Multicast

Software Multicast

- Support applications that need to send messages to more than one recipient
- Make use of hardware multicast features (e.g., ethernet) when available
Multicast by serial transmission

- The application could just serially send a copy of the message to each recipient. But:
  - Large amount of overhead time consumed at the app
  - Delay for the “last recipient” can be unacceptable
  - App has to keep track of who’s in the multicast group (could be large and it may change dynamically)
- The network ought to be able to take care of it

IP multicast

- The idea:
  - Recipients form a “multicast group”
  - Sender sends a packet to a “multicast address” to have it copied to everyone in the multicast group (IPv4 Class “D”)
  - Recipients join or leave the multicast group at will
IGMP

• Internet Group Management Protocol
  – Protocol used to join or leave a multicast group

• Host desiring to join uses IGMP to notify local router with address of multicast group
  – How do hosts learn multicast group addresses? Usually out of band.

Multicast routing

• Routing needs to be tailored
  – Setting up individual routes to each member of the group may not be optimum overall
Example: Multicast OSPF

- Easy to extend Dijkstra’s algorithm to find minimum spanning trees (“multicast trees”)
- Special “Group Membership Packet” added to link state advertisement packet types
- Internet thought of as “islands” of multicast-enabled devices – multicast traffic tunnelled between them

The papers

- How was this all worked out?
  - Testbed networks and initial implementations (MBONE…) {Adam’s paper}
- How will it work under IPv6?
  - Supported by provisions built into the protocol {Nic’s paper}