Efficient and Highly Available Peer Discovery: A Case for Independent Trackers and Gossiping

György Dán Ilias Chatzidrossos
Royal Institute of Technology (KTH)
Stockholm, Sweden

Niklas Carlsson
Linköping University
Linköping, Sweden

Proc. IEEE P2P, Kyoto, Japan, Aug/Sept. 2011

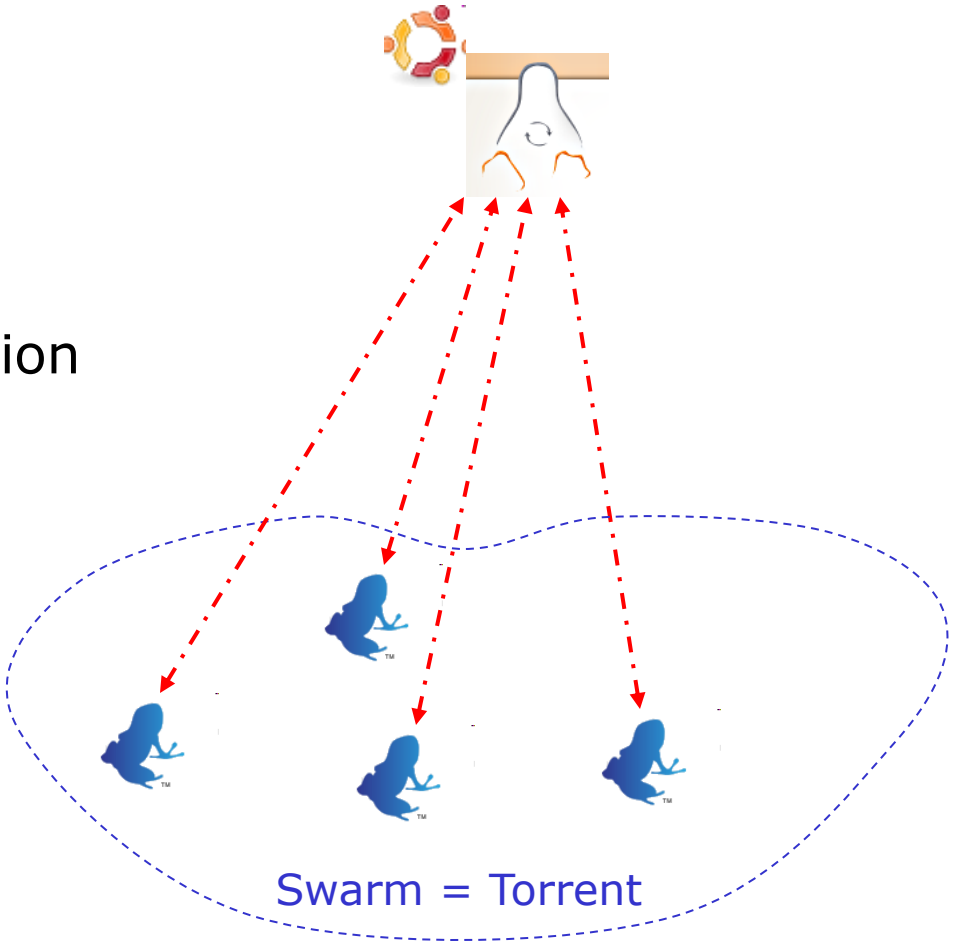
Background

BitTorrent

- Arguably biggest source of p2p traffic
- Contents split into many small pieces
  - Pieces are downloaded from both leechers and seeds
- Distribution paths are dynamically determined
  - Based on data availability
- At least one overlay per content

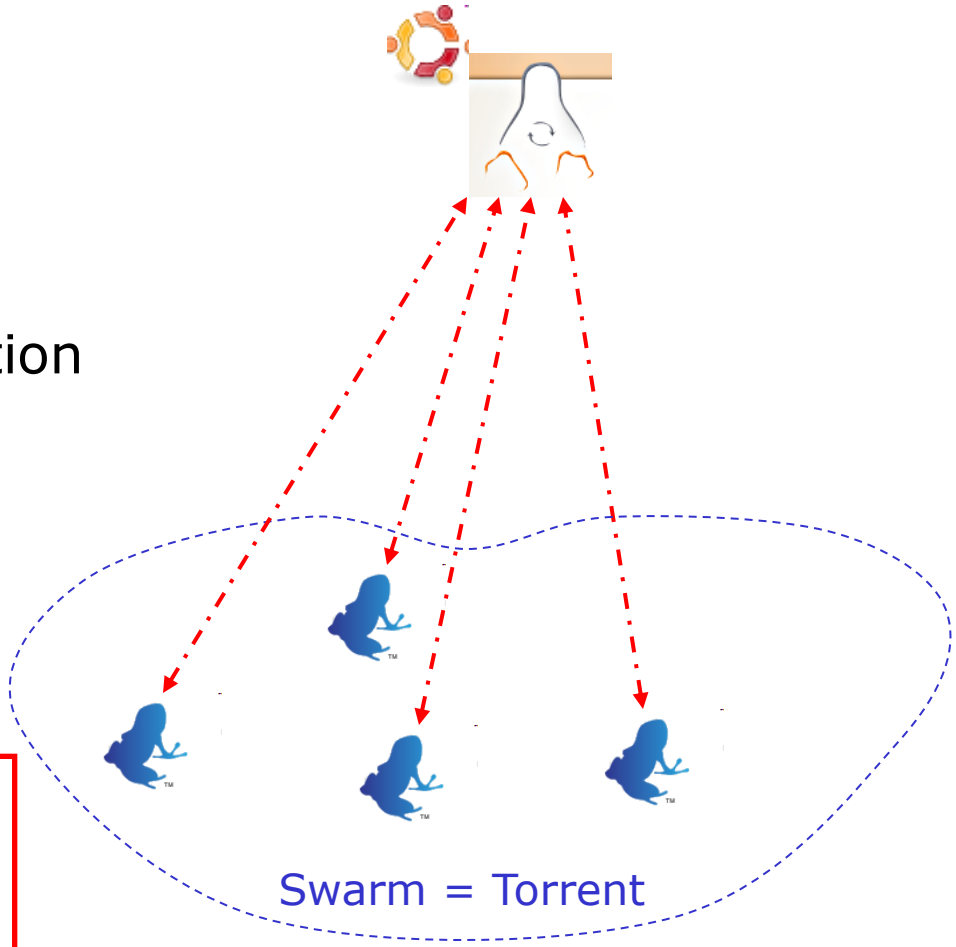
Background
Peer discovery in BitTorrent

- Torrent file
  - "announce" URL
- Tracker
  - Register torrent file
  - Maintain state information
- Peers
  - Obtain torrent file
  - Announce
  - Report status
  - Peer exchange (PEX)
- Issues
  - Central point of failure
  - Tracker load



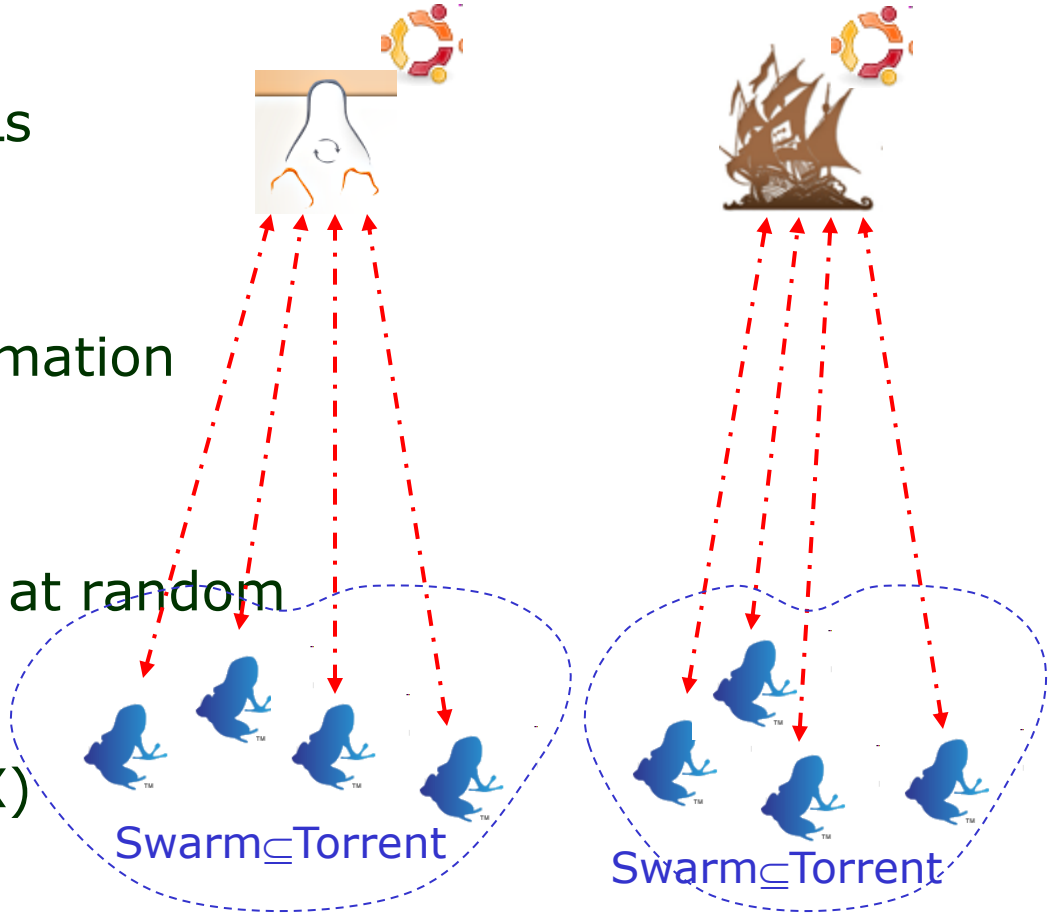
Background
Peer discovery in BitTorrent

- Torrent file
  - "announce" URL
- Tracker
  - Register torrent file
  - Maintain state information
- Peers
  - Obtain torrent file
  - Announce
  - Report status
  - Peer exchange (PEX)
- Issues
  - Central point of failure
  - Tracker load



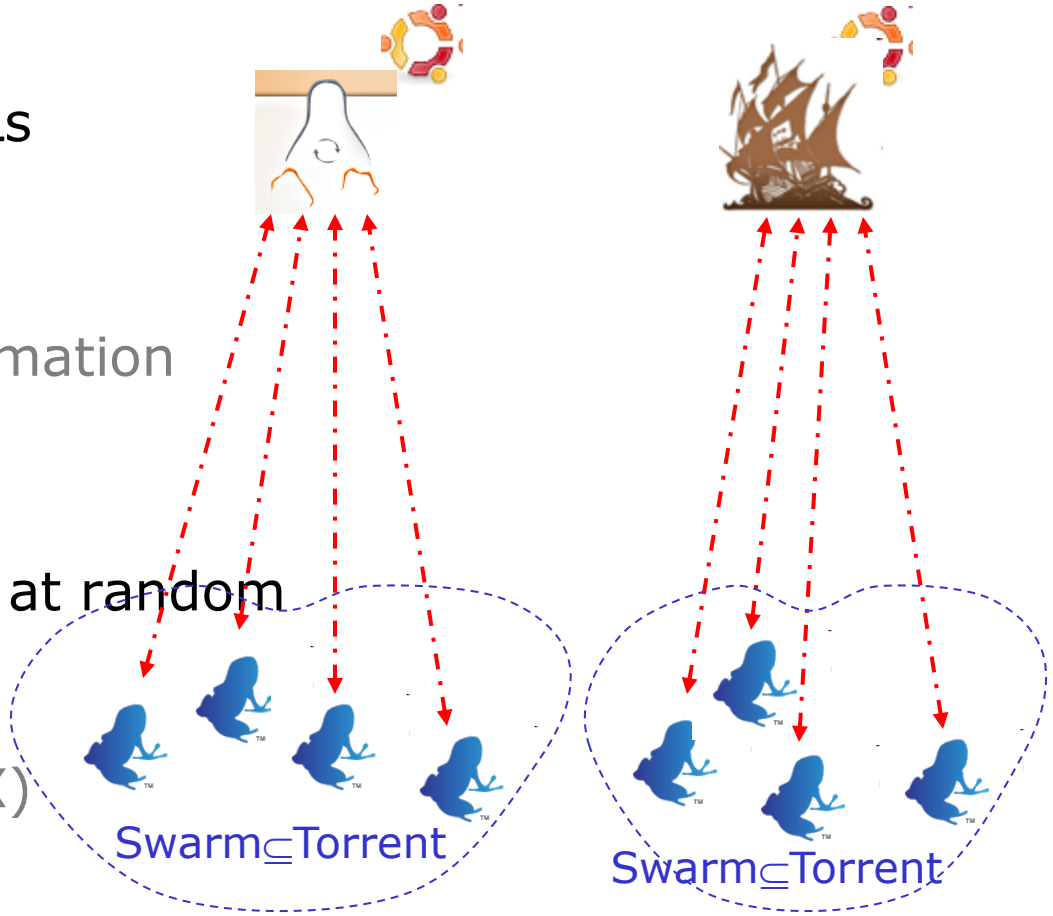
Background
Multi-tracked torrents

- Torrent file
  - "announce-list" URLs
- Trackers
  - Register torrent file
  - Maintain state information
- Peers
  - Obtain torrent file
  - Choose one tracker at random
  - Announce
  - Report status
  - Peer exchange (PEX)
- Issue
  - Multiple smaller swarms



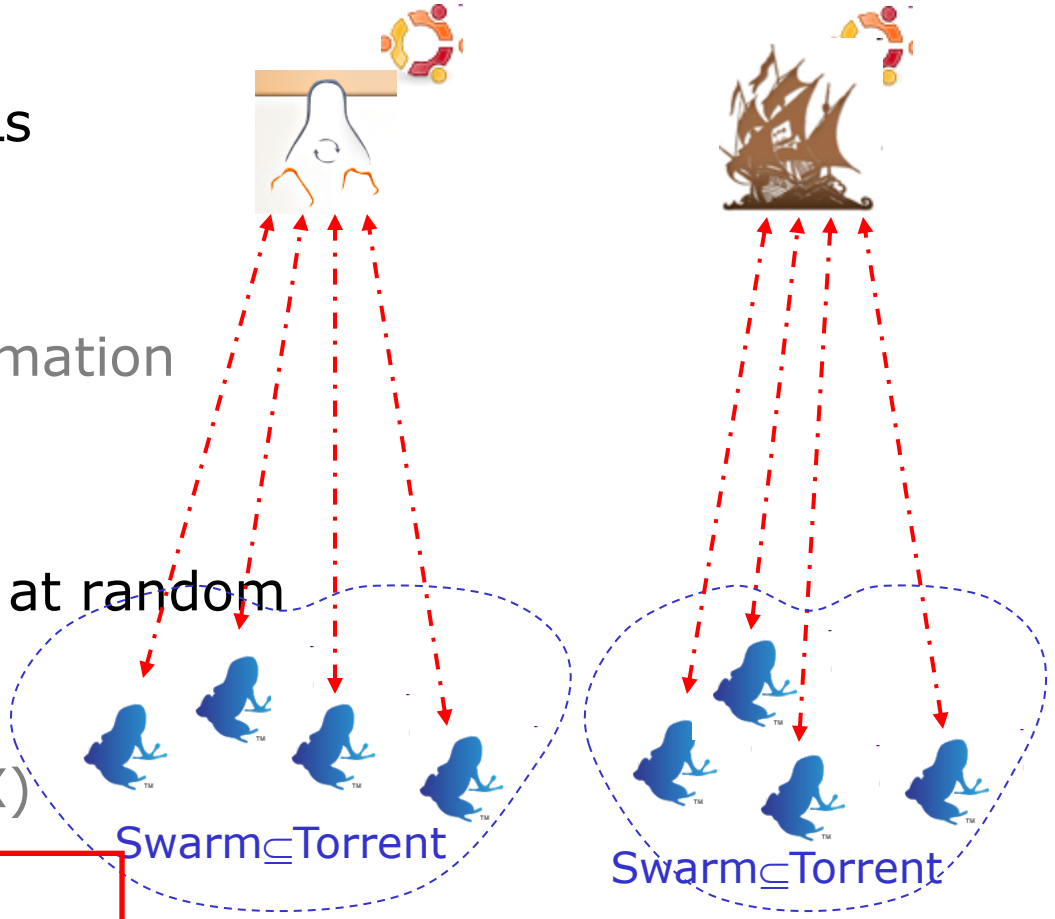
Background
Multi-tracked torrents

- Torrent file
  - "announce-list" URLs
- Trackers
  - Register torrent file
  - Maintain state information
- Peers
  - Obtain torrent file
  - Choose one tracker at random
  - Announce
  - Report status
  - Peer exchange (PEX)
- Issue
  - Multiple smaller swarms



Background
Multi-tracked torrents

- Torrent file
  - "announce-list" URLs
- Trackers
  - Register torrent file
  - Maintain state information
- Peers
  - Obtain torrent file
  - Choose one tracker at random
  - Announce
  - Report status
  - Peer exchange (PEX)
- Issue
  - Multiple smaller swarms

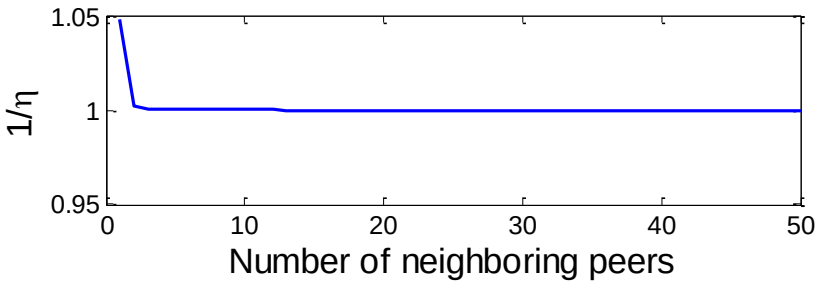


Scalable … Why an issue??

BitTorrent efficiency vs. swarm size

Early analytical model

$$\eta \approx 1 - \left(\frac{\log N}{N} \right)^k$$

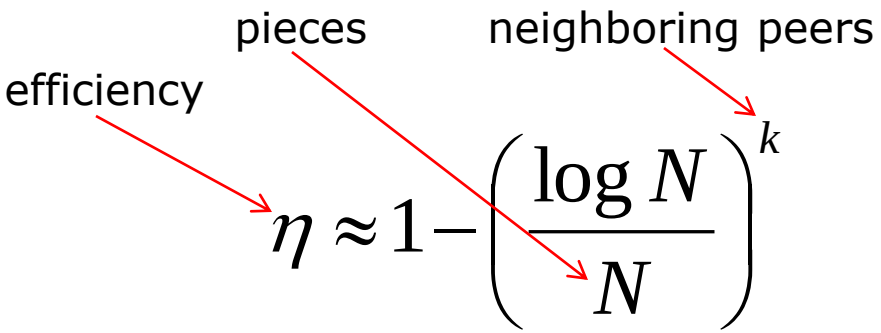


D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", Proc. ACM SIGCOMM, 2004

Scalable … Why an issue??

BitTorrent efficiency vs. swarm size

Early analytical model

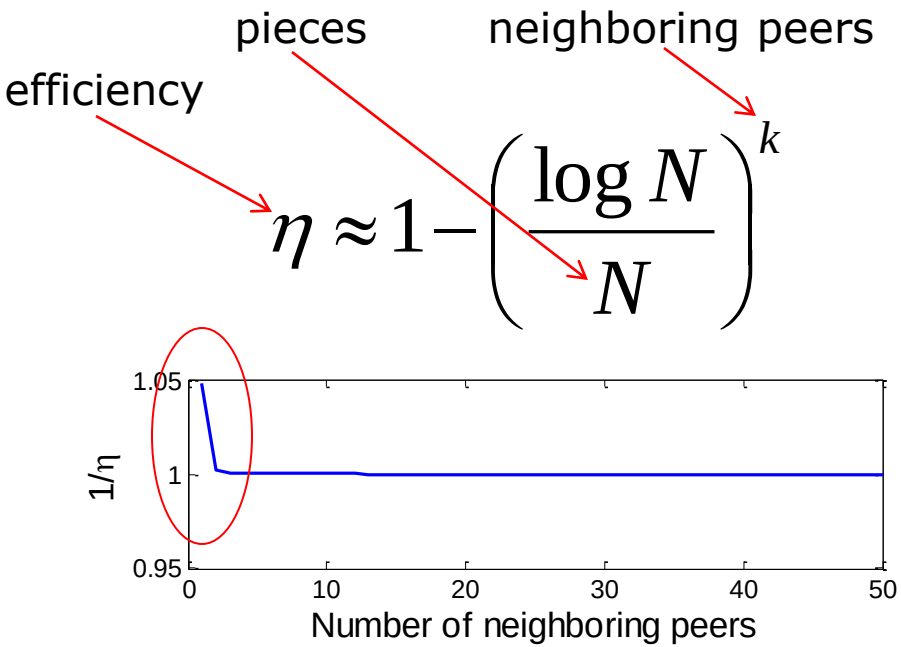


D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", Proc. ACM SIGCOMM, 2004

Scalable … Why an issue??

BitTorrent efficiency vs. swarm size

Early analytical model



D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", Proc. ACM SIGCOMM, 2004

Scalable … Why an issue??

BitTorrent efficiency vs. swarm size

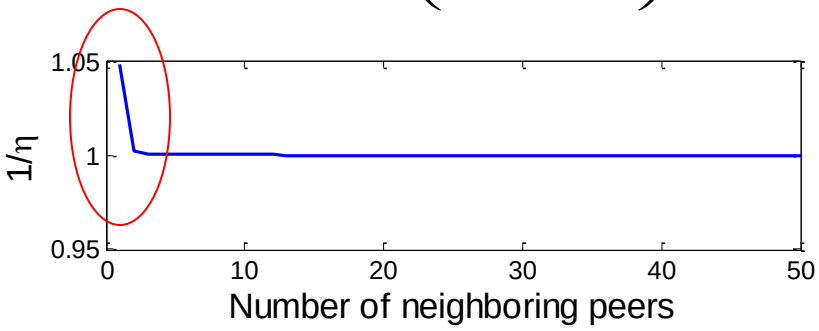
Early analytical model

pieces

neighboring peers

efficiency

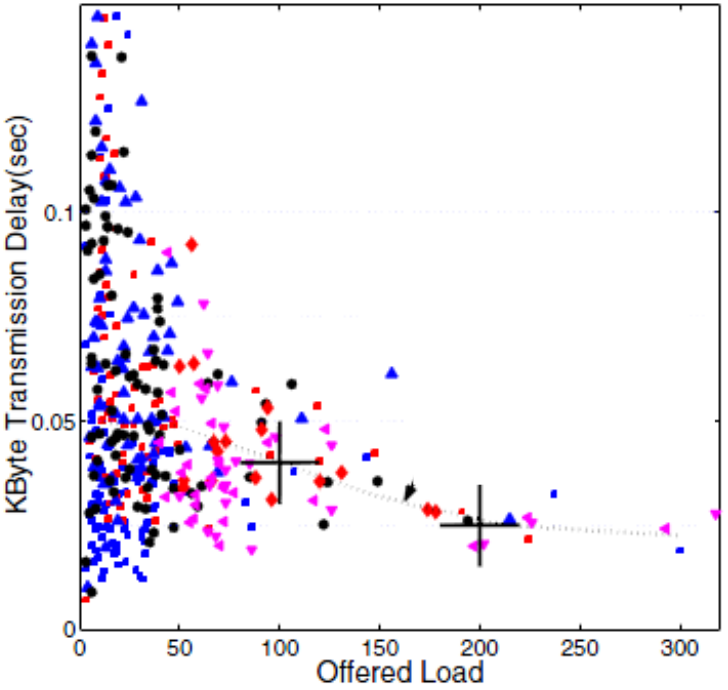
$$\eta \approx 1 - \left(\frac{\log N}{N} \right)^{k}$$



D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", Proc. ACM SIGCOMM, 2004

Early measurements

Measured time to transmit 1KB, based on 500 torrents



X. Yang, G. de Veciana,"Service Capacity of Peer to Peer Networks," Proc. IEEE INFOCOM 2004

Scalable … Why an issue??

BitTorrent efficiency vs. swarm size

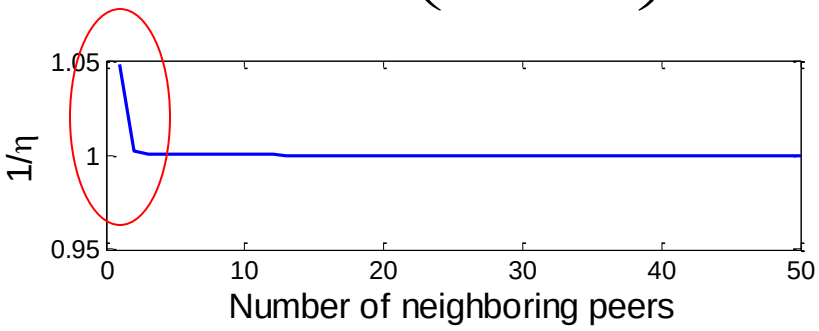
Early analytical model

pieces

neighboring peers

efficiency

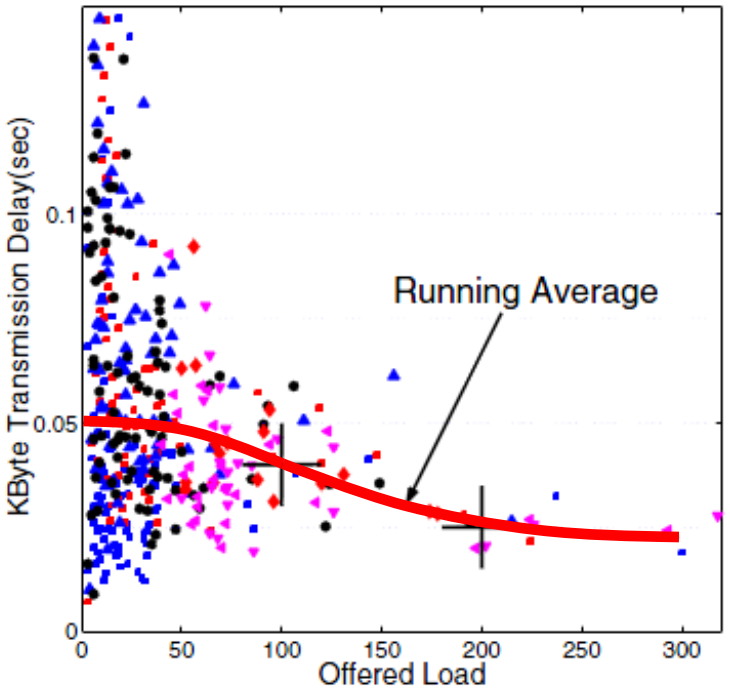
$$\eta \approx 1 - \left(\frac{\log N}{N} \right)^{k}$$



D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", Proc. ACM SIGCOMM, 2004

Early measurements

Measured time to transmit 1KB, based on 500 torrents

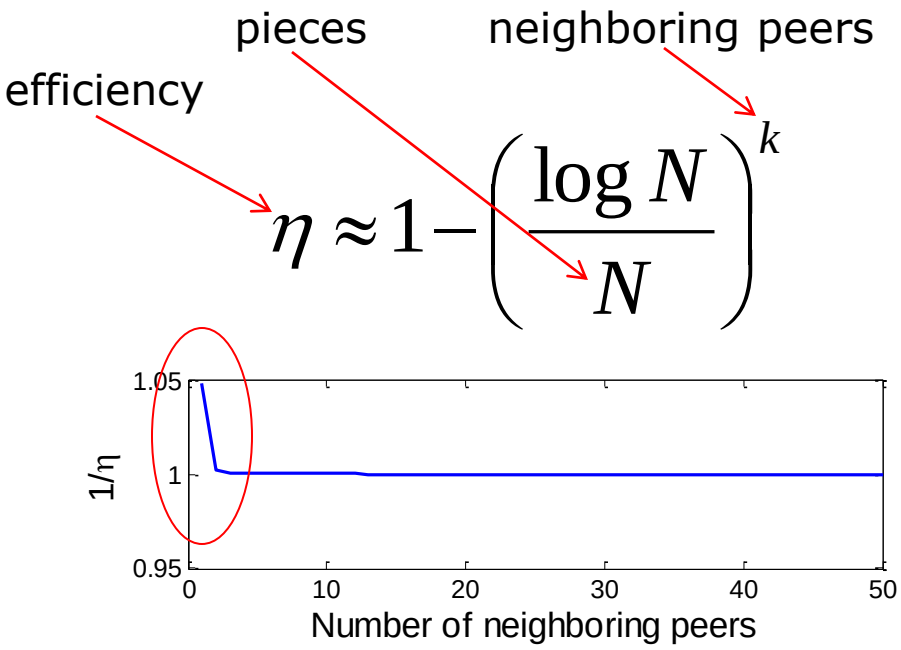


X. Yang, G. de Veciana,"Service Capacity of Peer to Peer Networks," Proc. IEEE INFOCOM 2004

Scalable … Why an issue??

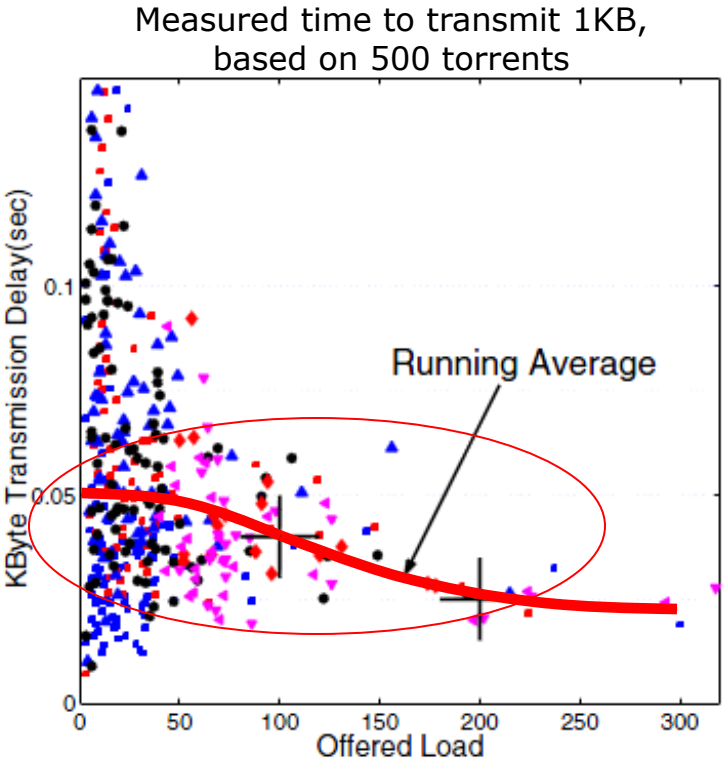
BitTorrent efficiency vs. swarm size

Early analytical model



D. Qiu, R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks", Proc. ACM SIGCOMM, 2004

Early measurements



X. Yang, G. de Veciana,"Service Capacity of Peer to Peer Networks," Proc. IEEE INFOCOM 2004

Measurements
Two basic datasets

- Screen scrapes of *www.mininova.org*
  - Popular torrent search engine
  - 1,690 trackers (721 unique)

- Tracker scrapes of known trackers (Oct. 10-17, 2008)
  - 2.86 million unique torrents
  - Roughly 20-60 M concurrent peers (depending on day)
  - 330,000 swarms overlap with screen scrape

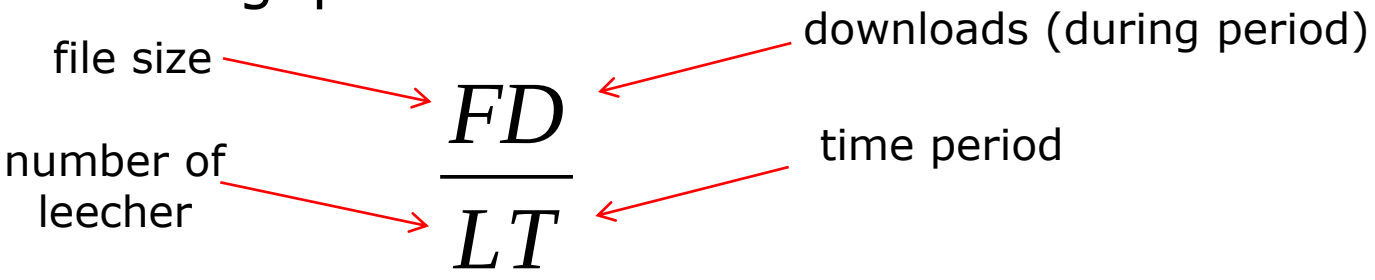
Throughput vs. swarm size

- Throughput estimation

$$\frac{FD}{LT}$$

Throughput vs. swarm size

- Throughput estimation



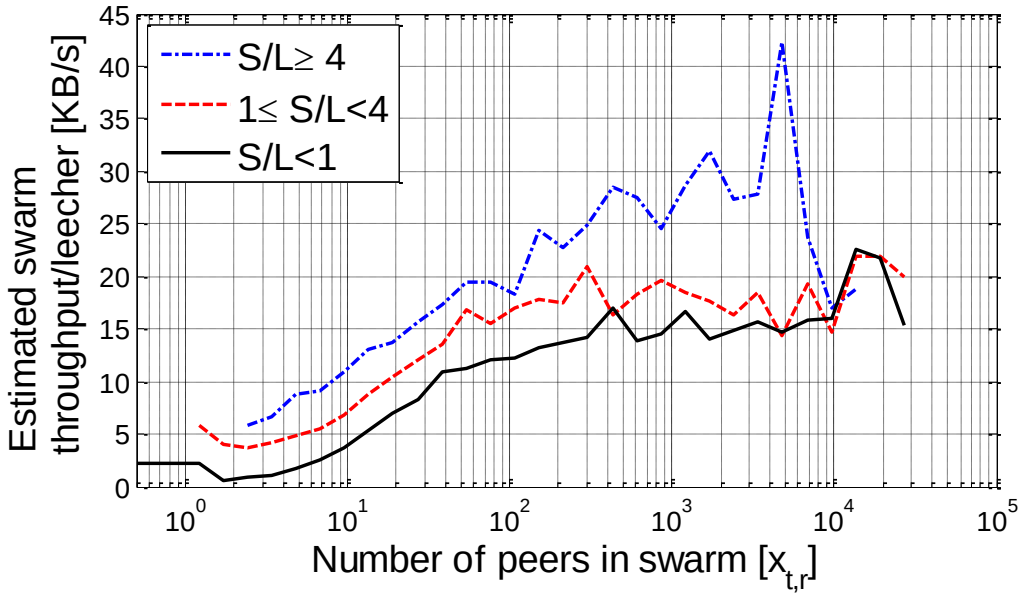
Throughput vs. swarm size

- Throughput estimation

file size → FD ← downloads (during period)

number of leecher → LT ← time period

$$\frac{FD}{LT}$$



0700 UTC 11-12.Oct.2008

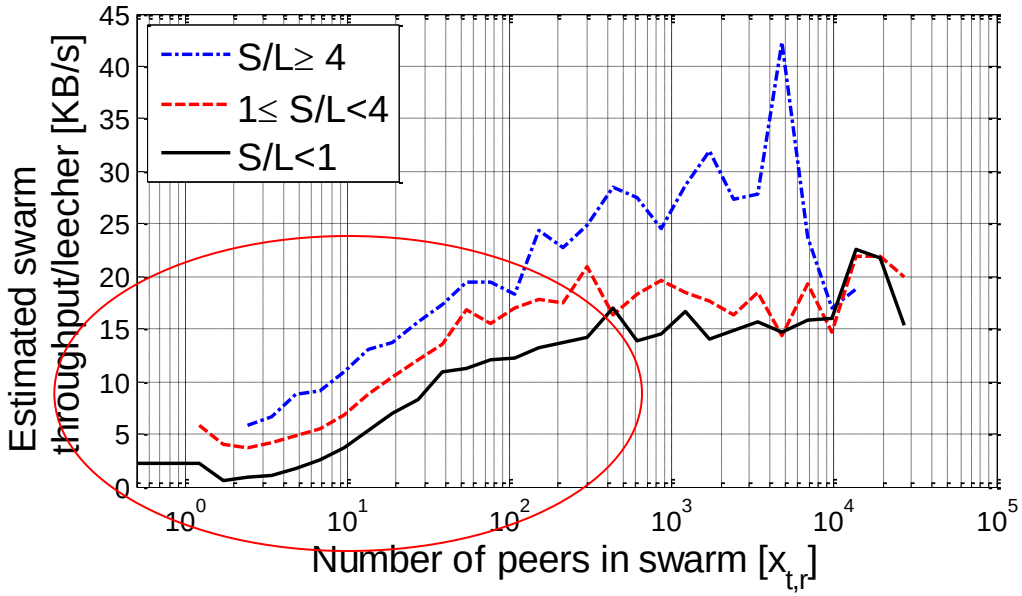
Throughput vs. swarm size

- Throughput estimation

file size → FD ← downloads (during period)

number of leecher → LT ← time period

$$\frac{FD}{LT}$$



The performance of small swarms is worse

Dynamic Swarm Management
Improving BitTorrent performance

- Trade-off in multi-tracking
  - Load sharing and increased availability
  - Smaller swarm sizes $\Rightarrow$ lower throughput
- Goals of dynamic swarm management
  - Efficient peer discovery
    - Avoid swarm partitioning (performance penalty)
  - High availability
    - Independent trackers
    - Load balancing (for large torrents)
  - Small overhead
    - Management traffic (at trackers and peers)

Candidate approaches

- Tracker-based protocol
  - Requires trackers to be modified (e.g., DSM)

G.Dán, N.Carlsson, "Dynamic Swarm Management for Improved BitTorrent Performance", Proc. of IPTPS 2009

- Torrent-wide DHT
  - Consistency and stale routing tables under churn
  - Overhead

- Peer-based protocols
  - Independent trackers and gossiping
  - Transparent to the trackers
  - Constant overhead independent of torrent size

Candidate approaches

- Tracker-based protocol
  - Requires trackers to be modified (e.g., DSM)

G.Dán, N.Carlsson, "Dynamic Swarm Management for Improved BitTorrent Performance", Proc. of IPTPS 2009

- Torrent-wide DHT
  - Consistency and stale routing tables under churn
  - Overhead

- Peer-based protocols
  - Independent trackers and gossiping
  - Transparent to the trackers
  - Constant overhead independent of torrent size

What have we learned so far?

- Good peer discovery mechanisms important
  - Small torrents bad ...
- Centralized peer discovery (single central tracker)
  - Single point of failure
  - No load balancing opportunities
- Multi-tracker approach
  - Connect with all trackers => High overhead
  - Connect with one tracker => Disjoint sets (smaller swarms)

Main question addressed

Is possible to achieve highly available and efficient peer-discovery, which avoids the formation of disjoint swarms, at low overhead by employing independent trackers and relying only on a gossip protocol?

Two protocols

- Random Peer Migration (RPM)
- Random Multi-Tracking (RMT)

Randomized Peer Migration (RPM)

- Slightly Modified BitTorrent peer behavior
- Component 1: Peer migration
  - Randomly chosen peer changes swarm
  - Intensity of migration (β) [non trivial]
- Component 2: Peer EXchange Protocol (PEX)
  - Peers exchange neighborhood info using gossiping

Random Multi-Tracking (RMT)

- Slightly Modified BitTorrent peer behavior
- Component 1: Multi-tracked Peers
  - Random arriving peer connects to *k* trackers
  - Intensity of multi-tracking (β) [non trivial]
- Component 2: Peer EXchange Protocol (PEX)
  - Multi-tracked peers exchange neighborhood info using gossiping

Random Multi-Tracking (RMT)

- Slightly Modified BitTorrent peer behavior
- Component 1: Multi-tracked Peers
  - Random arriving peer connects to *k* trackers
  - Intensity of multi-tracking (β) [non trivial]
- Component 2: Peer EXchange Protocol (PEX)
  - Multi-tracked peers exchange neighborhood info using gossiping

Peer migration (using RPM)

- How to pick a good migration rule??

Peer migration (using RPM)

- How to pick a good migration rule??

Migration probability

Make choice after downloaded of the file

Peer migration (using RPM)

- How to pick a good migration rule??

Migration probability $\frac{1}{x_r}$

Make choice after downloaded $\frac{1}{\beta(|R(t)|-1)}$ of the file

Parameter: scrape intensity (overhead) ~ β

Peer migration (using RPM)

- How to pick a good migration rule??
- Rate out of a swarm r

Migration probability $\frac{1}{x_r}$

Make choice after downloaded $\frac{1}{\beta(|R(t)|-1)}$ of the file

Parameter: scrape intensity (overhead) ~ β

Peer migration (using RPM)

- How to pick a good migration rule??
- Rate out of a swarm r

Migration probability $\frac{1}{x_r}$

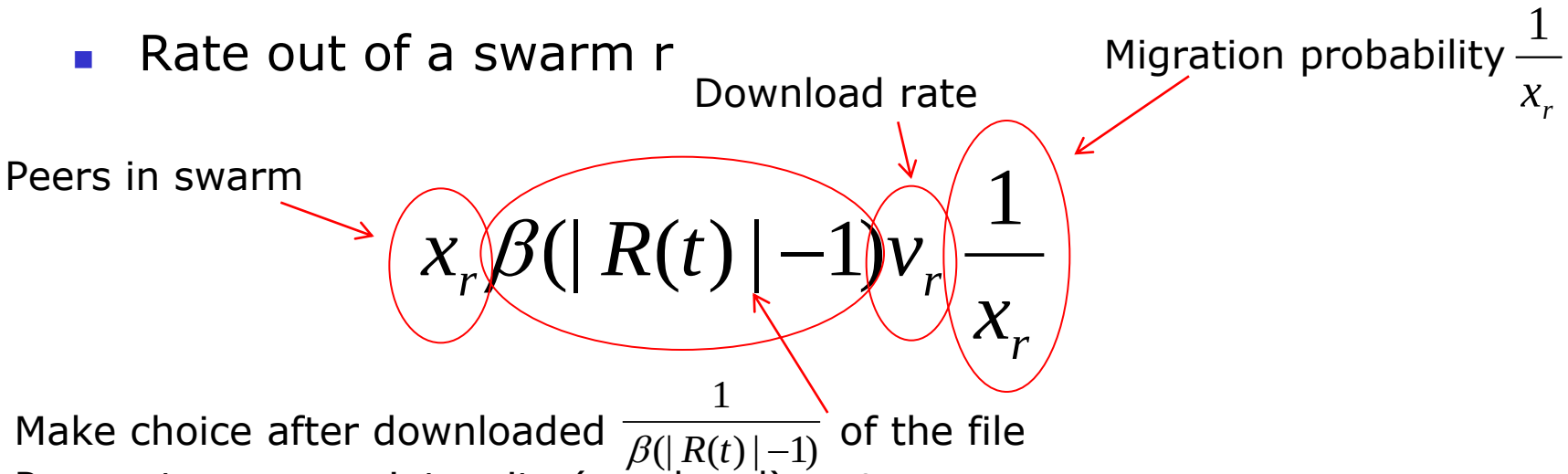
$$x_r \beta(|R(t)|-1) v_r \frac{1}{x_r}$$

Make choice after downloaded $\frac{1}{\beta(|R(t)|-1)}$ of the file

Parameter: scrape intensity (overhead) ~ β

Peer migration (using RPM)

- How to pick a good migration rule??
- Rate out of a swarm r



Peer migration (using RPM)

- How to pick a good migration rule??

- Rate out of a swarm r

Migration probability $\frac{1}{x_r}$

Download rate

Peers in swarm

$$x_r \beta(|R(t)|-1)v_r \frac{1}{x_r}$$

Make choice after downloaded $\frac{1}{\beta(|R(t)|-1)}$ of the file

Parameter: scrape intensity (overhead) ~ β

Note: Independent of swarms size

Peer migration (using RPM)

- How to pick a good migration rule??

- Rate out of a swarm r

Migration probability $\frac{1}{x_r}$

Download rate

Peers in swarm

$$x_r \beta(|R(t)|-1)\nu_r \frac{1}{x_r}$$

Make choice after downloaded $\frac{1}{\beta(|R(t)|-1)}$ of the file

Parameter: scrape intensity (overhead) ~ β

Note: Independent of swarms size

- Similarly, average in rate from trackers $r' \in R(t) \backslash \{r\}$
  - Destination chosen uniform at random

$$\frac{1}{|R(t)|-1} \sum_{r' \in R(t) \backslash r} \beta(|R(t)|-1)\nu_{t,r'} = \sum_{r' \in R(t) \backslash r} \beta \nu_{t,r'}$$

Peer migration (using RPM)

- How to pick a good migration rule??

- Rate out of a swarm r

Migration probability $\frac{1}{x_r}$

Download rate

Peers in swarm

$$x_r \beta(|R(t)|-1)\nu_r \frac{1}{x_r}$$

Make choice after downloaded $\frac{1}{\beta(|R(t)|-1)}$ of the file

Parameter: scrape intensity (overhead) ~ β

Note: Independent of swarms size

- Similarly, average in rate from trackers $r' \in R(t) \backslash \{r\}$
  - Destination chosen uniform at random

$$\frac{1}{|R(t)|-1} \sum_{r' \in R(t) \backslash r} \beta(|R(t)|-1)\nu_{t,r'} = \sum_{r' \in R(t) \backslash r} \beta \nu_{t,r'}$$

Note: Rates equal when download rates in swarms are equal

Mixing Performance

- *Virtual swarm size*
  - Fraction internal and external ($\bar{y}_{t,r}$) peers known in swarm

$$M_{t,r} = \frac{x_{t,r} + \sum_{r' \in R(t) \setminus \{r\}} \bar{y}_{t,r'}}{x_t}$$

- Average virtual swarm size

$$M_t = \frac{1}{x_t} \sum_{r \in R(t)} x_{t,r} M_{t,r}$$

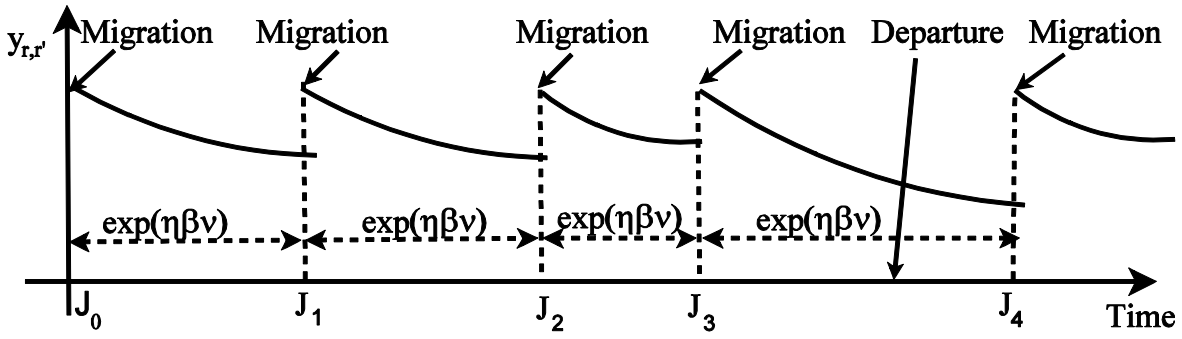
- Without swarm management

Load balancing

$$M_t = \sum_{r \in R(t)} \left(\frac{x_{t,r}}{x_t} \right)^2 \geq \frac{1}{|R(t)|}$$

RPM Protocol Performance

- Lower bound under exponential assumption (holding,migration)
- η share of peers implements RPM, look at tracker r
  - External peers known time z after last migration $y_{t,r}(z) \geq pe^{-\mu z}$



- Renewal-reward process $\{(J_i,R_i): i \geq 0\}$

$$E[J_{i+1} - J_i] = \frac{1}{\eta\beta\nu}$$

$$E[R_i] = \int_0^\infty \left(\int_0^h pe^{-\mu z} dz \right) \eta\beta\nu e^{-\eta\beta\nu h} dh = p\frac{1}{\mu + \eta\beta\nu}$$

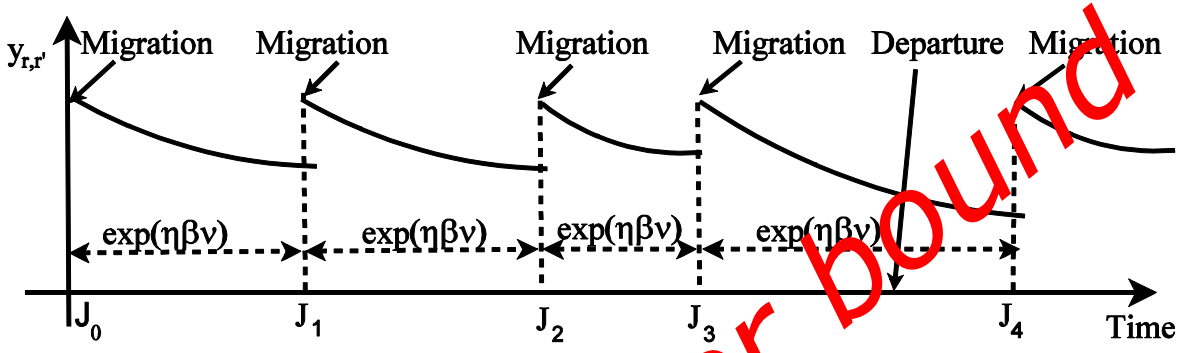
$$\bar{y}_{t,r'} \geq \frac{E[R_i]}{E[J_{i+1} - J_i]} = p\frac{\eta\beta\nu}{\mu + \eta\beta\nu}$$

$$\nu \geq \mu \Rightarrow \bar{y}_{t,r} \geq \frac{\eta\beta}{1+\eta\beta}$$

$$\frac{\partial f}{\partial \beta} > 0, \frac{\partial^2 f}{\partial \beta^2} < 0$$

RPM Protocol Performance

- Lower bound under exponential assumption (holding,migration)
- η share of peers implements RPM, look at tracker r
  - External peers known time z after last migration $y_{t,r}(z) \geq pe^{-\mu z}$



- Renewal-reward process $\{(J_i,R_i): i \geq 0\}$

$$E[J_{i+1} - J_i] = \frac{1}{\eta\beta\nu}$$

$$E[R_i] = \int_0^\infty \left(\int_0^h pe^{-\mu z} dz \right) \eta\beta\nu e^{-\eta\beta\nu h} dh = p\frac{1}{\mu + \eta\beta\nu}$$

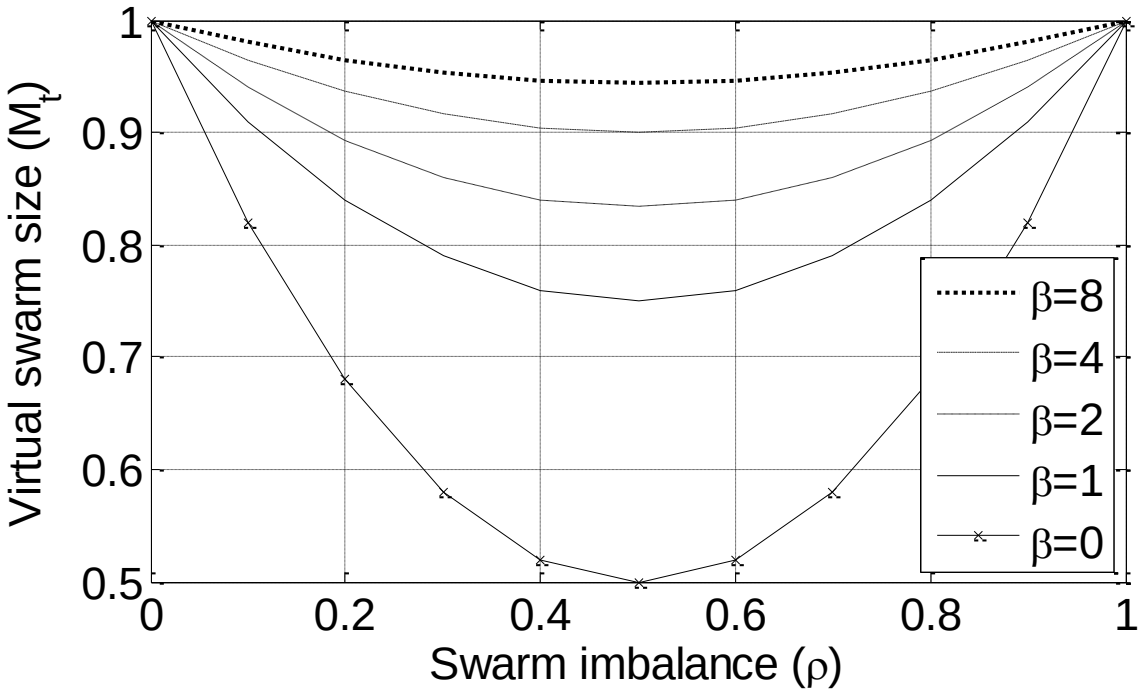
$$\bar{y}_{t,r'} \geq \frac{E[R_i]}{E[J_{i+1} - J_i]} = p\frac{\eta\beta\nu}{\mu + \eta\beta\nu}$$

$$\nu \geq \mu \Rightarrow \bar{y}_{t,r} \geq \frac{\eta\beta}{1+\eta\beta}$$

$$\frac{\partial f}{\partial \beta} > 0, \frac{\partial^2 f}{\partial \beta^2} < 0$$

Derive lower bound

Mixing efficiency (RPM)
Swarm imbalance

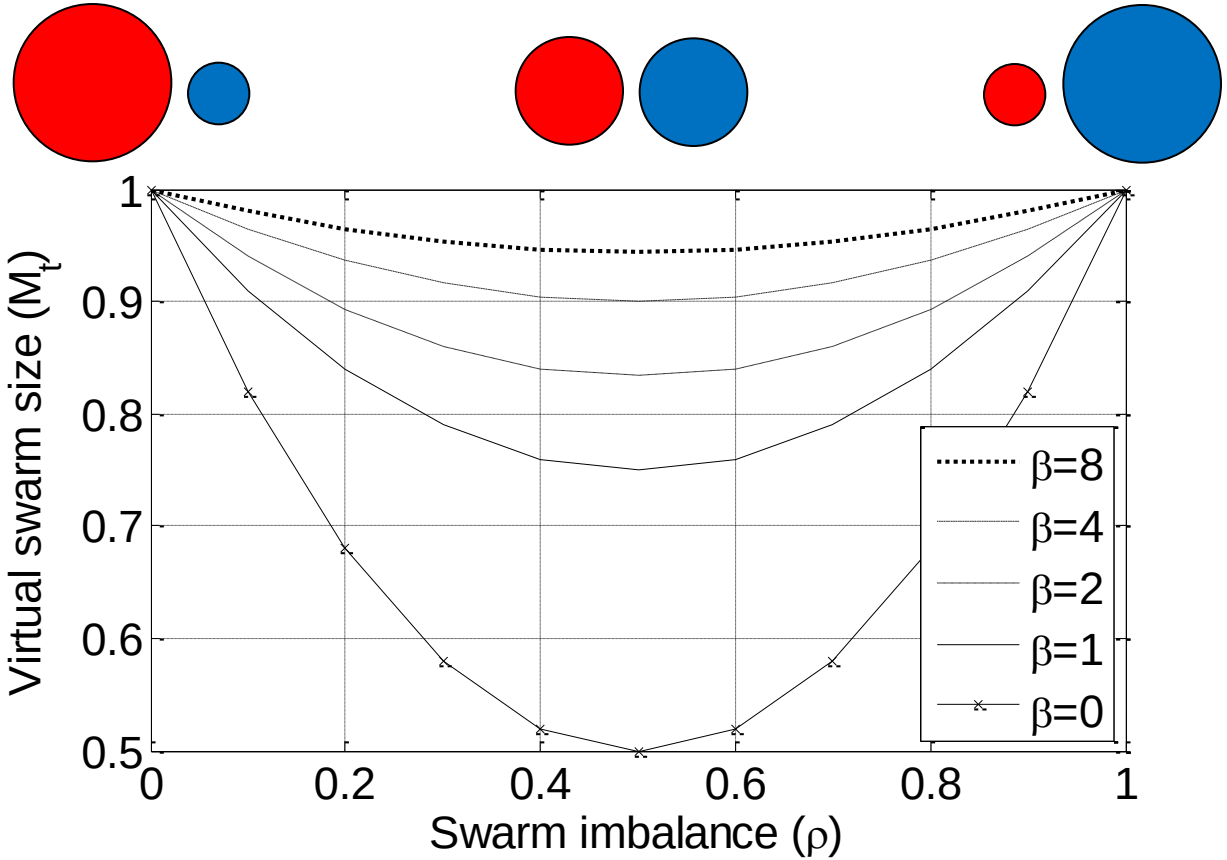


$$\rho = \frac{x_{t,r_1}}{x_{t,r_1} + x_{t,r_2}}$$

|R(t)|=2, η=1,μ=v,p=∞

Mixing efficiency (RPM)

Swarm imbalance

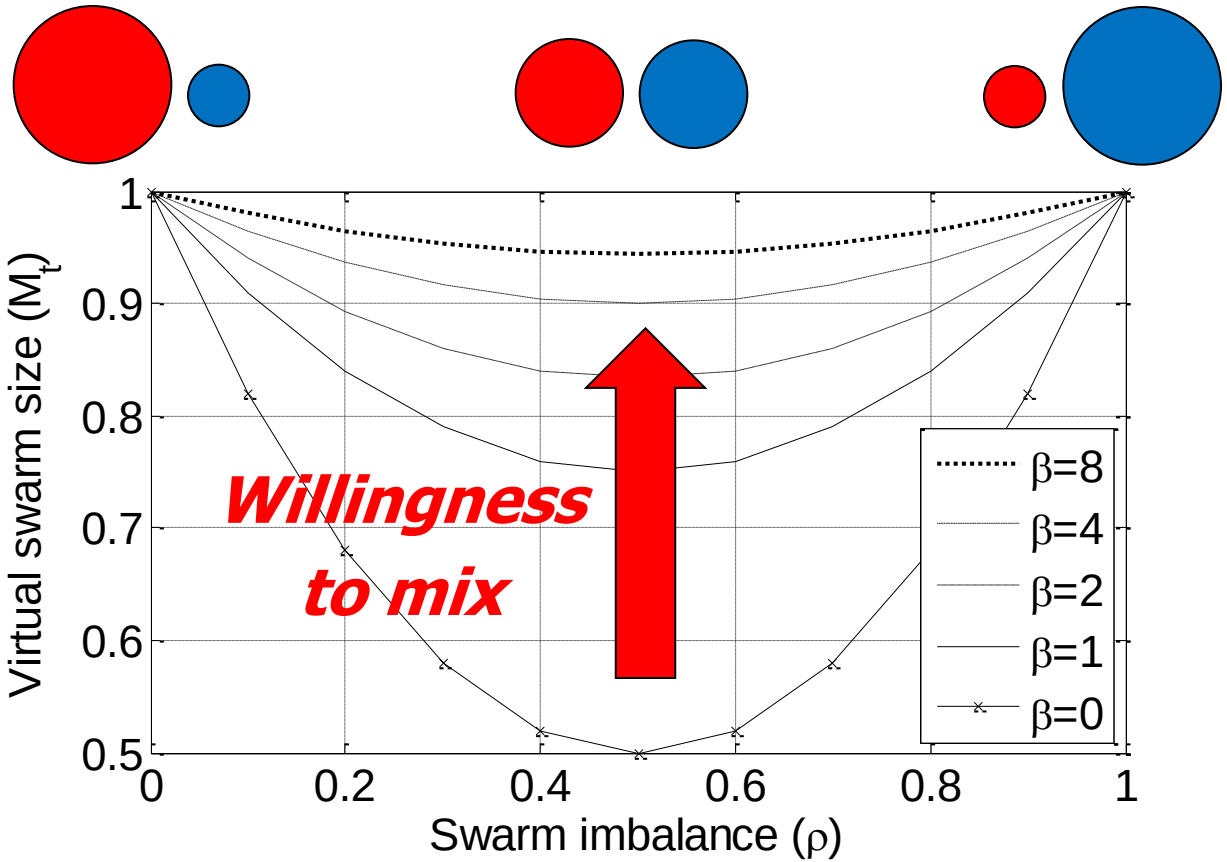


$$\rho = \frac{x_{t,r_1}}{x_{t,r_1} + x_{t,r_2}}$$

|R(t)|=2, η=1,μ=v,p=∞

Mixing efficiency (RPM)

Swarm imbalance



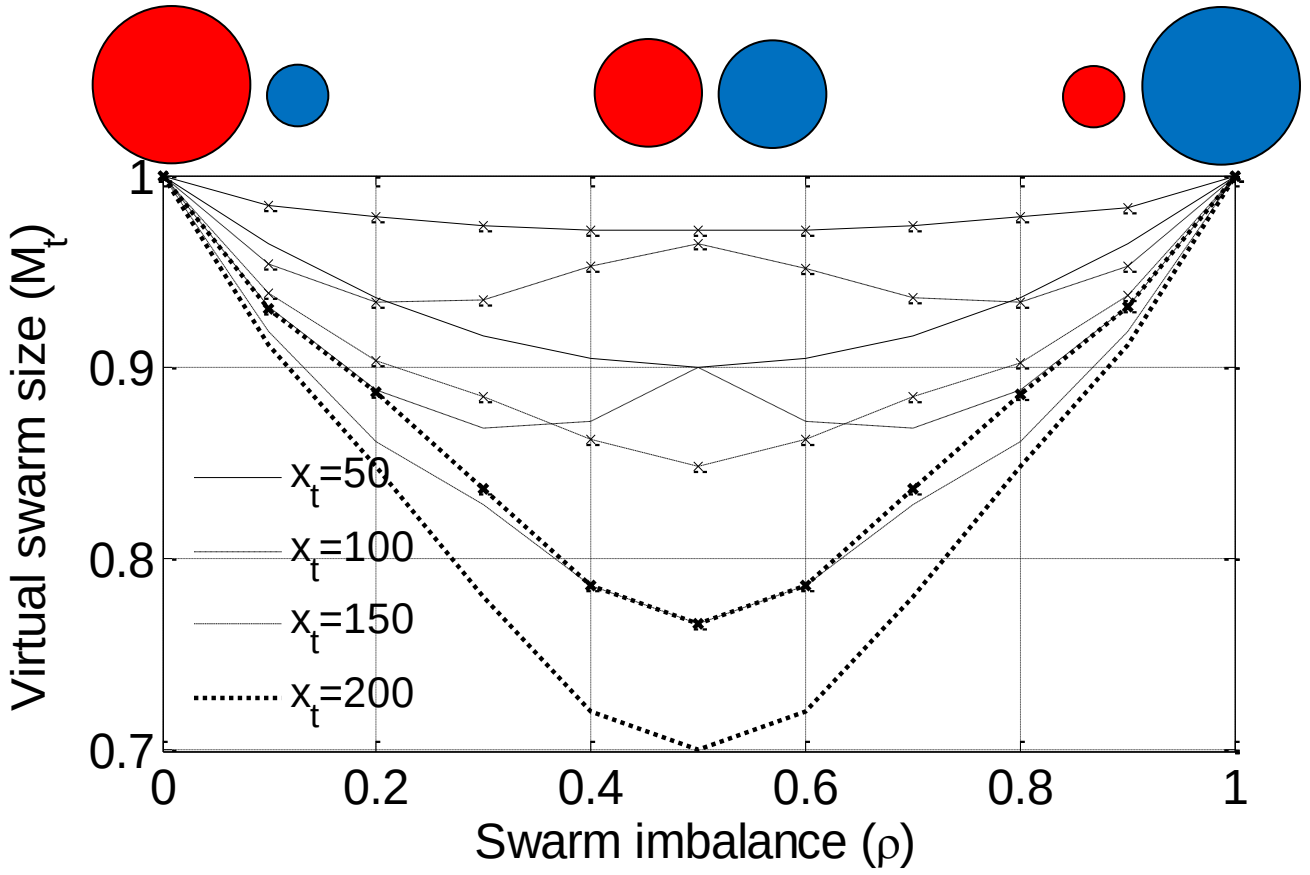
$$\rho = \frac{x_{t,r_1}}{x_{t,r_1} + x_{t,r_2}}$$

|R(t)|=2, η=1,μ=v,p=∞

Note: Diminishing returns

Mixing efficiency (RPM)

Swarm imbalance (limited peer memory)

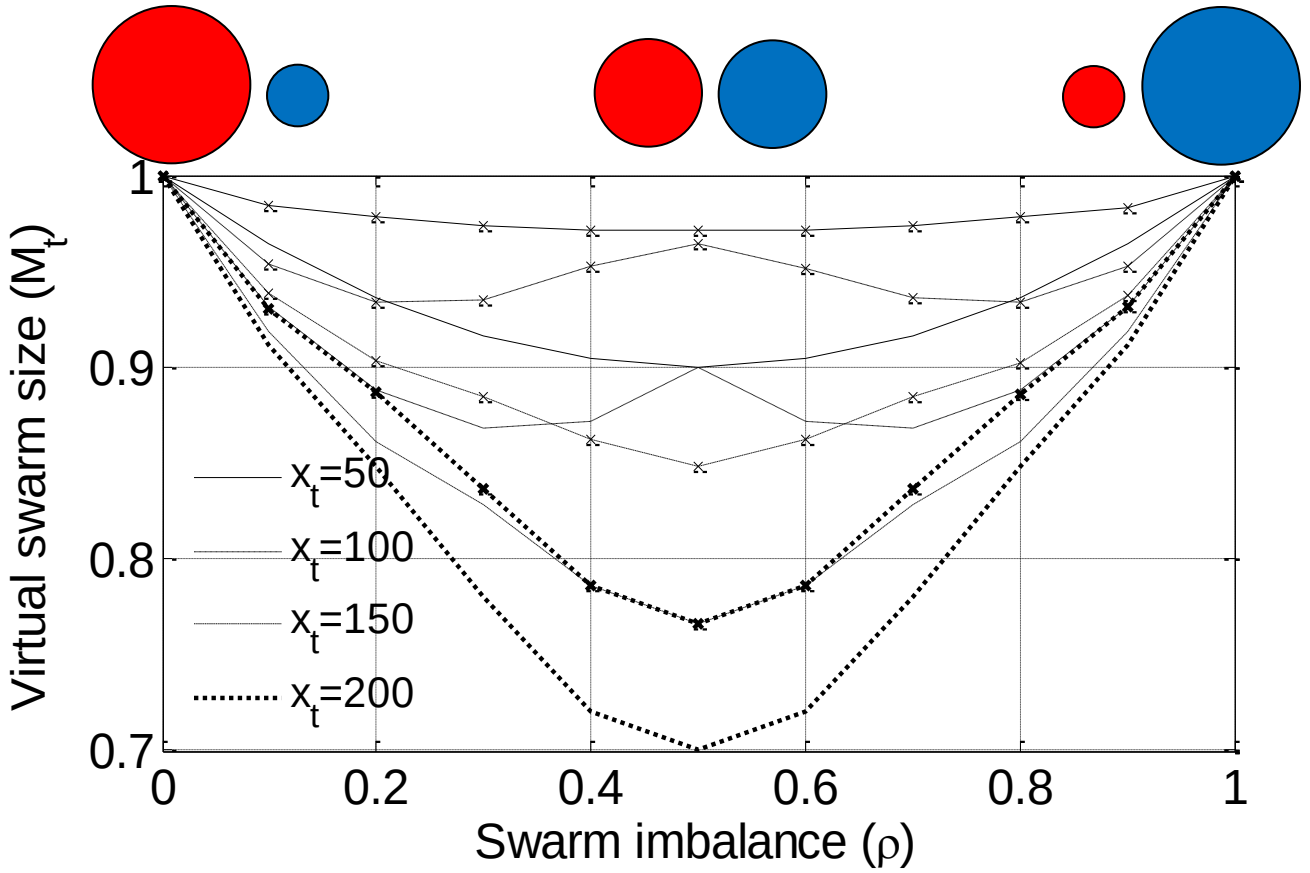


|R(t)|=2, ηβ=4,μ=v,p=50

(model and simulations 'x')

Mixing efficiency (RPM)

Swarm imbalance (limited peer memory)

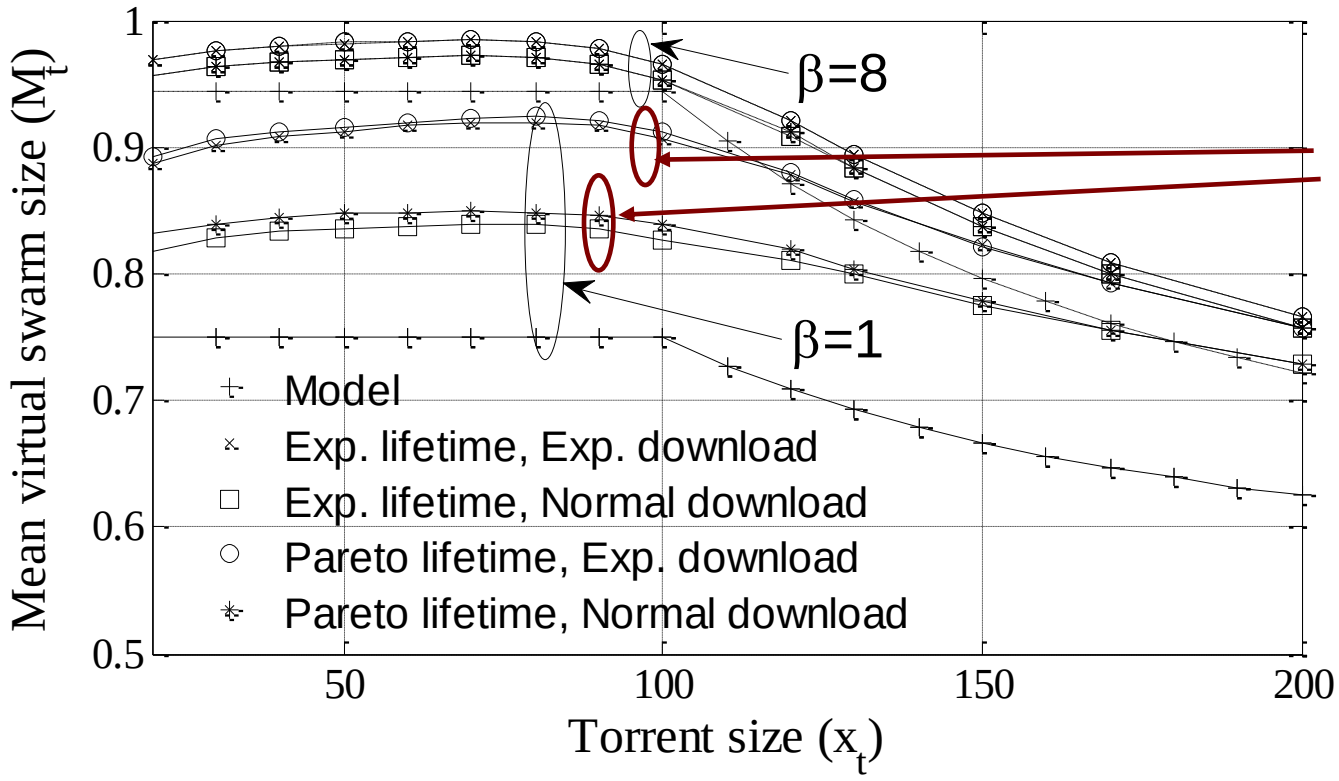


|R(t)|=2, ηβ=4,μ=v,p=50
(model and simulations 'x')

Note: Balanced no longer worst case

Mixing Efficiency

RPM vs torrent size (analytic + simulations)

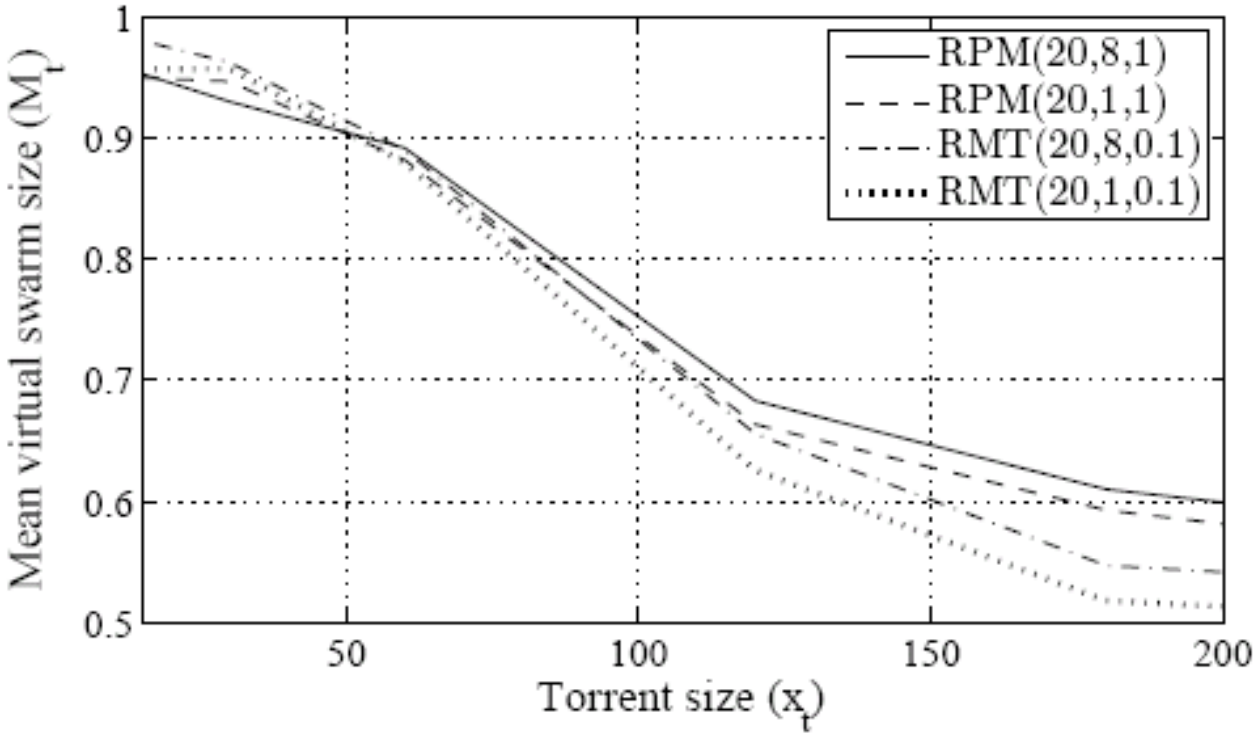


Download time distribution matters (β=1)

|R(t)|=2, η=1,μ=v,p=50

Mixing Efficiency

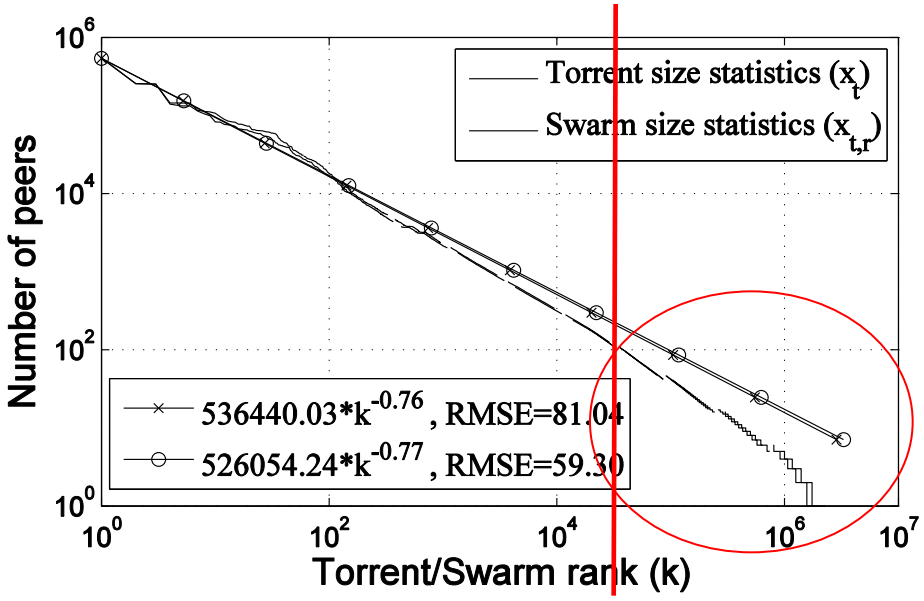
RPM vs torrent size (experiments rTorrent)



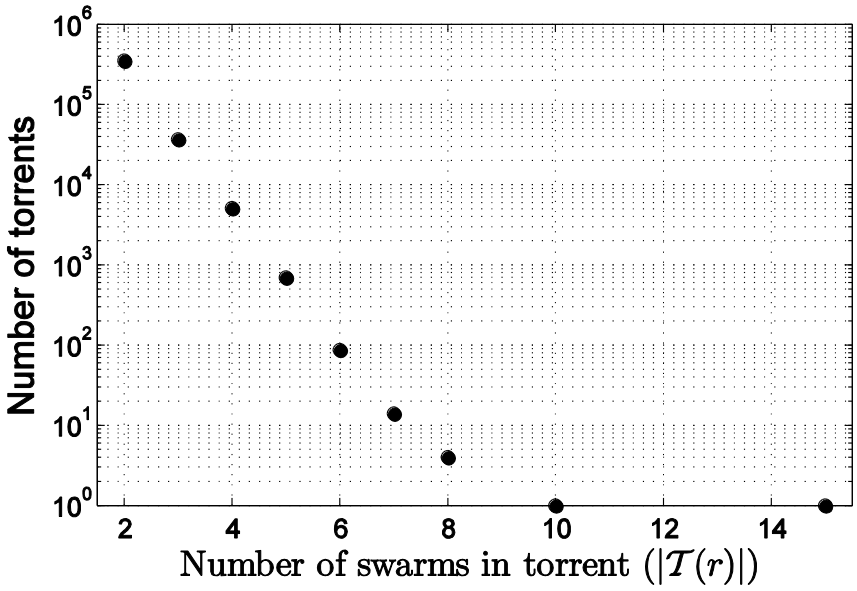
|R(t)|=2, η=1,μ=v,p=50

Case study
BitTorrent measurements

- Most swarms are small
  - Power-law: Long tail of moderately popular files
  - 99% of swarms smaller than 200 peers; half of the peers

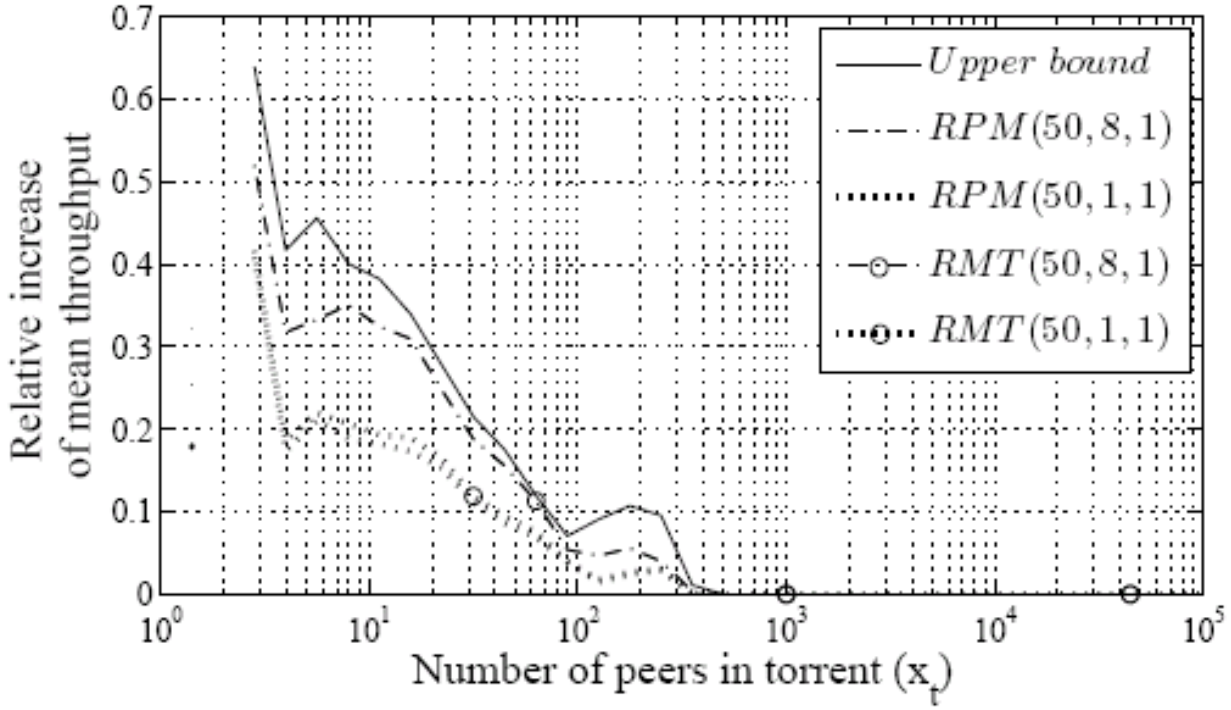


- Many torrents consist of several swarms
  - ~350.000 (small) multi-tracked torrents



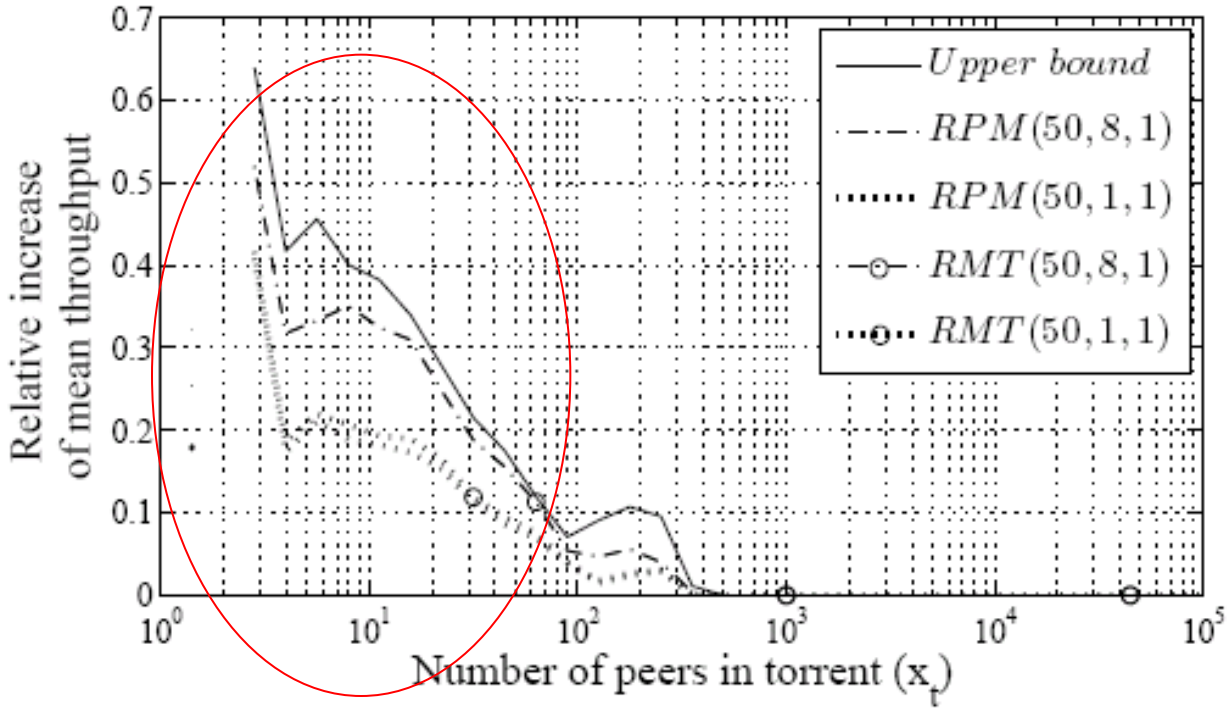
G.Dán, N.Carlsson „Power-law revisited: A large scale measurement study of P2P content popularity”,
Proc. of IPTPS 2010

Throughput improvement
RPM/RMT with parameters (p, ηβ, μ/v)



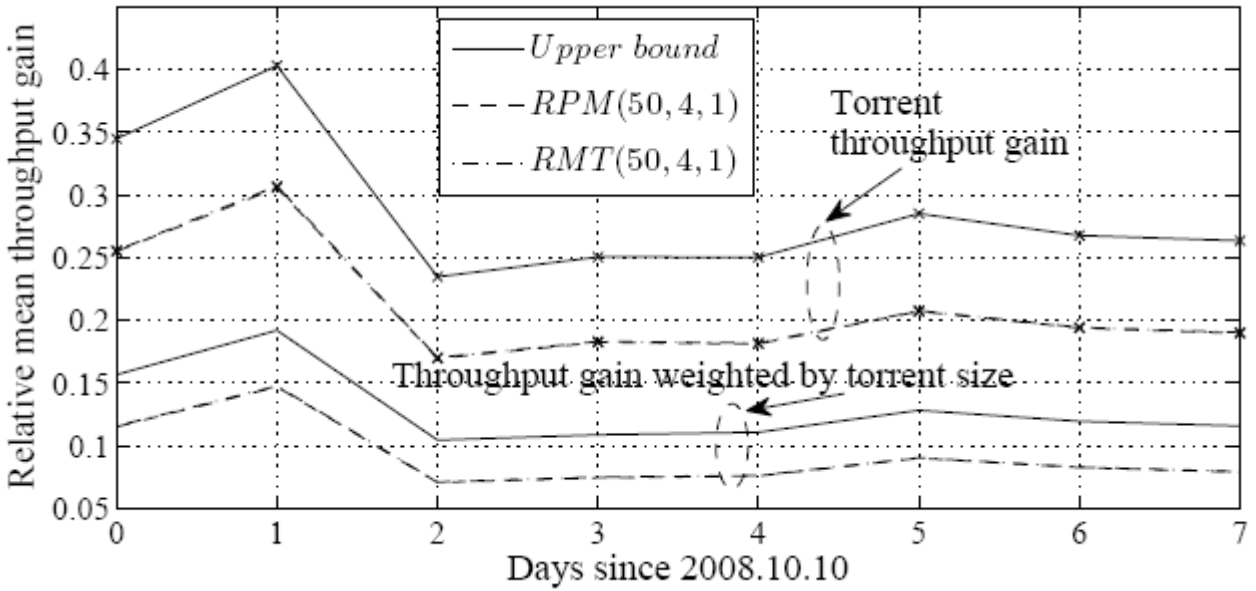
- Substantial improvement
- Close to upper bound
- Decreasing marginal gain in β

Throughput improvement
RPM/RMT with parameters (p, ηβ, μ/v)



- Substantial improvement
- Close to upper bound
- Decreasing marginal gain in β

Throughput improvement
RPM/RMT with parameters (p, ηβ, μ/v)



- Torrents with <300 peers
- Average throughput gain similar across days

Summary of Contributions

- Two distributed protocols for swarm management
  - Independent trackers
  - Gossip protocol
  - Constant overhead, independent of swarm size
- Analytical model (based on renewal theory)
  - Simulations and experiments validate the model
- Large-scale measurement evaluations
  - The performance of small swarms is worse
  - Most swarms are small
  - Many torrents consist of several swarms
  - Assess potential throughput gains

Thank you!

Efficient and Highly Available Peer Discovery: A Case for Independent Trackers and Gossiping

---

Niklas Carlsson (niklas.carlsson@liu.se)