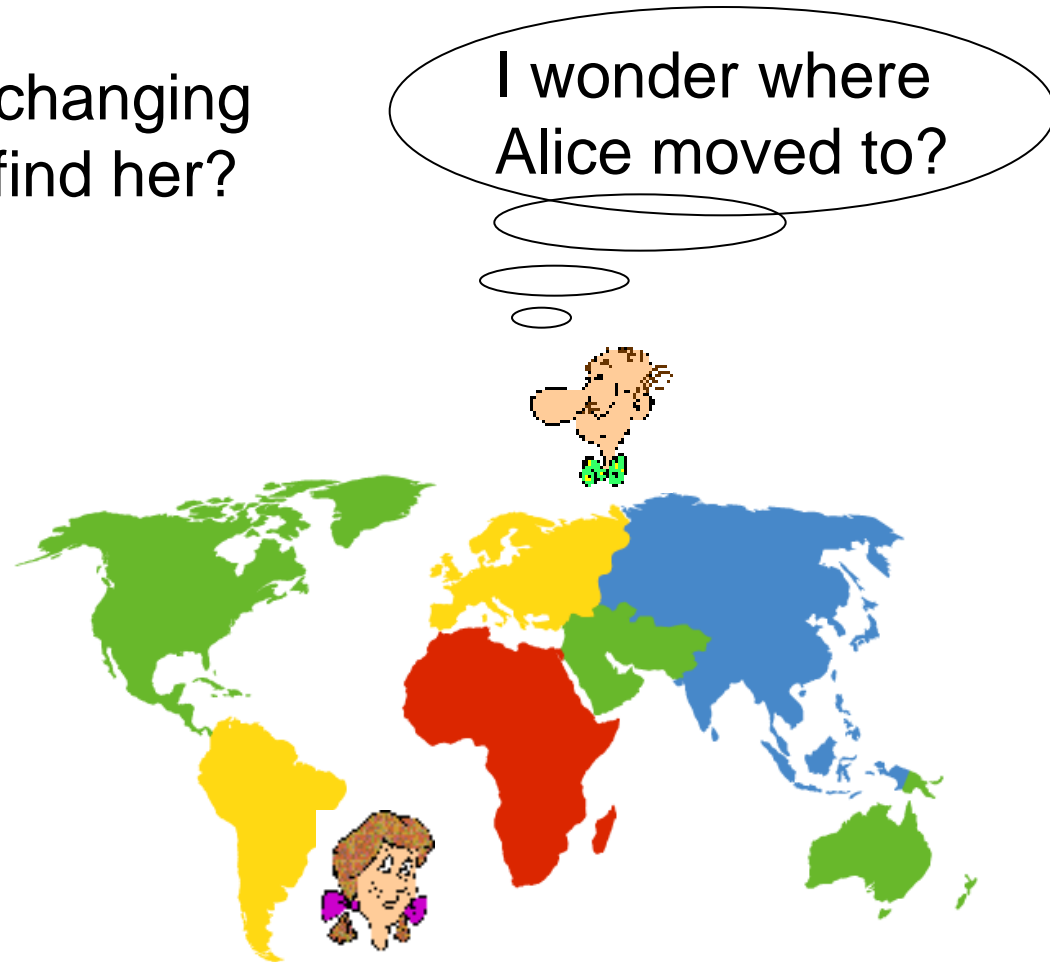


foreign agent: entity in visited network that performs mobility functions on behalf of mobile.

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?



Network Layer II

- 4.6 Multicast
 - Broadcast routing
 - Multicast routing
 - Multicast routing protocols
- 4.7 Mobility
 - What is Mobility?
 - Network layer mobility concepts and principles
 - Mobile IP

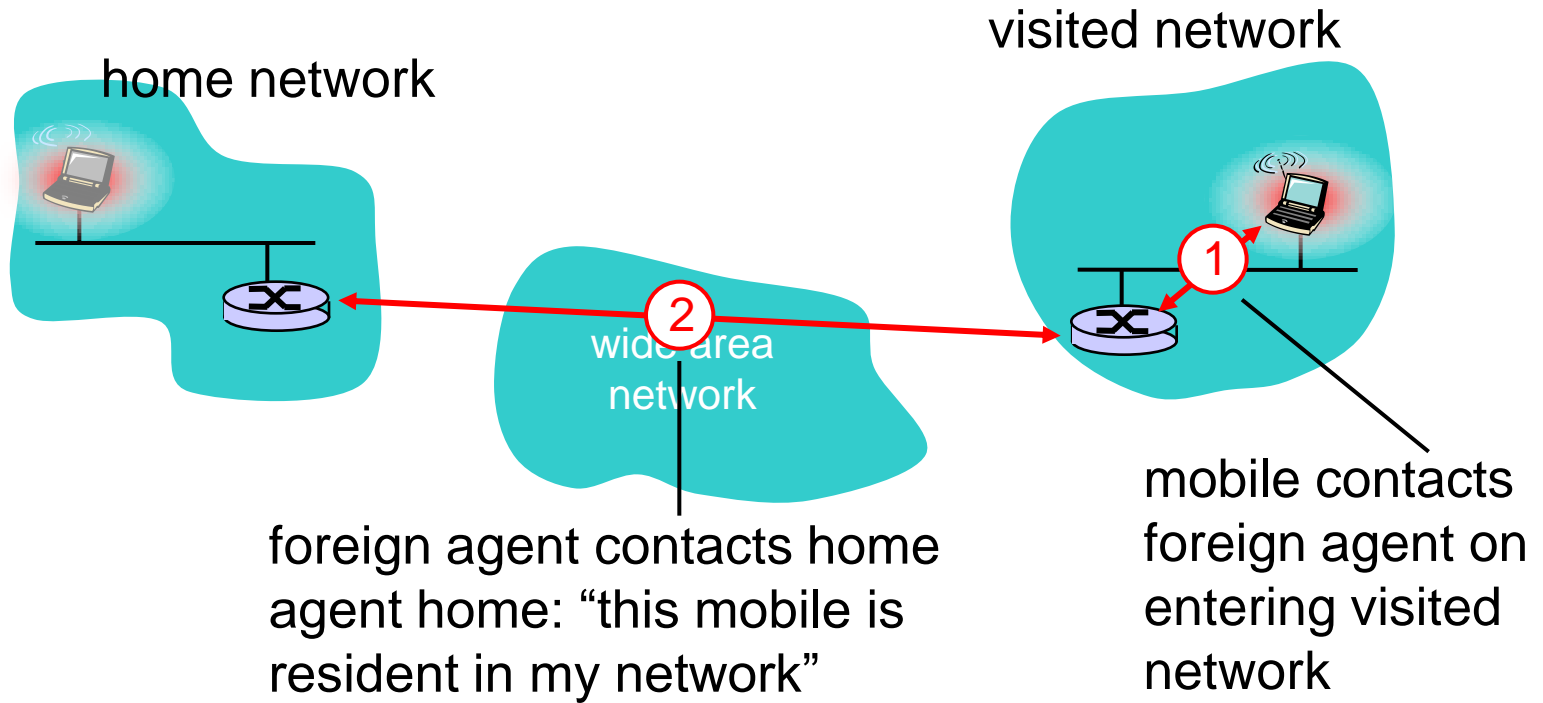
Mobility: approaches

- Let routing handle it
 - routers advertise permanent address of mobile-nodes via usual routing table exchange.
 - routing tables indicate where each mobile located
 - no changes to end-systems
 - **does not scale well!**

Mobility: approaches

- Let end-systems handle it
 - **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

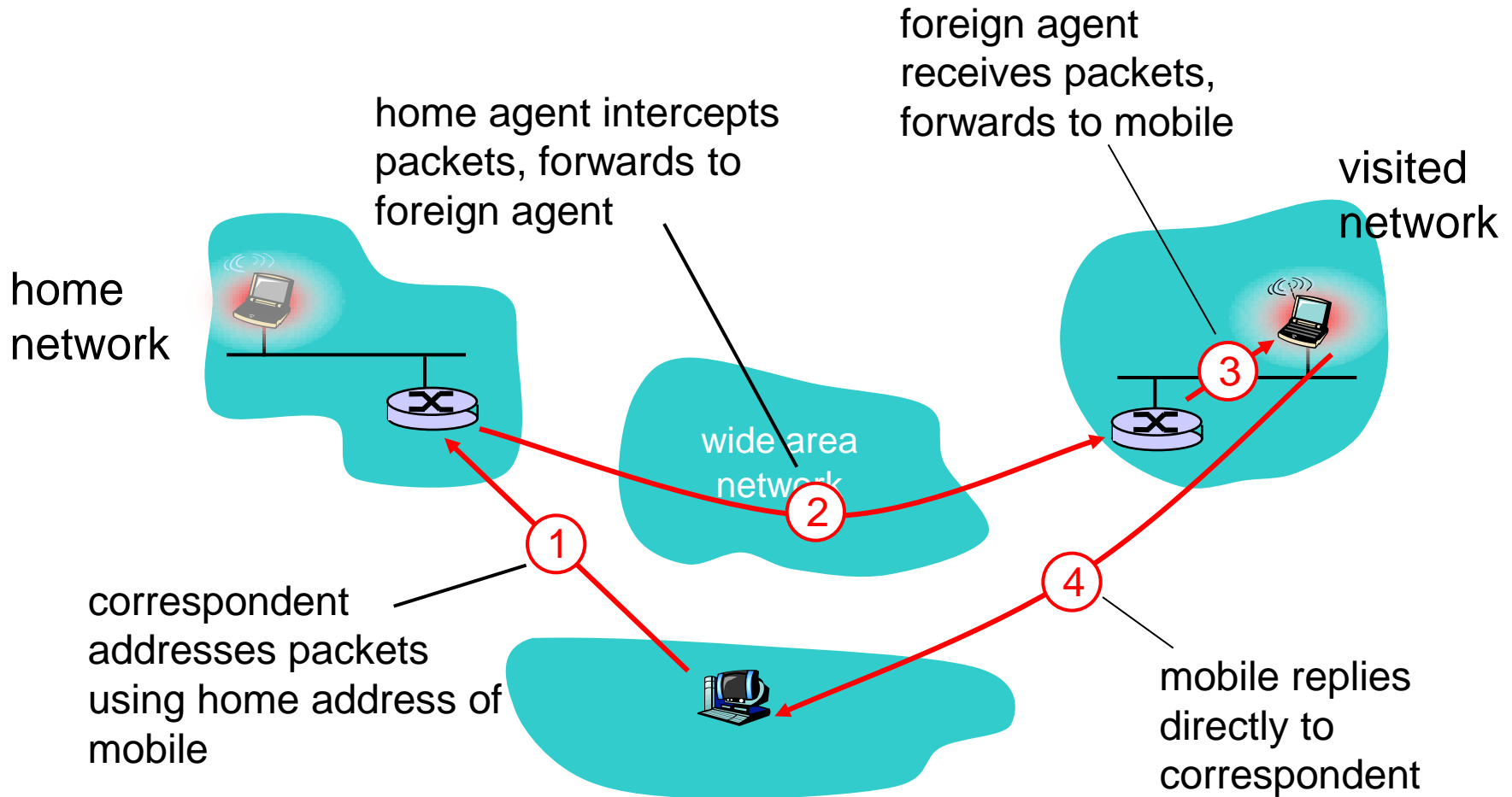
Mobility: registration



End result:

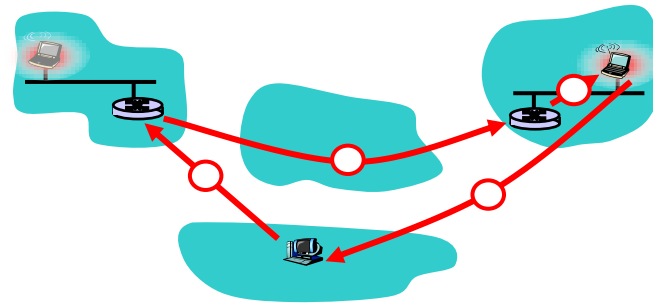
- Foreign agent knows about mobile
- Home agent knows location of mobile

Mobility via Indirect Routing



Indirect Routing: comments

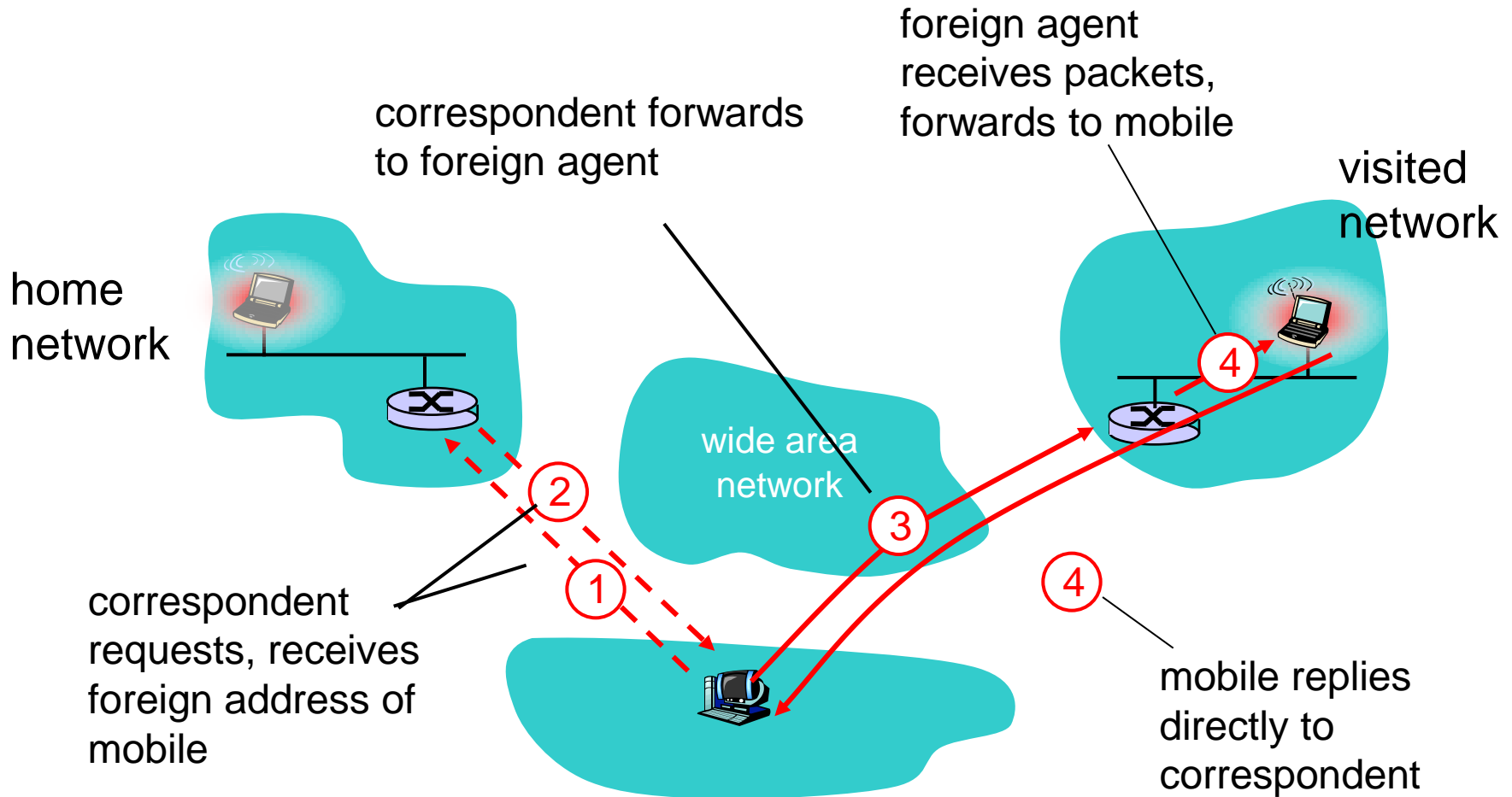
- Mobile uses two addresses:
 - **permanent address**: used by correspondent (hence mobile location is *transparent* to correspondent)
 - **care-of-address**: used by home agent to forward datagrams to mobile
- foreign agent functions may be done by mobile itself
- **triangle routing**: correspondent-home-network-mobile
 - inefficient when correspondent, mobile are in same network



Indirect Routing: moving between networks

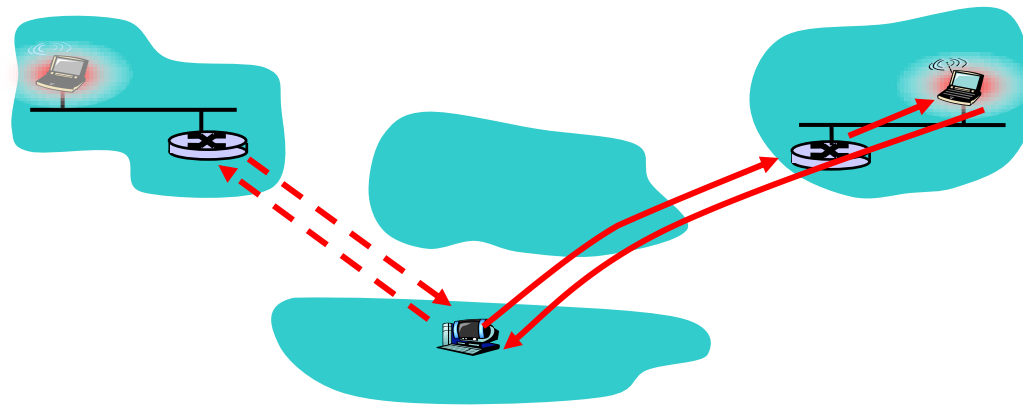
- suppose mobile user moves to another network
 - registers with new foreign agent
 - new foreign agent registers with home agent
 - home agent update care-of-address for mobile
 - packets continue to be forwarded to mobile (but with new care-of-address)
- mobility, changing foreign networks
transparent: *ongoing connections can be maintained!*

Mobility via Direct Routing



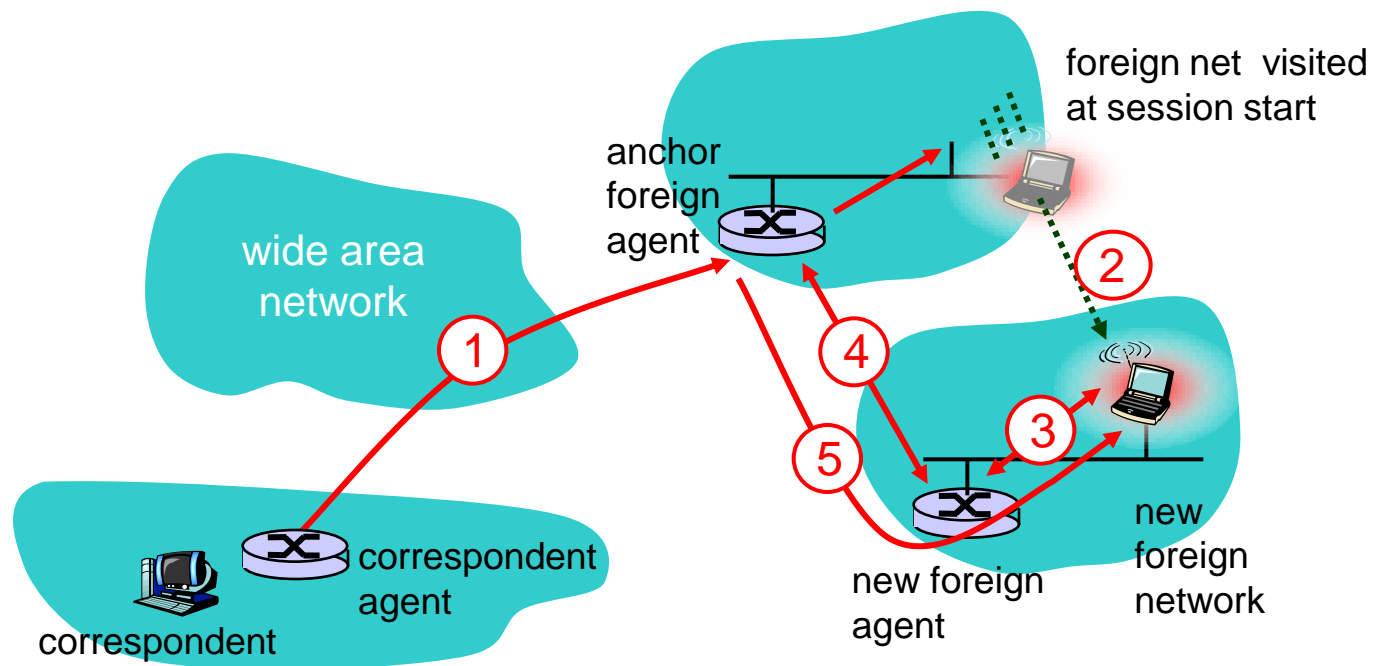
Mobility via Direct Routing: comments

- overcome triangle routing problem
- **non-transparent to correspondent:**
correspondent must get care-of-address from home agent
 - what if mobile changes visited network?



Accommodating mobility with direct routing

- anchor foreign agent: FA in first visited network
- data always routed first to anchor FA
- when mobile moves: new FA arranges to have data forwarded from old FA (chaining)



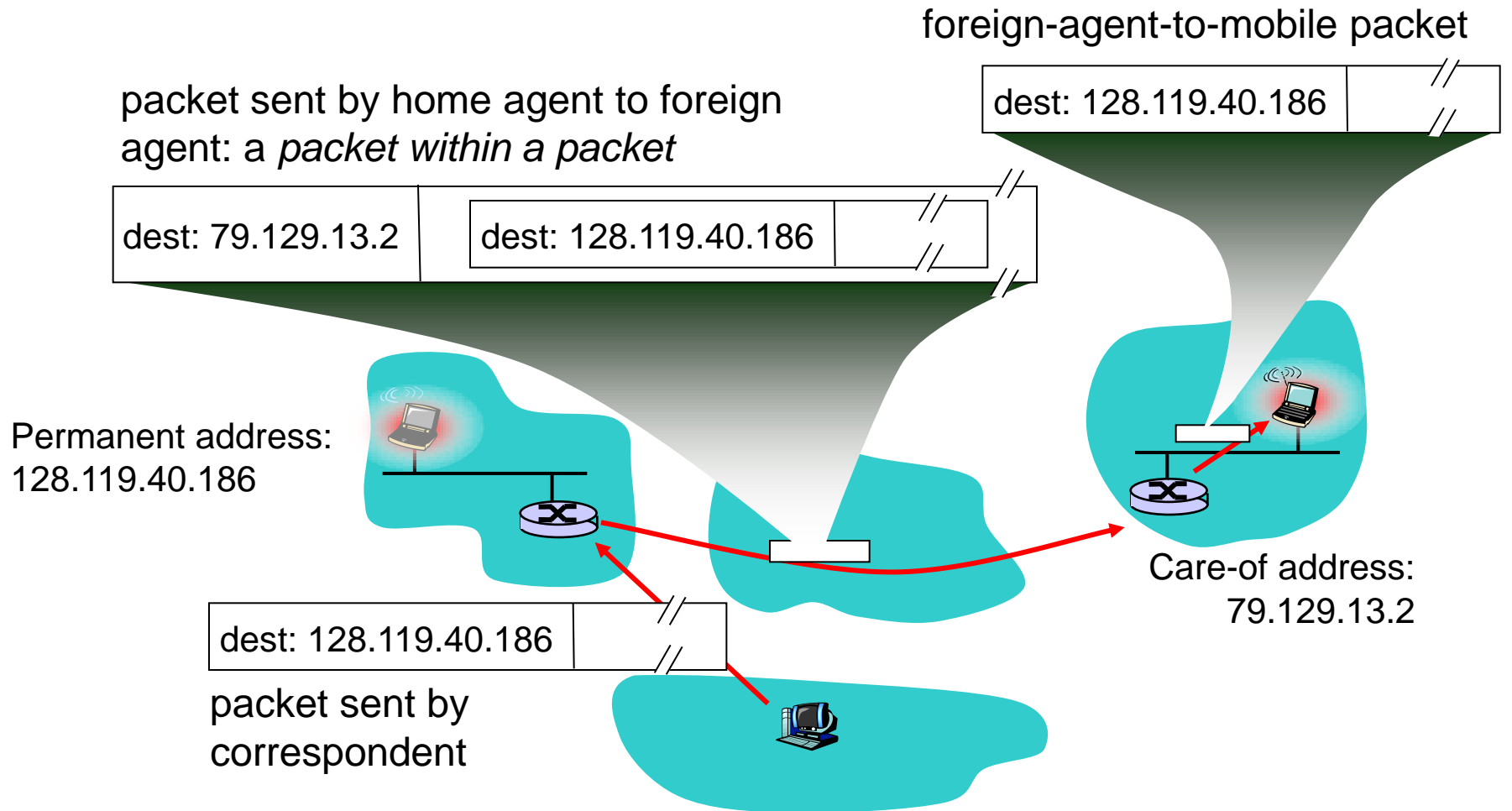
Network Layer II

- 4.6 Multicast
 - Broadcast routing
 - Multicast routing
 - Multicast routing protocols
- 4.7 Mobility
 - What is Mobility?
 - Network layer mobility concepts and principles
 - Mobile IP

Mobile IP

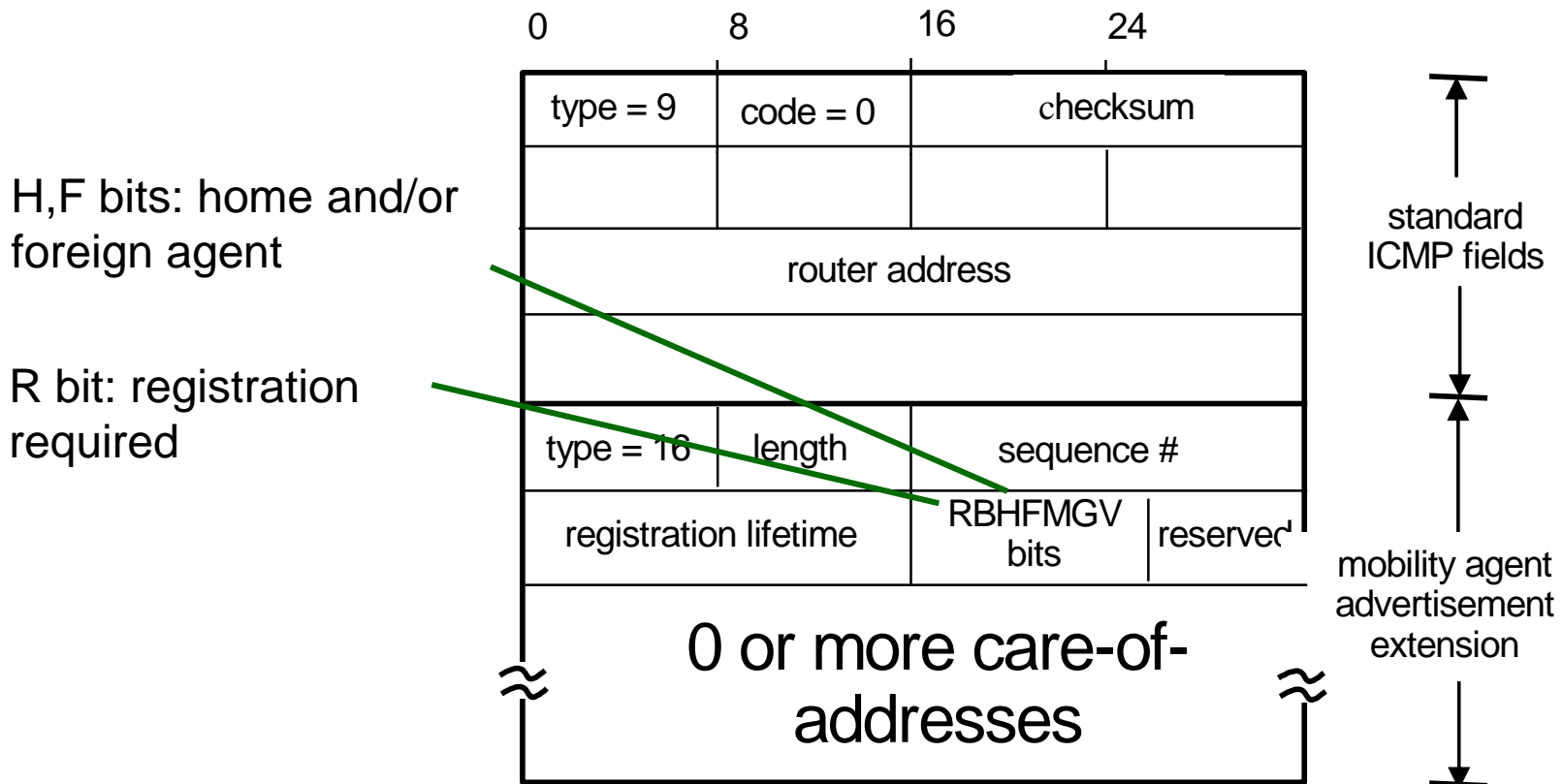
- RFC 3344
- has many features we've seen:
 - home agents, foreign agents, foreign-agent registration, care-of-addresses, encapsulation (packet-within-a-packet)
- three components to standard:
 - indirect routing of datagrams
 - agent discovery
 - registration with home agent

Mobile IP: indirect routing

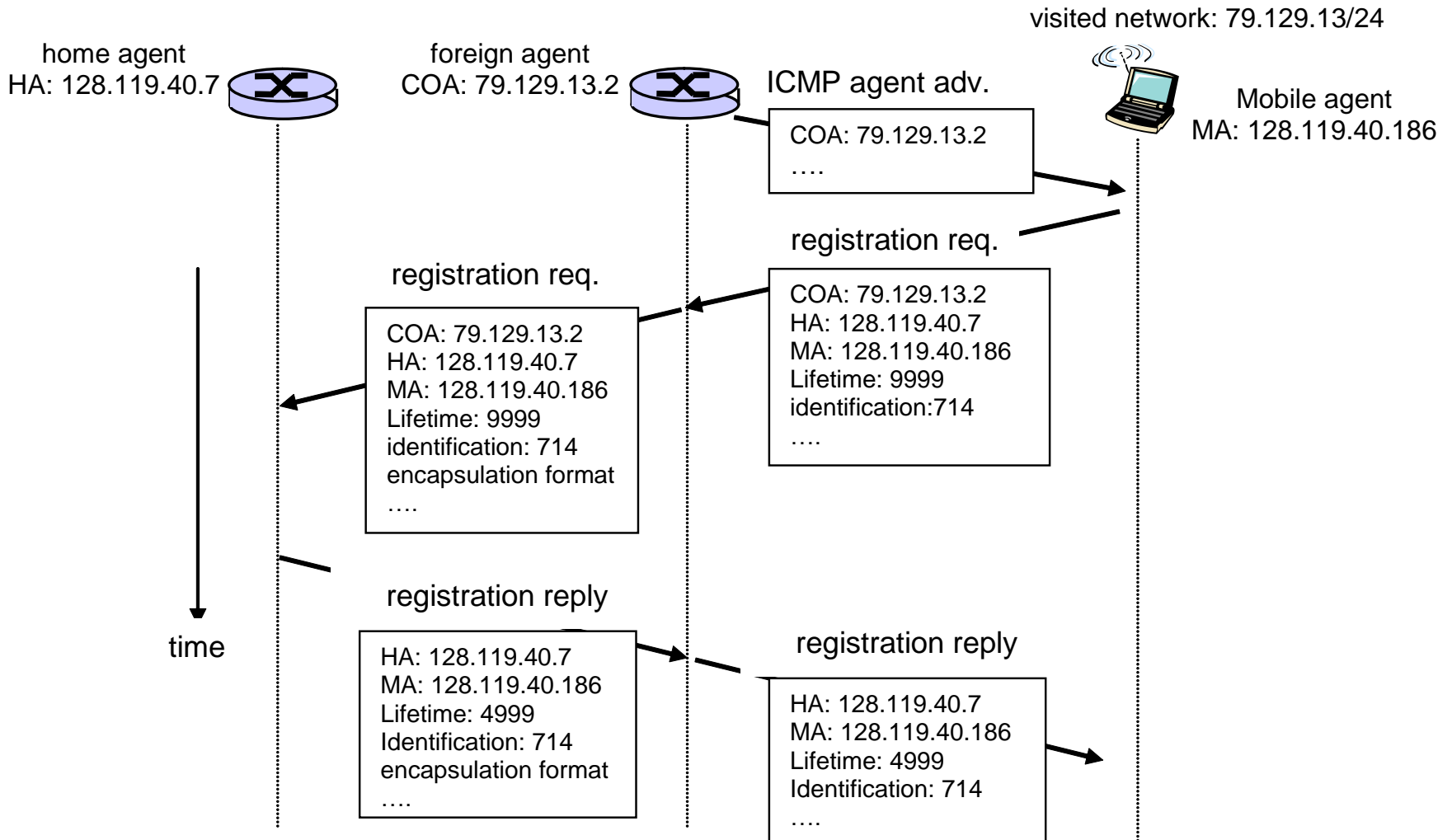


Mobile IP: agent discovery

- **agent advertisement:** foreign/home agents advertise service by broadcasting ICMP messages (typefield = 9)



Mobile IP: registration example



Thank you

Any questions?