On Thieves and Terrorists in Peer-to-Peer Systems

Stefan Schmid

Distributed Computing Group

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Introduction

THANK YOU !

- Thank you for the invitation! ☺

- Myself:

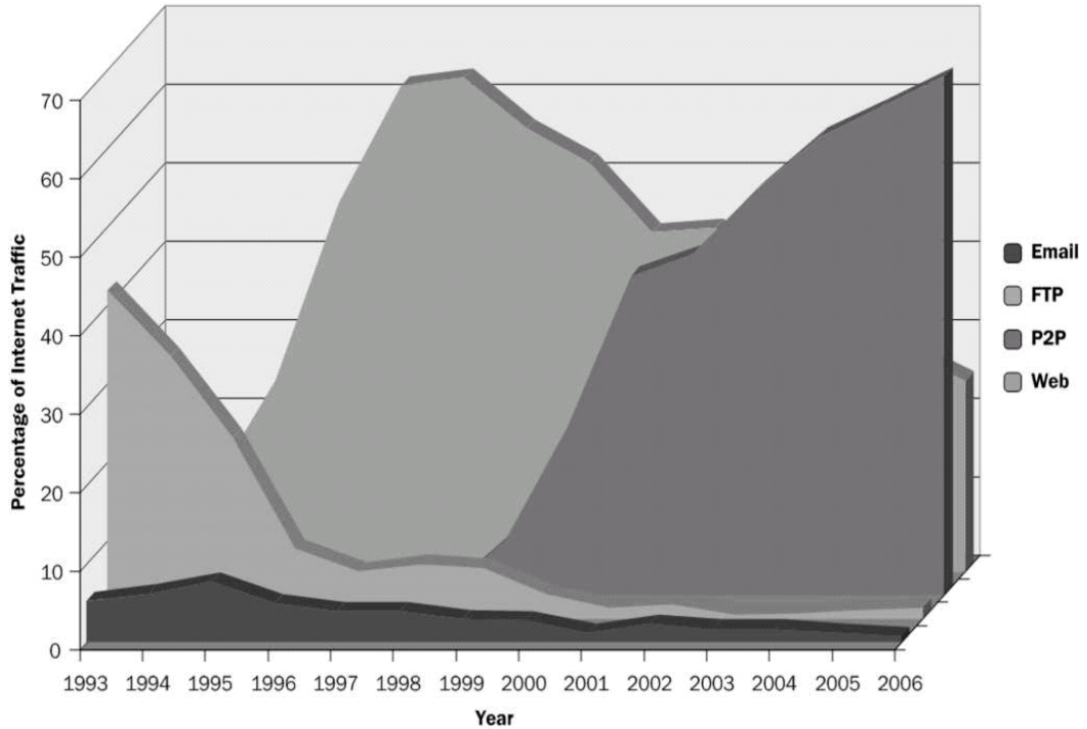
  - MSc in CS at ETH Zurich, Switzerland

  - 3rd year PhD student of the Distributed Computing Group of Prof. Roger Wattenhofer

  - For more details, see http://dcg.ethz.ch/members/stefan.html

Peer-to-Peer Systems (1)

- We all know: BitTorrent, eMule, Kazaa, Tribler, etc.
  - important: accounts for much Internet traffic today! (source: *cachelogic.com*)



Peer-to-Peer Systems (2)

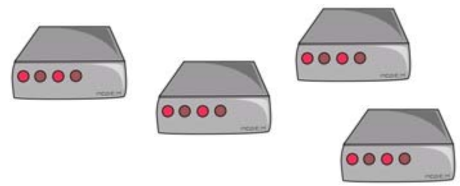
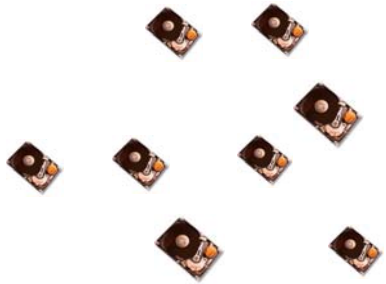
- Besides file sharing, also interesting for large-scale computations, media streaming, etc.
  - projects at DCG: Pulsar p2p streaming (e.g., IPTPS), Kangoo networking, etc.

- Advantages of p2p paradigm
  - Scalability
  - Low costs (e.g., cheap content distribution, don‘t have to pay ISP)
  - Fault-tolerance
  - …

- Challenges
  - Machines under the control of individual users
  - Dynamics / Churn
  - Selfishness and other forms of non-cooperation

The Importance of Cooperation

- Peer-to-peer computing is based on the resource contribution of the constituent parts, the peers
  - e.g., upload bandwidth, disk space, CPU cycles, etc.

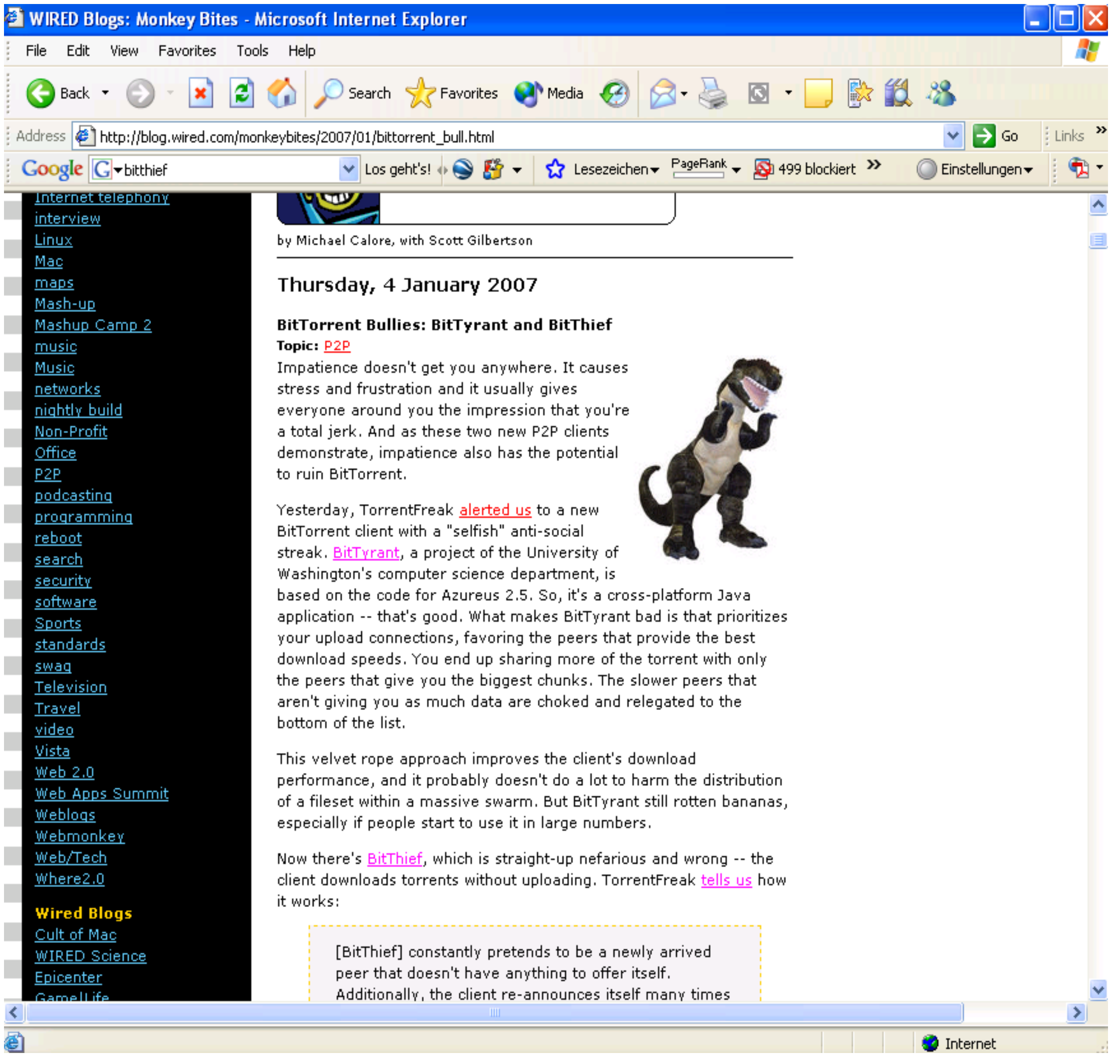




- Who is cooperative? Three models for participants in p2p computing:



What Are P2P Users Really Like??

- Reactions to our free riding BitThief client…



Reactions

- Reactions to our free riding BitThief client…

"Anyhow, bitthief is a client which I've been waiting for so long, I mean.. bitcomet bent the rules but never really broke any of them.. that much Bitthief is an interesting client in that it openly says "fuck you, and fuck your swarm" to the torrent community. I wonder how fast this will get banned at every tracker alive. As others have said, this makes bittyrant look like a sunday school boy."

A fan!

-----Original Message-----
From: Warren Henning [mailto:warren.henning@gmail.com]
Sent: Friday, January 12, 2007 3:03 PM
To: lochert@tik.ee.ethz.ch; schmiste@tik.ee.ethz.ch; wattenhofer@tik.ee.ethz.ch
Subject: Stop distributing BitThief, you jerks!

BitTorrent is a beautiful thing and you are intentionally fucking it up by distributing software that is apparently specifically designed to attack the entire basis of the function of BitTorrent, software that serves no legitimate purpose.

Luckily it apparently requires having a JRE installed right now, and the knuckle-dragging numbskulls you've worked so hard to cater to are probably too lazy to install that.

You people piss me off.

Warren Henning

Not a fan!

… and no, we do not cooperate with RIAA etc.! ☺

Non-Cooperation: A Challenge

- Why be selfish? E.g, no direct benefits from cooperation, anonymity, etc.

- Why be malicious? E.g., RIAA, etc.

- In spite of the topic‘s importance, cooperation is not enforced effectively by most of today‘s p2p systems!

- Reasons?
  - Not necessary?
  - Not possible?

Talk Outline

- BitThief: Free riding in BitTorrent is easy

Based on our HotNets'06 paper

- When Selfish Meets Evil:
  A game-theoretic framework for gaining insights into selfish and malicious behavior in distributed systems
  - Can system be exploited or not?

Based on our PODC'06 paper

Talk Outline

- **BitThief: Free riding in BitTorrent is easy**

- When Selfish Meets Evil:
  A game-theoretic framework for gaining
  insights into selfish and malicious behavior
  in distributed systems

BitTorrent (1)

- Many peers (a swarm) share the same file
  - File divided into pieces

- How to find swarm for file?
  - Described in a metafile (torrent file)
  - It is obtained from websites (HTTP request)
  - Torrent file contains information about tracker
  - Also stores hash values for piece verification

- Peers join swarm through tracker
  - Tracker coordinates interactions between peers
  - e.g., it tells peers about other participants
  - Maintains a list of currently active peers, returns random subset upon request
  - Peers periodically contact tracker (f = 15 min, less for BitThief!)

BitTorrent (2)

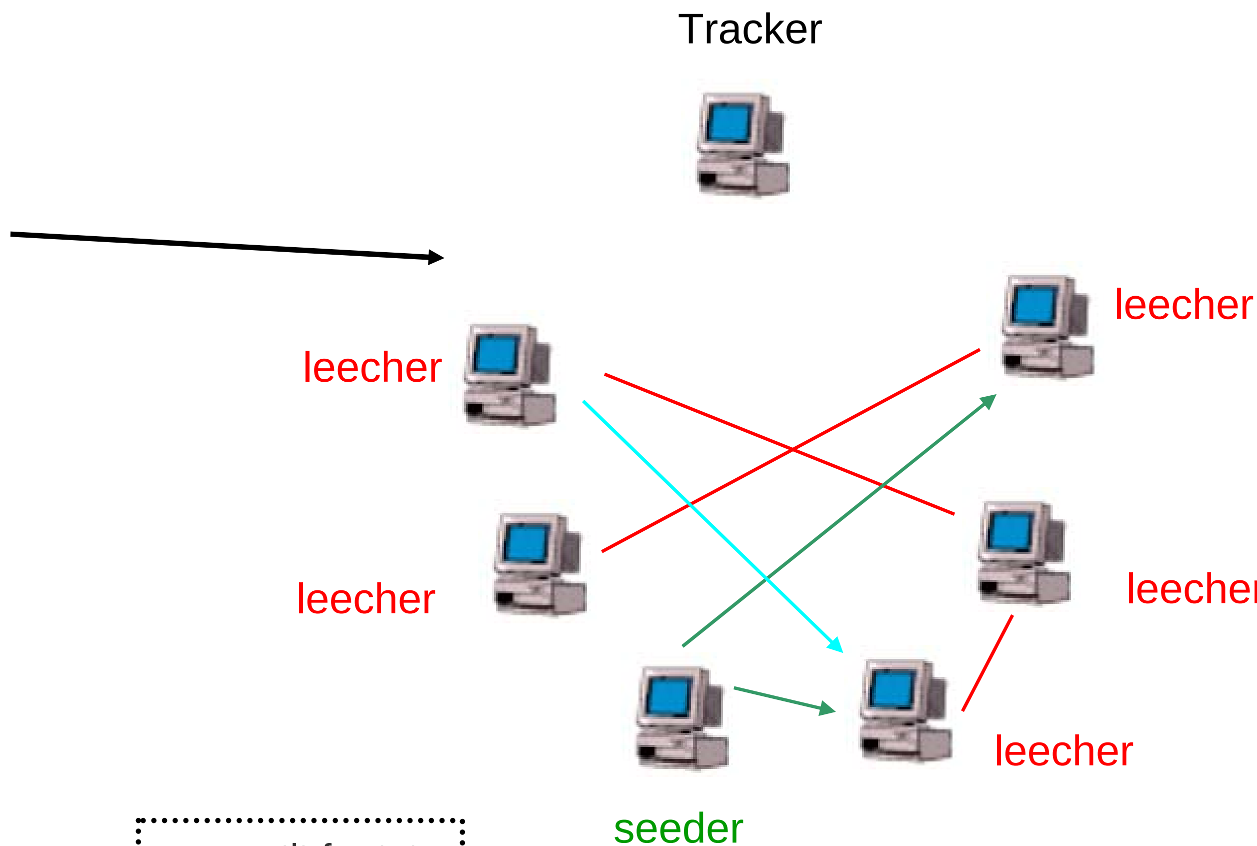
- In each torrent, there are seeders and leechers
  - Seeder: Already downloaded the entire file, provide the pieces for free (round robin)
  - Leechers: Upload only to peers which give something in return (tit-for-tat), i.e., upload to peers (at same rate) which gave best download rates over last 10sec; however, also unchoking a fixed number of peers (help to bootstrap & find new, potentially better peers)

- In spite of fairness mechanism, BitTorrent can be cheated.

BitTorrent (3)

website with .torrent file

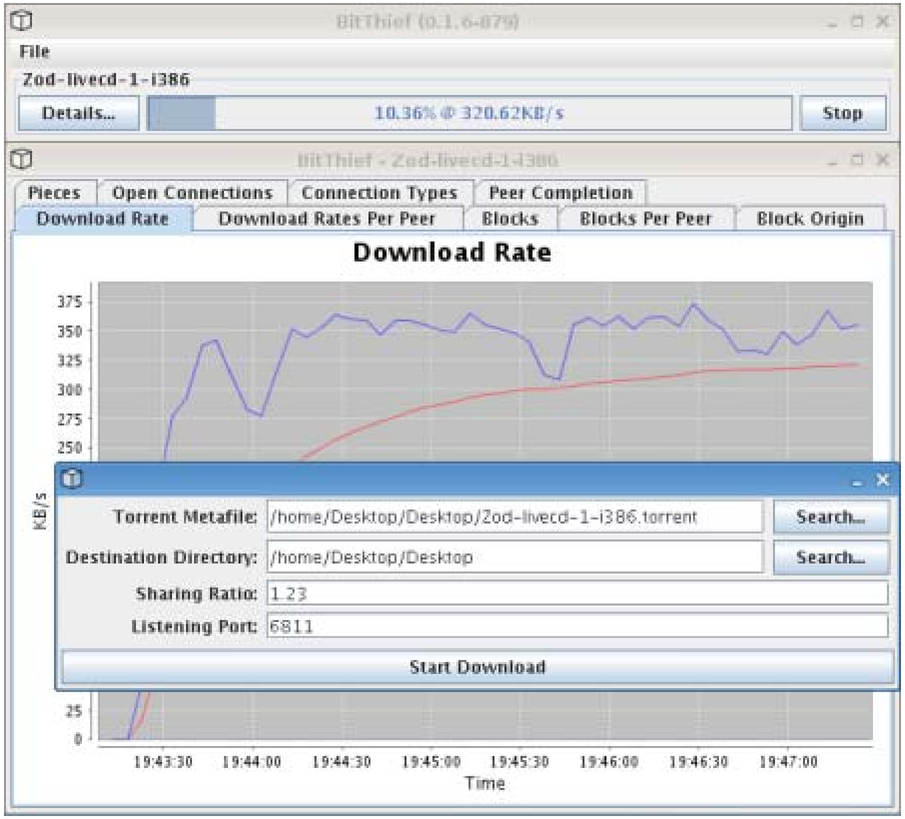
- Tracker address
- Verification data
- ....



- tit-for-tat
- unchoking
- seeding

BitThief (1)

- Our BitThief client is a Java client which achieves fast downloads without uploading at all – in spite of BitTorrent's incentive mechanism!



BitThief (2)

- BitThief‘s three tricks!

  - Open as many TCP connections as possible (no performance problem!)

  - Contacting tracker again and again, asking for more peers (never banned during our tests!)

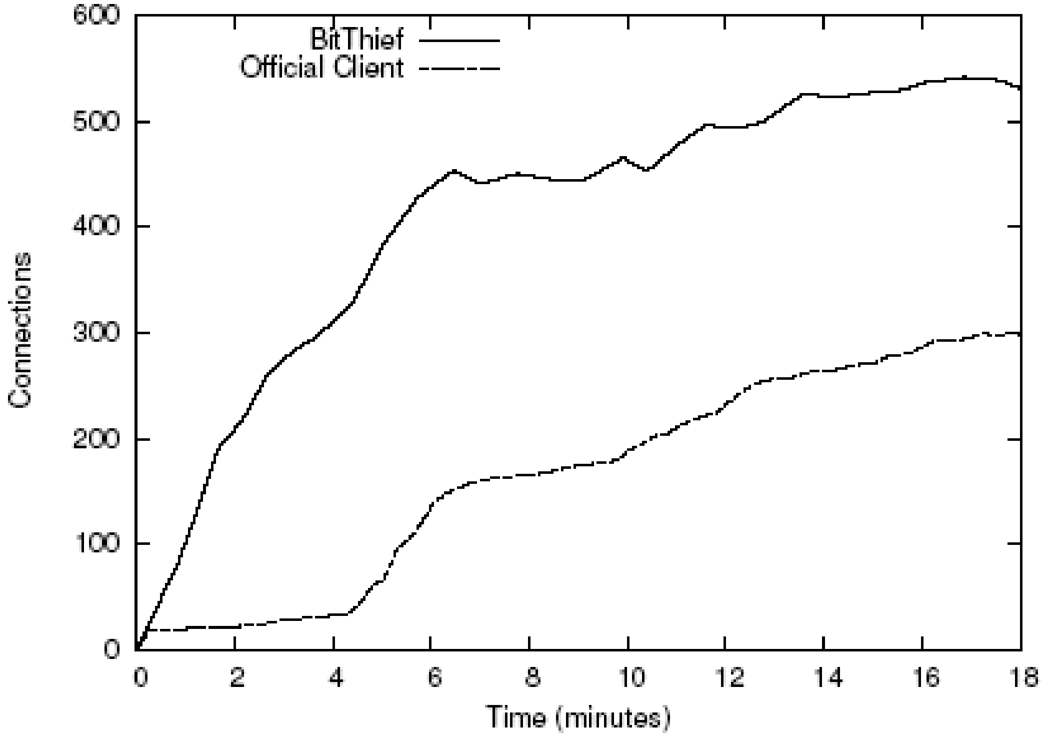
  - Pretend being a great uploader in sharing communities (tracker believed all our tracker announcements)

  -> Exploit optimistic unchoking

  -> Exploit seeders

  -> Exploit sharing communities

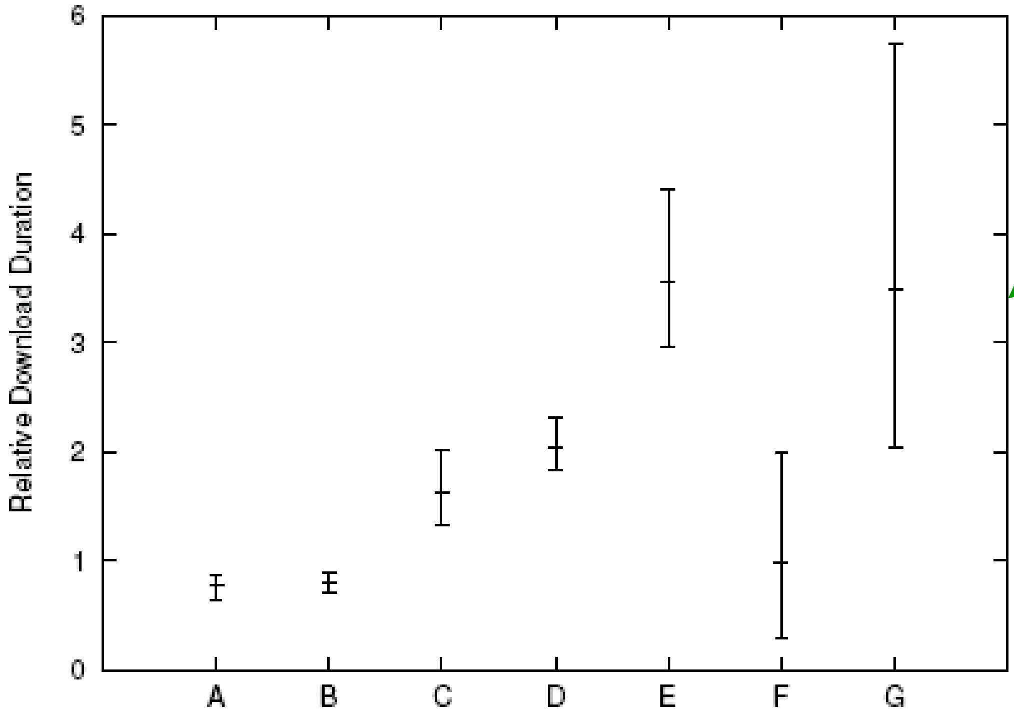
Open TCP Connections



Results for Different Torrents (w/ Seeders)

2

compared to official client (with unlimited number of allowed connections)



4 BitThief with public IP and open TCP port

number of peers announced by tracker

max peers found by BitThief

1

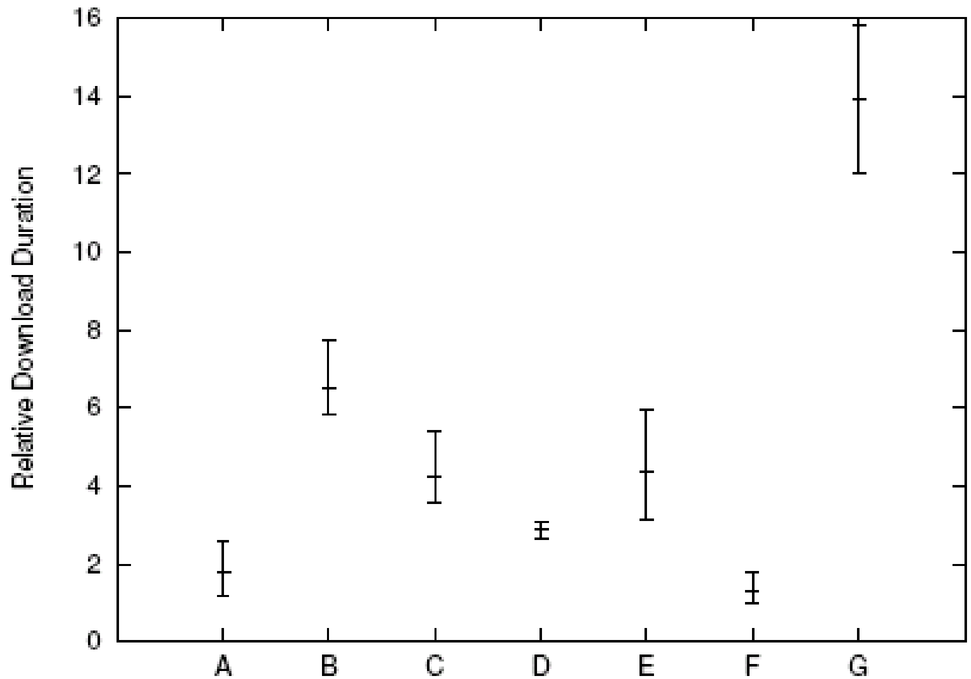
	Size	Seeders	Leechers
A	170MB	10518 (303)	7301 (98)
B	175MB	923 (96)	257 (65)
C	175MB	709 (234)	283 (42)
D	349MB	465 (156)	189 (137)
E	551MB	880 (121)	884 (353)
F	31MB	N/A (29)	N/A (152)
G	798MB	195 (145)	432 (311)

3

- All downloads finished!
- Fast for small files (fast startup), many peers and many seeders!

(Seeders identified by having message)

Results w/o Seeders



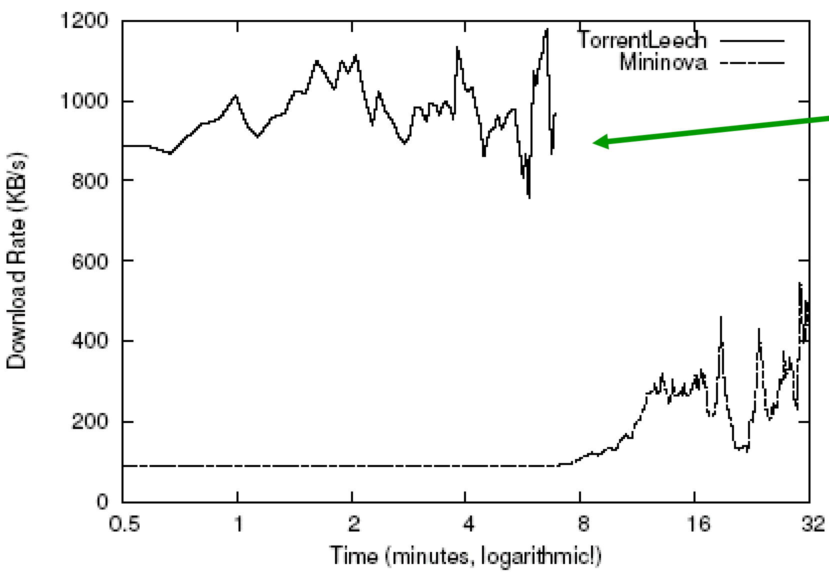
- Seeders detected with bitmask / have-message
- Even without seeder it‘s fast!
- Unfair test: Mainline client was allowed to use seeders!

	Size	Seeders	Leechers
A	170MB	10518 (303)	7301 (98)
B	175MB	923 (96)	257 (65)
C	175MB	709 (234)	283 (42)
D	349MB	465 (156)	189 (137)
E	551MB	880 (121)	884 (353)
F	31MB	N/A (29)	N/A (152)
G	798MB	195 (145)	432 (311)

Results in Sharing Communities (1)

- Sharing communities ban peers with low sharing ratios
- Uploading is encouraged; user registration required
- It‘s been observed that peers usually stay longer online in these communities! (interesting for BitThief!)
- Many seeders
- Client can report uploaded data itself! (tracker announcements)

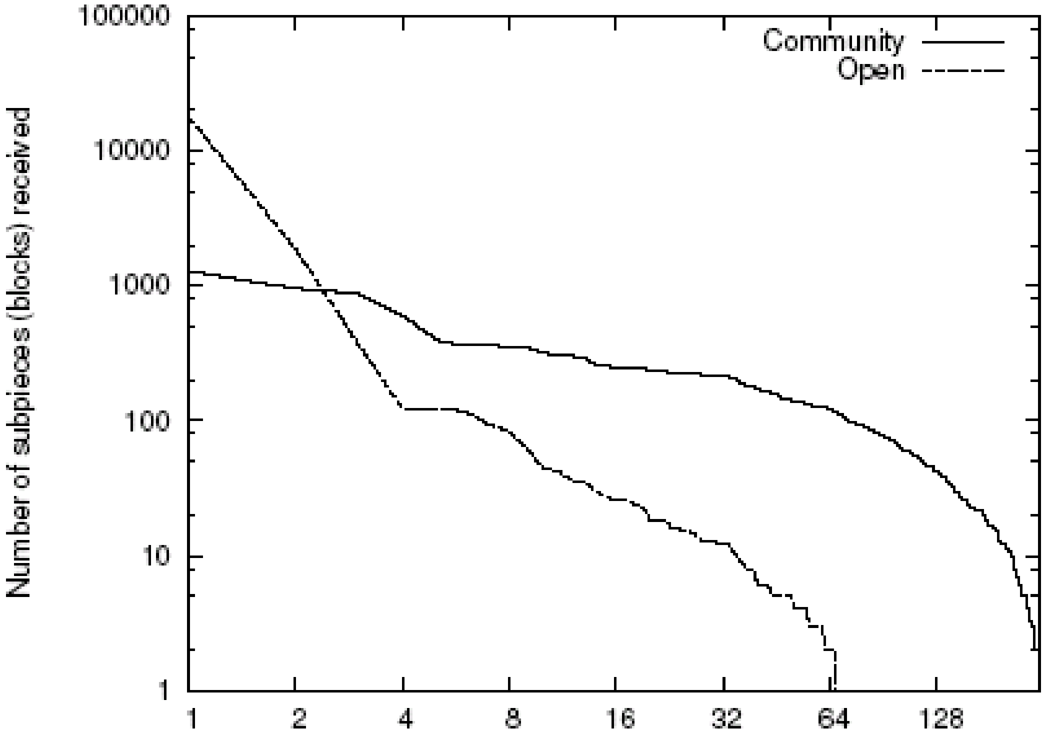
  - As tracker does not verify, it‘s easy to cheat!



4 x faster!
(BitThief had a faked sharing ratio of 1.4; in both networks, BitThief connected to roughly 300 peers)

Results in Sharing Communities (2)

- In communities, contribution is more balanced



Some Tricks that Did not Work

- Some tricks did not work for BitThief

  - Announce many available pieces (0%-99% all the same, 100% very bad, considered a seeder)

  - Upload garbage (easier with mainline client than with Azureus; Azureus remembers from which it has got most subpieces/blocks and tries to get all from him; otherwise you are banned)

  - Sybil attacks with same IP address

  - …

See paper for more details!

BitThief Conclusion

- BitTorrent is one of the few systems incorporating fairness.
- Still, it can be cheated easily.

  - Many exploits not tested yet, e.g., more peers via DHT, ISP caching, etc.
- How to do better?

  - Difficult: „Venture capital" for new peers needed!

  - First ideas, e.g., BitTorrent's fast extension (free piece set based on IP)
- Will people be selfish and use BitThief? We don't know. Currently ~100 different IPs per day…

  (Wanna try…? ☺ dcg.ethz.ch -> BitThief)

We believe that it is crucial to improve existing mechanisms!

When is BitThief Fast Compared to Other Clients?

- In environments with many seeders.

  - Connect to many of them and download quickly.

- In environments with only one slow seeder.

  - Exploit optimistic unchoking slots of other leechers (which are starving).

- **But not** in the presence of one fast seeder

  - Leechers are busy with tit-for-tat, saturared upload slots, only optimistic unchoking is left.

Final Note on Related Work: BitTyrant

- BitTyrant is a selfish client presented at NSDI 2007
- Authors find many sources of unwanted altruism in BitTorrent
  - Long convergence time until good neighbors are found
  - Equally splitting bandwidth between neighbors, independently of their upload (as long as they are in active set)
  - …

  ⇒ sublinear growth of download rate compared to upload rate („progressive tax“)
- BitTyrant avoids this altruistic behavior, uploading only as much as necessary. (unlike BitThief…)
- More details -> see their paper

Talk Outline

- BitThief: Free riding in BitTorrent is easy

- When Selfish Meets Evil:
  A game-theoretic framework for gaining
  insights into selfish and malicious behavior
  in distributed systems

Talk Outline

- BitThief: Free riding in BitTorrent is easy

- **When Selfish Meets Evil:**
  A game-theoretic framework for gaining insights into selfish and malicious behavior in distributed systems

Towards Understanding Non-Cooperation

- How to reason about non-cooperation in peer-to-peer computing?
- How to come up with incentive mechanisms which enforce contributions? When are such mechanisms needed at all?
- Tools of algorithmic game theory and mechanism design!

Game Theory

- Game theory answers the question:
  How much worse does a system perform compared to a optimal solution if all players are selfish?
  -> Degradation quantified by the notion of Price of Anarchy

- A large Price of Anarchy indicates that a system needs a better incentive mechanism which ensures collaboration

- Less frequently studied: What is the effect of having malicious players among the selfish players?
  - We will introduce the Price of Malice to quantify this!
  - Large Price of Malice -> malicious players can do a lot of harm!
  -> need mechanism to defend against attackers!

Game Theory

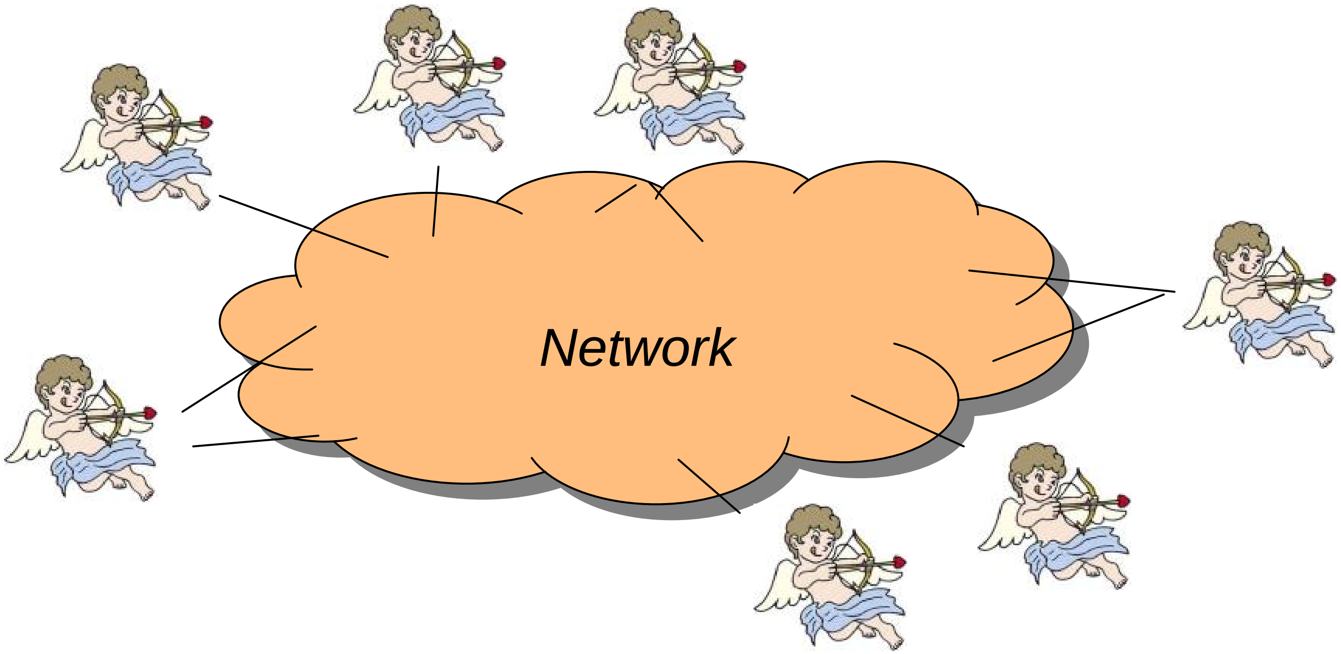
- In the following, we give a sample game-theoretic analysis of the impact of having malicious and selfish players.

- Sample game: virus incolation

Modeling Distributed Systems

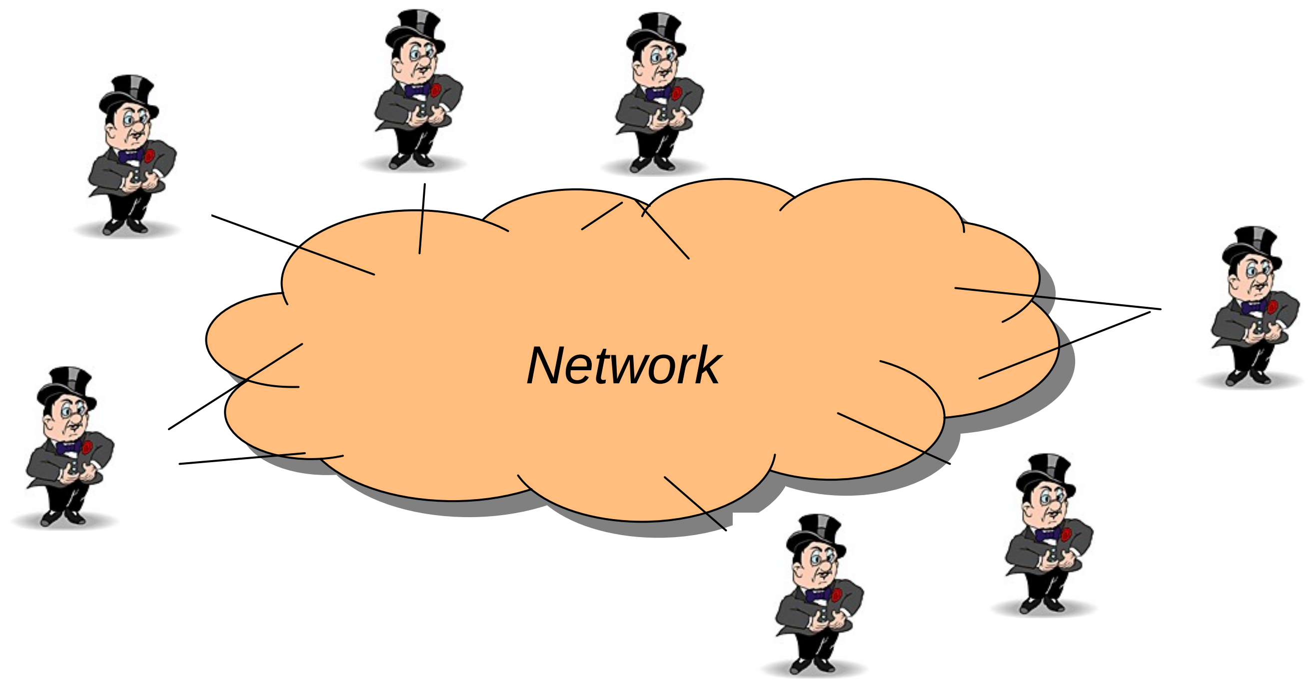
- One possibility to model a distributed system:

  all participants are benevolent ("seeders in BitTorrent")



Selfishness in Networks

- Alternative: Model all participants as selfish (e.g. BitThief!)



Classic game theory: What is the impact of selfishness on network performance…? (=> Notion of price of anarchy, etc.)

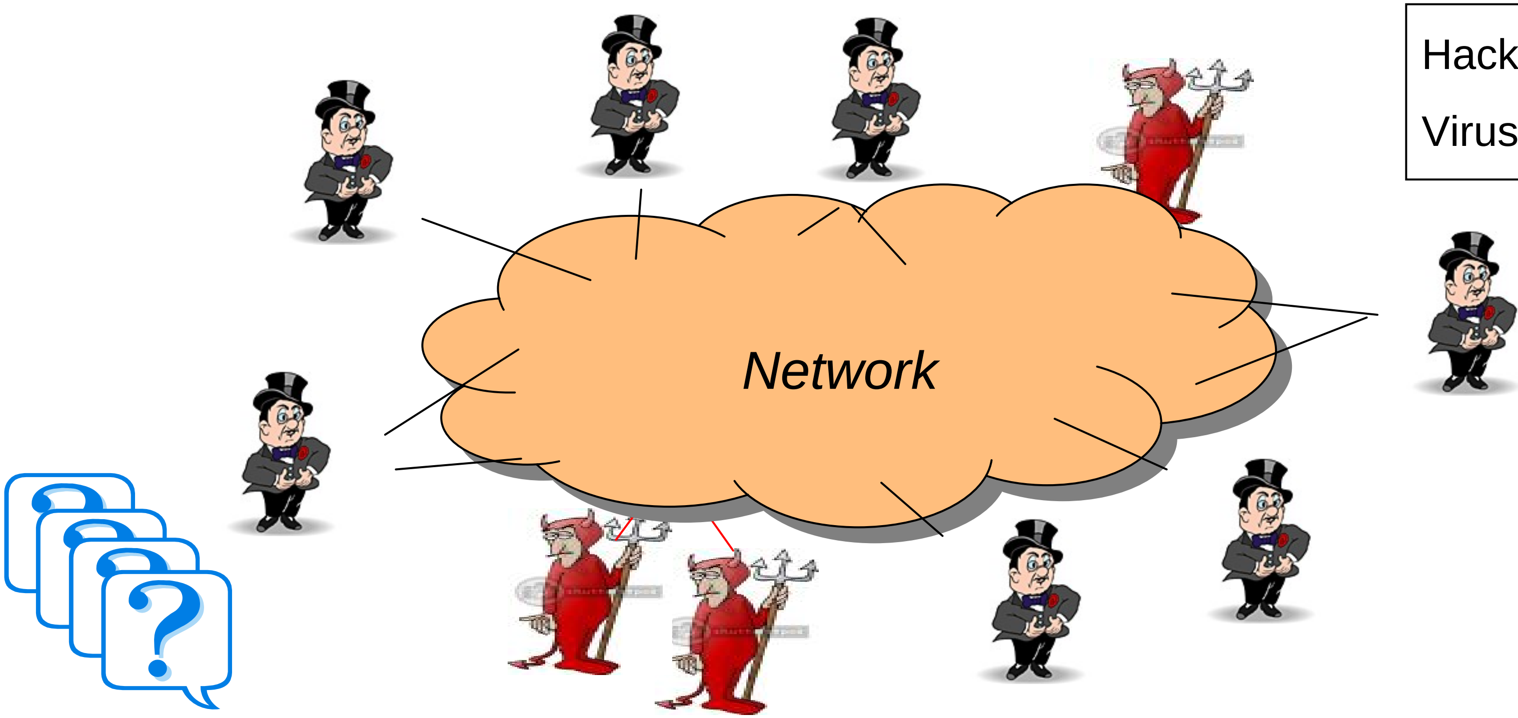
When Selfish meets Evil…

- But selfishness is not the only challenge in distributed systems!

→ Malicious attacks on systems consisting of selfish agents

Hackers, Polluters

Viruses, DOS attacks, …



What is the impact of malicious players on selfish systems…?

Some Definitions

peer, agent, node, host,…

- Goal of a selfish player: minimize its own cost!
- Social cost is the sum of costs of selfish players
- Social Optimum (OPT)
  - Solution yielding minimal social cost of a given problem instance
  - "solution formed by collaborating players"!
- Nash equilibrium
  - "Result" of selfish behavior
  - State in which no player can reduce its costs by changing its strategy
- Price of Anarchy
  - Captures the impact of selfishness by comparison with optimal solution
  - Formally: social costs of worst Nash equilibrium divided by optimal social cost

$$PoA := \frac{\text{worst Nash equilibrium}}{\text{social optimum}}$$

Inverse when considering utilities

"Byzantine Game Theory"

- Game framework for malicious players
- Consider a system (network) with n players
- Among these players, s are selfish
- System contains b=n-s malicious players
- Malicious players want to *maximize* social cost!
- Define Byzantine Nash Equilibrium:

Social Cost:

Sum of costs of *selfish* players:

$$Cost_{tot} = \sum_{i \in Selfish} cost_i(a)$$

A situation in which no selfish player can improve its perceived costs by changing its strategy!

Of course, whether a selfish player is happy with its situation depends on knowledge about the malicious players!

Do they know that there are malicious players? If yes, it will take this into account for computing its expected utility! Moreover, a player may be risk averse or not (reaction), etc.

Actual Costs vs. Perceived Costs

- Depending on selfish players' knowledge, actual costs (-> social costs) and perceived costs (-> Nash eq.) may differ!
- Actual Costs: $cost_i(a)$ ← Players do not know !

  → The cost of selfish player i in strategy profile a
- Perceived Costs: $\widehat{cost}_i(a)$ ← Byz. Nash Equilibrium

  → The cost that player i expects to have in strategy profile a, given preferences and his knowledge about malicious players!

← Many models conceivable →

Risk-averse…

Risk-loving…

Rational…

Nothing…,

Number of malicious players…

Distribution of malicious players…

Strategy of malicious players…

"Byzantine Game Theory"

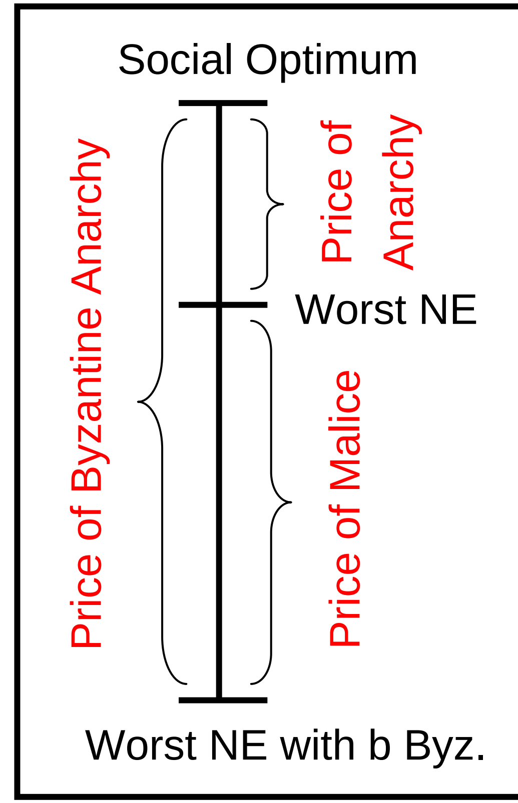
- Price of Anarchy: $PoA := \frac{\text{worst Nash equilibrium}}{\text{social optimum}}$

- We define Price of Byzantine Anarchy:

$$PoB(b) := \frac{\text{worst Byz. NE with } b \text{ malicious players}}{\text{social optimum}}$$

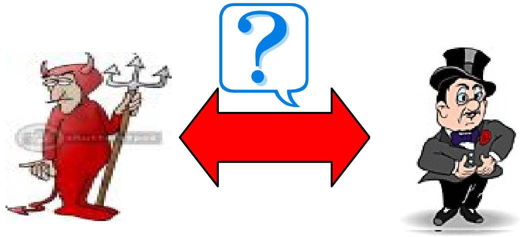
- Finally, we define the Price of Malice!

$$PoM(b) := \frac{\text{worst NE with } b \text{ malicious players}}{\text{worst NE}}$$



The Price of Malice captures the degradation of a system consisting of selfish agents due to malicious participants!

Remark on “Byzantine Game Theory”



- Are malicious players different from selfish players...?

- Theoretically, malicious players are also selfish...
  .... just with a different utility function!

Everyone is selfish!

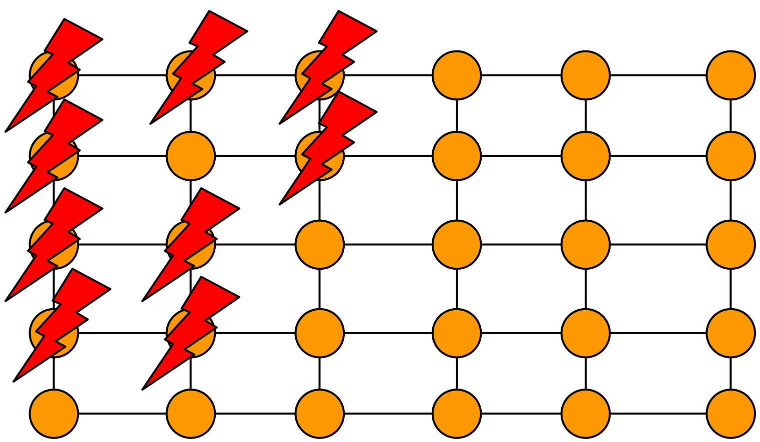
→ Difference: Malicious players‘ utility function depends inversely on the total social welfare! (not on individual ones!)

→ When studying a specific game/scenario, it makes sense to distinguish between selfish and malicious players.

Sample Analysis: Virus Inoculation Game

- Given n nodes placed in a grid (for simplicity)
- Each peer or node can choose whether to install anti-virus software
- Nodes who install the software are secure (costs 1)
- Virus spreads from a randomly selected node in the network
- All nodes in the same insecure connected component are infected (being infected costs L, L>1)

→ Every node selfishly wants to minimize its expected cost!



<u>Related Work:</u>

The VIG was first studied by Aspnes et al. [SODA'05]

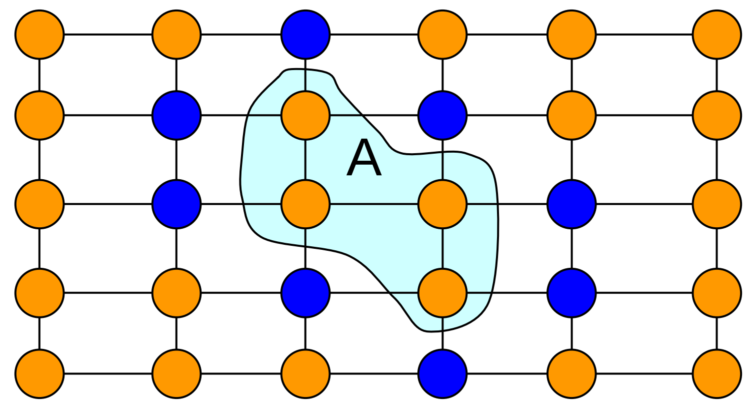
- Approximation algorithm
- General Graphs
- No malicious players

Virus Inoculation Game

- What is the impact of selfishness in the virus inoculation game?
- What is the Price of Anarchy?
- Intuition:

Expected infection cost of nodes in an insecure component A: quadratic in |A|

|A|/n * |A| * L = $|A|^2$ L/n



Total infection cost: $Cost_{inf} = \frac{L}{n} \sum_i k_i^2$

k_i: insecure nodes in the ith component

Total inoculation cost: $Cost_{inoc} = \gamma$

γ: number of secure (inoculated) nodes

Optimal Social Cost

$$Cost_{OPT} = \Theta\left(n^{2/3} L^{1/3}\right)$$

Price of Anarchy:

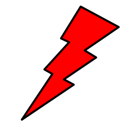
$$PoA = \Theta\left(\sqrt[3]{\frac{n}{L}}\right)$$

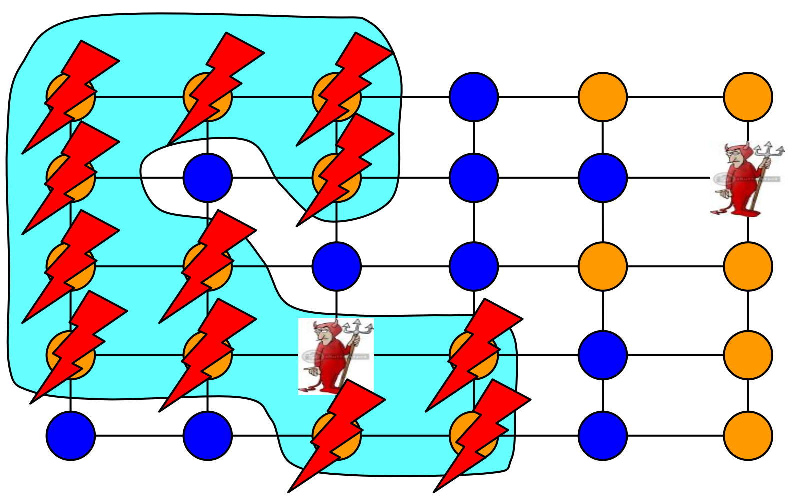
Adding Malicious Players…

- What is the impact of malicious agents in this selfish system?
- Let us add b malicious players to the grid!
- Every malicious player tries to maximize social cost!

  → Every malicious player pretends to inoculate, but does not!
- What is the Price of Malice…?

  → Depends on what nodes *know* and how they *perceive threat*!





Distinguish between:

- Oblivious model
- Non-oblivious model
  - Risk-averse

Price of Malice – Oblivious case

- Nodes do not know about the existence of malicious agents!
- They assume everyone is selfish and rational
- How much can the social cost deteriorate…?

- At most every selfish node can inoculate itself → $Cost_{inoc} \leq s$

- Total infection cost is given by: $Cost_{inf} = \frac{L}{n} \sum_i k_i \cdot l_i$

  (because component i is

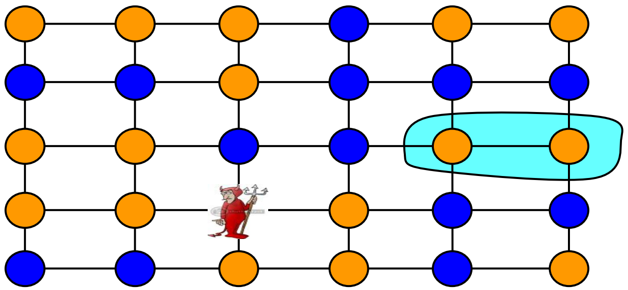
  hit with probability k_i/n)

Size of attack component i (*including* Byz.)

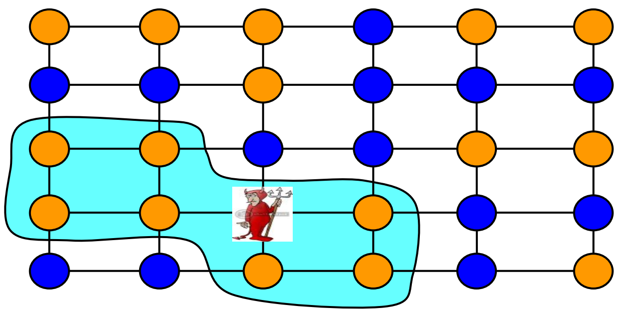
#selfish nodes in component i

Price of Malice – Oblivious case

- Total infection cost is given by: $Cost_{inf} = \frac{L}{n} \sum_i k_i \cdot l_i$
- For all components without any malicious node → $\overline{Cost_{inf}^{Byz}} \in O(s)$ (similar to analysis of PoA!)

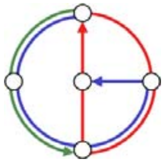


- Consider a component i with b_i>0 malicious nodes: $\sum_i b_i = b$
- In any Byz NE, the size of an attack component is at most n/L.



$$k_i \leq (b_i + 1) \cdot \frac{n}{L} + b_i$$
$$l_i \leq (b_i + 1) \cdot \frac{n}{L}.$$

it can be shown → $Cost_{inf}^{Byz} \in O\left(\frac{b^2 n}{L}\right)$



Price of Malice – Oblivious case

- Social cost is upper bounded by $O\left(s + \frac{b^2 n}{L}\right)$

- The Price of Byzantine Anarchy is at most

for b<L/2

$$PoB(b) \in \frac{O\left(s + \frac{b^2 n}{L}\right)}{\Theta(s^{2/3} L^{1/3})} \in O\left(\left(\frac{n}{L}\right)^{1/3} \cdot \left(1 + \frac{b^2}{L} + \frac{b^3}{sL}\right)\right)$$

Because PoA is $\Theta\left(\left(\frac{n}{L}\right)^{1/3}\right)$

- The Price of Malice is at most

$$PoM(b) \in O\left(1 + \frac{b^2}{L} + \frac{b^3}{sL}\right)$$

Oblivious Case Lower Bound

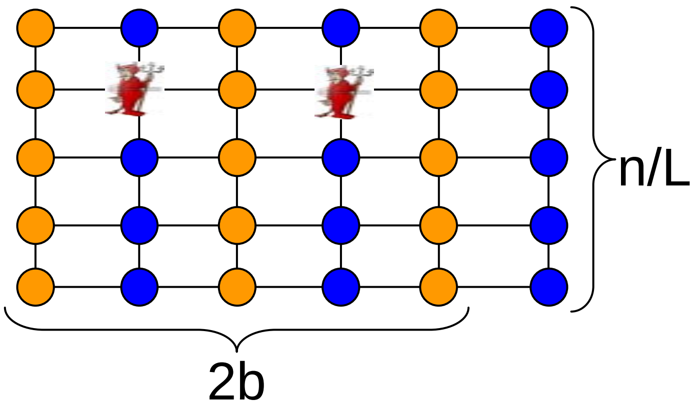
- In fact, these bounds are tight!

→ bad example: components with large surface

(Many inoculated nodes for given component size

=> bad NE! All malicious players together,

=> one large attack component => large BNE)



→ this scenario is a Byz Nash Eq.

in the oblivious case.

→ With prob. ((b+1)n/L+b)/n,

infection starts at an insecure or a malicious node of attack component of size (b+1)n/L

$$Cost_{inoc} = s/2 - b$$

→ With prob. (n/2-(b+1)n/L)/n, a component of size n/L is hit

Combining all these costs yields $\Omega\left(s + \frac{b^2 n}{L}\right)$

Price of Malice – Oblivious case

- Nodes do not know about the existence of malicious agents!
- They assume everyone is selfish and rational
- Price of Byzantine Anarchy is:

This was Price of Anarchy…

$$PoB(b) = \Theta\left(\left(\frac{s}{L}\right)^{1/3} \cdot \left(1 + \frac{b^2}{L} + \frac{b^3}{sL}\right)\right)$$

- Price of Malice is:

$$PoM(b) = \Theta\left(1 + \frac{b^2}{L} + \frac{b^3}{sL}\right)$$

- Price of Malice grows more than linearly in b
- Price of Malice is always ≥ 1

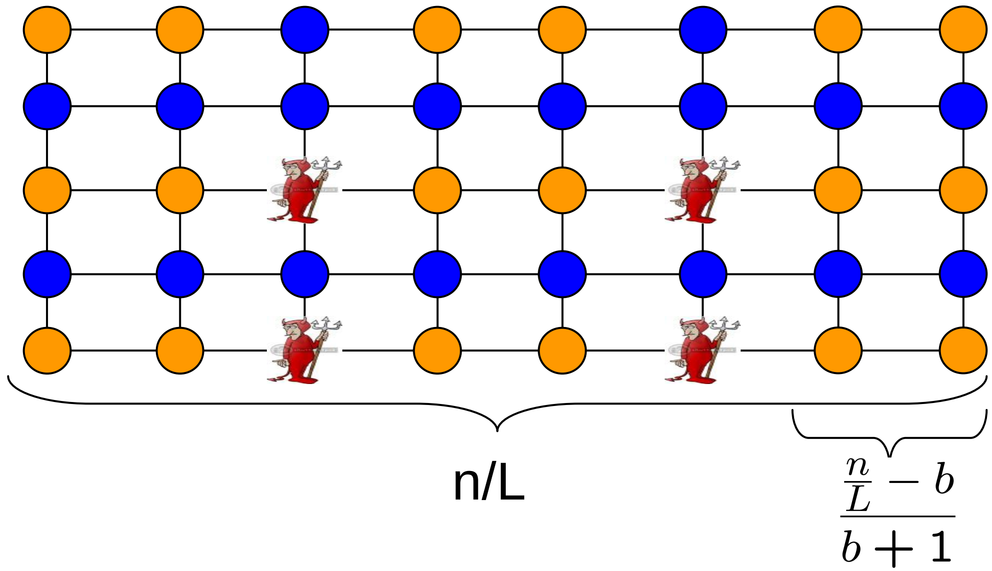
This is clear, is it…?!

→ malicious players cannot improve social welfare!

Price of Malice – Non-oblivious case

- Selfish nodes know the number of malicious agents b
- They are risk-averse
- The situation can be totally different…
- …and more complicated!
- For intuition: consider the following scenario…: more nodes inoculated!

Each player wants to minimize its maximum possible cost



This constitutes a Byzantine Nash equilibrium!

Any b agents can be removed while attack component size is at most n/L!

Conclusion

- Game-theoretic analysis

  - Large price of anarchy -> need incentive mechanism

- Byzantine game theory

  - Large price of malice -> need to do something! But what?

  - E.g., keep malicious players off from the beginning!

Future Work

- Plenty of open questions and future work!
- Virus Inoculation Game
  → The Price of Malice in more realistic network graphs
  → High-dimensional grids, small-world graphs, general graphs,…
  → How about other perceived-cost models…? (other than risk-averse)
  → How about probabilistic models…?
- The Price of Malice in other scenarios and games
  → Routing, caching, etc…
  → Can we use Fear-Factor to improve networking…?

The Last Slide!

THANK YOU !

- BitThief: How to be selfish in BitTorrent!
- Byzantine game theory: Tool to understand impact of non-cooperative behavior
- Questions and Feedback?
- Your work? Discussion?

Free Riding in BitTorrent is Cheap
Thomas Locher, Patrick Moor, Stefan Schmid, and Roger Wattenhofer.
5th Workshop on Hot Topics in Networks (HotNets), Irvine, California, USA, November 2006.
Documents: paper pdf ps slides pdf meta bibtex

When Selfish Meets Evil: Byzantine Players in a Virus Inoculation Game
Thomas Moscibroda, Stefan Schmid, and Roger Wattenhofer.
25th Annual Symposium on Principles of Distributed Computing (PODC), Denver, Colorado, USA, July 2006.
Documents: paper pdf ps meta bibtex