A Case For OneSwarm

Tom Anderson
University of Washington

http://oneswarm.cs.washington.edu/

With: Jarret Falkner, Tomas Isdal, Alex Jaffe, John P. John, Arvind Krishnamurthy, Harsha Madhyastha and Mike Piatek
The Move to Cloud Computing

Increasing amounts of data and computing moving into data centers:

- Search, web, email, social networking, video, ...
- Document preparation, collaborative work, ...

Advantages:

- Professional system management
- High availability
- Efficient for data analysis, data mining
- Lower administrative costs than do-it-yourself
- Statistical multiplexing of resources
A Gathering Storm?

Some downsides for the typical user:

- Forced to **trust** your provider wrt data ownership, revocation, lock-in, privacy
  - Facebook, Kindle, iPhone, ACM
- Cloud is a **natural monopoly**
  - Reliability/flash crowd issues for new entrants
  - Biz model: Attract developers, users will follow
  - No plausible avenue for an “open source” cloud
- Cloud forces an advertising model on the long tail
  - YouTube, Obama’s Inaugural, community sharing
- **Censorship** of the cloud is increasingly common
We're really sorry, but due to datacenter temperature issues beyond our control Last.fm is currently offline.

Please bear with us as we scramble to catch overheating DC/AC inverters... stay tuned, we'll be restoring regular service as soon as possible. Thanks for your patience.

In the meantime, check out our twitter account for updates on the situation.


Español Debido a un problema de energía con el centro de datos ajenó a nuestro control, estamos intentando mantener last.fm con un número de servidores mínimos durante este fin de semana. Resumiremos nuestro servicio acostumbrado lo antes posible. Gracias por vuestra paciencia.

Português Por motivos de força maior, haverá redução de energia em nossos servidores, fazendo com que a capacidade de armazenamento de dados do site fique drasticamente reduzida durante este fim de semana. O serviço voltará ao normal assim que possível. Obrigado pela sua paciência.

Русский Из-за перебоев с электроэнергией по независящим от нас обстоятельствам, сайт будет работать в режиме экономного использования серверов на выходных. Оставайтесь с нами, мы вернемся к обычному режиму работы в ближайшее время.

Спасибо за терпение.

Svenska Okonterrollerbara strömbroblem tvingar oss att försöka köra Last.fm på ett väldigt begränsat antal serverar denna helgen. Vi kommer att gå tillbaka till vår vanliga kapacitet så snart vi kan.

Tack för ert tålamod.

Türkçe Datacenter’ımızda ortaya çıkan kontrolümüz dişindaki elektrik probleminden dolayı bu hafta sonu Last.fm’nin sunucularını minimuma indirgeme durumundayız. En kısa zamanda normal hizmetimizi vermeye devam edeceğiz. Sabrıınız için teşekkür ederiz.

日本語 弊社でのコントロール範囲外での、データセンターサイドのパワープロブレムの為、本週末Last.fmのサービスのパワーサップが大幅に減少されます。いつでも通常サービスで復旧に向けておりますが、しばらくの間大変ご迷惑をおかけします。ご了承くださいませ。

中文（简体） 由于数据中心电源问题，本周末某些服务可能无法提供。我们将尽快恢复正常服务。由此带来的不便，敬请谅解。
The P2P alternative

Examples: Skype, BitTorrent

- **Abundant resources and scalability**
  Plentiful idle capacity, automatic scaling

- **High availability**
  Multiple data sources, geographically distributed

- **Democratizes distribution**
  Anyone can publish & access, without infrastructure
Technical challenges

- **Abundant resources**
  Users throttle contributions and free-ride

- **High-availability**
  Users are transient and depart after downloading

- **Democratizes distribution**
  Open protocols make monitoring behavior easy
OneSwarm

1. Open protocol, no infrastructure required
2. P2P system with robust contribution incentives
3. Share publicly, privately, or anonymously
4. Widely used: 1M+ downloads, 12 languages

http://oneswarm.cs.washington.edu/
Outline

• A case study: BitTorrent
  • Weak contribution incentives degrade performance
  • Wholesale lack of privacy
• OneSwarm
  • Robust incentives via one hop reputations
  • Privacy control via friend-to-friend sharing
• Scatter
  • Scalable consistent key-value storage
Outline

- A case study: BitTorrent
  - *Weak contribution incentives* degrade performance
  - Wholesale *lack of privacy*
- OneSwarm
  - *Robust incentives* via one hop reputations
  - *Privacy control* via friend-to-friend sharing
- Scatter
  - *Scalable consistent resilient* key-value storage
A case study: BitTorrent

- P2P scalability depends on **user contributions**
  But, users are sometimes **stingy**

- Early P2P designs were hampered by **free-riding**
  e.g., in Gnutella, 70% of users didn’t contribute

- Currently popular networks explicitly include **contribution incentives**
  e.g., tit-for-tat servicing in BitTorrent

Does tit-for-tat work?
BitTorrent overview

M joins the system by obtaining a random subset of current peers from a coordination service.
BitTorrent overview
BitTorrent overview
BitTorrent overview

1 2 3 4

S

M

A

B
BitTorrent overview

Diagram showing the BitTorrent network with computers labeled S, M, A, B and colored squares indicating data pieces.
Tit-for-tat in BitTorrent

Choosing *peers* and *rates*:

1. **Sort peers** by incoming data rate

2. **Reciprocate with** top $k$, $k \sim \sqrt{\text{capacity}}$

3. Send each peer selected in (2) an *equal split* of capacity

<table>
<thead>
<tr>
<th>Peer</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>17</td>
</tr>
<tr>
<td>B</td>
<td>13</td>
</tr>
<tr>
<td>C</td>
<td>8</td>
</tr>
<tr>
<td>D</td>
<td>5</td>
</tr>
</tbody>
</table>

If $k=2$, reciprocate with A and B, sending to each at an equal rate
Tit-for-tat in practice

- Does tit-for-tat be work in practice? – No
- Tit-for-tat is *ineffective for most swarms*
- Even when it influences performance, *tit-for-tat can be cheated*
Performance & Availability

- Tit-for-tat does not significantly influence most swarms.
- 25% of swarms are completely unavailable.
- A 100-fold increase in contribution yields only a 2.7x improvement.
- 74% of download rates are less than 50 KBps.
- 25% of swarms are completely unavailable.

Download performance for 13,353 swarms over 1 week.
Problem

- BitTorrent’s incentives are ineffective.

- Is there a fundamental lack of interest? – No
  Most peers download and quickly leave.
  *Tit-for-tat only applies when downloading*

- Is there a fundamental lack of capacity? – No
  Average capacity is 8X average performance
  *Many peers contribute less than their full capacity*
End-host capacities

65% of peer upload capacities $\leq 100$ KBps

End-host capacity is heavily skewed
Peers compete for reciprocation over a small range

70% of send rates fall within an 8 KBps range
Designing *BitTyrant*

- **BitTyrant**: a strategic BitTorrent client

- **Key idea**: maximize return on investment by adjusting selected peers and rates

- **Cost**: upload rate to peer: $u_p$
  **Benefit**: download rate from peer: $d_p$

- BitTyrant dynamically estimates these rates
Selecting peers & rates

Each TFT round, order and reciprocate with peers:

\[
\frac{d_0}{u_0}, \frac{d_1}{u_1}, \frac{d_2}{u_2}, \frac{d_3}{u_3}, \frac{d_4}{u_4}, \ldots
\]

choose \( k \mid \sum_{i=0}^{k} u_i \leq \text{cap} \)

After each round, for each peer:

- **If peer reciprocates:**
  - \( d_p \leftrightarrow \text{direct observation} \)
  - ...and continues to do so:
    - Reduce \( u_p \)

- **No reciprocation:**
  - Increase \( u_p \)
Single BitTyrant client

BitTyrant provides consistent performance

BitTyrant finds the point of diminishing returns

Download time (seconds) vs. Upload cap (KB/s)
Incentives in BitTorrent

- BitTyrant shows that tit-for-tat provides a weak *contribution incentive*, particularly for high capacity peers.

- Data quickly becomes unavailable since *users have no incentive to contribute once a file has finished downloading*. 
Outline

• Two problems:
  • Weak contribution incentives in existing designs
  • Wholesale lack of privacy

• OneSwarm
  • Robust incentives via one hop reputations
  • Privacy control via friend-to-friend sharing
Goal

• *Problem*: Incentives apply only when users are *actively downloading* and only for a *single swarm*

• Peers should be *rewarded for all contributions* across swarms even when not downloading
A simple fix?

- **Local history:**
  Each peer remembers all contributions and reciprocates across time and swarms

- A contributes to B because in a *future swarm*, B will recognize A and *reciprocate*

- But, *repeat interactions are rare*
  (1% of peers in our traces)
Indirect reciprocation

- Direct reciprocation is insufficient because *most peers directly interact with few others*

- A solution requires *indirect reciprocation*

*What path-length is required?*
One hop is enough

- 97% of peers are indirectly linked by less than 0.0002% of the most popular peers
One hop is enough

- 97% of peers are indirectly linked by less than 0.0002% of the most popular peers

A contributes to I

Swarm 1
One hop is enough

- 97% of peers are indirectly linked by less than 0.0002% of the most popular peers

In another swarm, I contributes to B

Swarm 2
One hop is enough

- 97% of peers are indirectly linked by less than 0.0002% of the most popular peers

When A and B meet, they exchange control traffic recognizing A’s contribution to B and enabling indirect reciprocation.
One hop reputations

- Two parts:
  1. A *protocol* that provides verifiable information
  2. A *default policy* that rewards contribution
One hop protocol

• What enables long-term identification?
  Persistent identity: self-generated key pair

• What do intermediaries do?
  Maintain accounting information and provide signed verification receipts

• How are intermediaries discovered?
  Gossip during connection setup
One hop default policy

• How are intermediaries selected?
  *Popularity*

• Why serve as an intermediary?
  *Priority service*

• How are peers evaluated?
  For each shared intermediary I:
  
  \[
  ( I's \, \text{valuation of } P ) \times ( \text{Local valuation of } I )
  \]

*The one hop protocol permits different policies per-peer*
Does it work?

• Key result:
  97% of pairs of peers share *at least one intermediary*
  Median value: 134 intermediaries per pair

• Additional evaluation shows:
  • Time to reciprocation is low
  • Overhead is limited
  • New users are quickly bootstrapped
  • Download performance improves in practice
Outline

• Two problems:
  • Weak contribution incentives in existing designs
  • Wholesale *lack of privacy*

• OneSwarm
  • *Robust incentives* via one hop reputations
  • *Privacy control* via friend-to-friend sharing
P2P Privacy

- No centralized control
  Open protocols make monitoring behavior easy

Instead of a service provider monitoring you, *anyone* can monitor your behavior
Monitoring is widespread

• We monitored \textit{tens of millions of people} with a small cluster

• Others monitor as well – we attracted \textit{hundreds} of copyright complaints

• Monitoring is sometimes \textit{inconclusive} and can be \textit{manipulated}

\url{http://dmca.cs.washington.edu/}
Spoofing results

<table>
<thead>
<tr>
<th>Host type</th>
<th>Complaints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop machine (1)</td>
<td>5</td>
</tr>
<tr>
<td>IP Printers (3)</td>
<td>9</td>
</tr>
<tr>
<td>Wireless AP (1)</td>
<td>4</td>
</tr>
</tbody>
</table>

False positives generated
Outline

• Two problems:
  • Weak contribution incentives in existing designs
  • Wholesale lack of privacy

• OneSwarm
  • Robust incentives via one hop reputations
  • Privacy control via friend-to-friend sharing
Motivating question

• Can we build a P2P data sharing system that provides users with control over their privacy? – Yes

• Key technique: leverage social networks
Privacy goal

*Give users more control* over how they share and obtain their data

1. Publicly with anyone (as in BitTorrent)
2. With specific people (permissions)
3. With anyone (without being monitored)

*Non-goals:* complete anonymity

Can we achieve a significant improvement in privacy without a significant loss of performance?
Previous solutions

- Solution 1: Put *Tor* in front of BitTorrent
  - Problems:
    - Requires a fraction of clients to be public
    - Discouraged for *performance* reasons

- Solution 2: *Freenet*
  - Problems:
    - Poor bulk data *performance*
    - Requires users to store other peoples data
Existing systems

Performance cost for privacy is significant!
Threat model

**OneSwarm goal:** Protect the users from revealing who is sharing / downloading which items

- **The Attacker:**
  - Controls a *limited number* of overlay nodes
  - Can do *anything* on nodes it controls
    - inject/sniff traffic, correlate data
  - **Cannot:** Sniff / inject traffic into *arbitrary network links*
Friend-to-friend sharing

- Exchange data directly *with friends only*
- Distant data obtained using *forwarding*

- Flooding discovers *multiple sources* using *multiple paths* per-source

- Users *control their own data* and can define *per-friend permissions* using persistent identities
Forwarding example

Point-to-point connections secured using SSL bootstrapped by each user’s public key
Forwarding searches
Forwarding data

Data source

David

Search origin

Me

Dan

John

Tom
Forwarding data

Transport is analogous to BitTorrent, with paths treated as peers.
View from the receiver

Search origin

John

Dan

Me

Search origin
View from the sender

David

Data source

Tom

Dan

?
Randomized delays, probabilistic forwarding, and a mix of trusted and untrusted peers reduce vulnerability to attacks.
Design challenges

• Locating mobile peers
  • Publish IP at peer-specific key in DHT

• Robustness despite sparse social networks
  • Support long-term binding with untrusted peers
  • Repeatable but probabilistic forwarding
  • Randomized delays to foil timing attacks

• Efficient search without revealing src and dst

• Maintain performance despite long paths
An impractical approach?

- Isn’t *overhead* a problem?
- Search by *controlled flooding*
- *Multi-source, multi-path transfers* improve performance

- OneSwarm works well in practice
- Performance *comparable to BitTorrent*
- *Outperforms alternatives* (Tor, Freenet)
Multi-path performance

Median single-path performance < 10 KBps

Multi-path transfers substantially improve performance
Multi-path transfers substantially improve performance.

Median multi-path transfer rate exceeds 400 KBps.
Performance

Transfer times for 120 machines

Cumulative fraction of transfers vs. Download time (seconds)

- BitTorrent
- OneSwarm
- 90% Tor/BitTorrent

Transfer times for 120 machines
Performance

OneSwarm reduces the performance cost of privacy
Summary

• OneSwarm provides a *practical P2P alternative* to data sharing with centralized trust

• Key mechanisms:
  • Persistent identities enable *long-term relationships*
  • Privacy-preserving sharing via a *social network*