Do incentives build robustness in BitTorrent?

Rakesh Gopinath Nirmala Arunakumari Yedurupaka

I. Motivation

One of the rudimentary problems in peer to peer systems is resource consumption by a peer without any contribution termed as free-riding. To overcome this problem, BitTorrent with its Tit-for-Tat (TFT) reciprocity strategy provides more download rates to peers contributing resources to the swarm. But, it is skeptical as to whether this popular incentives belief really holds for peers exhibiting different upload and download behaviors. If yes, is the performance improvement gained proportional to the amount of contribution. To answer these questions, the authors have developed a simple model to validate this belief and measure the performance gain by using diverse real-world BitTorrent traffic.

II. Contributions

Firstly, authors capture and analyze the real traffic of BitTorrent clients deployed across the globe. Based on the analyses of upload and download rates, equal split rates, active set size, neighborhood size, number of optimistically choked peers etc., they developed a model to find sources of altruism and explained how it can be exploited. Based on the evaluations, they also invalidate the widely flourished incentives belief.

Secondly, they have designed a new BitTorrent client (BitTyrant) based on the observations from the model incorporating features for strategic peers to benefit, evaluated it on real and synthetic swarms and results are presented showing performance gains. The paper also discusses a handful of cheat strategies for the peers to get better download rates.

III. Solution

The proposed solution is an optimized BitTorrent client for strategic peers. The model observations that led to the development of BitTyrant are as explained. Each BitTorrent peer connected to a set of random peers called neighbors exchange control messages regarding the data they possess and interested blocks. Based on upload rates, in each round each peer selects an active set to transmit actual data at equal split rates. Using opportunistic measurement techniques and several assumptions, various parameters of BitTorrent are derived, analyzed and a model is built to find sources of altruism. The measurements suggests that TFT takes time for the high capacity peers to reach a steady state, low bandwidth peers almost never earn reciprocation yet receive data at faster rates due to optimistic unchokes, performance improvement gained by high capacity peers is not proportional to their amount of contribution and active set size has to be largely increased for high capacity peers to improve their download rates proportionally.

BitTyrant is designed based on the popular BitTorrent client Azureus with an aim to maximize reciprocation. To do so, strategies to find peers with higher reciprocation bandwidth, to increase the number of the reciprocating peers in the active set incrementally and to reduce the upload contribution to each peer without affecting download rates from those peers are implemented. The BitTorrents unchoke algorithm is modified to incorporate changing active set size and varying upload rate per peer at run-time. With a less overhead, protocol policy was also modified to increase the local neighborhood size which increases the active set size and thus, maximize the download throughput. From the evaluation, BitTyrant provides more consistent performance with about 70% median gain for a 1Mbps client. It discovers the reducing performance gains and reallocates the bandwidth among connections. This encourages high-capacity peers to take part in multiple swarms in parallel. However, with BitTyrant new peers face elongated bootstrap periods. It could also limit performance gains for all users if selfish peers increases in the system.

IV. Strong Points

1) The performance modeling invalidates the long held belief and clearly elucidates the irregularities related to performance gain and the upload contributions of the incentives based approach and also explains the various ways of exploiting altruism.
2) BitTyrant is developed without any change to the underlying BitTorrent protocol but only policy changes by implementing various strategies discussed.
3) BitTyrant introduces dynamic active sets and the bandwidth reallocation on a per-peer basis and provides a clear picture of how these changes affect the download performances. This way it also fosters the high-capacity peers to participate in multiple swarms.

V. Weak Points

1) In allowing high capacity peers to take part in multiple swarms, BitTyrant reduces download rates for other users in the swarm thus decreasing the swarm performance. Low-capacity peers are most hit being unable to compete with the high-capacity peers.
2) Also, multiple swarm participation and bandwidth reallocation increases the number of TCP connections and reduces the contribution to the system as a whole gradually. This is not so encouraging.
3) The approach though is beneficial to the single users might turn the system more unfair and biased to high-capacity peers that operate with selfish motives.

VI. QUESTIONS

A. Briefly explain the most important piece selection policy used by BitTorrent to improve piece diversity.

BitTorrent Piece Selection operation is controlled by four different policies as explained.

- Random First Policy A peer initiates the download process by randomly selecting the pieces to download. The protocol operates in this stage until it has downloaded at least four pieces fully after which it employs Rarest First Policy.
- Rarest First Policy When the peer is not operating in neither the Random nor Game mode policy, it randomly selects the next piece to download from its local list of scarcely (rarest) available pieces (less available).
- End Game Mode Policy A peer pulls off the remaining pieces by operating as per this policy. If a peer has pending requests for blocks, it initiates requests for those blocks to be received to all its connected peers containing those blocks. Later, the cancel messages are sent to those peers that received the send requests earlier once for every received block.
- Strict Priority Policy - Once a block belonging to a specific piece is downloaded, this policy constraints a peer to prioritize the download of remaining blocks of that corresponding piece.

B. How this policy helps reduce download times?

The BitTorrent Piece Selection policy reduces the download rate in the following ways.

- The algorithm with its Rarest Pieces Policy increases the number of instances and diversity of the less available pieces and thereby reducing the long wait times and increasing the chances of parallel downloads.
- With its Strict Priority policy, peers always try to download the complete pieces instead of having incomplete ones and can, hence, fulfill the download requests of other connected peers with less delays.
- Also, with the parallel download requests, the peer experience less latency and downloads the pieces in a much faster rate.