XMSF Overlay Multicast Status Report

Mark Pullen and Dennis Moen
George Mason University
{dmoen,mpullen}@gmu.edu

Network Service Requirements for Real Time Distributed Virtual Simulation

-XMSF Report, 2002

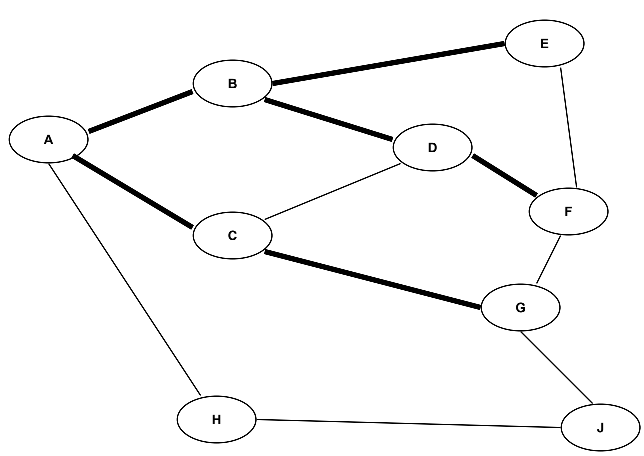
- Network Quality of Service (QoS)
  - end-to-end capacity, latency, jitter, and packet loss in a statistical sense
- Multicast
  - many-to-many group communication
- Reliable Multicast Transport
  - high confidence of delivery
- End-to-end network status and performance monitoring
  - need to know what the network is doing for you
- Multi-sensor systems
  - must manage streaming data with low latency

What is multicasting?

- Multicasting: sending a packet to a group of addresses in a network
  - broadcast: sending to all addresses - impractical in WAN
- IPmc allows packets to be sent to “group” addresses
  - delivered in parallel
  - “class D” address 224.0.0.0 and greater
  - LANs have inherent broadcast/multicast capability
- In WAN, set of all nodes in a multicast group forms a tree
  - grows from the node that starts the group
- Routers at forks in tree must replicate packets
  - commercial routers available that perform this function
- Multicast requires a different routing protocol from unicast
  - routing creates the multicast tree

Network with Multicast Tree

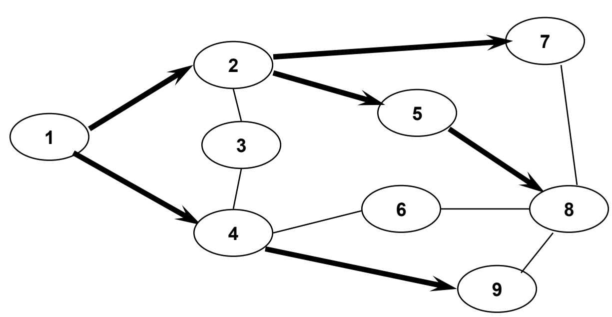
the network duplicates the packets



IP Multicast Directionality

- IPmc uses a many-to-many model of packet delivery
  - equivalent to full duplex for multicast
  - any participating host can send to the group
  - useful for collaborative-style applications
  - a very appealing model but increases complexity of the network layer considerably
- IETF recently developed an approach called Source Specific Multicast (SSM) that uses a one-to-many model
  - equivalent to half duplex for multicast
  - only one host can send to the group
  - useful for presentation-style applications

One-to-Many Multicast

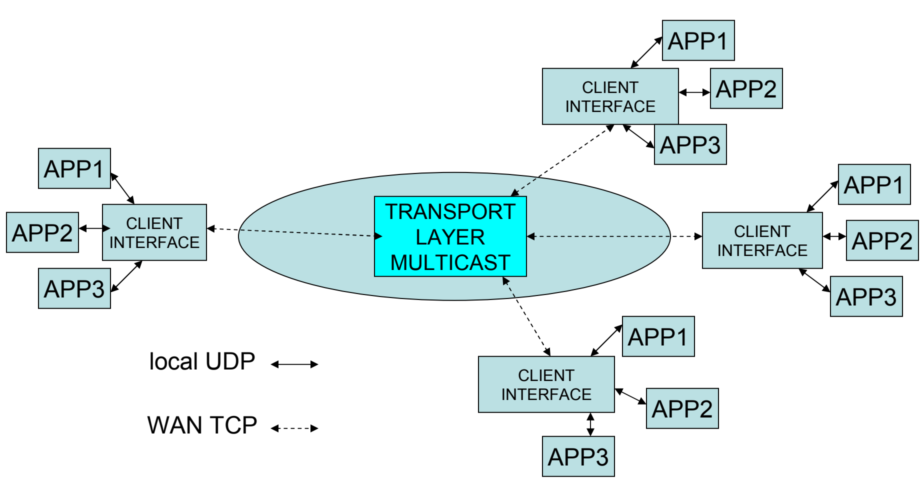


Internet Multicast Services Today

- IP multicast over the Internet not widely deployed
- IETF focus is on one-to-many multicast
- Commercial viability lacking for IP multicast across the Internet
- Result: interest in multicast based in end systems not network
  - End to end argument: push complexity up the stack
  - Example: TCP is complex, IP is simple

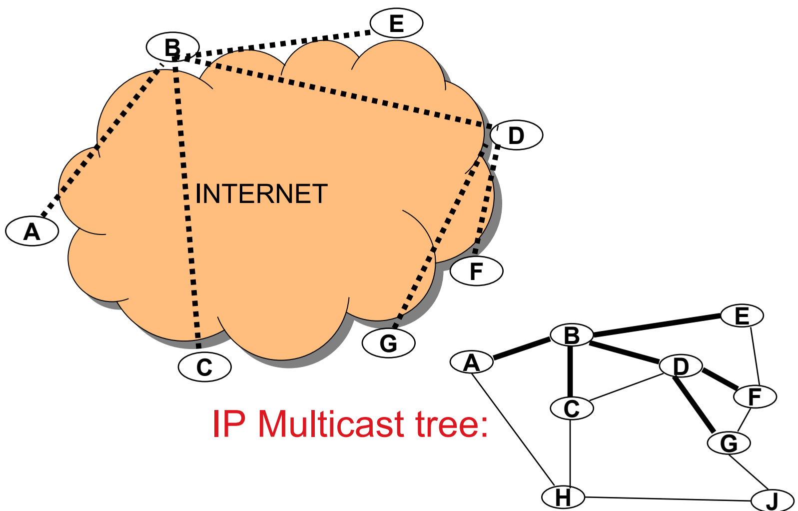
Application Layer Multicasting

Network EducationWare System

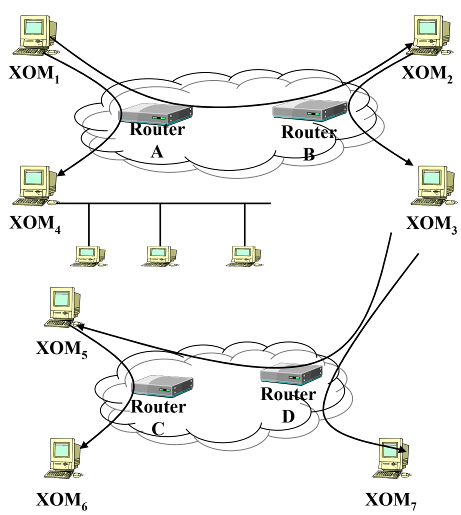


Overlay Multicast

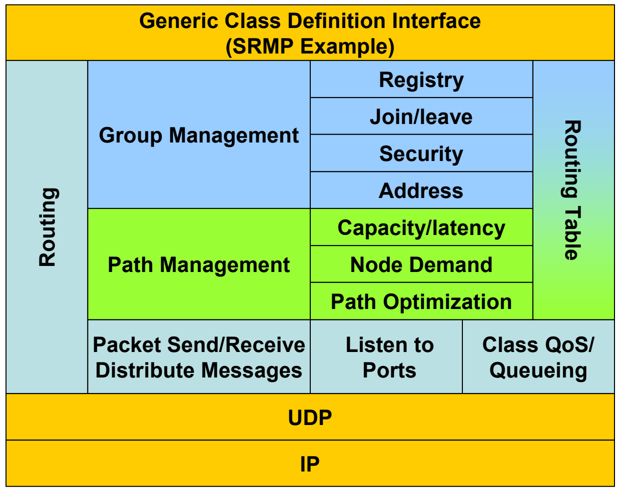
provided by a host-based middle layer



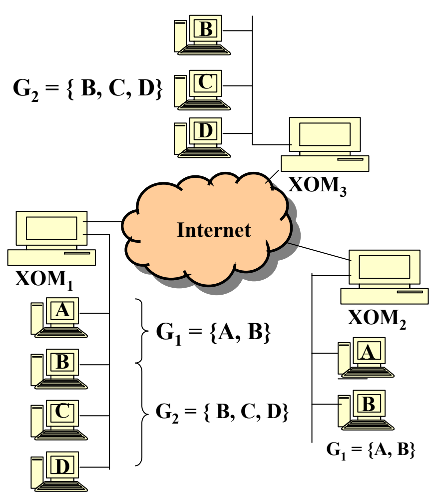
XOM Overlay



XOM Architectural Layers



XOM Group Membership



Application B sending implies routing
to group $G_3 = \{G_1 \cup G_2\}$

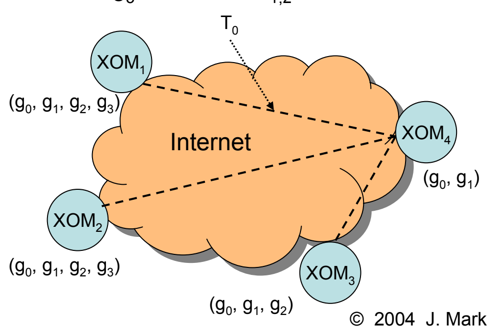
Group Aggregation Overlay
(Optimum Path Overlay)

Multicast Groups

Group	Members
g_0	$XOM_{1,2,3,4}$
g_1	$XOM_{1,2,3,4}$
g_2	$XOM_{1,2,3}$
g_3	$XOM_{1,2}$

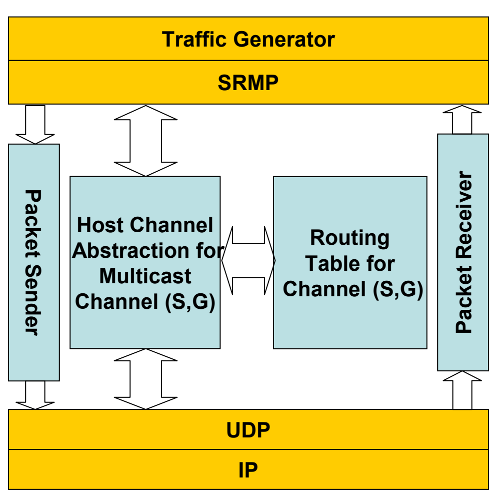
Aggregate Trees

Tree	Tree Links (arcs)
T_0	1-4, 4-2, 4-3

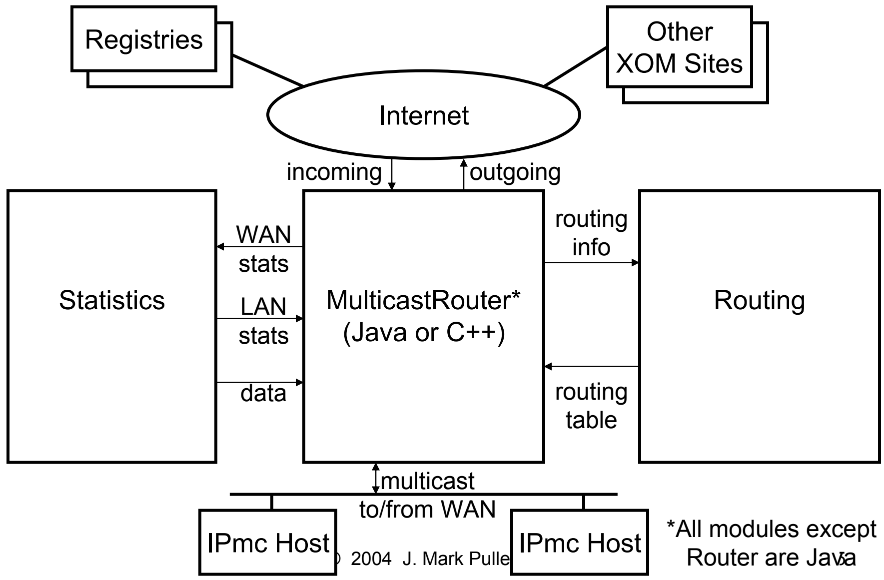


Groups g_0, g_1, g_2, g_3 share one aggregate tree T_0. T_0 is a perfect match for g_0 and g_1, but is a leaky match for g_2 and g_3. Trades off path utilization inefficiency for lower path Management overhead.

XOM Functional Model
Prototype Test Scenario

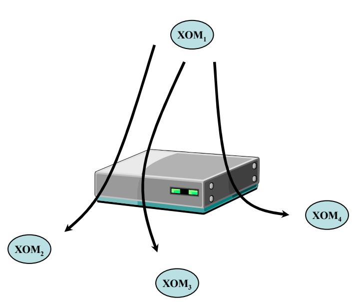


XOM Prototype

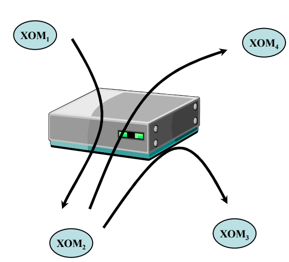


Prototype Lab Test Scenarios

- With DIS-like traffic:
  - Java version 3000 messages per second
  - C++ version 8000 messages per second

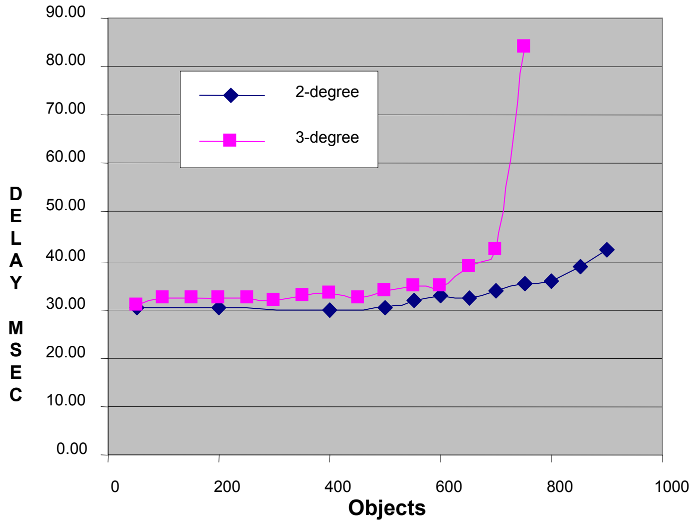


Test 1. XOM *n*-degree of 3

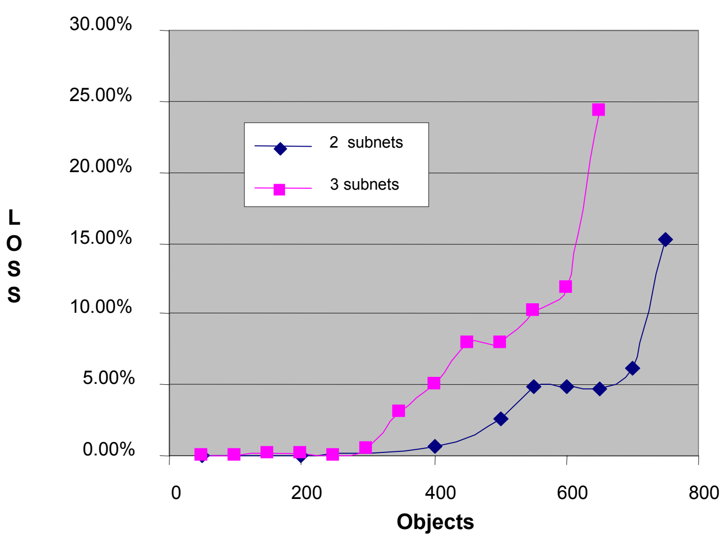


Test 2. XOM *n*-degree of 2

XOM Prototype Message Delay



XOM Prototype Message Loss Ratio



Setting Up the XOM Prototype

- One host per subnet runs XOM
  - Also can run applications
  - But for highest performance a separate host is best
- Current prototype does not support a registry
  - Each XOM must know the IP address of all others
- The XOMs exchange routing data and create a separate tree per source XOM
- Two configurations available
  1. Pure Java
  2. Router module assisted by C++ (better performance)
- Multicast applications on LAN hosts run without modification
  - XOM is transparent except in received packet “from” address

XOM Prototype Command Line Parameters

registryAddress	0 for now
numberOfMulticastGroups	count of groups/ports we will support
lowestMCAddress	first group address to multicast from the subnet, dotted decimal notation (other addresses follow in sequence)
lowest port	first UDP port to multicast (each address will get one port, in sequence)
routingUpdateInterval	time in ms between routing updates
useTCP	0 for now
partnerHostAddresses	up to 20 XOM host IP addresses in dotted decimal notation; these partners will be used without checking registry

Conclusions and Future Work

Initial results indicate overlay networking is a promising strategy for providing many-to-many multicast in the open internetwork environment of XMSF.

We are working on an architecture specification based on the properties of distributed simulation traffic plus recent networking research.

A working experimental prototype is available on our website. We plan to enhance this with better performance display.

NPS is working on a Web-service-based registry and routing information system.

Our system will be demonstrated at I/ITSEC, supporting a multicast HLA federation and Web Service-based Interest Management (WSIM).

Publications

Available on http://netlab.gmu.edu--

Pullen, J., "Reliable Multicast Network Transport for Distributed Virtual Simulation," *Proceedings of the 3rf IEEE Workshop on Distributed Simulation and Real-Time Applications,* 1999

Moen, D. and J. Pullen, "A Performance Measurement Approach for the Selectively Reliable Multicast Protocol for Distributed Simulation," *Proceedings of the 5h IEEE Workshop on Distributed Simulation and Real-Time Applications,* 2001

Moen, D. and J. Pullen, "Enabling Real-Time Distributed Virtual Simulation over the Internet Using Host-based Overlay Multicast" *Proceedings of the 7th IEEE Workshop on Distributed Simulation and Real-Time Applications,* 2003

Moen, D. and J. Pullen, "Implementation of Host-based Overlay Multicast to Support Web Based Services for RT-DVS," *Proceedings of the 8th IEEE Workshop on Distributed Simulation and Real-Time Applications,* 2004