# What If Some Peers Are More Equal than Others?

Stefan Schmid

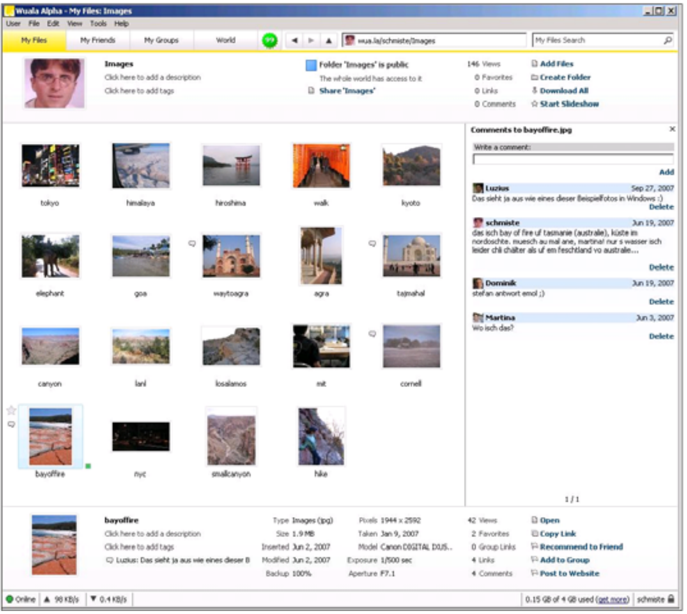
*joint work with:*
Thomas Locher, Yvonne Anne Oswald,
Dominic Meier, Patrick Moor, David Mysicka,
Christian Scheideler, Roger Wattenhofer

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

International Workshop on Mobility, Algorithms,
and Graph Theory in Dynamic Networks (IMAGINE)
Reykjavik, Iceland
July 12, 2008

# Peer-to-Peer Technology

- Well-known p2p systems
  - Internet telephony: *Skype*
  - File sharing: *BitTorrent, eMule, ...*
  - Streaming: *Zattoo, Joost, ...*



Images
Click here to add a description
Click here to add tags

Folder 'Images' is public
The whole world has access to it

146 Views
0 Favorites
0 Links
0 Comments

Comments to bayoffire.jpg
Write a comment:

Luzius Sep 27, 2007
Das sieht ja aus wie eines dieser Beispielfotos in Windows :)

schmiste Jun 19, 2007
das isch bay of fire uf tasmanie (australie), küste im nordoschte. muesch au mal ane, martina! nur s wasser isch leider chli chälter als uf em feschtland vo australie...

Dominik Jun 19, 2007
stefan antwort emol ;)

Martina Jun 3, 2007
Wo isch das?

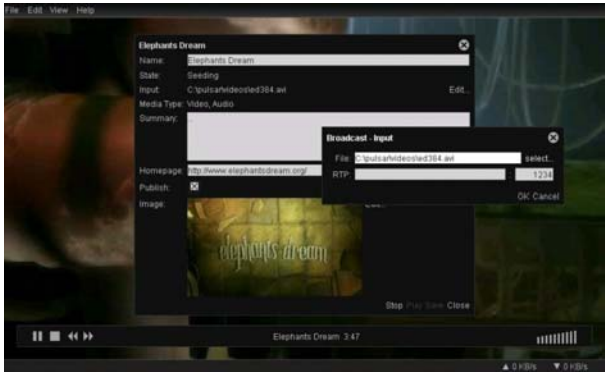
bayoffire
Click here to add a description
Click here to add tags
Luzius: Das sieht ja aus wie eines dieser B

Type Images (jpg)
Size 1.9 MB
Inserted Jun 2, 2007
Modified Jun 2, 2007
Backup 100%

Pixels 1944 x 2592
Taken Jan 9, 2007
Model Canon DIGITAL IXUS...
Exposure 1/500 sec
Aperture F7.1

42 Views
2 Favorites
0 Group Links
4 Links
4 Comments

● Online ▲ 98 KB/s ▼ 0.4 KB/s 0.15 GB of 4 GB used (get more) schmiste



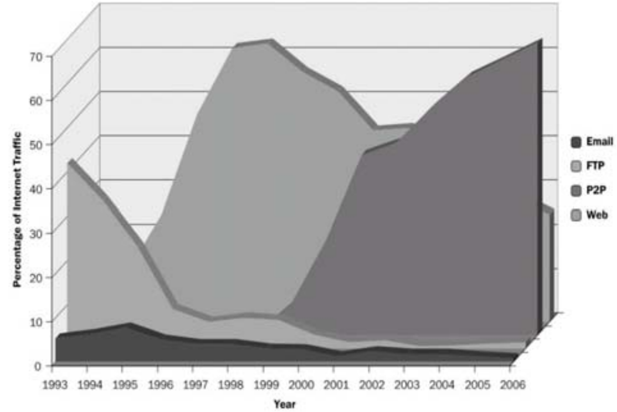
Elephants Dream
Name: Elephants Dream
State: Seeding
Input: C:\pulsar\videos\ed384.avi
Media Type: Video, Audio
Summary:
Homepage: http://www.elephantsdream.org/
Publish: ☒
Image:

Broadcast - Input
File: C:\pulsar\videos\ed384.avi
RTP: 1234

- Other (well-known?) systems
  - Pulsar streaming system (e.g., *tilllate* clips?)
  - Wuala online storage system

- Impact: Accounts for much Internet traffic!

  (source: *cachelogic.com*)



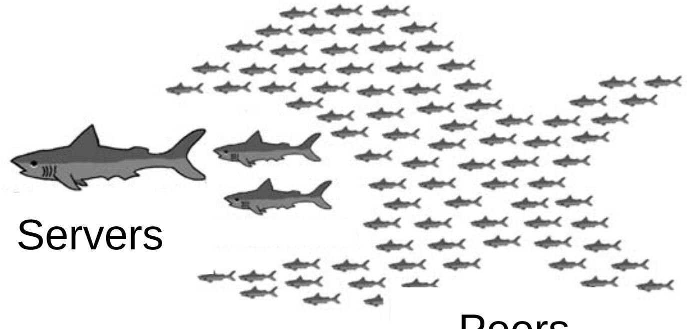
# The Paradigm

- Key concepts

  - Machines (peers) in the network: consumer and producer of resources
  - Use of decentralized resources on the edge of the Internet (e.g., desktops)

- Benefits

  - Scalability: More resources in larger networks („cake grows“)
  - Robustness: No single point of failure
  - Can outperform server-based solutions
  - Cheap: start-up vs Google



- Therefore:

  - No need for expensive infrastructure at content distributors
  - Democratic aspect: Anyone can publish media contents / speeches

# A Challenge

- In practice, peer-to-peer is not synonym for „from equal to equal“
  - Rather some peers may be „more equal than others“!

- E.g.
  - Some peers want to be consumers only (but not producers) of resources
  - Some peers may be malicious
  - Some peers may be social
  - Different capabilities (e.g., better Internet connection)

- These differences must not be ignored
  - E.g., punish selfish behavior
  - E.g., ensure efficiency despite heterogeneity

# State of the Art

- Peer-to-peer systems: no effective solutions for many inequality problems today

- Example 1: BitThief client downloads entire files from BitTorrent without uploading

- Example 2: Censorship attacks in the Kad network (malicious peer)

  - Peer assumes corresponding IDs

- Example 3: Solutions for heterogeneity challenge often simplistic

  - Cheated incentive mechanism: Kazaa Lite client hardwires user contribution to maximum

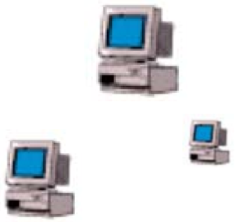
  - Limited heterogeneity: two peer type approach of Gnutella or Kazaa

# Talk Outline

- Case Study 1: Non-Cooperation in BitTorrent Swarms *(HotNets 2006)*

- Case Study 2: Malicious Peers in the Kad Network *(under submission)*

- Analysis of Social Behavior in Peer-to-Peer Systems *(EC 2008)*



- SHELL: A Heterogeneous Overlay Architecture *(ongoing work)*

- Conclusion and Research Problems

# Case Study BitThief:
# Free-riding Peers in BitTorrent

# BitThief: BitTorrent

- BitTorrent = one of the most popular p2p systems
  - Millions of simultaneous users

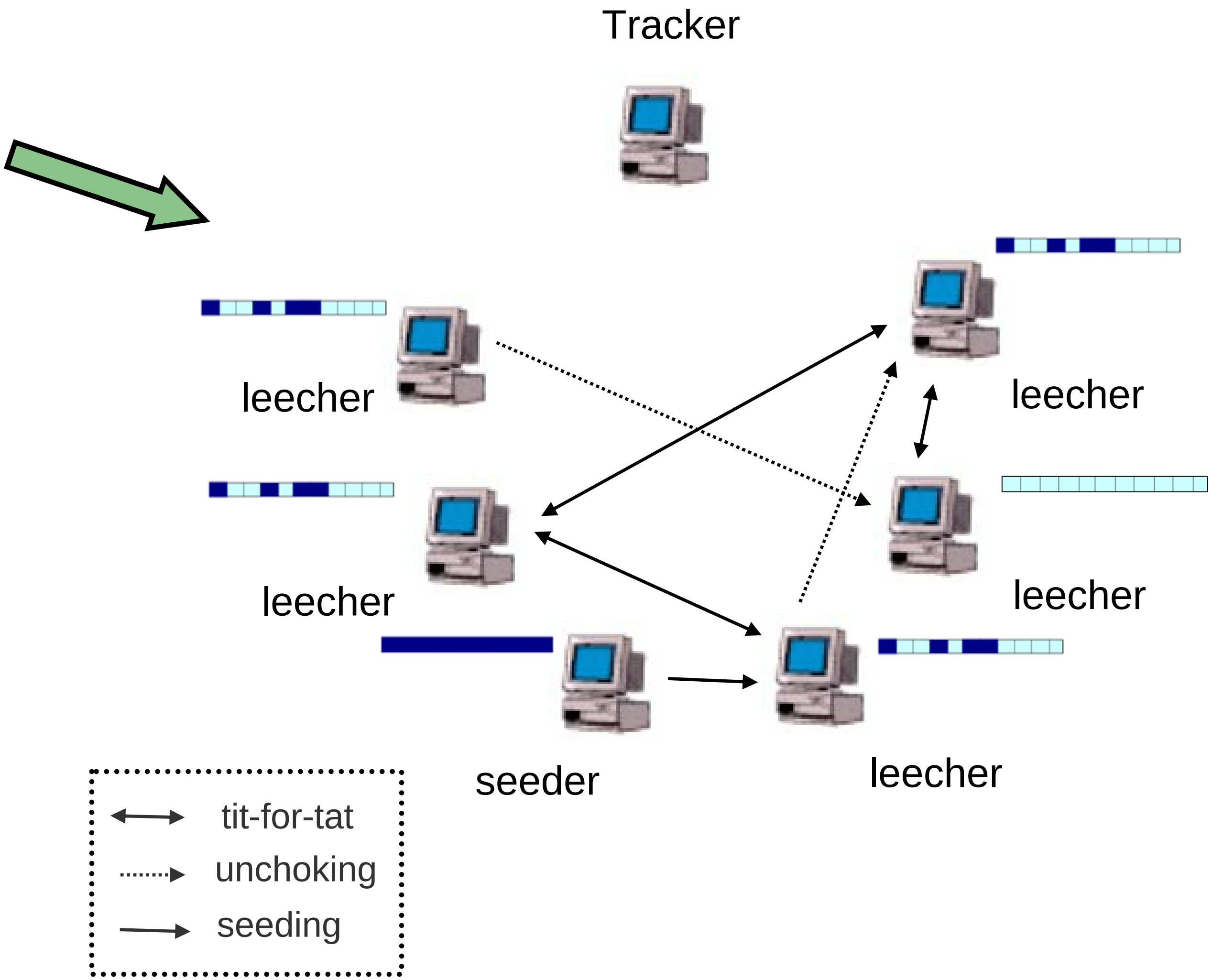
- One of the few systems incorporating incentive mechanism

- Basic principle
  - Peers interested in same file are organized by a tracker in a swarm
  - File is divided into pieces (or „blocks“)
  - Distinguish between seeders (entire file) and leechers (not all pieces)
  - Peers have different pieces which are exchanged in a tit-for-tat like manner
  - Bootstrap problem: peers optimistically unchoke neighbors (round-robin = give some pieces „for free“)

# BitThief: BitTorrent Swarms

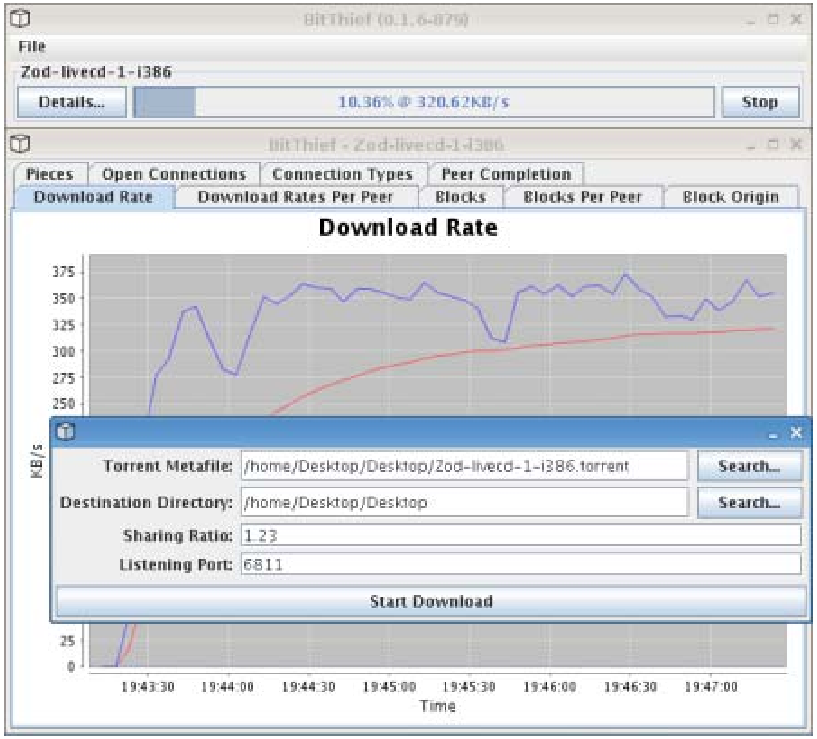
website with .torrent file

- tracker address
- verification data
- ….



# BitThief: Goal

BitThief = proof of concept Java client (implemented from scratch) which achieves fast downloads without uploading at all – in spite of BitTorrent‘s incentive mechanism!



Zod-livecd-1-i386

10.36% @ 320.62KB/s

Torrent Metafile: /home/Desktop/Desktop/Zod-livecd-1-i386.torrent

Destination Directory: /home/Desktop/Desktop

Sharing Ratio: 1.23

Listening Port: 6811

# BitThief: Tricks

BitThief‘s three tricks:

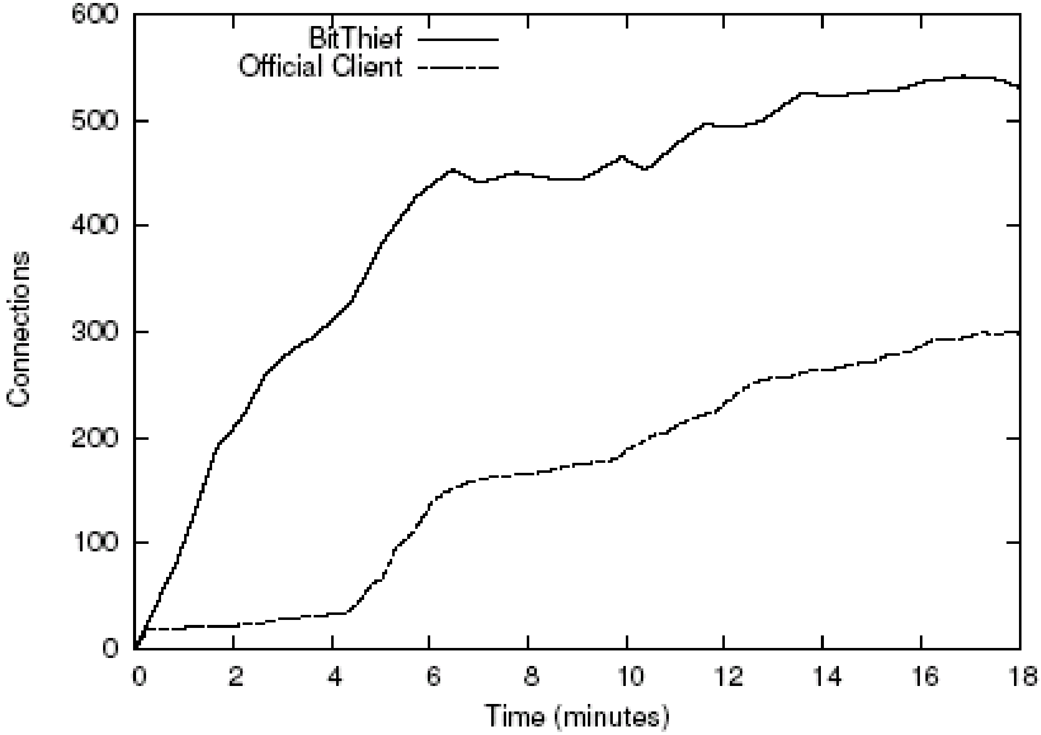
- Open as many TCP connections as possible
- Contacting tracker again and again, asking for more peers (never banned!)
- Pretend being a great uploader in sharing communities

=> Exploit optimistic unchoking

=> Exploit seeders

=> Exploit sharing communities

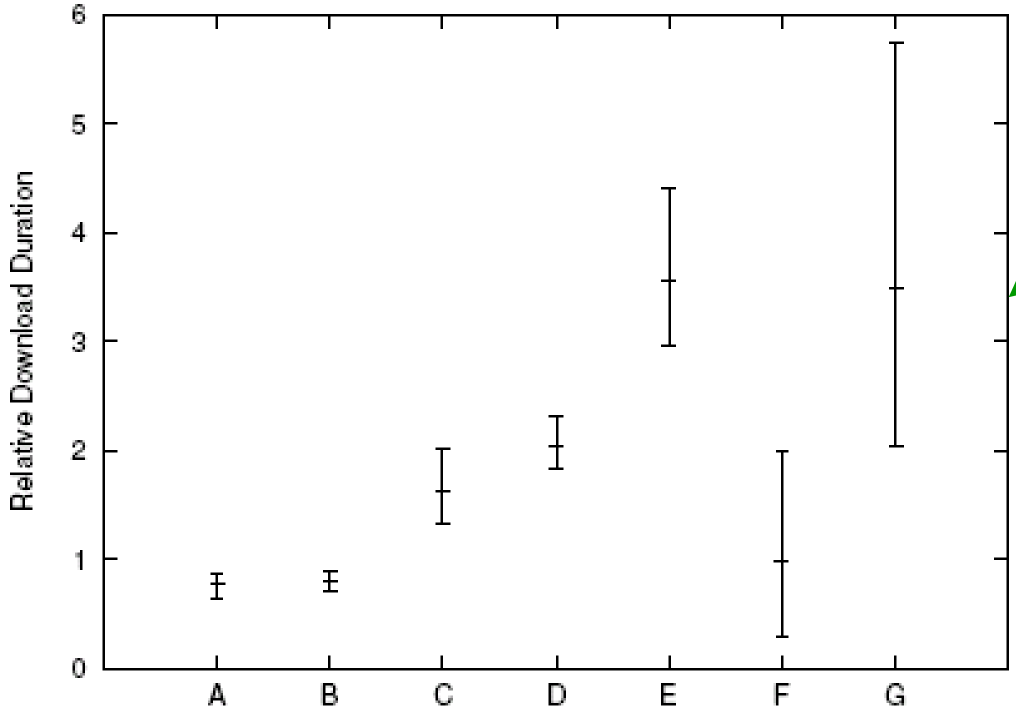
# BitThief: Connect to More Neighbors…



# BitThief:  Results (with Seeders)

2

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

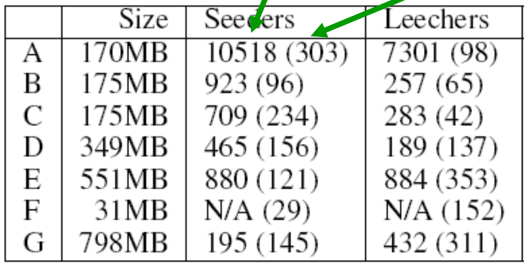


4 BitThief with public
IP and open
TCP port

number of peers
announced
by tracker

max
peers found
by BitThief

1

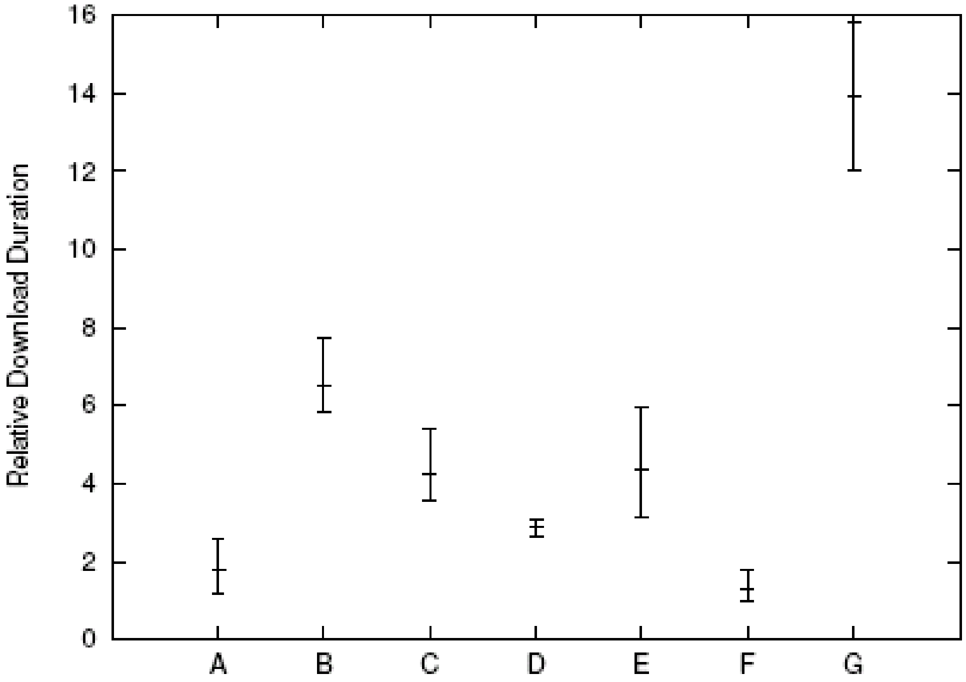


|   | Size | Seeders | Leechers |
|---|---|---|---|
| A | 170MB | 10518 (303) | 7301 (98) |
| B | 175MB | 923 (96) | 257 (65) |
| C | 175MB | 709 (234) | 283 (42) |
| D | 349MB | 465 (156) | 189 (137) |
| E | 551MB | 880 (121) | 884 (353) |
| F | 31MB | N/A (29) | N/A (152) |
| G | 798MB | 195 (145)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

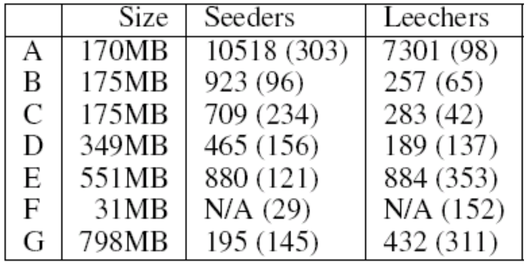
3

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

# BitThief: Results (without Seeders)



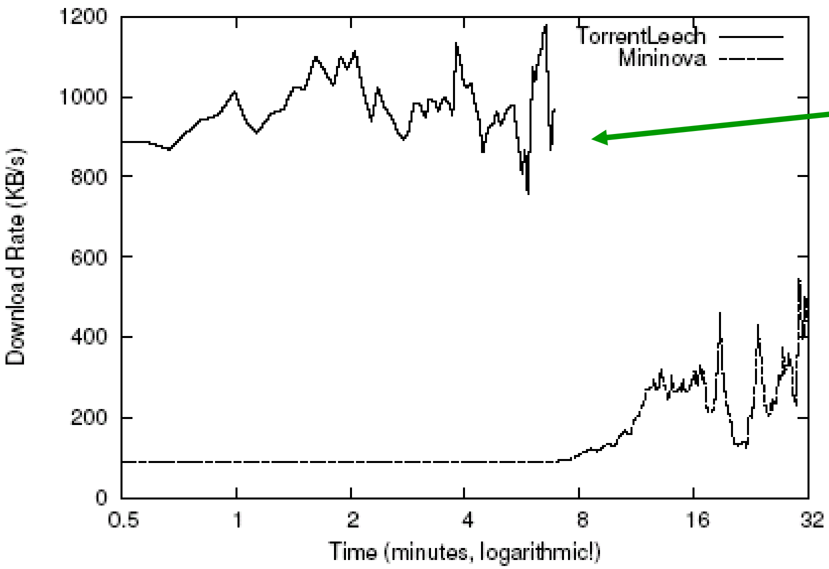
- Seeders detected with bitmask / have-message
- Even without seeder it‘s fast!
- Unfair test: Mainline client was allowed to use seeders!



| | 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) |

# BitThief: Sharing Communities (1)

- Closed / private swarm
  - Tracker requires user registration
  - Monitors contributions, bans peers with low sharing ratios
- Client can report uploaded data itself! (tracker announcements)
  - As tracker does not verify, it‘s easy to remain in community...
  - ... and communities are often a paradise for BitThief.



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

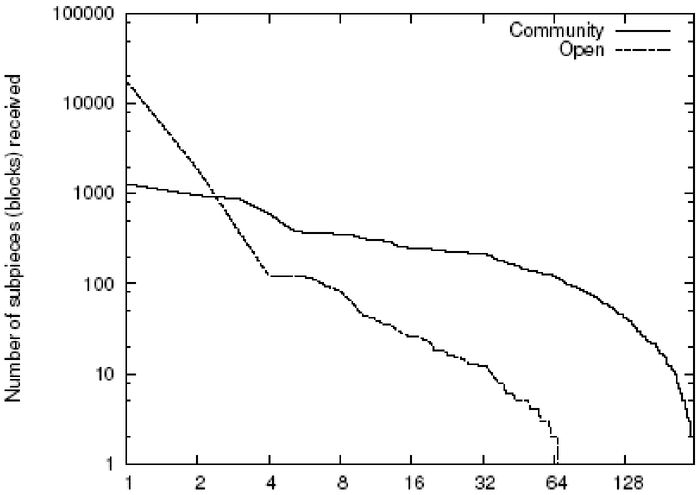
# BitThief: Sharing Communities (2)

- In communities, contribution is more balanced

- Reason?

  - Peers want to boost ratio?

  - Users more tech-savvy? (less firewalled peers? faster network connections?)



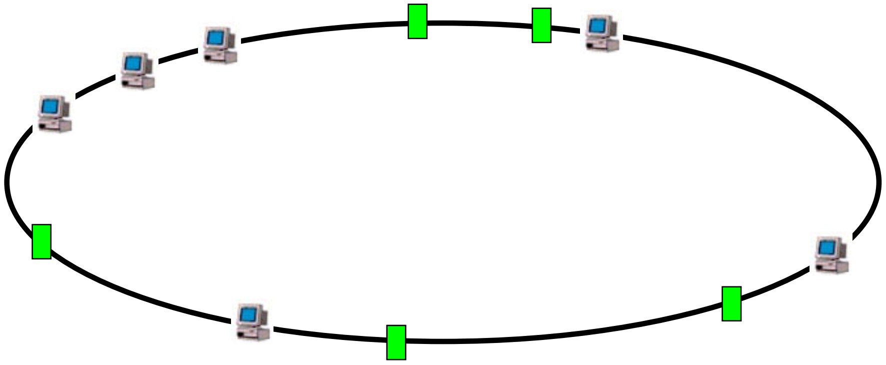
# Case Study Kad: Censorship in Kad

Under submission / PhD thesis

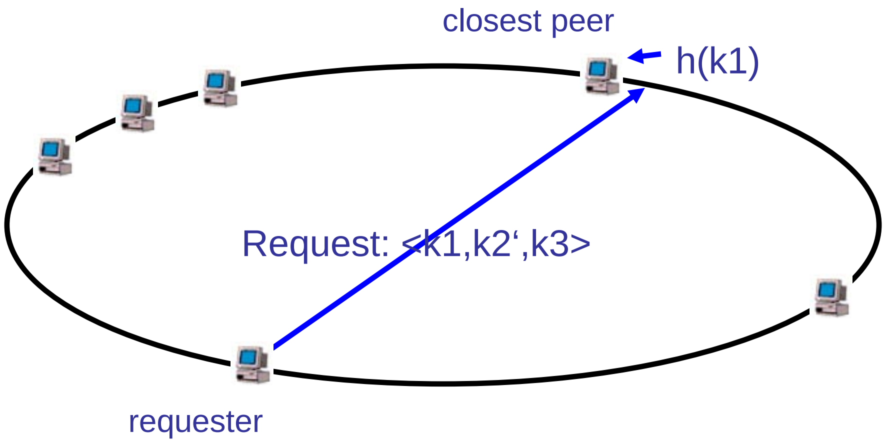
# Kad: The Kad Network

- Kad = one of the first widely used distributed hash tables (DHT)
  - A structured peer-to-peer system where the index is stored distributedly
  - In literature, DHTs have been studied for years (Chord, Pastry, etc.)

- Basic principle
  - Consistent hashing
  - Peers and data items with identifiers chosen from [0,1)
  - (Pointers to) data items stored on closest peers

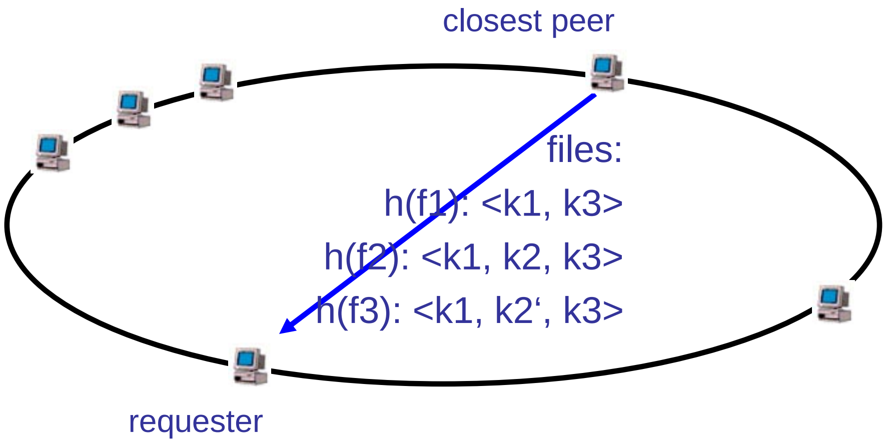


# Kad: Keyword Request



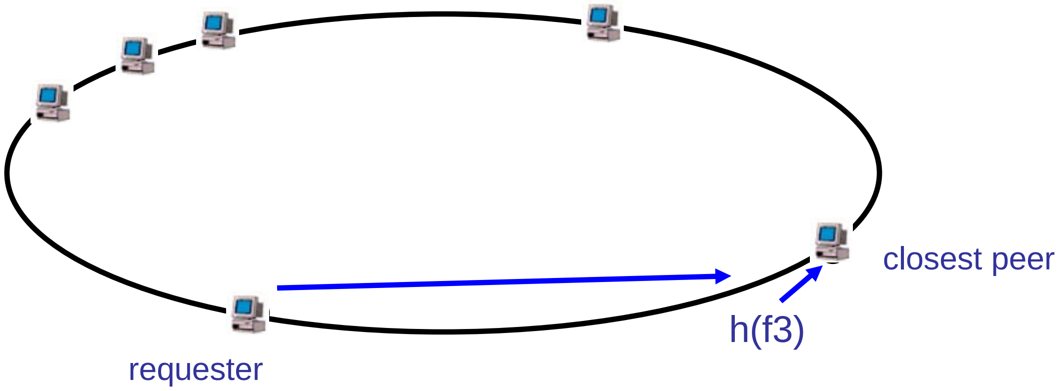
Lookup only with first keyword in list. Key is hash function on this keyword, will be routed to peer with Kad ID closest to this hash value. This peer is responsible for files stored with this first keyword.

# Kad: Keyword Request



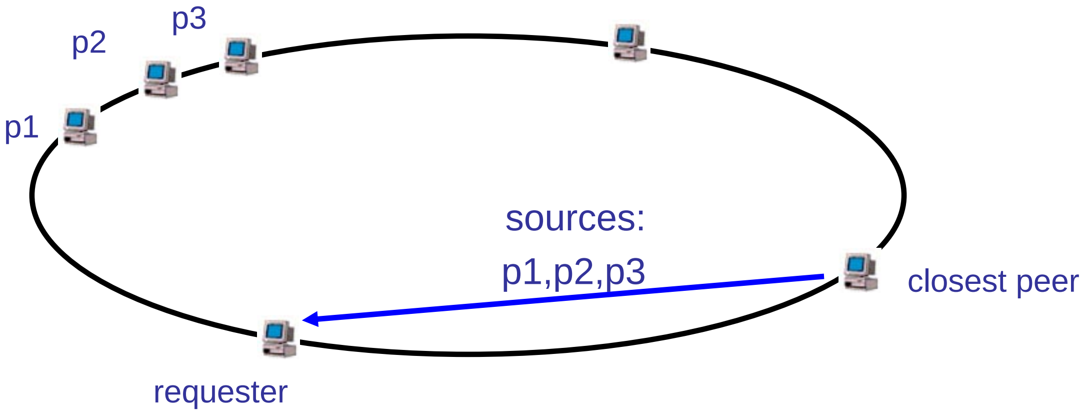
Peer responsible for this keyword returns different sources (hash keys) together with keywords.

# Kad: Source Request



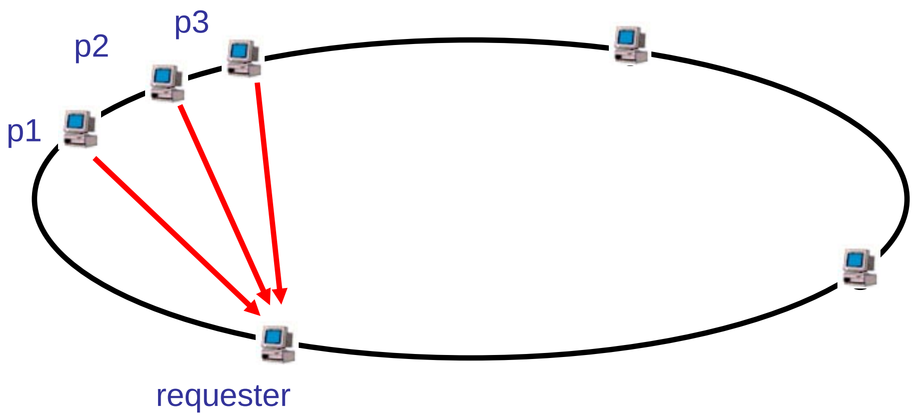
Peer can use this hash to find peer responsible for the file.

# Kad: Source Request



Peer provides requester with a list of peers storing a copy of the file.

# Kad: Download

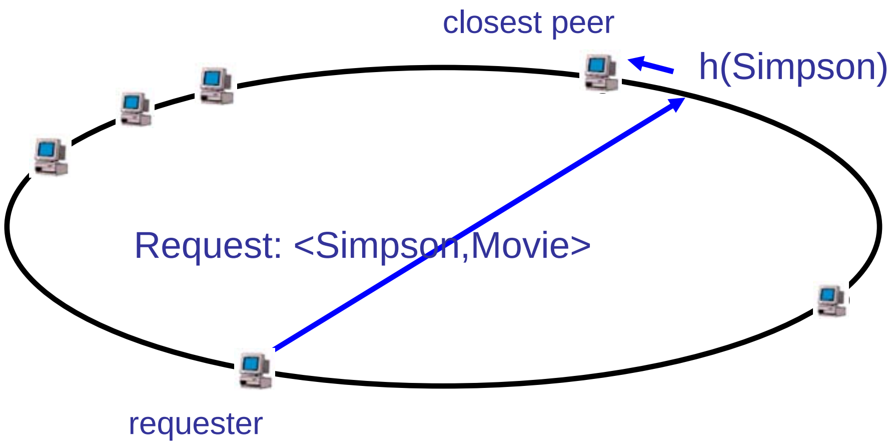


Eventually, the requester can download the data from these peers.

# Kad: Censorship

- Kad network has several vulnerabilities

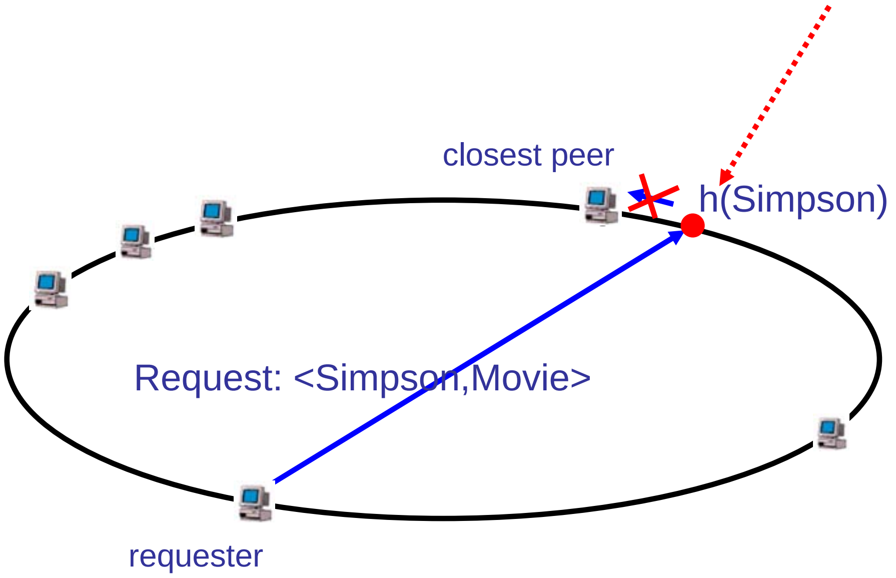
- Example: malicious peers can perform censorhip attack
  - Simply by assuming the corresponding IDs (peer insertion attack)
  - No prescribed ID selection method or verification



# Kad: Censorship

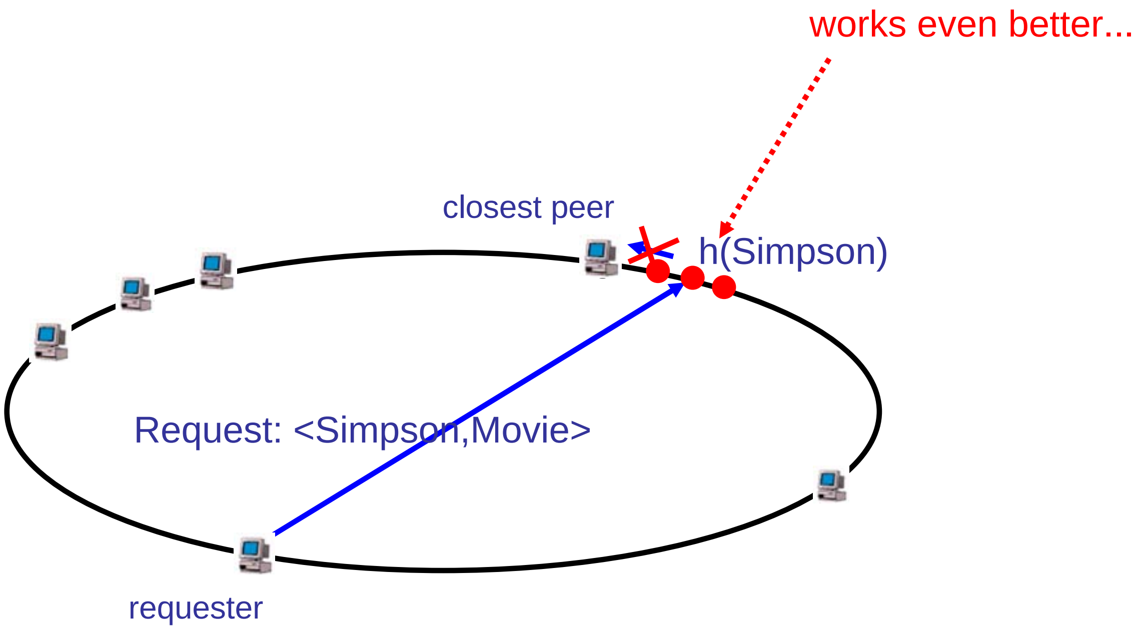
- Censoring contents in Kad

If peer is inserted here, it can block (or spy on) keyword requests for „Simpsons“, „Simpsons Movie“, etc.



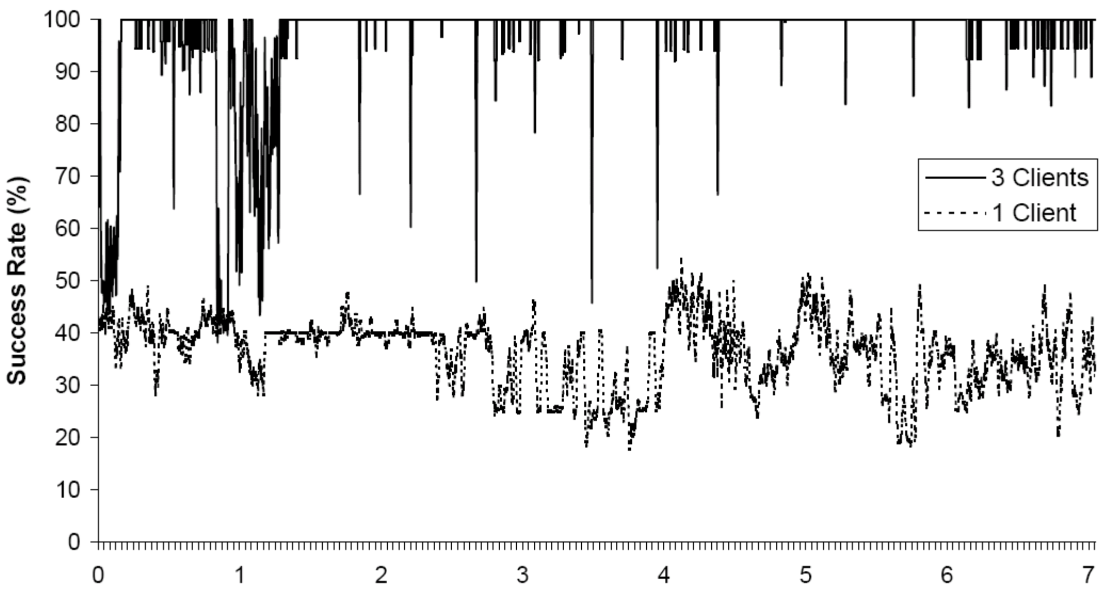
# Kad: Censorship

- Censoring contents in Kad



# Kad: Censorship

- Some results



- Similarly for source requests
- There are also other censorship attacks (e.g., pollute cache of other peers)
- Plus eclipse and denial of service attacks (e.g., pollute cache such that requests are forwarded to external peers)...

# Easy to Fix?

# BitThief and Kad Attacks

- BitThief

  - Optimistic unchoking can be exploited

  - Just do pure tit-for-tat? Bootstrap problem...

  - Fast extension: subset of pieces only (limited „venture capital“)

  - What if participants are not directly interested in each other?

- Kad Attacks

  - Do not accept too much information from same peer (e.g., publish attack)

  - Bind ID to peer... But how?

  - Bind to IP? But what about NATs where many peers have same ID? And what about dynamic IP addresses? Lose credits?

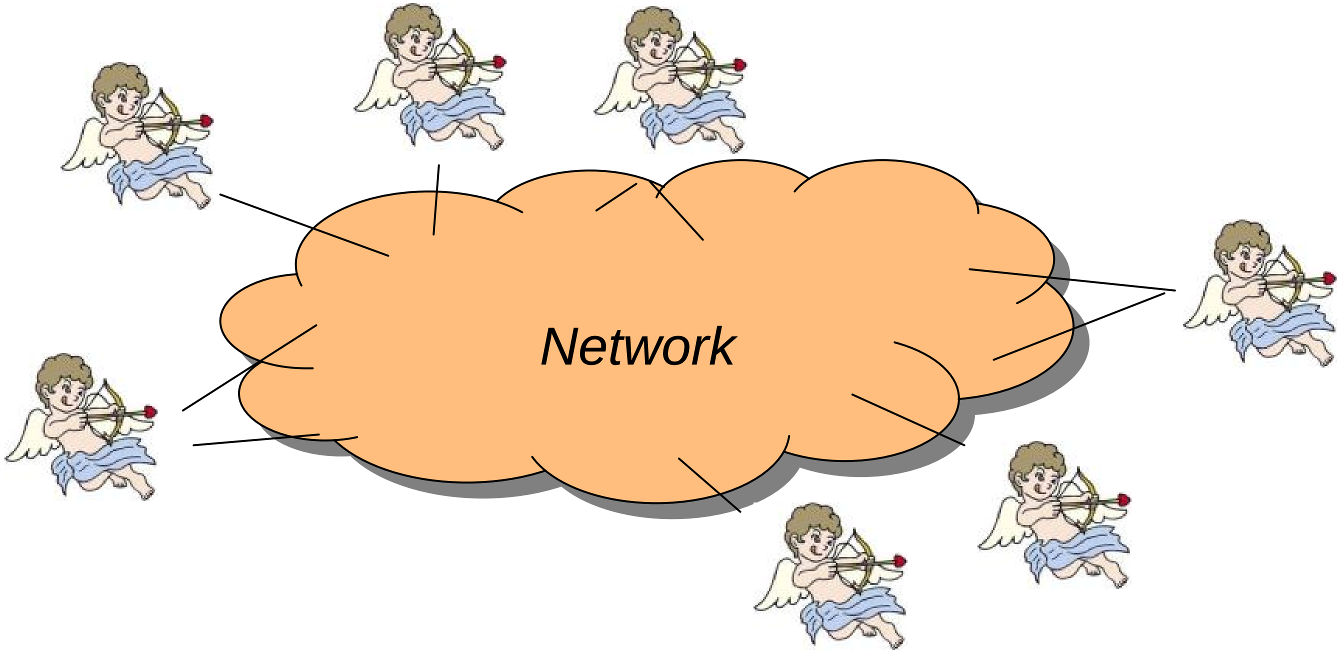
  - Generate ID, e.g., by hashing a user phrase? But due to sparsely populated ID space, it‘s still easy to generate IDs close to the object...

# What is the Impact?
# (Extended) Game Theory…

# Modelling Peers (1)

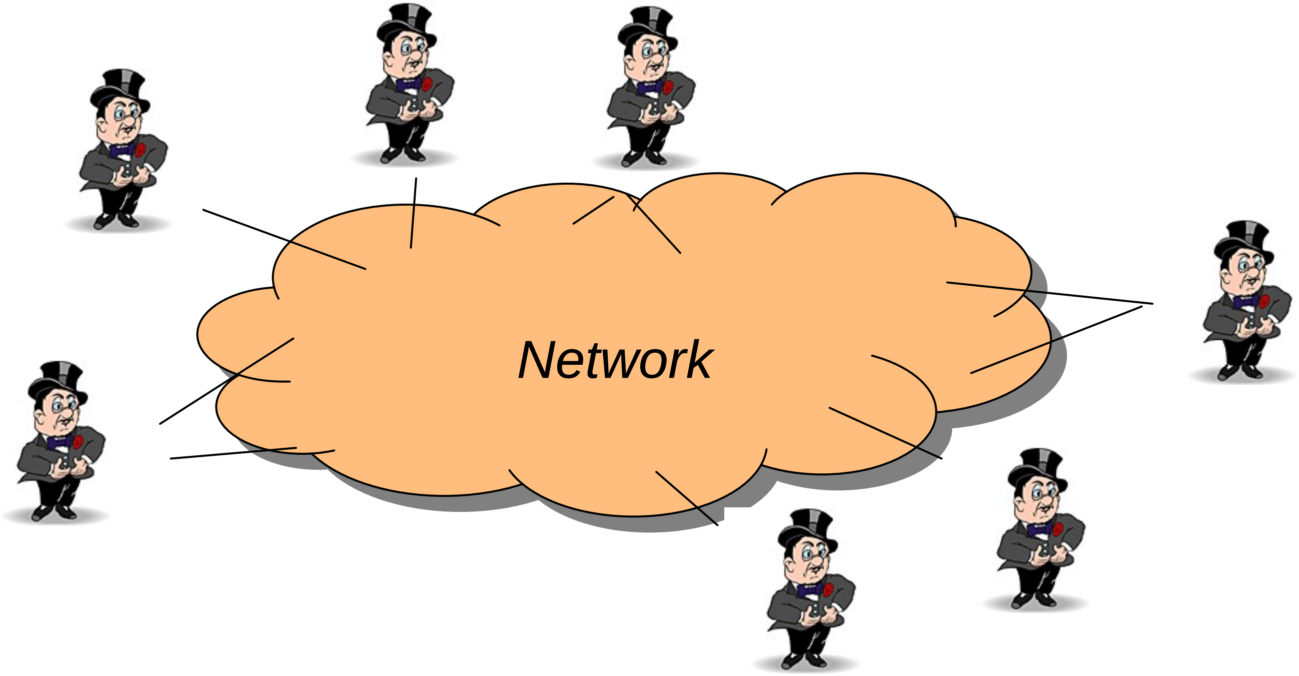
- Game theory is formalism to study uncooperative behavior
  - mainly selfish individuals (e.g., Price of Anarchy)

- Model for peer-to-peer network?



# Modelling Peers (2)

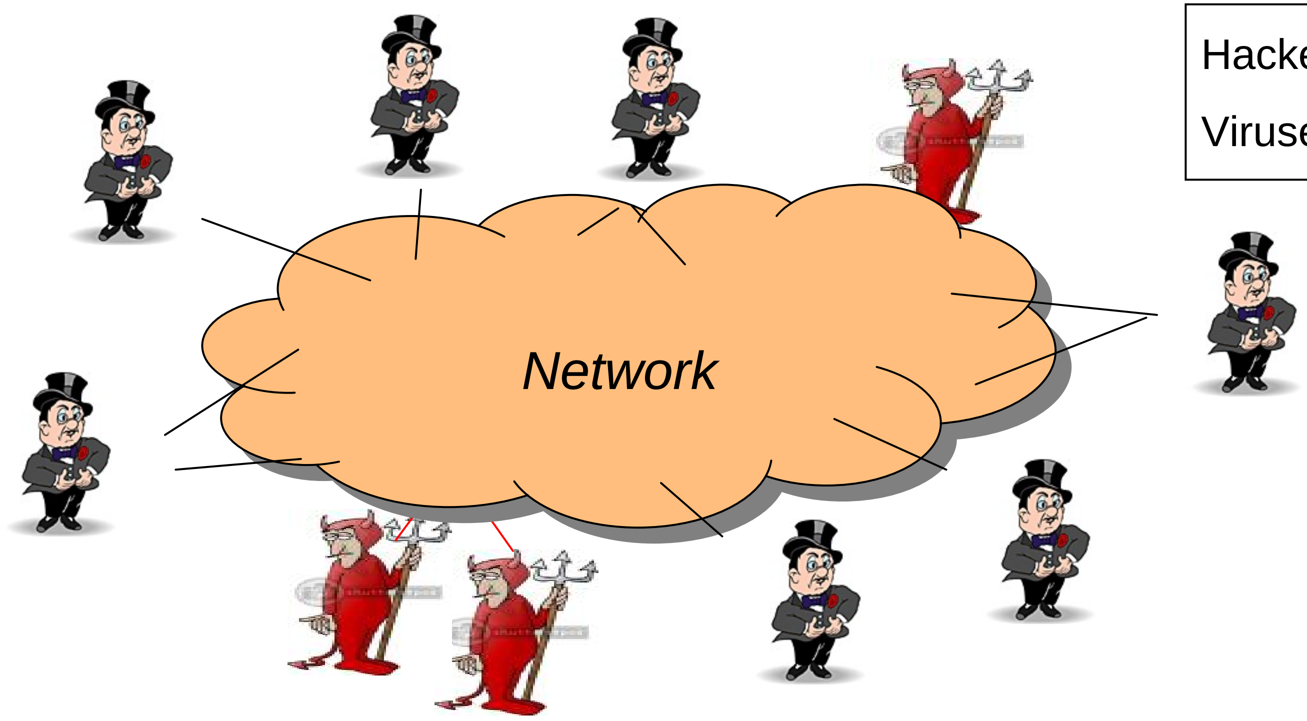
- Game theory models participants as selfish players
  - Seek to maximize their utility



# Modelling Peers (3)

- We extended this model and introduced malicious players

  - seek to minimize social welfare



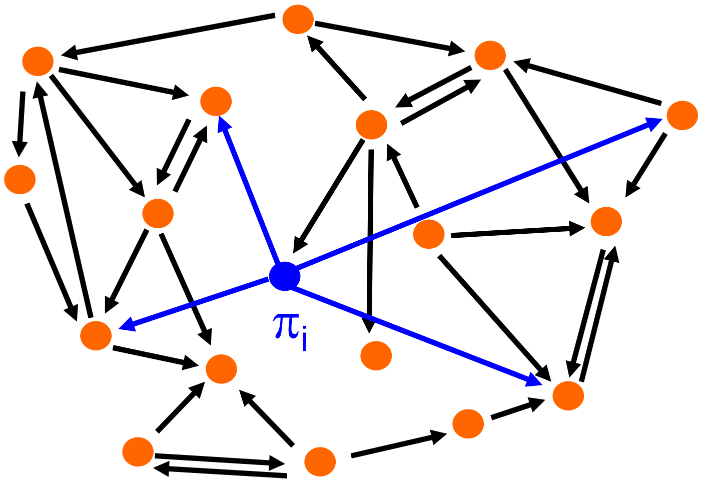
Hackers, Polluters,

Viruses, DOS attacks

# Impact of Selfish Players

- Study of strategic behavior in unstructured peer-to-peer topologies

- Some results of network creation game (PODC 2006)
  - Price of Anarchy can be large
  - Nash equilibria may not exist (instability!)
  - NP-hard to decide whether a given network will stabilize



# Impact of Malicious Players

- What is impact of malicious players in selfish networks?

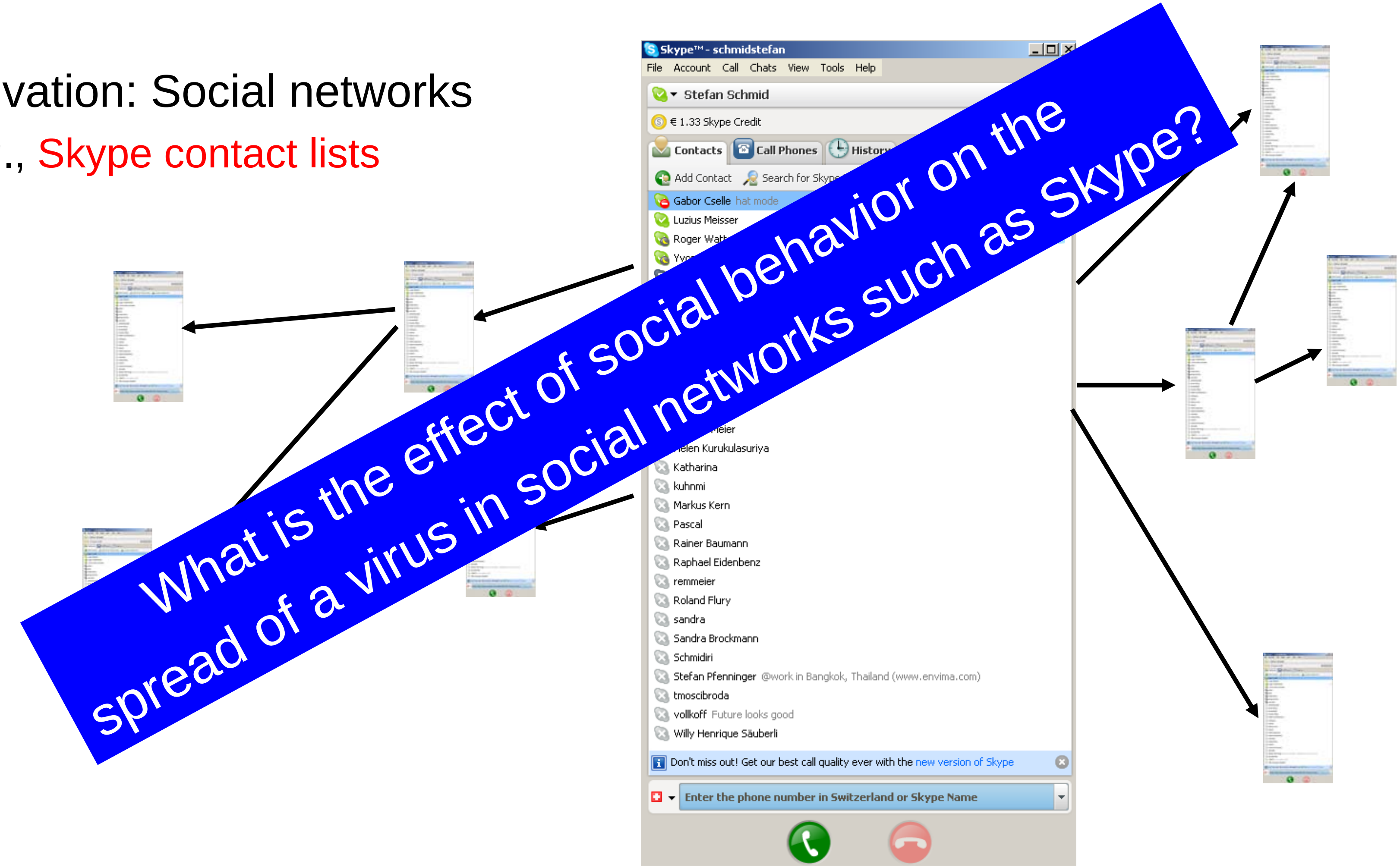
- Depends on
  - Knowledge of selfish players on malicious players
  - How selfish players react to this knowledge (neutral, risk-averse, etc.)

- Some results (PODC 2006) for a virus inoculation game
  - If selfish players are oblivious, malicious players reduce social welfare
  - If players non-oblivious and risk-averse, social welfare may improve!
  - Phenomenon called fear factor or windfall of malice

# Impact of Social Players?

- In the following, we want to study social peers

- Motivation: Social networks
  - E.g., Skype contact lists



What is the effect of social behavior on the spread of a virus in social networks such as Skype?

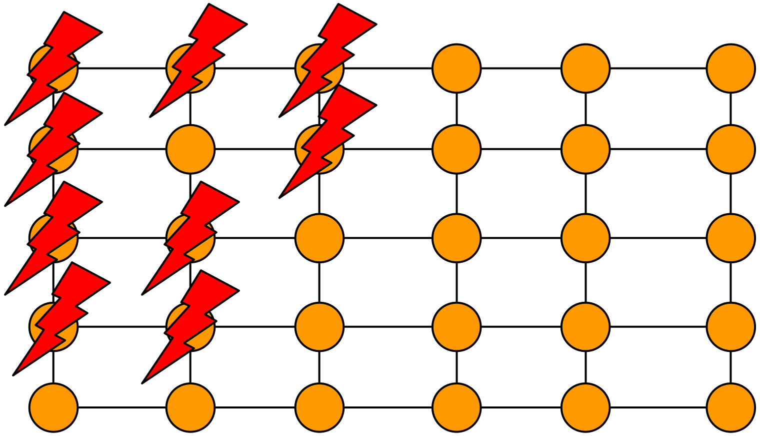
Gabor Cselle hat mode
Luzius Meisser
Roger Watt
Yvo
...eier
...elen Kurukulasuriya
Katharina
kuhnmi
Markus Kern
Pascal
Rainer Baumann
Raphael Eidenbenz
remmeier
Roland Flury
sandra
Sandra Brockmann
Schmidiri
Stefan Pfenninger @work in Bangkok, Thailand (www.envima.com)
tmoscibroda
vollkoff Future looks good
Willy Henrique Säuberli

Don't miss out! Get our best call quality ever with the new version of Skype

# A Sample Game

- Sample game: virus inoculation

- The game
  - Network of n peers (or players)
  - Decide whether to inoculate or not
  - Inoculation costs C
  - If a peer is infected, it will cost L>C



- At runtime: virus breaks out at a random player, and (recursively) infects all insecure adjacent players

# Modelling Peers...

- Peers are selfish, maximize utility

- However, utility takes into account friends‘ utility
  - „local game theory“

- Utility / cost function of a player
  - Actual individual cost:

$k_i$ = attack component size

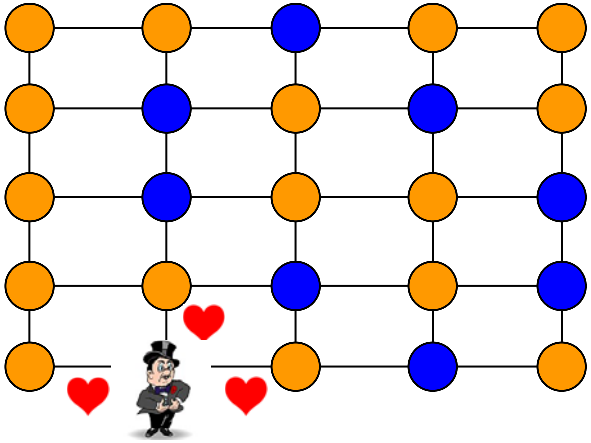
$$c_a(i,\vec{a}) = a_i \cdot C + (1-a_i)L \cdot \frac{k_i}{n}$$

$a_i$ = inoculated?

  - Perceived individual cost:

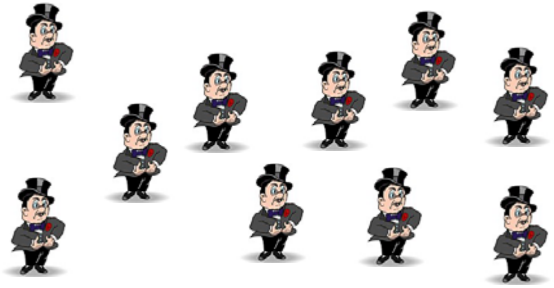
$$c_p(i,\vec{a}) = c_a(i,\vec{a}) + F \cdot \sum_{p_j \in \Gamma(p_i)} c_a(j,\vec{a})$$

F = friendship factor,
extent to which players care about friends



# Social Costs and Equilibria

- In order to quantify effects of social behavior...



- Social costs
  - Sum over all players‘ actual costs

- Nash equilibria
  - Strategy profile where each player cannot improve her welfare...
  - ... given the strategies of the other players
  - Nash equilibrium (NE): scenario where all players are selfish
  - Friendship Nash equilibrium (FNE): social scenario
  - FNE defined with respect to perceived costs!
  - Typical assumption: selfish players end up in such an equilibrium (if it exists)

# Evaluation

- What is the impact of social behavior?

- Windfall of friendship

  - Compare (social cost of) worst NE where every player is selfish (perceived costs = actual costs)...

  - ... to worst FNE where players take friends‘ actual costs into account with a factor F (players are „social“)

# Windfall of Friendship

- Formally, the windfall of friendship (WoF) is defined as

$$\Upsilon(F,I) = \frac{\max_{NE} C_{NE}(I)}{\max_{FNE} C_{FNE}(F,I)}$$

instance I describes graph, C and L

- WoF >> 1 => system benefits from social aspect

  - Social welfare increased

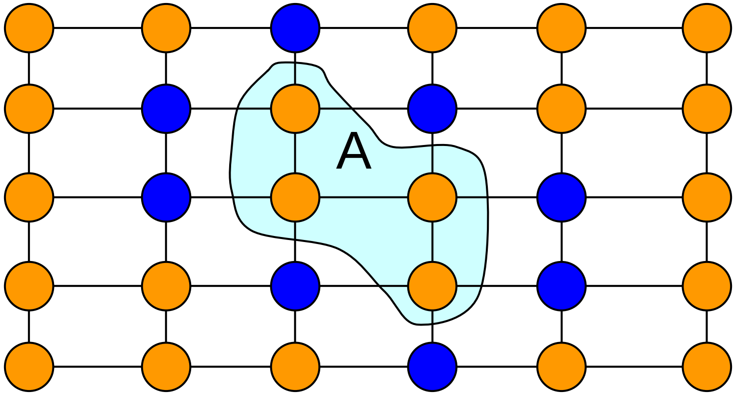
- WoF < 1 => social aspect harmful

  - Social welfare reduced

# Characterization of NE

- In regular (and pure) NE, it holds that...

- Insecure player is in attack component A of size at most Cn/L
  - otherwise, infection cost
    > (Cn/L)/n * L = C

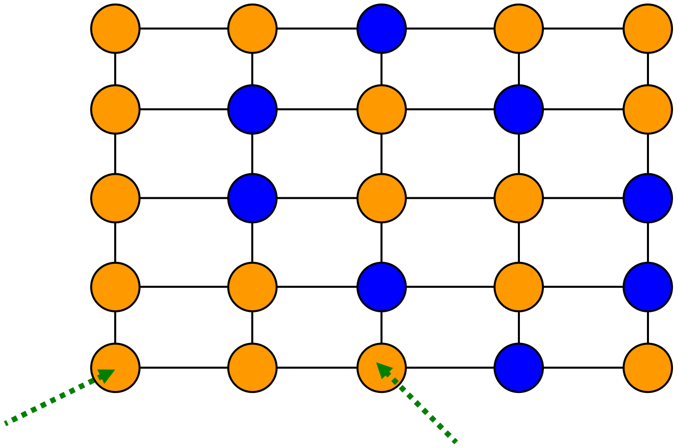


- Secure player: if she became insecure, she would be in attack component of size at least Cn/L
  - same argument: otherwise it‘s worthwhile to change strategies

# Characterization of Friendship Nash Equilibria

- In friendship Nash equilibria, the situation is more complex

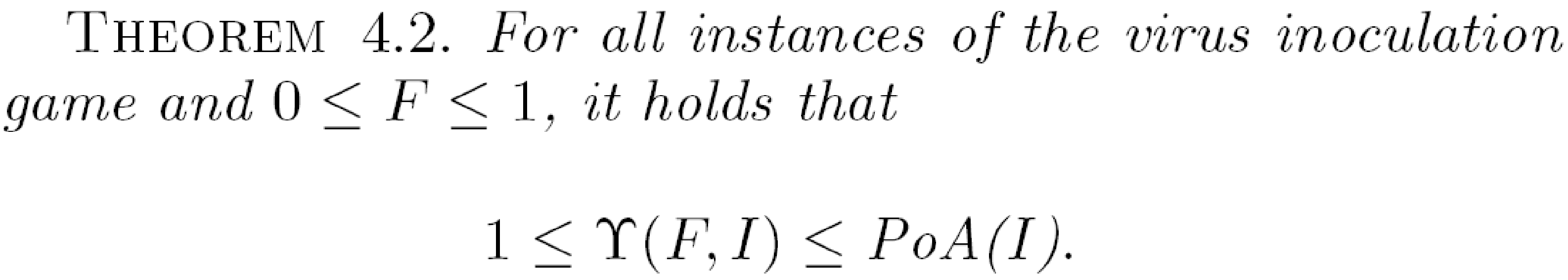
- E.g., problem is asymmetric
  - One insecure player in attack component may be happy...
  - ... while other player in same component is not
  - Reason: second player may have more insecure neighbors



not happy, two insecure neighbors
(with same actual costs)

happy, only one insecure neighbor
(with same actual costs)

# Bounds for the Windfall



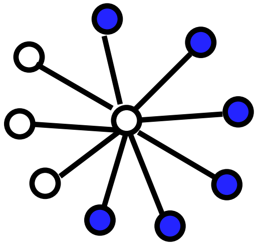
THEOREM 4.2. *For all instances of the virus inoculation game and $0 \leq F \leq 1$, it holds that*

$$1 \leq \Upsilon(F, I) \leq PoA(I).$$

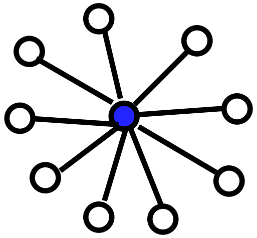
- It is always beneficial when players are social!
- The windfall can never be larger than the price of anarchy
  - Price of anarchy = ratio of worst Nash equilibrium cost divided by social optimum cost
- Actually, there are problem instances (with large F) which indeed have a windfall of this magnitude („tight bounds“, e.g., star network)

# Example for Star Graph

- In regular NE, there is always a (worst) equilibrium where center is insecure, i.e., we have n/L insecure nodes and n-n/L secure nodes (for C=1):



Social cost = (n/L)/n * n/L * L + (n-n/L) ~ n



- In friendship Nash equilibrium, there are situations where center *must* inoculate, yielding optimal social costs of (for C=1):

Social cost = „social optimum“
= 1 + (n-1)/n * L ~ L

WoF as large as maximal price of anarchy in arbitrary graphs (i.e., n for constant L).

# A Proof Idea for Lower Bound

- WoF $\geq$ 1 because...:

- Consider arbitrary FNE (for any F)

- From this FNE, we can construct (by a best response strategy) a regular NE with at least as large social costs
  - Component size can only increase: peers become insecure, but not secure
  - Due to symmetry, a player who joins the attack component (i.e., becomes insecure) will not trigger others to become secure
  - It is easy to see that this yields larger social costs

- In a sense, this result matches our intuitive expectations...

# Monotonicity

But the windfall does not increase monotonously:
WoF can decline when players care more about their friends!

- Example again in simple star graph...

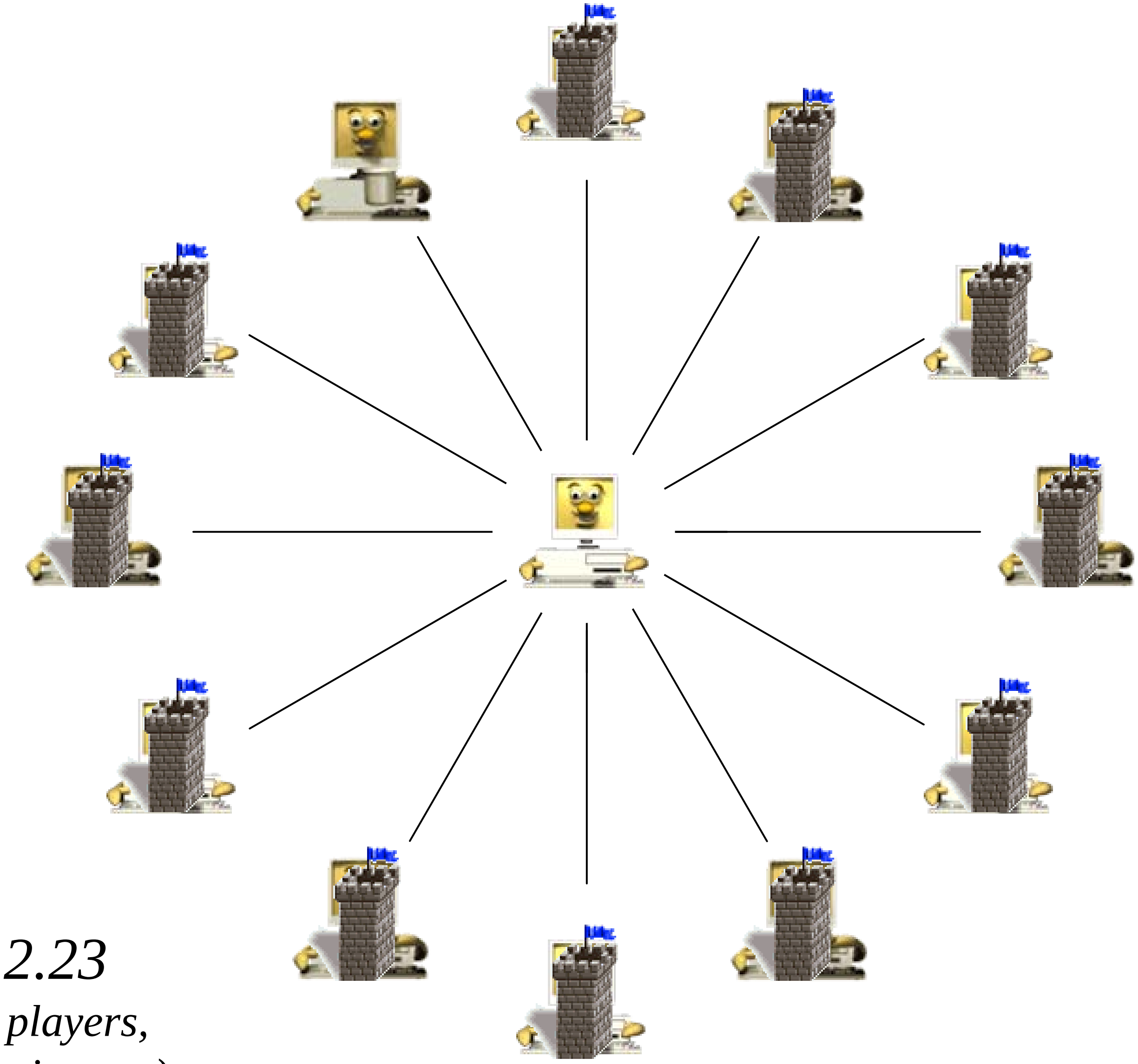
# Monotonicity: Counterexample

*n = 13*

*C = 1*

*L = 4*

*F = 0.9*



*total cost = 12.23*

*(many inoculated players, attack component size two)*

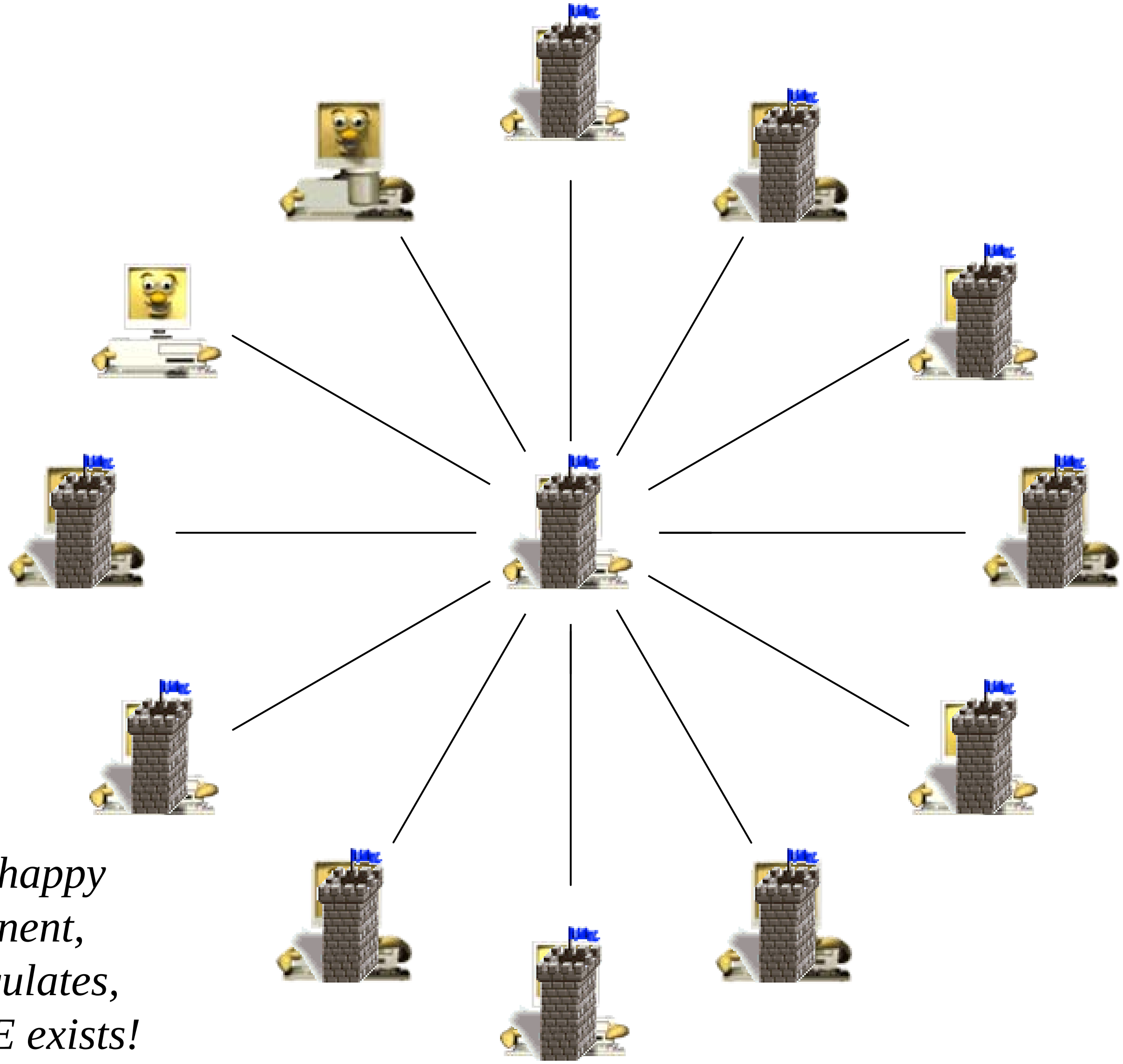
# Monotonicity: Counterexample

*n* = 13

*C* = 1

*L* = 4

*F* = 0.1



*Boundary players happy with larger component, center always inoculates, thus: only this FNE exists! total cost = 4.69*

# Further Results and Open Problem

- More results in paper...
  - e.g., better bounds for windfall on special graphs
  - e.g., NP hardness (best and worst FNE)

- Many exciting open problems!

  - Example 1: existence of equilibria and convergence time (asymmetry!)
  - Potential functions used for regular equilibria cannot be adopted directly...
  - Example 2: study of windfall on class of social networks
  - Example 3: multihop scenario (transitive friendship?)
  - Example 4: alternatives for worst case equilibria
  - Example 5: experimental verification in practice? Monotonicity in reality?

# Other Forms of Inequality?
# Heterogeneous Capabilities…

Under construction...

# Heterogeneous Peers...

- Peer-to-peer machines have different

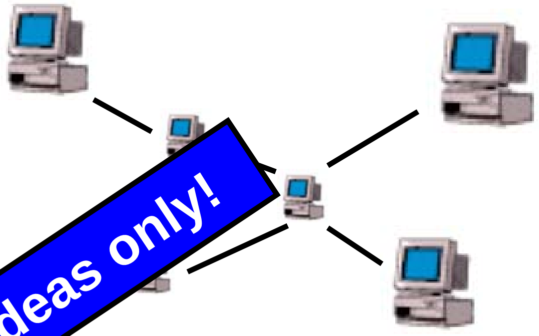
  - Internet connections
  - CPUs
  - Hard disks
  - Operating systems
  - ...

- But still, peers need to collaborate, in an efficient way

- Intersting problem
  - E.g., conflict with incentive compatibility
  - Should a (cooperative) weak peer be supported by stronger peers?
  - Threat: strategic behavior? Is peer weak or just selfish?

# The Basic Problem

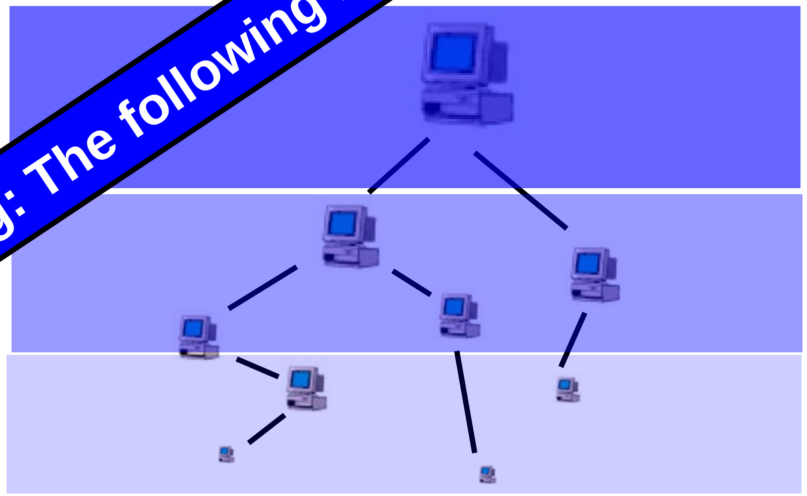
- Motivation: strong peers cannot make full use of the system if they can only interact indirectly via weak peers



- Idea: clustering of peers with roughly s[illegible]apability!

  - in a heap-like manner

**Warning: The following results are first ideas only!**



# The Distributed SHELL Heap

- What is a distributed heap?

- We assume that peers have a key / rank / order / id

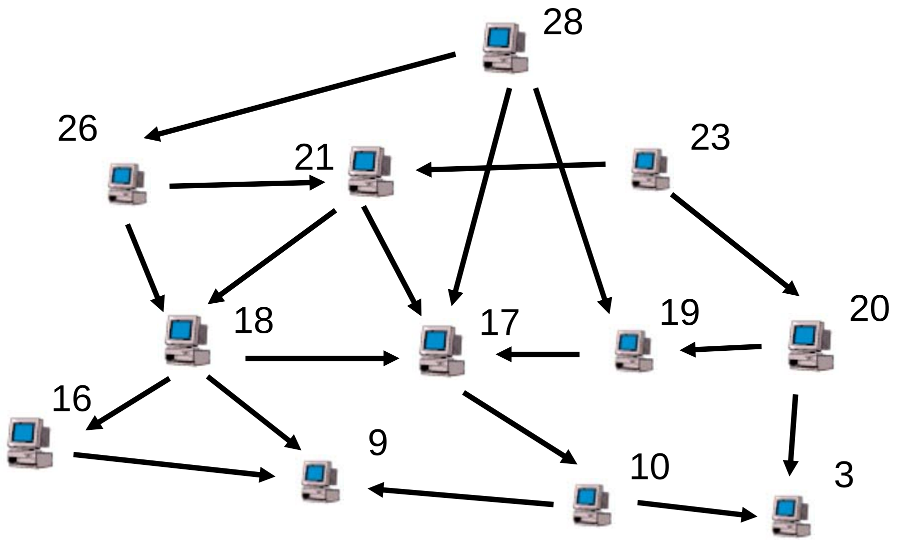
  - for example: inverse of peer capability

- (Min-) heap property: peers only connect to peers of lower rank

  - for example: peers only connect to stronger peers

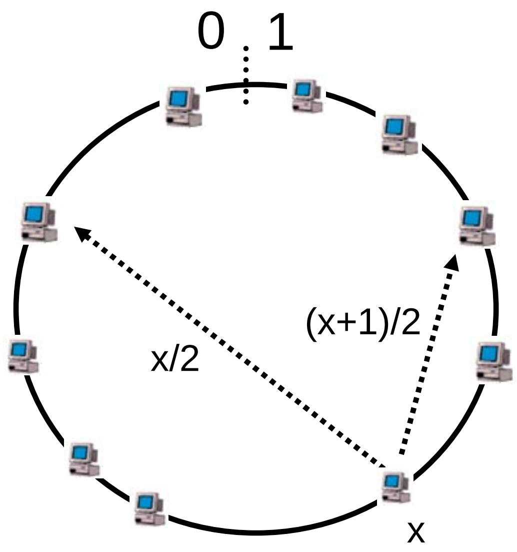
  - SHELL constructs a directed overlay

    (routing along these edges only)

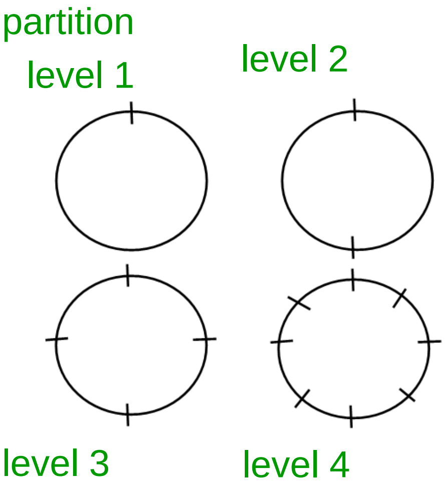


# The SHELL Topology (1)

- Continuous-discrete approach: de Bruijn network



- Problem: de Bruijn neighbor may have larger rank

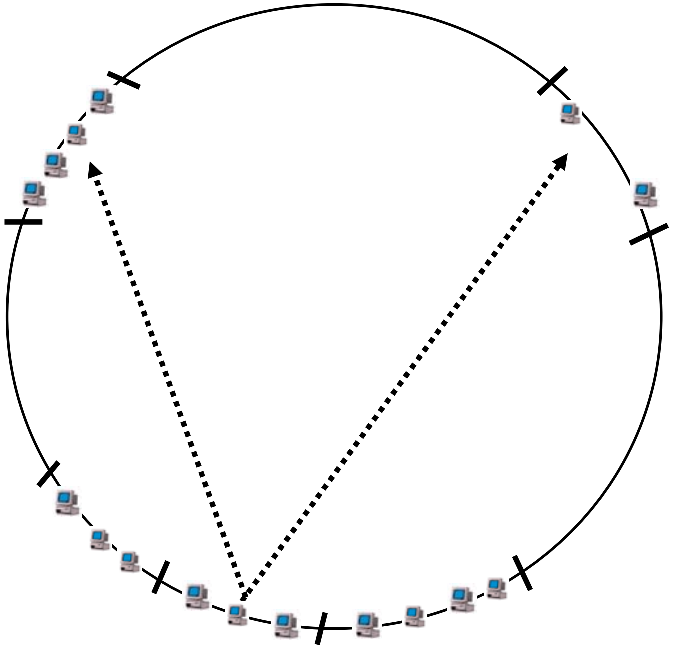


- Solution
  - peer at position x connects to all lower-ranked peers in an interval around x/2 and (x+1)/2
  - i.e., space divided into intervals
  - size of interval depends on number of low-rank peers there
  - larger degree, but still logarithmic diameter etc.

# The SHELL Topology (2)

- Peer connects to all peers of lower order in
  - Level-i home interval (interval which includes position x of peer)
  - Adjacent level-i intervals to home
  - de Bruijn intervals: intervals which include position x/2 and (x+1)/2

- What is level i?
  - Level i chosen s.t. there are at least c log $n_p$ lower order peers in interval
  - $n_p$ = total number of peers in system with lower order
  - $n_p$ can be estimated, in the following we assume it is given



# Routing (1)

- Interesting routing properties on heap

- Routing paths: if peer p is weaker than peer p‘, a request sent from p to p‘ only traverses peers which are stronger than p

  - „augmenting paths“

- E.g., live streaming: quality of transmission depends on weaker of the two peers, but not on peers in-between

- General routing policy: route according to de Bruijn rules, and choose highest-ranked peer to forward message in interval

  - yields low congestion

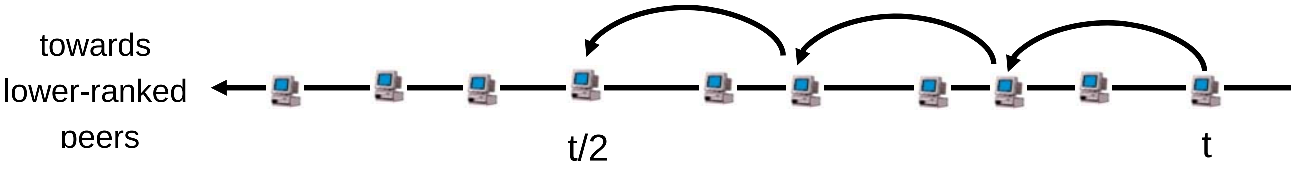
# Routing (2)

- Congestion: *each* peer makes a lookup of a *random* ID

  - Two phase routing: first forward along outgoing edges to a peer whose home interval includes ID, then make descent if necessary to higher-ranked peers

- Proof idea low congestion in first phase

  - Recall: among all lower-ranked peers, forward packet to highest-ranked peer
  - Thus, it can be shown that *w.h.p.*, reached peer whose interval includes ID is of rank at least t/2 when starting from ranked t peer
  - I.e., packet does not travel to too low-ranked peers
  - Therefore, peer of order t only receives packets from peers of ranks t+1..2t!



# Other Application: Robust Information System

- Approach also useful as robust distributed information system

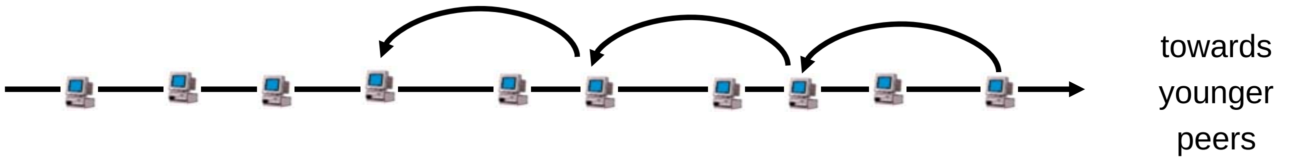
- For instance, robustness to Sybil attacks

- Sybil attack: at time t, an arbitrary number of malicious peers join the system

  - E.g., try to overload system with bogus requests

- Idea: build same de Bruijn heap, but use different peer ranks
  - Instead of rank ~ peer capacity, we use rank ~ join time
  - Thus: peers only connect to older peers
  - i.e., we want to maintain join time order in our distributed system

# Robustness to Sybil Attacks (1)

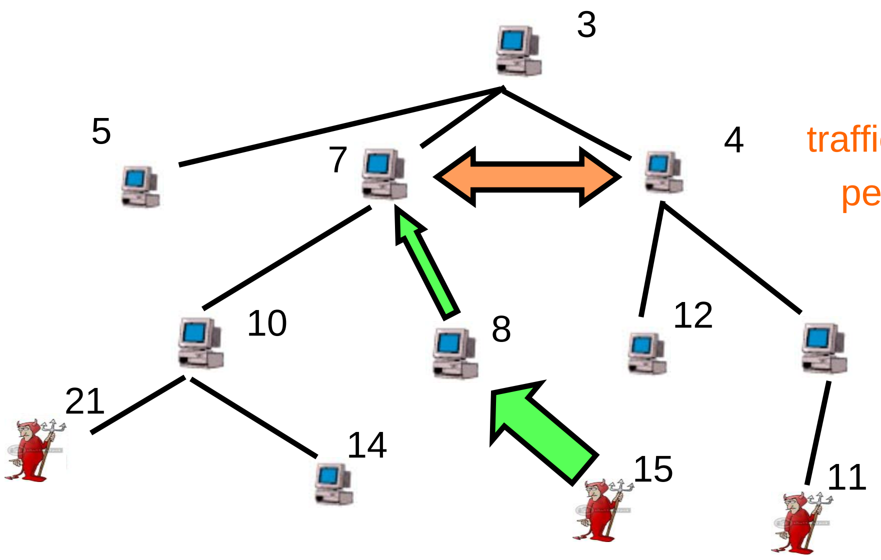


- Requests do not travel to younger peers
  - Path between old peers does not include any young / Sybil peers
  - Hence, older peers are immune against this Sybil attack!
  - Compare to heterogeneous SHELL system: paths without weak peers...

- Additional advantage: older peers have larger remaining session times
  - According to measurement studies
  - higher robustness to churn

# Robustness to Sybil Attacks (2)

- Yields min heap



traffic between older peers unaffected

higher peers can perform a rate control algorithm

attack originates from lower peers

- In addition, in case of congestion, a rate control algorithm could be used towards „lower peers“ in order to prevent newly joined peers to overload the system

  - Traffic only from younger to older peers, i.e., upward the tree

# Conclusion and Open Problems

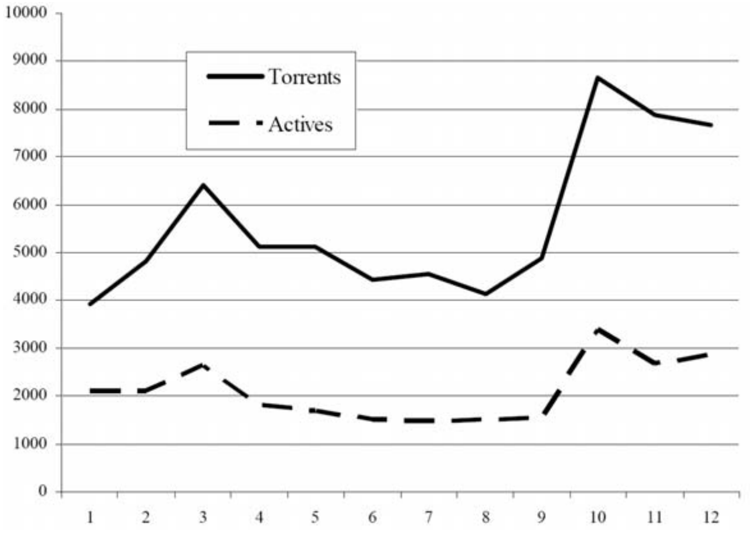
# Conclusion (1)

- Presence of unequal participants is interesting and important challenge in peer-to-peer computing

  - Unequal peers = peers which voluntarily or involuntarily do not contribute the same amount of resources as/to other peers

  - How to distinguish the two cases in a distributed environment?

- Reality check: are people selfish?



# Conclusion (2)

- Solutions to this problem have useful consequences
  - Most importantly: efficiency
  - E.g., side benefits: heap structure can also be used to make information systems robust to Sybil attacks

- Many open problems...

# Open Problems (1)

- Incentive compatible peer-to-peer computing?
  - Mechanism design...
  - E.g., how to prevent BitThief from free-riding in BitTorrent?
  - Tit-for-tat is good, but how to solve the bootstrap problem for newly joined peers?
  - E.g., fast extension: newly joined peers get a random subset of pieces for free (subset depends on peer ID) => venture capital
  - Unlike optimistic unchoking, BitThief could no longer download entire files...

- And what about other systems where peers are not directly interested in same file?

  - Difficulty depends on application, e.g., live streaming easier

# Open Problems (2)

- Robust peer-to-peer computing?

  - How to prevent our attacks on Kad?

  - Idea: do not allow arbitrary IDs... E.g., ID may depend on IP address of peer!

  - But: What if many peers behind NAT share same IP?

  - And what about dynamic IPs? Kad ID should be stable (e.g., no loss of credits etc.)

  - Idea: User generated phrase which is hashed?

  - But: Attacks still possible, as generated Kad ID must only approximately match to be censored keyword (sparsely populated ID space)

  - General rule (e.g., against publish attack): do not accept too much information from same IP address...

- Why are these attacks not used today?

  - Measurement studies show that there are indeed large ID clusters (Sybil attack?)

# Open Problems (3)

- Formal analysis with game theory?
  - Heterogeneous population (e.g., different BitTorrent clients)
  - Need to model different types of players
  - Effects of these players on social welfare?
  - How to exploit these phenomena?

- Insights can be relevant in other areas
  - e.g., how to foster cooperation / how to ensure high quality in Wikipedia?

Takk!

*Most results published in PhD thesis, download from*

*http://www14.informatik.tu-muenchen.de/personen/schmiste/*