Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies

European Seventh Framework Project FP7-2008-ICT-216259-STREP

Deliverable D2.6
Comprehensive Test-bed and Trial Parameter Set Definition (Part II)

The SmoothIT Consortium

University of Zürich, UZH, Switzerland
DOCOMO Communications Laboratories Europe GmbH, DOCOMO, Germany
Technische Universität Darmstadt, TUD, Germany
Athens University of Economics and Business - Research Center, AUEB-RC, Greece
PrimeTel Limited, PrimeTel, Cyprus
Akademia Gorniczo-Hutnicza im. Stanislawa Staszica W Krakowie, AGH, Poland
Intracom S.A. Telecom Solutions, ICOM, Greece
Julius-Maximilians Universität Würzburg, UniWue, Germany
Telefónica Investigación y Desarollo, TID, Spain

For more information on this document or the SmoothIT project, please contact:

Prof. Dr. Burkhard Stiller
Universität Zürich, CSG@IFI
Binzmühlestrasse 14
CH—8050 Zürich
Switzerland

Phone: +41 44 635 4355
Fax: +41 44 635 6809
E-mail: info-smoothit@smoothit.org

Document Control

Title: Comprehensive Test-bed and Trial Parameter Set Definition (Part II)

Type: Public

Editor(s): Maximilian Michel

E-mail: michel@docomolab-euro.com

Author(s): Sergey Kuleshov, Simon Oechsner, Sergios Soursos, Rafal Stankiewicz. Maximilian Michel

Doc ID: D2.6-v1.0.doc

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.1	January 10, 2010	Maximilian Michel	First version, Merged chapter bullet list
V0.2	February 2, 2010	Maximilian Michel	Updated bullet list
V0.3	February 23, 2010	Maximilian Michel	Updated bullet list
V0.4	March 8, 2010	Sergey Kuleshov, Simon Oechsner, Sergios Soursos, Rafal Stankiewicz. Maximilian Michel	First merged version from all contributions
V0.5	March 22, 2010	Sergey Kuleshov, Simon Oechsner, Sergios Soursos, Rafal Stankiewicz. Maximilian Michel	Updates of "ETM parameters" and "Parameters", added "introduction", "executive summary" and "conclusions"
V0.6	March 26, 2010	Sergey Kuleshov, Simon Oechsner, Sergios Soursos, Rafal Stankiewicz. Maximilian Michel	Updates of "External Trial" section and "Evaluation parameters"
V0.7	April 1, 2010	Simon Oechsner, Sergey Kuleshov	Updates of "External Trial" section
V0.71	April 6, 2010	Konstantin Pussep	Internal review update I
V0.72	April 7, 2010	George Stamoulis	Internal review update II
V0.73	April 9, 2010	Dirk Staehle	Internal review update III
V0.8	April 12, 2010	Maximilian Michel	Integrated reviewers feedback and merged updates from partners for all sections
V1.0	April 14, 2010	Maximilian Michel	Final content and layout update

Table of Contents

1 Executive Summary 7
2 Introduction 8
3 Parameter Description 9
3.1 Swarm Parameters 9
3.1.1 Swarm Size Parameters 9
3.1.2 User Settings 10
3.1.3 Content Parameters 11
3.2 Client Input Parameters 11
3.3 ISP-related Parameters 13
3.3.1 Autonomous System Parameters 14
3.3.2 User Distribution and Characteristics 14
3.3.3 Technical Inter-domain Link Parameters 15
3.3.4 Business Inter-domain Link Parameters 15
3.3.5 Network Status Information 16
4 Updated Experimental Setup 16
4.1 Internal Trial 16
4.1.1 Simple Scenario for Internal Trial 16
4.1.2 Topology 16
4.2 Description of Simulation Scenarios 19
4.2.1 Complex Scenario 19
5 ETM Configuration Parameters 22
5.1 BGP-Loc ETM Mechanism 22
5.1.1 Client 22
5.1.2 SIS 23
5.2 IoP ETM Mechanism 23
5.3 HAP ETM Mechanism 24
5.3.1 SIS 24
5.3.2 NMS 25
5.4 Summary 25
6 External Trial Experimental Setup 27
6.1 Restrictions 30
6.1.1 Technical Limitation 30
6.1.2 Legal aspects 31
6.2 Initial and Support Seeding 31
6.2.1 Topology 33
6.2.2 G-Lab Setup 34
6.2.3 Timeline 35
6.3 Assessment of Results and Measurements 35
6.3.1 Measurements to Evaluate Win/No-Lose/Lose for ISP 35
6.3.2 Measurements to evaluate win/no-lose/lose for end-user 36
6.3.3 Measurements to Evaluate Win/No-Lose/Lose for Content Provider 37
6.3.4 Summary of Generic Measurements 37
6.4 Experiment 1 - Reference 38
6.4.1 Topology 38
6.4.2 Measurements 38
6.5 Experiment 2 – BGP-Loc and IoP 38
6.5.1 Topology 39
6.5.2 Experiment-specific Parameters Values 39
6.5.3 Experiment-specific Measurements 40
6.5.4 External effects 40
6.6 Experiment 3 – HAP and BGP-Loc 40
6.6.1 Topology 41
6.6.2 Experiment-specific Parameters values 41
6.6.3 Experiment-specific Measurements 42

7 Summary **43**

8 References **44**

9 Abbreviations **45**

10 Acknowledgements **46**

11 Appendix A - Simulation Support Studies for External Trial **47**

12 Appendix B - Adaptive Server Allocation for Peer-assisted Video-on-Demand52

13 Appendix C – Highly Active Peer and Next Generation Networks **61**

13.1 Scenario 61

13.2 ETM Modes 63

13.3 Increased Availability of Content & Incentives 63

13.4 HAP Algorithm 64

13.4.1 General view 64

13.4.2 Peer Rating 65

13.4.3 Dimensioning of HAPs 66

13.4.4 Collecting Behavior Statistics 66

13.4.5 HAP Promotion 67

(This page is left blank intentionally.)

1 Executive Summary

The project SmoothIT aims at defining, developing, and testing Economic Traffic Management (ETM) mechanisms to optimize the traffic impact of overlay applications on underlay networks in such a way that all network operators, overlay providers, and application users benefit from the approach undertaken; this corresponds to the so-called TripleWin situation. The ETM mechanisms are to be tested and demonstrated via a range of simulations, internal and, finally, external trials. To this end, WP2 “Theory and Modeling” investigates theoretical foundations and prepares theoretical grounds for running the trials and associated experiments.

This deliverable is the second part of the work defining the simulation, test-bed, and trial parameters and settings. It is building upon work done at the earlier stages of the project in terms of identifying system requirements, ETM models and setup of the trials, particularly the internal trial which have already been defined in [D2.5], executed and reported in [D2.4]. As the first part [D2.5] concentrated on the overall simulation and test-bed settings, this deliverable D2.6 is concerned with the configuration, execution, and pre-evaluation of the external trial that will mainly be held in the network of Primetel in Cyprus This, together with the fact that certain settings were found to be changed due to new discoveries made in simulations or in the internal trial, necessitated updating certain parts of that document.. Thus, they have been updated in this document D2.6 here.

The main areas covered in the present document are as follows:

- Update of parameters presented in the first part of the deliverable, including parameters regarding the BitTorrent/Tribler system as a whole, the client software, and the ISP network. Additionally, update of the list of experiments for simulations and internal trials.

- Complete list of parameters regarding the configuration of the ETM mechanisms developed throughout the project and selected for evaluation in the external trial, and list of parameters that will be used for the evaluation of the impact of these different ETM mechanisms on the system’s performance.

- Definitions of different scenarios that will be tested in the external trial, including specific configuration parameter values for each specific test scenario and also scenario specific evaluation parameters.

Specifically, this deliverable sets the external trial in following way: Selected ETM mechanisms, namely the BGP-Locality (BGP-Loc), ISP-owned Peer (IoP), and Highly Active Peer (HAP) together with a reference setting without ETM mechanisms will be tested in a sequential way. The influence on the traffic patterns of the ISP’s users will be measured together with the impact on the user-experienced quality of service. Due to the tight constraints for the quality of service level for the ISP’s users in this commercial environment, a mechanism called Initial Support Seeding will be used, to prevent the occurrence of a non-acceptable quality level.

The final external trial evaluation will be presented in another deliverable, namely D4.2.

2 Introduction

The main target of this document is to support SmoothIT's external trials by first identifying a complete set of relevant parameters, and second by specifying values or value ranges for these parameters. This document is the basis for the work to be conducted in WP4, the final results and assessments of the external trial will be reported in D4.2.
The ETM mechanisms to be tested in the external trial were selected and described by WP2 in deliverables D2.2 [D2.2] and D2.3 [D2.3]. The application to be used in the external trial is Video-Streaming, as described in deliverable D1.2 [D1.2]. The respective application is Tribler/NextShare [NextShare], a BitTorrent variant, for Live Streaming/VoD. The results of this deliverable will be used in WP4, for the setup and implementation of the external trial. The results of the external trails will be found in deliverable D4.2.

This document is the second and final part of two deliverables concerned with the specification of parameters for the internal and external trial. The first deliverable [D2.5] was mainly targeting the internal trials and simulation studies. Hence, this deliverable concentrates on the parameters for setting up, executing and evaluating the external trials. The parameters are described and as much as possible fixed to concrete values before the final simulations and the external trial. Additionally, updates on certain parts of the previous deliverable are given, where necessary. In addition, new simulation scenarios and a new scenario for the internal trial are described.

This document is organized in the following way: Section 3 expands D2.5; it describes the updated parameters for the system, the client software and the ISP network. The following section provides another update of D2.5 in terms of scenarios for internal trial and simulations. Section 5 provides a list of the parameters needed for the configuration of the three ETM mechanisms already selected and specified by the project that will be evaluated in the external trial. This list of parameters was not shown previously in D2.5 and is introduced in this document. Finally, Section 6 describes in detail the three different experiments that will be used in the external trials. General settings common to all scenarios, such as the Primetel network topology are shown, together with restrictions that are associated with the external trials. This document closes in Section 7 with a summary and certain concluding remarks. In the appendix results from simulation studies can be found, that were conducted in order to support correct parameter settings for the external trial.

3 Parameter Description

This section describes all parameters, which are relevant to trial and simulation scenarios and proper evaluation of these results with the target of identifying if TripleWin has been achieved. Some of these parameters can be influenced, and thus are free variables, while some others can only be observed, but should still be recorded for valid evaluation results.

These parameters are divided into three categories: those that describe the properties of the given swarm, those related to user settings or behavior and, finally, parameters related to an ISP (Internet Service Provider), such as topology, network properties and so on. The swarm parameters are further labeled as being generic, i.e. applicable to any torrent swarm or VoD-specific i.e. relevant to video-on-demand streaming with Tribler only.

In addition to D2.5 [D2.5], we specify for each parameter in which experiments the according value can be influenced. While this can generally be done in simulations and the internal trial, there are many values that cannot be set for the PrimeTel users in the external trial. Details for the according settings of G-Lab peers can be found in Section 6.

We also provide default values to some typical parameters that are used in case no other values are specified in the according experiment.

3.1 Swarm Parameters

The overlay network is defined by Tribler, which implements a variation of the BitTorrent protocol (Tribler is the original name of the software; the version which is used in the SmoothIT trial is NextShare [NextShare], a successor to Tribler being built within the EU-project P2P-NEXT [P2PNEXT]). Tribler modifies BitTorrent policies in order to be more streaming-friendly, but full compatibility is maintained. The following parameters need to be set or observed for every swarm during the trials and simulations to run the tests and interpret the results.

Swarm parameters are categorized into three groups: content related parameters, user setting related parameters, and parameters describing the swarm size. For the parameters that have a default value which is unlikely to be changed in the experiments, this value is given as well.

3.1.1 Swarm Size Parameters

Number of peers (Generic)

This parameter defines the total number of peers that are actively or passively belonging to a swarm. A peer belongs to a swarm when the swarm's torrent file is present in its torrent list. A peer is actively participating in the swarm when the client is running and passively when the client is switched off. This value can only be set in simulations and internal trial.

Peer availability (Generic)

Peers may fail or leave a swarm at any time when they switch the client on and off. The probability that a peer is available at any time is given by the peer availability. The peer availability is the share of active peers in the swarm. This value can only be set in simulations and internal trial.

Initial number of seeds (Generic)

This parameter defines the initial number of seeds, i.e. peers having the entire content, which actively participate in the swarm at the beginning of the experiment. This value can be set in all experiments.

Initial number of leechers (Generic)

This parameter defines the number of leechers, i.e. peers not having all content, which actively participate in the swarm at the beginning of the experiment. This value can be set in all experiments.

Initial leecher content share (Generic)

This parameter defines the random or in order share of the content that a given leecher has at the beginning of the experiment. By default, new leechers have no content. This value can be set in all experiments.

Total seed capacity (Generic)

The total seed capacity expresses the total upload capacity provided by seeds or servers that already possess the content at the beginning of the experiment and offer it continuously. This value can be set in all experiments.

Arrival process (Generic)

The arrival process describes how peers, in particular leechers, join the swarm. Typical types of arrival processes are flash crowd or Poisson arrivals. This value can only be set in simulations and internal trial, in the external trial this is a result of user behavior.

In order to get an impression of the arrival process in the external trial, the number of peers in the swarm can be measured at fixed time intervals; based on these measurements a model of peers arrival process can be deduced.

3.1.2 User Settings

Share of free-riders (Generic)

This parameter defines the share of peers that do not provide bandwidth to the system. In the context of BitTorrent like protocols a free-rider uses a modified BitTorrent client that either does not upload at all and profits from optimistic unchoking or actively tries to minimize the ratio of upload and download bandwidth. This value can only be set in simulations and internal trial.

Share of firewalled/NAT peers (Generic)

This parameter defines the share of peers that cannot accept incoming connections. It can only be set in simulations and internal trial.

Download bandwidth (Generic)

This parameter defines the maximum bandwidth peers reserve for downloading. This bandwidth is set by the user and has to be smaller than the downstream access speed. This value can only be set in simulations and internal trial.

Upload bandwidth (Generic)

This value defines the maximum bandwidth peers reserve to forward video streams to other peers. This bandwidth is set by the user and has to be smaller than the upstream access speed. This value can only be set in simulations and internal trial.

3.1.3 Content Parameters

Video and audio codec (VoD)

This parameter defines the codec used to encode the shared file. The choice of codec not only affects the compression rate and quality but also how sensitive Quality of Experience (QoE) would be to delayed blocks. Some examples of video/audio codecs are: x264 [FH264], a free video encoder that implements the H.264 codec [H264], offering excellent image quality at low bitrates. It is, however, CPU-intensive (Central Processing Unit). For audio, the most modern codec is AAC [AAC]. This value can be set in all experiments.

Bit rate of video files (VoD)

The bit rate of a file and its codec influence the final video quality. With x264, it is possible to achieve DVD-like (Digital Versatile Disk) quality with 2000 Kbit/s, and average quality with 500 Kbit/s. AAC (Advanced Audio Coding) audio can achieve excellent quality at 128 Kbit/s, and average quality at 64 Kbit/s. This value can be set in all experiments.

Size/length of video files (Generic/VoD)

The file size is the average bit rate multiplied by the length of the video (in seconds). This value can be set in all experiments.

Popularity (Generic)

This parameter defines the distribution of torrent popularity, determining the request rate for each file. This value can only be set in simulations and internal trial. In the external trial this parameter can be influenced by the choice of the content offered to the users.

Chunk/Block size (Generic)

A file/video is subdivided into chunks that are again subdivided into blocks. The chunk and block sizes are specified in the torrent file and may hence be set individually for every shared file. The actual block and chunk sizes often depend on the volume of the shared content. In the context of Tribler the terms piece and sub-piece are alternatively used. This value can be set in all experiments.

3.2 Client Input Parameters

Input client parameters are typically fixed for a certain client and may not be changed by the user. They can have a large impact on how the overlay is formed, and on the potential influence of ETM mechanisms.

Upload/download slots

These parameters show the number of peers that the client will attempt to connect to simultaneously in order to upload/download content. By default, the number of upload slots is limited to 4. The number of download slots is unlimited. This value can be set in all experiments.

Seeding behavior

The seeding behavior describes how a user behaves after downloading all content, i.e. when she changes from leecher to seeder. The time a user continues participating in the swarm as a seeder may be specified by either the seeding ratio or the seeding time. Both alternatives are described below.

In the special case of VoD, the seeding behavior may also start after a user has finished watching the video although the downloading process has already finished some time before. This value can only be set in simulations and internal trial.

Seeding ratio

This is the upload/download ratio that the client will achieve before it automatically stops seeding (offering) this content to the overlay. It is defined as $SR = \frac{data\ uploaded}{data\ downloaded}$. This value can be set in all experiments. In the external trial, it only has an effect as long as the user does not switch off the client earlier.

Seeding time

This is the time during which the client will seed (offer) a file to the overlay, after the completion of the download or finishing watching the video. Once this time expires, the client will automatically stop seeding this content. It is defined as

$ST = SeedStopTime - DownloadCompletionTime$.

This value can be set in all experiments, in the external trial: as long as the user does not switch off the client earlier.

Neighbors min

When a client has reached this number of connected clients it will stop initiating new connections. By default, this value is set to 40 connections. It can be set in all experiments.

Neighbors max

When a client has reached this number of neighbors it will reject any further remotely initiated connection requests. By default, this value is 80. It can be set in all experiments.

Initial seeding

This is a Boolean parameter. If it is set to true, then this client will perform “super seeding” once it becomes a seeder. This means that it will offer the rarest parts of the file first to the overlay, in order to reduce the data the original seed needs to upload to boot-strap this overlay. This algorithm is *experimental*. For more information cf. [BT1]. This value can be set in all experiments.

End game mode

This is a Boolean parameter. If it is set to true, then this client will enter “end game” mode at the end of the download, in order to receive the final parts (which are slower) more quickly. This mode is experimental has more parameters that are implementation-dependent. For more information cf. [BT2]. This value can be set in all experiments.

Timeout values

The timeout parameter specifies the time reserved for the attempt to open a new connection. If a connection is not established before the timeout, then it is presumed to have failed and is aborted. This parameter is defined as:

$$Timeout = ConnectionInitiationAbortTime - ConnectionInitiationTime$$

This value can be set in all experiments.

Tit-for-tat parameters

Overlay clients (such as BitTorrent clients) may play tit-for-tat games in order to set their upload/download speed limits. The upload rate of a client is correlated to the download rate of that client; if a client limits its upload rate then the other peers will limit their upload rate to this client. These parameters are usually implementation-dependent. In Tribler, the give-to-get variant is used, cf. [MPME08]. This value can be set in all experiments.

Chunk/peer selection strategy

These parameters store the strategy used for chunk selection (e.g. Local rarest first/Sequential/Priority set based) and peer selection (e.g., Random/Locality aware). Both of these parameters may include strategies to use SIS (SmoothIT Information Service) data for chunk and/or peer selection. In Tribler, the default chunk selection is Priority set based. This value can be set in all experiments.

Video streaming parameters

These parameters control the video streaming and include the size of the buffer, the partitioning of the buffer in priorities (e.g. the first 10s are high priority, the next 40s medium priority and the rest are low priority), playout buffer size, etc. These parameters are mostly implementation-dependent. Tribler uses 10s for the size of the high priority set, and 40s for the size of the mid-priority set by default. This value can be set in all experiments.

3.3 ISP-related Parameters

This section provides an overview of all parameters that are related to the ISP, the ISP's underlying network, as well as the discussion about the values of these parameters in the internal and external trial.

The most important variables for the trial specification regarding the underlying network capabilities can be classified in the following types:

- Network topology
  - o Intra-domain topology
  - o Inter-domain information
- Business related information
- Users distribution and characteristics.
- Network status

These are explained in the sequel.

3.3.1 Autonomous System Parameters

Hierarchy level

This parameter defines the level of the AS (Autonomous System) according to the Internet hierarchy: Tier-1, Tier-2 and Tier-3. This value can be set/is fixed in all experiments, i.e. in internal trial or simulations this parameter can be set, while in the external trial this parameter is given by the actual network topology.

Number of PoPs (Points of Presence)

This parameter defines the number of PoPs within an AS. This value can be set/is fixed in all experiments.

Intra-domain core delay

This parameter defines the propagation delay in the IP core of the AS, i.e. between two PoPs. The intra-domain domain core delay does not include queuing delays caused by congestion. Serialization delay is considered to be insignificantly small and is therefore ignored. This value can only be set in simulations and internal trial.

Bandwidth of the core network

The bandwidth in the core network corresponds to the smallest link bandwidth in the core network. Typically, this bandwidth should be much larger than the access bandwidth. In general, it is assumed that the P2P end-to-end bandwidth is not limited by the core bandwidth. This value can only be set in simulations and internal trial.

Access type composition

An AS typically comprises access networks with different access types, e.g. different DSL access speeds or optical access networks. An access type is characterized by its

- upstream bandwidth
- downstream bandwidth
- access delay

This value can only be set in simulations and internal trial.

3.3.2 User Distribution and Characteristics

Number of users

This parameter specifies the number of users in the experiment. It might be either a constant or a randomized number specified by its mean value. Users may belong to several swarms. This value can only be set in simulations and internal trial.

Number of swarms

This parameter defines the Number of swarms considered in the experiment simultaneously. This value can be set in all experiments.

Distribution of users across the ASes

This parameter specifies how the users are spread over the different ASes present in the network topology. This distribution might be homogeneous, i.e., on average the number of users is the same in all ASes, or skewed, like in real swarms. This value can only be set in

simulations and internal trial, while it can be influenced to a certain extent in the external trial.

Distribution of users within an AS

This parameter specifies how the users within one AS are spread over the different PoPs. This might again be homogeneous or skewed. This value can only be set in simulations and internal trial.

Distribution of users per access type

This parameter specifies the percentage of users per access type present in an AS. This value can only be set in simulations and internal trial.

Distribution of users per swarm

This parameter specifies the percentage of users per swarm in experiments with multiple swarms. In general, we assume that the distribution of users per swarm is independent of the AS. This value depicts the relative popularity of different files. It can only be set in simulations and internal trial.

Number of seeds per domain and its associated access type

This parameter specifies how many seeds there are in an AS and the access type of the seeds/servers. This value can be set in all experiments.

Percentage of SIS usage

This parameter specifies the percentage of peers in the AS that use the SIS service. If several ETM mechanisms are used in parallel different groups of SIS users might be distinguished. This value can be set in all experiments.

3.3.3 Technical Inter-domain Link Parameters

Delay

This parameter specifies the packet delay between the two PoPs connected by the inter-domain link. This value can only be set in simulations and internal trial.

Bandwidth

This parameter specifies the bandwidth of the inter-domain link. This value can only be set in simulations and internal trial.

3.3.4 Business Inter-domain Link Parameters

Type of interconnection agreement

This parameter specifies the interconnection agreements, e.g. peering or transit agreements, for more details cf. [D1.1]. This value can only be set in simulations and internal trial, while it is fixed for the external trial.

Cost model associated to inter-domain transit link usage

Different models are in place in theory and practice on how to charge inter-domain link usage with transit agreement. Examples are 95th Percentile, per volume, etc. For the external trial we are going to consider 95th Percentile model and the amount of inter-domain traffic. For more details, cf. [D1.1].

3.3.5 Network Status Information

Status of intra- and inter-domain links

This parameter specifies the utilization of intra- and inter-domain links. It is typically given in a granularity of several minutes. This value can only be set in simulations and internal trial.

4 Updated Experimental Setup

In what follows, updates for the simulation scenarios and the internal trial setup are given in order to complete the set of parameters for internal trials and simulations and attain a complete documentation thereof. This means that these are changes and additions to respective settings as presented in the first part of this document ([D2.5]).

4.1 Internal Trial

In addition to the description of the internal trial in [D2.5], another experiment has been performed in the internal testbed. The results will appear in deliverable 4.1. The setup and the parameters are shown below.

This is an update to [D2.5], specifically to Section 7.1, describing a scenario for the internal trial.

4.1.1 Simple Scenario for Internal Trial

This scenario is a simple setup to demonstrate in an easily comprehensible way the merits of SIS-enabled Locality (BGP-Loc) and ISP-owned Peer (IoP) mechanisms. It is used in the SmoothIT showcase to show in a reasonable time with only two ISPs' networks a quickly observable reduction of the inter-domain traffic or the impact of the IoP.

A preliminary description of results from this scenario can be found in the Annex C of [D3.3]. Further results will be included in deliverable 4.1.

4.1.2 Topology

Similarly to the external trial setup described later on, in this scenario there are two ISPs, where peers can be located (see Figure 1). ETM mechanisms, in this case IoP and BGP-Loc, are only applied in ISP A, while ISP B does not employ any ETM mechanisms.

As shown in Figure 2, this internal trial experiment is based on the ModelNet network emulator (described in [D3.1]). The setup uses two computers to emulate the network of the previously described two ISPs, and to execute the tracker, the SIS server, the IoP, and the peers.

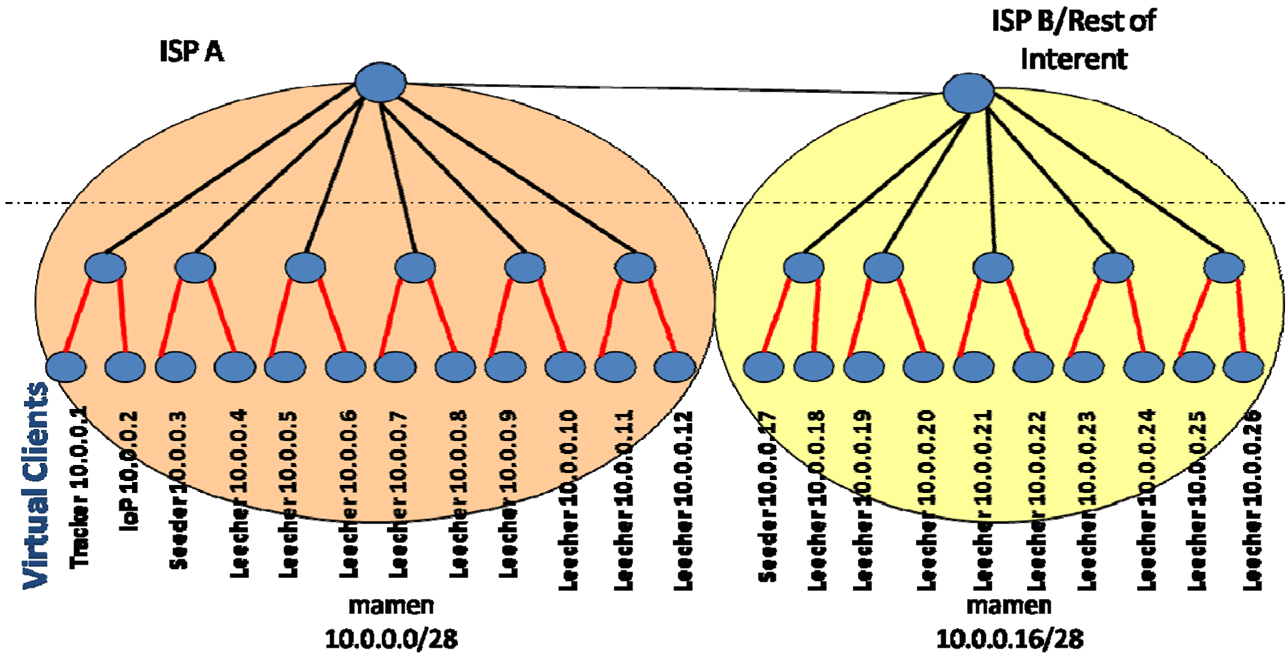


Figure 1: Topology of the simple scenario for the internal trial

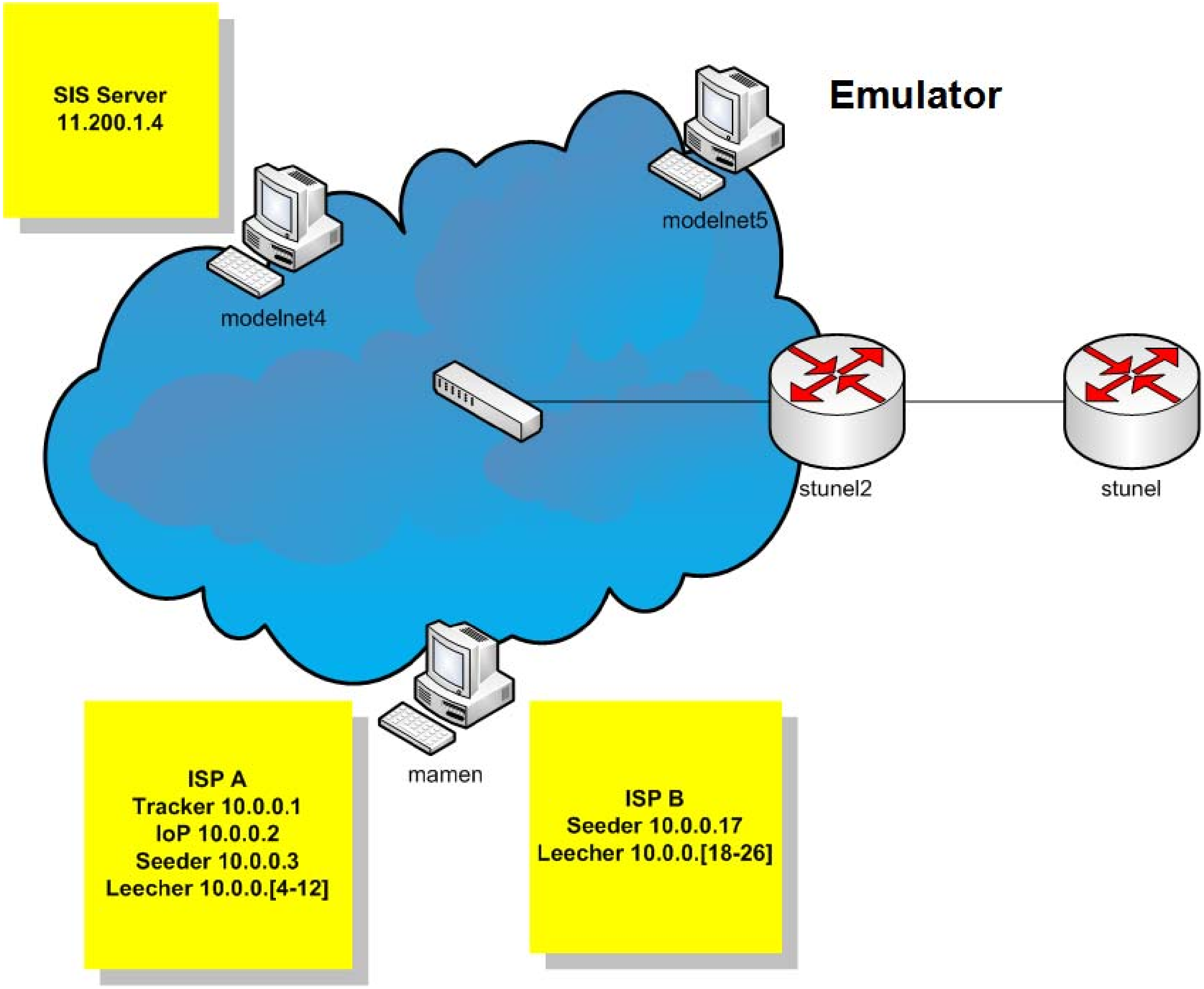


Figure 2: Emulator setup for the simple scenario for the internal trial

4.1.2.1 General Parameters

The experiment emulates the network and 20 peers running in two different sub networks (ISP A, ISP B). The application the peers are running is Tribler; a video file is distributed using this application. The seed ration in this scenario is 0.1 (see Table 1).

Table 1: General parameters for simple scenario

	Mean Swarm Size	**Overlay**	**Content**	**Seed percentage**
General Parameter	20	Tribler	10 MB	10%

4.1.2.2 AS Definition

The simple internal trial scenario consists of two end-customer ISPs. Details can be found below in Table 2. A description of the terminology of the parameters used in Table 2 can be found in [D2.5], Section 6, which explains the introduced scenario description template.

Table 2: ASes definition for simple internal trial scenario

	End-Customer ISP A	End-Customer ISP B
Quantity	1	1
Tier	3	3
Relative Local Swarm Size	1	1
Relative Seed Share	0	0
Swarm/Peer behavior	Steady Swarm/ Leecher	Steady Swarm/ Leecher
Category	High Mixed Quality	High Mixed Quality
SIS Usage Percentage	100%	0%

4.1.2.3 Exemplary Experimental Setup

4.1.2.3.1 SIS gains

This experiment offers insights on the effectiveness of SIS usage for the different ISPs in a scenario with two ISPs, whereas only one ISP applies ETM mechanisms. (cf. Table 3).

Table 3: SIS usage for the simple internal trial scenario

	End-Customer ISP A
SIS Usage Percentage	0% - 100%

4.2 Description of Simulation Scenarios

4.2.1 Complex Scenario

This scenario should depict the effects of SIS-enabled locality in the case of multiple ASes in different distances from each other in terms of number of hops. Additionally, the effects of peering ASes can be investigated.

4.2.1.1 Topology

The topology shown in Figure 3 offers a network structure with a variation in inter-AS hops between different ISPs.

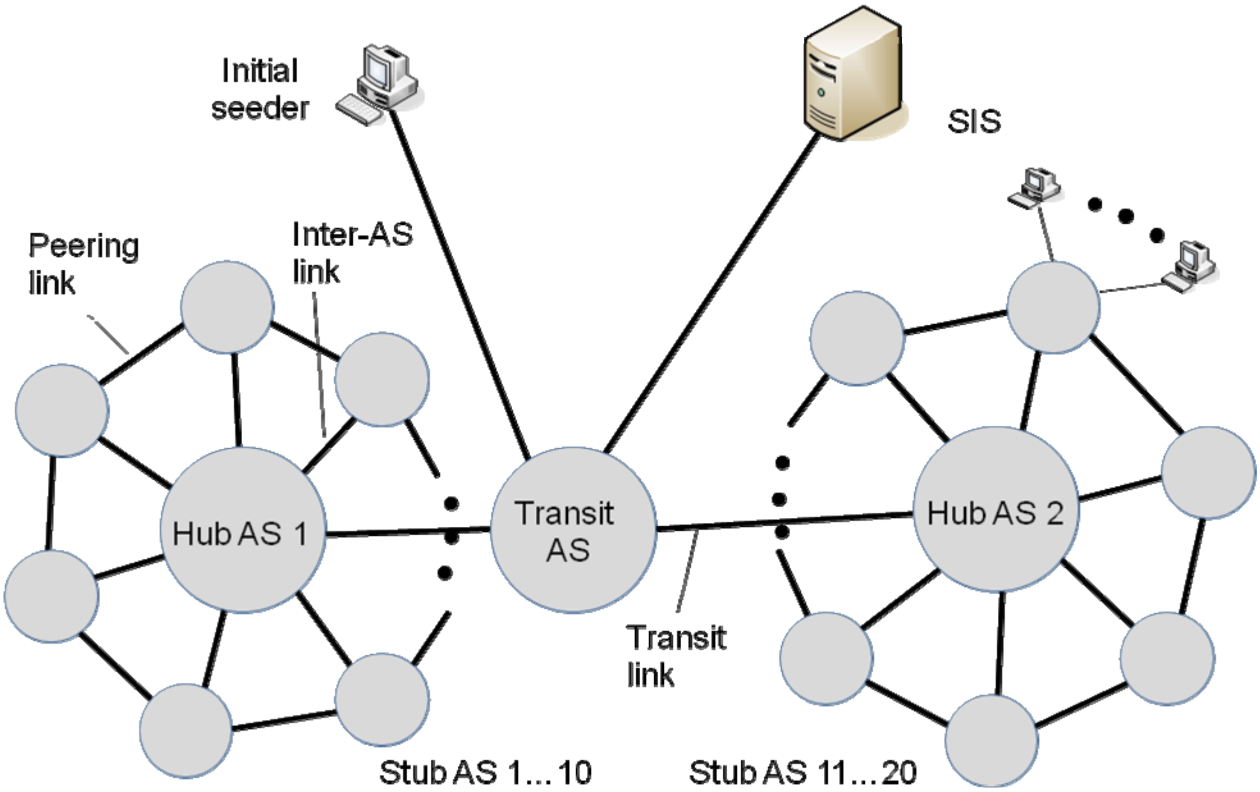


Figure 3: Complex simulation scenario topology

4.2.1.2 Generic Parameters

As in the symmetric scenario (presented in [D2.5]), these parameters are flexible due to the easier handling of the simulations. Parameter settings can be found in Table 4.

Table 4: General parameters for complex scenario

	Mean Swarm Size	**Overlay**	**Content**	**Seed percentage**
General Parameter	120 - 200	BitTorrent, Tribler	Simpsons	2-3%

4.2.1.3 AS Definition

The topological structure of the complex scenario is as flexible as in the symmetric scenario. Thus, the attributes of the different AS may still vary in the simulations. Default parameter settings can be found in Table 5.

Table 5: ASes definition for complex scenario

	End-Customer ISP	Medium ISP	Transit Provider
Quantity	20	2	1
Tier	3	2	1
Relative Local Swarm Size	1	0	0
Relative Seed Share	0	0	1
Swarm/Peer behavior	Steady Swarm/ Leecher	-	-
Category	High Mixed Quality	Transit	Transit
SIS Usage Percentage	100%		-

4.2.1.4 *Exemplary Experimental Setup*

4.2.1.4.1 SIS gains

This experiment provides insights on the effectiveness of SIS usage for the different ISPs in a multi-ISP setting (cf. Table 6).

Table 6: SIS usage for complex scenario

	Type 2 ISP
SIS Usage Percentage	0% - 100%

4.2.1.4.1.1 Different inter-connection links

This experiment investigates the influence of different interconnection links with and without SIS in conjunction with peering. Parameter settings can be found in Table 7.

Table 7: Interconnection links for complex scenario

	Inter-AS links	**Other**
Seed factor	Fast (unlimited) symmetric, 3 Mbit/s	Fast (unlimited) symmetric
SIS Usage Percentage	0%/100%	

5 ETM Configuration Parameters

The purpose of this section is to provide an overview of the parameters applicable for each of the three ETM mechanisms that were selected for implementation and deployment in the internal and/or the external trials. The parameters are categorized according to the entity where they are applicable:

- client (normal peer),
- SIS (SmoothIT Information Service),
- IoP (ISP-owned Peer),
- NMS (Network Management System), applicable to Highly Active Peer (HAP) only.

Most of these parameters are derived from the specification of each ETM mechanism, presented in [D2.3]. The absence of certain parameters is due to the fact that not the entire functionality described in WP2 was actually implemented in WP3. Hence, certain functionality, and thus parameters are missing. Please refer to [D3.3] for the actual implementation of the ETM mechanisms.

A short description is provided per parameter, as well as some typical values that can be used in the trials, wherever applicable. Note that these parameters, especially the ones related to the client overlay application, are not visible to the end users. A typical end user could not change their value. However, there exists the risk that technically skilled users could change some of those values.

5.1 BGP-Loc ETM Mechanism

The objective of this first ETM mechanism is to promote the localization of overlay traffic. This is achieved by both the client and the server sides. The parameters for both sides are specified below.

5.1.1 Client

There are two overlay functions that, when biased, can promote locality. These are

- the neighbor selection and
- the unchoking overlay procedures.

For BGP-Loc, or any other locality-promoting mechanism, these procedures can be biased in favor of local peers. In Biased Neighbor Selection (BNS), the parameter *f* denotes the ***percentage of connections*** that is devoted to highly-rated peers, while the remaining ones are devoted to peers selected by the Tit-for-Tat (T4T) or the Give-to-Get (G2G) overlay procedures. A typical value for *f*, used in the simulative as well as experimental performance evaluation is *f=0.9*.

Regarding the Biased Optimistic Unchoking (BOU), the ***number of slots*** devoted for the biased optimistic unchoking and/or the ***optimistic unchoking probability*** are the parameters for configuration. Typically, only one slot is available for optimistic unchoking, so the number of slots is equal to 1. For biasing the optimistic unchoking probability among the candidate peers, the typical procedure is to select the best (in terms of locality) peer suggested by the SIS and unchoke it. If more than one peers are considered as local (receive the same SIS rating) then the client should arbitrarily select one of them, i.e. with uniform distribution.

5.1.2 SIS

For the server-side support of BGP-based locality promotion, there are no parameters that can be configured. One can consider the ***weights of the rating metrics*** as configuration parameters, but in order to capture the concept of BGP routing preferences, the weights are defined by the algorithm itself. Of course, the administrator of the SIS can define his own weights in order to materialize a different rating algorithm.

5.2 IoP ETM Mechanism

The IoP mechanism introduces an ISP-owned peer into the overlay. Since IoP is transparent to the rest of the overlay and, the configuration parameters are IoP internal, either obtained through communication with the SIS or from a local configuration file. The ***number of IoPs*** to be inserted in a domain is the first factor to consider. However, as stated in [D2.3], we only consider the simple case where only one IoP is launched per domain.

One important parameter of the IoP is the network resources allocated to it, namely the ***uplink and downlink capacity*** of its access link. The more upload capacity the IoP has the more local peers he can serve and the more localized the overlay traffic will be. On the other hand, the more download capacity the IoP has, the faster the content will become available to the local part of the swarm. The specific values are related to the dimensioning of the network and the access capacity of the "average" peer in the ISP's domain.

Another crucial parameter of the IoP is the ***minimum upload and download bandwidth*** the IoP can assign per swarm. This again has to do with the access capacity of the average peer in the ISP's domain, how many swarms are typically active in a domain and in how many of them the IoP decides to participate in.

Another parameter having to do with the physical characteristics of the machine hosting the IoP is its ***storage space capacity***. The larger the capacity is, the more content the IoP can store in its cache. Due to the caching policies implemented in the IoP, unpopular content will eventually be replaced. Having however more storage capacity can help serve swarms belonging to the long-tail of the content.

The ***period after which the IoP asks the SIS for new swarms*** and the ***period for re-allocating the available bandwidth*** to the swarms the IoP is connected are also two important factors. Typically, the time period to ask for new swarms should not be too short.

It is expected that four to five times per day are more than enough for the IoP to react to the changes in popularity for certain swarms. Respectively, the period to re-allocate the bandwidth among the swarms should not be less than an hour, because very frequent changes to bandwidth allocations may result in an unstable behavior of the IoP. A typical value would be half or one third of the time it takes for the IoP to ask for new swarms.

Moreover, related to the previous parameter, is the ***percentage of old swarms*** that the IoP will keep participating after the update of popular swarms. A typical value for this would be around 40%-50%, in order to keep participating in some swarms that even though may not be popular enough in the specific monitoring period, it may be advantageous to keep them in order to cover the probability of re-appearing as popular ones.

Another parameter need to be decided is whether the IoP should unchoke ***remote (non-local) peers*** or not. Obviously by unchoking remote peers, the IoP can attain the entire content faster but the decrease of the uplink and downlink inter-domain traffic will not be significant. By not unchoking remote peers the content might be downloaded slower, but the final effects on the inter-domain traffic will be important.

Finally, the ***number of swarms*** that the IoP can join in parallel is a parameter for consideration. This parameter is in direct relation with or, better, derives from the available upload/download capabilities of the IoP and the minimum upload/download bandwidth to assign per swarm.

Note that in this deliverable, only the parameters that have been both specified in the mechanisms specification, to be found in [D2.3] and have been implemented (cf. [D3.3]) are presented. There are a few parameters that even though they have been initially specified, they were simplified during the implementation phase. These are the ***unchoking policy*** and the ***number of unchokes*** for the IoP in the leeching phase. These are overridden by the respective parameters of the BGP-Loc ETM mechanism, namely the *number of unchoking slots* and *optimistic unchoking probability*. In other words, in the leeching phase, the IoP behaves as every other BGP-Loc enabled client.

5.3 HAP ETM Mechanism

In the HAP ETM mechanism, the SIS server and the NMS of the ISP cooperate to decide which peers should be promoted to HAPs so as to help the local overlay community to experience a higher QoE (e.g. better download times). This is achieved by the ISP (through the NMS) granting more network resources to certain peers that are expected to help (e.g. seed more) other local peers. It is made obvious that the only two entities that should be configured to support the HAP functionality are the SIS and the NMS.

In Appendix C the current status of the specifications of the HAP as of the writing of this document can be found.

5.3.1 SIS

The version of the HAP to be validated in the course of the trials is combined with BGP-Loc, where plain BGP-Loc mechanism is adjusted to give higher rating to active HAPs. Thus the usage of the additional resources by local peers is promoted. The ***additional weight given to*** HAPs when performing peers rating is the first parameter to be defined

for SIS side of the experiment. The absolute value is of no importance as long as the HAP gets a higher rating than all other peers.

The second parameter is actually a family of ***weights*** (namely p_1, p_2 and p_3) that affect the HAP rating function. These weights define the importance of the three factors that form the rating formula, i.e. the percentage of upload traffic directed to local peers, the seeding ration and the percentage of upload traffic actually used by the overlay. Since up to now no special importance has been given to any of the three factors, the typical value for all weights is equal to one. Note however that we offer the possibility for the ISP to alter the way these factors affect the final outcome of the HAP rating function.

It is also necessary to define the duration of time interval ***t*** for which the aggregation of the behavioral statistic is performed and which also corresponds to the frequency of HAP promotion. This is closely related to the capabilities of the NMS, but it is estimated that very frequent changes in the status of local peers will not serve the original purpose of the mechanism, since a stability of decisions is required for a certain time period. Typically, updating of the status of peers on a daily basis is expected to serve well the desired purpose. The last parameter to be defined is the duration of historical data ***D*** to be considered.

5.3.2 NMS

Other decisions, similar to the ones for the IoP ETM mechanism, are summarized in the following three parameters: the ***total extra throughput capacity*** to be offered in the domain, i.e. allocated to the HAPs, the ***number of HAPs*** that should simultaneously exist and the ***uplink and downlink capacity*** assigned to them. Of course, these three parameters are highly correlated. Indeed, it suffices to determine only two of them, e.g. the amount of resources to be allocated per HAP can be found by dividing the total amount of network resources available to the domain with the number of HAPs to be deployed.

5.4 Summary

In this section we provide a summary of all the parameters required for each ETM mechanism (based on specification provided in [D2.3]) as well as which of them have been eventually implemented (based on implementation documentation at [D3.3]).

Table 9: Summary of all ETM-related parameters

Parameter Name	**Implemented**
BGP-Loc ETM mechanism	
Percentage of connections (devoted to highly rated peers)	Yes
Number of slots (for biased optimistic unchoking)	Yes

Biased optimistic unchoking probability	Yes
Weight of rating metrics	Yes
IoP ETM mechanism	
Number of IoPs	Yes
Download bandwidth	Yes
Upload bandwidth	Yes
Minimum upload bandwidth per swarm	Yes
Minimum upload bandwidth per swarm	Yes
Number of swarms to join	Yes
Storage space	Yes
Period to ask for new swarms	Yes
Period to reallocate bandwidth	Yes
Percentage of old swarms	Yes
Unchoke remote peers	Yes
Unchoking policy	No (replaced by *biased optimistic unchoking probability* of the BGP-Loc case)
Number of unchokes	No (replaced by *number of slots* of the BGP-Loc case)
HAP ETM mechanism	
Weight for new peer rating formula	Yes
Weights for HAP rating formula	Yes
Total extra throughput capacity (in the domain)	Yes
Upload bandwidth	Yes
Download bandwidth	Yes
Number of HAPs	Yes
Period to renew HAPs	Yes

6 External Trial Experimental Setup

The organization of external trial faces several restrictions that limit the number of ETMs that can be evaluated.

To encourage the real users to participate in the trial, the distributed content should be attractive for them. The content is provided by third party companies/organizations not participating in the project. The content is copyrighted and restricted to Primetel users only. According to the agreements with content providers, 15 full length movies as well as a larger movie trailers are available.

The number of Primetel customers that may be invited to participate in the main stage of the external trial is limited to 25.000 in total. According to PrimeTel's estimation of interest level of users, it is expected that around 1000 users will actively participate.

To separate the effect imposed by the deployment of the various ETM mechanisms, separate scenarios are defined for the different ETM mechanism. Also a separate trial setup specific for different ETM mechanisms is necessary. In order to be able to validate the results and evaluate credibly the effect of ETM mechanism deployment, a *reference scenario* is defined, where no ETM mechanism will be applied. This amounts to SIS not performing any rating, IoP not joining any swarms and no HAP's being promoted. To ensure that no interference happens between the ETM mechanisms and, given the limited duration of the trials and user involvement level, the experiments will run sequentially.

Users may not be willing to watch a given movie more than once (e.g. under different scenarios). Thus the set of available full length movies needs to be divided between the different scenarios. A smaller set of movies will be offered in each scenario. Another important issue is that users' interest for participation may decrease in the course of the trial and the number of actively participating users decreases. To keep the users' interest at a similarly high level in all scenarios, additional promotions will be announced by Primetel.

Taking into account the expected number of active users, the necessity to run scenarios sequentially and the limited number of available movies that must be shared between experiments, the number of ETMs evaluated in the trial must be limited. On the basis of several issues, including ETM mechanism specification, complexity, simulation based evaluation, etc., the following ETM mechanisms were selected to be tested and evaluated in the external trial:

- IoP combined with BGP-Loc and
- HAP combined with BGP-Loc.

These are considered as the most promising, innovative, and original solutions developed by the project. Thus, there will be in total 3 experimental scenarios, including a *reference scenario*, run sequentially.

Table gives the values for the parameters defined in Section 3, as far as defined by the time of this deliverable.

Table 10: Parameter settings for external trial

Parameter Name	PrimeTel users	G-Lab clients
Swarm Parameters		
Number of peers	Not configurable (depends on user interest)	Min. 2, max. 10 (see **Fehler! Verweisquelle konnte nicht gefunden werden.**)
Peer availability	Not configurable (user behavior)	100% during download
Initial number of seeds	0 + Support Seeder	0
Initial number of leechers	0	2
Initial leecher content share	0%	0%
Total seed capacity	Not configurable (user behavior)	0 Kbit/s
Arrival process	Not configurable (user behavior)	Not pre-configurable (coupled to PrimeTel arrival process, see **Fehler! Verweisquelle konnte nicht gefunden werden.**)
Share of free-riders	Not configurable (user behavior)	0%
Share of firewalled/NAT peers	Not configurable (user behavior)	0%
Download bandwidth	Not configurable (can be set by user, depends on access speed)	1 Mbit/s
Upload bandwidth	Not configurable (can be set by user, depends on access speed)	512 Kbit/s
Video and audio codec	H264/mpga	
Bit rate of video files	0.7 — 1 Mbit/s (VBR)	

Size/length of video files	1-5m for trailers, ~90m for movies	
Popularity	Not configurable (user behavior)	Not configurable (coupled to PrimeTel users)
Chunk/block size	512 KB/16KB (depends on file size)	
Client Input Parameters		
Upload/download slots	4/∞ (default)	
Seeding behavior	Not configurable (user behavior)	Seeding time 0 after ‘watching’ the video
Neighbors min	5	
Neighbors max	15	
Initial seeding	No (Tribler default)	
End game mode	Yes (Tribler default)	
Timeout values	Tribler default	
Tit-for-tat parameters	Give-to-get (Tribler default)	
Chunk/peer selection strategy	Priority set based (Tribler default)	
Video streaming parameters	10s high prio./40s mid prio. (Tribler default)	
ISP-related Parameters		
Hierarchy Level	Tier 3 (fixed)	n/a
Bandwidth of the core network	(fixed)	1 Gbps
Access type composition	(fixed)	512 Kbit/s upstream, 1 Mbps downstream, delay < 1ms
Number of users	Not configurable (depends on user interest)	2 < N < 10
Number of concurrent active swarms	Not configurable (depends on user interest)	
Distribution of users across the ASes	Not configurable (depends on PrimeTel user interest)	
Distribution of users within an AS	Not configurable (user activity dependent)	n/a
Distribution of users per access type	Not configurable (user activity	n/a

	dependent)	
Distribution of users per swarm	Not configurable (user behavior, content popularity)	Coupled to PrimeTel user behavior
Number of seeds per domain and its associated access type	Not configurable (user behavior, content popularity)	0
Percentage of SIS usage	0%/100% (depending on experiment)	0%
Delay	Not defined yet (to be measured shortly before trial)	
Bandwidth	Not defined yet (to be measured shortly before trial)	
Status of intra- and inter-domain links	Not defined yet (to be measured shortly before/during trial)	Not defined yet (to be measured shortly before/during trial)

6.1 Restrictions

One of the main restrictions of the external trials comes from the permissions for content distribution, which is restricted to Cyprus only. Therefore in all of the topologies used for the experiments all peers outside the PrimeTel network must be represented by headless clients. This also has a side effect in terms of users' feedback, which will only be available for one AS at a time. All other clients will only be able to provide automated monitoring information.

In order to have successful trials and maintain users' interest during the whole period it is important to maintain acceptable level of QoE. Therefore the experiments will provide limited opportunities to test extreme situations of bandwidth starvation inside PrimeTel's AS. This scenario can only be simulated inside systems with automated clients.

It is also important to understand that, due to the nature of the external trial, the situation implemented will be the one when most of the peers participating in the swarm are using the corresponding ETM mechanism. This restriction does not allow testing the early adoption stage when a small part of the swarm will be located in an SIS-enabled domain.

6.1.1 Technical Limitation

Here, we describe the technical limitations for the experiment, mainly related to the limitations of the G-Lab experimental facility ([G-LAB]). G-Lab is an experimental facility based on the PlanetLab software ([PlanetLab]), which allows for a good control over used nodes. Parameters like upload and download capacity of clients installed on G-Lab nodes are configurable, while the necessary infrastructure (processing power, physical link bandwidths, virtualization tools) is provided by the facility.

However, the nodes in the G-Lab network are more stable and reliable than PlanetLab nodes, and G-Lab sites exist at the premises of SmoothIT partners UniWue and TUD.

Therefore, G-Lab is more suited for usage in the external trial than other facilities or testbeds. The limitations described below should not affect the outcome of the external trial negatively for the expected swarm sizes (see also Appendix A).

G-Lab Upload Capacity

The total upload capacity from all peers installed at the University of Wuerzburg G-Lab site is limited to a maximum of 10 Mbps. This bandwidth may not be exceeded and can only be reached during short peak times. On the average, the upload capacity will be below 5 Mbps.

Thus, also the number of swarms that can be supported in parallel by seed capacity in the G-Lab network is limited. As leecher peers are to be installed to generate bandwidth demand outside of the PrimeTel network, this potential seeder capacity is further reduced.

Number of IP Addresses

The number of distinct, externally visible IP addresses that can be used for the installation of peers is limited to 25, i.e., the number of nodes at the University of Wuerzburg G-Lab site. More peers can be installed per node, but will then share the same (external) IP address.

BGP-based SIS rating

Since there is no access to BGP routers for nodes at the G-Lab sites, the standard SIS server cannot be used in this environment. Thus, peers installed in G-Lab need to be either preconfigured with SIS ratings for other G-Lab peers and peers in the PrimeTel network, or a ‘dummy’ SIS needs to be installed that returns hard-coded ratings.

Adaptation of peer capacity

Once a peer is configured with an upload and download capacity and started on a G-Lab node, this capacity cannot be changed during the runtime of the peer. Thus, the HAP ETM cannot be applied to G-Lab peers.

6.1.2 Legal aspects

All content is provided for the purposes and duration of SmoothIT trials only. Any access to the content outside PrimeTel network should be restricted to the headless clients operating in SmoothIT partner networks or to the real client in the network of the content owner. Furthermore, the content needs to be encrypted.

6.2 Initial and Support Seeding

The most realistic and at the same time bandwidth consuming case, especially for smaller Tier 3 ISPs, is the one when content originates from a different AS. Furthermore this is the case that allows testing and demonstrating locality improvement and associated benefits. However, due to limitations imposed by G-Lab on outgoing traffic it is not feasible to place there the initial seeders. Nevertheless, the experiments are designed in such a way that, given the static location of the initial seeds in the network, they can be considered as

external, even though physically connected to the PrimeTel's AS. For the purposes of the trials the BGP routing table, used exclusively by SIS server for metering, will be modified in such a way that initial seeds would get the same or lower rating as G-Lab peers. Furthermore, as all traffic from the initial seeds is accounted for, the assessment can easily consider these to be a special case.

One of the challenges faced during the design of external trials experiments is the requirement to maintain good or at least acceptable level of QoE for all participating customers in order to keep their interest in the project. The target level of experience is when customer experiences no stalling during playback and startup time is minimal given the restriction of the access bandwidth and the bit-rate of the video. For this purpose in all experiments a ***support seeder*** will be available, which will only join the swarms in case one of QoE metrics for one of the users in PrimeTel domain falls below a threshold value. The support seeder is configured to connect only to "starving" (or underserved) local peers. Thus it will not be announced by the tracker, in order to avoid side effects on the ETM assessment. This support seeder can be located in any part of the experimental setup, but from practical aspects and given the above mentioned limitation of G-Lab it is best to be placed inside the PrimeTel AS. G-Lab peers don not need to be supported by this seed, since assessment of ETM performance is done on the results acquired in PrimeTel's domain as the one employing the ETM.

For the purposes of the assessment of the external trials results, upload traffic generated by the support seed can be considered as a metric negatively correlated to the generalized QoE that would apply in the absence of this seed. On the other hand, in the presence of the support seed, QoE in PrimeTel's customers is expected to be similar in all scenarios. Thus, gathering of QoE related feedback from users by means of questionnaires is essentially meaningless and will not be carried out.

The decision on which peers are starving and therefore require support is based on the state of the play-out buffer which is communicated to the modified version of the tracker. Once a peer is identified as being starving, the support seed is instructed to connect to this peer. More details on the algorithm can be found in Appendix B.

The support seeder configuration is identified by two important parameters. The first is *suffer time,* which identifies how long the support seed waits from the time a peer reported an empty layout buffer until it starts supporting the seed. As we attempt to keep QoE at the optimal level, but also allow for some fluctuations in the swarm, setting this parameter to 5s is expected to yield good results. The second parameter is the *amount of bandwidth* dedicated to the support seeder. As all content will only be available locally, the uplink bandwidth is relevant. This should cover for the worst case scenario, i.e. all potentially parallel peers in PrimeTel's AS leeching without a single seed available. Given the expected participation rate and the bit rate of the video, a 10 Mbps link should be sufficient and cover all potential needs.

6.2.1 Topology

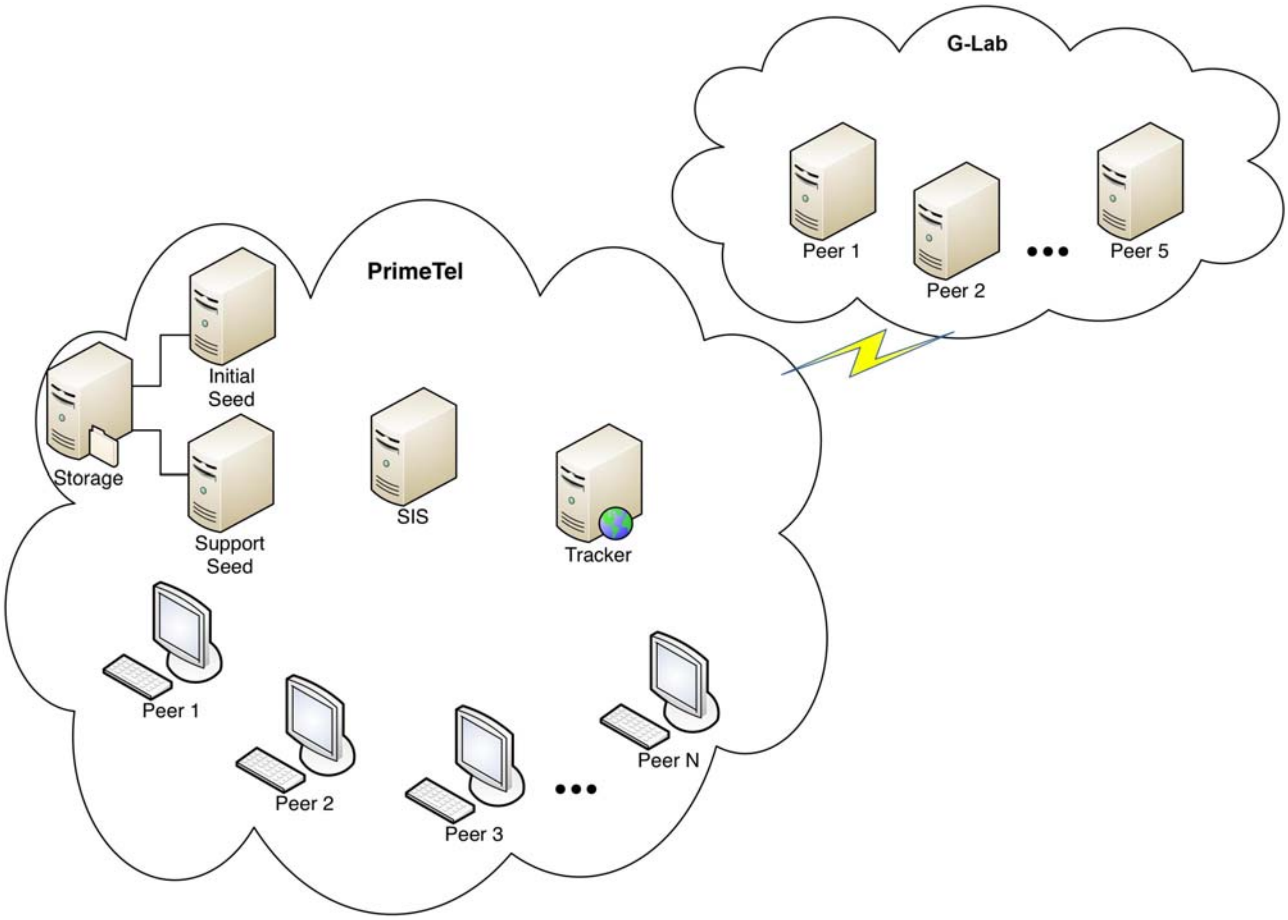


Figure 4: Network layout as used in the external trials.

While the setup of different experiments might add specific elements or properties to the topology, they will all share the same general layout that is shown in Figure 4.

All experiments include two ASes, PrimeTel and G-Lab, which is the minimal number required for considerable assessment of the ETM mechanisms being tested. The PrimeTel AS will include

- real users (Peer 1..N),
- the tracker,
- SIS,
- initial seed, and
- the support seeder.

The latter will only support the swarms if QoE metrics drop below the threshold. The G-Lab AS will host up to 10 headless peers simulating realistic arrival process.

The two ASes will be interconnected via a backbone link; no special peering agreement between the two systems will be assumed.

6.2.2 G-Lab Setup

In this section, we define the behavior of the peers that can be controlled during the experiment, namely the G-Lab peers.

The peers in G-Lab generate demand for the content to match the demand in the PrimeTel network and emulate 'the rest of the world'. Otherwise, the complete seeder capacity would always be utilized by the leechers in the PrimeTel network, while no traffic would flow from the PrimeTel network to the G-Lab sites. Thus, no effect of ETM mechanisms on the traffic distribution would be observed in the absence of G-Lab peers.

Using the G-Lab testbed in the external trial allows testing biased unchoking, as the G-Lab peers generate additional demand outside the local AS. This however requires keeping the number of G-Lab peers close to number of active local peers to avoid all seeder slots from being occupied by external peers. At the same time G-Lab peers will allow to validate the neighbor selection process, as they will also upload content while they are still downloading and they are a possible alternative in the neighbor selection for the local peers.

G-Lab has a fixed amount of resources available for the purposes of the external trial, which puts an upper bound on the number of peers running simultaneously. All peers have an equal amount of resources available, which approximately matches an average peer in PrimeTel's domain. The bandwidth dedicated to each peer is 512 Kbps for the upload and 1 Mbps for the download. Given the G-Lab limitation this allows for up to 10 simultaneously running peers. Should a need arise to have more artificial peers (e.g. if participation rate in PrimeTel is higher than expected) it will be possible to extend the G-Lab capacity by using nodes of other partners or to place extra peers inside the PrimeTel network. In the latter case the SIS can be configured to consider these peers as external. However this should only be used as a last resort measure.

It is expected that the largest effects will be observed if the number of leechers in the PrimeTel network is roughly equal to the number of leechers in the G-Lab network. Thus, the peer arrival process in G-Lab should adapt to the number of leechers in the PrimeTel network until the G-Lab capacity is exhausted (10 peers if restricted to UniWue only). However, at least two peers should always be running in G-Lab at all times.

To achieve this, a headless peer is started in G-Lab whenever a new peer from the PrimeTel network contacts the tracker, as long as the maximum value of 10 peers is not reached. Thus, each peer demanding upload capacity in PrimeTel is matched by a peer in G-Lab as long as the swarm is small.

Finally, in order not to influence the seeder capacity, the headless G-Lab peers go offline as soon as they have finished "watching" the content, thus they will be seeding the content or part of the content for the duration of the video. The G-Lab peers do not promote locality but use the reference client behavior.

Simulation results (see Appendix A) show that, depending on the number and behavior of clients in the PrimeTel network, the demand generated by the G-Lab peers should lead to a measureable improvement at least w.r.t. the inter-domain traffic. More details about assessing the benefits for ISP and the user, and whether win/win or win/no-lose apply are provided in section 6.3.1.

6.2.3 Timeline

The external trials are separated into three phases for the three experiments in the following order:

- Reference,
- IoP+BGP-Loc, and
- HAP+BGP-Loc.

Each phase will last one month. During the first phase, only half of the content is made available. The second phase gives access to the remaining half. It is expected that by the last phase of the trials users' interest might drop. In order to ensure good participation rate some extra incentives might be given to customers in form of promotion. Detailed definition of these, however, falls outside the scope of the deliverable and are to be handled by the marketing team of PrimeTel as part of the WP4 work. This sequence is expected to keep customers' interest at constant level.

6.3 Assessment of Results and Measurements

Assessment of results and validation of TripleWin scenario in the context of external trials might not be straightforward due to the limited size of the swarms, and thus the possibly limited effect they will have of traffic patterns in general. Under such circumstances direct measurements of inter-domain traffic will most probably not reveal any visible changes as the traffic generated by the trials will be small compared to the overall bandwidth usage. The requirement to maintain a sufficient QoE for the users poses another challenge, ruling out the possibility of direct measurements at the client side.
This Subsection presents the means of assessing the benefits for each of the parties (and whether a "win", a "no-lose" or a "lose" applies) based on the external trial results and the measurements should be taken during all three experiments for assessment purposes. Certain experiments require extra measurements and these are described in the respective sections.

6.3.1 Measurements to Evaluate *Win/No-Lose/Lose* for ISP

A Win situation for the ISPs is the one that involves a reduction of costs associated with the inter-domain traffic. However, as described above given the scale of the trials limited by expected users' participation rate it is hard to expect noticeable changes to PrimeTel's traffic pattern. Therefore it is suggested to evaluate the win/no-lose/lose for the ISP indirectly by measuring:

- The traffic incurred from all hits on remote peers
- The count of *missed local opportunities* (see below)

Together these metrics allow assessing the gain of the ISP in terms of inter-domain traffic and the level of locality introduced by the respective ETM mechanism.

A *Missed Local Opportunity* (MLO) is defined as an event where a local peer is downloading a chunk from a remote location (also referred to as *hit*) when there is another local peer seeding this chunk wherefrom its downloading was possible according to resource availability. In order to identify MLOs the following information is required:

- All hits to remote peers (i.e. G-Lab or any other peer considered as remote for the purposes of the trials).
- Upload bandwidth utilization of the peers
    - Total
    - Used by the Tribler application
- Upload bandwidth limitation set in the client
- Real-time information about files/chunks availability in the local domain.

All of the above metrics except for content availability are directly obtainable via monitoring component or from the NMS. Identifying effective content availability, however, is more complex and requires local peers to report each complete chunk download as well as the events when they go online or offline. This can be achieved by extending the measurement component of the client to include reports on any downloaded chunk.

It is expected that if ETM mechanisms behave as expected a significant decrease in the number of MLOs should be observed as compared to the reference scenario. It should not be expected for this number to go to zero level, however, due to the specifics of the unchoking algorithm.

6.3.2 Measurements to evaluate *win/no-lose/lose* for end-user

The evaluation of *win/no-lose/lose* for an end-user is basically related to QoE. The most important metric for video applications is stalling (the number of occurrences during playback, its duration etc.). However, as already mentioned, to maintain users' interest in the participation in the trial it was decided that a perceivable quality degradation of the video playback should be avoided. To achieve this, the use of the support seed to prevent video stalling was introduced. If there is a danger of playback stalling, a given peer download missing block from the support seed. As a result, a real user should not experience video stalling and always receive the same high quality playback.

The evaluation of *win/no-lose/lose* for an end-user is still possible. It is performed by measuring the traffic downloaded from the support seed. The more content a peer downloads from the support seed, the lower the video quality is experienced by that peer. In order to assess the situation it is necessary to record the traffic generated by the support seed on a per-peer per-swarm basis. This granularity of measurements is possible only if the data is provided by the support seed itself.

Additionally, an objective evaluation of QoE is performed in order to ensure that the support seed operates as expected and account for any singularities in the final assessment should they arise. The objective QoE is evaluated by constantly monitoring the state of the playout buffer. This allows identifying stalling and effective startup times. If video stalling occurs the reason should be determined (e.g. the support seed was not working properly) and, if possible, an appropriate action should be taken, e.g. an adjustment of the settings to prevent video stalling in the future. To be able to analyze the reasons for video stalling and related circumstances it is necessary to collect information on the timestamp of the stalling occurrence, stalling duration, the number of missing chunks, details on the neighbor set etc.

Measurements of the QoE will be made both for real customers and for headless clients. The latter results will be used to evaluate in addition the consequences of deploying ETM mechanisms in one domain on the performance of the peers in other domains.

The measurements will be performed by a measurement module embedded in client software.

6.3.3 Measurements to Evaluate *Win/No-Lose/Lose* for Content Provider

Overlay providers' Win/no-lose/lose situation is assessed based on the content availability in the whole overlay. This statistic is similar to the one collected in order to identify MLOs with the sole difference that G-Lab peers should also be included in the measurement.

Moreover, the load on support seed can be considered as the secondary metric which is relevant for the peer-assisted CDNs which also maintain customer QoE fixed by providing the missing bandwidth though the content provider's servers.

6.3.4 Summary of Generic Measurements

This section presents the summary of all measurements to be taken during external trials including interval/event that will trigger the measurement, the source from which it will be collected (either ISP's NMS or the monitoring functionality built-in into the client application),) and any additional parameters that need to be recorded along with the measurement.

Description: Total upload bandwidth utilization by each peer

Parameters: IP address

Source: NMS

Granularity: 1 minute

Description: Tribler-generated upload bandwidth utilization by each peer

Parameters: IP address

Source: NMS

Granularity: 1 minute

Description: Upload bandwidth limitation set by customer

Parameters: peer ID

Source: Monitoring

Granularity: Event based

Description: Peer goes Online/Offline

Parameters: peer ID, IP address

Source: Monitoring

Granularity: Event based

Description: Chunk download complete

Parameters: peer ID, source peer ID, chunk ID, swarm ID

Source: Monitoring

Granularity: Event based

Description: Peer joins swarm

Parameters: peer ID, swarm ID

Source: Monitoring

Granularity: Event based

Description: Play-out buffer state

Parameters: peer ID, buffer state

Source: Monitoring

Granularity: 1 minute

6.4 Experiment 1 - Reference

The goal of this experiment is to gain data about the behavior of the Tribler swarm without using any enhancement mechanism. This data will be used, to measure the impact of the previously described mechanisms, by comparing this experiment's data against the results of the two previous experiments.
As this experiment does not include any mechanisms, that require additional configuration, all configuration settings are as in the general case.

6.4.1 Topology

The topology for this experiment is unchanged from the initially shown network layout.

6.4.2 Measurements

The measurements done for this experiment will not include any other data as previously described for the general case.

6.5 Experiment 2 – BGP-Loc and IoP

The purpose of this experiment is to evaluate the performance of the combination of two ETM mechanisms, i.e. the BGP-based locality promotion mechanism (BGP-Loc) and the ISP-owned Peer (IoP). The reason why we selected to study their combination and not the effects from their separate deployment is that simulation (cf. [PSS09]) and internal trial studies (cf. Appendix C of [D3.3]) have shown that their combination achieves a "win-win" situation for both the ISP and the peers, while when separated we reach a "win-no lose" or a "no lose-win" situation respectively. Thus, the outcome of such an experiment would provide a clear view on whether this combination leads to a TripleWin situation.

6.5.1 Topology

Since the ETM mechanisms will be deployed in only one AS in the experiment while all the second one will play the role of the “rest of the Internet”, it is obvious that both the SIS server and the IoP will be placed inside PrimeTel’s domain. The exact placement of the SIS is of no great importance since only singaling information will be exchnaged with the local peers. However, the placement of the IoP needs further attention. Even if it’s feasible (from the ISP’s network planning perspective) to allocate a large amount of resources to an IoP that resides at the edges of the domain (access network), such a placement should be avoided. The IoP should reside at a central point of the domain, attached or close to the core network of the domain, so that all local peers can benefit from its presence and be served with satisfactory rates.

6.5.2 Experiment-specific Parameters Values

This experiment contains two types of peers: normal peers (be it real customers or headless clients) and the ISP owned peer. While they run same codebase, the configuration is different in the context of the experiment.

The normal peers would employ biased overlay procedures (Neighbor Selection and Unchoking) as described in Section 5 to promote locality. To normal peers no other experiment specific parameters are relevant and the typical values should be used, as defined in [D2.3].

Regarding the IoP, as already mentioned in Section 5, the most important decision is the allocation of network resources to it. Assuming that the average access speeds of PrimeTel-customers ranges from 300 to 4000 Kbps, with 80% of user have access speeds of 2 Mbps or less and taking the expected participation rate into account, it is reasonable to anticipate at most 20 peers simultaneous distributed among the swarms. In order for the IoP to account for the potential request and yet keep room for peers to exchange content between themselves, a typical value for the access characteristics of the IoP should be at least 5 Mbps in both directions with a possibility of increasing it up to 10 Mbps if needed due to an unexpectedly high participation rate.

All the BGP-Loc and IoP related parameters along with their values to be used for the first experiment as summarized in the table below.

Table 8: Parameter settings for experiment 1 of external trials

Parameter Name	Value
BGP-Loc ETM mechanism	
Percentage of connections (devoted to highly rated peers)	0.9
Number of slots (for biased optimistic unchoking)	1
Biased optimistic unchoking probability	Uniform (among equally high rated peers)
Weight of rating metrics	Automatically set (by the BGP-based rating algorithm)

IoP ETM mechanism	
Number of IoPs	1
Download bandwidth	5 Mbps
Upload bandwidth	5 Mbps
Minimum upload bandwidth per swarm	1.5 Mbps
Minimum upload bandwidth per swarm	1.5 Mbps
Number of swarms to join	10
Storage space	10 GB
Period to ask for new swarms	Every 6 hours
Period to reallocate bandwidth	Every 3 hours
Percentage of old swarms	50%
Unchoke remote peers	False

6.5.3 Experiment-specific Measurements

In addition to generic measurements this experiment will require monitoring the IoP link utilization (load) and also gathering the total volume of traffic uploaded and downloaded.

6.5.4 External effects

The number of peers inside and outside PrimeTel's domain will play a significant role to the outcome of the experiment. If too few peers reside outside, then it is expected that hardly any difference for the ISP will be observed, with respect to the reference scenario and in terms of inter-domain traffic exchanged. If the content is already inside the network, then applying direct (BGP-Loc) or indirect (IoP) traffic localization techniques will not do much better than letting the overlay work the standard way. However, significant change in support seed traffic can be observed thus affecting QoE metric for the customers.

6.6 Experiment 3 – HAP and BGP-Loc

The goal of this experiment is twofold. First, it shall show that the combination of these HAP and BGP-Loc ETM mechanisms leads to the TripleWin situation as defined for the SmoothIT project in general. Moreover there results will be compared with the IoP assisted BGP-Loc scenario to show that the introduction of HAPs leads to an improved QoE for customers without extra resources from ISP such as an extra peer. The other goal is to show that an increase in upload and download bandwidth given to the HAPs is a kind of

incentive that would lead to measurable changes in peers' behavior encouraging them to act more collaboratively towards the overlay in general thus leading to improved content availability and reduction in number of MLOs. In fact, this issue can only be investigated in the external trial, since it is dependent on the influence in the behavior of humans.

6.6.1 Topology

While the general topology of this experiment stays the same as described in Section 6.2.1, there are a few important changes that take place on the side of SIS integration in PrimeTel's network. The most important one is that SIS will provide an interface to PrimeTel's NMS which will be used to transfer the list of customers to be promoted to HAP status. Furthermore, SIS will have an interface to the billing system to get a list of customers who currently have HAP status.

6.6.2 Experiment-specific Parameters values

The main parameter that defines HAP ETM performance is the amount of additional bandwidth allocated for this purpose and the increase given to each of the HAPs which is limited by line properties and issues of financial feasibility. Given the restrictions of ADSL technology and the fact that many customers already have a 512 Kbps upload link, it is feasible to offer an increase of upload capacity of extra 512 Kbps. In order to ensure that customers actually see a clear benefit of HAP status, their download bandwidth can be increased by 1 Mbps. The total amount of resources dedicated for HAP purposes by PrimeTel is 10 – 20 MBps, depending on actual user participation.

All the HAP related parameters along with their values to be used for the second experiment as summarized in the table below.

Table 9: Parameter settings for experiment 2 of external trial

Parameter Name	**Value**
HAP ETM mechanism	
Weight for new peer rating formula	1
Total extra throughput capacity (in the domain)	10 Mbps (so as to be comparable with the IoP case)
Upload bandwidth	unaffected
Download bandwidth increase	1 Mbps
Upload bandwidth increase	512 Kbps
Number of HAPs (t)	10
Period to renew HAPs	Once a day

History length (***D***)	7 days

6.6.3 Experiment-specific Measurements

In addition to generic measurements this experiment will require detailed traffic profiles of the peers promoted to HAP including their local and total upload traffic as well as their download traffic for the whole period of being a HAP. Furthermore, it is important to distinguish in measurements the traffic generated by Tribler vs. all other applications that can be done on per-port basis. Combined these measures allow to measure financial feasibility of HAP promotion.

7 Summary

This deliverable provides the second part of the parameter set used for simulations, internal and external trial. Its main focus is on the external trials that will be executed in the PrimeTel network and partly in the G-LAB testbed. For that purpose, where possible exact values are provided for the configuration of the client software, trackers and SIS components, as well as the planned process of experiments for the external trials. Finally, this deliverable provides the metrics and evaluation parameters that will be needed for WP4 to assess and disseminate the results of the experiments and the effects of the SmoothIT overlay traffic management mechanisms on the ISP, the content provider, and the end users' side.

Specifically, this document provides definitions of three experiments to be performed in the external trial. These experiments are designed to test the BGP-Loc, the IoP, and the HAP ETM mechanisms, which are output of previous work (cf. [D2.3]) of the SmoothIT project. To measure the improvement introduced by those ETM mechanisms, another experiment is defined in which none of those ETM mechanisms are used.

8 References

[D1.1]. The SmoothIT project: Deliverable D1.1 – Requirements and Application Classes and Traffic Characteristics (Initial Version), March 2009

[D2.2] The SmoothIT project: Deliverable 2.2: ETM Model and Components (Initial Version); December 2008

[D2.3] The SmoothIT project: Deliverable 2.3: ETM Model and Components and Theoretical Foundations (Final); September 2009

[D2.5] The SmoothIT project: Deliverable 2.5: Comprehensive Test-bed and Trial Parameter Set Definition (Part I); March 2009

[D3.1] The SmoothIT project: Deliverable 3.1: Economic Traffic Management Systems Architecture Design (Initial Version); September 2008

[D3.3] The SmoothIT project: Deliverable 3.3: Documentation of Engineering and Implementation (Final); March 2010

[FH264] VideoLAN – x264. http://www.videolan.org/developers/x264.html

[H264] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra: Overview of the H.264/AVC video coding standard, Circuits and Systems for Video Technology, IEEE Transactions on Circuits and Systems for Video Technology, Vol.13, No.7, pp. 560-576, July 2003

[AAC] Apple – AAC Audio. http://www.apple.com/quicktime/technologies/aac/

[BT1] BitTorrent specification – Super Seeding, http://wiki.theory.org/BitTorrentSpecification#Super_Seeding

[BT2] BitTorrent specification – End Game, http://wiki.theory.org/BitTorrentSpecification#End_Game

[MPME08] J. J. D. Mol, J. A. Pouwelse, M. Meulpolder, D. H. J. Epema, H. J. Sips: Give-to-Get: Free-riding-resilient Video-on-Demand in P2P Systems; In Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 6818 (2008)

[PAGSS10] Konstantin Pussep, Osama Abboud, Florian Gerlach, Ralf Steinmetz, Thorsten Strufe: Adaptive Server Allocation for Peer-assisted Video-on-Demand. In: Proceedings of the 24th IEEE International Symposium on Parallel and Distributed Processing IPDPS, April 2010

[PSS09] I. Papafili, S. Soursos, G. D. Stamoulis, Improvement of BitTorrent Performance and Inter-Domain Traffic by Inserting ISP-owned Peers, 6th International Workshop on Internet Charging and QoS Technologies (ICQT'09), Aachen, Germany, May 2009

[NextShare] P2P NextShare Homepage, http://www.livinglab.eu/

[P2PNEXT] P2P-Next Project Homepage, http://www.p2p-next.org/

[PlanetLab] Planetlab – An open platform for developing, deploying, and accessing planetary-scale services, http://www.planet-lab.org/

[G-Lab] G-Lab Homepage, http://www.german-lab.de/

9 Abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, Accounting
ADSL	Asymmetric Digital Subscriber Line
AS	Autonomous System
BGP	Border Gateway Protocol
BNS	Biased Neighbor Selection
BT	BitTorrent
BOU	Biased Optimistic Unchoking
BU	Biased Unchoking
CAPEX	CAPital EXpenditures
CP	Content Provider
DPI	Deep Packet Inspection
DSL	Digital Subscriber Line
ETM	Economic Traffic Management
ETMS	Economic Traffic Management System
G2G	Give-to-Get
HAP	Highly Active Peer
IoP	ISP-owned Peer
ISP	Internet Service Provider
NGN	Next Generation Networking
NMS	Network Management System
OP	Overlay Provider
OPEX	OPerating EXpenditures
P2P	Peer-to-Peer
QoS	Quality of Service
SIS	SmoothIT Information Service
SLA	Service Level Agreement
SmoothIT	Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies
STB	Set-top Box
STREP	Specific Targeted Research Project
T4T	tit-for-tat

10 Acknowledgements

This deliverable was created not only by the authors as stated in the top of this document, but also with the help of the whole WP2 team of SmoothIT. Additionally, the authors would like to express their gratitude towards the internal SmoothIT reviewers George Stamoulis, Konstantin Pussep and Dirk Staehle, who gave valuable and helpful comments.

11 Appendix A - Simulation Support Studies for External Trial

To judge the effects that can be expected in the external trial, a simulation study has been conducted that uses its topology and setup as far as it is known. The arrival process of the G-Lab peers is changed in the simulation to have a fixed number of 10 peers all the time. Peers in the PrimeTel network stay online until they have watched the video plus an exponentially distributed seeding time, with a default mean of 1 minute. The number of peers in the PrimeTel network is fixed as well, since no better assumption on the arrival process of these peers can be made. Each peer has 6 Mbps downlink and 512Kbps uplink capacity. In the default scenario, a single seed with default Tribler behavior is placed in the PrimeTel network. It has an upload capacity of 5 Mbps.
The video distributed is quite large with 154 MB size and 22 minutes runtime. However, since no exact figures on the content used in the external trail can be given yet, it serves as a baseline for comparison.

The results show that, depending on the number of peers in the PrimeTel network, a positive effect of the BGP-Loc mechanism on the traffic as seen from the PrimeTel network should be observable, cf. Figure A1. The change in bandwidth consumption is shown for a number of 5, 10, 20 and 50 peers in the PrimeTel network. It should be noted that the absolute values of bandwidth are in the range of 100 Kbps, i.e., it will be difficult to observe these on an aggregate link.

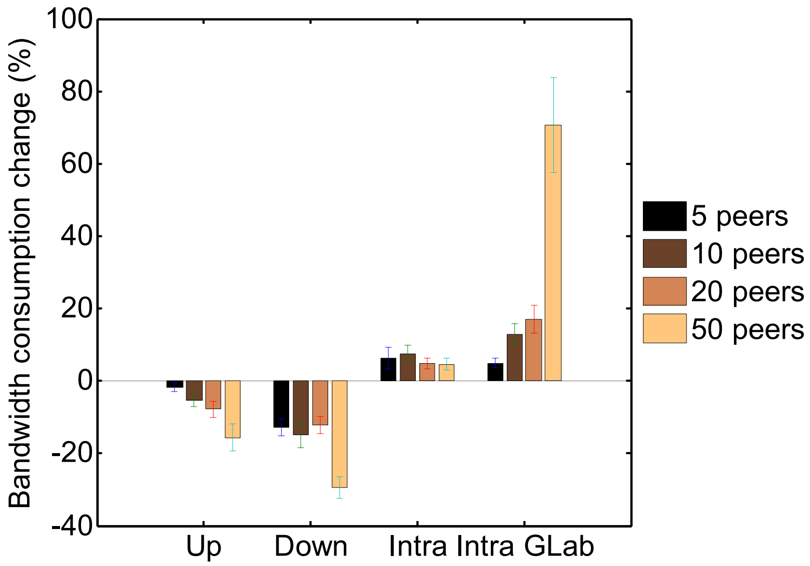


Figure A1: Bandwidth consumption change for different peer populations

Since the seeding capacity is quite low, the stalling times are very high, cf. Figure A2. This should be avoided by the addition of the support seed. As expected, more seed capacity is needed for larger peer populations, since the upload capacity of the peers is less than needed to scale with the streaming bandwidth demand for the evaluated video size.

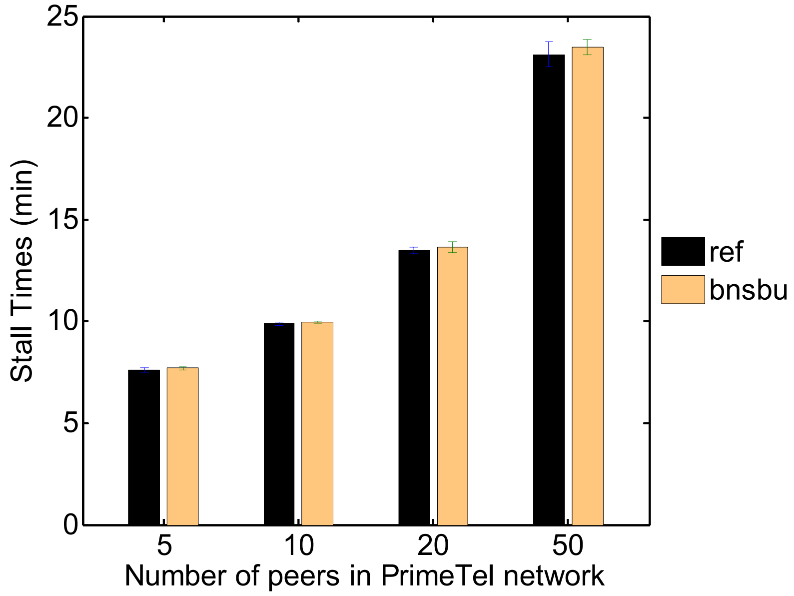


Figure A2: Stalling times for different peer populations

Longer seeding times should not decrease the effect seen by promoting locality, cf. Figure A3, and obviously have a positive effect on the stalling times, cf. Figure A4. The chosen values of 1, 10 and 30 minutes mean seeding time are sample values, since no reliable assumptions about the behavior of the PrimeTel peers can be made. A population of 10 peers in the PrimeTel network was used in this experiment.

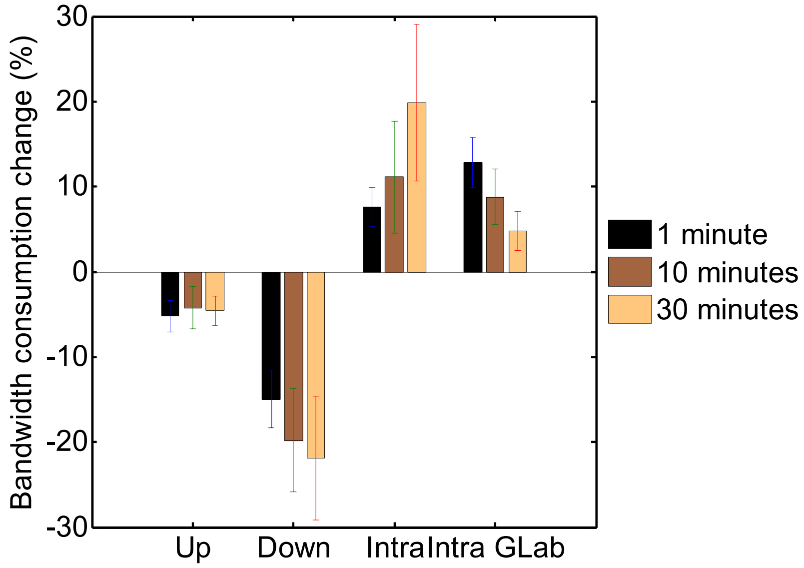


Figure A3: Bandwidth consumption change for different seeding times

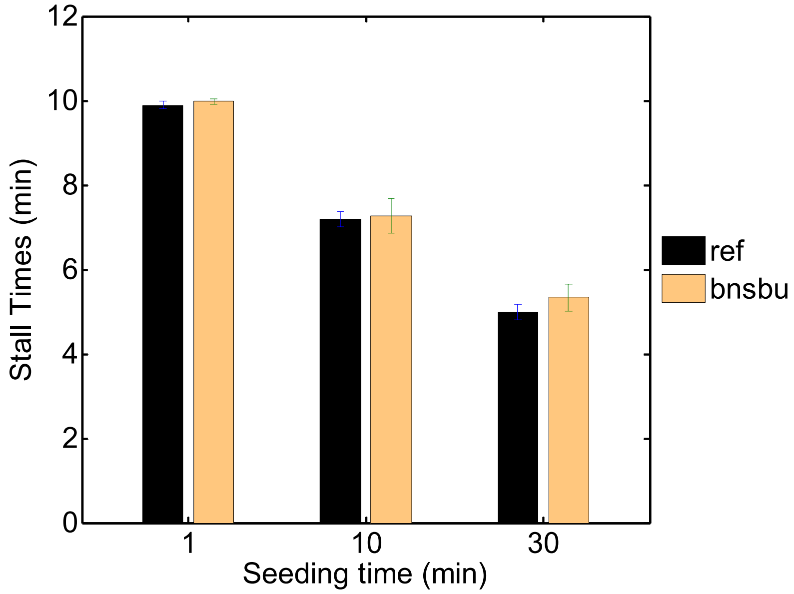


Figure A4: Stalling times for different seeding times

To judge whether the QoE can be improved enough with additional seeding capacity, we simulated a scenario with 10 PrimeTel peers and a seeding time of 10 minutes, increasing the seeding capacity to 20 Mbps. The results are shown in Figures A5 and A6. The traffic reduction is decreased, since the seeders do not promote locality in this scenario. This even leads to an increased upload. However, the stalling times of the PrimeTel peers are significantly reduced.

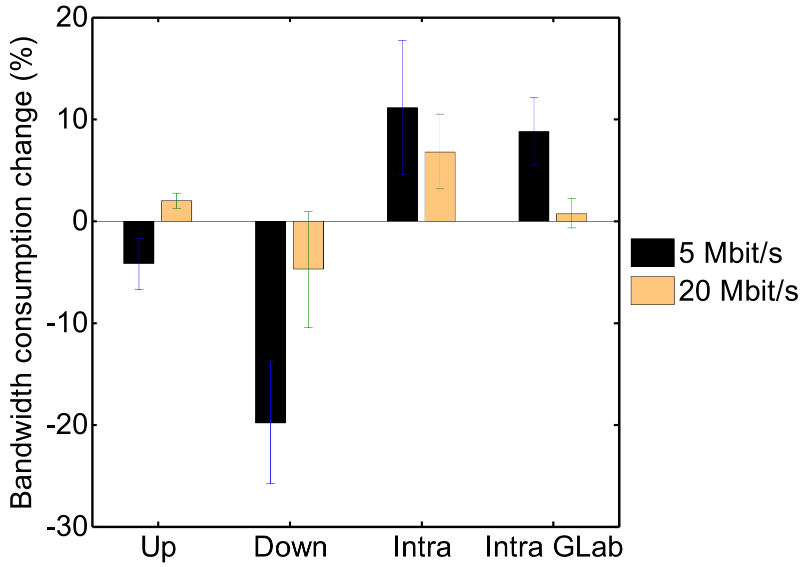


Figure A5: Bandwidth consumption change for different seed capacities

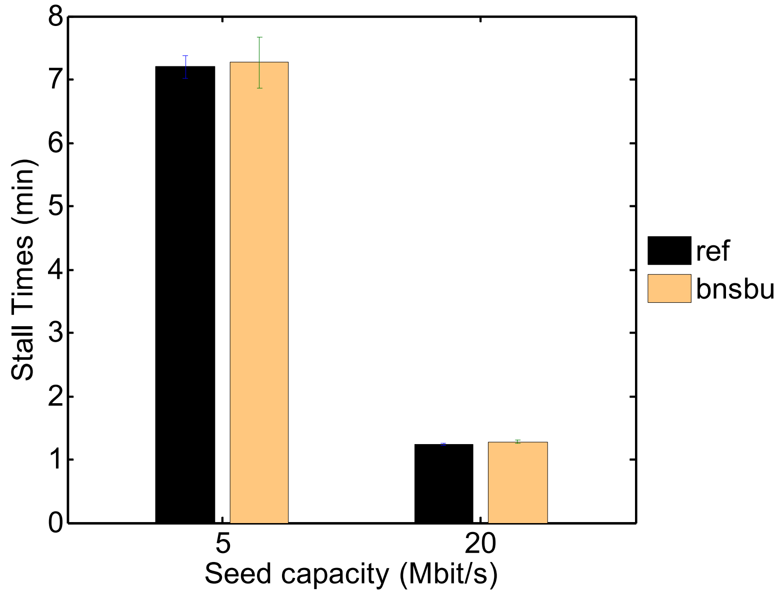


Figure A6: Stalling times for different seed capacities

It is expected that the swarm sizes in the external trial will be relatively small, leading to a reduction of the efficiency of the BNS mechanism in the BGPLoc ETM. Therefore, we evaluated whether a reduction of the standard neighbor set sizes of 40 default and 80 maximum neighbors will lead to better results. We reduced the number of default neighbors to 10 and the maximum number of neighbors to 15 in the standard scenario with 10 peers in the PrimeTel network and a seed capacity of 5 Mbps. The results, shown in Figures A7 and A8, show that this setting indeed should lead to a higher traffic reduction. On the other hand, it may influence stalling times negatively; however, this can probably be managed by the supporting seed capacity.

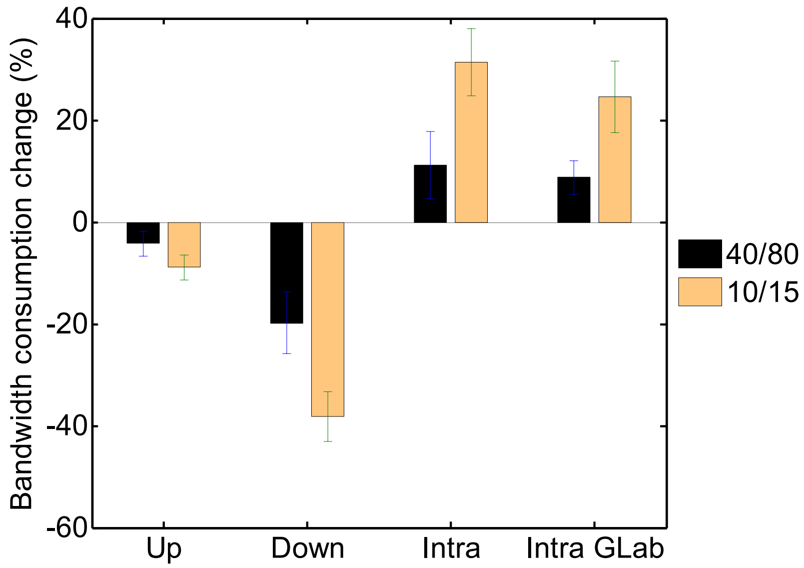


Figure A7: Bandwidth consumption change for different neighbor set sizes

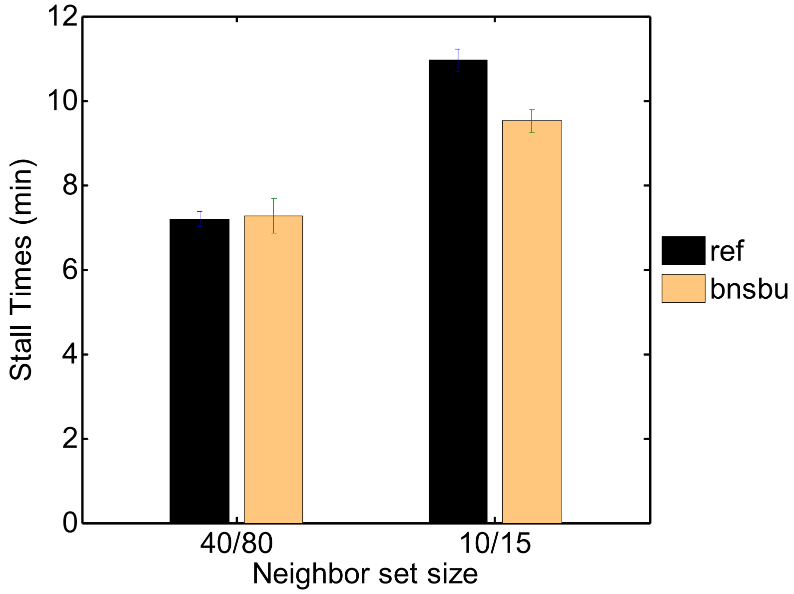


Figure A8: Stalling times for different neighbor set sizes

12 Appendix B - Adaptive Server Allocation for Peer-assisted Video-on-Demand

(This page is left blank intentionally.)

Adaptive Server Allocation for Peer-assisted Video-on-Demand

Konstantin Pussep*, Osama Abboud*, Florian Gerlach*, Ralf Steinmetz*, Thorsten Strufe§
*Multimedia Communications Lab, Technische Universität Darmstadt
Email: {pussep,abboud,steinmetz}@kom.tu-darmstadt.de
§Peer-to-Peer Networks, Technische Universität Darmstadt
Email: strufe@cs.tu-darmstadt.de

***Abstract*—Dedicated servers are an undesirable but inevitable resource in peer-assisted streaming systems. Their provision is necessary to guarantee a satisfying quality of experience to consumers, yet they cause significant, and largely avoidable cost for the provider, which can be minimized. We propose two adaptive server allocation schemes that estimate the capacity situation and service demand of the system to adaptively optimize allocated resources. Extensive simulations support the efficiency of our approach, which, without considering any prior knowledge, allows achieving a competitive performance compared to systems that are well dimensioned using global knowledge.**

I. Introduction

Peer-assisted Video-on-Demand (VoD) streaming systems are an attractive way to distribute video content through the Internet at low cost [1], [2]. They combine the scalability of Peer-to-Peer (P2P) systems, where users contribute their resources, and the service level guarantees of server-based systems.

A common solution is to organize peers in a mesh-based overlay where peers and servers contribute their upload capacity. For this, the video is split into segments that are initially downloaded from the content provider's servers, and later on redistributed by downloading peers.

A content provider utilizing a peer-assisted VoD solution aims at achieving the desired user streaming experience, while keeping the load at its servers as low as possible. Therefore, the provider should monitor the system behavior and adjust the contribution of the servers depending on the system performance.

However, such systems are highly dynamic because users arrive and depart from the system unexpectedly. Additionally, the resources of users are heterogeneous: some users have high bandwidth capacities and others quite low. This especially applies to the upload capacities, that are typically much lower than download capacities [3]. Thus, dynamic number of concurrent viewers and fluctuating capacities of peers should be alleviated by servers' contribution.

Unlike in pure server-based systems, server dimensioning in peer-assisted systems is more complex. The reason is that not only the demand but also the upload capacity of the system is dynamic. The main point is how much bandwidth can be contributed by peers and how much bandwidth must be provided by servers.

In this paper we address the issue of server allocation depending on system capacity deficit or surplus without knowing user behavior and capacities in advance. Additionally, the server bandwidth must be utilized efficiently to reduce the startup delay and stall times for video playback, which are the main quality metrics for the users.

To achieve this, we employ *adaptive server allocation policies* that allow available servers or peers with cached content to join and leave the overlay on demand. This way, servers can contribute their spare upload capacity otherwise, for example, to join other overlays with bandwidth demand, pro-actively upload some files, or even simply reduce costs if the upload is paid per volume. By minimizing the number of active servers the costs and overhead for the owners are reduced to the required minimum, while the users receive the desired quality of service as perceived in over-provisioned systems. We propose two policies, each one with a different metric to evaluate the streaming performance of the distribution overlay. Our performance evaluation analyzes the performance of the proposed adaptive policies, and further compares it with the static server allocation.

The paper is structured as follows: background on peer-assisted VoD is presented in Section II. Our system model is presented in Section III. In Section IV we present the adaptive server allocation approach and the details of the proposed policies. Performance evaluation is covered in Section V, while Section VI presents the related work. Finally, Section VII concludes the paper.

II. Background

We consider a typical scenario of a mesh-based peer-assisted VoD overlay where a content provider tries to reduce distribution costs by letting peers upload parts of the content. A media file is divided into segments and initially injected from content provider's servers. The servers upload segments to a subset of peers which exchange these segments with each other and, therefore, contribute their upload capacity. Peers exchange the information about available and desired

segments with each other and choose segments to download depending on their playback deadlines, that means, segments closer to the current playback position are prioritized. Typically, most urgent segments are managed in a high priority set whose size can range from few seconds to a minute of the video. Since the high priority set roughly corresponds to the playout buffer of the video player, peers are interested in keeping it full.

For the startup time we assume that peers start the playback once their playout buffer is filled and the remaining download time is lower than the duration of the video, to avoid playback stalling that occurs if segments close to the playback deadline are missing.

Peers are free to join and leave the overlay at any time and sometimes desired segments are not available in the neighborhood. Because of this, it is difficult to guarantee continuous segment supply for the whole download duration. *Prefetching* is a technique to smooth the streaming experience of users by downloading segments out of order. This way better segment distribution and bandwidth utilization are possible.

A special *indexing* server is utilized to track the active peers and to provide the peer lists for contact management. While the indexing server has the global view of the system, the information might be partially outdated due to the reporting intervals.

A. Give-to-Get Protocol

In order to utilize high capacity peers efficiently we utilize the Give-to-Get (G2G) system [4], a mesh-based protocol for VoD streaming. Here, uploaders prefer peers that turn out to be good forwarders. The video segments are divided into 3 sets: high, mid and low priority. The first one is the most crucial, since it corresponds to the playout buffer. Therefore, the main objective of the system is to fill this buffer fast and keep it filled during the playback, while segments from other sets are downloaded to enable prefetching.

Unlike the original G2G approach, we do not skip delayed segments of the video, but rather stall the playback until the playback buffer can be filled. This appears to be more suitable for a VoD application, since most users would prefer to see parts of the movie with delay instead of skipping them completely.

III. SYSTEM MODEL

We assume that the content provider has access to a pool of servers, that can be either run by himself or rented, such as Amazon Elastic Compute Cloud (EC2)[1]. In such a scenario the content owner typically pays for the total volume of data uploaded from its servers, while Amazon also charges for usage time. Therefore, the provider is interested to avoid unnecessary uploads if the segments can be served by other peers. Uploading too few segments will result in unsatisfied users, while too much segments uploaded by the servers will result in unnecessary high costs. Furthermore, the reduction of unnecessary server online times allows content provider to reduce their energy consumption.

In order to estimate the bandwidth supply and demand in peer-assisted streaming we use the following notation:

- S: totally available servers (passive and active)
- S': active servers (subset of S)
- u_s: upload capacity per server
- L: peers active in distribution overlay
- u_l, d_l: upload and download capacity of peer l (in homogeneous case also denoted as u and d)
- r: video bitrate ($r \leq d_l$)
- $d_r = f * r$: required download speed ($r \leq d_r \leq d$), where f is the *prefetching factor* of the system
- g = peer upload utilization factor

As already explained in Section II prefetching is required to speedup the startup time and pre-load future segments. A suitable prefetching factor can differ depending on the utilized protocol. For example, in the G2G protocol f should be ≥ 1.2 [4].

On the other hand, the peer upload utilization factor g is the ratio of the average upload speed of peers to their upload capacity. Ideally, this values should be close to 1 but peers might be not able to exploit their upload capacity due to the lack of segments required by other peers (content bottleneck). In real systems utilization values around 0.8 have been reported [5].

We can easily compute the total required upload capacity as $R_{total} = |L| \cdot d_r$ and the total available upload capacity as $U_{total} = |L| * g * u + |S| * u_s$.

For acceptable streaming performance we then need $U_{total} \geq R_{total}$ and therefore

$$|S| \geq max(|L| \cdot \frac{d_r - g \cdot u}{u_s}, 0) \qquad (1)$$

While the above model applies for homogeneous upload and download capacities, in the heterogeneous case, we obtain:

$$U_{total} = \sum_{l \in L} u_l \cdot g + \sum_{s \in S'} u_s \qquad (2)$$

$$R_{total} = \sum_{l \in L} r \cdot f = |L| \cdot f \cdot r \qquad (3)$$

Our goal is to find the minimal subset $S' \subset S$ with:

$$\sum_{s' \in S'} u_{s'} \geq max(\sum_{l \in L}(d_r - u_l \cdot g), 0) \qquad (4)$$

[1] http://aws.amazon.com/ec2/

Here the right side expresses the missing upload capacity of the peers (under assumption that they cannot provide enough upload capacity) while the left side is the total contribution of the selected active servers.

We can immediately see from Equation 4 that in systems with the upload capacity of peers being very high (say $u_l \geq \frac{r \cdot f}{g}$) the server contribution becomes marginal. They are only needed to assure content availability and catch up with fluctuations in user demand and upload supply. However, even for the upload utilization factor $g = 0.8$ and low prefetching factor $f = 1.2$ we obtain that $u_l \geq 1.2 \cdot r/0.8$; so even for a moderate video playback rates of 512 kbps the average upload capacity of peers must be almost 800 kbps.

Ideally, the servers could contribute just enough upload capacity to balance out the available bandwidth in the system. Since this situation can change any time due to departure or arrivals of peers, the content provider must allocate server bandwidth to avoid serious degradations in user experience.

IV. Server Allocation Policies

In order to provide high Quality of Experience (QoE) for the users, our system utilizes the architecture shown in Figure 1. The indexing server monitors the peers' performance (step 1), determines the required upload capacity, and allocates servers (step 2) to join the overlay (step 3) and provide the missing upload capacity to peers in the most efficient manner (step 4).

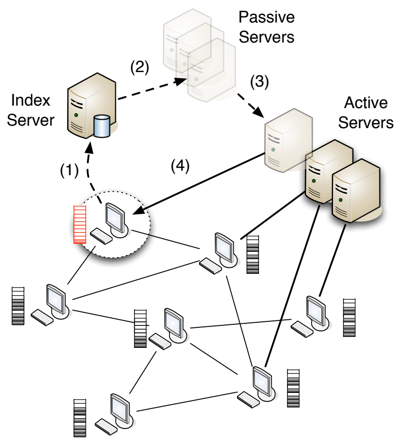


Figure 1. System Architecture.

Since users might abort the playback without watching the whole video, too aggressive prefetching might waste upload capacity. Therefore, exploiting the download capacity far beyond the video playback rate is not reasonable. Such peers can consume too much bandwidth and leave fast without contributing enough capacity in return. In order to avoid such unfair resource utilization, we limit the utilization of download capacity to a reasonable threshold, e.g, 2 or 4 times the video playback rate. On the other hand we limit the maximum upload utilization for high capacity users to the level of 2 times the playback rate.

A server allocation policy optimizes $S' \subset S$ according to the system parameters: the current number of peers $L(t)$, their upload and download bandwidth u_l and d_l, video bit rate r, and peer upload utilization factor g. These factors are measured by the indexing server, but might not be up-to-date due to the overlay dynamics. The factor f depends on the utilized streaming overlay and must be also taken into account while choosing S'.

An actual resource allocation policy is defined as $P = (M, \Delta, C)$ with the following components:

1) *Monitoring mechanism* M: this one collects the information about the overlay performance and state of single peers. This is done by the indexing server which is otherwise used for contact management. While the plain contact membership information is normally reported at a scale of several minutes, the streaming quality information must be updated more often to allow fast reactions on performance degradation.
2) *Decision metric* Δ: Different metrics can be used to decide when a server should join or leave the overlay. The candidate list includes the current server-to-peer ratio, missing upload bandwidth (difference between the total demand and the total available upload capacity), current download speed, and buffer states. We consider the (global) average download speed and playout buffer states as the most promising metrics.
3) *Connection management* C: Once a server decides to leave the overlay it can be done (1) immediately once the bandwidth excess was detected, (2) after finishing current transfers, or (3) once the connected peers are able to find new neighbors to replace the departing server. For the join procedure, the question is to which peers a new server should connect to. A possible improvement is also the decision to which peers to allocate the bandwidth.

Note that our system is managed by a provider, who controls the servers and the client software. In order to utilize the same architecture with open software and protocols, additional measures must be applied to prevent false reports from the users.

A. Global Speed Policy

The goal of this policy is to allocate minimal server resources necessary to balance the global overlay performance. To achieve this, the indexing server

monitors the *global speed* G – the average download speed across all active peers. When the global speed falls below the desired level, additional servers are allocated to the overlay. On the other hand, if the measured speed is too high some servers are removed from the overlay.

Thereby, the target speed is defined accordingly to the desired prefetching factor f' as: $G_{target} = r \cdot f'$ Note that this value is independent from the actual number of the online peers $|L(t)|$. The actual demand for additional servers can be expressed as $(G_{target} - G_{measured}) * |L(t)|$ based on the recent peer reports.

In order to achieve good accuracy, peers must report their download performance frequently. We found one report in five seconds being sufficient for an appropriate estimation of the global speed. Then the indexing server computes the average global speed $G_{measured}$ over the last five seconds, compares it with the target speed G_{target}, and allocates servers according to their difference. The indexing server performs this computations in short periods and adds or removes only one server at once, to avoid too big performance oscillations.

A server that receives an instruction to leave, waits until the transmission of current segments is finished to avoid upload of incomplete segments. Contrary, a join action can be performed immediately.

The indexing server must process $|L(t)|/5$ client reports per second, that might become a bottleneck in case of thousands of files and ten thousands of users. Another possible drawback of this solution is that some peers might experience bad performance even if the global speed is balanced.

B. Supporter Policy

While the previous policy uses the average overlay performance as a metric to allocate server upload capacity, the supporter policy concentrates on the peers experiencing bad performance. Therefore, it addresses the possible drawbacks of the previous solution in order to avoid

- bad experience for a subset of users, even if the average performance is fine and
- too frequent status reports to the indexing server from too many peers.

Instead of periodic reports, a peer sends only the *starving* status message to the indexing server in the case when it cannot download high priority segments fast enough. Once the indexing server receives a certain amount of such requests in a given time, a passive server is selected to become a *supporter* (see Figure 1).

A supporter accepts only a reduced number of neighbors, so that it can provide segments at a speed greater or equal to the playback rate. In order to supply starving peers as fast as possible, a supporter exclusively serves the peers assigned to it. As long as the supporter has free upload slots, additional suffering peers can be assigned.

A naive approach could consider peers as suffering in the stall state only. Instead we try to identify suffering peers before the actual stalling happens. Therefore, we start *watching* peers if they miss segments in the playout buffer. A peer that enters the *watched* state sends a suffering report to the indexing server. Since missing segments in the playout buffer can have a transient nature, watched peers are considered as *starving* only if they were not able to fill their buffer in the time interval, (called *suffer time*) being an important parameter of the mechanism. In the starving state a peer will be connected to active supporters that currently serve less than *maxPeers*. If no free supporter can be found, a new supporter is allocated, but only if the number of unserved starving peers reaches the *minPeers* threshold. Note that this condition also covers peers being too long in the startup phase.

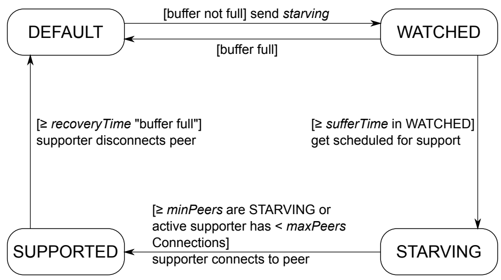


Figure 2. Downloader states in the supporter policy.

The algorithm to decide when and which peers should be considered as starving and receive help from supporters can be described as a peer state diagram with the following states (see Figure 2):

- *Default*: peer's playout buffer is full.
- *Watched*: peer is missing segments in the playout buffer recently.
- *Starving*: peer time in the watched state reaches the *sufferTime* threshold.
- *Supported*: total number of starving peers reaches the *minPeers* threshold.

Connected peers that left the suffering state and do not fall back in the *recover* time interval, are disconnected from the supporter. Finally, if all assigned peers have been served and no new peers have been assigned, the supporter leaves the overlay.

V. Performance Evaluation

In order to evaluate the performance of the proposed allocation policies, we implemented them in in an extended version of OctoSim simulator [6]. This discrete event-based simulator models data transfers at the level of file segments. As the underlying streaming

protocol, we implemented in the simulator the G2G protocol (see Section II-A). All simulation runs were repeated ten times and provide average values together with their standard deviation unless mentioned otherwise.

A. Goal and Metrics

In order to show the feasibility and benefits of the proposed approach we conduct two groups of experiments: (1) performance and sensitivity analysis of policy parameters and (2) comparison of adaptive policies with static server allocation. We expected our policies to eliminate stall times and to perform at least as good as the static allocation policy with perfect predictions of user behavior. Furthermore, the supporter policy should be able to avoid too many outliers experiencing much worse performance than average users.

We use the following QoE metrics to evaluate the performance of the streaming overlay: *startup delay* until the video can be played and *stall times* during the playback. Thereby, we apply the startup solution proposed for G2G: the playback starts after the initial playout buffer (which corresponds to the high priority set) is filled and the remaining download time plus 20% overhead is smaller than the video duration. We consider especially the 50th and 95th percentiles of the delays, expressing the maximum delays experienced by 50% or 95% of users, respectively.

B. Basic setup

The basic scenario used for the performance evaluation is shown in Table I. It models a Video-on-Demand scenario where a content provider offers short clips, such as trailers or news reports, and tries to reduce its costs and increase system scalability by deploying a peer-assisted system. The content provider can decide in advance how many servers to allocate to a particular video or use our adaptive allocation policy. The parameterization of policies is described in the respective Subsections V-C and V-D.

Table I
BASIC SETUP.

Parameter	Value
Simulation duration	30 minutes
Video length	5 minutes
Video bitrate	512 kbps
Available servers	up to 10
Server capacity (up)	2048 kbps
Peer capacity (up)	256, 512, 1024 kbps
Peer capacity (down)	2048 kbps
Peer distribution	0.3, 0.5, 0.2
Arrival rate (exp.)	6 peers/min
Playout buffer size	10 seconds
Departure rate	50% of video length on average

The peers are divided into three groups based on their upload capacities, representing slow (DSL), moderate (high-speed DSL or Cable), and fast (Ethernet) Internet connections similar to [7]. The relative size of these user groups is 0.3, 0.5, and 0.2, respectively. We also limit the maximum download capacity of peers to 2048 kbps to avoid fast peers consuming too much download bandwidth. In any case the download limit is four times the video playback rate and, therefore, sufficient even for an aggressive segment prefetching. Similarly, we limit the maximum upload capacity of Ethernet users to 1024 kbps to avoid too unfair resource utilization. Typically, even for high-end users greedy utilization of the upload capacity might result in throttling through the network operator. Each server has a upload capacity of 2048 kbps.

We model the session length of peers as follows:

$$T_{session} = min(T_{startup} + T_{playback} + T_{stall}, T_{departure})$$

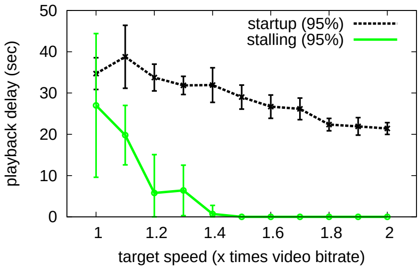
Here, users might stop the session without watching the complete video and, therefore, $T_{playback} \leq$ video length. The peers are enforced to upload the content as long as they watch it. We model the departure rate by letting peers on average abort the playback and depart from the overlay after 50% of video length after they start playing.

C. Global Speed Policy

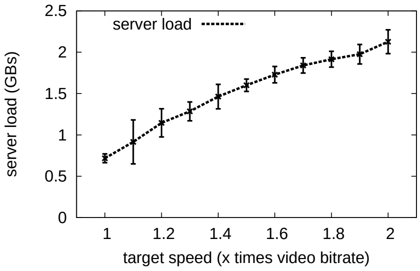
In this subsection we analyze the performance of the speed-based adaptation policy with respect to the desired *target* speed. As already explained in Section IV-A this policy tries to keep the global speed close to the desired target where the target is defined as f' times video bitrate. If f' is set too low, some nodes might experience undesired playback delays. If f' is chosen too high, too many servers will join the overlay. We expect to find a suitable value in the range $[1 : 2]$, where 1 corresponds to no capacity for prefetching and 2 allows fast prefetching.

The basic parameterization of the policy lets clients report their performance to the indexing server once every five seconds. The indexing server then calculates the average speed over the five seconds interval.

In order to find suitable values of f' in the utilized VoD system, we run a series of experiments with varying parameter values. Fig. 3 shows the 95th percentiles of startup delay and stall times (Figure 3(a)) and the relative upload contribution of servers (Figure 3(b)) for different values of f'. We observe that f' values close to 1 results in higher delays and unstable performance (high standard deviation), while keeping the server's contribution low. Here peers whose average upload rate is close to the playback rate provide most of the resources but cannot keep up with the arrival and departure rates. On the other hand, high f' result in much lower playback delays, while the server load grows significantly. We further see that f'



(a) Stall and startup delays (95th percentiles).



(b) Server load.

Figure 3. Impact of target speed factor on playback delay and upload.

equal to 1.5 is sufficient to avoid stalling, and achieve low playback delay while avoiding unnecessary load at servers.

D. Supporter Policy

For the supporter policy we are interested in understanding the impact of the following parameters:

1) *minPeers*: How many suffering peers must be present to allocate an additional server,
2) *maxPeers*: Maximum number of peers a supporter can take care of,
3) *sufferTime*: Time interval with not full playout buffer to consider a peer as suffering,
4) *recoveryTime*: Time interval with filled buffer after which a previously suffering peer is considered as recovered.

To evaluate the impact of these parameters, we fix the default configuration as follows: *minPeers* = 1, *maxPeers* = 4, *sufferTime* = 5 seconds, and *recoveryTime* = 20 seconds. Then we subsequently modify single values.

Figure 4(a) shows the impact of the minimum number of suffering peers to allocate an additional server. We observe that the median is quite insensitive regarding this parameter, while the values around 2 and 3 peers prevent too bad performance for outliers. This can be explained by the fact, that waiting for too many suffering peers results in bad performance for single peers in the suffering state.

The impact of the maximum number of peers to be handled by one supporter is presented in Figure 4(b). We can observe that the best delays are achieved if the supporter handles only 2-4 peers. Beyond 6 peers per supporter the 95th percentiles increases dramatically. At the first spot the performance increase when going from 1 to 2 peers per supporter might appear counterintuitive. However, allocating the whole supporter capacity to a single peer results in too fast prefetching of segments and, therefore, wasted upload capacity, since a peer might depart without consuming and uploading them to other peers.

Another interesting point is how long a peer should try to fill its buffer being considered as *suffering* by the the indexing server. It turns out that values around 4 seconds are optimal (see Figure 4(c)). This is roughly the half of the playout buffer size and, therefore, corresponds to our expectation that supporter should react on the suffering state before playback stalls occur.

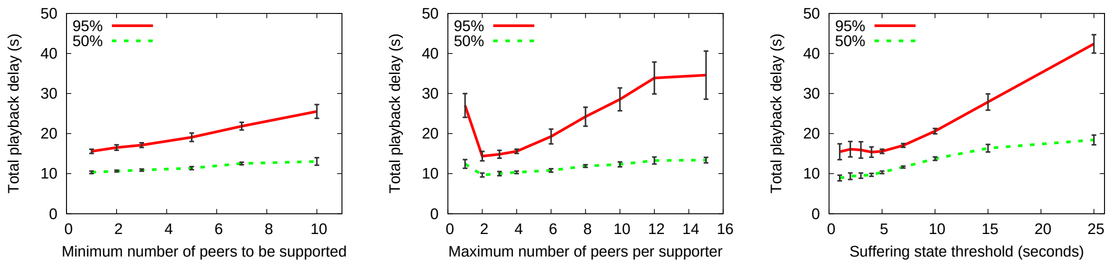
We also analyzed the impact of the recovery period with values between 1 and 20 seconds, but did not found a significant impact in our scenario. Both the server load and the startup delays were very close among the setup.

E. Comparison of Static and Adaptive Policies

In this subsection we compare the performance of adaptive policies with the static server allocation in order to see if there are static setups that can outperform them. For the static policy we let 1 to 10 servers stay in the network for the whole duration of the experiment. The adaptive policies can allocate up to 10 servers on demand. For the supporter policy we use the default parameters as specified in Subsection V-D and for the global policy the target speed factor is set to 1.5.

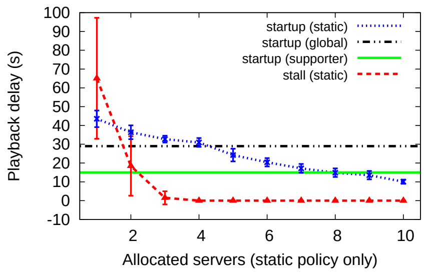
The results for the playback delays are reported in Figure 5(a). The figure shows startup and stall times for the static policy, and only startup delays for adaptive policies, since no stalling took place (standard deviations for adaptive policies are the same as in V-C and V-D). We can observe that with the static allocation, the system performs best with 4 servers, though some peers still experience playback stalling. With too few servers the delays are high while with many servers the delays go down due to bandwidth overprovisioning. We also observe that the adaptive policies exhibit comparable performance, while the supporter policy outperforms the global speed policy (50% smaller startup delay). The probable reason is that supporters upload to suffering peers in the first place, while the global speed and static policies allocate the bandwidth to random peers. Table II additionally summarizes the average values for all three policies with the optimal configuration.

In Figure 5(b) we also show the data volume uploaded by servers. We can see that in the static configuration the server contribution grows almost

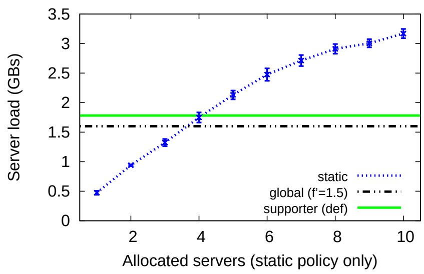


(a) minPeer threshold.
(b) maxPeers per supporter.
(c) Suffering time interval.

Figure 4. Impact of various parameters on the supporter policy performance.



(a) Stall and startup delays (95th percentiles).



(b) Data upload.

Figure 5. Comparison of static and adaptive policies.

linearly with the server count. With four static servers, they have to upload roughly the same amount of data compared with the adaptive policies. Though, the supporter policy uploads slightly more data than the global speed policy, it is able to achieve much lower startup delays (cf. Table II).

Table II
COMPARISON OF VARIOUS POLICIES (AVERAGE VALUES).

servers	server load	stalling (95%)	startup (95%)
static (4 servers)	1,75 GB	0s	31.0s
global ($f' = 1.5$)	1,60 GB	0s	29.0s
supporter (default)	1.78 GB	0s	15.5s

In summary, we observe that an adaptive policy allows a content provider to reduce costs by allocating available resources according to the overlay performance. This allows to achieve the performance of a well-dimensioned system without knowing the user demand and upload supply in advance. Even if the demand can be well estimated (here with 4 servers) the supporter policy deals better with the system dynamics and provides better streaming performance than the static and global policies.

VI. Related Work

Recent work on how to deal with temporary undercapacity in P2P systems can be classified into two categories: (1) server allocation policies and (2) alternative proposals that do not allocate additional servers.

Regarding the allocation of servers in peer-assisted systems, different proposals have been done for file sharing (mostly BitTorrent) and live streaming systems [8], [9], [10], [11]. In case of BitTorrent, Das et al. [8] propose to estimate the server demand in order to guarantee minimal download speed to users. Similarly, Rimac et al. [9] consider the dimensioning of servers for single and multiple BitTorrent swarms. In AntFarm [10] a coordinator allocates peers and servers to swarms to provide minimal service level. However, such systems concentrate only on the download speed of peers, that is not constrained by the playback positions and buffer states. Therefore, they don't deal with the issue of startup delays and stall times. Wu, Li and Zhao present a prediction algorithm to estimate server bandwidth demand for peer-assisted live streaming [11]. Their *streaming quality* metric counts the number of peers that have a buffer count $\geq$ 80%. Differently to us, they don't address the delay and startup delays explicitly. Unlike in VoD, live streaming does not deal with the issue of prefetching and its interplay with the streaming quality.

Alternative approaches to avoid undercapacity start with advanced network coding, segment scheduling, and peer-matching algorithms to improve the throughput and capacity utilization for P2P streaming, e.g, [12]. Kumar et al. [13] state that the ratio of slow and fast users determine the P2P VoD performance and propose admission control and scalable video coding to deal with system's undercapacity. Garbacki et al. [14] propose the usage of helper peers in order to avoid bad user experience. However, they

do not provide a metric how to calculate the desired number of helper peers. Huang et al. [1] analyze the impact of prefetching on peer-assisted VoD in surplus and deficit modes. They show that prefetching can significantly reduce the server load, especially when the bandwidth demand is close to the supply. Inspired by these results we propose to combine prefetching and adaptive server allocation to keep supply slightly higher than demand.

VII. Conclusion

We consider adaptive allocation of servers in peer-assisted Video-on-Demand streaming in order to avoid undercapacity and service degradation. To achieve this, we provide a simple demand model, and design two policies covering the demand monitoring, allocation decisions, and connection management for the servers. We show, by means of simulations, that adaptive policies can handle unknown user demand and provide high service level to the users while keeping the server load at a low level. On the other hand, focusing on suffering peers and supporting them preferentially can additionally improve service quality.

Acknowledgements

This work has been performed in the framework of the EU ICT Project SmoothIT (FP7- 2007-ICT-216259). The authors would like to thank all SmoothIT partners for useful discussions on the subject of the paper.

References

[1] C. Huang, J. Li, and K. W Ross, "Peer-Assisted VoD: Making Internet Video Distribution Cheap," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, 2007.

[2] Yan Huang, Tom Z.J. Fu, Dah-Ming Chiu, John C.S. Lui, and Cheng Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in *SIGCOMM*, 2008.

[3] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu, "Characterizing Residential Broadband Networks," in *Internet Measurement Conference*, 2007.

[4] J.J.D. Mol, J. A. Pouwelse, M. Meulpolder, D.H.J. Epema, and H.J. Sips, "Give-to-Get: An Algorithm for P2P Video-on-Demand," in *MMCN*, 2008.

[5] B. Li, S. Xie, Y. Qu, G. Y. Keung, C. Lin, J. Liu, and X. Zhang, "Inside the New Coolstreaming: Principles, Measurements and Performance Implications," in *INFOCOM*, 2008.

[6] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and Improving a BitTorrent Networks Performance Mechanisms," in *INFOCOM*, 2006.

[7] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?," *ACM SIGCOMM Computer Communication Review*, 2007.

[8] S. Das, S. Tewari, and L. Kleinrock, "The case for servers in a peer-to-peer world," in *IEEE International Conference on Communications*, 2006.

[9] I. Rimac, A. Elwalid, and S. Borst, "On Server Dimensioning for Hybrid P2P Content Distribution Networks," in *Peer-to-Peer Computing*, 2008.

[10] R.S. Peterson and E.G. Sirer, "Antfarm: Efficient Content Distribution with Managed Swarms," *NSDI*, 2009.

[11] C. Wu, B. Li, and S. Zhao, "Multi-channel live p2p streaming: refocusing on servers," in *INFOCOM*, 2008.

[12] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P.R. Rodriguez, "Is high-quality VoD feasible using P2P swarming?," in *WWW*, 2007.

[13] R. Kumar, Y. Liu, and K.W. Ross, "Stochastic fluid theory for P2P streaming systems," *INFOCOM*, 2007.

[14] P. Garbacki, D.H.J. Epema, J. Pouwelse, and M. Van Steen, "Offloading servers with collaborative video on demand," in *IPTPS*, 2008.

13 Appendix C – Highly Active Peer and Next Generation Networks

The main idea of this ETM mechanism is to boost overlay performance in the local AS by increasing upload capacities of selected local peers. This measure also acts as an extra incentive since increased upload capacity helps them to become better torrent community members. Such a peer becomes then a Highly Active Peer (HAP) by means of the Next Generation Networking (NGN) or by means of similar features provided by the Network Management System (NMS) of an ISP. Therefore, other peers are incited to download more from those local peers than from remote peers. Unlikely to the IoP mechanism, the content is offered by local peers running at the edge of the network and not at the ISP premises. Moreover, this mechanism aims to prevent the increased HAP capacity from being used by remote peers since this would increase inter-domain traffic and, therefore, ISP costs.

Unlike the QoS-awareness mechanism discussed in Deliverable D2.3, this mechanism does not consider SLA agreements between content providers and ISPs in the first place. Instead, the focus is on the agreements between users and ISPs. Additionally, this mechanism addresses the availability of the HAPs, since the peers should still offer the increased upload rate after they finish their downloads. Furthermore, unlike QoS-awareness, HAP is not a real-time mechanism and does not need to apply changes to customer's profile immediately, instead this can happen at longer intervals (up to the length of a day in the static case).

It is important to note that the goal of HAP ETM is twofold. On one side it tries to provide an immediate benefit to the ISP and the overlay network by providing additional local bandwidth. However, it also strives to provide a long-term benefit to all three players by providing users an incentive to change their behavior and become active seeders but also act nicely towards their ISP by promoting locality. The later goal can be achieved if the peers make use of SIS locality rating and follow the advice given by SIS. However, the HAP mechanism itself does not rely on whether peers have actually followed the recommendation of SIS or not, as this is only a mean of achieving the ultimate goal of reducing inter-domain traffic.

13.1 Scenario

The performance of peer-to-peer (P2P) overlays relies strongly on two factors: availability of the content and the upload bandwidth of participating peers. Considering the example of a peer-assisted video-on-demand streaming application, the "P2P effect" is getting stronger if some seeds are available and the total upload capacity of peers can satisfy the total download rate demand. The required download rate here must be at least as high as the video playback rate to achieve the "watch-while-you-download" behavior. Therefore, an ETM mechanism that aims at increasing the overlay performance, must address both content availability and available upload rate. This is not a trivial task since users tend to leave the swarm once they finish the download [SR06] and the upload capacity is typically sparse (e.g., 1:4 or even 1:8 for DSL connections).

From the ISP's point of view these measures only make sense if the additional upload capacity is consumed by local peers *and* the HAPs can be found (and connected to) in the local domain. These are exactly the challenges addressed by the ETM mechanism described in this section.

In this mechanism, an ISP reduces its costs and load on expensive inter-domain links by increasing the performance offered by local peers in terms of the offered bandwidth. On the other hand the mechanism also provides incentives to seed more content in the local domain, as such a compliant behavior will increase their chances of being selected as next HAP which brings certain benefits, such as increased download bandwidth (for the peer) and a chance to become better torrent community member by seeding more content via upgraded upload link (for the overlay). From a peer's perspective this can be achieved by following SIS advise and seeding popular content for longer time. HAP also offers a benefit to the whole overlay, since longer seeding time and increased bandwidth also increase the user satisfaction and can offload the initial seeds.

The resulting situation is shown in Figure 1. Here, the local part of the overlay contains two HAP candidates and several leechers. One candidate was promoted to a HAP and thus offers higher upload rates (thicker lines) than other peers. Note that a HAP can be a seeder and leecher in different swarms at the same time.

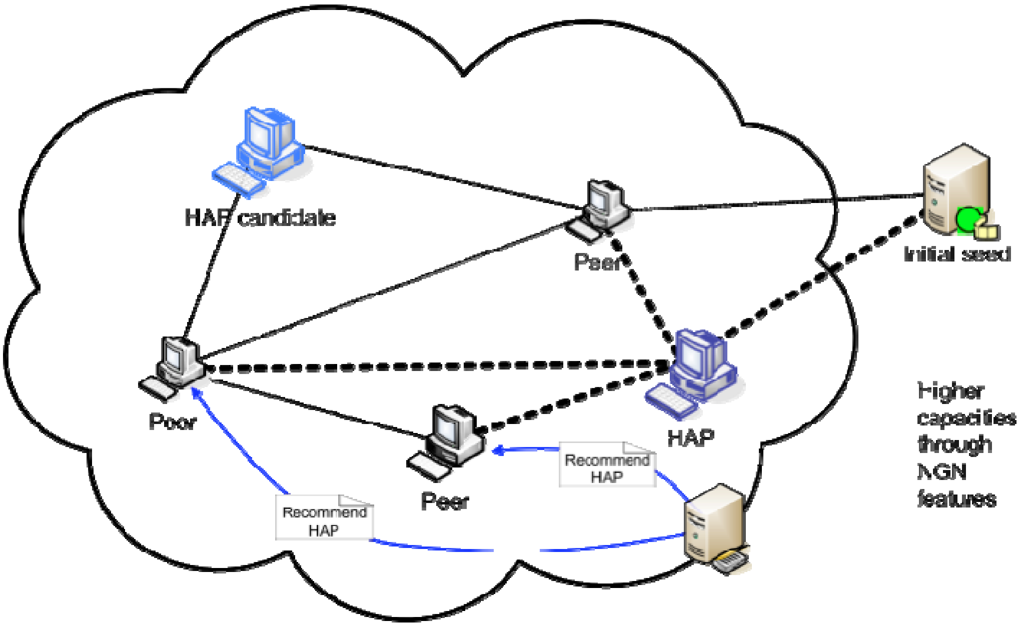


Figure 1: HAP and NGN Scenario

We expect the ETM mechanism to improve the performance of BitTorrent and similar applications (streaming applications such as the show case of SmoothIT – NextShare [P2PNext]).

In principal, any overlay relying on uploading and downloading of big amounts of data by peers can be considered. In order to benefit from the HAP mechanism, it must be possible to promote at least some of the overlay peers to HAPs. Ideally, those should be the peers that have the highest potential to serve local peers. Candidates could be peers staying online for a long time, or those sharing many popular files. The latter can be either estimated by the ISP or provided to the ISP through a suitable API (e.g. SIS-Client API, see D2.3 for details).

13.2 ETM Modes

One of the questions arising from the above description of the mechanism is whether the ETM should be applied to the overlay network as a whole or to each individual swarm in particular. The following issues are associated with this choice:

- How often can ISP change user profiles?
- Can ISP collect detailed monitoring information on per-swarm basis?
- Will peers feel any specific improvement enough to give them incentive to become HAPs?

In its basic form HAP ETM promotes peers to HAPs based on their behavior patterns in the overlay network in general irrespective of the immediate needs of a particular swarm. While it does not necessarily solve the problem of missing local bandwidth immediately, should it arise in any given swarm, it strives to change user behavior in medium to long term by giving them incentives to seed more content for longer periods of time and follow the SIS advice. This ETM is directly applicable to PrimeTel's scenario as its NGN equipment is only able to update customers profiles as often as once every 24 hours.

Dynamic extension can be applied to the Basic HAP ETM that attempts to promote peers to HAPs in those swarms, that suffer from lack of local upload bandwidth. This approach requires NGN mechanisms of the provider to be able to react in much short time scale, in the return providing solution to immediate problem of a given swarm. Long time effect on user behavior pattern might be less obvious as dynamic HAP doesn't have a direct correlation between actions of the user and his "reward".

In fact, in order to achieve a TripleWin situation, HAP should rely on one of the above described locality ETMs such as BGPLoc or dynamic locality. These are required in order to promote HAP as a more suitable peer, localize traffic and thus ensure a win situation for ISP as well.

13.3 Increased Availability of Content & Incentives

The mechanism strives to increase availability of highly demanded content on the overlay network. This requires local peers to offer the content after they finish the download and while there is some demand for it from local peers, i.e. increase "seeding time" if we apply BitTorrent's terminology and local part of seeding ratio. This can assure two important properties:

- Increase availability of the local upload capacity in the local domain: This allows local downloaders to achieve high QoE while reducing the amount of traffic exchange over expensive inter-domain links.
- Higher content availability: Even if the original seeds leave the overlay, the content stays available thanks to peers acting as edge caches and contributing their upload bandwidth to the relevant overlays.

One of the many ways to increase peers seeding time and seeding ratio, i.e. convince them to leave their clients run longer, is to provide clear and measurable incentives, which would make becoming HAP clearly beneficial for the end user. Increasing customer UL bandwidth is one such incentive as gives the chance to be a better torrent community member and thus increase the probability to be served by other peers. However, its effect is not always directly noticeable and might not be sufficient to many users, especially

inexperienced in P2P. Increase of DL link, on the other hand, is a very clear incentive and users can experience its effect immediately. It is also easily measurable and understandable by users.

Given the right ratio of additional costs, a clear triple win situation can be achieved. Peers get higher bandwidth/better QoS or some other advantage. ISP gains from more localized traffic and reduced inter-AS traffic costs. Overlay wins due to increased content availability and better seed distribution.

It is important to note that this service is not activated by user's request, but by ISP's request, as opposed to the QoS awareness ETM mechanism, where the user wants a better quality and pays something more to get it.

13.4 HAP Algorithm

13.4.1 General view

The basic operation of the mechanism is as follows:

1. Every X minutes each peer provides to SIS the summary of its activities since the time of the last report, which contains information about generated traffic to other peers, both in upload and download directions. (Note that another option would be to collect this data by the NMS means as discussed in the respective subsection).
2. Every D hours SIS aggregates peers' reports and selects those which satisfy the requirement to become HAP. *Note: The duration of the time slot D is to be determined from the simulations and is subject to technical restrictions. I.e. how often can customers' profiles be updated.*
3. SIS provides the list of peers to promote or downgrade to NMS.
4. NMS thus updates the upload and/or download bandwidth of those peers.

Optional steps if HAP is combined with BGP-Locality:

5. Clients ask SIS for peer ratings.
6. SIS assigns HAPs top rating (even higher than to local "normal" peers).

The next figure shows the sequence diagram of the above procedure while the following subsections describe the single steps and functionalities in detail.

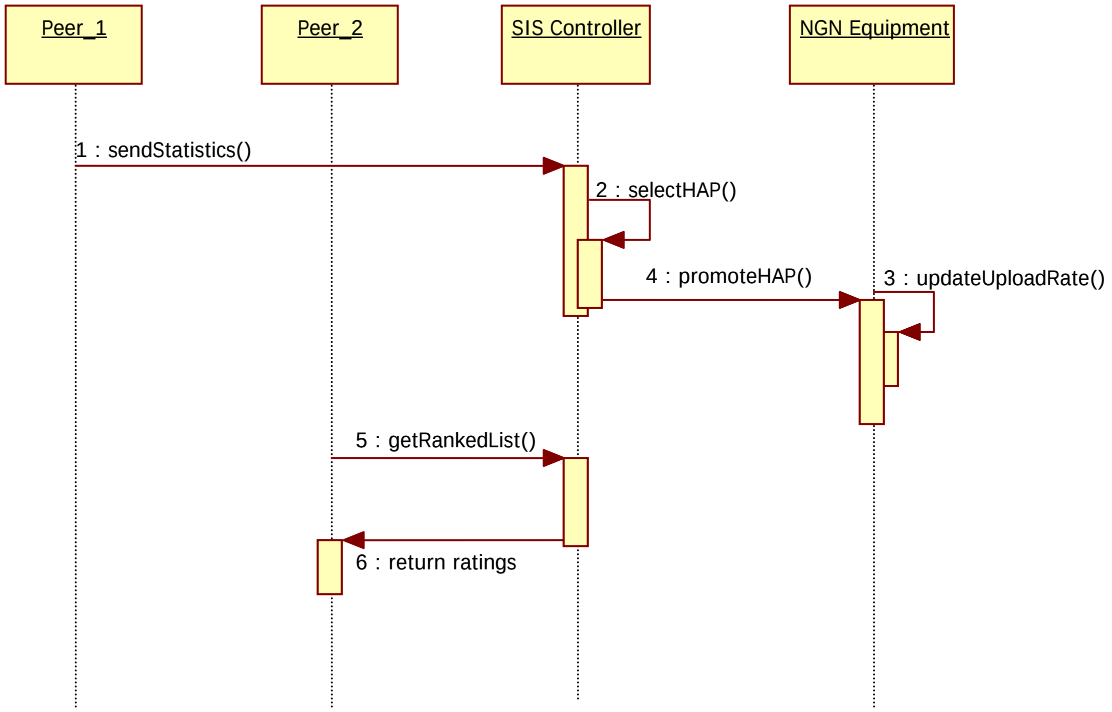


Figure 2: Sequence diagram of HAP promotion procedure (basic scenario).

13.4.2 Peer Rating

HAP selection algorithm is based on existence of a list **C** which comprises all HAP candidate peers. Prior to selection of HAP candidates, list **C** should be sorted based on parameters which define how much benefit to the local segment of overlay network this peer will provide. This relative parameter can be calculated based on history of the peer's behavior collected for different time intervals.

Another parameter which should be considered is whether the given peer had been acting as a HAP in the previous time slot. It is important to take this into account as HAP are more likely to get higher rating based on certain parameters, such as seeding ratio or local upload. Therefore, in order to be fare to all peers absolute values of local or total upload should not be taken into account but rather relatives ones with respect to available bandwidth.

The parameters considered when populating candidates list **C** are summarized bellow:

- **Local Upload L(t)** — upload traffic of the peer within the AS during time slot ***t***.
- **Total Upload T(t) —** total upload traffic of the peer during time slot ***t***.
- **Seeding Ratio S(t)** — seeding ratio measured for the duration of time slot ***t*** and aggregated for all swarms. The seeding ratio is calculated as S(t)=T(t)/(D(t) where D(t) is the data downloaded by the peer. The maximum value over all peers is denoted as S_max(t).
- **Upload Bandwidth B(t)** — the upload bandwidth of the peer accessible to it during time slot **t**.
- **Duration of time slot *d***

For a time slot ***t*** rating **R(t)** can be defined as

$$R(t) = p_1 \frac{L(t)}{dB(t)} + p_2 \frac{S(t)}{S_{max(t)}} + p_3 \frac{T(t)}{dR(t)}$$

Then the total resulting rating can be defined as

$$R = \sum_{t=0}^{D} e^{-t} R(t)$$

Here, t = 0 represents current time slot and t = 1 the previous one respectively.

This ensures that historical information is taken into account with the exponentially decaying weight. The value of D can be configured.

13.4.3 Dimensioning of HAPs

Now that we have our candidates list **C** sorted by R the top **N** peers are selected into subset **H** and promoted to HAP via NGN interface.

The upper boundary for number of peers **N** promoted to HAP in Basic ETM version depends on the amount of available resources. Let us define **AD** as download bandwidth increase and **AU** as upload bandwidth increase (both per HAP), **U'** and **D'** as total increase in upload and download respectively. Then we get the following expression for N:

$$N = Min(\frac{AD}{D'}, \frac{AU}{U'})$$

The values of ***D'*** and ***U'*** can be obtained either automatically via NMS interface or set by the ISP manually.

13.4.4 Collecting Behavior Statistics

The information about peers behavior can be collected either directly, from the clients via extensions in SIS protocol, or from ISP's metering tools. Furthermore, these two sources can complement each other thus reducing the possibility of the client providing false information with the purpose of having its link upgraded.

More specifically the ISP's metering component can provide the data on the up/down link traffic of the given peer-customer classified as either local or inter-AS. However, this information cannot normally be classified by application. And even though a typical customer would have quite small uplink utilization except for P2P applications, this metric would be very abuse-prone (malicious customers generating useless local upload). Moreover, it could lead to HAP-promotion of the peers who are not actually participating in the overlay. A further complication that is important to take into account is that many (especially large) ISPs might be reluctant to enable required monitoring procedures as they lead to significant performance degradation on the core routers.

On the other hand the client cannot provide aggregated statistics as it has no knowledge of peers classification. Therefore it would have to provide raw data on amount of

generated upload and download traffic per every peer. Aggregating this information SIS can calculate Local Upload, Total Upload, Total Download and thus the Seeding Ratio.

13.4.5 HAP Promotion

Once the set **H** of HAPs has been determined, SIS uses its interface to NMS to modify their UL/DL profiles respectively. Since a HAP is expected to boost the swarm/overlay performance in the local domain, it must receive a higher SIS rating than other peers. Additional possibility to recommend HAPs to normal peers is the planned feature of inserting additional peer addresses (here HAPs) into the SIS-rated peer list.