Improving Download Time & Traffic via Locality in BitTorrent Protocol

Master Thesis Defense

Author: Antonis Prelorentzos (prelorentzos@aueb.gr)
Supervisor: George Xylomenos (xgeorge@aueb.gr)
Co-Evaluator: George Polyzos (polyzos@aueb.gr)

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
Msc in Computer Science

30/06/2010

The Problem

- When a peer inserts in swarm, requests a pool of peers from tracker to contact and download the content
- This process is driven by the tracker with **random** way and it achieves a high level of robustness…
- …**but** what happens with download times and network traffic(inter-domain & intra-domain)?
- Is there any way to select “better” peers?

Types of Locality

There are 3 main types of Locality criteria in BitTorrent:

1) Locality based on Overlay network processes
2) Locality based on Underlay network processes
3) Locality based on statistics computed and decisions made about pieces of the content (Piece Selection Strategy)

The Idea

- Insert locality use with the addition of AS information of each peer advertised to the tracker
- The tracker will process the obtained information of the connected to him peers and will response to each peer with a list of peers with a guaranteed percentage of peers with the same ASID to the peer that made the request to him

What is needed

- A search of all the connected to the tracker peers to learn the size of the same ASID peers to each peer makes a request
- A calculation of inserted percentage of the same ASID peers according to the **factor a** (%) before the random algorithm runs which completes the remaining pool of peers sent

max_same_asid_peers = a * peers_size

The algorithm

```
if ( same_asid_peers <= max_same_asid_peers )  //(Case 1)
{
    //Place all these peers to the list will be sent to the peer
}
else  //(Case 2)
{
    while (added_peers < max_same_asid_peers )
    {
            //Select random peer from  same_asid_peer  table
            //Place this random peer to the list will be sent to the peer
    }
}
//Complete the rest of the list with random peers from all the peers table
```

Examples

Peers in swarm	Max peers in reply	Factor a (%)	Same ASID peers	Max Same ASID peers
50	10	0	5	0 (default case)
50	10	10	6	1 (case 2)
50	10	35	5	3 (case 2)
50	10	40	3	3 (case 1)
50	10	50	7	5 (case 2)

Limitations

1) The proposal will have purpose to work only if:

- There are a lot of peers that will request the content
- We are talking about symmetric-size topologies (GT-ITM topology used)
- The peers from the same AS are not bad enough

2) Even if we have many users we must respect

Tit-for-Tat and set the factor a logical value and not making cliques

Simulation Process

- Use of Omnet Network Simulator with INET and Oversim Frameworks
- Use of the BitTorrent module implemented by our students
- Addition of:

1) The extra information is sent by each peer to the tracker *and*
2) Our algorithm to process the list to the tracker

Simulation Parameters - Scenarios

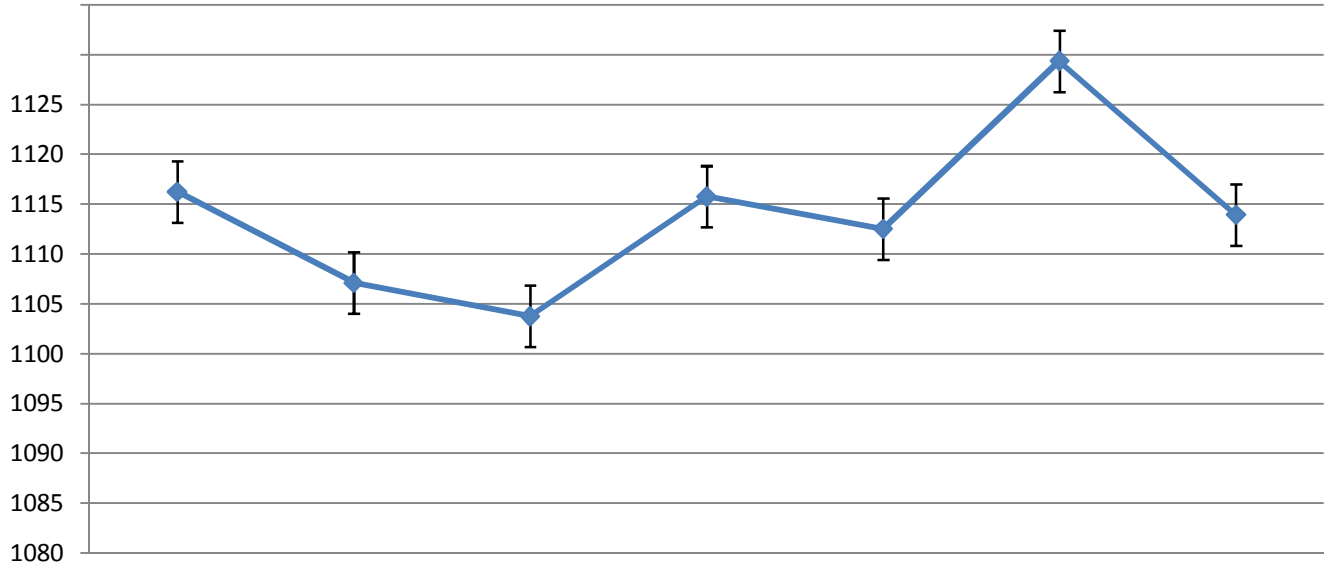
Simulation Parameters

- 50 terminals
- 150 MB file size
- 256 KB piece size
- 10 max peers in reply
- end game mode: true

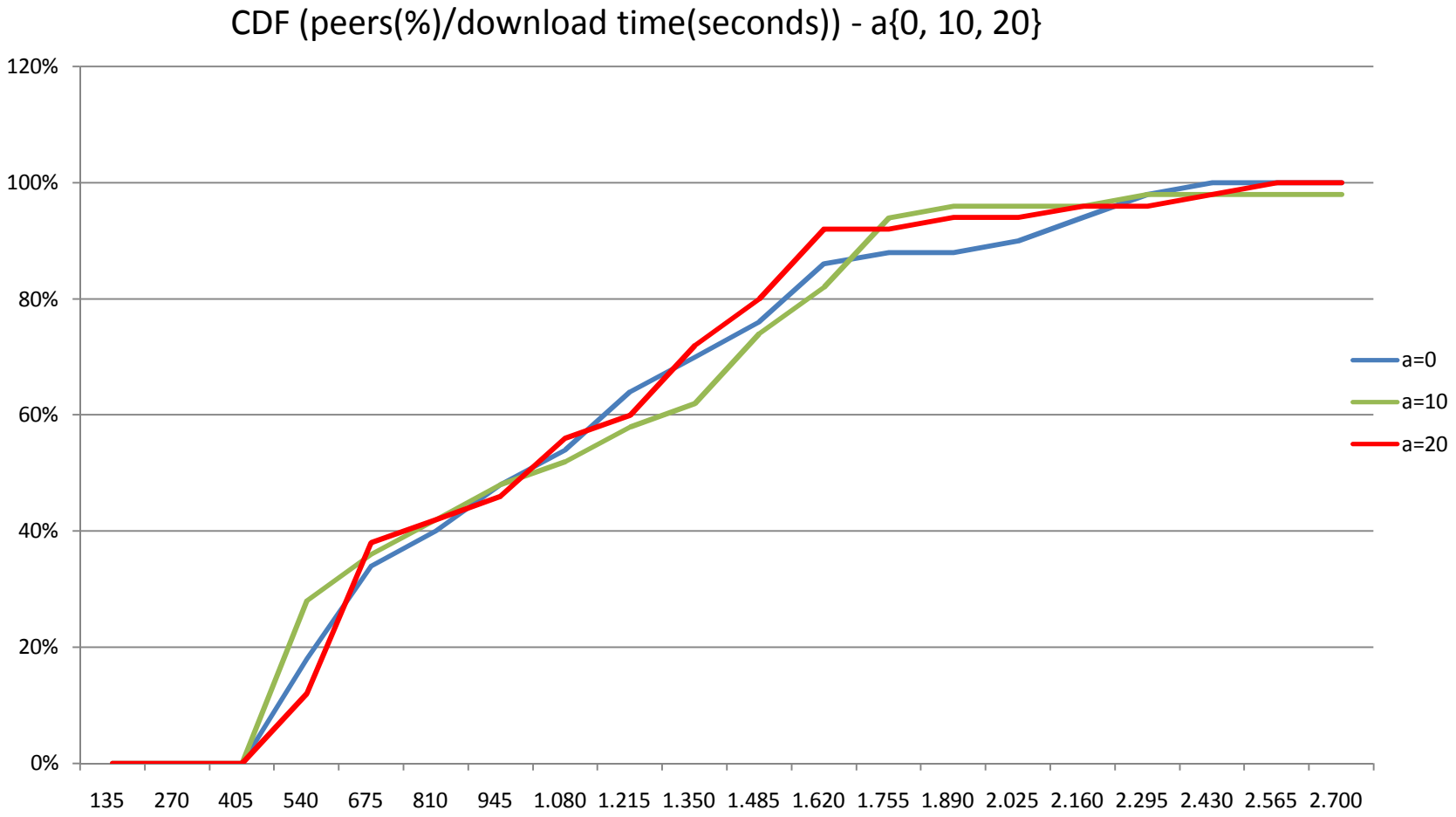
Simulation Scenarios (35 total)

- factor a for values {0, 10, 20, 30, 40, 50, 60}
- 5 different seed values

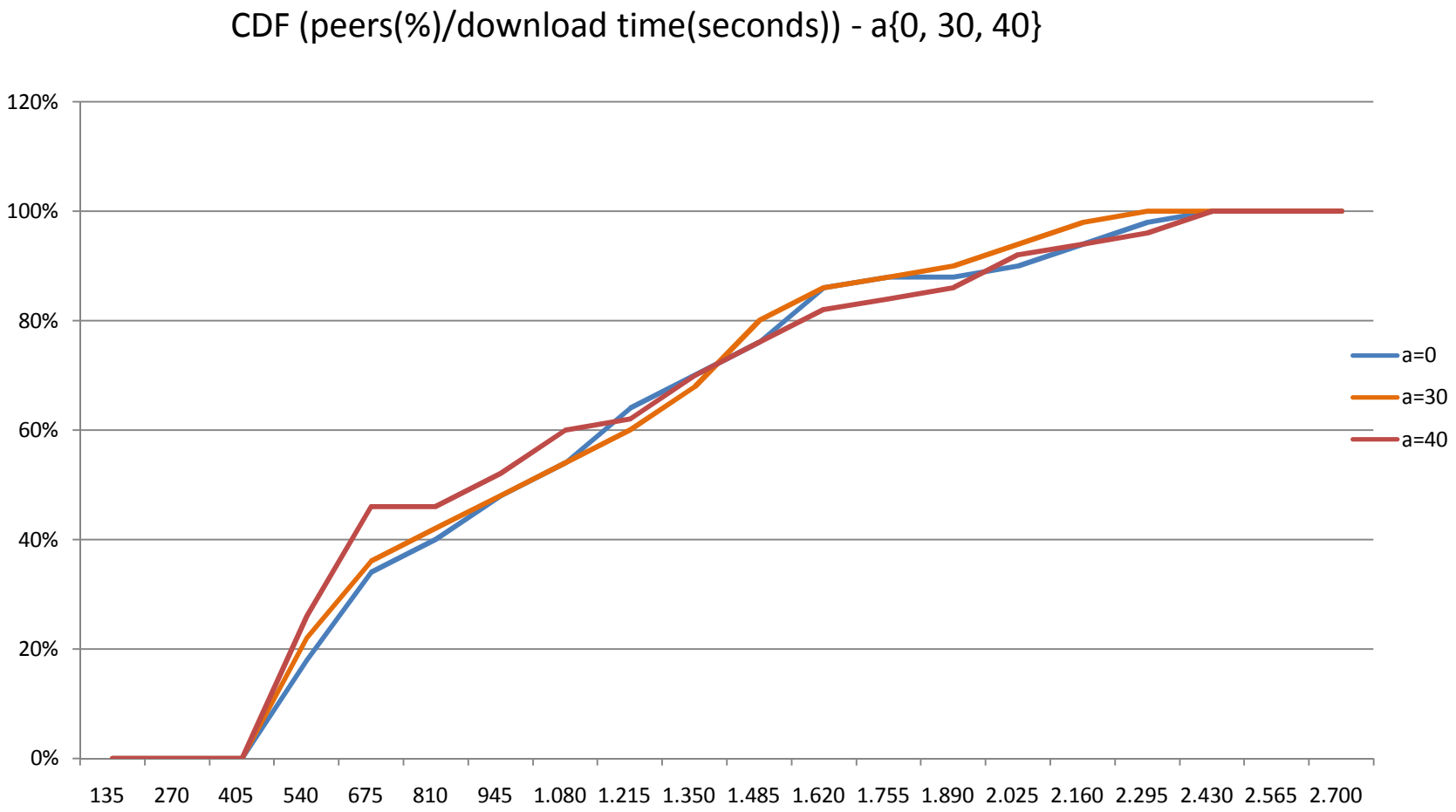
Results – Average Download Time



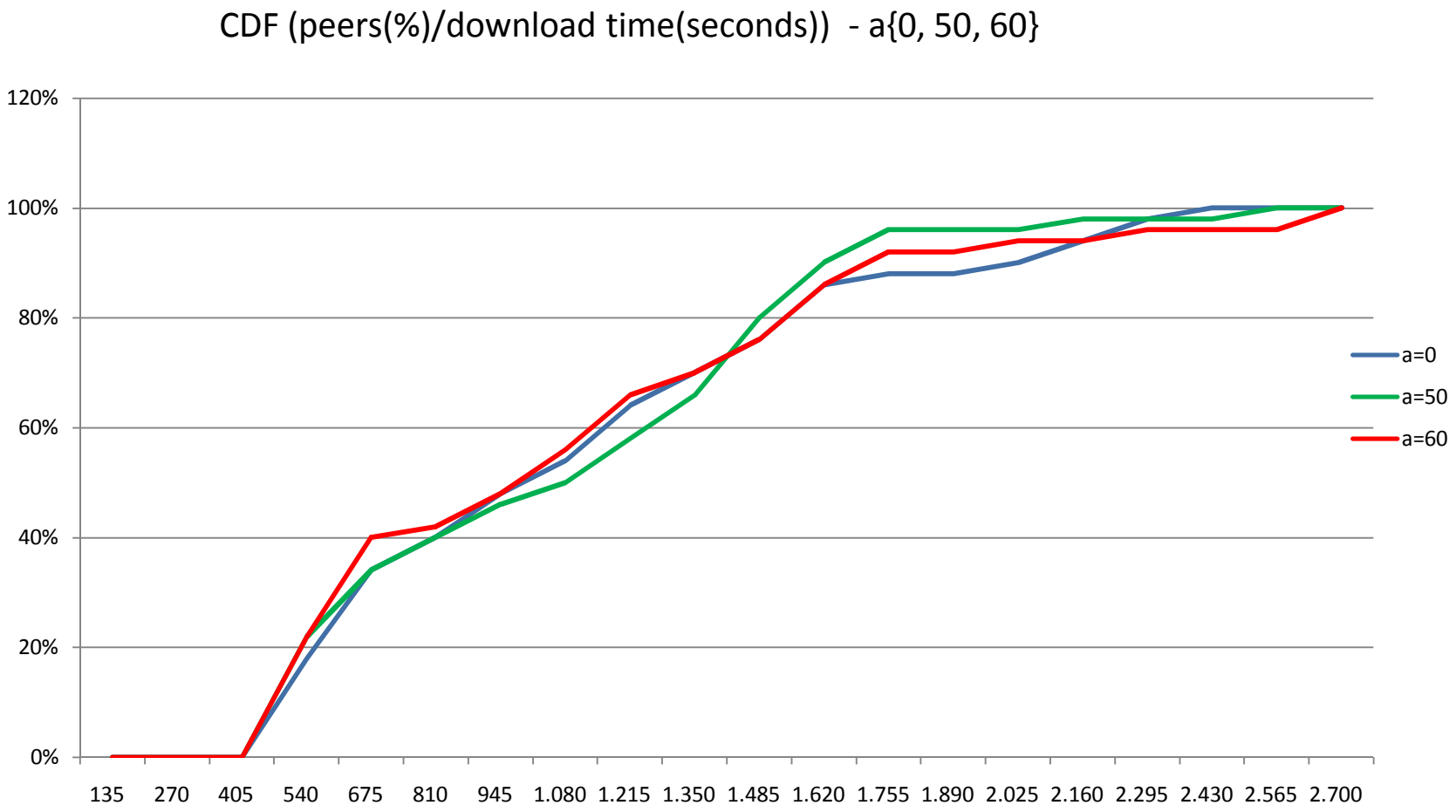
Results – CDF a {0, 10, 20}



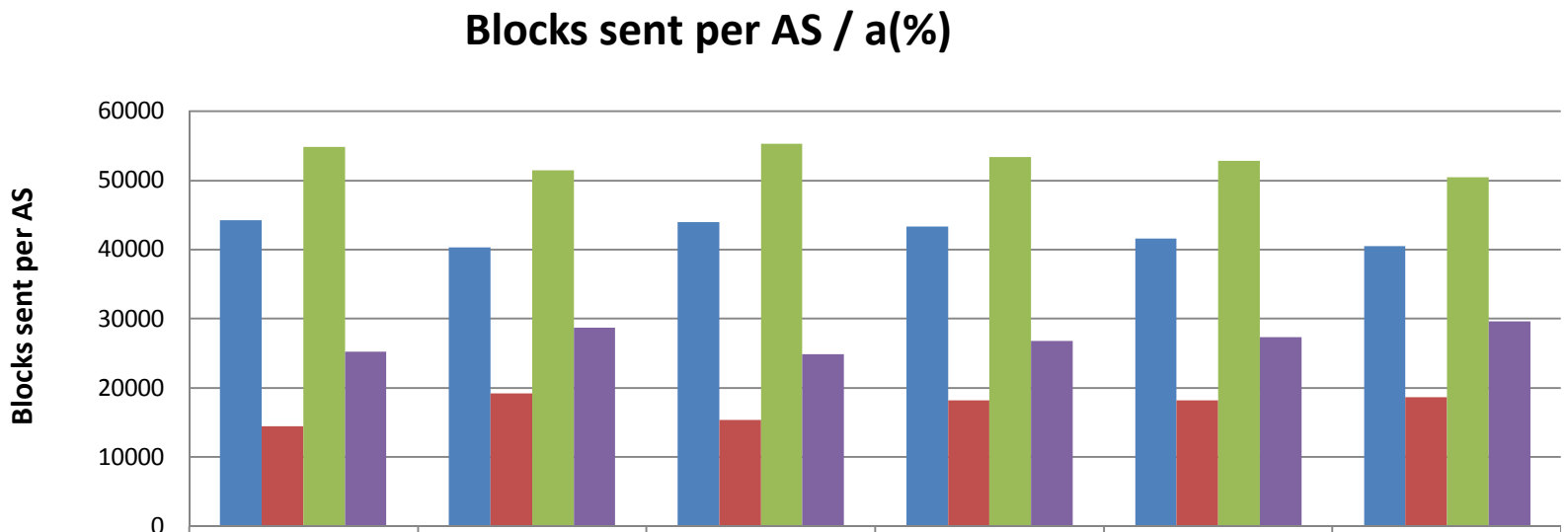
Results – CDF a {0, 30, 40}



Results – CDF a {0, 50, 60}



Results – Network Traffic



	a 1: 0	a 2: 10	a 3: 20	a 4: 30	a 5: 40	a 6: 50
Sent InterDomain	44208	40260,25	43899,75	43301	41554,75	40447,75
Sent IntraDomain	14460,25	19170,75	15372	18154	18204	18630
Received InterDomain	54809,25	51385,25	55222	53311	52776,75	50439,25
Received IntraDomain	25237,5	28687,25	24817,5	26753	27260,5	29606

Conclusions

- A factor between 10-20% gives a better download time than the completely random method
- The ingress domain traffic becomes bigger when the factor a is getting bigger
- A case of factor between 10-25% can give a good combinatorial result of these two parameters

Future Work

- Many types of improvement can be made to this “very hot” of Locality-aware area
- Maybe some tries for a better Piece Selection Strategy will give a good result for better performance of the protocol than Rarest First Strategy
- Collaboration of ISPs may has a big cost but if the traffic among them continues getting bigger and bigger will be necessary

Related Work - References

- Ioanna Papafili, George D. Stamoulis, Sergios Soursos, “Insertion of ISP-owned Peer & Locality Awareness in BitTorrent”,

  AUEB, (EuroNF workshop), Athens October 2008
- Ruben Cuevas, Nikolaos Laoutaris, Xiaoyuan Yang, Georgos Siganos, Pablo Rodriguez, “Deep Diving into BitTorrent Locality”, (Telefonica Research), July 2009
- Simon G. M. Koo , Karthik Kannan , C. S. George Lee, ”Neighbor-Selection Strategy in Peer-to-Peer Networks”, (Purdue University)
- Rong, L., Burnett, I. , ”BitTorrent in a dynamic resource adapting peer-to-peer network”, (Wollongong Univ., Australia), December 2005
- Marco Slot, Paolo Costa, Guillaume Pierre and Vivek Rai, “Zero-Day Reconciliation of BitTorrent Users with Their ISPs”, (VU University Amsterdam), August 2009

References

- K. Katsaros, V. P. Kemerlis, C. Stais and G. Xylomenos, "A BitTorrent Module for the OMNeT++ Simulator," Proc. 17th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), London, Great Britain, September 2009

- BitTorrent Development Community: BitTorrent Protocol Specification v1.0 (http://wiki.theory.org/BitTorrentSpecification)

- BitTorrent.Org: The BitTorrent Protocol Specification (http://www.bittorrent.org/beps/bep_0003.html)

- A.Varga OMNeT++ network simulator homepage (http://www.omnetpp.org)

Thank you for attending!

Special Thanks to:

- Mr G.Xylomenos for advising and guiding me
- C.Stais & K.Katsaros for helping, supporting and more, all this time up to now

Any Questions?

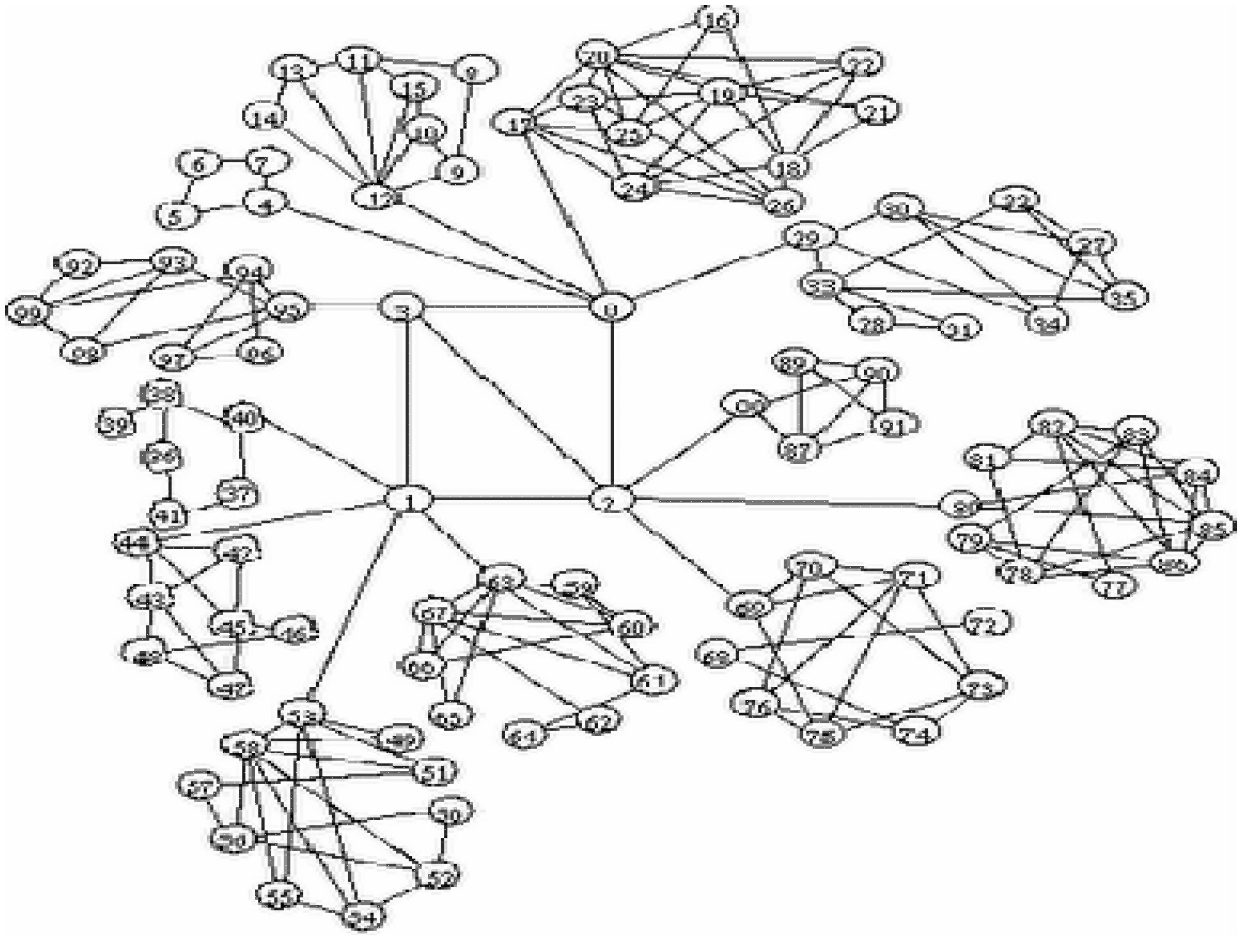
Backup Slides

ASID advertisement

- Advertisement of extra information of **ASIDs** between each peer and tracker

info_hash	**no_peer_id**
peer_id	**event (started, stopped, completed)**
port	**ip_address**
uploaded	**numwant**
downloaded	**key**
left	**tracker_id**
compact	**asid**

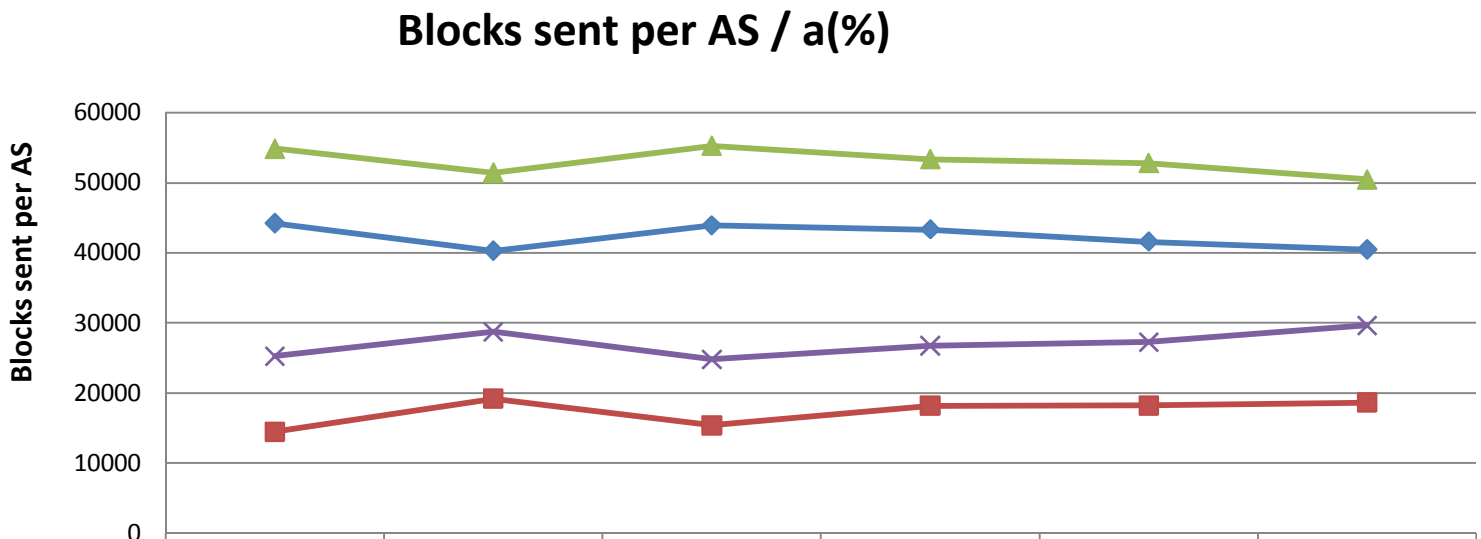
GT-ITM Topology example



Software used

- OMNeT++ 3.3
- INET for OverSim
- OverSim
- BRITE
- GT-ITM
- Ubuntu 8.04 Operating System
- Microsoft Excel 2007

Results – Network Traffic



	a 1: 0	a 2: 10	a 3: 20	a 4: 30	a 5: 40	a 6: 50
Sent InterDomain	44208	40260,25	43899,75	43301	41554,75	40447,75
Sent IntraDomain	14460,25	19170,75	15372	18154	18204	18630
Received InterDomain	54809,25	51385,25	55222	53311	52776,75	50439,25
Received IntraDomain	25237,5	28687,25	24817,5	26753	27260,5	29606