Freeriding

Seminar WS 05/06

Lukas Oertle

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Departement Computer Science

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Definition <<

- **Definition**
  - A *freerider* is a peer in a P2P System (like Gnutella, BitTorrent, Kazaa, etc.), who does not upload any data.
- **Technique**
  - Specially configured/modified client software
- **Reasons for freeriding**
  - Uploading may be more risky (legal reason) or immoral
  - Avoid the bandwith cost of uploading
  - Two peers behind firewalls may not be able to exchange files

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Definition <<

- **Seeder**
  - A seeder is a peer that has finished downloading a file but is still connected (or reconnected again) to the system and might upload to other peers.
- **Freeriding ratio**
  - Percentage of peers that are freeriders
- **Sharing ratio**
  - Total amount of data the peer has uploaded divided by the total amount of data the peer has downloaded:

$$\frac{data_{upload}}{data_{download}}$$

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

General <<

- **The Problem with freeriding**
  - If everybody is freeriding → Nobody can download anymore
  - Freeriding is not punished
  - Contributing is not rewarded
  - Systems may become inefficient
- **Wanted**
  - Solutions to solve the freeriding-problem
  - Incentive mechanisms
    - Reward and punishment

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prisoner‘s Dilemma (PD) <<

- We have two prisoners, both committed a criminal delict.
- Cross-examination of the to prisoners. But they do not have the possibility to talk to each other, neither during the cross-examination, nor before.
- The highest penalty for the delict is 5 year.
- The police negotiaties with both prisoners and offers them a reduced penalty if they confess and accuse the other prisoner.

- If *both of them don‘t confess*, the indices are not strong enough and both of them get a 2-year-penalty.
- If *one of them confess*, but the other one doesn‘t, the one who confesses get‘s no penalty, the other one gets the full penalty of 5 years.
- If *both of them do confess*, both get a penalty of 4 years.

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prisoner‘s Dilemma (PD) <<

- The Dilemma is paradox, as the individually (confess) best solution is not the same as the collectively best solution (don‘t confess).
- A payoff matrix results the following:

	B doesn‘t confess (cooperates with A)	**B confess (betrays A)**
A doesn‘t confess (cooperates with B)	**A: -2** **B: -2**	**A: -5** **B: 0**
A confess (betrays B)	**A: 0** **B: -5**	**A: -4** **B: -4**

- Temptation (T) for confess (freedom): 0
- Reward (R) for mutual „don‘t confess“: -2
- Punishment (P) for mutual betrayal: -4
- Suckers payoff (S) for one-sided trust: -5

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prisoner‘s Dilemma (PD) <<

- **Problems**
  - T > R and P > S
  - Thus, any rational (=self-interest) prisoner should choose to confess.
  - If A confess, he gets a smaller penalty, independent of the decision of B
- **Conclusion**
  - Both prisoners confess
  - The (globally) best strategy „no one confess“ is never used
- **Solution**
  - Iterated cross-examination (Iterated Prisoner‘s Dilemma)

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Iterated Prisoner‘s Dilemma <<

- In the iterated prisoner‘s dilemma, decisions of the other prisoners are remembered and influent future decisions
- There are many different strategies for the iterated prisoner‘s dilemma

- We focus on one: **TIT FOR TAT**

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Iterated Prisoner‘s Dilemma <<

- **TIT-FOR-TAT**
  - In the first cross-examination, the prisoner doesn‘t confess. Thereafter, he does what the other prisoner did in the previous cross-examination
  - IMPORTANT: Not every prisoner uses the same strategy. If all prisoners would uses TIT-FOR-TAT, nobody would ever confess.

- **Analysis**
  - A prisoner A which uses the TIT-FOR-TAT-strategy cannot defeat any prisoner B because A always doesn‘t confess first and continues to do so until B confesses.
  - But the difference between the penalty is very small: T – S
  - It occurs in two cases:
    - B always confesses. So A gets the penalty S in the first round.
    - B switches in the last round from „don‘t confess“ to confess
      A has no possibility to react on this decision and gets the penalty S

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

File Exchange in BitTorrent <<

- **General**
  - BitTorrent is a peer-to-peer-system as former Napster, Kazaa, Emule, Gnutella etc.
  - The files to be exchanged are segmented into *fragments*
  - Peers have different sets of fragments which they exchange among themselves
  - A peer which has all fragments of a file and continues uploading the fragments is a *seeder*
  - A speciality of BitTorrent is that also peers which have not the whole file, but only fragments of a file, also participate to the upload-process.

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

File Exchange in BitTorrent <<

- **Joining the distribution of fragments**
  1. A peer downloads the *torrent-file* of the target file (e.g. the mp3-file) by a well-knows web site.
     This *torrent-file* contains meta-information, e.g. the tracker-address.
  2. The tracker (serverside) contains information about the peers which have fragments for the specific file.
  3. Connections to the peers which provide fragments are established and the download process starts.

- After peers have successfully downloaded a fragment, they provide this fragment to the other peers downloading the same file → fragments are exchanged between peers

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BTs original mechanism <<

- **To avoid that freeriders only download fragment, but don‘t upload, BitTorrent has implemented an incentive mechanism:**
  - $Peer_0$ maintains the download-rates from all ist peers.
  - It allows uploading to $peer_1$ to $peer_b$ with the b highest download rates.

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BTs original mechanism <<

- **Optimistic uploading**
  - Another $peer_X$ gets the permission to download from $peer_0$. Thus, $peer_0$ maybe gets the permission to download from $peer_X$. If this $peer_X$ has a higher download-rate than one of $peer_1$ to $peer_b$, $peer_b$ is replaced by $peer_X$.
    $Peer_X$ is choosen by a round-robin mechanism which chooses every 30 seconds another peer.
  - ***Fast peers exchange with fast peers. Slow peers exchange with slow peers.***

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BT as Prisoner‘s Dilemma <<

- Fragment-Exchange between peers is the Prisoner‘s Dilemma
- Every peer has the largest benefit if it does not upload
- But two peers together have the largest benefit if both of them upload

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BT as IPD <<

- **Definition**
  - d = the utility of a download
  - u = the negative utility of an upload (cost of an upload)
  - Reward for mutual upload R = d-u
  - Temptation for „no upload“ T = d
  - Suckers payoff for one-sided download S = -u
  - Punishment for mutual „no upload“ P = 0
- **Some additional rules**
  - $T > R > P > S \leftrightarrow d > d-u > 0 > -u$
  - $2R > S + T$
    Otherwise the benefit of changing upload/no upload would be greater than permanent upload
- As BitTorrent peers have the possibility to stop uploading to another peer, we have an iterated prisoner‘s dilemma

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Proposed mechanism <<

- Incentive mechanism that is in the broadest sense a TIT-FOR-TAT strategy for the IPD.
- Each link to another peer is defined with a deficit
  u - d = upload amount - download amount
- Fragment size c
- We define a *nice factor* f (≥1) which defines the following equation:

  $$u - d \leq f \cdot c$$

- This factor determines the amount a peer will risk to get the chance to establish cooperation

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Proposed mechanism <<

- **We get a payoff matrix for the proposed mechanism**

	B uploads (cooperates with A)	**B doesn‘t upload**
A uploads (cooperates with B)	**A: R = d - u** **B: R = d - u**	**A: S = -u** **B: T = d**
A doesn‘t upload	**A: T = d** **B: S = -u**	**A: P = 0** **B: P = 0**

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Comparison <<

- **High-scoring peers have the following properties:**
  - They are nice:
    They upload as long as the other peer does. They never stop uploading first.
  - They are retaliatory:
    They stop uploading if the other peer does. So they protect them selves from being exploited.
  - They are forgiving:
    They restart uploading if the other peer also restarts uploading after an interruption.

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Comparison <<

- **The proposed mechanism has all of them.**
- **The original mechanism lacks:**
  - Niceness:
    Peers stop uploading first if they find a better link
  - Not retaliatory:
    The optimistic uploading gives potential freeriders the possibility to get 1/(b+1) of my upload bandwith. It can be a significant ressource leak.
  - Limited forgivness:
    Reconstructing mutual upload may be difficult within the time limit of 30 seconds.

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Evaluation <<

Facts and Assumptions

- We evaluate the old (original) and the new (proposed) mechanism in a case example
- We have a file of 33 MB divided into fragments of 128 KB
- We have 170 downloaders
- Peers are uniformely given upload rates from 1 to 100 KB
- We have only one seeder and every peer that has downloaded all fragments leaves the system

Two examples

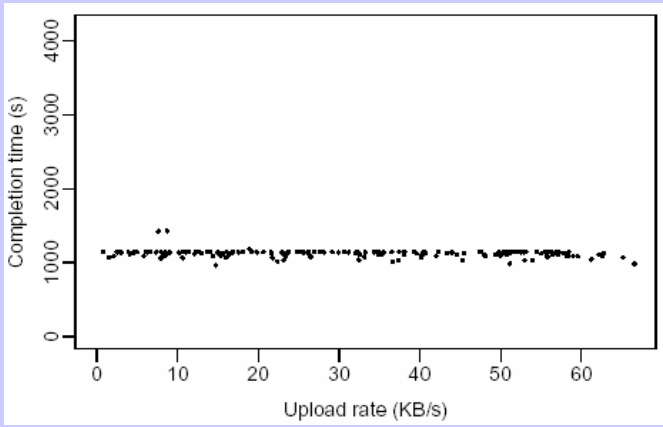
- An experiment with pure population (either OLD or NEW downloaders)
- Another one with OLD and NEW downloaders together, mixed population

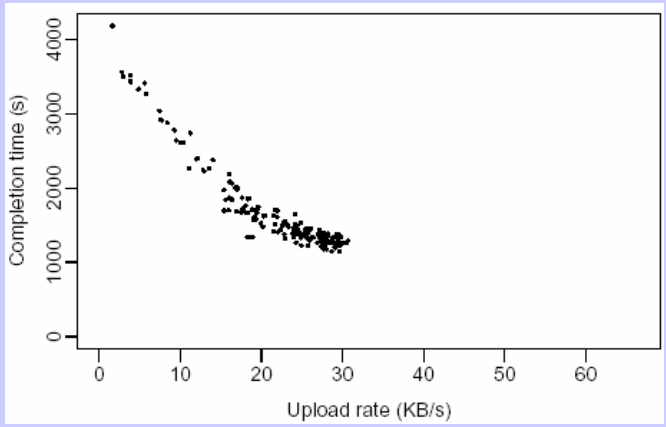
Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Evaluation <<

Pure population





OLD downloaders **NEW downloaders**

(Each spot corresponds to a peer)

- **Upload range**

  OLD: 0.78 KB to 66.56 KB

  NEW: 1.62 KB to 30.65 KB

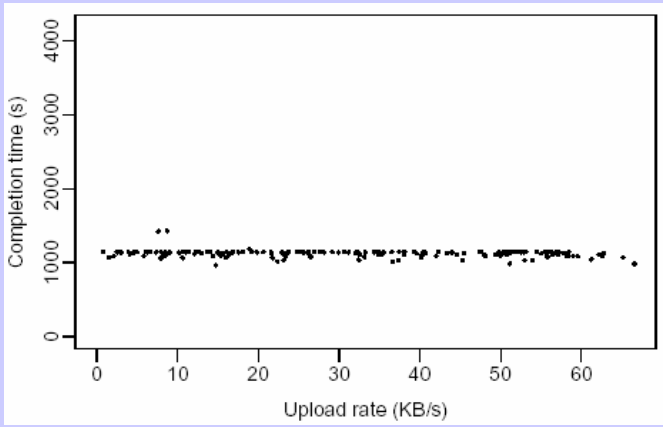
  The maximum upload rate is smaller in the new system as the fragment scheduling is less efficient (lack of diversity and OLD uploaders tend to maximize upload regardless of what the other side does)

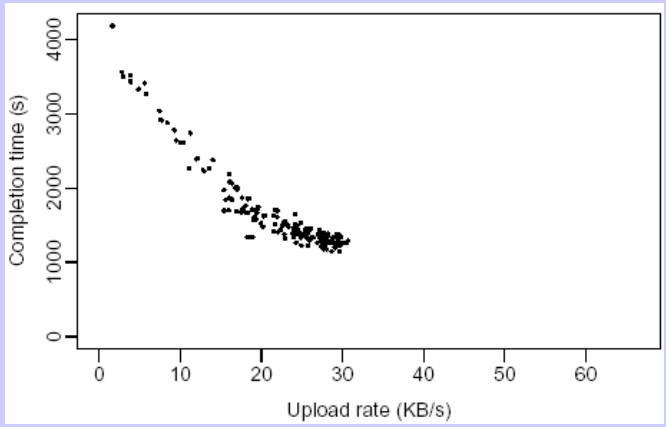
Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Evaluation <<

Pure population





OLD downloaders **NEW downloaders**

(Each spot corresponds to a peer)

- **Average completion time**

  OLD: 1123 s

  NEW: 1672 s

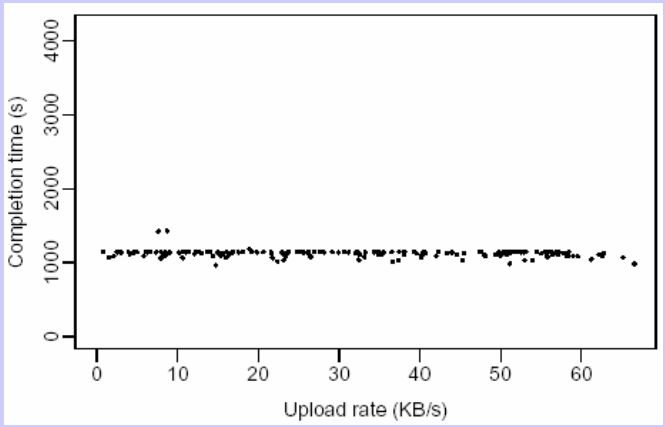
  Thats because of the bad upload rate of the new system

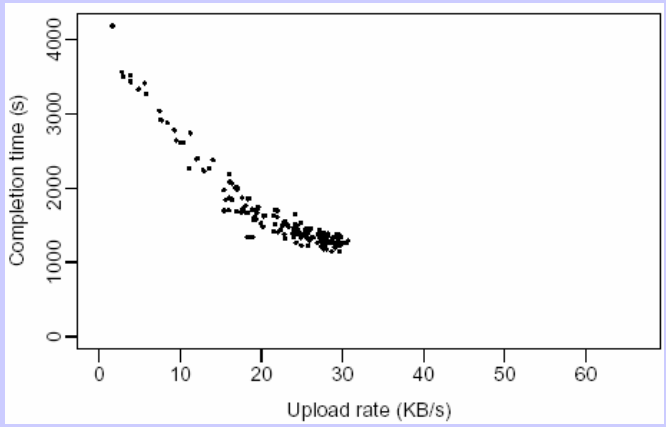
Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Evaluation <<

Pure population





OLD downloaders **NEW downloaders**

(Each spot corresponds to a peer)

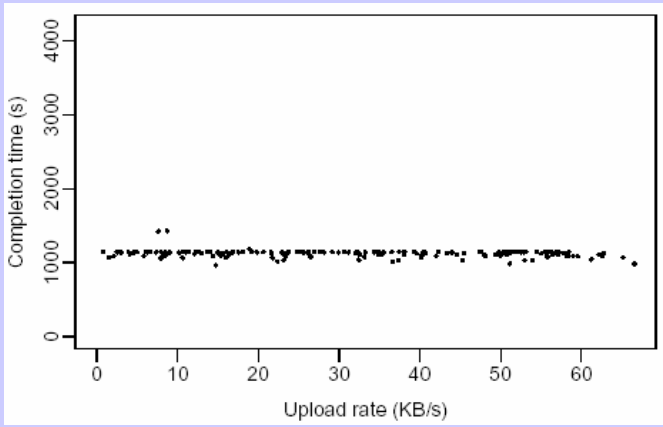
- We could argue, that the old system is better than the new one
- BUT: remember that we adjustet the upload bandwith uniformely between 1 KB and 100 KB
- A real system doesn‘t garantee this property and all peers could be freeriders (upload bandwith = 1 KB)
- The graph in this case would be completely different

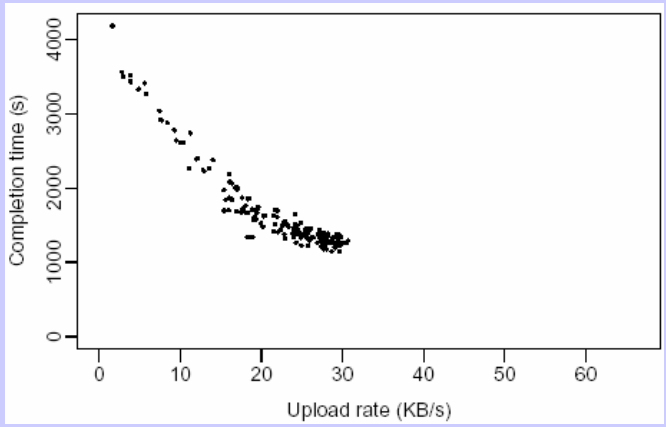
Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Evaluation <<

Pure population





OLD downloaders **NEW downloaders**

(Each spot corresponds to a peer)

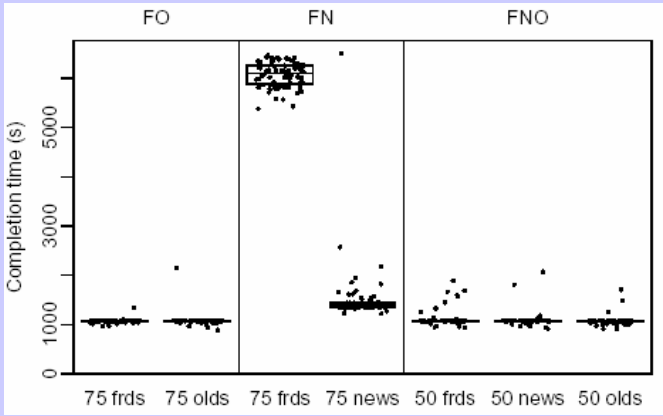
- Minimum peer deficit
  (= total upload amount – total download amount)
  OLD: -32 MB
  NEW: -9 MB

Lukas Oertle, 21.12.05

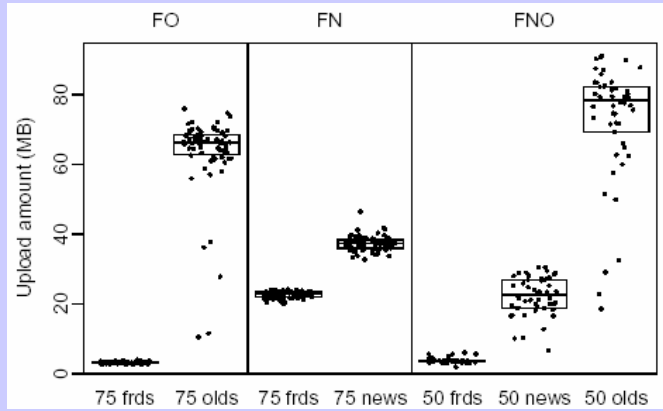
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Evaluation <<

Mixed population



Download completion time



Upload amount

(Each spot corresponds to a peer)

- Freeriders and OLD downloaders have almost the same completion time, but very different upload amounts
- In a system with freeriders and NEW downloaders, freeriders have no chance
- But as soon as OLD downloaders join the system, all incentives to avoid freeriding is destroyed

Lukas Oertle, 21.12.05

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Conclusion <<

- The old mechanism has the higher efficiency, but doesn‘t punish freeriders
- The new mechanism is not as efficient as the old one, but prevents freeriding
- Because the system gets abused as soon as old downloaders join the system, one must ensure that no old clients are around.
  - Make the incentive mechanism work on the serverside