Cooperation in Open Distributed Systems

Stefan Schmid
Distributed Systems 2008/9

Wireless: Many mobile phones today have WLAN (and even Skype)

P2P: Olympic games 2008 live-broadcast over peer-to-peer networks

„Social“ networks: Facebook, Xing, Twitter...
E.g. US elections 2008: Obama makes extensive use of Internet technologies
Challenges in Open Distributed Systems (1)

**Dynamic resources**
e.g., in peer-to-peer computing: worst-case churn, topological self-stabilization, etc.

**Robustness**
e.g., in wireless networks: jamming
e.g., in peer-to-peer networks: DoS attacks
(cf Leighton/Akamai)

**Efficiency**
e.g., how to deal with huge amounts of data?
e.g., low-overhead p2p live streaming?
Challenges in Open Distributed Systems (2)

Economics
- e.g., selfishness / altruism / malicious behavior
- e.g., mechanism design for anonymous, money-less networks

Heterogeneity
- e.g., live streaming with heterogeneous peers

Focus of this talk: Cooperation and heterogeneity in peer-to-peer systems!
Peer-to-Peer Technology

- Well-known p2p systems
  - P2P: contributions of participants
  - Internet telephony: Skype, file sharing: BitTorrent, eMule, ..., streaming: Zattoo, Joost, ...

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

- Impact: Accounts for much Internet traffic!
  (old source: cachelogic.com)

Nowadays only 20% (T-Lab measurements)?
Cooperation in Peer-to-Peer

- Peer-to-peer systems
  - open to „everybody“
  - rely on contributions
  - heterogeneous

- Non-cooperation: threat to the paradigm

- For example BitThief:
  - Proof of concept Java client
  - Downloads without uploading at all
  - despite BitTorrent‘s incentive mechanism!
BitThief’s Tricks

BitThief’s three simple tricks:
1. Open many TCP connections
2. Contacting tracker again and again, asking for more peers (never banned!)
3. Pretend being a great uploader in sharing communities

⇒ Exploit optimistic unchoking slots
⇒ „Exploit“ seeders
⇒ Exploit sharing communities
BitThief: Results (with Seeders)

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

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

<table>
<thead>
<tr>
<th>Size</th>
<th>Seeders</th>
<th>Leechers</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 170MB</td>
<td>10518 (303)</td>
<td>7301 (98)</td>
</tr>
<tr>
<td>B 175MB</td>
<td>923 (96)</td>
<td>257 (65)</td>
</tr>
<tr>
<td>C 175MB</td>
<td>709 (234)</td>
<td>283 (42)</td>
</tr>
<tr>
<td>D 349MB</td>
<td>465 (156)</td>
<td>189 (137)</td>
</tr>
<tr>
<td>E 551MB</td>
<td>880 (121)</td>
<td>884 (353)</td>
</tr>
<tr>
<td>F 31MB</td>
<td>N/A (29)</td>
<td>N/A (152)</td>
</tr>
<tr>
<td>G 798MB</td>
<td>195 (145)</td>
<td>432 (311)</td>
</tr>
</tbody>
</table>
Another Example: Non-cooperation in Kad

- Kad = one of the first widely used distributed hash tables (DHT)

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

* Attention: this is a simplification (factor 10 replication in „close“ tolerance zone)
Kad Censorship

• Several **vulnerabilities**

• **Example:** malicious peers can perform censorship 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

Request: <Simpson,Movie>

h(Simpson)

works even better...
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)...
BitThief and Kad Attacks: Easy to Fix?

• **BitThief**
  - Optimistic unchoking can be exploited
  - Just do pure tit-for-tat? Bootstrap problem...
  - Fast extension: subset of pieces only (limited „venture capital“)
  - No direct interest? E.g., inter-swarm incentives?

• **Kad Attacks**
  - Too much information from same peer (e.g., publish attack)
  - Bind ID to peer... But how?
  - Bind to IP? NATs yield same peer IDs? Dynamic IP addresses?
    - Credit loss?
  - Generate ID, e.g., by hashing a user phrase?
    - But sparsely populated ID space =>
    - easy to generate IDs close to the object...
Insights from Game Theory?

- A model for peer-to-peer networks?

- Game theory can help to find mechanisms
  - E.g., malicious players may be beneficial
  - E.g., too much altruism can be harmful
Heterogeneity

- Tight connections to the cooperation challenge

- E.g., streaming: Shall stronger peers support weaker ones?
  - If yes, what about selfishness?

- SHELL: Takes into account heterogeneity
  - Distributed oblivious heap
  - Paths between strong peers do not include weak peers
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: only connect to lower rank peers
  - for example: peers only connect to stronger peers
  - SHELL constructs a directed overlay
    (routing along these edges only)
The SHELL Topology

- 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.

- Oblivious: Very fast joins and leaves!
Routing

• Routing paths are **augmenting** (no weak peer between)

  ![Routing diagram]

  - towards lower-ranked peers
  - r/2
  - r

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

• **Congestion guarantee**
  - „first phase“ ends at peer rank at least t/2 w.h.p.
  - second phase short...
SHELL Solves Cooperation Problems!

- Approach also useful as robust distributed information system

- Idea: de Bruijn heap, but different peer ranks
  - Use rank ~ join time
  - Thus: peers only connect to older peers
  - i.e., maintain join time order

traffic between older peers unaffected

higher peers can perform a rate control algorithm

attack originates from lower peers

Stefan Schmid @ T-Labs, 2009
Conclusion

• Presence of unequal participants interesting and important challenge
  - Unequal = voluntarily or involuntarily little or no contribution
  - How to distinguish the two cases in a distributed environment?

• Reality check: are people selfish?

Thank you for your attention!
Some Literature

Dynamic resources
e.g., in peer-to-peer computing: worst-case churn (IPTPS 2005), topological self-stabilization (PODC 2009), etc.

Robustness
e.g., in wireless networks: jamming (DCOSS 2009)
e.g., in peer-to-peer networks: DoS attacks (SPAA 2009)

Efficiency
e.g., how to deal with huge amounts of data? (PODC 2008)
e.g., low-overhead p2p live streaming? (DISC 2007)
Some Literature

**Economics**
e.g., selfishness / altruism / malicious behavior (PODC 2006, EC 2008)
e.g., mechanism design for anonymous, money-less networks (INFOCOM 2009)

**Heterogeneity**
e.g., live streaming with heterogeneous peers (ICALP 2009)