Energy Efficient BitTorrent for Green P2P File Sharing

Giuseppe Anastasi

Pervasive Computing & Networking Lab (Perlab)

Dept. of Information Engineering, University of Pisa

E-mail: giuseppe.anastasi@iet.unipi.it

Website: www.iet.unipi.it/~anastasi/

UNIVERSITÀ DI PISA

PerLab

Acknowledgements

Ken Christensen, University of South Florida, USA

Ilaria Giannetti, University of Pisa, Italy

Marco Conti, Andrea Passarella, IIT-CNR, Italy

Helmut Hlavacs, University of Vienna, Austria

Koc University, Istanbul, July 5, 2012

Acknowledgments

PerLab

www.cost.esf.org

Action IC0804 – Energy Efficiency in Large-scale Distributed Systems

- ICT Action IC0804
  - Energy Efficiency in Large-Scale Distributed Systems
  - Starting date : 23/01/2009
  - End of action : 04/05/2013
  - Additional info at

http://www.irit.fr/cost804/

Overview

PerLab

- Introduction
- Motivations for
  - Energy Efficient Internet
  - Energy Efficient P2P File Sharing
- Energy Efficient BitTorrent
  - Green BitTorrent
  - Proxy-based BitTorrent
  - Download Sharing
- Conclusions

The Energy Problem

PerLab

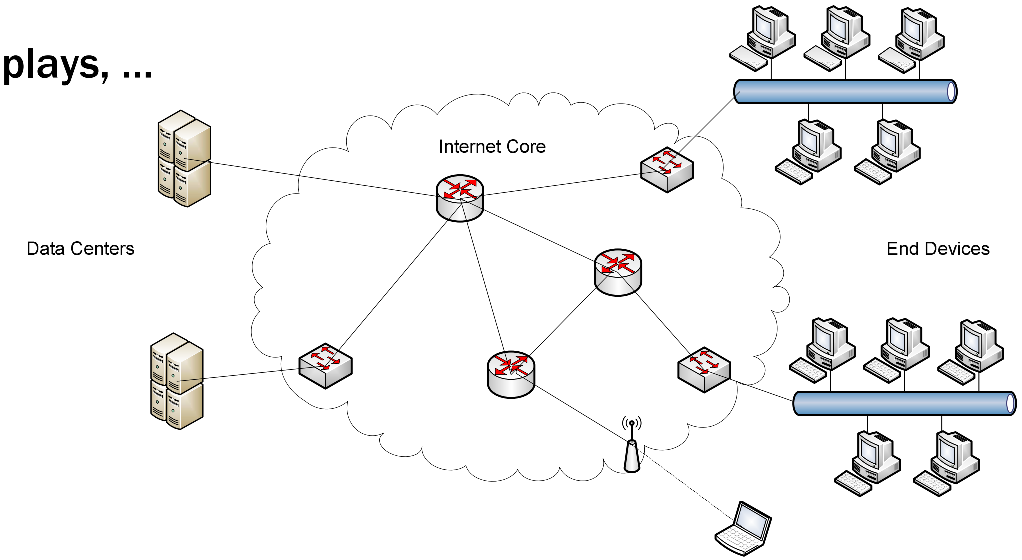
- Dramatic increase in the global energy consumption
  - Energy is a not renewable and limited resource
  - Environmental pollution and planetary overheating
- Energy consumption of Internet
  - 74TWh/year in US (equivalent to $ 6 billions)
  - 2-3% of the total energy consumption in US
  - About 1/3 of this energy could be saved by simple power management techniques

(Source: Lawrence Berkeley National Laboratory, USA, 2006)

Where Energy is consumed

PerLab

- Internet Core
  - Routers , Switches, Access Points, Links
- Data Centers
  - Servers
- User Devices
  - PCs, Printers, Displays, …



Edge Devices

PerLab

- Most of the energy wastes occurs at edges
  - Data centers
  - User devices
- Energy Consumption
  - Data Centers: 2 TWh per year
  - User Devices: 16 TWh per year
    - ⇨ User devices are widespread and very numerous
    - ⇨ User devices are often left powered on even if idle
    - ⇨ People typically do not pay attention to energy issues

(Source: Lawrence Berkeley National Laboratory, USA, 2006)

On-going Research Projects

PerLab

- EARTH
  - Energy Aware Radio and neTwork Technologies
  - IP project, EU FP7, https://www.ict-earth.eu/
- BCG^2
  - Beyond Cellular Green Generation
  - Green Touch Consortium
- ECONET
  - low Energy COnsumption NETworks
  - 3-year IP project, EU FP7, https://www.econet-project.eu/
- TREND
  - Towards Real Energy-efficient Network Design
  - NoE, EU FP7, http://www.fp7-trend.eu/
- CoolEmAll
  - Energy efficient data centers
  - EU FP7, http://www.coolemall.eu/

Energy Efficiency in the Internet Core

PerLab

- Re-Engineering
  - More energy-efficient network devices through
    - ⇨ Energy-Efficient Silicon
    - ⇨ Complexity Reduction
- Dynamic Adaptation
  - The capacity of network elements is dynamically modulated so as to meet actual traffic loads
    - ⇨ Performance Scaling
    - ⇨ Idle Logic
- Sleeping/Standby
  - Unused network/device portions are put in low-power mode

R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, **Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures**, *IEEE Communications Surveys and Tutorials*, Vol. 13, N. 2, 2011.

Personal Equipments

PerLab

- Some statistics about people behavior
  - 43,5% of UK population uses PC at work and
    - ⇨ 18% *never* powers it off
    - ⇨ 16% *sometimes* powers it off
  - Energy wastage corresponding to
    - ⇨ 153 millions of €
    - ⇨ 700.000 tons of CO2
- Motivations for not powering off
  - It is no so important
  - It takes some time and I am always in a hurry
  - I simply forget to power off
  - I don't want to lose my work
  - Nobody else turns PC off , so ...
  - ....

(Source: National Energy Foundation, UK, 2007)

Causes for energy wastes

PerLab

- PCs left on for
  - Laziness, Omissions, ...
- PCs intentionally left on for maintaining connectivity
  - Remote login
  - Automatic software upgrades
- PCs intentionally left on for
  - P2P file sharing applications

Possible Solutions

PerLab

- Centralized Shutdown
  - Already used in data centers and labs
  - No flexibility

Laziness
Omissions

- Power Manager [Chi10]
  - Better flexibility
  - Luca Chiaraviglio, Marco Mellia, PoliSave: Efficient Power Management of Campus PCs, IEEE SoftCOM - 18th *International Conference on Software, Telecommunications and Computer Networks*, Bol, Croatia, September 2010

- Context-aware Power Management [Har05]
  - Uses low-power sensors/devices to predict the user's intention to use/not use the PC
  - [Har05] C. Harris, V. Cahill, Power Management for Stationary Machines in a Pervasive Computing Environment, *Proc. 38th Hawaii International Conference on System Sciences*, 2005.

Possible Solutions

PerLab

- **Network Connectivity Proxy (NCP) [Jim08]**
  - **Based on proxying + Magic Packet**
    - ⇨ **Somniloquy [Aga09]**
    - ⇨ **Sleep Server [Aga10]**

Permanent Connectivity

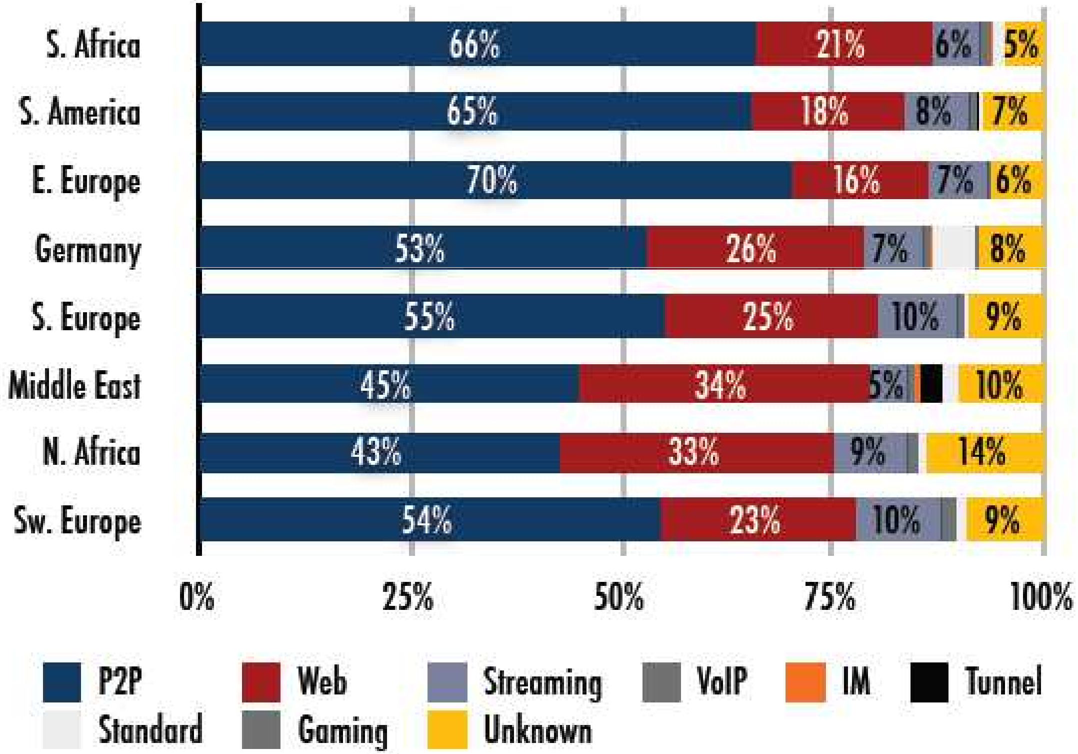
- [Jim08] M. Jimeno, K. Christensen, B. Nordman, **A Network Connection Proxy to Enable Hosts to Sleep and Save Energy**, *Proc. IEEE International Performance Computing and Communications Conference*, pp. 101-110, December 2008.
- [Aga09] Y. Agarwal, S. Hodges, J. Scott, R. Chandra, P. Bahl, R. Gupta, **Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage**, Proceedings USENIX Symposium on Networked System Design and Implementation (NSDI, 2009), Boston, MA, USA, April 22-24, 2009.
- [Aga10] Y. Agarwal, S. Savage, and R. Gupta, **SleepServer: Energy Savings for Enterprise PCs by Allowing them to Sleep**, Proceedings of the USENIX Annual Technical Conference, June 2010.

- **Energy-aware Applications and Protocols**

P2P Applications

PerLab

P2P Traffic in Internet

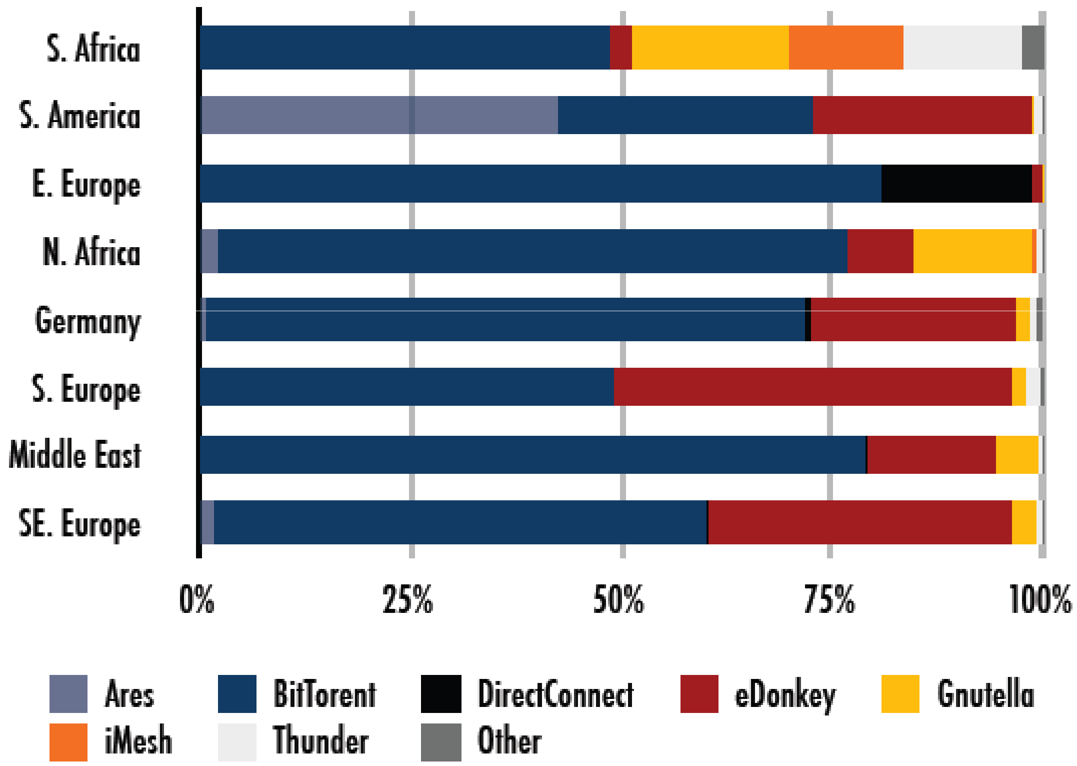


Source: Ipoque 2008 / 2009

P2P Applications

PerLab

P2P Traffic originated by BitTorrent



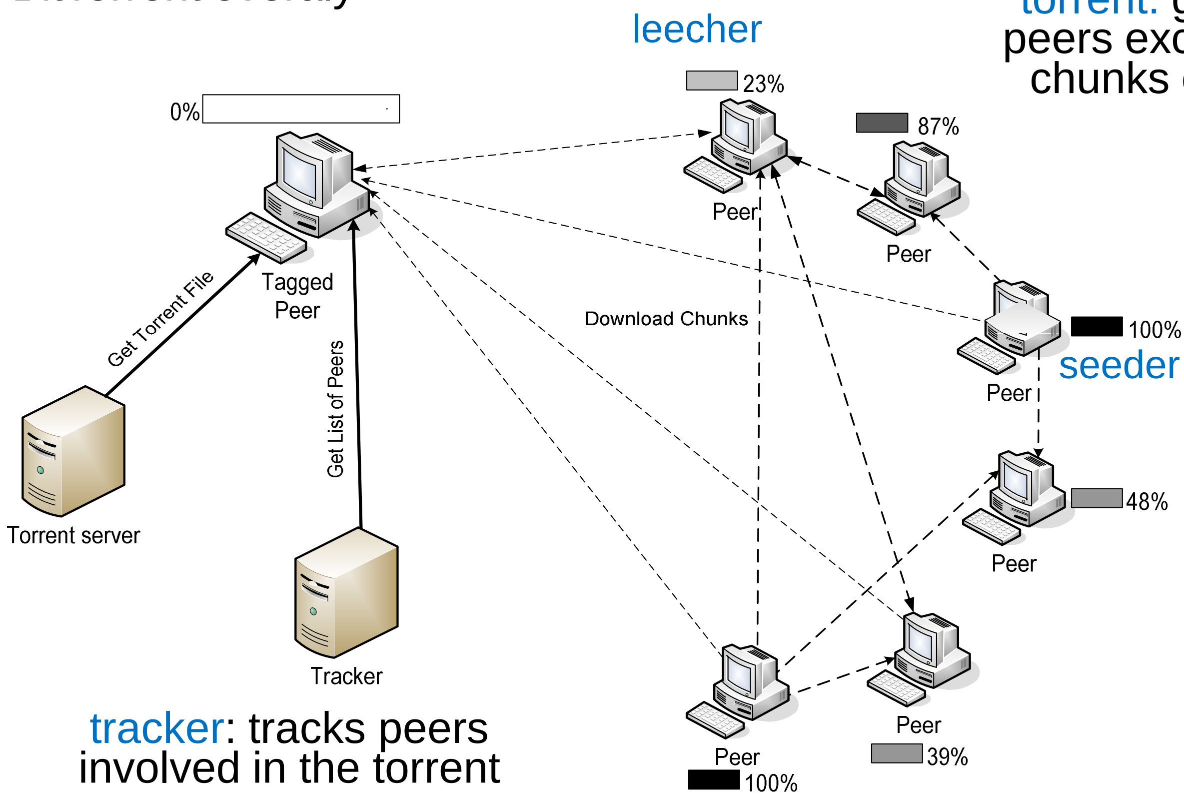
Source: Ipoque 2008 / 2009

BitTorrent

BitTorrent Terminology

peer: node of the BitTorrent overaly

torrent: group of peers exchanging chunks of a file

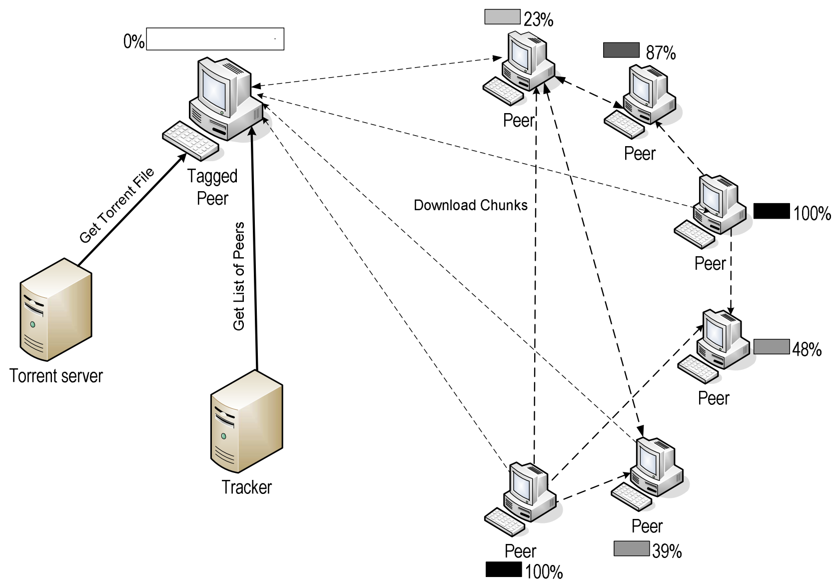


tracker: tracks peers involved in the torrent

BitTorrent Protocol

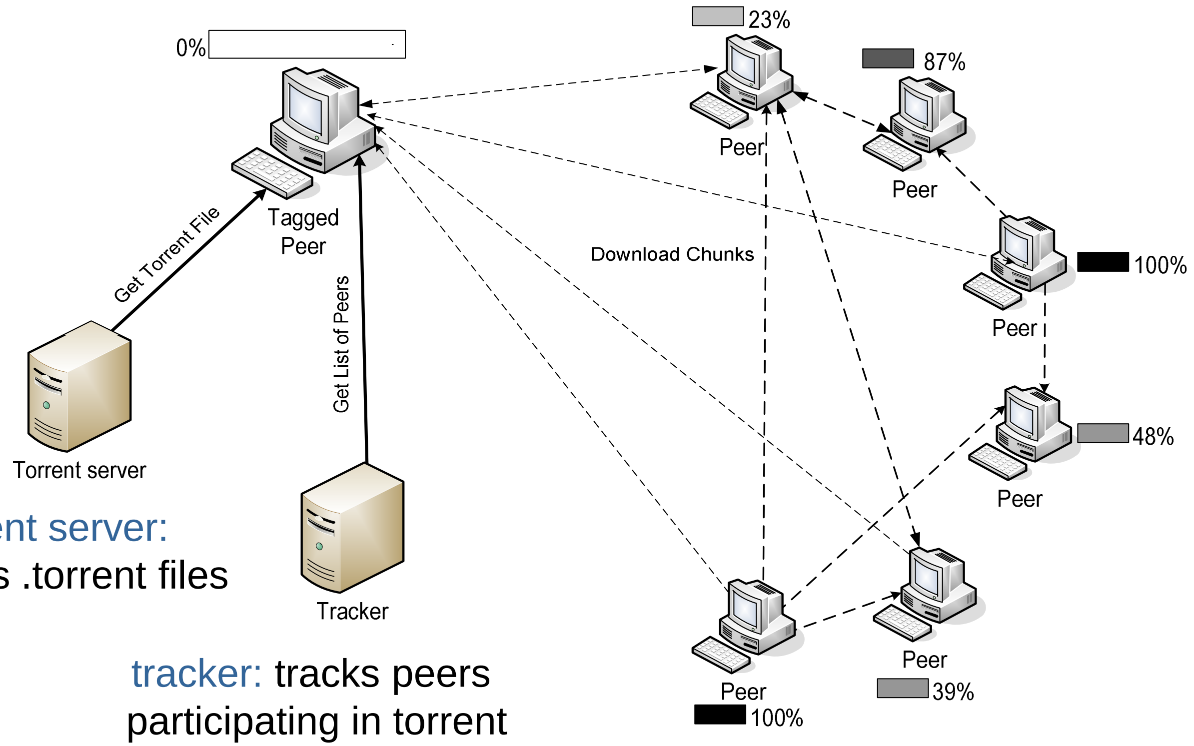
PerLab

- File divided into 256KB *chunks*.
- Peers download chunks from a multitude of other peers
  - Instead from a single server, as in the traditional C/S approach
- While downloading, peers upload chunks to other peers
- Once a peer has entire file, it may (selfishly) leave or (altruistically) remain
  - Peers may come and go



BitTorrent Protocol

PerLab



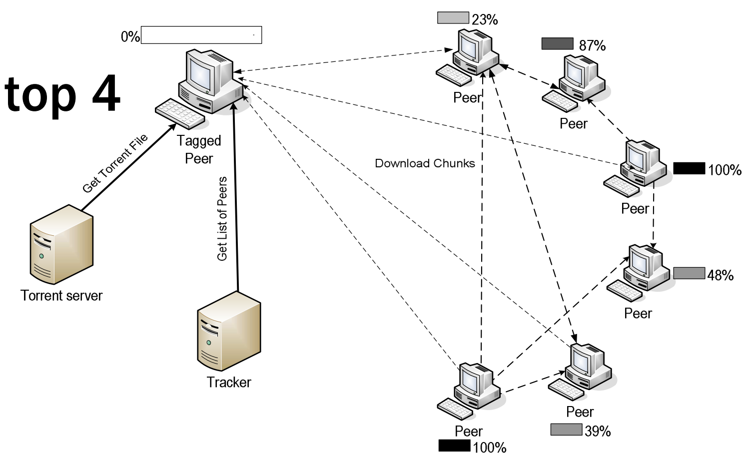
torrent server: contains .torrent files

tracker: tracks peers participating in torrent

Tit-for-Tat Policy

PerLab

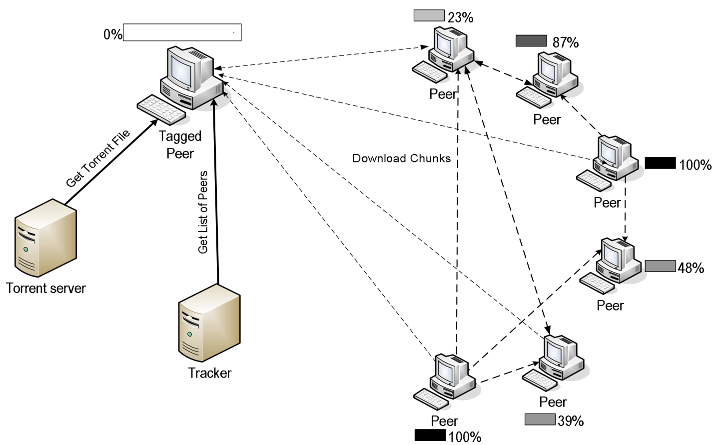
- A peer continuously measures the bit rate achieved by each of its neighbors
- And uploads chunks to the 4 neighbors from which it is achieving the *highest bit rate*
  - re-evaluate top 4 every 10 secs
- Every 30 secs: randomly select another peer, starts sending chunks
  - newly chosen peer may join top 4
  - "optimistically unchoke"



Rarest First Policy

PerLab

- At any given time, different peers have different file chunks
- Periodically, a peer asks each neighbors for the list of chunks they have.
- And sends requests for missing chunks, giving priority to chunks that are less spread
  - *rarest first*



BitTorrent and Energy Efficiency

BitTorrent

P2P and Energy Efficiency

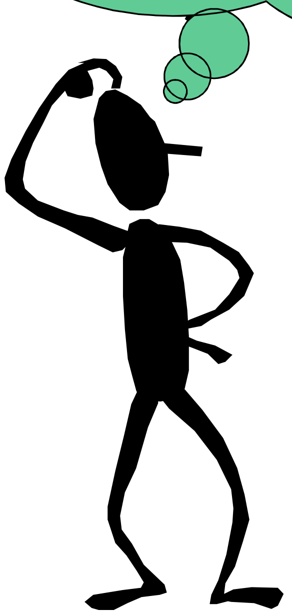
PerLab

- Energy Efficiency in BitTorrent (and P2P) has not received significant attention so far
  - Most of the proposed optimizations aimed at improving performance
  - Only indirectly address energy efficiency
- Motivations
  - Energy consumption is distributed among a large number of peers
  - No central entity pays for the energy consumed by P2P applications (e.g., BitTorrent)

Key Question

PerLab

Is it more energy efficient to download a song from BitTorrent rather than iTunes?



S. Nedevschi, S. Ratnasamy, J. Padhye
Hot Data Centers vs. Cool Peers,
Proc. Workshop on Power Aware Computing and Systems (HotPower 2008),
San Diego, CA, USA, Dec 7, 2008.

P2P vs. Client-Server

PerLab

- Energy consumption model of P2P and CS is not a trivial task
- CS is typically more energy efficient than P2P in the *network side*
  - Multiple hops needed to distribute the file between peers
  - Membership maintenance
- P2P is likely to be more energy efficient than CS at end systems
  - Most of the energy consumed at data centers is due to cooling
  - They do not consider the baseline energy consumption of PCs

S. Nedevschi, S. Ratnasamy, J. Padhye, **Hot Data Centers vs. Cool Peers**, *Proc. Workshop on Power Aware Computing and Systems (HotPower 2008),* San Diego, CA, USA, Dec 7, 2008.

P2P vs. Client-Server

PerLab

- P2P is a mean of reducing energy consumption at data centers
  - Users are both content consumers and content distributors
- P2P pushes energy consumption out of data centers
  - into the homes of users

J. Blackburn, K. Christensen, **A Simulation Study of a New Green BitTorrent**, *Proc. International Workshop on Green Communications (GreenComm 2009)*, Dresden, Germany, June 2009.

BitTorrent and Energy Efficiency

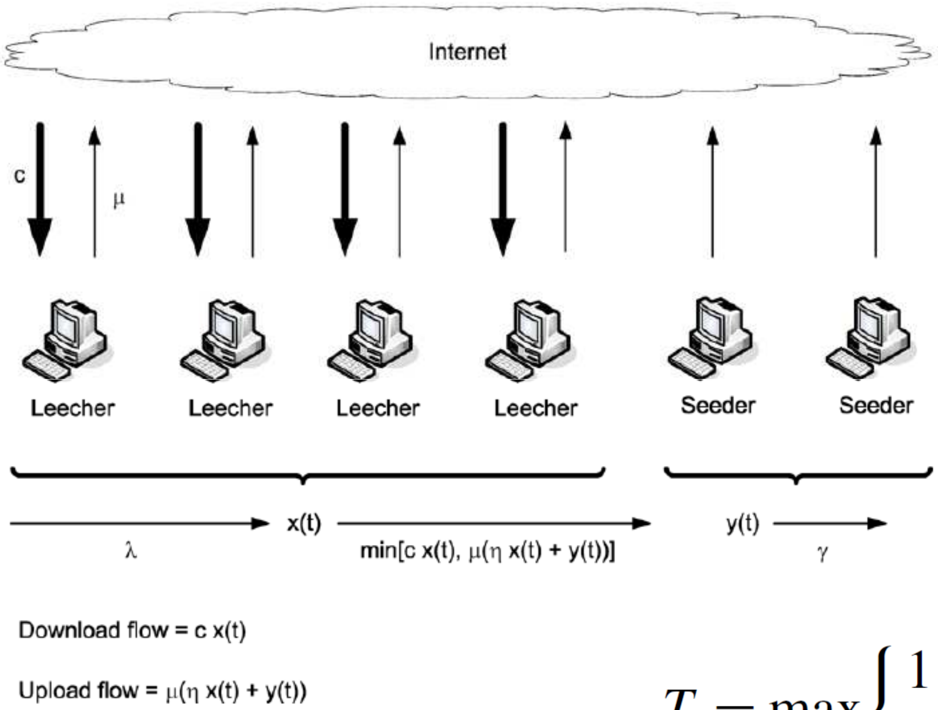
PerLab

BitTorrent is not "energy friendly" for users

- Peers must remain connected during the entire download process
  - Powering off a peer stops the download process
- Coordinated strategies for energy efficiency are unfeasible
  - They would be in contrast with the BitTorrent design paradigm

Average Download Time

Popular Files



$$T = \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \frac{1}{\gamma}\right)\right\}$$

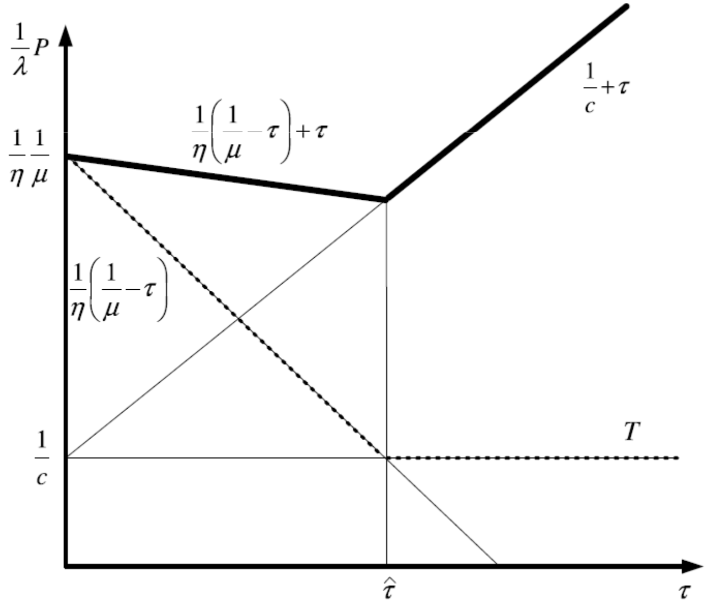
D. Qiu, R. Srikant, **Modeling and Performance Analysis of BitTorrent-like Peer-to-Peer Networks**, *Proceedings of ACM Sigcomm 2004*, Portland, Oregon, USA, pp 367–378.

Average Power Consumption

PerLab

Popular Files – Each peer consumes one unit of power

$$P = \lambda \max\left\{\frac{1}{c}, \frac{1}{\eta}\left(\frac{1}{\mu} - \tau\right)\right\} + \lambda\tau$$



Optimal Average Seeding Time

$$\hat{\tau} = \frac{1}{\mu} - \frac{\eta}{c}$$

H. Hlavacs, R. Weidlich, T. Treutner, **Energy Efficient Peer-to-Peer File Sharing**, *Journal of Supercomputing*, to appear. Available online at http://www.springerlink.com/content/k56811t52n134811/

Limits of the previous approach

- The optimal avg. seeding time depends on a number of parameters
  - which are not under the user control
  - and vary over time
- The optimal seeding time is unknown to the user
- The user is typically selfish and lazy
  - He is mainly interested in getting a copy of the file
    - ⇨ Not in optimizing the energy consumption of the system at the cost of an increase in his/her personal consumption
  - He typically leaves the swarm *after some time* the file has been downloaded

Proposed Approaches

PerLab

- **Green Bit Torrent**
  - J. Blackburn, K. Christensen, **A Simulation Study of a New Green BitTorrent**, *Proc. International Workshop on Green Communications (GreenComm 2009)*, Dresden, Germany, June 2009.
- **Proxy-based BitTorrent**
  - G. Anastasi, I. Giannetti, A. Passarella, **A BitTorrent Proxy for Green Internet File Sharing: Design and Experimental Evaluation**, *Computer Communications*, Vol. 33, N. 7, pp. 794-802, May 2010. Elsevier.
  - I. Giannetti, G. Anastasi, M. Conti, **Energy-Efficient P2P File Sharing for Residential BitTorrent Users**, *Proc. IEEE International Symposium on Computers and Communications (ISCC 2012),* Cappadocia, Turkey, July 1-4, 2012.
- **Download Sharing**
  - H. Hlavacs, R. Weidlich, T. Treutner, **Energy Efficient Peer-to-Peer File Sharing,** *Journal of Supercomputing*, available online at http://www.springerlink.com/content/k56811t52n134811/.
- **Energy-Efficient Mobile BitTorrent**
  - I. Kelenyi, A. Ludanyi, J. Nurminen, I. Pusstinen, **Energy-efficient Mobile BitTorrent with Broadband Router Hosted Proxies**, Proc. *IFIP Wireless and Mobile Networking Conference (WMNC 2010)*, Budapest, Hungary, October 13-15, 2010.
  - I. Kelenyi, A. Ludanyi, J. Nurminen, **BitTorrent on Mobile Phones – Energy Efficiency of a Distributed Proxy Solution**, Proc. *International Green Computing Conference (IGCC 2010)*, Chicago, USA, August 15-18, 2010.

Green BitTorrent

A Simulation Study of a New Green BitTorrent

Jeremy Blackburn and Ken Christensen
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida USA
jhblackb@mail.usf.edu, christen@cse.usf.edu

Proc. international Workshop on Green Communications (GreenComm 2009), Dresden, Germany

Abstract— **The use of P2P technologies, such as BitTorrent, to distribute legal content to consumers is actively being explored as a means of reducing both file download times and the energy consumption of data centers. This approach pushes the energy use out of the data centers and into the homes of content consumers (who are also then content distributors). The current BitTorrent protocol requires that clients must be fully powered-on to be participating members in a swarm. In this paper, we show that simple changes to the BitTorrent protocol, including long-lived knowledge of sleeping peers and a new wake-up semantic, can enable clients to sleep when not actively downloading or uploading yet still be responsive swarm members. Using ns-2 we simulate a green BitTorrent swarm. We show that significant energy savings are achievable with only a small performance penalty in increased file download time.**

commercial offerings such as the VUDU on-demand movie service [22] are available that allow users to access digital video files via a P2P network. Open problems in digital rights management, charging models, and excessive bandwidth use between ISPs for P2P are currently being explored and solved (for example, by Choffnes and Bustamente [3]).

We foresee a future where residences connected to a broadband link may have a file sharing device, or unit, as part of a set-top box, or on the side of the house. This unit could be owned and controlled by the operator and would allow for on-demand content access to the local customer and distribution of operator-owned content to other customers. With this possible future model, energy use for content distribution will move largely from the data center to the consumer. If we assume

Motivations

PerLab

- Using the optimal seeding time is unfeasible
  - It cannot be calculated
  - The user is selfish and lazy
- Green BitTorrent tries to implement the idea behind the above ideal approach
  - Peers should remain available after they have completed their download
  - At the same time, they should not consume energy for others
  - The user should not be directly involved

Basic Idea

PerLab

- Extended version of BitTorrent to allow power management at the user' s PC
- Allows BitTorrent peers to sleep …
  - when they are not downloading/uploading

  … while remaining active members of the swarm
- Ensures backwards compatibility
  - Green BitTorrent peers can operate with legacy BitTorrent peers
    - ⇨ Some performance degradation

Challenge

PerLab

- BitTorrent relies on TCP connections
  - A peer establish at TCP connections with its neighbors
    - ⇨ at most `max_connect`
- A client stopping a TCP connection is considered disconnected
- Challenge:
  - ⇨ Is there a way to maintain TCP connectivity while sleeping?

States of another peer

PerLab

- Connected
  - A peer that this peer has an active TCP connection with.
    - ⇨ File pieces can be uploaded and downloaded on the connection
- Sleeping
  - A peer that has disconnected its TCP connection with this peer.
    - ⇨ The TCP connection must be re-established before file pieces can be uploaded or downloaded.
- Unknown
  - A peer that has been given to this peer by the tracker, and it is unknown if the peer is sleeping or awake.

Additional mechanisms

PerLab

- Inactivity timer
  - Restarted whenever a download/upload activity occurs
- Magic packet
  - Used to wakeup a remote sleeping peer
  - The receiving network interface triggers a wakeup of the PC

Event Management

PerLab

- Upon inactivity timer expiration
  - close all TCP connections
  - enter *sleep* state
- If (count of connected peers < max_connect)
  - randomly selected peer in my peer list
  - send *wake-up* message to peer p
  - try to connect to peer p
  - if (TCP connection established)

    then p.state = connected

    else remove peer p from my peer list

Event Management

PerLab

- Upon receiving a wakeup message from peer *p*
  - if (TCP connection is established from peer p) then my.state = connected
  - send my file contents bit-field to peer p
  - run choking algorithm

Simulation Setup

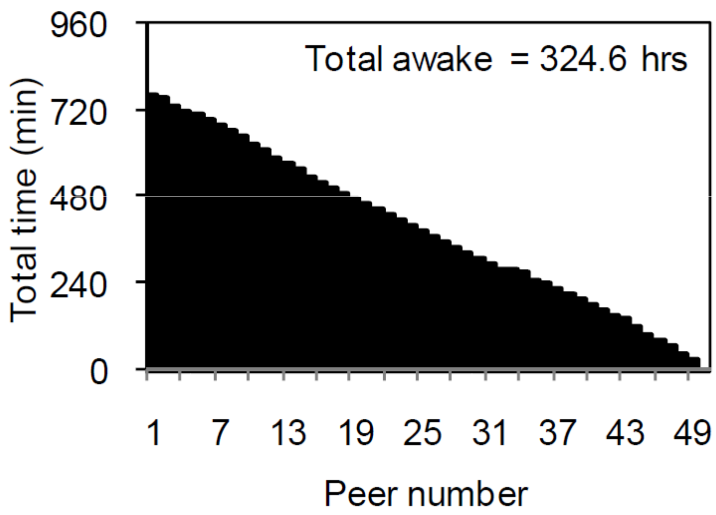
PerLab

- NS-2 simulation tool
- Single swarm
  - 50 peers (a typical large swarm)
  - one additional initial seed peer (never sleep in any of the cases)
- max_connect =5
- File size =1 GB (small video file)
- Upload data rate = 2 Mbps
- Download data rate = 10 Mbps
- Wake-up/Sleep transition time = 300 ms
- Peer inter-arrival time = Poisson
  - 0, 0.5, 1, 2, 4, 8, 16, 32 msec

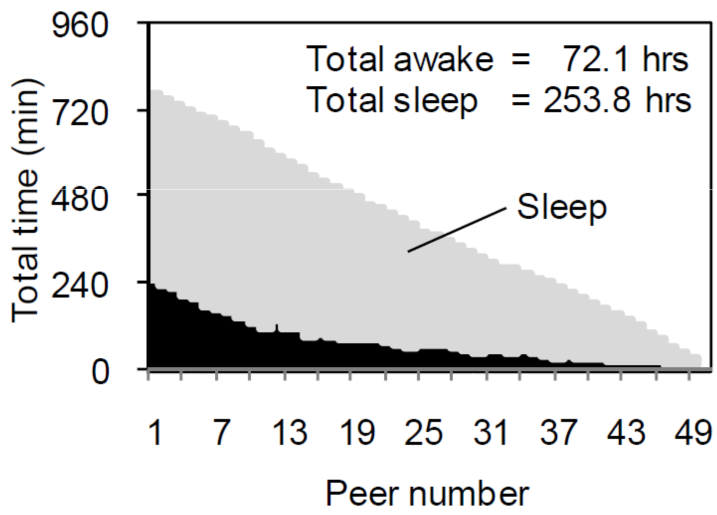
Energy Consumption

PerLab

Legacy BitTorrent



Green BitTorrent



Energy Savings = 77.8%

Increase in the Download Time

PerLab

	Mean client interarrvial time							
Client number	**0 *min***	**0.5 *min***	**1 *min***	**2 *min***	**4 *min***	**8 *min***	**16 *min***	**32 *min***
Client #1	4.8 %	1.5 %	1.8 %	1.9 %	1.9 %	1.9 %	2.1 %	1.6 %
Client #25	6.7	3.7	–0.7	5.2	11.0	44.8	30.9	18.1
Client #50	7.0	4.6	6.0	15.5	32.8	18.7	9.4	8.2
Average for all	4.9	3.4	3.5	5.2	13.7	23.0	22.0	18.3

Limits of Green BitTorrent

PerLab

- Green BitTorrent relies on magic packets to wakeup sleeping peers
  - This mechanism may not be available on some NICs
- Privacy/security issues
  - A remote peer is able to know whether your PC is active or sleeping
  - A malicious user could wake up your PC, just to increase your energy consumption
  - An attacker could sleep and wake up your PC with a high frequency to damage your hardware

Proxy-based BitTorrent

Computer Communications 33 (2010) 794–802

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

ELSEVIER

A BitTorrent proxy for Green Internet file sharing: Design and experimental evaluation

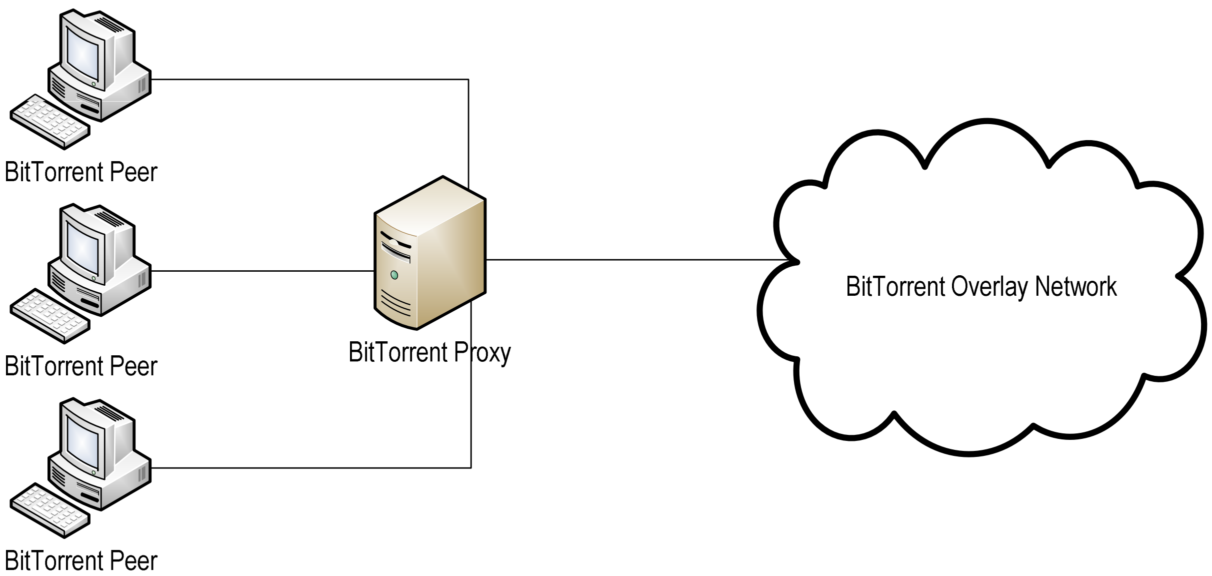
Giuseppe Anastasi [a], Ilaria Giannetti [a], Andrea Passarella [b,*]

[a] Dept. of Information Engineering, University of Pisa, via Diotisalvi 2, 56122 Pisa, Italy
[b] IIT-CNR, via G. Moruzzi 1, 56124 Pisa, Italy

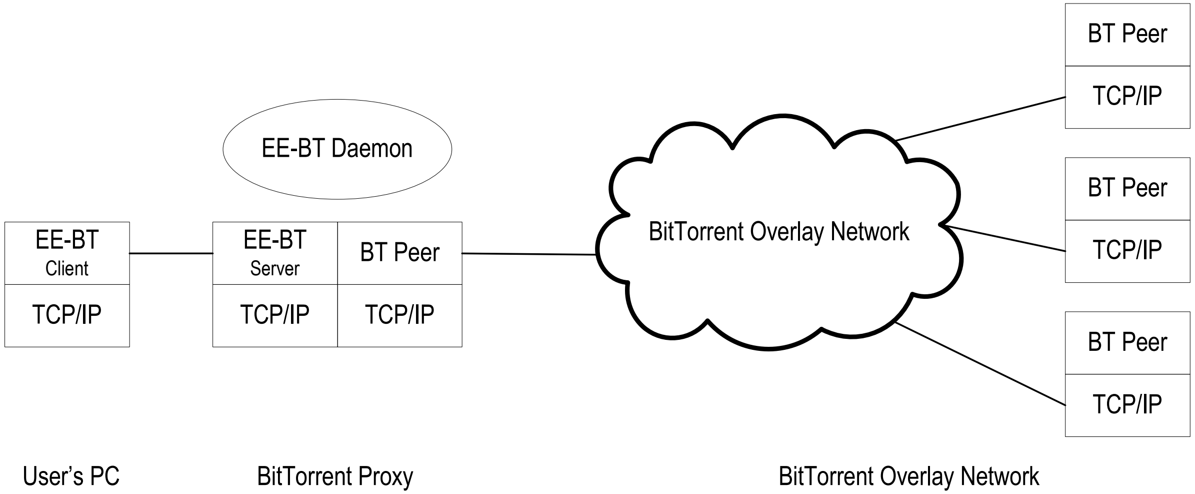
EE-BitTorrent

PerLab

- Proxy-based version of BitTorrent
- One BitTorrent Proxy for a large number of peers (PCs)



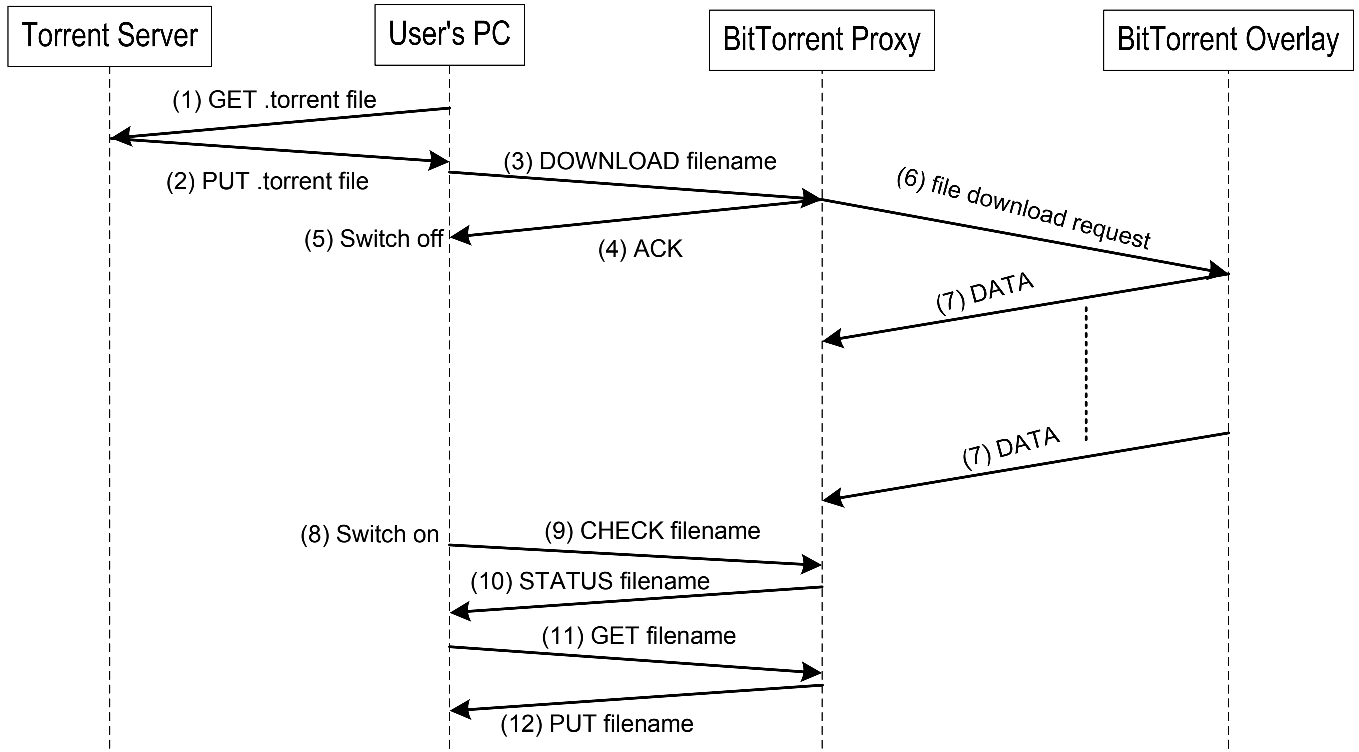
Architecture



- EE-BitTorrent (EE-BT)
  - Clients and Proxy (clients side)
  - Client/Server scheme
- Traditional BitTorrent (BT Peer)
  - Proxy (P2P Network side)

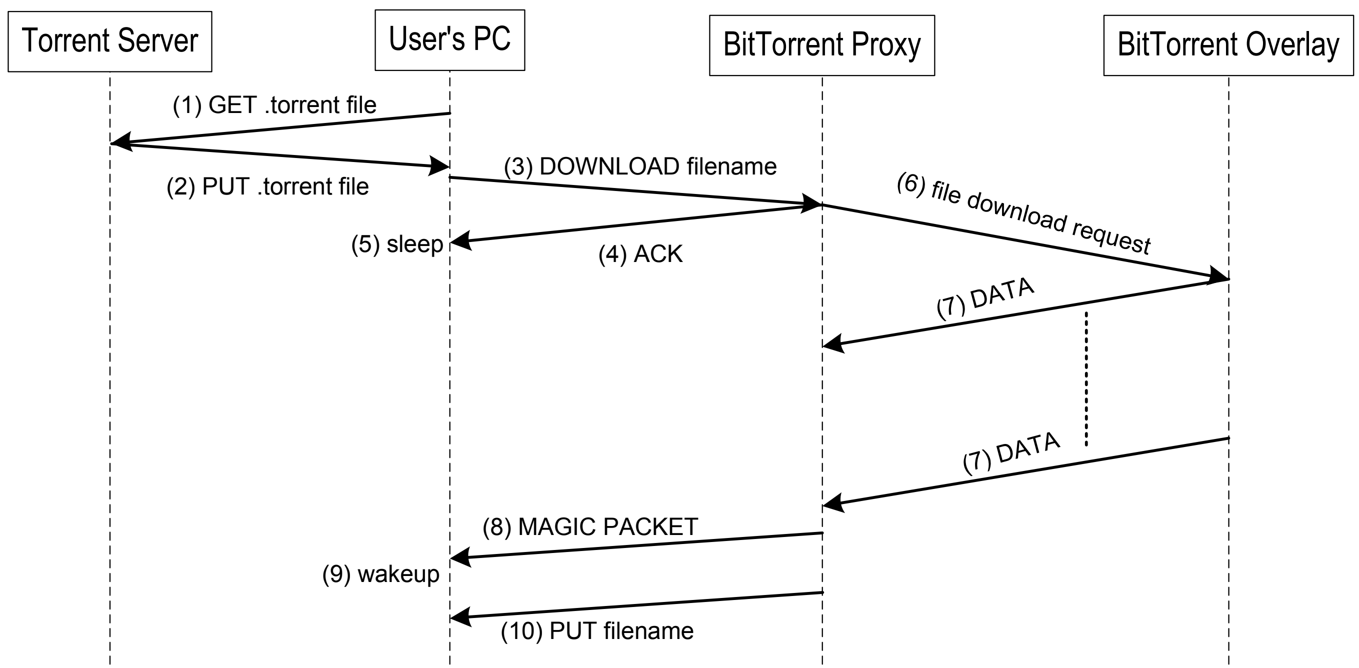
EE-BT Protocol – version 1

PerLab



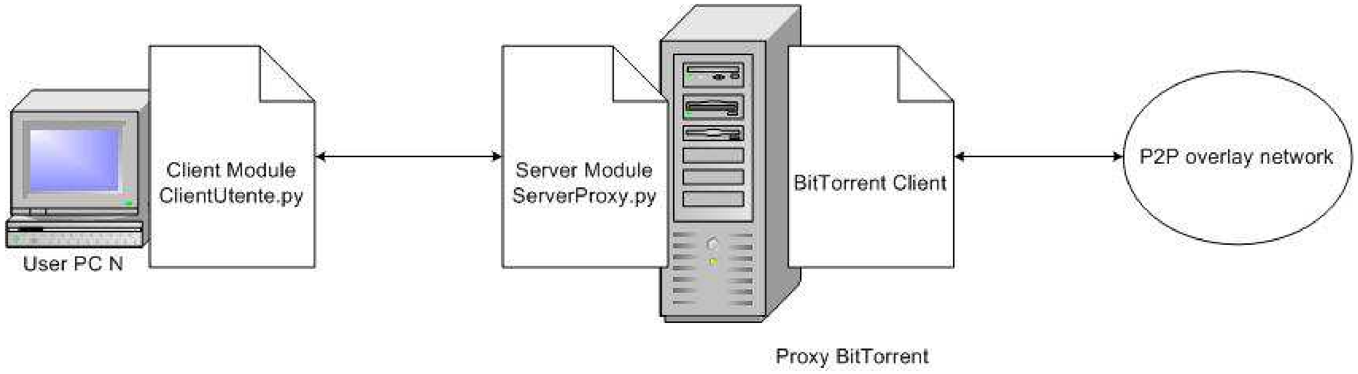
EE-BT Protocol – version 2

PerLab



Implementation

PerLab



- Energy Efficient BitTorrent modules
  - Client – Server scheme
    - Server (Proxy)
    - Client (user PC)
- Programming language: Python
- Libtorrent Rasterbar: library for BitTorrent
- Command-line BitTorrent client

Where to place the BitTorrent Proxy?

PerLab

- Departmental Scenario
  - Dedicated Machine
  - Machine already used for some other services
  - Better control (in addition to energy efficiency)
- Residential Users
  - Several Options
    - ⇨ Provided (for free) by ISP
    - ⇨ Cloud Proxy leased to users
    - ⇨ Proxy maintained and shared by a group of users (Social Proxy)
  - The proxy should have a high-speed connection
  - The proxy should be as close as possible to users

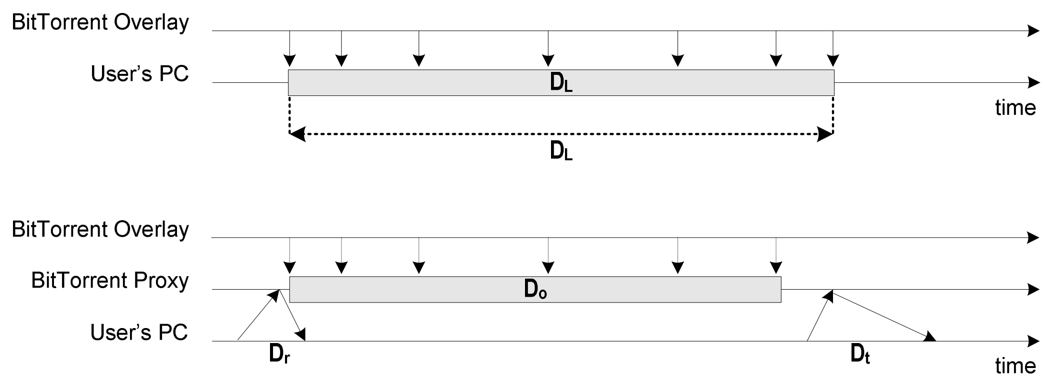
Performance metrics

PerLab

- **Relative Energy Saving**

$$S = 1 - \frac{E_P}{E_L}$$

- **Assumptions**
  - **All PCs and proxy have the same power consumption**



$$S_{usr}(n) = 1 - \frac{D_P(i)}{D_L(i)} = 1 - \frac{D_r(i) + D_t(i)}{D_L(i)}$$

$$S_{sys}(n) = 1 - \frac{\sum_{i=1}^{n} D_o(i) + \sum_{i=1}^{n} [D_r(i) + D_t(i)]}{\sum_{i=1}^{n} D_L(i)}$$

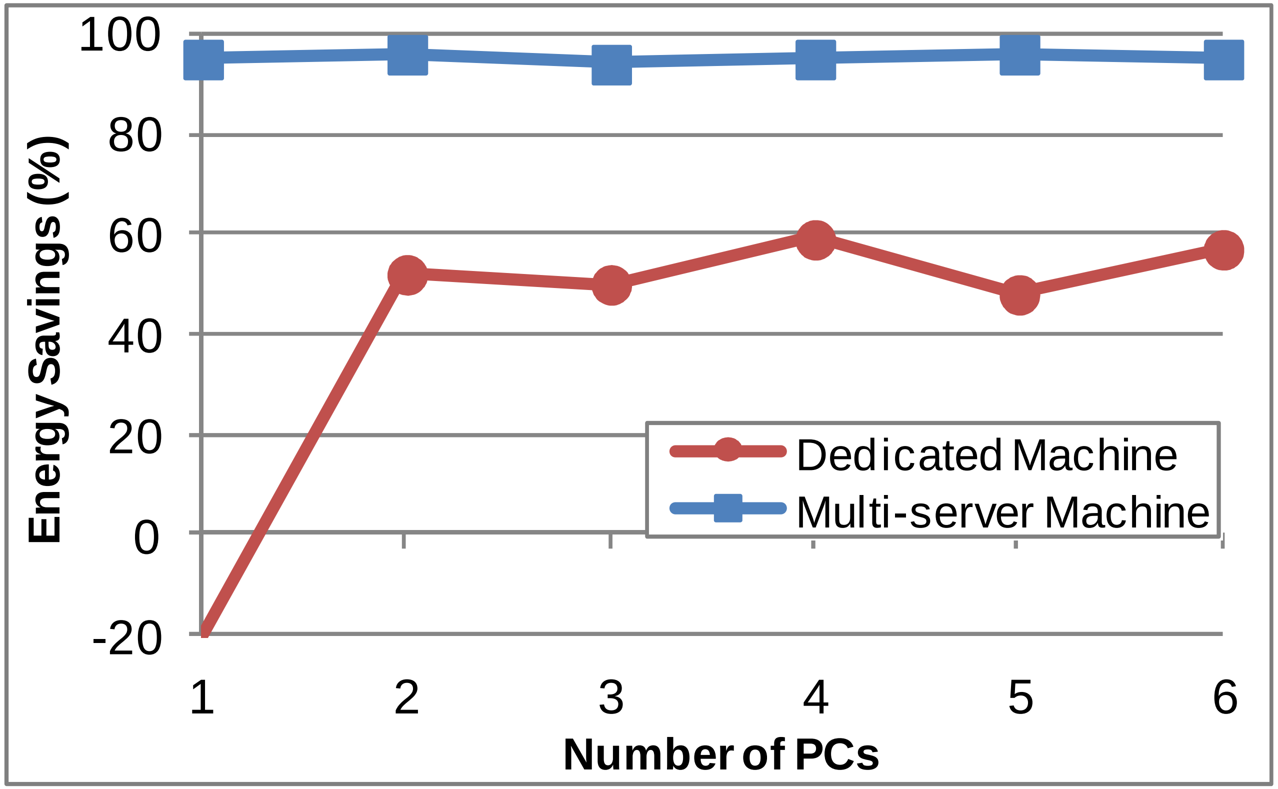
Dept. Scenario

PerLab

- Ethernet LAN
  - Both the proxy and peers are connected to the same LAN
  - 100 Mbps
- Downloaded files
  - Size: ~4GB [3.95 GB – 4.71 GB]
  - Initial number of seeds: 200 – 800
- Methodology
  - Interleaved experiments with Legacy BitTorrent and Proxy-based BitTorrent
  - Experiments replicated several times per day and in different days

Relative Energy Savings

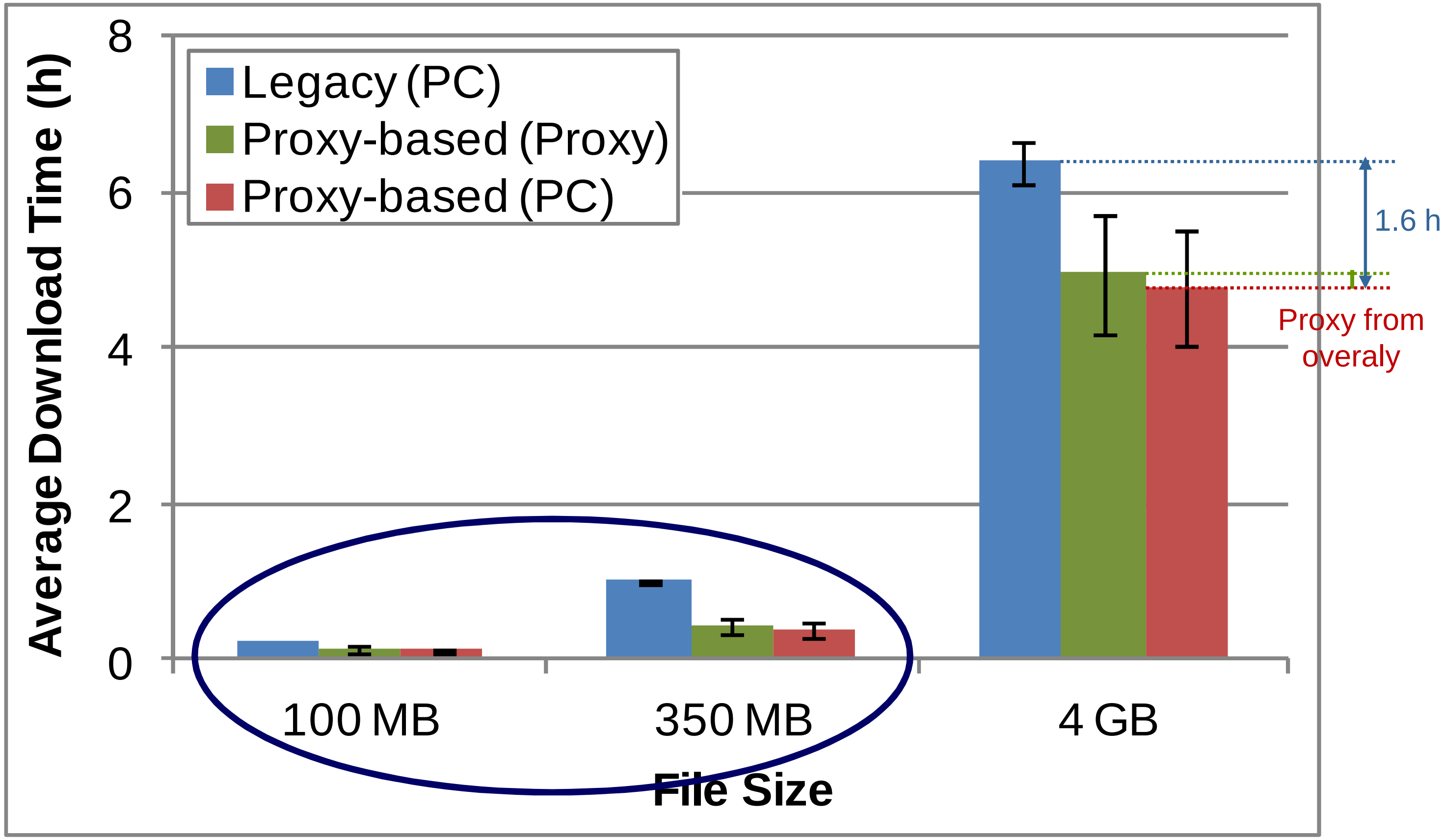
PerLab



Residential Scenario

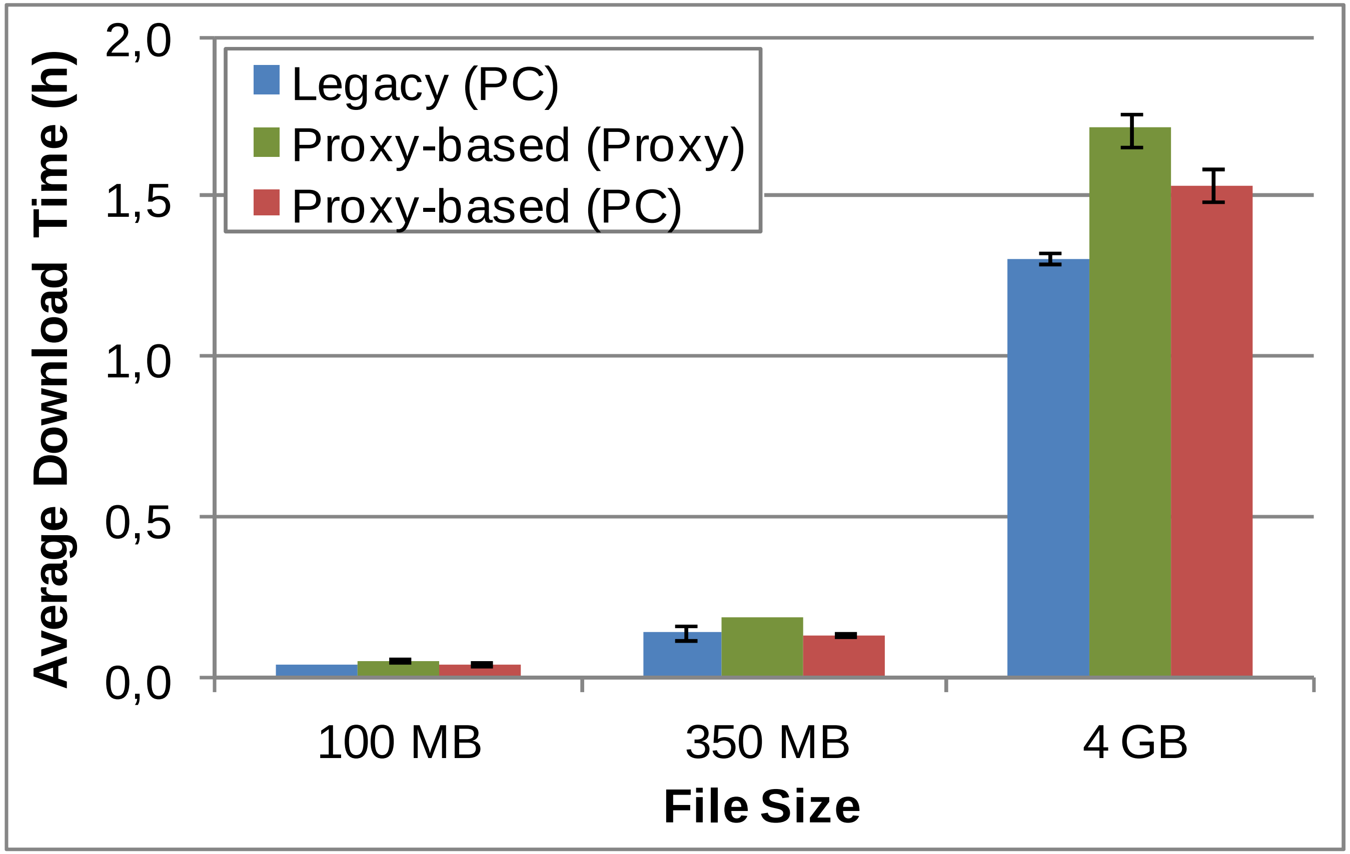
- ADSL Access Network
  - Downlink rate: up to 8 Mbps
  - Uplink rate: up to 512 Kbps
- Users
  - Up to 4 different users, at different locations
    - ⇨ A, B, C, D
- File Types:
  - 135 MB (Audio CD, MP3)
  - 350 MB (Episode of a TV Series, AVI)
  - 4 GB (Ubuntu 10.10 Distribution, ISO)

Single User at Location A



Single User at Location B

PerLab



Why a so different performance?

PerLab

- Same nominal uplink/downlink rates
  - 512 Kbps, 8 Mbps
- Very different actual uplink rate
  - The actual uplink rate at A is significantly lower than that experience at B
- *Tit-for-tat* penalizes users with low uplink rate
  - In the proxy-based version the download time only depends on the downlink rate

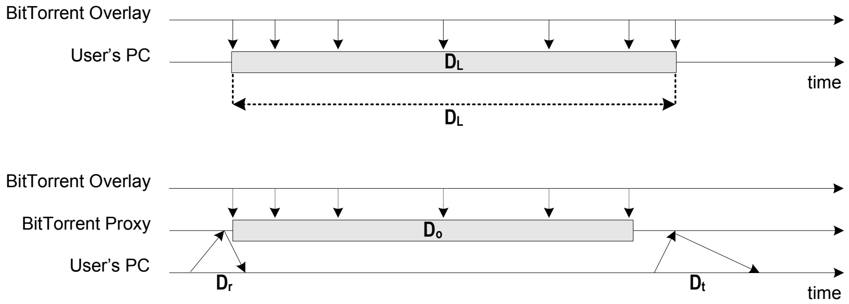
Location	Legacy	EE-BitTorrent
A	0.89 Mbps	2.86 Mbps
B	6.66 Mbps	6.18 Mbps

Single User

- **Energy Consumption**

$$S_{usr}(n) = 1 - \frac{D_r(i) + D_t(i)}{D_L(i)}$$

$$S_{sys}(n) = 1 - \frac{\sum_{i=1}^{n} D_o(i) + \sum_{i=1}^{n} \left[D_r(i) + D_t(i) \right]}{\sum_{i=1}^{n} D_L(i)}$$



Location	$S_{usr}(1)$	$S_{sys}(1)$
A	25,4%	-52,1%
B	-16,8%	-147,3%

Energy Savings

PerLab

- User and System Energy Savings

Location		1 PC	2 PCs	3 PCs	4 PCs
A	S_{usr}^{A}	25.4%	41,8%	33,2%	28.8%
	S_{sys}	-52.1%	-1,2%	0,8%	4.8%
B	S_{usr}^{B}	-16.8%	-14.5%	-19.1%	-19.8%
	S_{sys}^{B}	-147.3%	-1.2%	0.8%	4.8%

Lessons learned in residential scenarios

PerLab

- The available uplink rate strongly depends on the specific location
  - For a given location it also vary significantly over time
- EE-BitTorrent outperforms the legacy BitTorrent when the available uplink rate is low
- Otherwise, the legacy BitTorrent is the more convenient approach
- Selecting a priori the most convenient option is almost impossible

Adaptive BitTorrent (AdaBT)

AdaBT: Basic Idea

- Goal
  - Choosing the most convenient option, depending on the operating condition at run time
- Upon receiving a request for a file
  - Estimates the time required for downloading the file
    - ⇨ from the BitTorrent overlay (legacy BitTorrent)
    - ⇨ from the configured BitTorrent Proxy
  - Selects the most convenient option
- When using EE-BitTorrent
  - If file already available, starts download
  - Otherwise, switches off and reconnects later

AdaBT: Algorithm

PerLab

1	AdaBT_start_time = time();
2a	Start Legacy BitTorrent; Wait for the first chunk of data;
3a	Legacy_start_time=time();
4a	Wait for Q data;
5a	Legacy_end_time = time();
6a	T_L=Legacy_end_time-Legacy_start_time
7a	$R_L=Q/T_L$ // estimated rate with legacy protocol
2b	request S data from Proxy;
3b	Proxy_start = time();
4b	Wait for Q data;
5b	Proxy_end_time = time();
6b	T_P=Proxy_end_time-Proxy_start_time
7b	$R_P=Q/T_P$ // estimated rate from proxy
8	AdaBT_end_time=time();
9	T_{OH} = AdaBT_end_time-AdaBT_start_time;
10	**if** $(L/R_P+T_{OH}) < \alpha \cdot ((L-Q)/R_L)$ **then** use EE-BitTorrent
11	**else** use legacy BitTorrent;

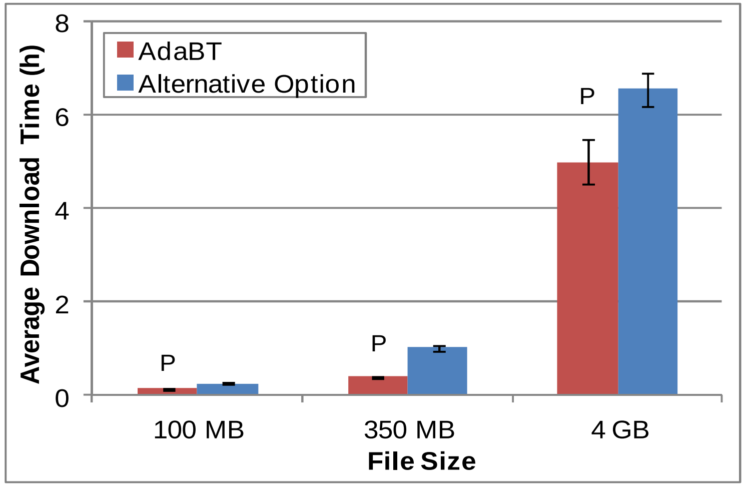
Giannetti, G. Anastasi, M. Conti
Energy-Efficient P2P File Sharing for Residential BitTorrent Users
Proc. *IEEE International Symposium on Computers and Communications (ISCC 2012)*, Cappadocia, Turkey, July 1-4, 2012.

AdaBT: Experimental Results

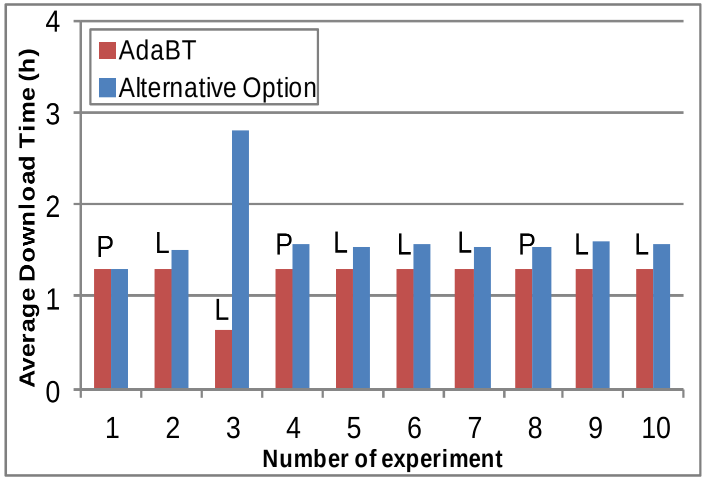
PerLab

File size = 4 GB

Location A



Location B



Summary

PerLab

- Energy Efficiency in BitTorrent
- EE-BitTorrent
  - Proxy-based solution
  - Implementation
- Experimental Analysis
  - In a dept. scenario EE-BitTorrent provides a significant reduction in energy consumption
  - In a residential scenario performance is strongly influenced by the available uplink rate
- AdaBT
  - adaptive solution that dynamically selects the most convenient option, based on operating conditions

BitTorrent with Download Sharing

J Supercomput
DOI 10.1007/s11227-011-0602-8

Energy efficient peer-to-peer file sharing

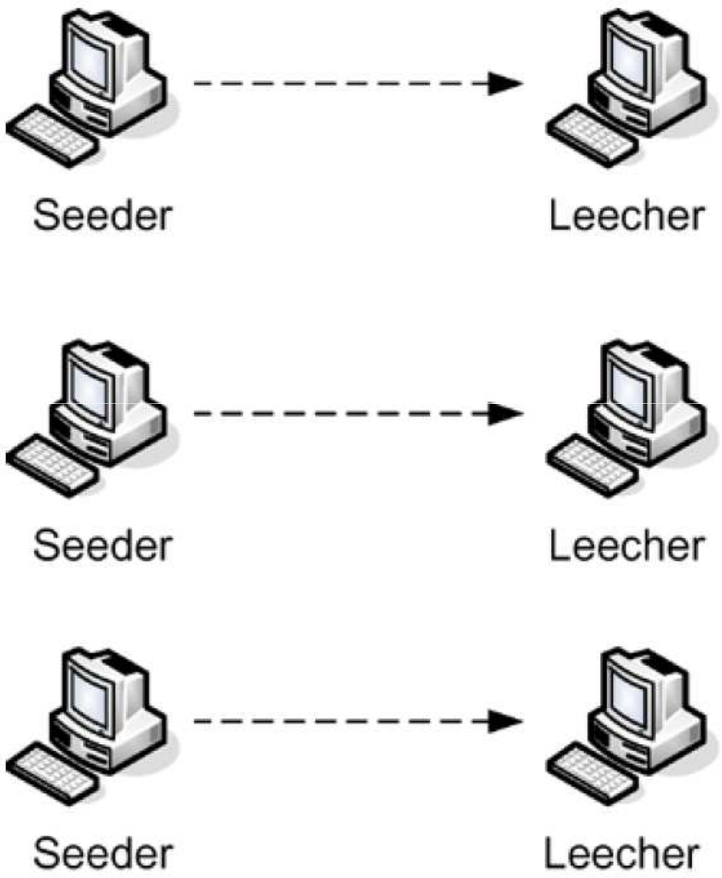
**Helmut Hlavacs · Roman Weidlich ·
Thomas Treutner**

Overview

PerLab

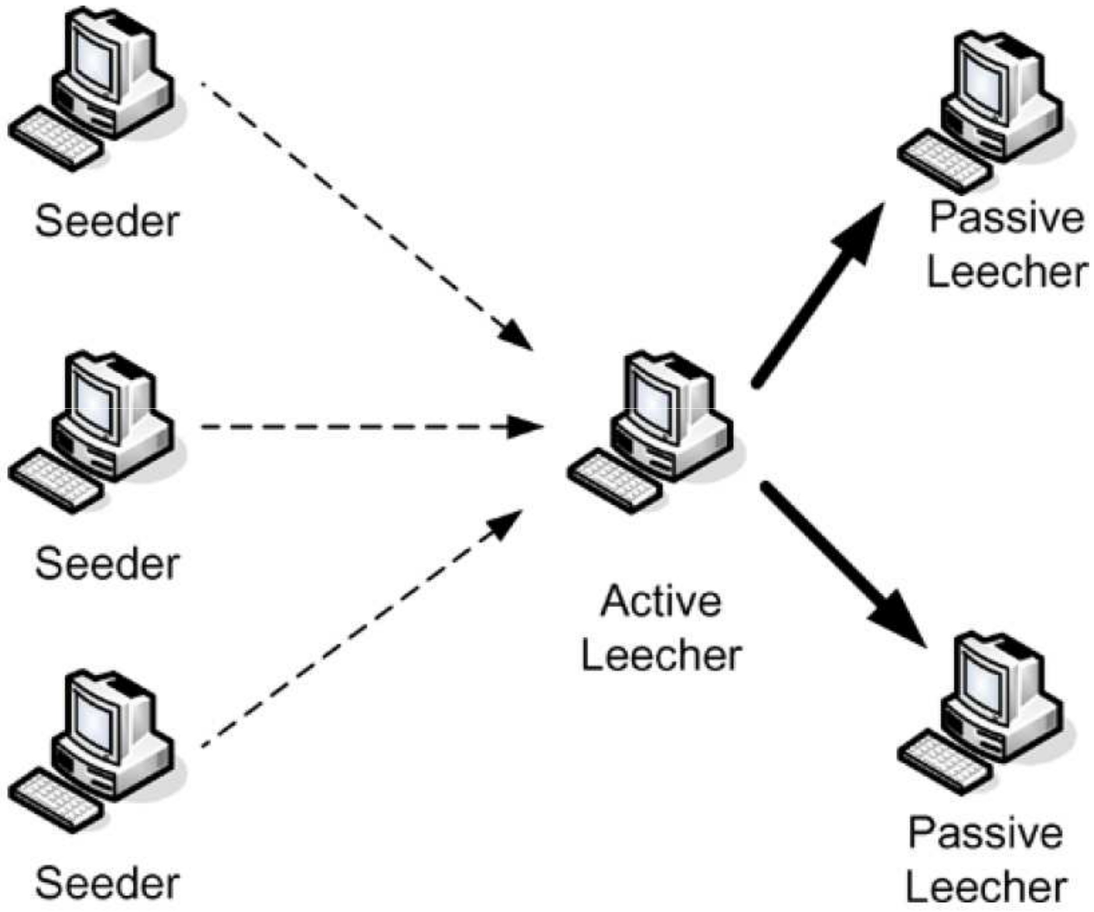
- Mainly targeted to *unpopular files*
  - File for which there is a limited number of available copies (i.e., seeders)
- Basic idea
  - Aggregate downloads on intermediate active leechers and hibernate the other leechers (passive leechers)
- Comparison with proxy-based BitTorrent
  - The download is delegated to another peer (active leecher)
  - The active peer is not a dedicated machine

Download of Unpopular Files



- Very few seeders
  - Each seeder serves one or more leechers
  - P2P degenerates to client-server
  - Low performance
  - It could be convenient to aggregate the upload capacity of seeders

Download Sharing



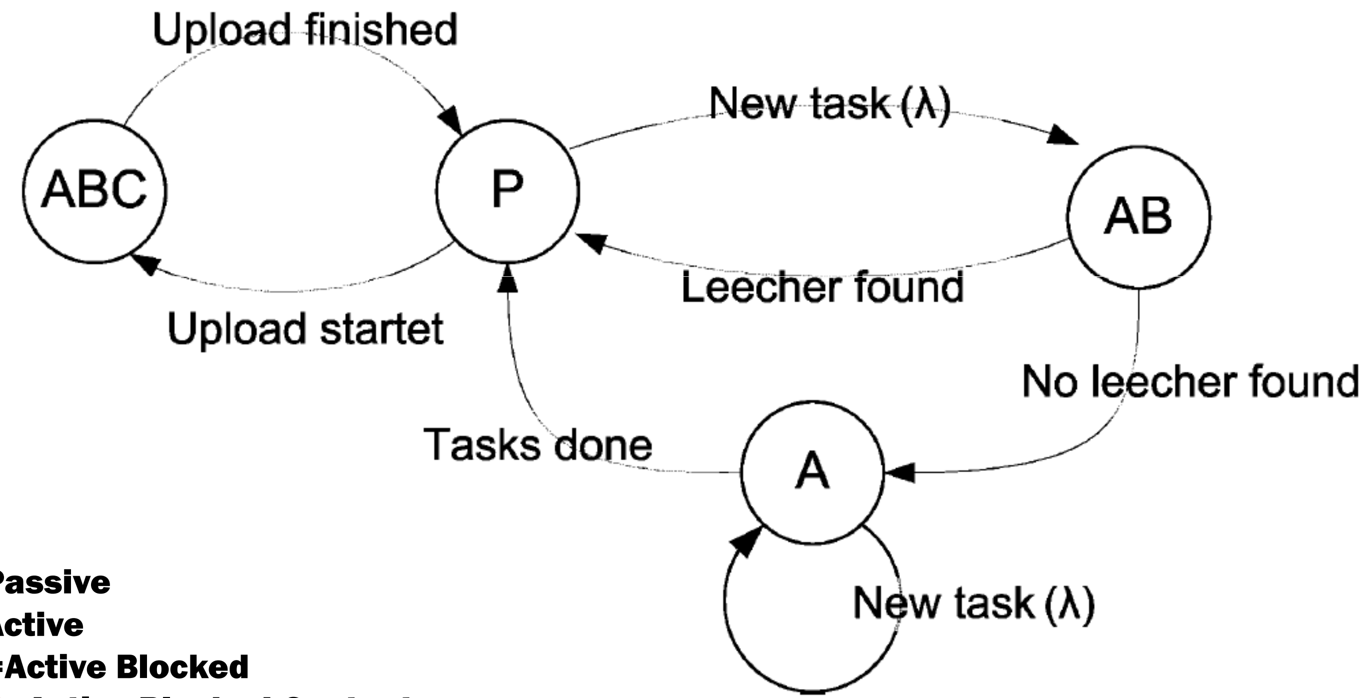
Download Sharing

- A random subset is selected to serve as active leechers
  - The leechers with the fastest link to seeders should be selected
  - All the other leechers are hibernated (passive leechers)
- Active leechers serve as proxies for passive leechers
  - Once the downloads are completed, files are transferred to passive leechers
  - It is assumed that active and passive leechers are close to each other (e.g., neighboring homes)

Dowload Sharing

PerLab

- State cycle of a leecher



P=Passive
A=Active
AB=Active Blocked
ABC=Active-Blocked-Content

Simulation Environment

PerLab

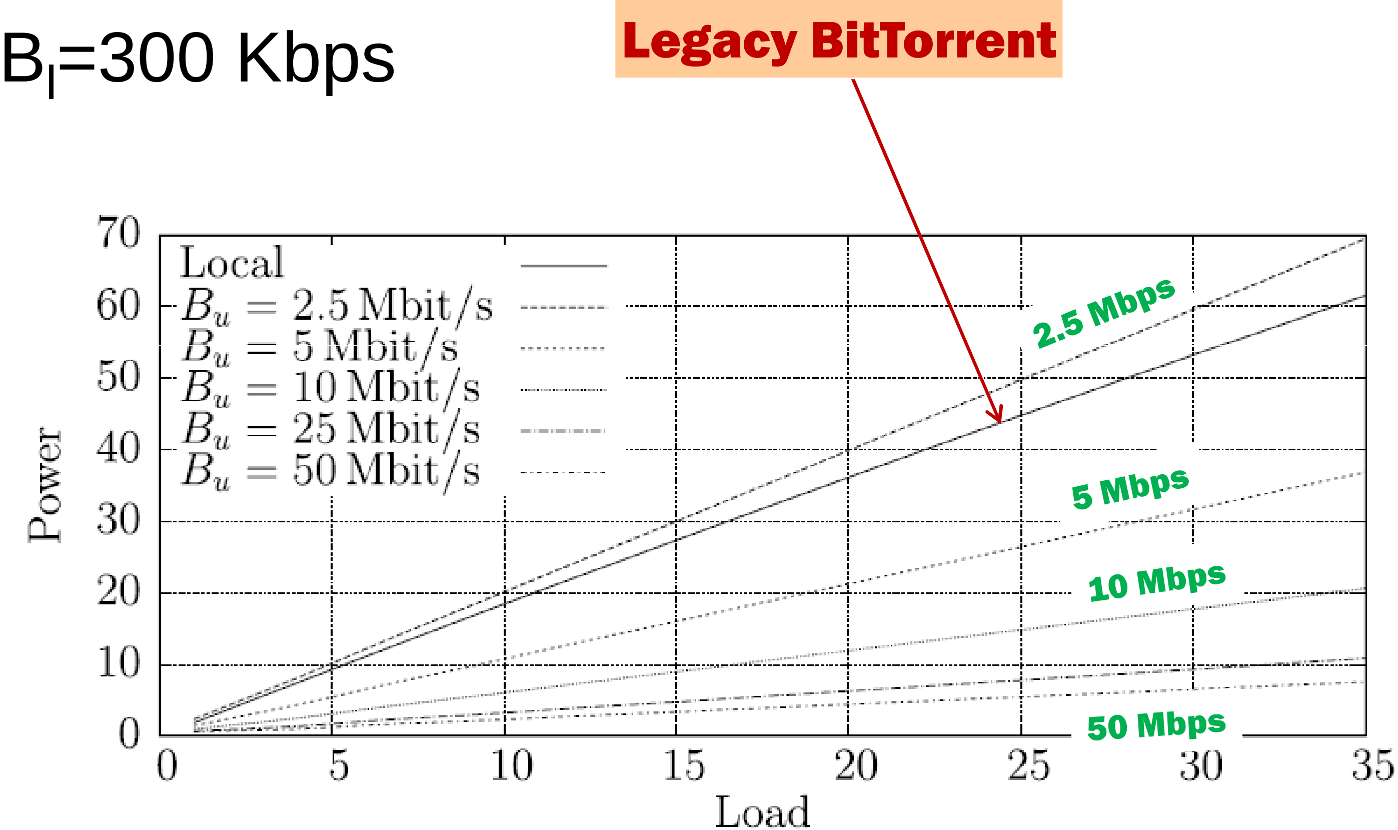
- N leechers → Aggregate Load
  - Load=avg. number of tasks per week per leecher
- B_d= Downlink Bandwidth
- B_u= Uplink Bandwidth
- B_l= Download rate achieved with legacy BitTorrent
  - Unpopular files
  - Assumed equal to 300 Kbps
- Service time = 8 hours
- Simulation time = 1 year

Download Sharing

PerLab

N=1000 leechers

B_l=300 Kbps



Remarks on Download Sharing

PerLab

- Targeted to unpopular files
  - For which B_l is typically low
- Uses a proxy-based approach
  - Active Leechers act as proxies for passive leechers
- Efficient only when $B_u >> B_l$
  - More suitable for future residential access networks
  - Currently $B_d > B_u$
    - ⇨ The transfer rate from the active leecher to the passive leecher is limited by the downlink bandwidth of the latter

Conclusions

Conclusions

PerLab

- Energy Efficiency in the Internet
  - At the user side
- Energy efficiency in P2P File Sharing
  - BitTorrent
- Energy Efficient BitTorrent
  - Green BitTorrent
  - Proxy-based BitTorrent
  - Download Sharing

Thank you for your attention!