

p2p Fairness and Performance

Jason But

jbut@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology



BitTorrent Introduction

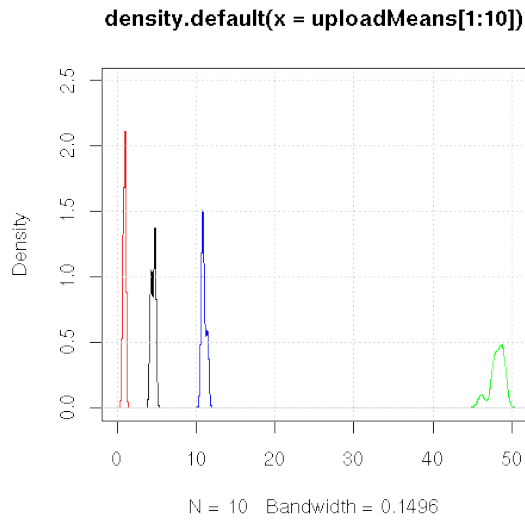


What is BitTorrent?

- File sharing application
- Peer-2-peer based
- Distributed in nature

Why is it Interesting?

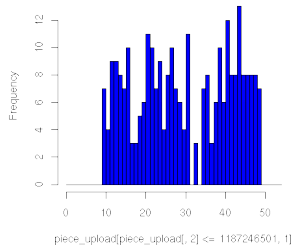
- Each node is independent
- How well does the algorithm ensure that fairness is maintained between each participating node
- Is the available bandwidth usage maximised



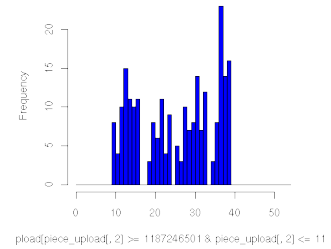
- Each class exhibits an even spread of uploaded data rates
- Higher capacity clients contribute more to transfer



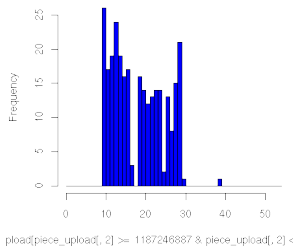
stogram of piece_upload[piece_upload[, 2] <= 118724]



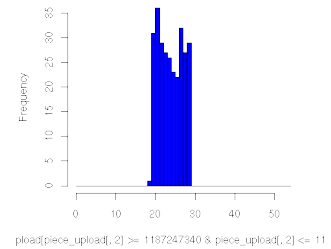
_upload[piece_upload[, 2] >= 1187246501 & piece_uplc

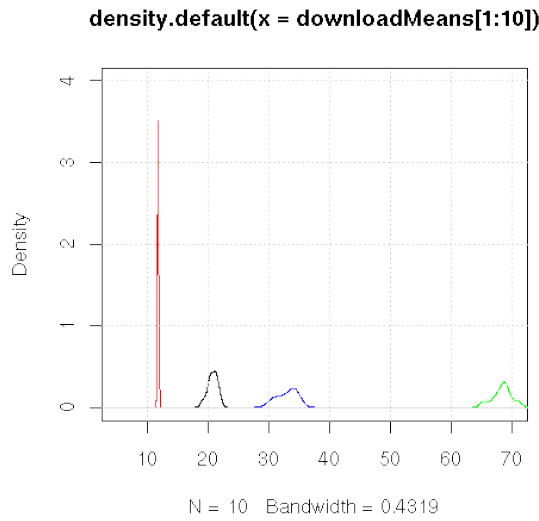


_upload[piece_upload[, 2] >= 1187246887 & piece_uplc



_upload[piece_upload[, 2] >= 1187247340 & piece_uplc

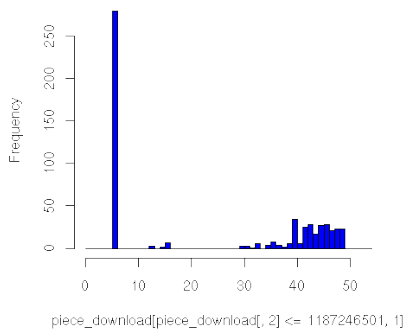




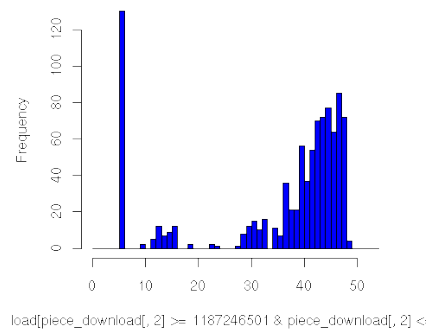
- Clients download according to capacity
- No one client within class is favoured



gram of piece_download[piece_download[, 2] <= 1187



mload[piece_download[, 2] >= 1187246501 & piece_dc



Future Work - Local Caches



- BitTorrent skews traffic witnessed by ISPs – more p2p increases the amount of traffic in/egressing the ISP network
- Can the ISP benefit by installing a local cache
- Local BitTorrent client connected at high-speed (say 100Mb/s)
- Servicing only ISP customers

Initial Results

- Are promising
- Indicate that a high-speed node will preferentially seed content to locally connected nodes
- Doesn't greatly impact on performance of remote nodes



Future Work - Improving Protocol



- BitTorrent is based on the TCP protocol
 - Data transfer is serial
 - But data flow can be split in two - Control and data
- Is there an alternative that may allow us to improve the performance?
- Explore the possibility of using SCTP as an underlying protocol with multiple streams
- SCTP socket API may actually simplify the code/implementation

