Do incentives build robustness in BitTorrent?

Ronghui Gu
ronghui.gu@yale.edu

Agenda

- Introduction
- BitTorrent Overview
- Modeling Altruism in BitTorrent
- Building BitTyrant
- Evaluation
- Conclusion

Introduction

MAIN IDEA

- Free-Ride
  - Consuming resources without contribution
  - Fundamental problem in P2P systems
- BitTorrent
  - Use “Tit-for-Tat” strategy for discouraging free-riders
  - Upload more → download more
- Question
  - Can we cheat?
  - Download without upload or upload less

Introduction

CONTRIBUTION

- Contribution
  - Shows BitTorrent is not robust with strategic users
  - Model altruism in BitTorrent
    - Upload more than necessary
- BitTyrant
  - A selfish and strategic BitTorrent client
  - Carefully select peers and contribution rates
  - Raise the download speed with the same contribution

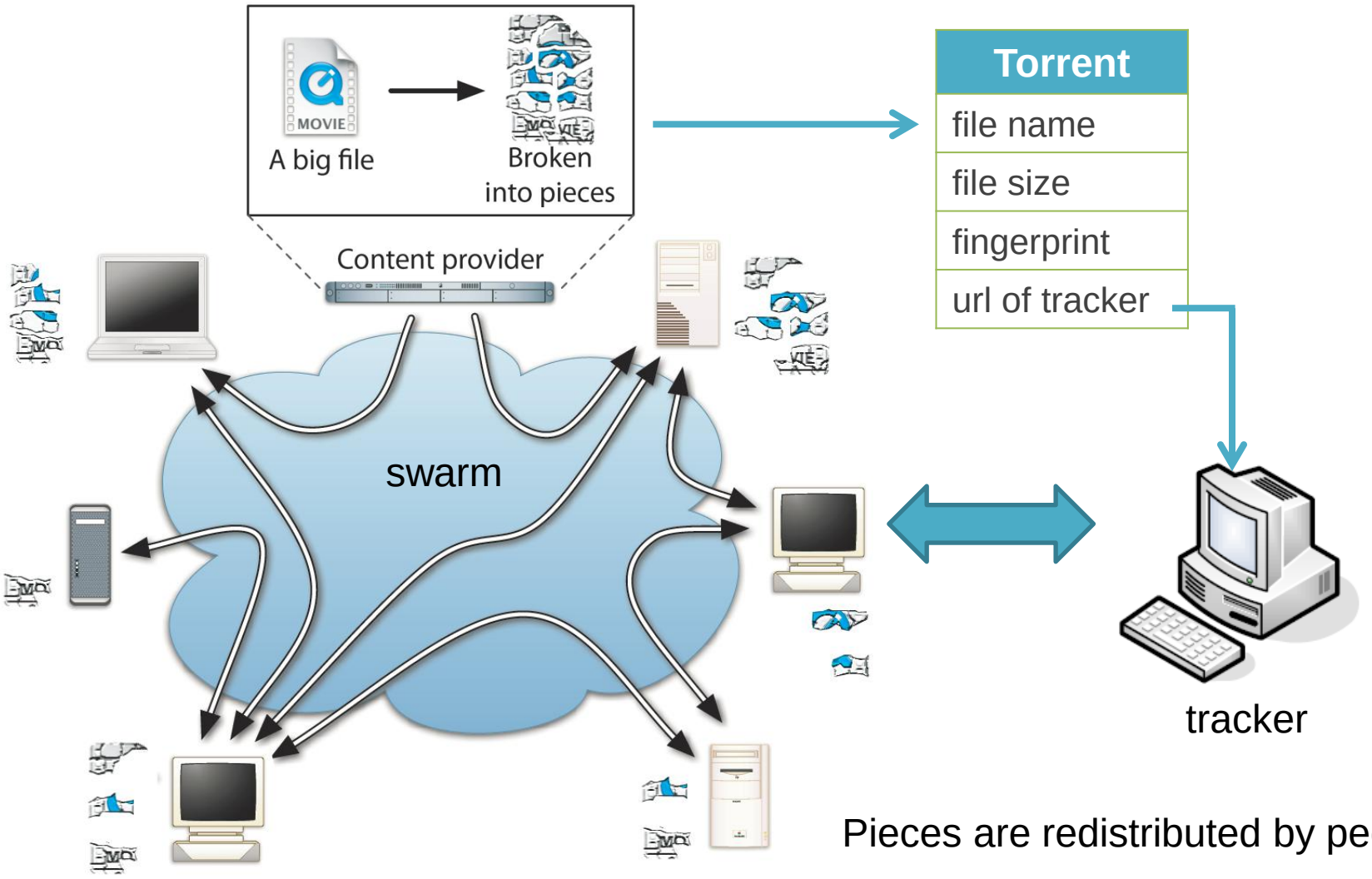
BitTorrent Overview

WHAT IS IT?

- A P2P file sharing protocol
  - Bulk data transfer
  - Account for 40%~70% of internet traffic (Feb.2009)
- True P2P: no single server
- Tracker: keep track of active peers in the swarm
- Swarm: all peers sharing a torrent
- Seeds: users with a complete file
- ……

BitTorrent Overview

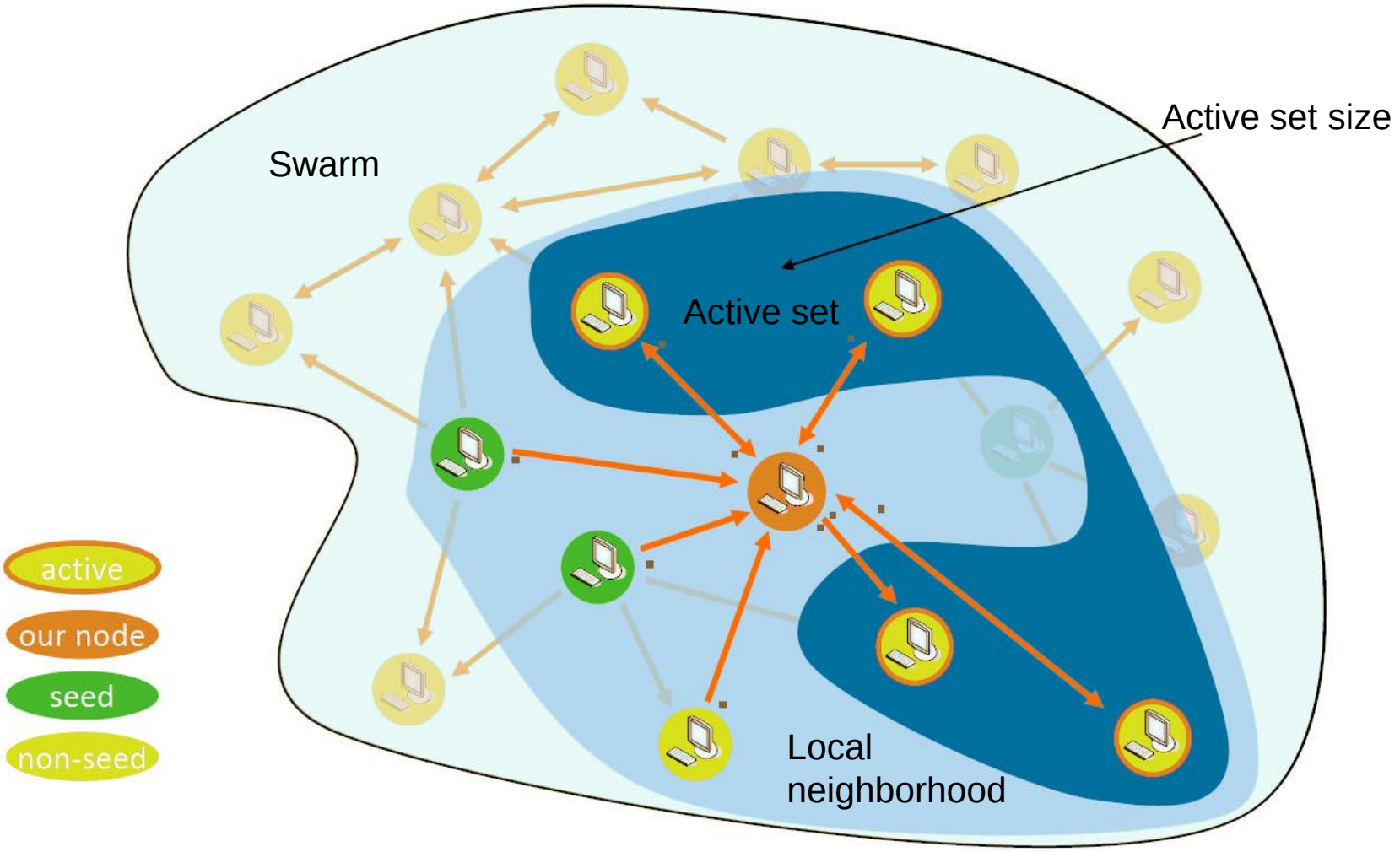
How does it work(1)?



Pieces are redistributed by peers

BitTorrent Overview

How does it work(2)?



BitTorrent Overview

THE STRATEGY!

- Tit-for-Tat strategy
  - Famous in game theory (Prisoners' Dilemma)
  - Do what others did to him in the last round
  - Forgiveness: cooperate with a few lucky guys
- In the BitTorrent context
  - Grant upload capacity to

    n best uploaders + ω optimistically unchoked peers
  - Active set size = n
  - Equal split rate = upload capacity / (n + ω) ?
    - Match same rate and difficult to be stable
  - Choke (stop uploading to) peers that perform badly

BitTorrent Overview

Sounds Great?

- A fast client with 90 total upload capacity
- Will choose top 2 uploaders and 1 unchoker

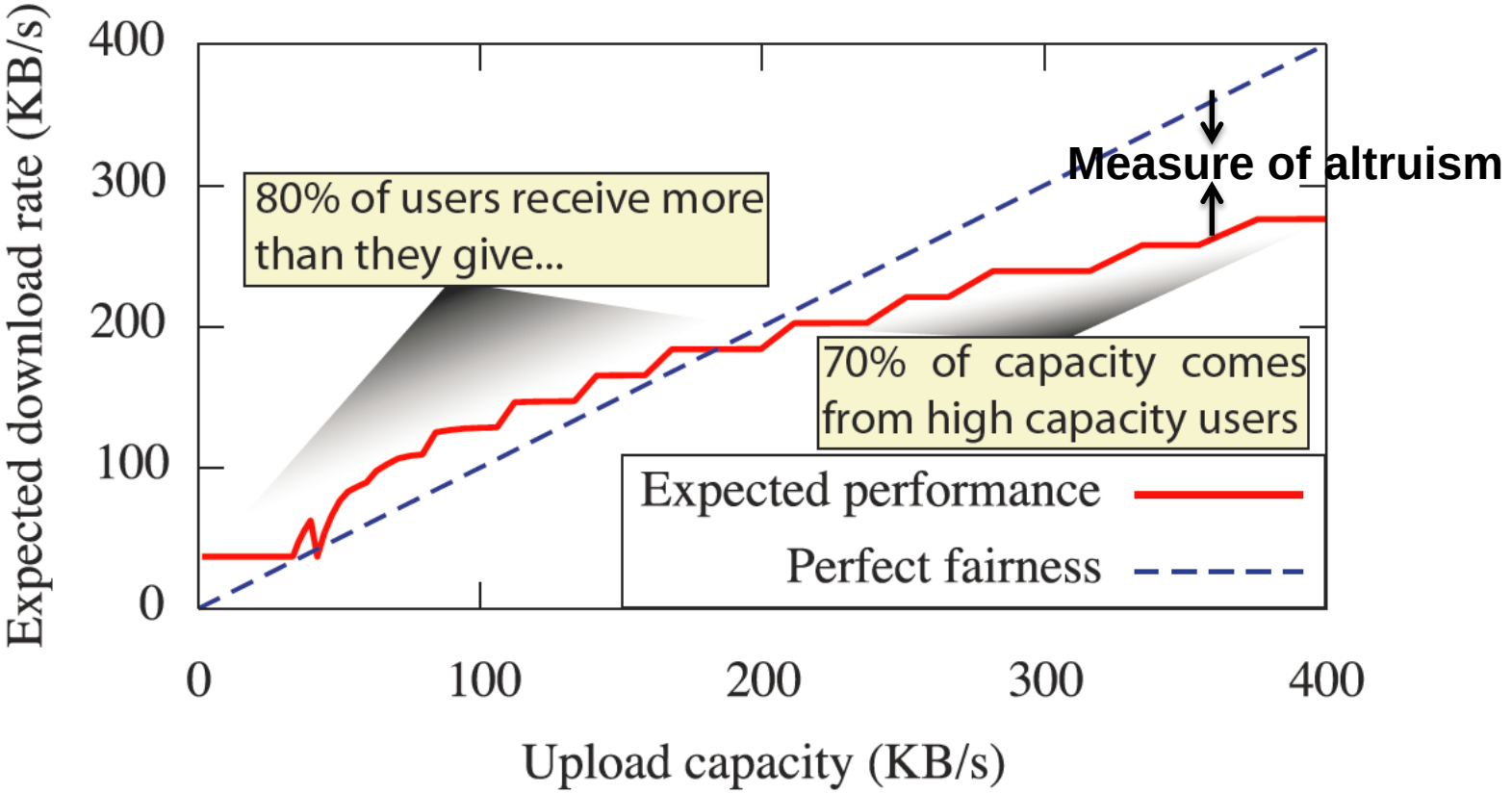
	peer	*received rate*	*sent rate*
Top		20	30
Top		10	30
		9	0
Unchoker		1 (capacity)	30

- For machine has LOTS of upload
  - Most peers are slower, even top ones
  - Pay much more than get
- For slow machines
  - Have no chance to be top
  - However, could get welfare
  - Waste all the upload capacity
- For all peers in the active set
  - Get the same reciprocation
  - sent rate = equal split rate
  - Why not just send at rate 10?
- For the 3rd one
  - He will get paid if he sent 1 more

Altruism

Modeling Altruism in BitTorrent

OBSERVATION (1)

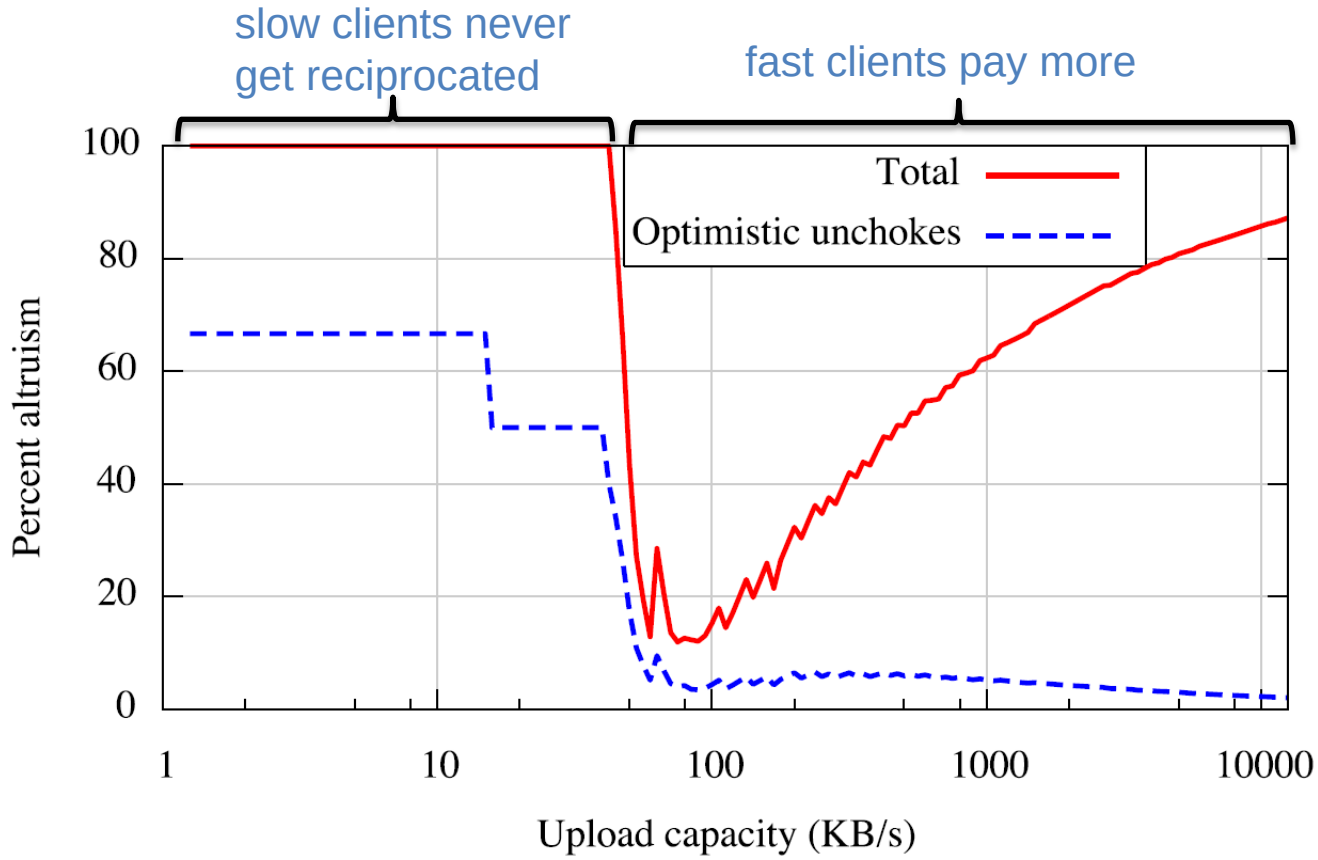


- The sub-liner growth suggests the unfairness (high capacity)

Modeling Altruism in BitTorrent

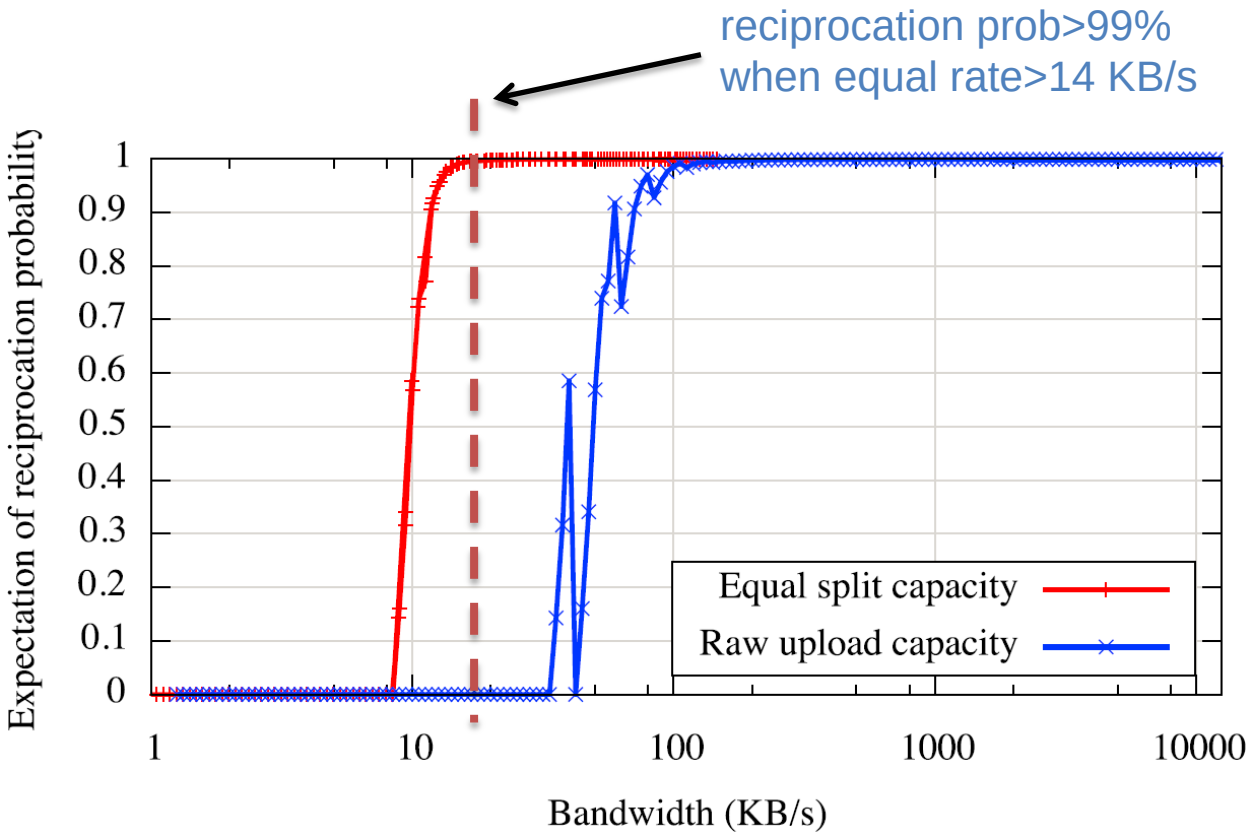
OBSERVATION (2)

- Altruism: *any* upload contribution that can be withdrawn without loss in download performance



Modeling Altruism in BitTorrent

OBSERVATION (3)



- Reciprocation probability as a function of equal split rate
- The sharp jumps due to the increase of active set size $\lfloor\sqrt{0.6r}\rfloor - \omega$

Building BitTyrant

A SELFISH CLIENT!

- Based on Azureus Client :
  - Most popular in traces
- Main idea
  - Exploit unfairness and minimize altruism
  - Dynamically choose *how many* and *which* peers to send data
- Mechanisms
  - Choose “best” peers
  - Deviate from equal split

Building BitTyrant

ALGORITHM

- Maintain d_p and u_p of peer p
  - d_p: download performance from p
  - u_p: rate to earn reciprocation from p
- Algorithm
  - Rank peers by the ratio d_p/u_p
  - Select top ones until reach the upload capacity

$$\underbrace{\frac{d_0}{u_0}, \frac{d_1}{u_1}, \frac{d_2}{u_2}, \frac{d_3}{u_3}, \frac{d_4}{u_4}}_{\text{choose } k \mid \sum_{i=0}^{k} u_i \leq cap}, \cdots$$

Building BitTyrant

EXAMPLE

- A BitTyrant client with 21 total upload capacity

peer	*received rate*	*required send rate*	*benefit/cost ratio*
	9	3	3.00
	20	10	2.00
	10	8	1.25
	1	1.6	0.625

Sum=21

Building BitTyrant

SOME PROBLEM?

- Similar to knapsack problem
  - It's a greedy algorithm, not the best
  - Maybe dynamic programming method is better

peer	*received rate*	*required send rate*	*benefit/cost ratio*
	20	10	2.00
	15	8	1.88
	12	8	1.5

A client with 17 total upload capacity

- Greedy: 20
- Dynamic: 27

Building BitTyrant

REASONS?

- Faster in large scale system
- More robust to
  - Estimation error
  - Churn and other network conditions
- Even they're true
  - Still could be improved

Building BitTyrant

ESTIMATION?

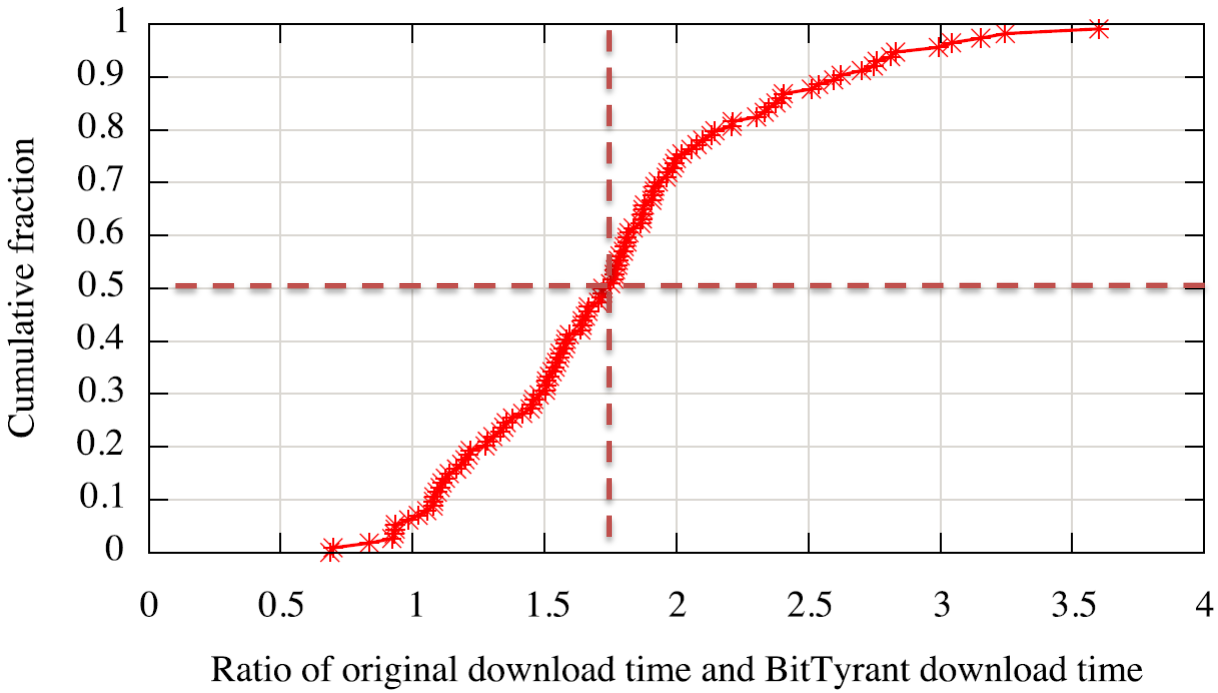
- Initialization
  - According to the bandwidth distribution
- After each round
  - If peer p not unchoke us: $u_p \leftarrow (1+\delta)u_p$
  - If peer p unchoke us: $d_p \leftarrow$ observed rate
  - If peer p unchoke us for the last r rounds:

$$u_p \leftarrow (1-\gamma)u_p$$

Evaluation

SINGLE AND MULTIPLE USERS

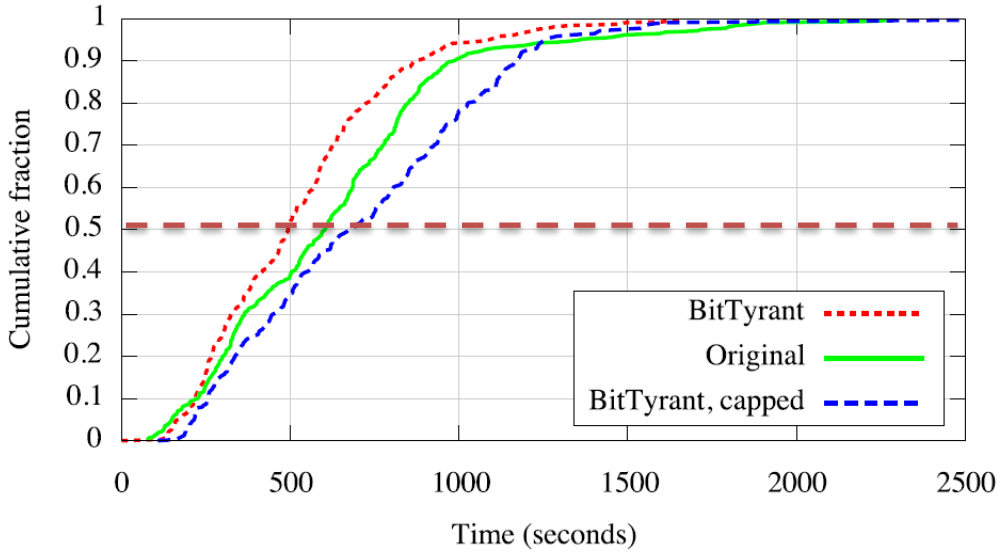
- Single BitTyrant user
  - The CDF of the ratio of download time
  - The median of performance is a factor of 1.72



Evaluation

SINGLE AND MULTIPLE USERS

- Multiple BitTyrant users
  - Strategic: use BitTyrant and contribute excess capacity
    - The performance will be improved
  - Strategic & selfish: doesn't give back excess capacity
    - The performance decreases dramatically



Conclusion

WHAT THE PAPER HAS DONE

- Shows BitTorrent is not robust with strategic users
- Model altruism in BitTorrent
- BitTyrant
  - Exploit altruism in BitTorrent
  - The performance of a client is improved

THANKS FOR YOUR TIME

QUESTIONS?

Appendix

IMPROVEMENT!

- Improved algorithm
  - Select top ones until exceed the upload capacity
  - Suppose there are n peers in the active set
    - Allocate u_i to peer i, where $i < n$
    - Allocate the rest to peer n