Data Communication & Networks
G22.2262-001

Session 10 - Main Theme
IP Multicast

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Agenda

- Introduction to Multicast
- Multicast Addresses
- IP Multicast
- Reliable Multicast
- Pragmatic General Multicast (PGM)
- Reliable Multicast Protocol (RMP)
- Conclusion

Part I

Introduction to Multicast

Cast Definitions

- Unicast - send to one destination (198.122.15.20)
- General Broadcast - send to EVERY local node (255.255.255.255)
- Directed Broadcast - send to subset of nodes on LAN (198.122.15.255)
- Multicast - send to every member of a Group of “interested” nodes (Class D address).
- RFC 1112 (an easy read!)

Why Multicast, Why Not Unicast?

- Unicast:
  - Many applications require same message sent to many nodes (10, 100, 1000, n)
  - Same message transits network n times.
  - n messages requires n*(CPU time) as 1 message
  - Need to deliver “timely” information.
    - Message arrives at node n >> node 1

Why Multicast, Why Not Broadcast?

- Broadcast:
  - Send a copy to every machine on the net
  - Simple, but inefficient
  - All nodes “must” process the packet even if they don’t care
  - Wastes *more* CPU cycles of slower machines (“*broadcast radiation*”)
  - General broadcast cannot be routed
  - Directed broadcast is limited in scope (to machines on same sub-net or same domain)

Multicast Applications

- News/sports/stock/weather updates
- Software distribution
- Video-conferencing, shared whiteboards
- Distributed interactive gaming or simulations
- Email distribution lists
- Database replication

IP Multicast - Concepts

- Message sent to multicast “group” of receivers
  - Senders need not be group members
  - Each group has a “group address”
  - Groups can have any size
  - End-stations (receivers) can join/leave at will
  - Data Packets are UDP (uh oh!)

IP Multicast Benefits

- Distribution tree for delivery/distribution of packets (i.e., scope extends beyond LAN)
  - Tree is built by multicast routing protocols. Current multicast tree over the internet is called MBONE
- No more than one copy of packet appears on any sub-net
- Packets delivered only to “interested” receivers => multicast delivery tree changes dynamically
- Non-member nodes even on a single sub-net do not receive packets (unlike sub-net-specific broadcast)

Part II

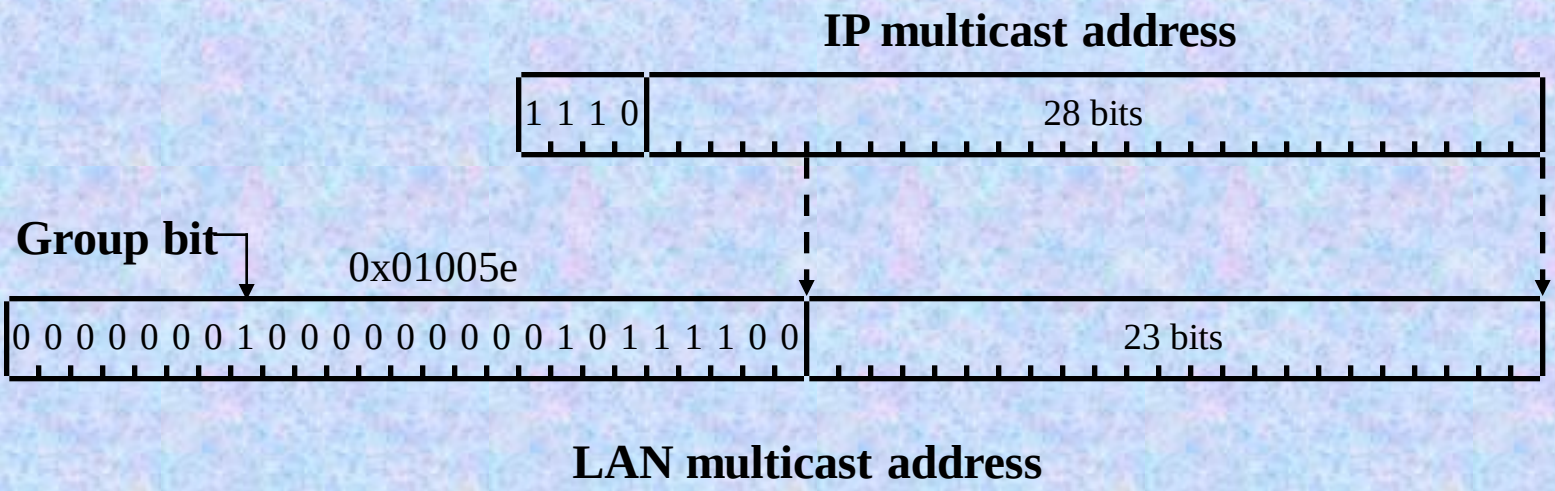
Multicast Addresses

Multicast Addresses

- Class D addresses: 224.0.0.0 - 239.255.255.255
- Each multicast address represents a *group of arbitrary size, called a "host group"*
- Addresses 224.0.0.x and 224.0.1.x are reserved.
  - See assigned numbers RFC 1700
  - Eg: 224.0.0.2 = all routers on this sub-net
- Addresses 239.0.0.0 thru 239.255.255.255 are reserved for private network (or intranet) use

Link-Layer Multicast Addresses

Ethernet and other LANs using 802 addresses:



Lower 23 bits of Class D address are inserted into the lower 23 bits of MAC address (see RFC 1112)

Part III

IP Multicast

NIC, IP Stack, Apps Cooperate!

- **Idea** - NIC does not accept packets unless some app on this node wants it (avoid non-productive work).
- How does it know which packets to accept?
  - App “joins” group with Class D address
  - IP stack gives class D info to NIC
  - NIC builds “filter” to match MAC addresses
  - Latch on to:
    - Own address, MAC broadcast, or “Group” address

IP Multicast - Sending

- Use normal IP-Send operation, with multicast address specified as destination
- Must provide sending application a way to:
  - Specify outgoing network interface, if >1 available
  - Specify IP time-to-live (TTL) on outgoing packet
  - Enable/disable loop-back if the sending host is/isn't a member of the destination group on the outgoing interface

IP Multicast - Receiving

- Two new operations
  - Join-IP-Multicast-Group(group-address, interface)
  - Leave-IP-Multicast-Group(group-address, interface)
- Receive multicast packets for joined groups via normal IP-Receive operation

Multicast Scope

- Scope: How far do transmissions propagate?
- Implicit scope:
  - Reserved Mcast addresses => don't leave subnet.
- TTL-based scope:
  - Each multicast router has a configured TTL threshold
  - It does not forward multicast datagram if TTL <= TTL-threshold
  - Useful at edges of a large intranet as a blanket parameter

IGMP

- Router sends *Host Membership Query* to 224.0.0.1 (all multicast hosts on sub-net)
- Host responds with *Host Membership report* for each group to which it belongs, *sent to group address (*224.0.0.2)
- Router periodically broadcasts query to detect if groups have gone away
- Hosts send reports when join or leave group

IP Multicast in Java

- Java has a Multicast Socket Class
- Use it to “join” a multicast group.

```
MulticastSocket s = null;
InetAddress group = null;

try  {
  group = InetAddress.getByName(“227.1.2.3”);
  s = new MulticastSocket(5555);
  s.joinGroup(group);
} catch (UnknownHostException e)  {

} catch (IOException e)  {

}
```

IP Multicast in Java

- Receive DatagramPackets on a MulticastSocket

```
DatagramPacket recv = new DatagramPacket(buf,
buf.length);
try  {
    s.receive(recv);
} catch (IOException e)  {
    System.out.println("mcastReceive: " +
e.toString());
    return;
}
// get message
String msg = new String(recv.getData(),
recv.getOffset(), recv.getLength());
```

IP Multicast in Java (cont.)

- To send, just send a DatagramPacket to the multicast address, port (no need to use a MulticastSocket, although you could)

```
group = InetAddress.getByName("227.1.2.3");
s = new DatagramSocket();

DatagramPacket snd = new DatagramPacket(buf, buf.length,
                                        group, 5555);
try  {
  s.send(snd);
} catch (IOException e)  {
  System.out.println("mcastSend: " + e.toString());
  return;
}
```

Let's See Multicast in Action

Part IV

Reliable Multicast

Reliable Multicast

- Remember: IP Multicast uses IP - it's unreliable!
- We need a "reliable" multicast
- Let's review what we mean by "reliable"
  - message gets to ALL receivers
  - sending order is preserved
  - no duplicates
  - no corruption

Reliable Multicast (cont.)

- **Problems:**
  - Retransmission can make reliable multicast as inefficient as replicated unicast
    - Ack-implosion if all destinations ack at once
    - Nak-implosion if all destinations nack at once
  - Source does not know # of destinations. Or even if all destinations are still “up”
  - One bad link affects entire group
  - Heterogeneity: receivers, links, group sizes

Reliable Multicast (cont.)

- RM protocols have to deal with:
  - Scalability
  - Heterogeneity
  - Adaptive flow/congestion control
  - Reliability
- Not all multicast applications need reliability of the type provided by TCP. Some can tolerate reordering, delay, etc
  - Let's look at two very different approaches
    - PGM (Pragmatic General Multicast)
    - RMP (Reliable Multicast Protocol)

Part V

Pragmatic General Multicast (PGM)

Pragmatic General Multicast

- Cisco's reliable multicast protocol
- Central design goal
  - **Simplicity** of operation yet provide for ***scalability and network efficiency***
- Not the perfect solution for all requirements, but for most, ***P****retty* ***G****ood*!

Overview of PGM

- **Component**s
  - *Source* : transport-layer originators of PGM data packets
  - *Receiver* : transport-layer consumers of PGM data packets
  - *Network element* : network-layer entities in the intervening network – typically routers, but need not be
- **Receiver initiated repair protocol**
  - Based on NAK

PGM Packet Types

- **ODATA**
  - Data packets from a source
- **SPM** (Source Path Message)
  - Sent by sources: used to establish reverse path from receivers to sources
- **NAK**
  - Sent by receivers to ask for repairs.
  - Forwarded upstream along source path
- **NCF** (NAK Confirm)
  - Sent by network elements to NAKers
- **RDATA**
  - Data packet resent from a source in reply to a NAK

PGM Transmit Window

- **Problem**: NO Acks, so:
  - Senders don't know when all receivers have packets
  - Therefore, how long does sender retain packet?
- **Solution** (sort of)
  - Define a window (range of sequence #s)
  - Keep only packets in window
  - Update window edges periodically to receivers
  - Receivers can only ask for resend (NAK) of packets in the window

Source Functions

- **Data Transmission**
  - Multicast ODATA packets within transmit window a given transmit rate
- **Source Path State**
  - Multicast SPMs to establish source path state in PGM network elements
- **NAK Reliability**
  - Sources multicast NCFs in response to any NAKs they receive
- **Repairs**
  - Multicast RDATA packets in response to NAKs received for data packets within the transmit window

Source Functions (cont.)

- **Transmit Window Advance**
  - Sources MAY advance the trailing edge of the window according to one of a number of strategies, for example:
    - keeping the window at a fixed size in bytes
    - keeping the window at a fixed number of packets
    - keeping the window at a fixed real time duration.
- **Trailing edge is advanced** in ODATA and SPM packets.
- Question: *What defines the leading edge?*

Receiver Functions

- **Source Path State**
  - Use SPMs to determine the last-hop PGM network element for a given TSI to which to direct their NAKs
- **Data Reception**
  - Receive ODATA within transmit window and eliminate any duplicates (and order packets!)
- **Repair Requests**
  - Receivers unicast NAKs to last-hop PGM network element.
  - Receiver MUST repeatedly transmit a given NAK until it receives a matching NCF

Receiver Functions (cont.)

- **NAK Suppression**
  - Receivers suppress NAKs for which a matching NCF or NAK is received during the NAK transmit back-off interval
- **Receive Window Advance**
  - Receivers immediately advance their receive windows upon receipt of any PGM data packet or SPM within the transmit window that advances the receive window

Network Element Functions

- **Source Path State**
  - Network elements intercept SPMs and use them to establish source path state for the corresponding TSI before multicast forwarding them
- **NAK Reliability**
  - Network elements multicast NCFs to the group in response to any NAK they receive
  - For each NAK received, create repair state recording the TSI, the sequence number of the NAK, and the input interface on which the NAK was received

Network Element Functions (cont.)

- **Constrained NAK Forwarding**
  - Network elements repeatedly unicast forward only the first copy of any NAK they receive to the upstream PGM network element on the distribution path for the TSI until they receive an NCF in response
- **NAK Elimination**
  - Network elements discard exact duplicates of any NAK for which they already have repair state and respond with a matching NCF
- **Constrained RDATA Forwarding**
  - Network elements use NAKs to maintain repair state consisting of a list of interfaces upon which a given NAK was received, and they forward the corresponding RDATA only on these interfaces

Operation – No Errors

- Sources multicast ODATA packets
  - Packet contains source ID, seq #, Window trailing edge, data
  - Network Elements forward packets to receivers (could be applications nodes or other NEs)
- Sources periodically multicast SPM packets (also contain source ID, seq #, Window edges)
- Network Elements use SPMs to build a reverse path to the source (keyed by Source ID, called TSI in PGM)
  - NE must forward SPM to downstream NE!

Receiver Operation: Packet Loss

- Receiver detects missing packet possibly from:
  - ODATA sequence number
  - SPM sequence number
  - Receipt of NCF
- Start a random (bounded) timer
  - If missing packet arrives (ODATA or RDATA) cancel (success)
  - If another NCF arrives (same seq. #), restart timer
  - If timer expires, send NAK, restart timer
  - If too many retries, cancel (fail)

Network Element Operation: Packet Loss

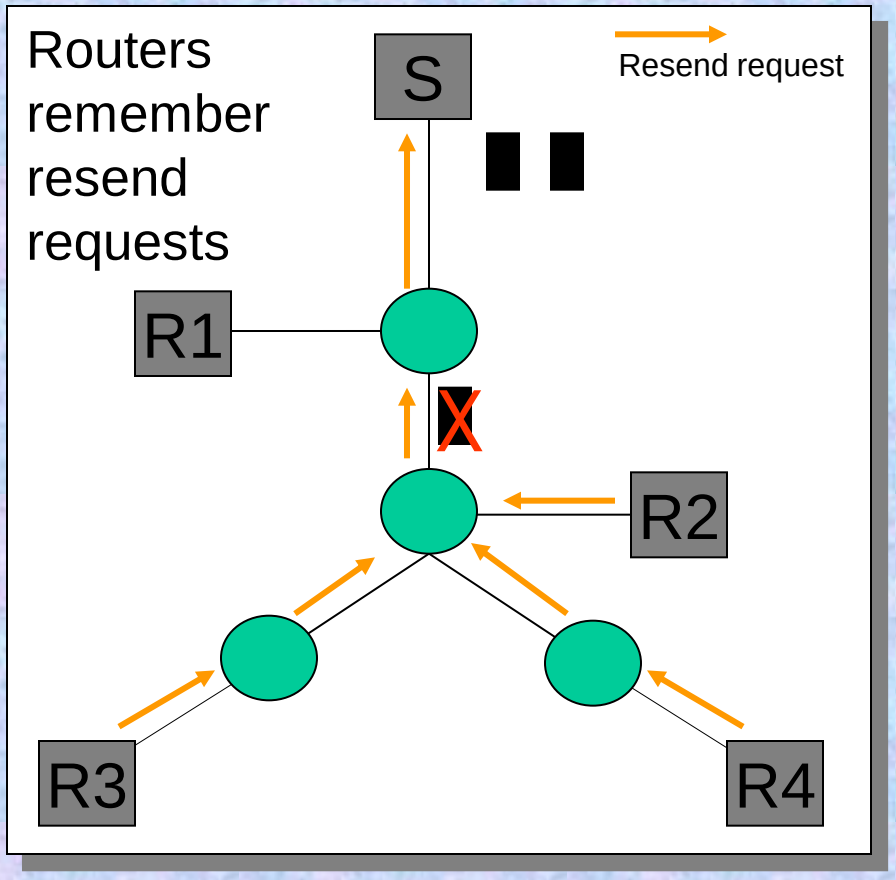
- Receiver unicasts NAK to NE
- NE builds a “repair state” for this request ONLY if NAK is from a local group member
- NE multicasts NCF to group (limit scope using TTL) – now all receivers in local group know
- NE unicasts NAK to next upstream PGM hop for this source (using TSI from NAK packet and SPM info)
- This process is repeated at each successive upstream NE

Source Operation: Packet Loss

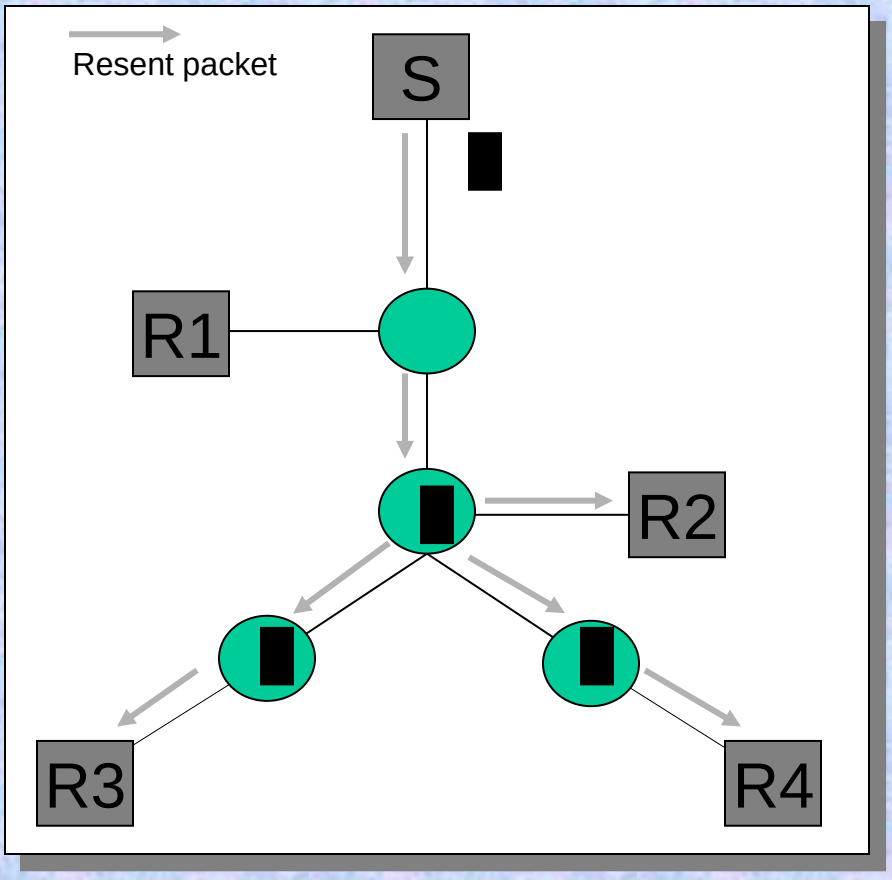
- Source of requested packet gets NAK
- Source multicasts NCF
- Source multicasts RDATA packet (but only if the requested packet is still in the transmit window)

Pragmatic General Multicast

Packet 1 reaches only R1;
R2, R3, R4 request resends



Packet 1 resent to R2, R3, R4;
Not resent to R1



Part VI

Reliable Multicast Protocol (RMP)

RMP

- Messages from multiple senders to multiple destinations
- Quality of Service (QoS) Selection on a per message basis
- Supports total ordering of all messages
- Transparent failure recovery (Reformation)
- Uses Post Ordering Rotating Token algorithm, extension of work by Chang & Maxemchuk

RMP: Quality of Service

- **Unreliable** - same QOS as UDP
- **Reliable** - delivery guaranteed, but not order or 1-copy
- **Source Ordered** - all messages from same source ordered, 1-copy
- **Totally Ordered** - all receivers see exactly same sequence from all sources

About Total Ordering

- Hosts A, B, & C send messages into the group (A1, A2, B1, B2, C1, C2)
- Suppose:
  - Host D gets A1, B1, B2, C1, A2, C2
  - Host E gets C1, A1, A2, B1, C2, B2
  - Host F gets B1, A1, C1, A2, B2, C2
- What if results at Hosts D, E, F depend on order of all messages? In this case, results will be different!
- Total ordering means D, E, F have to see same sequence!

RMP - Concepts

- Each process (an App) has an ID
- Processes form groups (based on mcast Addr, Port, TTL)
- All processes see “membership list”
- Members agree among themselves what the “order” of messages should be (when total order QOS is selected)

RMP – Token Passing

- One member holds a “token”
- The token holder assigns “global” sequence number to each correct packet
- Every packet has unique id (GroupID:Pid:seqno)
- Token Holder “passes” token to next member in list (token pass is mcast!)
- Token packet contains global sequence numbers and packet IDs

RMP – Ack, Nak

- Since token pass is mcast, all members see it (usually)
- The token is the Ack packet
- Since Ack packet contains ids of packets, receiving process sees what it might have missed!
- If missing packets, send Nak packet to request packets (Nak is mcast!)

RMP – Avoiding Nak Implosion

- Before sending Nak, start random timer Tr
- If see missing packets, kill Tr, accept packets
- If see Nak for same packets before Tr, kill Tr, start new timer Tn
- If Tn or Tr, send Nak

RMP – Protocol Model

(1) B is TS

(2) A sends (1)

(3) B assigns TS 1 to (A,1), passes token to C.

(4) C* has nothing, so passes token (NULL Ack) to A with TS = 2

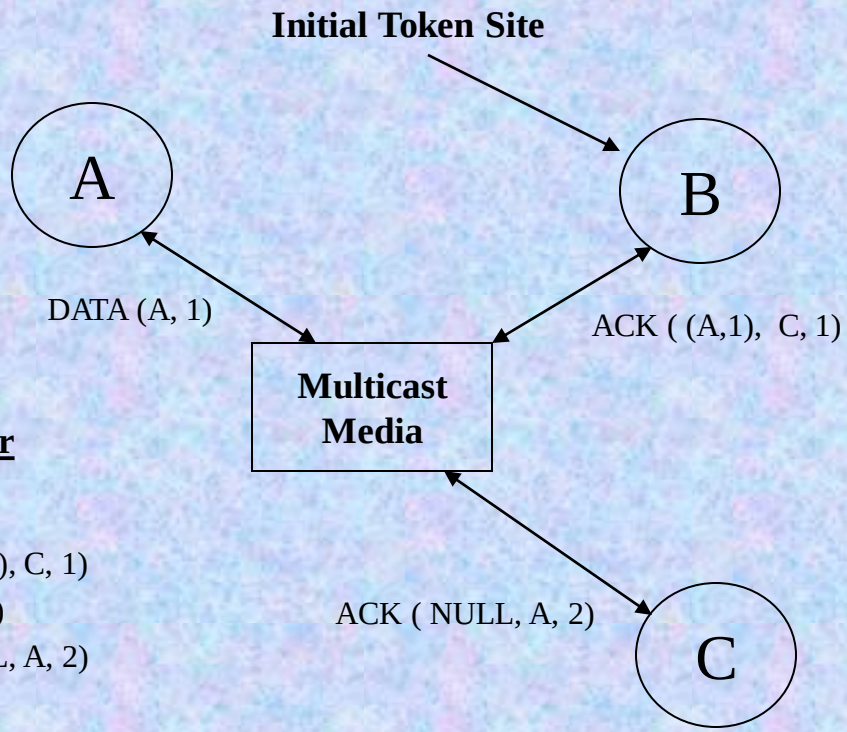
(5) Packet (A,1) can be committed after token gets back to B

Event Order

DATA (A, 1)

ACK ((A, 1), C, 1)

ACK (NULL, A, 2)



Imposed Order

Timestamp	Event
1	ACK ((A, 1), C, 1)
	DATA (A, 1)
2	ACK (NULL, A, 2)

* Cannot accept token token unless have all packets being assigned global timestamps

Packet Reception vs. Delivery

- When providing total ordering:
  - Cannot deliver packets to app when received.
  - Must wait until:
    - timestamp is assigned
    - packet is stable at all nodes (Token is passed n times in group of size n)
- When providing source ordering, can deliver packet if have all previous packets from source
- Question: When can a node delete a packet?

Packet Stability

- **Rule:** you can delete a packet if no other group member will need it
  - When is that?
    - When the site that that assigned a global time stamp to the packet becomes the token site again!
    - How to know this? Answer: Ordering Queue
      - list of all packets in order of time stamp
      - when list has n + 1 Ack packets, delete oldest Ack and lower numbered packets

RMP - Observations

- **Good:**
  - Use to build distributed Applications where *virtual synchrony* is required
  - Fault Tolerant - group reforms after network partition
- **Not So Good:**
  - Does not scale well (maybe to 100 nodes)
    - Question: Why do you think this is so?
  - More complex protocol than PGM
  - Implementations are no longer available (except mine), but specs with FSMs are available

Part VII

Conclusion

Assignment & Readings

- Readings
  - IP Multicast Backgrounder (PDF File)
  - Chapter 4, section 4.7 - multicast routing
  - RFC 1022 (IP Multicast Extensions) (optional)
  - RFC 3208 (PGM) (optional)

Next Session:
Performance in Queuing Systems