19. Multicast Communication

- Multicast within a subnet
- Reverse path forwarding
- IGMP
- PIM
- Scalable multicast forwarding

Jon Turner
Multicast in a Subnet

- Within switched subnet, multicasting relies on Ethernet
  - IP multicast address embedded in Ethernet multicast address
    - more precisely, low 23 bits are embedded
  - switches deliver multicast packets to all hosts (simple case)
- Receiving application configures multicast socket (UDP)
  - bind to specific port, join multicast group
- Sending app sends UDP packet on multicast socket
  - use multicast address/port, set TTL to 1
Basic Multicast Receiver

/** usage: Rcvr mcAdr port iface */
import java.io.*; import java.net.*;
public class Rcvr {
    public static void main(String args[]) throws Exception {
        // form multicast group from command-line arguments
        int port = Integer.parseInt(args[1]);
        InetAddress group = new InetAddress(args[0], port);
        // open multicast socket and join group
        MulticastSocket sock = new MulticastSocket(port);
        NetworkInterface iface = NetworkInterface.getByName(args[2]);
        sock.joinGroup(group, iface);
        // create buffer and packet
        byte[] inBuf = new byte[1000];
        DatagramPacket inPkt = new DatagramPacket(inBuf, inBuf.length);
        while (true) {
            // receive packet and print payload
            sock.receive(inPkt);
            System.out.println(new String(inBuf, 0, inPkt.getLength(), "US-ASCII");
        }
    }
}
Basic Multicast Sender

/** usage: Sender mcAdr port iface ttl message */
import java.io.*; import java.net.*;
public class Sender {
    public static void main(String args[]) throws Exception {
        // from multicast group address from command-line arguments
        int port = Integer.parseInt(args[1]);
        InetSocketAddress group = new InetSocketAddress(
            InetAddress.getByName(args[0]), port);

        // open multicast socket, set interface and time to live
        MulticastSocket sock = new MulticastSocket();
        sock.setNetworkInterface((NetworkInterface.getByName(args[2])));
        sock.setTimeToLive(Integer.parseInt(args[3]));

        // create buffer and packet to send
        byte[] outBuf = args[4].getBytes("US-ASCII");
        DatagramPacket outPkt = new DatagramPacket(
            outBuf, outBuf.length, group);
        sock.send(outPkt); // send packet to group
        sock.close();
    }
}
Internet Group Multicast (RFC 2236)

- To forward among subnets, router needs to know about multicast subscribers
  - router multicasts *membership query* packets
  - hosts respond with *membership reports* after random delay
    - suppress duplicate reports; usually just one actual report per group

- Router forwards multicast packets among networks based on membership information
  - subscriptions timeout if not renewed periodically
IGMP Details

- **Message fields**
  - type – query, report, leave group
  - max response time – max wait time before responding to query
  - group address – subject multicast addr; 0 for “general query”

- **On each subnet, one router plays role of “Querier”**
  - router with smallest IP address acts as querier
  - issues periodic general membership queries
    - sent on all systems multicast address: 224.0.0.1

- **Hosts send reports to subject multicast address**
  - in response to general queries or when first joining a group

- **Leave group message sent by host when leaving**
  - sent to all routers group (224.0.0.2)
  - querier responds with a group-specific query
    - verify that no other group members still out there
IGMP Snooping (RFC 4541)

- **Objective**
  - use group membership to limit propagation of multicast packets
  - learn which hosts are in which group by snooping on IGMP
  - invented by switch vendors to limit multicast traffic in large switched networks

- **Suppression of duplicate membership reports** by IGMP interferes with snooping
  - so send report packets only to routers, not to hosts
    - implies switches must be able to detect which ports go to routers

- **Forwarding based on group membership**
  - forwarding can use either Ethernet or IP addresses
    - IP addresses preferred due to non-exact multicast address mapping
    - but not an option for simple Ethernet switches
Multicast Issues in Larger Networks

- IGMP is not always enough
  - suppose a multicast has members in subnets $A$ and $C$, but not $B$
    - routers will not forward packets across $B$
  - can work if group members are in contiguous subnets

- Multiple paths among subnets can produce multicast duplicates
  - need to restrict routing of each multicast group to a tree
  - requires routers to maintain forwarding information for each multicast group
Efficient distribution of multicast packets requires that packets be forwarded along a tree

- in principle, any tree joining subnets can be used and each multicast can use a different tree
- internet multicast generally uses reverse of unicast routes
  - so routing based on combination of (source addr, multicast addr)
  - means multicast with many senders requires more routing state
  - more efficient to use shared tree through “rendezvous point”
Multicast Using Shared Tree (PIM)

- Rendevous point (RP) serves as multicast tree root
  - different groups have different RPs, routers can determine assigned RP for each multicast address
- When Designated Router (DR) discovers new subscriber
  - sends join message to RP, creating multicast state along path
- Sending hosts just send packets to multicast address
  - initially, DR may forward to RP inside unicast packet
Sending to Multicast Group

- Non-subscribing sender’s packets sent to RP by DR
  - RP extracts multicast packet and forwards on shared tree
- RP may initiate join back towards source
  - adds multicast routing state along path, merges with RP tree
  - when native packets reach RP, it tells DR to stop encapsulating
Switch to More Direct Path

- Router in tree may choose to switch to more direct path
  - sends join back towards source
  - after packets arrive on direct path, prune links in original path from source
- Note: expensive for multicasts with many senders
More About PIM (RFC 4601)

- PIM stands for Protocol Independent Multicast
  - PIM can use routing data from any routing protocol
    - e.g. unicast routing state, or multicast BGP
- Routers must be able to determine RP for any address
  - can use statically configured mapping based on address ranges
  - or, a PIM domain can use a Bootstrap Router (BSR) to distribute information used to map a multicast address to an RP
    - domain has a defined set of RPs distributed by BSR
    - PIM routers use a hash function to select an RP in this set
      - e.g. if $N$ candidate RPs, hash(multicast address) mod $N$ selects one
- PIM generally used only within single routing domain
  - RFC omits essential details for inter-domain routing
    - e.g. how are RPs coordinated among different domains
  - BGP extensions may eventually enable inter-domain multicast
IGMPv3 (RFC 3376)

- Adds support for source-specific multicast using “filters”
  - include-filter specifies a list of acceptable senders
    - multicast packets are delivered to socket only if sender is on list
  - exclude-filter specifies a list of blocked senders
  - hosts have “interface filters” derived from socket filters
  - not supported by java multicast sockets; linux does support it
- Queries and membership reports are extended to handle “filtered multicasts”
  - new source-specific query that includes a sender list for target destination address
  - reports can cover multiple groups and includes a filter per group
- IGMPv3 ignores membership reports from other hosts
  - so all IGMPv3 hosts in a multicast group will respond to a query
- Can operate in “compatibility mode” with older versions
Which Trees are Best?

- Reverse path forwarding routes multicast packets on shortest paths
  - tends to copy packets near the source
  - copying near destination reduces # of "long-distance packets"

- Best multicast route is a Steiner tree
  - Steiner tree for a vertex subset $S$ is shortest subtree that includes all vertices in $S$
  - problem is NP-complete, so no efficient algorithm known for finding optimal solution
  - good approximations based on minimum spanning tree
Other Issues with IP Multicast

- Separate protocol needed to reserve network capacity
  - RSVP protocol (RFC 2205) can be used for this purpose
  - but cannot select route based on required network capacity
    - QoS routing has been discussed but never implemented

- No “private” multicast groups
  - no way to get short-term exclusive use of a multicast address
  - no way to limit subscribers or limit who can send to an address
    - makes it awkward to use for multi-party teleconferencing
    - privacy requires encryption

- Largely unavailable in public internet
  - unresolved technical issues for inter-domain routing
  - no economic benefit to network providers
    - multicast senders benefit as they need not duplicate traffic
    - but ISPs lose money if senders reduce internet access link rate