

Incentives for P2P Systems

Daniel Garrison
CS 699

George Mason University
November 4, 2004



Incentives for P2P Systems



- Incentives Build Robustness in BitTorrent
 - Bram Cohen
 - Workshop on Economics of Peer-to-Peer Systems, 2003
- Incentives-Compatible Peer-to-Peer Multicast
 - Tsuen-Wan “Johnny” Ngan, et al.
 - 2nd Workshop on Economics of Peer-to-Peer Systems, 2004

Incentives Build Robustness in BitTorrent

Bram Cohen

*based on slides by Fan Bin, Karthik
Tamilmani, Brian Liao*



Background

- Traditional Client/Server Sharing
 - Performance deteriorates rapidly as the number of clients increases
- Free-riding in P2P network
 - Free riders only download without contributing to the network.

BitTorrent Design Overview



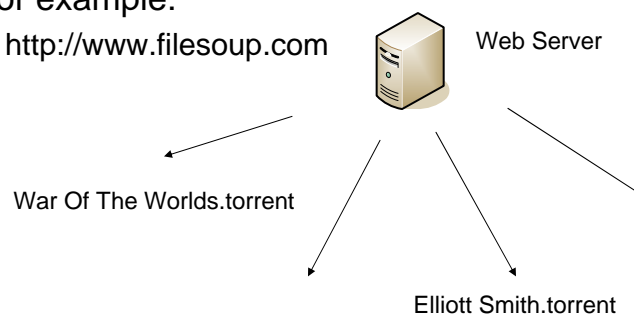
- New file-sharing strategy:
 - Minimize the file hosting cost
 - Increase scalability
- Based on Tit-for-tat
- Pieces of files
- Critical elements:
 - Web server
 - *Client browser*
 - BitTorrent tracker
 - Peers
 - Seeder, Leecher, Reseeders

5

Critical Elements



- A web server
 - Provide a 'metainfo' file by HTTP
 - For example:
 - <http://www.filesoup.com>



6

Critical Elements



- The .torrent file
 - Static 'metainfo' file contains necessary information :
 - File
 - Pieces
 - Checksum
 - IP address of the Tracker



Elliot Smith.torrent

7

Critical Elements



- A BitTorrent tracker
 - Non-content-sharing node
 - Track peers
 - For example:
 - <http://diganote.com:8080/announce>
 - <http://www.musicorrents.net:6969/announce7>

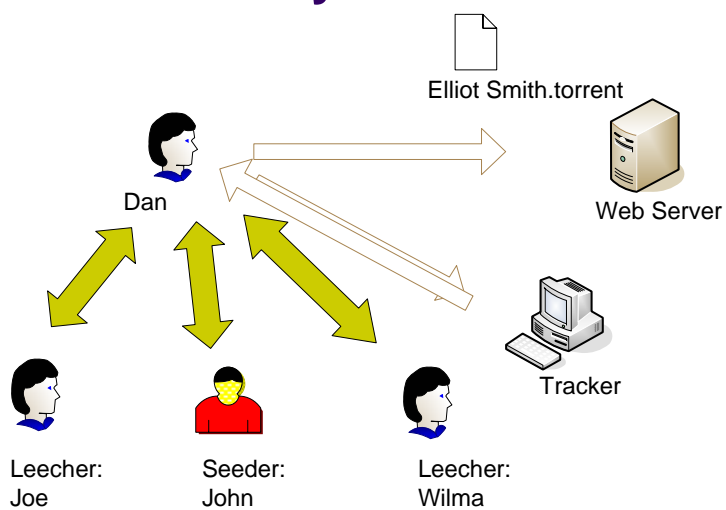
8

Critical Elements

- The peers (end users)
 - Seeder
 - complete file
 - Leecher
 - part (or none) of the file.
 - Reseeder
 - helping the community

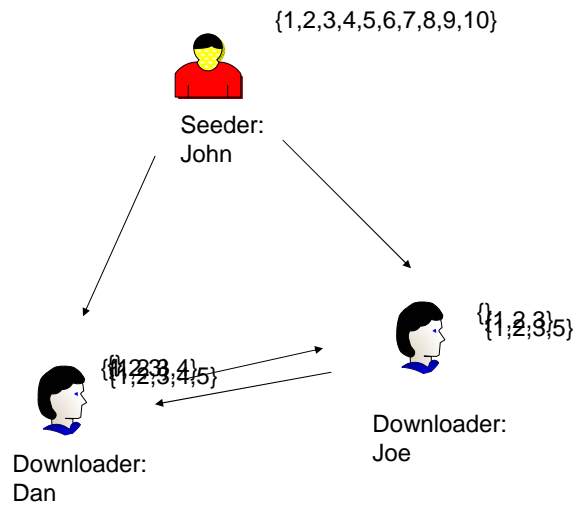
9

Connectivity



10

Piecing



11

Piece Selection



- Strict Priority
 - First Priority
- Rarest First
 - General rules
- Random First Piece
 - Special case, at the beginning
- Endgame Mode
 - Special case, in the end

12

Peer Selection



- Built-in incentive mechanism:
 - *Choking Algorithm*
 - *Optimistic Unchoking*
 - *Anti-snubbing*

13

Choking Algorithm



- Choking
 - A temporal refusal to upload
- Fixed number of leechers remain unchoked
- Peer performance determination
 - Uses 20-second average
- Choking refreshed every 10 seconds

14

Optimistic Unchoking



- An additional leecher remains unchoked
- Peer performance determination not used
- Rotated every 30 seconds

15

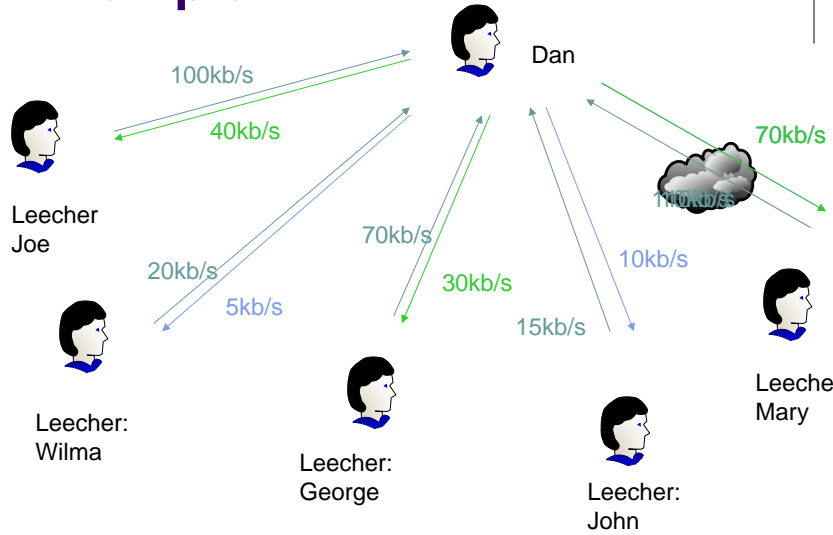
Anti-snubbing



- Exception to one optimistic unchoking
- Leecher may be snubbed by all peers
- Snubbing determination
 - Leecher receives no data for 60s
- Refuses to upload to it
 - May be chosen for optimistic unchoking

16

Example



17

Download Completed

- New choking decisions
 - Give best upload rate
 - Choked leechers

18

Conclusion



- BitTorrent is a well thought-out protocol that embraces aspects of influenced cooperation and self-optimizing mechanisms.
- BitTorrent proposes solutions for current optimization and scalability problems.

19

Incentives-Compatible Peer-to-Peer Multicast

Tsuen-Wan “Johnny” Ngan
with Dan Wallach and Peter Druschel
Rice University





Background

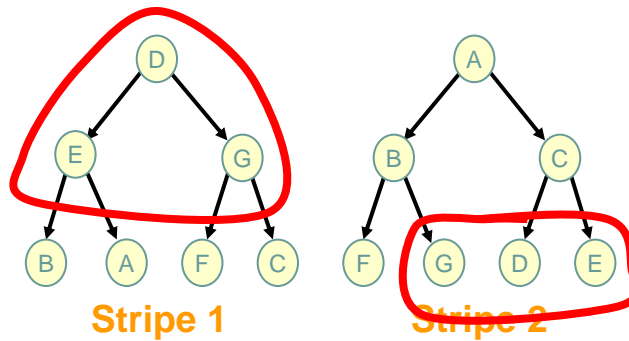
- P2P multicast
- Existing systems rely heavily on cooperation
- Most incentivized solutions not suitable
 - Storage solutions
 - Bandwidth solutions
- Do not map onto multicast fairness
 - Rely on tit-for-tat
 - Multicast trees are usually static

21



SplitStream Concept

Interior node is a leaf node in the other trees

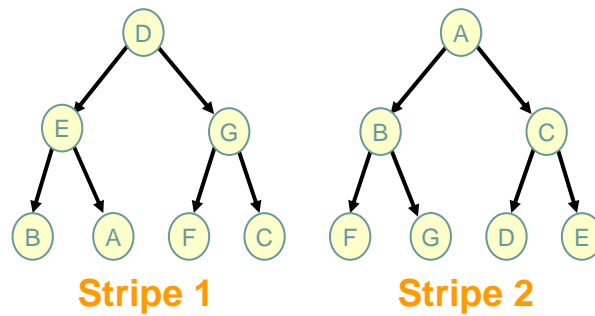


22

SplitStream Reliability



Peer failure only affects a single stripe

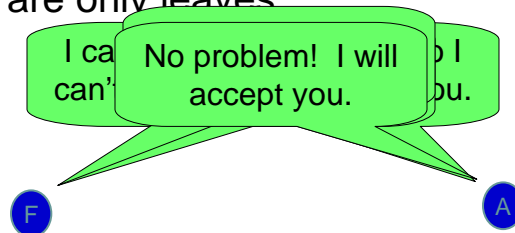


23

Freeloading Model



- Assume rationality
- Nodes can refuse to accept children
- Nodes can refuse forwarding data
- Nodes are only leaves



24

Design Overview



- Distinguish nodes with selfish behavior
- Periodic tree reconstructions
 - Avoid suffering forever
 - Potentially reversing parent-child relationships
- Measure various metrics
 - Combine to form a robust policy

25

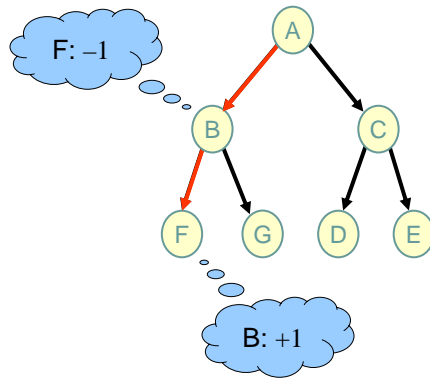
Naïve



- Disallows data dropping
- Disadvantages:
 - Wastes bandwidth
 - Can still refuse children

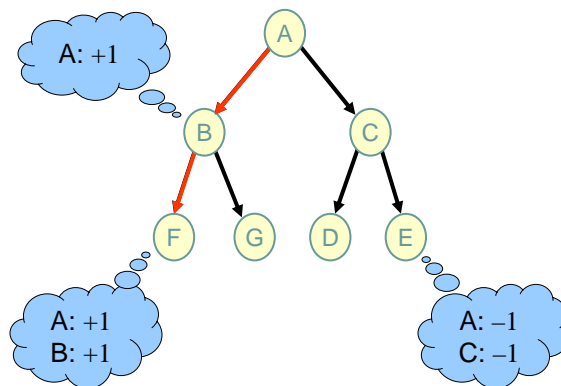
26

Pairwise Debt



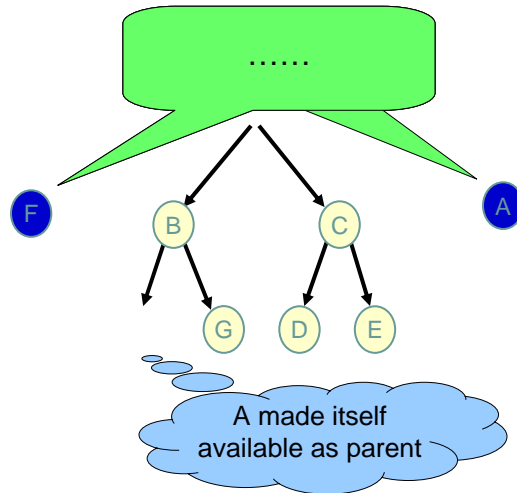
27

Ancestor Rating (Confidence)



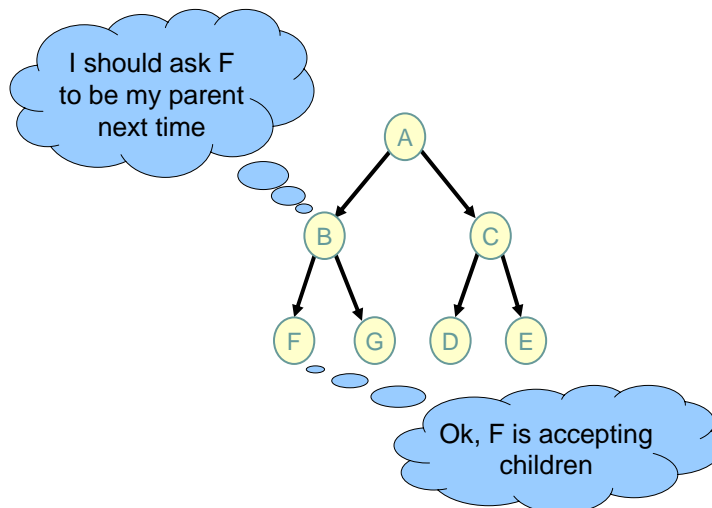
28

Parental Availability (PA)



29

Reciprocal Request



30

Truthfulness of Parents



- Hash chain used to verify path
- Hashes message digest with nodeId

31

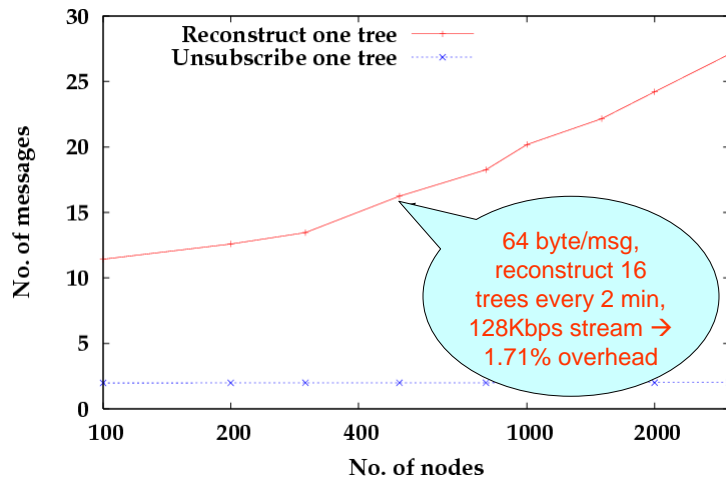
Experiments



- On SplitStream, part of FreePastry
- Stochastic model for node proximity
 - 500 nodes randomly distributed on a plane
- Each node subscribe to 16 trees
- Good nodes accept up to 16 children

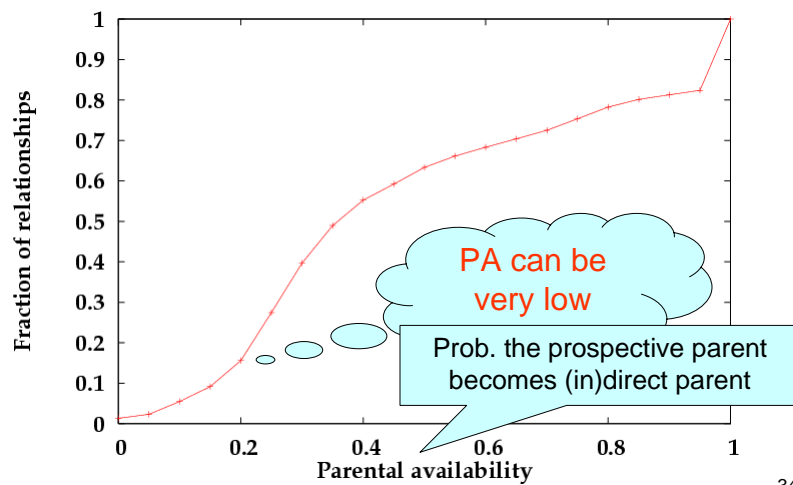
32

Tree Reconstruction Cost



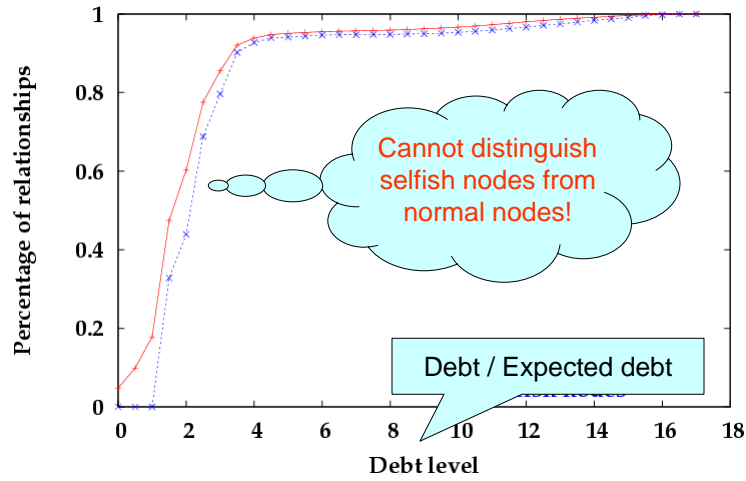
33

Parental Availability (PA)



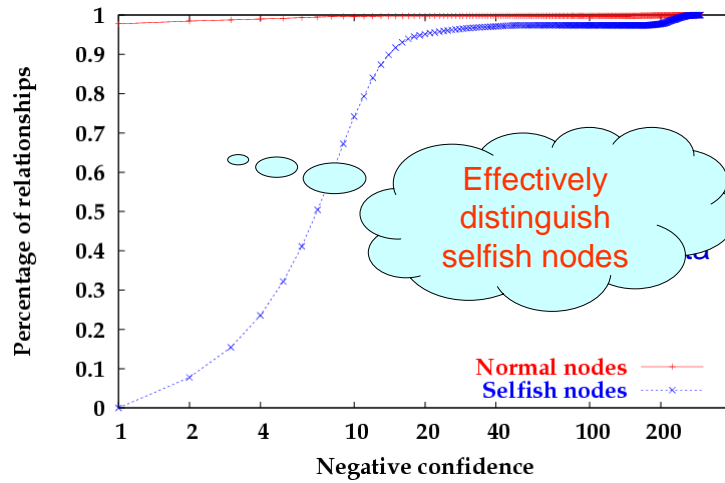
34

Debt Level



35

Confidence



36

Enforcing Policy



- A simple policy using the above schemes
- Two types of selfish nodes
 - Refuse to accept children
 - Accept children but refuse to forward data
- Different start time to freeload
 - Begin cheating immediately
 - Start only after time 32

37

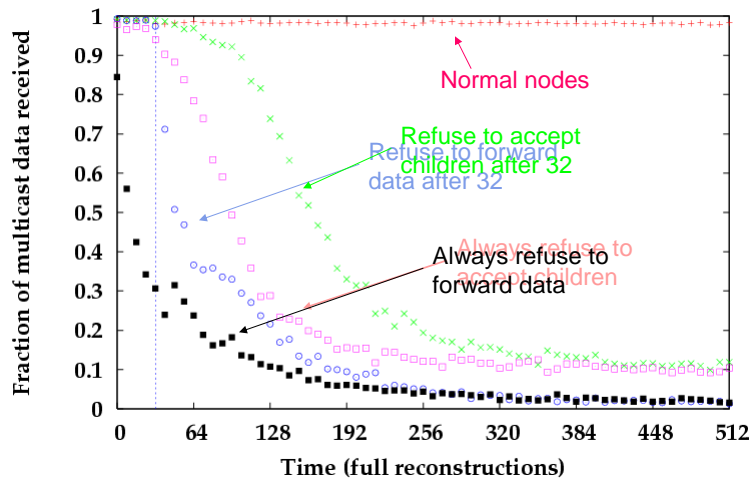
The Policy



- Not to use debt level
- Normal nodes will not serve those with
 - Confidence < -2
 - PA < 0.44 and confidence < 0.2
- Positive confidence/PA decay over time
- Allow preemption if 0.1 higher in PA
- Reciprocal requests if requests are 8 times more

38

Result from Enforcing Policy



39

More on Policy

- Increasing selfish nodes reduces the reception of normal nodes
 - 4% selfish nodes → 90% reception
- Can use encoding
 - Receive above a certain fraction of data to decode anything at all
 - Freeloaders get no service, would probably leave

40

Concluding Remarks



- Mechanism effective by tracking only first-hand observed behavior
- Low network and computation overhead
- Future work:
 - Robustness against more freeloaders
 - Learn parameters using Bayesian approach
 - Study dependence on multicast application, p2p substrate, and network topology