TDK - Team Distributed Koders
Distributed Systems I

Fairness in P2P Streaming Multicast:

Research Paper Presentation

Team Members:

Kumar Keswani
John Kaeuper
Jason Winnebeck

Team Report II
1/24/07

Presentation Topics

- Research Paper Presentation
  - SplitStream
  - Incentives-Compatible P2P Multicast
  - Taxation in P2P Streaming Broadcast
- Future Work

Paper 1: SplitStream

- SplitStream: high-bandwidth multicast in cooperative environments
  - Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles
  - October 19 - 22, 2003

Problems

- Single tree-based multicast systems poor choice for P2P network

1. Small number of interior nodes bear forwarding burden
2. Only acceptable if interior nodes are highly-available, dedicated infrastructure routers
3. Many multicast applications need high bandwidth, so many nodes can't handle forwarding
4. Poor fault-tolerance – if one node fails, some nodes receive none of original content
5. Poor scalability

Solutions

- *Split* the original content into *k* stripes and multicast each stripe in a separate tree
- Nodes join trees of stripes they want to receive and specify upper-bound on number of children they will accept
- Solution has 2 main goals:
  1. Forest of trees is interior-node-disjoint
  2. Forest must satisfy node bandwidth constraints

Solutions (continued)

- Forwarding load is now distributed
- System more fault-tolerant (applications using SplitStream can use data encodings to reconstruct content from less than *k* stripes)
- Enhanced scalability

Feasibility of Forest Construction

- <u>Def</u>: for node set N and source set $S \subseteq N$, it is possible to connect nodes such that each node $i \in N$ gets I_i <u>distinct</u> stripes and has no more than C_i children (I_i is desired indegree and C_i is forwarding capacity)

- 2 conditions for feasibility:
  1. Necessary (but not sufficient): $\sum_{\forall i \in N} I_i \leq \sum_{\forall i \in N} C_i$
  2. Sufficient: if node can forward more than it wants to receive, it must receive (or originate, if source) all k stripes: $\forall i : C_i > I_i \Rightarrow I_i + T_i = k.$
  3. High probability of feasibility if 2 conditions met and there is reasonable spare capacity:

$$C = \sum_{\forall i \in N} C_i - \sum_{\forall i \in N} I_i$$

Implementation of Solution

- Basic architecture: Scribe group communication system on top of Pastry overlay protocol

- Pastry: P2P overlay network

  1. Nodes assigned 128-bit *nodeId*
  2. Messages sent with 128-bit *keys* - message routed to node with nodeId numerically closest to key, called the key's *root*

- Scribe: application-level group communication system upon Pastry

  1. Multicast groups (trees) given pseudo-random Pastry keys called *groupId* (groupId's root is root of multicast tree)
  2. Multicast trees formed by combining Pastry routes from group members to groupId's root

Solution Design

- Recall interior-node-disjoint goal

1. How? Scribe trees are formed from Pastry routes between tree members and the groupId (the tree root), and Pastry routes messages to nodeId's sharing progressively longer prefixes with groupId
2. so interior nodeId's share some digits with groupId
3. Simply make all groupId's differ in most significant digit – then trees will be interior-node-disjoint

Solution Design (continued)

- Recall node bandwidth satisfaction goal

1. Inbound bandwidth satisfied by joining trees of desired stripes
2. Satisfying outbound bandwidth involves orphaning nodes: if node attempts to be child of parent with exhausted outbound bandwidth, child taken, but then some child (possibly same child) is orphaned
   1. First, parent tries to orphan child of tree in which the parent's nodeId shares no prefix with that tree's groupId
   2. If no such child, pick child w/ shortest common prefix w/ groupId
   3. Orphan attempts to be child of its former siblings; 1) + 2) applied recursively until orphan finds parent or no siblings share a nodeId prefix with the tree groupId
   4. If orphan cannot find parent, it anycasts to the Spare Capacity Group; DFS of SCG will find node in stripe tree needed by orphan

Paper 2

- Incentives-Compatible Peer-to-Peer Multicast
  - The Second Workshop on the Economics of Peer-to-Peer Systems
  - July 2004

Goals

- Have nodes observe their peers to:
  - Prevent freeloading
    - Nodes that refuse to forward packets
    - Nodes that refuse to accept children
  - Detect Freeloaders
  - Stop servicing Freeloaders

Fairness Mechanism 1
Debt Maintenance

- Consider two nodes A and B.
- A sends a stream of data to B.
- Both the nodes A and B keep a track of record.
- Both A and B know B owes A a debt of one packet.
- If debt exceeds some threshold value, A refuses to service B.

Fairness Mechanism 2
Ancestor Rating

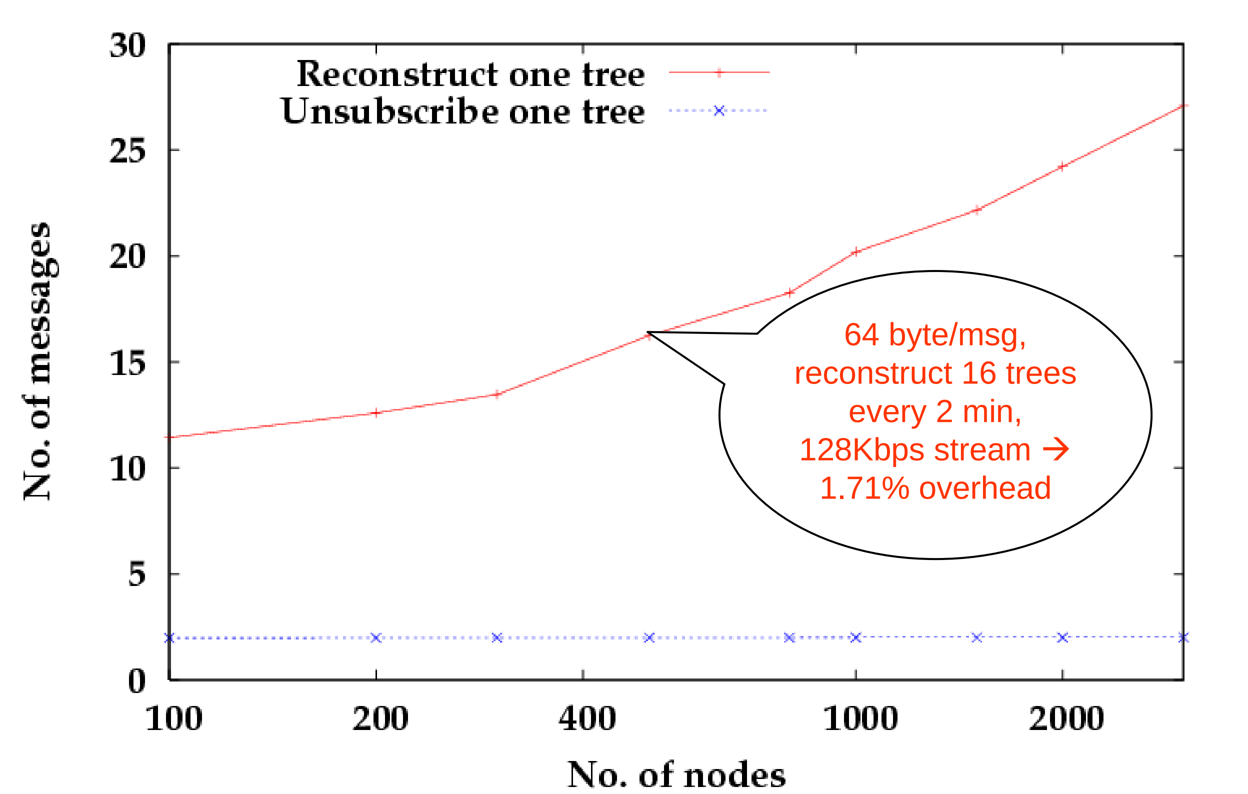
- An extension to Debt Maintenance
- Apply debts not only to immediate parents but to all of its ancestors
- If a packet is not received by the child it assigns ‘equal blame’ to all its ancestors
- Reduce the confidence level of each node in the path to the root
- If packet is received, all ancestors get equal credit and confidence level is increased

Fairness Mechanism 3
Tree Reconstruction

- Periodically rebuild the forest trees to identify freeloaders
- Keeps a track of debts in parent-child role by rebuilding the tree periodically
- Identifies ‘innocent nodes’ blamed because of their child’s selfish behavior
- Only selfish nodes will keep on accumulating debt

Tree Reconstruction Cost

- Figure shows average number of messages sent by each node in order to construct a tree



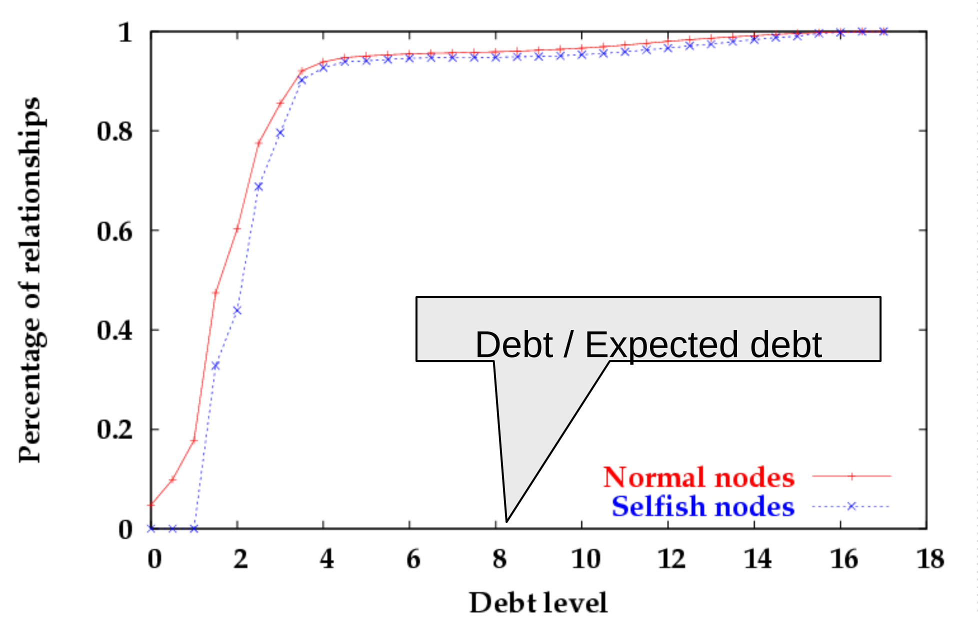
Other Fairness Mechanisms

- Parental Availability
  - If any parent continuously refuses to accept children, child identifies it as a freeloader
- Reciprocal Requests
  - If for A and B, A is much more often the child, allow B to break standard join protocol and try to join A
- Sybil Attack Prevention
  - New nodes start on a “probation period”

Results: Debt Maintenance

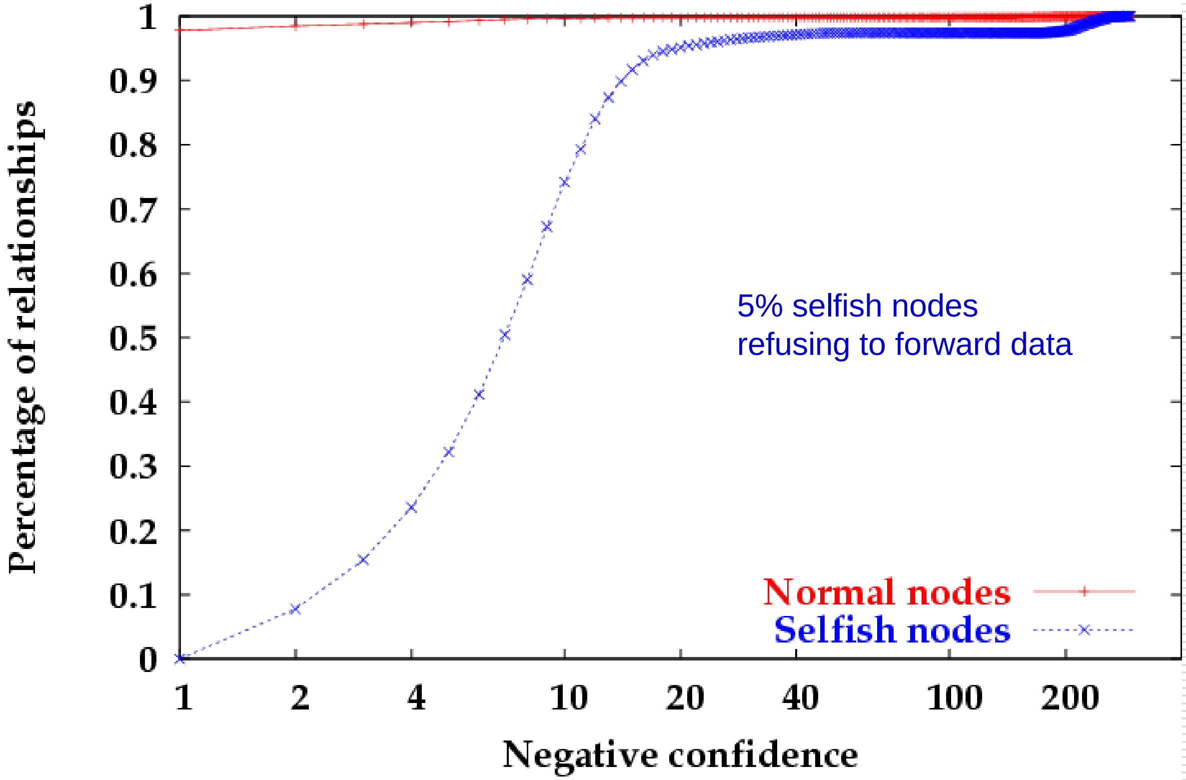
- Sensitive to tree reconstruction method
  - Pastry, for example, chooses similar trees on each rebuild because it favors local paths.

$$\text{Debt Level} = \frac{\text{accumulated debts/credits}}{\sqrt{\text{total transfers}}}$$



Results: Ancestor Rating

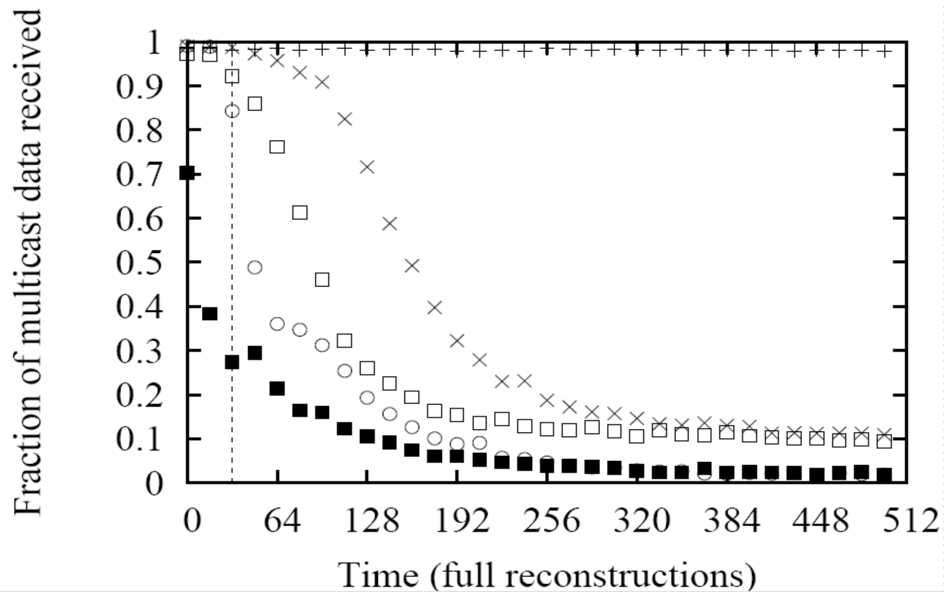
- Figure shows negative confidence distribution after 256 full tree reconstructions



Experiment

- 500 nodes with 4 selfish nodes
- 2 types of selfish nodes
- Nodes will forward to children unless its child's
  - Confidence value <-2 or
  - Parental Availability <0.44 and Confidence value <0.2

Type	Count	Description
+	496	Normal nodes
×	1	Refuse to accept children after 32
⊡	1	Always refuse to accept children
⊙	1	Refuse to forward data after 32
■	1	Always refuse to forward data



Paper 3: Taxation

- A case for taxation in peer-to-peer streaming broadcast
  - Proceedings of the ACM SIGCOMM Workshop on Practice and theory of incentives in Networked Systems
  - September 2004

Taxation: Goals

- Goal: Improve on *bit-for-bit* P2P streaming model to maximize social welfare
- Social welfare: Aggregate of utility, which is benefit minus cost
- Idea: Achieve through increasing contribution of resource-rich peers
- Work Based on ESM: http://esm.cs.cmu.edu/

Taxation: Environment

- Resource-poor (cable, DSL) versus resource-rich peers
- In P2P streaming “the publisher of the video stream has the means to enforce taxation and the will to maximize their collective social welfare”
  - Means: Proprietary software (the viewer)
  - Will: Better overall video quality means more viewers
- Strategic peers: maximize utility

Taxation: Utility

- Utility is defined as benefit minus cost
- Benefit is based entirely on received bandwidth (content quality)

$$b(r) = \sqrt{r}$$

- Cost is based on percentage of outgoing bandwidth

$$c(f,F) = \alpha * \sqrt{F} * p(f,F)$$

$$p(f,F) = \beta * \left(\frac{f}{F}\right) + (1-\beta) * \left(\frac{f}{F}\right)^4$$

$$\alpha = 0.75, \beta = 0.5$$

Taxation: Tax Schedule

- Properties of a good tax scheme
  - Asymmetric roles and power
  - Public and fixed tax schedule
  - Fairness (horizontal and vertical)
  - Budget Balanced

Taxation: Tax Schedule

- Linear tax schedule based on receive rate **r** and contribution **f**
  - $f = \max(t * (r - G), 0)$
- Based on two fixed parameters
  - t – tax rate (fixed)
  - G – demogrant (dynamic)
- Demogrant is a form of base income

* For an economic perspective on demogrants, see http://bostonreview.net/BR25.5/phelps.html

Taxation: Implementation

- Entitled bandwidth is G + f
- Nodes assign priority to trees, highest priority for each entitled tree, then decreasing order for all others
- Higher priority (entitled) nodes preempt lower priority in join process
- Publisher dynamically adjusts G:
  - Start with G as 0
  - Increase G by one each round until budget is balanced

Taxation: Strengths and Weaknesses

- Strengths
  - Improves social welfare in heterogeneous environments
  - Linear scheme simple to implement and works as well as non-linear
- Weaknesses
  - Tax rate must be chosen by publisher
  - Protocol relies heavily on trust in client software

Future Work

- Main importance is the 2 goals of the SplitStream design
  1. We will implement interior-node-disjoint forest
  2. We will implement forest satisfying bandwidth constraints

- We will not be using Pastry and Scribe

- Focus is on enforcing fairness through:
  1. Debt maintenance
  2. Ancestor rating
  3. Attempt to incorporate taxation scheme with previous fairness algorithms

References

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