TDK - Team Distributed Koders
Distributed Systems I

Fairness in P2P Streaming Multicast:

Software Requirements and Design

Team Members:

Kumar Keswani
John Kaeuper
Jason Winnebeck

Team Report III
2/7/07

Presentation Topics

- Software Requirements
  - Goals
  - Simulation
  - Inputs and Outputs
- Software Design
  - Class Structure
  - Pastry Algorithm
  - Freeloader Detection and Response
    - Debt Maintenance
    - Ancestor Rating
- Current Progress

Software Requirements: Goals

Explore the effectiveness of various mechanisms for enforcing fairness and incentivizing social welfare in a multi-tree peer-to-peer multicast system by using the ideas discussed in our research papers. The evaluation will be done with a discrete event simulation.

Software Requirements: Simulation

- What will not be simulated
  - Network protocol messages to structure the network
  - Routers and networks between the nodes
  - Network congestion (tree algorithms will ensure no overloads)
- What we will simulate
  - Individual packets being forwarded between nodes
  - Direct connection between nodes
  - Optional percentage of packet loss on a per-node basis
  - Algorithms
    - Node fairness detection algorithms
    - Tree construction algorithms
    - Publisher taxation algorithms

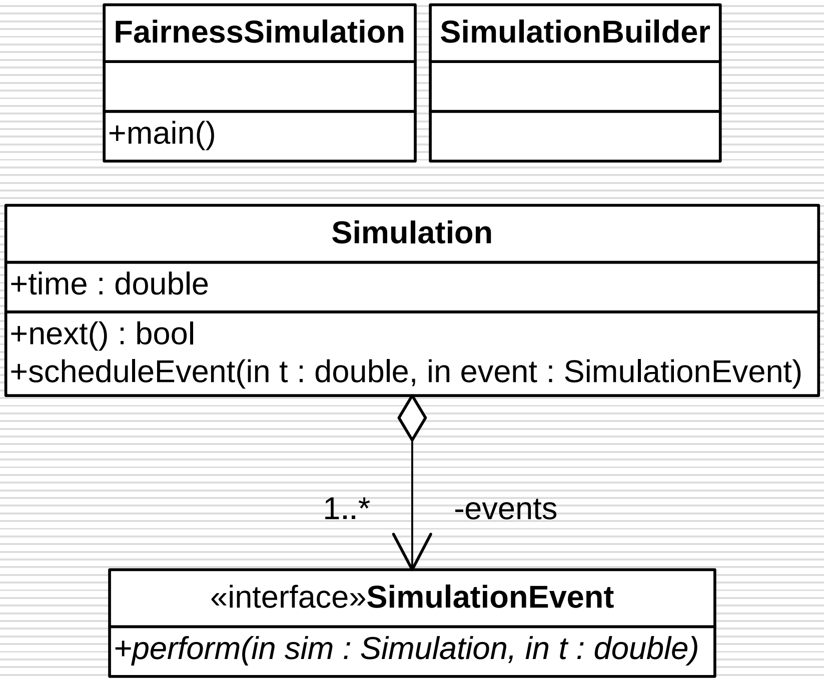
Software Requirements: Inputs

- List of nodes and their configuration
  - Behavior algorithm
  - Percentage of packet loss
  - Inbound and outbound bandwidth capacity
  - Time of entry into and time of departure from the multicast
- Stripe information
  - Number
  - Bits per second
  - Packet size
- Simulation duration

Software Requirements: Outputs

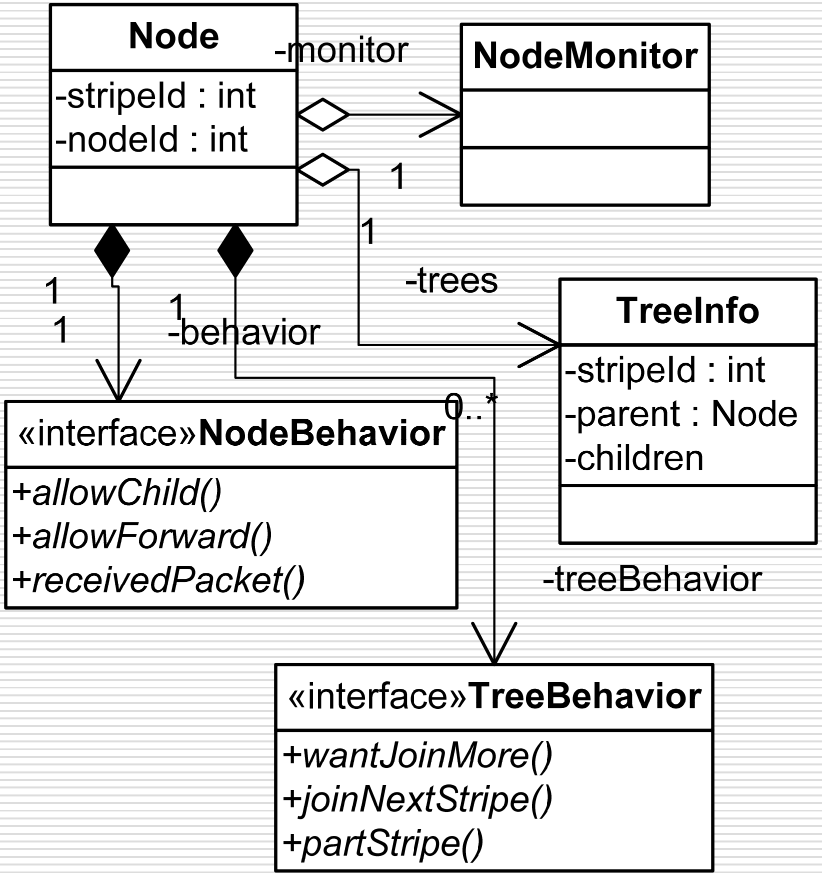
- Percentage of packets received by node behavior group over time
- Cumulative distribution function (CDF) of nodes by behavior group and debt level at end of simulation
- CDF of nodes by behavior group and negative confidence at end of simulation
- Measurements of social welfare

Software Design: Simulation



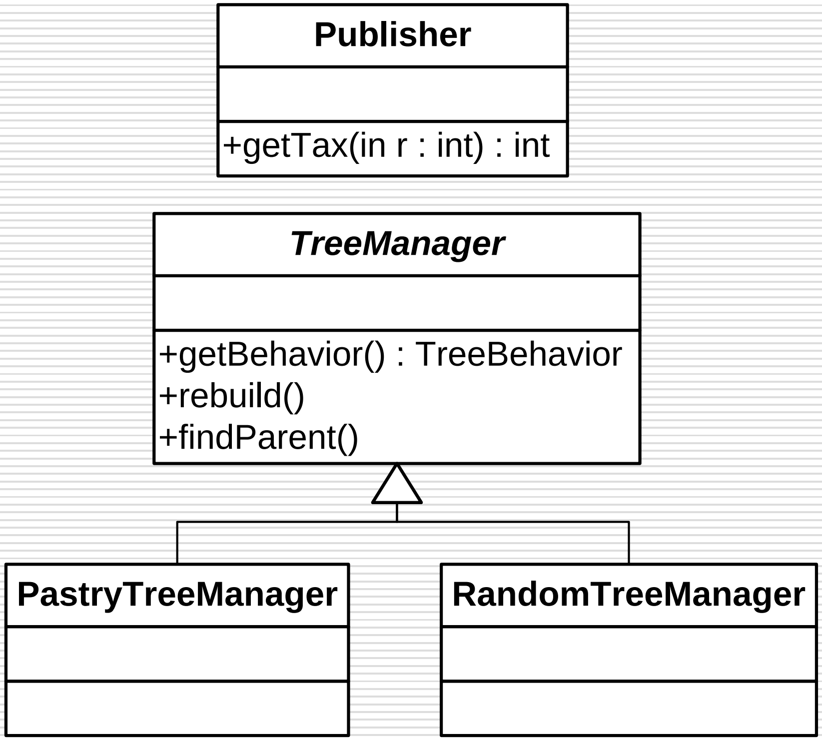
- Core classes
- FairnessSimulation and SimulationBuilder load inputs and construct all objects
- Simulation processes events in order by time

Software Design: Node



- Node is customized with a network behavior and tree behavior
- Reports events to NodeMonitor to track simulation results
- Maintains information for every tree (stripe)

Software Design: Tree Construction



- Node's TreeBehavior communicates with TreeManager during rebuild
- Specific TreeManager instance implements algorithms
- Publisher will adjust number of maximum children a node must take if taxation is enabled

Tree Construction Algorithms: Pastry Tree Algorithm

- Core algorithm executed during tree construction event:
- findParent() method will be called for each Node for each stripe they want to join:
  - Construct a Pastry route from Node to stripe tree root, following Nodes with progressively longer prefix matches between their nodeID's and stripeID (the Pastry routing method)

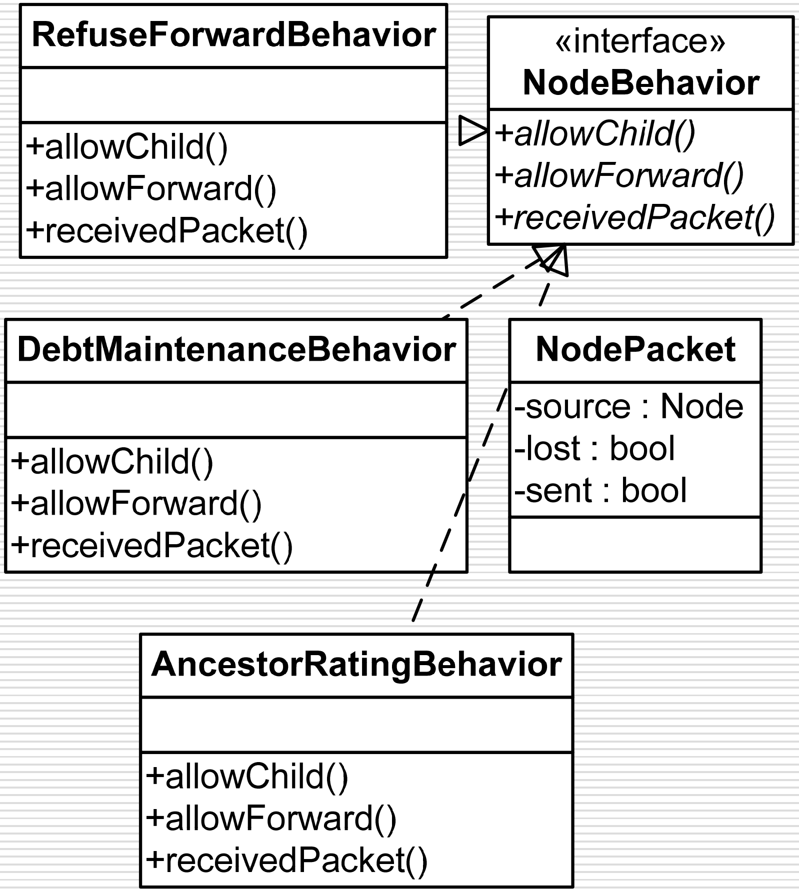
Tree Construction Algorithms: Pastry Tree Algorithm (continued)

- orphanChild(): method called for Nodes along path exceeding max children
  - selects orphan
  - finds parent from former siblings
  - if cannot find parent, search Spare Capacity Group (SCG)
- searchSCG():
  - Searches SCG list to find parent for orphan
  - SCG just a list of spare capacity nodes, not a tree as in real SplitStream mechanism
  - This mecanism may break the interior node disjoint property, although unlikely

Tree Construction Algorithms: Random Tree Algorithm

- In Pastry Tree Algorithm, order that Nodes (with given set of desired stripes for each) added to forest by findParent determines forest structure
- But in Random Tree Algorithm forest structure will be different each time regardless of join order. Nodes take turns at random to join a random stripe under a random parent.

Software Design: Sending Packets



- Nodes receive NodePackets during simulation
- Even “lost” and “unsent” packets are given to track statistics in NodeMonitor
- Packet forwarding and receiving cause interactions with NodeBehavior

Software Design: Freeloader Detection and Response

- Debt Maintenance
  - Track debts of immediate peers (parent and child)
- Ancestor Rating
  - Track confidence in all nodes in path to root and in immediate children
- Each node keeps independent track of its own view of its peers.
  - i.e. A's confidence of B may be different than C's confidence of B.

Freeloader Detection and Response
Debt Maintenance

- When A sends a packet to B, A does:
- If (A.debt[B] >= threshold)
  - Packet sent to B has ‘sent’ flag false
  - Else, send packet to B with ‘sent’ flag true and A.debt[B]++
- When B receives a packet:
- If (packet.sent && !packet.lost)
  - B.Debt[A]--

Freeloader Detection and Response Ancestor Rating

- When A sends a packet to B, A does:
- If (A.confidence[B] < threshold)
  - Packet sent to B has ‘sent’ flag false
  - Else, send packet to B with ‘sent’ flag true
- When B receives a packet:
- If (packet.sent && !packet.lost)
  - Increase by 1 B’s confidence of A and all of its parents
  - Else, decrease B’s confidence of A and all of its parents

Freeloader Detection and Response
Ancestor Rating

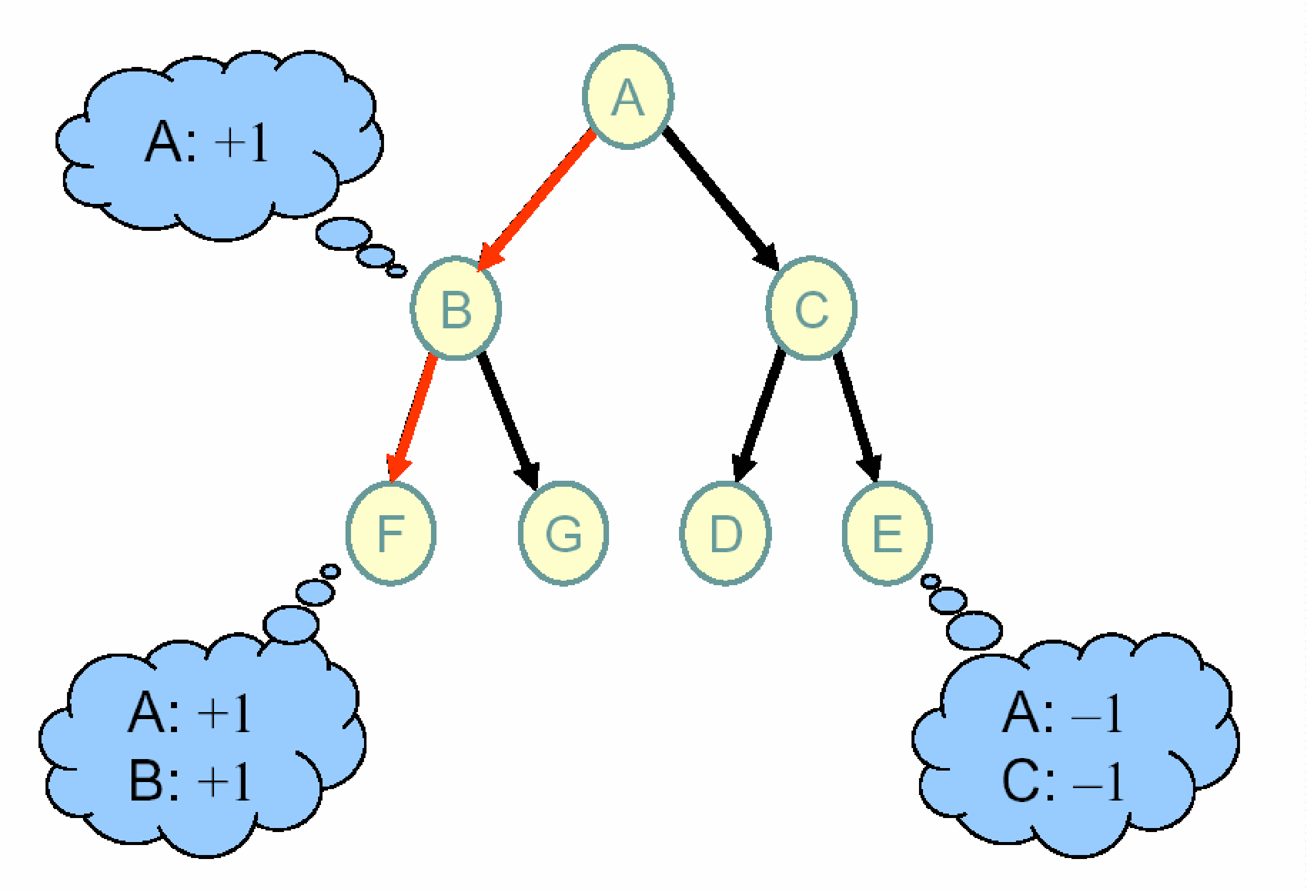


Diagram from: Daniel Garrison, CS 699,George Mason University,November 4,2004 http://cs.gmu.edu/~setia/cs699/incentives.pdf

Current Progress

- Design is completed
- Construction of stub classes
- No implementation to demo yet

Research Papers

1. Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A., and Singh, A. 2003. SplitStream: high-bandwidth multicast in cooperative environments. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (Bolton Landing, NY, USA, October 19 - 22, 2003). SOSP '03. ACM Press, New York, NY, 298-313. DOI= http://doi.acm.org/10.1145/945445.945474
2. T. W. J. Ngan, D. S. Wallach, and P. Druschel. Incentives-Compatible Peer-to-Peer Multicast. In The Second Workshop on the Economics of Peer-to-Peer Systems, July 2004. http://citeseer.ist.psu.edu/ngan04incentivescompatible.html
3. Chu, Y. 2004. A case for taxation in peer-to-peer streaming broadcast. In Proceedings of the ACM SIGCOMM Workshop on Practice and theory of incentives in Networked Systems (September 2004). ACM Press, New York, NY, 205-212. DOI= http://doi.acm.org/10.1145/1016527.1016535