Multicast

- Operates over UDP
- Key ideas:
  - single packet sent to multiple sources via a single address
    - multicast address
  - multicast group = recipients on multicast address
    - one address per group
  - source is *not aware* of the recipients!
  - recipients may join or leave at any time

MBone

- Multicast backbone network
- Multicast connectivity between academic, corporate, and research sites
- “Virtual” multicast
  - multicast packets tunnelled between sites
  - each site had multicast enabled locally
- First large-scale multicast network

IGMP

- Internet Group Management Protocol
- Protocol used for multicast group management
  - communication between a host and its local router
    - join or leave multicast groups
    - periodically confirm group membership
- RFC 988
Multicast routing

- DVMRP (Distance vector multicast routing protocol)
- MOSPF (link-state)
- PIM (Protocol independent multicast)

DVMRP

- Default routing mechanism originally
- Idea: distance vector routing applied to multicast
- Procedure: “broadcast and prune”
  - each LAN has designated multicast router
  - router receives multicast packet: forwards it on all links except the incoming one
  - “prune” packets sent by routers that do not want to receive messages from that group

MOSPF

- Very similar to OSPF
- Routers store extra state about multicast groups
- Shortest-path multicast tree constructed

PIM

- Idea: multicast routing that’s independent of the underlying routing protocol
  - use smarter strategies than “broadcast and prune”
- Types:
  - PIM-DM: dense mode
  - PIM-SM: sparse mode
### PIM-SM
- Routers explicitly join and leave multicast groups
- 2 message types: join and prune
- Each group has a **rendezvous point**
  - handles group membership
- Can construct 2 types of multicast tree:
  - **shared**:
    - used by all senders in the group
  - **source-specific**:
    - used by only one sender

### PIM-SM: shared source tree
- Router sends join to RP via unicast
- Intermediate routers note join, update forwarding tables
- RP is the root of the tree
- Sources that are not part of the tree must tunnel multicast packets to the RP

### PIM-SM: source-specific tree
- If a source is not part of the tree and is sending a large number of packets, the RP may decide to “invite” the source to join the tree
  - RP sends join to source
  - routers learn path as in shared-source tree
  - slightly different forwarding table entry

### Why hasn't multicast been widely deployed?
- Requires intelligence at the routers
  - state information
  - violates “end-to-end” principle
- $$$
  - harder to charge for multicast traffic
- Scalability problems
- No perceived need
  - unicast works “well enough”
  - everything tunneled over HTTP these days
Application-layer multicast

- Current workaround to provide multicast-like service
- CDNs, etc.
- Idea: set up an *overlay network*
  - set of “distribution points”
  - content distributed to distribution points via unicast
  - these points distribute the content locally (ideally, by multicasting it) or to other distribution points in a “tree” (by unicast)

ICMP

- Internet Control Message Protocol
- Layer 3 (“peer” of IP)
- ICMP message sent in IP packets
- Used to communicate error messages
  - “X is unreachable”
  - “time expired”
  - “are you alive”

ICMP message contents

- Type field: 1 byte
- Code field: 1 byte
  - indicates exact error
- Checksum: 2 bytes
  - entire message
- Data: variable size
  - typically contains IP header + first 8 bytes of datagram

Limitations of ICMP messages

- ICMP handling is low(er) priority
- ICMP message is never sent in response to ICMP message
- ICMP messages are not sent in response to broadcast or multicast packets
ICMP example: Ping

- ICMP echo request/echo reply messages
- Ping message:
  - Type = 0 (request) or 8 (reply)
  - Code = 0
  - identifier: 2 bytes
  - sequence number: 2 bytes
  - (data)

Ping: calculation of RTT

- Source stores original timestamp in data field of ICMP packet
- Receiver: does not change the data field
- Source receives reply, subtracts current time from timestamp

ICMP example: traceroute

- Source sends UDP datagrams with TTL = \(x + 1\), where \(x\) = previous TTL
  - addressed to destination address
- Intermediate hop: receives UDP datagram with TTL = 1
  - sends ICMP “Time exceeded” message
- Destination: UDP datagram will be addressed to nonexistent port
  - sends ICMP “Port unreachable” message

ICMP time exceeded message

- Type = 11
- Code = 0
- Data = IP header + first 8 bytes of datagram
ICMP port unreachable message

- Type = 3
- Code = 3
- Data: IP header + first 8 bytes of datagram