More on Network Layer

Outline:
- What is inside a router?
- IPv6
- Multicast routing
  - basic concept
  - approaches
  - protocols

Router Architecture Overview

Two key router functions:
- run routing algorithms/protocol (RIP, OSPF, BGP)
- switching datagrams from incoming to outgoing link

Input Port Functions

Decentralized switching:
- given datagram dest., lookup output port using routing table in input port memory
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Input Port Queuing

- Fabric slower that input ports combined -> queuing may occur at input queues
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward
- queuing delay and loss due to input buffer overflow!

Three types of switching fabrics

- Memory
- Bus
- Crossbar

Output Ports

- Buffering required when datagrams arrive from fabric faster than the transmission rate
- queuing delay and loss due to output port buffer overflow!
Summary on Router Construction

- Different types of switching fabrics
  - memory? bus? crossbar
  - increasing performance, increasing cost, decreasing flexibility
- Resource contention can be everywhere
  - switching fabric, output line speed
  - solution: buffers on input and output ports
  - buffer overflow ⇒ dropping packets

IPv6

- Initial motivation: address depletion ⇒ 32-bit address space completely allocated by 2008.
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
  - new “anycast” address: route to “best” of several replicated servers
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no “flag days”
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Two proposed approaches:
  - Dual Stack: some routers with dual stack (v6, v4) can “translate” between formats
  - Tunneling: IPv6 carried as payload in IPv4 datagram among IPv4 routers

Outline

- What is inside a router?
- IPv6
- Multicast routing
  - basic concept
  - approaches
  - protocols
Multicast

- Multicast: act of sending datagram to multiple receivers with single “transmit” operation
- Question: how to achieve multicast

Multicast via unicast

- Source sends N unicast datagrams, one addressed to each of N receivers

Network Multicast

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

Internet Multicast Service Model

- Class D Internet addresses reserved for multicast
- Service semantics:
  - Anyone can “join” (receive) multicast group
  - Anyone can send to multicast group
- What is needed? Infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group

Multicast Groups

- Class D Internet addresses reserved for multicast:
  - 1110 Multicast Group ID
  - 28 bits
- Service semantics:
  - Anyone can join (receive) multicast group
  - Anyone can send to multicast group
- What is needed? Infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group

Joining A Mcast Group: Two-step

- Local: host informs local multicast router of desire to join group: IGMP (Internet Group Management Protocol)
- Wide area: local router interacts with other routers to receive multicast datagram flow

IGMP: Internet Group Management Protocol

- Host: sends IGMP report when application joins mcast group
  - IP_ADD_MEMBERSHIP socket option
  - Host need not explicitly “unjoin” group when leaving
- Router: sends IGMP query at regular intervals
  - Host belonging to a mcast group must reply to query
Wide-area Multicast Routing
- **Goal:** find path/paths (tree) connecting routers having local mcast group members
- **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

**Shared tree**

**Source-based trees**

### 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:
  
  \[
  \text{If} \ (\text{mcast datagram received on incoming link on shortest path back to center}) \\
  \text{then} \ \text{flood datagram onto all outgoing links} \\
  \text{else} \ \text{ignore datagram}
  \]

### Reverse Path Forwarding: example
- result is a source-specific reverse SPT

### Reverse Path Forwarding: pruning
- forwarding tree contains subtrees with no mcast group members
  - no need to forward datagrams down subtree
  - “prune” msgs sent upstream by router with no downstream group members

### Shared-Tree: Steiner Tree
- Steiner Tree: minimum cost tree connecting all routers with attached group members
- problem is NP-complete
- excellent heuristics exists
- 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:

Internet Multicasting Routing

- DVMRP: distance vector multicast routing protocol, based on RPF (reverse path forwarding) and pruning
  - first and the most commonly supported
- PIM: protocol independent multicast
  - separate the dense and sparse mode
  - PIM dense-mode is based on RPF/pruning, similar to DVMRP
  - PIM sparse-mode is based on Center-based Trees

Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).