CS 640 Introduction to Computer Networks

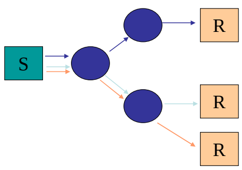
Lecture14

Today's lecture

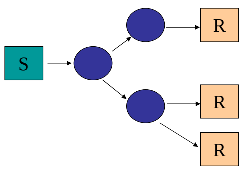
- Network layer multicast
- Transport layer – UDP

One to many communication

- Application level one to many communication
- Multiple unicasts



- IP multicast



Types of Multicast

- At network-layer
  - Topic of this lecture
- Sequence of unicasts
  - Separate streams of unicast traffic for each destination from the source
  - Does not require support at network-layer
- Application-layer multicast
  - Based on unicasts
  - Constructs an overlay structure
  - Source unicasts to a subset of receives, these receivers unicast to another subset, which unicast to another subset and so on to reach the whole multicast group

Why Multicast

- When sending same data to multiple receivers
  - Better bandwidth utilization
  - Less host/router processing
  - Quicker to join
- Application
  - Video/Audio broadcast (One sender)
  - Video conferencing (Many senders)
  - Real time news distribution
  - Interactive gaming

IP multicast service model

- Invented by Steve Deering (PhD. 1991)
  - It's a different way of routing datagrams
- RFC1112 : Host Extensions for IP Multicasting - 1989
- Senders transmit IP datagrams to a "host group"
- "Host group" identified by a class D IP address
- Members of host group can be anywhere in the Internet
- Members join and leave the group and indicate this to the routers
- Senders and receivers distinct (a sender need not be a receiver)
- Routers listen to all multicast addresses and use multicast routing protocols to manage groups

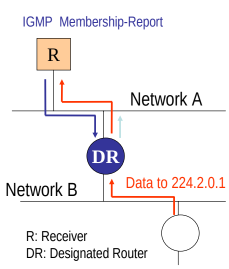
IP multicast group address

- Things are a little tricky in multicast since receivers can be *anywhere*
- Class D address space
  - high-order three 3bits are set
  - 224.0.0.0 ~ 239.255.255.255
- Allocation is essentially random – any class D can be used
  - Nothing prevents an app from sending to any multicast address
  - Customers end hosts and ISPs are the ones who suffer
- Some well-known address have been designated
  - RFC1700
  - 224.0.0.0 ~ 224.0.0.25
- Standards are evolving

Getting Packets to End Hosts

- Packets from remote sources will be forwarded by IP routers onto a local network only if they know there is at least one receiver for that group on that network
- Internet Group Management Protocol (IGMP, RFC2236)
  - Used by end hosts to signal that they want to join a specific multicast group
  - Used by *routers* to discover what groups have have interested member hosts on each network to which they are attached.
  - Implemented directly over IP

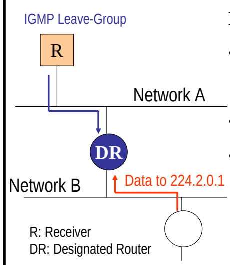
IGMP – Joining a group



Example : R joins to Group 224.2.0.1

- R sends **IGMP Membership-Report to 224.2.0.1**
- DR receives it. DR will start forwarding **packets for 224.2.0.1** to Network A
- DR periodically sends **IGMP Membership-Query to 224.0.0.1** (ALL-SYSTEMS.MCAST.NET)
- R answers **IGMP Membership-Report to 224.2.0.1**

IGMP – Leaving a group



Example : R leaves from a Group 224.2.0.1

- R sends **IGMP Leave-Group**
- DR receives it.
- DR stops forwarding **packets for 224.2.0.1** to Network A if no more 224.2.0.1 group members on Network A.

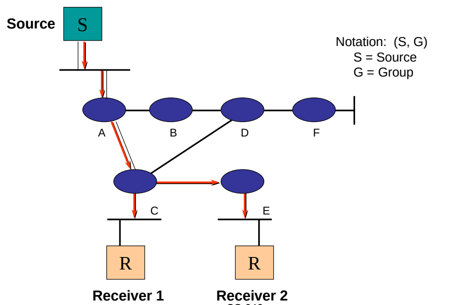
Challenges in the multicast model

- How can a sender restrict who can receive?
  - need authentication, authorization
  - encryption of data
  - key distribution
  - still an active area of research

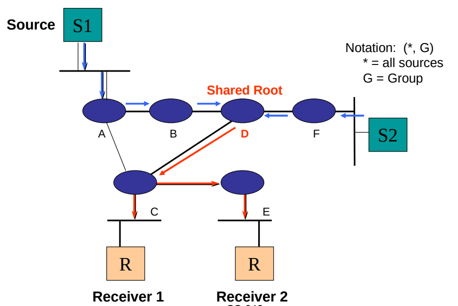
IP multicast routing

- Purpose: share group information among routers, to implement better routing for data distribution
- Distribution tree structure
  - Source tree vs. shared tree
- Data distribution policy
  - Opt in (ACK) type vs. opt out (NACK) type
- Routing protocols used together with IGMP

Source distribution tree



Shared distribution tree



Source tree characteristics

- Source tree
  - More memory O (G x S) in routers
  - Optimal path from source to receiver, minimizes delay
  - Good for small number of senders, many receivers (e.g. Radio broadcasting application)

Shared tree

- Characteristics
  - Less memory O (G) in routers
  - Sub-optimal path from source to receiver, may introduce extra delay (source to root)
  - May have duplicate data transfer (possible duplication of a path from source to root and a path from root to receivers)
- Good for
  - Environments where most of the shared tree is the same as the source tree
  - Many senders with low bandwidth (e.g. shared whiteboard)

Data distribution policy

- Opt out (NACK) type
  - Start with "broadcasting" then prune branches with no receivers, to create a distribution tree
  - Lots of wasted traffic when there are only a few receivers and they are spread over wide area
- Opt in (ACK) type
  - Forward only to the hosts which explicitly joined to the group
  - Latency of join propagation

Protocol types

- Dense mode protocols
  - Assumes dense group membership
  - Source distribution tree and NACK type
  - **DVMRP** (Distance Vector Multicast Routing Protocol)
  - **PIM-DM** (Protocol Independent Multicast, Dense Mode)
  - Example: Company-wide announcement
- Sparse mode protocol
  - Assumes sparse group membership
  - Shared distribution tree and ACK type
  - **PIM-SM** (Protocol Independent Multicast, Sparse Mode)
  - Examples: Space Shuttle Launch in the '90s

RPF(reverse path forwarding)

- RPF algorithm takes advantage of the IP routing table to compute a multicast tree for each source.
- RPF check
  1. When a multicast packet is received, note its source (*S*) and interface (*I*)
  2. If *I* belongs to the shortest path from *S*, forward to all interfaces except *I*
  3. If test in step 2 is false, drop the packet
- Packet is never forwarded back out the RPF interface!

DVMRP
exchange distance vectors

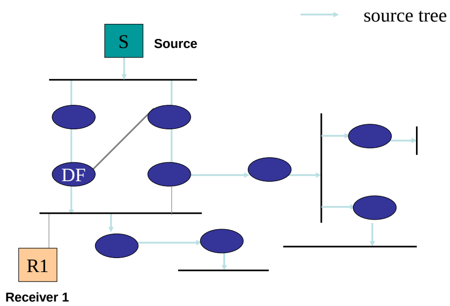
- If not all routers in the network support DVMRP, then unicast tunnels are used to connect multicast enabled networks
- Each router maintains a ‘multicast routing table’ by exchanging distance vector information among routers
  - First multicast routing protocol ever deployed in the Internet
    - Similar to RIP
  - Constructs source trees for each group using reverse path forwarding
    - Each tree provides a shortest path between source and each receiver
- There is a “designated forwarder” in each subnet
  - Multiple routers on the same LAN select designated forwarder by lower metric or lower IP address (discover when exchanging metric info.)
- Once tree is created, it is used to forward messages from source to receivers

DVMRP
broadcast & prune

- Flood multicast packets based on RPF (Reverse path forwarding) rule to all routers.
- Leaf routers check and send prune message to upstream router when no group member is on their network
- Upstream router prunes the interface with no dependent downstream router.
- *Graft* message to create a new branch for late participants
- Restart forwarding after prune lifetime (standard : 720 minutes)
- draft-ietf-idmr-dvmrp-v3-09.txt (September 1999)

DVMRP (1)

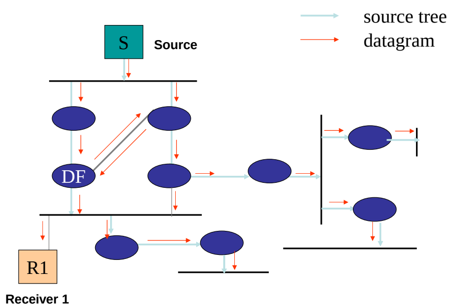
form a source tree by exchanging metric



CS 640

DVMRP (2)

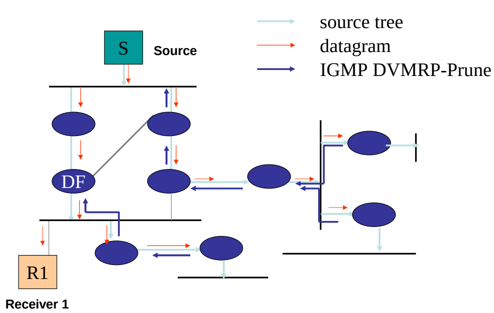
broadcast



CS 640

DVMRP (3)

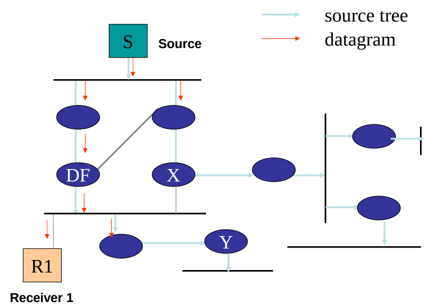
prune



CS 640

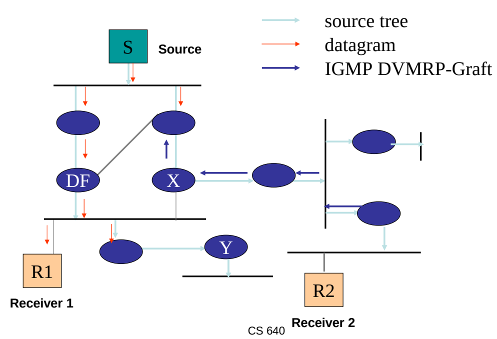
DVMRP (4)

X and Y pruned



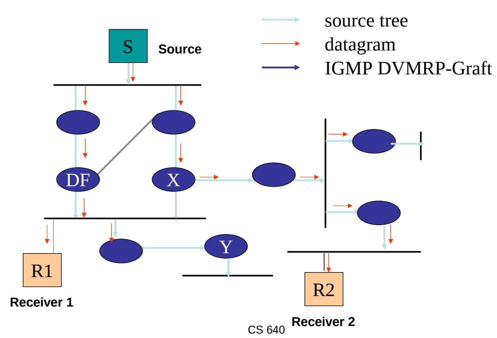
DVMRP (4)

New member



DVMRP (4)

New branch



Today's lecture

- Network layer multicast
- Transport layer – UDP

Layering and Encapsulation Revisited

- Each layer relies on layers below to provide services in black box fashion
  - Layering makes complex systems easier to understand & specify
  - Makes implementation more flexible
  - Can make implementation bigger and less efficient
  - Layers are implemented by protocols – rules for communication
- Data from applications moves up and down protocol stack
  - Application level data is chopped into packets (segments)
  - Encapsulation deals with attaching headers at layers 2, 3, 4

End-to-End Protocols

- Underlying network is *best-effort* so it can:
  - drop messages
  - re-orders messages
  - delivers duplicate copies of a given message
  - deliver messages after an arbitrarily long delay
- Common end-to-end services do:
  - guarantee message delivery
  - deliver messages in the same order they are sent
  - deliver at most one copy of each message
  - support synchronization
  - allow the receiver to flow control the sender
  - support multiple application processes on each host

Basic function of transport layer

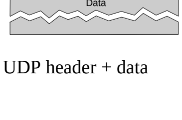
- How can processes on different systems get the right messages?
- *Ports* are numeric locators which enable messages to be demultiplexed to proper process.
  - Ports are addresses on individual hosts, not across the Internet
- Ports are established using *well-know* values first
  - Port 80 = http, port 53 = DNS
- Ports are typically implemented as message queues
- Simplest function of the transport layer is multiplexing/ demultiplexing of messages

Other transport layer functions

- Connection control
  - Setting up and tearing down communication between processes
- Error detection within packets
  - Checksums
- Reliable, in order delivery of packets
  - Acknowledgement schemes
- Flow control
  - Matching sending and receiving rates between end hosts
- Congestion control
  - Managing congestion in the network

User Datagram Protocol (UDP)

- Unreliable and unordered *datagram* service
- Adds multiplexing/demultiplexing
- Adds reliability through optional checksum
- No flow or congestion control
- Endpoints identified by ports
  - servers have *well-known* ports
  - see **`/etc/services`** on Unix
- Header format



- Optional checksum
  - Computed over pseudo header + UDP header + data

UDP Checksums

- Optional in current Internet
- Covers payload + pseudoheader
- Pseudoheader consists of 3 fields from IP header: protocol number (TCP or UDP), IP src, IP dst and UDP length field
  - Pseudoheader enables verification that message was delivered between correct source and destination.
  - IP dest address was changed during delivery, checksum would reflect this
- UDP uses the same checksum algorithm as IP

UDP in practice

- Minimal requirements make UDP very flexible
  - Any end-to-end protocol can be implemented
    - Remote Procedure Calls (RPC)
    - TCP can be implemented using UDP
- Examples
  - Most commonly used in multimedia applications
    - These are frequently more robust to loss
  - RPC's
  - Many others…