Chapter 4 roadmap

4.1 Introduction and Network Service Models
4.2 VC and Datagram Networks
4.3 What’s Inside a Router
4.4 The Internet (IP) Protocol
4.5 Routing Algorithms
4.6 Routing in the Internet
4.7 Broadcast and Multicast Routing
4.8 Mobility

Multicast Routing: Problem Statement

Goal:
find a tree (or trees) connecting
routers having local mcast group members

- tree: not all paths between routers used
- source-based: different tree from each sender to rcvrs
- shared-tree: same tree used by all group members

Approaches for building mcast trees

Approaches:

- 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

...we first look at basic approaches, then specific protocols adopting these approaches
Shortest Path Tree

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

Reverse Path Forwarding

- rely on router’s knowledge of unicast shortest path from it to sender
- each router has simple forwarding behavior:

  \[
  \text{if } \begin{cases} \text{mcast datagram received on incoming link on shortest path back to center} \\
  \text{then flood datagram onto all outgoing links} \\
  \text{else ignore datagram} \end{cases}
  \]

Reverse Path Forwarding: example

- 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 mcast group members
- No need to forward datagrams down subtree
- "Prune" msg sent upstream by router with no downstream group members

Legend:
- Source
- Router with attached group member
- Router with no attached group member
- Prune message
- Links with multicast forwarding

Shared-Tree: Steiner Tree

- Steiner Tree: minimum cost tree connecting all routers with attached group members
- Problem is NP-complete
- Excellent heuristics exist
- Not used in practice:
  - Computational complexity
  - Information about entire network needed
  - Monolithic: rerun whenever a router needs to join/leave

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:

![Diagram of center-based trees]

LEGEND
- router with attached group member
- router with no attached group member
- path order in which join messages generated

Chapter 4 roadmap

4.1 Introduction and Network Service Models
4.2 VC and Datagram Networks
4.3 What's Inside a Router
4.4 The Internet (IP) Protocol
4.5 Routing Algorithms
4.6 Routing in the Internet
4.7 Broadcast and Multicast Routing
4.X Mobility

What is mobility?

- spectrum of mobility, from the network perspective:

  - no mobility
  - high mobility

  mobile user, using same access point
  mobile user, connecting/disconnecting from network using DHCP
  mobile user, passing through multiple access point while maintaining ongoing connections (like cell phone)
Mobility: Vocabulary

- **Home network**: permanent "home" of mobile (e.g., 128.119.40/24)
- **Permanent address**: address in home network, can always be used to reach mobile (e.g., 128.119.40.186)
- **Home agent**: entity that will perform mobility functions on behalf of mobile, when mobile is remote

Mobility: more vocabulary

- **Permanent address**: remains constant (e.g., 128.119.40.186)
- **Care-of-address**: address in visited network (e.g., 79.129.13.2)
- **Visited network**: network in which mobile currently resides (e.g., 79.129/12/24)
- **Home agent**: entity in visited network that performs mobility functions on behalf of mobile.
- **Correspondent**: wants to communicate with mobile (e.g., 128.119.40.186)

How do you contact a mobile friend:

- I wonder where Alice moved to?
- 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: approaches

- **Let routing handle it:** routers advertise permanent address of mobile-nodes-in-residence via usual routing table exchange.
  - routing tables indicate where each mobile located
  - no changes to end-systems
- **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: approaches

- **Let routing handle it:** routers advertise permanent address of mobile-nodes-in-residence via usual routing table exchange.
  - routing tables indicate where each mobile located
  - no changes to end-systems
- **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

Not scalable to millions of mobiles

Mobility: registration

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

End result:
- Foreign agent contacts home agent: “this mobile is resident in my network”
- Mobile contacts foreign agent on entering visited network
- Home agent contacts home network

Mobility via Indirect Routing

- 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

Forwarding datagrams to remote mobile

- packet sent by home agent to foreign agent: a packet within a packet
- packet sent by correspondent

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

- correspondent requests, receives foreign address of mobile
- correspondent forwards to foreign agent
- foreign agent receives packets, forwards to mobile
- mobile replies directly to correspondent

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)
- three components to standard:
  - agent discovery
  - registration with home agent
  - indirect routing of datagrams

Chapter 5: The Data Link Layer

Our goals:
- understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control: done!
- instantiation and implementation of various link layer technologies

Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM and MPLS
Link Layer: Introduction

Some terminology:
- hosts and routers are nodes
- communication channels that connect adjacent nodes along communication path are links
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a frame, encapsulates datagram

Data-link layer has responsibility of transferring datagram from one node to adjacent node over a link.

Link layer: context

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide rdt over link

Transportation analogy
- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Link Layer Services

Framing, link access:
- encapsulate datagram into frame, adding header, trailer
- channel access if shared medium
- “MAC” addresses used in frame headers to identify source, dest
  - different from IP address!

Reliable delivery between adjacent nodes
- we learned how to do this already (chapter 3)
- seldom used on low bit error link (fiber, some twisted pair)
- wireless links: high error rates
  - Q: why both link-level and end-end reliability?
Link Layer Services (more)

- **Flow Control:**
  - Pacing between adjacent sending and receiving nodes

- **Error Detection:**
  - Errors caused by signal attenuation, noise.
  - Receiver detects presence of errors:
    - Signals sender for retransmission or drops frame

- **Error Correction:**
  - Receiver identifies and corrects bit error(s) without resorting to retransmission

- **Half-duplex and Full-duplex**
  - With half duplex, nodes at both ends of link can transmit, but not at same time

Adaptors Communicating

- Link layer implemented in “adaptor” (aka NIC)
  - Ethernet card, PCMCI card, 802.11 card

- Sending side:
  - Encapsulates datagram in a frame
  - Adds error checking bits, rdt, flow control, etc.

- Receiving side:
  - Looks for errors, rdt, flow control, etc.
  - Extracts datagram, passes to receiving node

- Adapter is semi-autonomous

- Link & physical layers