Course Road Map & Next Steps

Weeks 1-6
- Core subject material covered
  - (network arch, MAC, Ethernet Switching, IPv4, IPv6, Routing, IP Multicast)

Weeks 6-7
- Mid-term review
- Mini-net small projects completed (+in-class session)
- Protocol project intro.

Weeks 8-9
- Unified Protocol spec Complete
- Interactive Protocol Design Sessions

Weeks 10-14
- Additional topics covered (QoS, mobility, TCP & future networks)
- Focus on protocol design project work
- Protocol Project Submissions
- Final Exam (multiple choice)
## Network Protocols Studied so far

<table>
<thead>
<tr>
<th>Network</th>
<th>Architecture</th>
<th>Algorithm</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.3 Ethernet</td>
<td>Shared link</td>
<td>CSMA/CD</td>
<td>Ethernet packet</td>
<td>6B S/D addresses</td>
</tr>
<tr>
<td>802.11 WiFi</td>
<td>Shared radio link</td>
<td>CSMA/CA</td>
<td>Ethernet+ packet</td>
<td>S/D and intermediate</td>
</tr>
<tr>
<td>SW Ethernet</td>
<td>Campus net</td>
<td>Self-learning bridge, Spanning Tree</td>
<td>Ethernet packet, control msg</td>
<td>6B S/D, control, etc.</td>
</tr>
<tr>
<td>Basic IP</td>
<td>Wide area network</td>
<td>Bellman Ford/DV, Djikstra</td>
<td>IPv4 packet ICMP OSPF, RIP</td>
<td>4B S/D; Class A,B,C addr TTL etc.</td>
</tr>
<tr>
<td>Advanced IP</td>
<td>Global Network</td>
<td>DV with full path + local policy</td>
<td>BGP packets: update etc.</td>
<td>CIDR S/D addresses; AS #, path vectors, ...</td>
</tr>
<tr>
<td>IP Multicast</td>
<td>Wide area net</td>
<td>...Today’s topic</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IP Multicast

• Introduction
• Internet Group Management Protocol (IGMP)
• Routing Protocols
  – Intra-domain (DVMRP, MOSPF, PIM)
  – Inter-domain (MBGP, MSDP) – not covered here
Multicast: one sender to many receivers

- **Multicast**: act of sending datagram to multiple receivers with single “transmit” operation
  - One-to-many, many-to-many

- **Question**: how to achieve multicast

**Multicast via unicast**
- source sends N unicast datagrams, one addressed to each of N receivers
  - Redundant traffic around sender
  - Keep track of all the IP addresses to send to

Routers forward unicast datagrams
Multicast: one sender to many receivers

- **Multicast:** act of sending datagram to multiple receivers with single “transmit” operation
  - One-to-many, many-to-many

- **Question:** how to achieve multicast

**Network multicast (IP Multicast)**

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

Multicast routers (red) duplicate and forward multicast datagrams
Multicast: one sender to many receivers

- **Multicast**: act of sending datagram to multiple receivers with single “transmit” operation
  - One-to-many, many-to-many

- **Question**: how to achieve multicast

**Application-layer multicast** (P2P)
- end systems (“hosts”) involved in multicast copy and forward unicast datagrams among themselves
  - “host” becomes router

P2P hosts duplicate and forward multicast datagrams
Internet Multicast Service Model

multicast group concept:
- Each group has its own IP multicast address
- A host can join or leave freely
- Routers forward multicast datagrams (with destination address of the group’s multicast address) to hosts that have “joined” that multicast group
Multicast groups

- class D Internet addresses reserved for multicast:
  - 1110 Multicast Group ID
  - 28 bits
  - i.e. 224.0.0.0 to 239.255.255.255.

- host group semantics:
  - anyone can “join” (receive) or leave multicast group
  - anyone (not even a member) can send to multicast group
  - no network-layer identification of hosts members

- needed: infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group
Mapping IP Multicast Address to Ethernet Address

- Ethernet MAC Addresses: 48 bits
  - broadcast: all 1s, ff:ff:ff:ff:ff:ff
  - multicast: multicast flag (the lowest bit of the 1st octet) = 1
  - 01-00-5E-00-00-00 to 01-00-5E-7F-FF-FF for IP multicast
- IP multicast group address mapped to the lower order 23 bits of MAC address
- not one-to-one mapping, one Ethernet mcast addr ↔ 32 IP mcast addrs
IPv6 Multicast Addresses (RFC 2375)

- low-order flag indicates permanent / transient group; three other flags reserved
- scope field:  
  - 1 - node local  
  - 2 - link-local  
  - 5 - site-local  
  - 8 - organization-local  
  - B - community-local  
  - E - global  
  - (all other values reserved)
Joining a mcast group: two-step process

- **local**: host informs local mcast router of 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 need not explicitly “unjoin” group when leaving
- **router:** sends IGMP query at regular intervals
  - host belonging to a mcast group must reply to query
How IGMP Works

- on each link, one router is elected the “querier”
- querier periodically sends a Membership Query message to the all-systems group (224.0.0.1), with TTL = 1
- on receipt, hosts start random timers (between 0 and 10 seconds) for each multicast group to which they belong
How IGMP Works (cont.)

- when a host’s timer for group G expires, it sends a Membership Report to group G, with TTL = 1
- other members of G hear the report and stop their timers
- routers hear all reports, and time out non-responding groups
Source Specific Multicast

• Source Specific Multicast: a receiving host specifies (source, mcast group) to join
  – receive multicast packets addressed to the group and only if they are from the specific sender (one-to-many)

• Any source multicast (ASM): many-to-many
IGMP

**IGMP version 1**

- **router**: Host Membership Query msg broadcast on LAN to all hosts
- **host**: Host Membership Report msg to indicate group membership
  - randomized delay before responding
  - implicit leave via no reply to Query
- RFC 1112

**IGMP v2**: additions include

- group-specific Query
- Leave Group msg
  - last host replying to Query can send explicit Leave Group msg
  - router performs group-specific query to see if any hosts left in group
  - RFC 2236

**IGMP v3**:

- Join/Leave specific $S$ in $G$
- RFC 3376
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

![Source-based trees](image1.png)  ![Shared tree](image2.png)
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

![Diagram of shortest path tree]

**LEGEND**
- S: source
- router with attached group member
- router with no attached group member
- link used for forwarding, i indicates order link added by algorithm
Reverse Path Forwarding

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

\[
\text{if (mcast datagram received on incoming link on shortest path back to source)}
\]

\[
\text{then flood datagram onto all outgoing links}
\]

\[
\text{else ignore datagram}
\]
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” msgs sent upstream by router with no downstream group members

LEGEND
S: 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 exists
- not used in practice:
  - computational complexity
  - information about entire network needed
  - monolithic: rerun whenever a router needs to join/leave
Shared-Tree: Steiner Tree

Kruskal’s algorithm

Select min weight edge

Expand the tree with next min weight

Expand the tree with next min weight, avoiding loops
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:

LEGEND
- router with attached group member
- router with no attached group member
- path order in which join messages generated
Current Intra-Domain Multicast Routing Protocols

DVMRP — Distance-Vector Multicast Routing Protocol
  flood-and-prune,
  unidirectional per-source trees,
  builds own routing table

MOSPF — Multicast Extensions to Open Shortest-Path First Protocol
  broadcast membership,
  unidirectional per-source trees,
  uses OSPF routing table
Current Intra-Domain Multicast Routing Protocols (cont.)

PIM-DM — Protocol-Independent Multicast, Dense-Mode
   broadcast-and-prune, unidirectional per-source trees,
   uses unicast routing table (Protocol Independent)

PIM-SM — Protocol-Independent Multicast, Sparse-Mode
   uses meeting places ("rendezvous points"),
   unidirectional per-group or shared trees,
   uses unicast routing table (Protocol Independent)

CBT — Core-Based Trees
   uses meeting places ("cores"),
   bidirectional shared trees,
   uses unicast routing table
The First Intra-Domain Routing Protocol: DVMRP
Distance-Vector Multicast Routing Protocol (DVMRP)

DVMRP consists of two major components:

1. a conventional distance-vector routing protocol (like RIP) which builds, in each router, a routing table like this:

<table>
<thead>
<tr>
<th>Subnet (Destination)</th>
<th>shortest dist (cost)</th>
<th>via interface (NextHop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>i1</td>
</tr>
<tr>
<td>b</td>
<td>5</td>
<td>i1</td>
</tr>
<tr>
<td>c</td>
<td>3</td>
<td>i2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

2. a protocol for determining how to forward multicast packets, based on the routing table and routing messages
Example Topology
Phase 1: Truncated Broadcast

flood iff the packet arrives over the link that is on the shortest path to $S$
first packet from source s to multicast group g is forwarded using Reverse Path Forwarding (RPF) algorithm:

if a multicast packet arrives from the interface that, according to the routing table, is on the shortest path back to the source,

then forward the packet on all* other interfaces

else drop the packet

*exceptions:

when more than one router attached to a link, only the router with the shortest distance back to the source forwards onto that link (or, in case of a tie, the router with lowest IP address)

on a “leaf” link (relative to the source) do not forward the packet if there are no group members on that link
Phase 2: Pruning
when a packet reaches a router for whom there are no permitted outgoing interfaces, that router sends a prune message to its predecessor on the path back to the source
if the reception of a prune message causes predecessor now to have no remaining outgoing interfaces, it then sends a prune message to its predecessor
routers keep state remembering what prunes they have sent and received; the state is discarded after a (relatively long) timeout

(Notes for slide above)
Steady State
now, packets flow down only those branches that lead to members of the multicast group when the prune-state times out, if there is still multicast traffic from s to g, truncated broadcast happens again, triggering prunes again; if the traffic has stopped, nothing more happens and no state remains for traffic from s to g
Grafting on New Receivers
if a new group member appears on a pruned-off link (as detected by IGMP), the upstream router for that link sends *graft* messages to undo the effect of any prune messages sent, regarding that group
Steady State after Grafting
after the graft message has undone the pruning, multicast packets can now flow down the branch to the new member

there are, of course, many more details; if you are interested, please read the current spec: draft-ietf-idmr-dvmrp-v3-08.txt (not RFC-1075!), which can be found via the IETF web page, http://www.ietf.org
Multicast Routing: MOSPF
Multicast OSPF (MOSPF)

- an extension to OSPF (Open Shortest-Path First), a link-state, intra-domain routing protocol specified in RFCs 1584 & 1585
- multicast-capable routers indicate that capability with a flag in their link-state messages
- routers include in their link-state messages a list of all groups that have members on the router’s directly-attached links (as learned through IGMP)
Link state: each router floods link-state advertisement
Multicast: add membership information to “link state”

Each router then has a complete map of the topology, including **which links have members of which multicast groups**
Z has network map, including membership at X and Y
Z computes shortest path tree from S1 to X and Y
Z builds multicast entry with one outgoing interface
W, Q, R, each build multicast entries
Link-state advertisement with new topology (may be due to link failure) may require recomputation of tree and forwarding entry. Link WZ failed in the diagram below.
Link state advertisement (T) with **new membership (R3)** may require incremental computation and addition of interface to outgoing interface list (Z). Similarly, disappearance of a membership may cause deletion an interface from an outgoing interface list. Link **WZ is back to normal**.
Multicast Routing: PIM
Protocol Independent Multicast (PIM)

• "Protocol Independent"
  - does not perform its own routing information exchange
  - uses unicast routing table made by any of the existing unicast routing protocols

• PIM-DM (Dense Mode) - similar to DVMRP, but:
  - without the routing information exchange part
  - differs in some minor details

• PIM-SM (Sparse Mode), or just PIM - instead of directly building per-source, shortest-path trees:
  - initially builds a single (unidirectional) tree per group, shared by all senders to that group
  - once data is flowing, the shared tree can be converted to a per-source, shortest-path tree if needed
PIM Protocol Overview

• Basic protocol steps

- routers with local members send **Join messages** towards a **Rendezvous Point (RP)** to join shared tree

- routers with local sources **encapsulate data to RP**

- routers with local members may initiate data-driven **switch to source-specific, shortest-path tree**
Phase 1: Build Shared Tree

- Shared tree after R1, R2, R3 join
- Join message toward RP

Join message toward RP

Join G

R4
Phase 2: Sources Send to RP

unicast encapsulated data packet to RP

RP decapsulates, forwards down Shared tree
Phase 3: Stop Encapsulation

Join G for S1

Join G for S2

(S1,G)

(S2,G)

(*.G)

Join G for S1

R1

R2

R3

R4
Phase 4: Switch to Shortest Path Tree

shared tree

Join messages toward S2

S1

S2

RP

R1

R2

R3

R4
Phase 5: Prune (S2 off) Shared Tree

S2 distribution tree

Shared tree

Prune S2 off Shared tree where iif of S2 and RP entries differ
Clean Slate Protocol Design Project
“k-out-of-N Multicast”

Initial Discussion
(detailed project instructions to be given later)
Objective

- Design and implement a k-out-of-n packet datagram multicast network
- A message is sent from one sender to 1-3 recipients
  - Each packet will have n= 1, 2, or 3 destinations
  - Multicast message to “best k (=<n)” destinations
  - Routing protocol chooses the ‘best’ multicast path
- **Project Format:** Like an industry standards committee
  - Each group proposes a solution to the class
  - Class as a whole combines proposals for final draft
  - All groups implements code based on this draft
Network Architecture & Assumptions

- Small network with <256 hosts and routers
- Max number of destinations is 3 (n)
- Packet to be delivered to k out of n destinations
- No direct links between end-nodes
- Loss probability on each link = $p$

Dest = a, b, c

Data
Routing Algorithm

- Define an algorithm that delivers $k=1,2$, $N=3$ packets efficiently
- Goal is to find a mcast tree with least number of packet-hops
- In this example, red mcast tree vs. blue
- Variations of Steiner tree? Simpler heuristics (longest common path?)

Dest = d1, d2, d3

Data

Red # pkt-hops = 8
Blue # pkt-hops = 6
Proposal Requirements

• Proposal consists of a protocol and related algorithms
• Protocol includes:
  – **How to address different network elements?**
    • Fixed static address vs. Dynamic address allocation
    • Bit structure of address and how it is used
  – **How do nodes/routers discover each other?**
    • Periodic hello messages vs. hello with ACK
    • Provisioning or not for router/node failure?
  – **What is the baseline routing protocol?**
    • Link State vs. Distance Vector vs. Other variations such as on-demand routing or controlled flooding
  – **What ARQ scheme to use?**
Proposal Requirements

• Algorithmic component includes:
  – **How to multicast once you know the destination nodes?**
    • Whether to follow one of known multicast schemes?
    • Better, more efficient, simpler schemes?
    • Determining where to split and copy the packet?
    • What requirements from routing protocol?
    • Multicast efficiency measured in packet-hops

• Use ppt template for group proposals to be posted on project website
Today’s Homework

- Peterson & Davie,
- 4th Ed - 4.58, 4.60, 4.63
- 5th Ed – 5.15, 5.17, 5.20

Due on Friday 3/6; Reminder: **Midterm 3/13**