Destabilizing BitTorrent's Clusters to Attack High Bandwidth Leechers

Florian Adamsky[1], Hassan Khan[2], Muttukrishnan Rajarajan[1], Syed Ali Khayam[2] and Rudolf Jäger[3]

[1]City University London, England
[2]National University of Sciences & Technology, Islamabad, Pakistan
[3]THM University of Applied Sciences, Friedberg, Germany

BitTorrent's Choking Algorithm

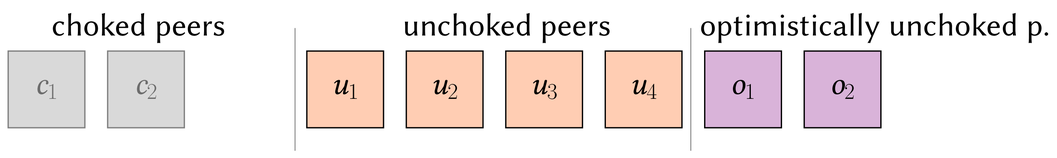
- Every leecher maintains two bits of state for every connection
  - choked/unchoked: no data to be sent to the other party until unchoked
  - interested/uninterested: the other party has data that a leecher wishes to acquire
- Data will be sent whenever one side is interested and other is unchoked
- A leecher unchokes its peers based on their upload rate
- Choking algorithm works in following way
  - The leecher orders the peers in its peer set according to their upload rate
  - The leecher then unchokes $n-1$ peers (n = number of parallel downloads allowed)
  - After every three executions of choking algorithm, a peer is chosen at random and optimistically unchoked
- The above algorithm is executed after every 30 seconds

Hypothesis

The current choking algorithm is greedy and it selects leechers by considering the upload bandwidth for the last round only. The greedy nature of BitTorrent choking algorithm can be exploited to break peer clustering

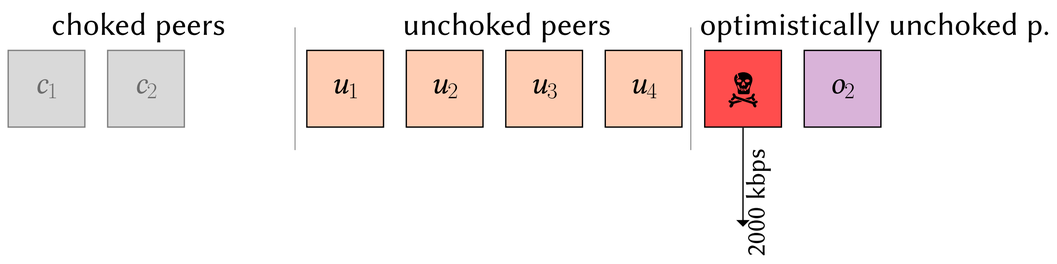
Attack - Step 1

- Introduce a malicious peer $L_{☠}$ with high bandwidth capabilities into the swarm
- $L_{☠}$ waits until a victim L_{av} optimistically unchokes $L_{☠}$



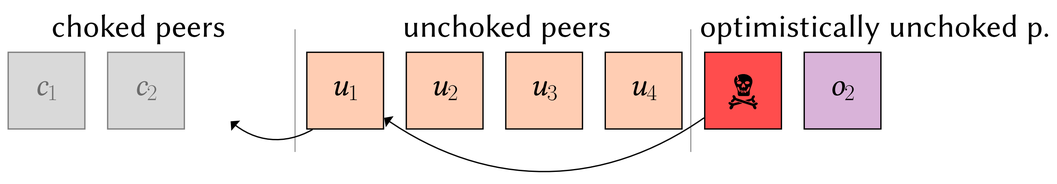
Attack - Step 2

- $L_{☠}$ provides high upload bandwidth for about ◷ 30 seconds



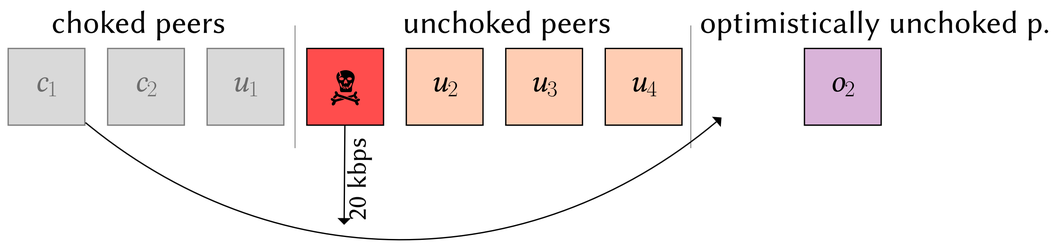
Attack - Step 3

- L_{av} adds $L_{☠}$ to its active peer set where $L_{☠}$ gets more time to upload



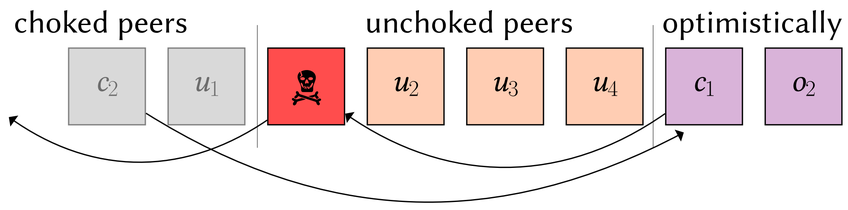
Attack - Step 4

- $L_{☠}$ drops its upload bandwidth significantly



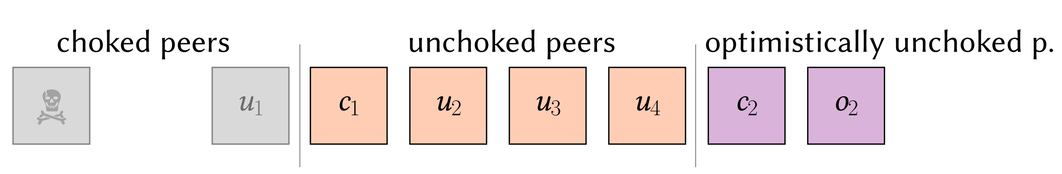
Attack - Step 5

- Chances are that $L_{☠}$ will remain on unchoked peer list for another round due to greedy nature of BitTorrent's choking algorithm.



Attack - Step 6

- L_{av} will try to get L_{u1} again, but this peer has found a new peer
- L_{av} has to take another peer, which might not be able to provide as efficient upload bandwidth as L_{u1}



Peer Affect of Proposed Attack on Peer Clustering

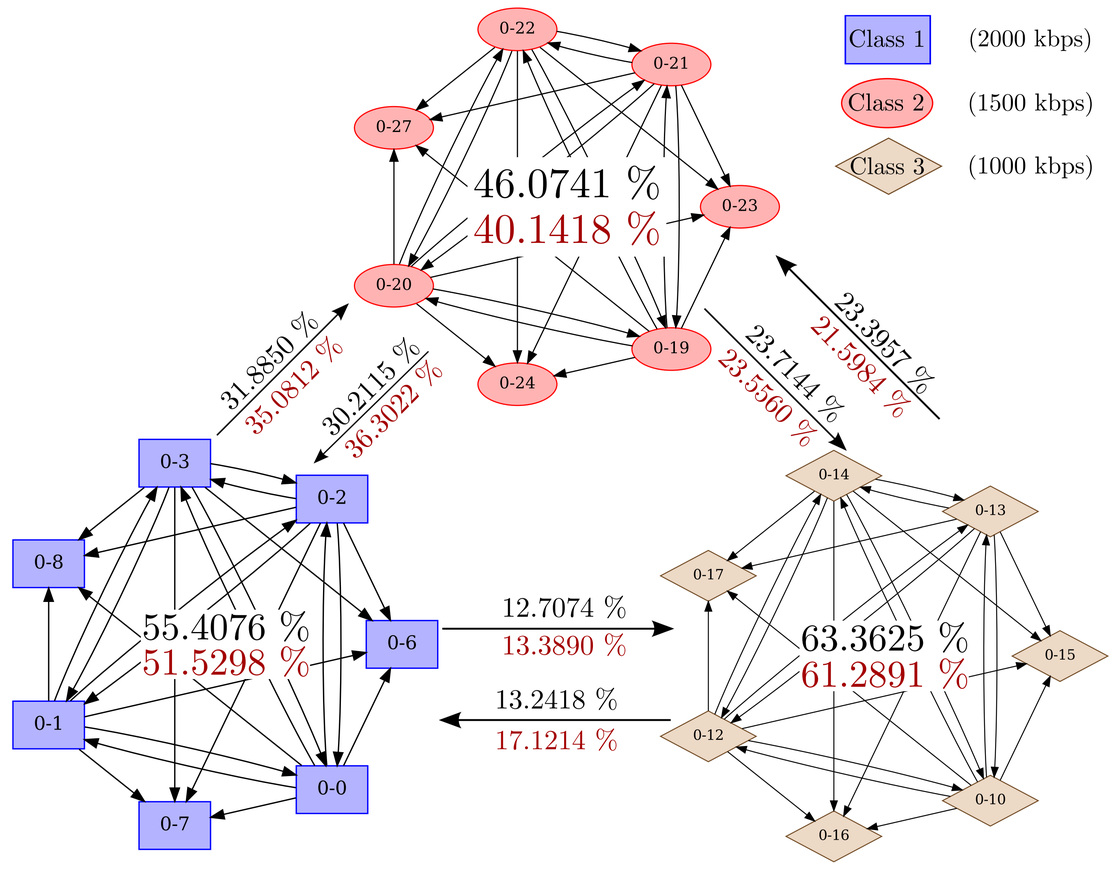


Figure: Affect of proposed attack on intra- and inter-cluster activity. The proposed attack increases inter-cluster activity and decreases intra-cluster activity and consequently wastes valuable bandwidth

Arrows show the direction of data transfer. Seeders are omitted for simplicity. Results without malicious peer are given in black & results with 1 malicious peer are given in red.

Affect of Proposed Attack on Download Times

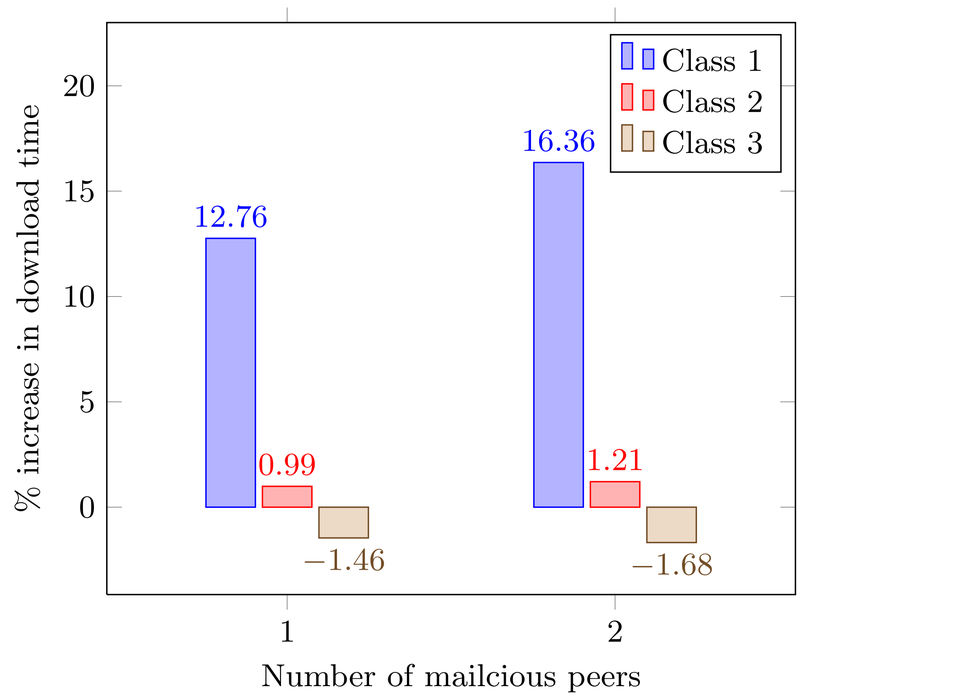


Figure: Increase/decrease in download times of different peer classes when malicious peers are introduced. It should be noted that the high bandwidth peers suffers the most

Class 1 - 2000 Kbps; Class 2 - 1500 Kbps; Class 3 - 1000 Kbps