Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies

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

Deliverable D2.3
ETM Models and Components And Theoretical Foundations (final)

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: ETM Models and Components and Theoretical Foundations

Type: Public

Editor(s): Konstantin Pussep, Christian Gross

E-mail: Konstantin.Pussep@kom.tu-darmstadt.de, Christian.Gross@kom.tu-darmstadt.de

Author(s): Konstantin Pussep, Christian Gross, Simon Oechsner, Michael Makidis, Peter Racz, Maria Angeles Callejo Rodriguez, Maximilian Michel, Zoran Despotovic, Sergios Soursos, Eleni Agiatzidou, Ioanna Papafili, George Stamoulis, Piotr Chołda, Zbigniew Duliński, Mirosław Kantor, Rafał Stankiewicz

Doc ID: D2.3-v1.0.doc

Amendment History

Version	Date	Author	Description/Comments
V0.1	18.05.2009	Christian Gross, All	First version, Merged chapter bullet lists
V0.2	16.06.2009	Christian Gross, All	Merged second (extended) versions of sections
V0.3	30.06.2009	Christian Gross, Konstantin Pussep	Added Introduction, Exec. Summary and Conclusion, updated HAP.
V0.4	03.09.2009	Konstantin Pussep	Integrated contributions for sections 3, 4, 5, 6 and 7.
V0.5	03.09.2009	Konstantin Pussep	(Cross-) References cleanup, authors list. Fixed non-capital letters in headings, unifiead and sorted references and abbreviations.
V0.6	06.09.2009	Konstantin Pussep	Updated “executive summary”, “introduction”. Integrated section 3.4 dynamic BGP locality, updated parameter table for the IoP section. Extended “summary and conclusion”, fixed references.
V0.7	07.09.2009	Sergios Soursos, Konstantin Pussep	Updated section 8, minor update of section 3.4, fixed various layout features.
V0.8.1	09.09.2009	Christian Gross, Konstantin Pussep	Updated Acknowledgement, Introduction, Executive Summery, and Conclusion, final cleanup for the internal review
V0.8.2	13.09.2009	Marian Callejo	Internal review, part 1
V0.8.3	15.09.2009	Sergey Kuleshov	Internal review, part 2
V0.8.4	19.09.2009	George D. Stamoulis	Internal review, part 3
V0.8.5	25.09.2009	Konstantin Pussep, Christian Gross	Integrated reviewers’ feedback, merged the updates from partners for all sections.
V0.9	02.10.2009	Christian Gross	Spellchecking and update of tables in section 4.1.2
V0.91	06.10.2009	Sergey Kuleshov	Update HAP chapter
V1.0	14.10.2009	Konstantin Pussep	Final content and layout updates.

Legal Notices

The information in this document is subject to change without notice.

The Members of the SmoothIT Consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the SmoothIT Consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Table of Contents

(This page is left blank intentionally.)

1 Executive Summary

The purpose of this deliverable is to provide the final specifications of the Economic Traffic Management (ETM) mechanisms developed within the SmoothIT project. This document makes a selection of the most promising algorithms from the initial set presented in Deliverable 2.2 [D2.2], refines and specifies them in detail. This deliverable also presents the theory and modelling results obtained so far for the ETM analysis. The presented mechanisms and specifications will be used as input to Work Package 3 (WP3) where the proposed approaches will be implemented in the SmoothIT prototype, and to Task T2.4 where they will be further analysed through simulations.

An important objective of this document is to provide a theoretical view of the ETM mechanism, without covering pure implementation details that are part of WP3's architecture design. For this reason, this document analyses ETM mechanisms also from the point of view of their architectural implications and their combination opportunities.

The most comprehensive ETM mechanism is the SIS-enabled locality-awareness based on Border Gateway Protocol (BGP) data, since it is the first approach the SmoothIT project focuses on. This deliverable also presents preliminary simulation results showing that BGP-based locality-awareness can shift a significant amount of traffic from inter-domain to intra-domain links. They further highlight the necessity of a proper support for locality in the overlay application itself, since only a reasonable utilization of the BGP-based hints can lead to the desired traffic pattern. Furthermore, the ratio of overlay peers that are aware of the locality mechanism in use have strong impact on the efficiency of this approach. Future analysis is needed to assess the impact on different Internet Service Provider (ISP) topologies. Additionally, the deliverable analyses an extension of the mechanism to cope with the dynamic load of the inter-domain links since, as shown by simulation, the usage of locality-aware mechanisms is more efficient for the end user in the case of congested interconnection links.

Further investigations target the possibility to exchange information between different SIS (SmoothIT Information Service) instances. This can help to cope with the information asymmetry in local and remote SIS instances, for example regarding the BGP routes preferred by remote ISPs and the bandwidth profile of remote peers. Different scenarios for Inter-SIS cooperation are considered, covering different kinds of inter-ISP relationships (peering vs. transit and source ISPs).

The three ETM mechanisms: ISP-owned Peer (IoP), QoS-awareness, and Highly Active Peers (HAP) target the improvement of the user experience in an ISP-friendly manner according to the Honey-Pot scenario (as defined in Deliverable D3.1 [D3.1]). Here the ISP offers a higher QoS to users that behave in an ISP-friendly manner. The mechanisms can be combined with the original SIS-enabled locality to achieve the TripleWin situation in a more reliable way. It must be noticed, however, that the specifications of these mechanisms are less mature (cf. SIS-enabled locality) but the ongoing evaluations will help to choose the most promising approach for the final SmoothIT showcase.

The rest of this deliverable is structured as follows: Section 3 introduces the SIS-enabled locality-awareness followed by Inter-SIS mechanisms in Section 4. Afterwards, the concept of the ISP-owned Peer is presented in Section 5, whereas Section 6 addresses Quality-of-Service awareness. How to use dynamic network capabilities in order to realize the Highly Active Peer mechanism is explained in Section 7. Section 8 deals with Theory and Modelling of ETM mechanisms, followed by the summary section. The Appendix presents additional details for the SIS-enabled locality-awareness mechanism, detailed simulation results of this mechanism, the details of the Markov chain evolution (used in Subsection 8.2), and finally the download BGP-rating algorithm.

2 Introduction

The SmoothIT project aims at defining, developing, and testing Economic Traffic Management (ETM) mechanisms to optimize the traffic impact of overlay applications on underlay networks in order to accomplish a TripleWin situation such that the network operators, overlay providers, and application users benefit from the approaches undertaken. This deliverable presents the results of the Tasks 2.1, 2.2 and 2.3 of Work Package 2 (WP2) that had the objectives of giving an overview of Self-Organizing Mechanisms for ETM, specifying models for different ETMs and defining the theoretical foundations for the Economic Traffic Management (ETM) approaches.

Deliverable D2.2 [D2.2] has already proposed a multitude of approaches for ETM in overlays. Five approaches have been selected for further study and specification. These are presented in this document; in particular:

Section 3 provides the definition of the *SIS-enabled locality-awareness* mechanism, together with the more general *Generic Peer Rating Module,* which implements this mechanism. This ETM mechanism uses BGP information in order to rate remote peers. A complete algorithm for the SIS itself is described, as well as adaptations on the client-side that can utilize the locality information provided by the SIS in a way that would be beneficial for both clients and ISP. In addition, evaluation results of this ETM are presented, which are derived from the simulations described in Appendix B. The performance evaluation shows a general potential for savings in inter-domain traffic (and consequently in the associated costs) as well as for download time reductions in cases where there exist inter-domain bottleneck links.

The *Inter-SIS mechanism* (see Section 4) is an ETM approach that aims to minimize expensive transit links usage by introducing co-operation between different SIS-servers (such as exchanging parameter and topology information). Thus, the range of operation of ETM mechanisms can be extended covering multiple Autonomous Systems (ASes). The proposed extensions cover the scenarios where the upload and download routes between two peers are different (route asymmetry), and the scenarios where a local SIS (that receives a query from a local peer) requires additional information to rate remote peers. In such a situation the local SIS can access the SIS responsible for the remote peers to obtain more detailed underlay information. Different models for Inter-SIS cooperation are considered, depending on the business relationship among the involved ISPs.

ISP-owned Peer (IoP) is another ETM mechanism introduced by WP2. It aims at increasing the level of traffic locality within an ISP and at improving the performance experienced by users of Peer-to-Peer (P2P) applications (see Section 5). The IoP is a regular peer running the overlay protocol, e.g. BitTorrent, which is controlled by the ISP itself and which is equipped with high upload capacities. Thus, other peers prefer downloading content from the IoP instead from peers outside the AS, which gives rise to the aforementioned improvements.

QoS-awareness mechanism (Section 6) utilizes the Quality of Service (QoS) capabilities of Next Generation Networks (NGN) to guarantee certain Quality of Service levels within an AS for a specific overlay application. By doing so a TripleWin situation is accomplished as the ISP makes additional revenue by selling Service Level Agreements (SLA) to content / overlay providers, while the overlay providers are able to provide better services and the users experience better Quality of Experience (QoE).

Section 7 presents the *Highly Active Peer (HAP)* approach that is similar to the concept of the IoP but moves the functionality from the core (IoPs run by an ISP) to the edge of network (user premises). Based on swarm statistics the ISP selects peers to which a higher upload bandwidth is allocated and, therefore, the corresponding peers become HAPs. Given the higher upload bandwidth the HAP is offering, other peers participating in the overlay and running the overlay protocol tend to download content from the HAPs as this decreases their download time. The HAP approach differs from the IoP mechanism as the HAP stores content in a distributed way whereas the IoP includes consolidated storage capabilities.

All the ETM mechanisms mentioned above are shown in an abstract representation in Figure 1. As one can see, most ETM mechanisms consist of a client-side and a server-side component. For example, in the HAP mechanism, on one hand the SIS server must be aware of which peers should become HAPs, while on the other hand the client must support HAP capabilities.

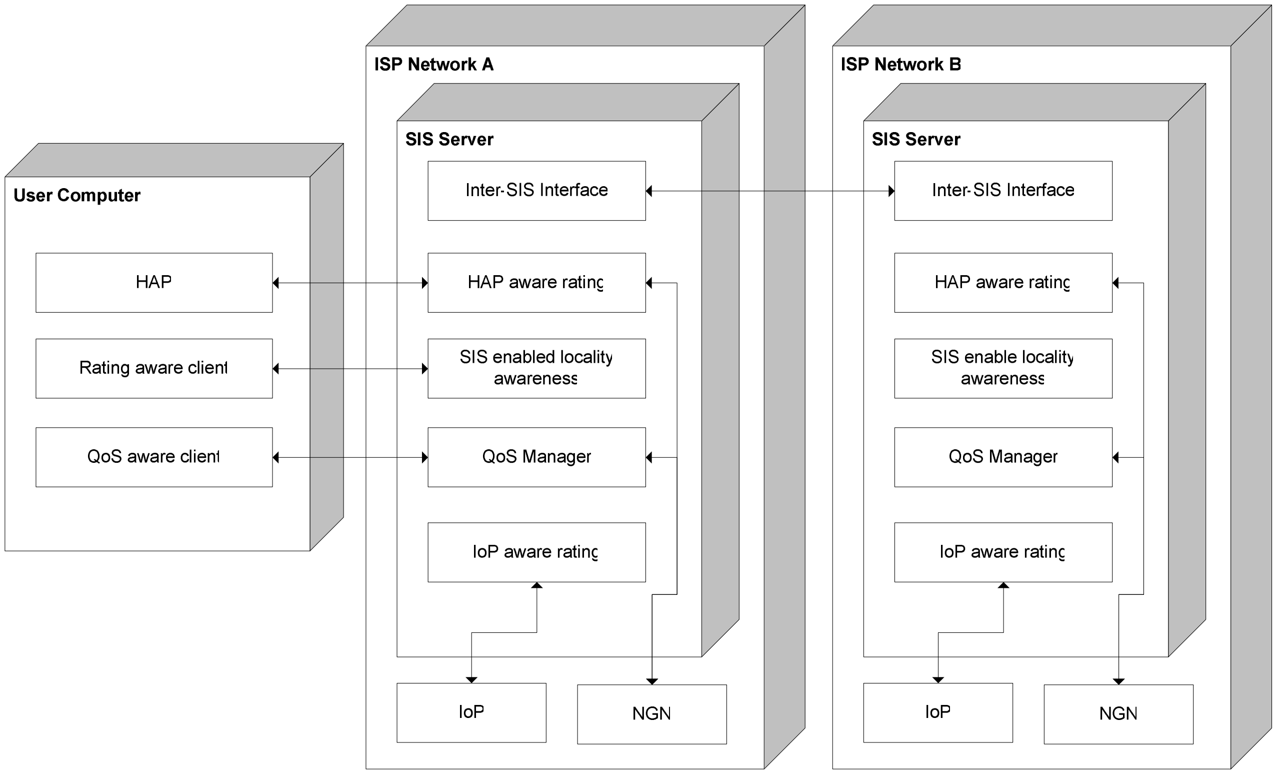

Figure 1: SmoothIT Information Service Architecture Overview

Other components, such as the IoP, are directly located in the ISP network in order to ensure high bandwidth link. Using Inter-SIS communication, different SIS servers are able to co-operate in a mutually beneficial way, e.g. exchange information about peers located in different ISP networks.

Different ETM approaches can be combined, thus leading to an increased overall efficiency. For example, the IoP or the HAP approaches can be extended to use BGP locality information to prefer uploading content only to local peers instead of distributing their upload capabilities to all the peers in the swarm. Another example of collaboration is the exchange of ETM-specific information between different SIS servers through the Inter-SIS interface in case two ISPs have a peering agreement. This way, the functionally of an ETM mechanism can be extended to cover multiple AS system.

Furthermore, this deliverable presents the work on modelling and theoretical foundations for ETM performed in Task T2.3. The corresponding Section 8 covers locality games between ISPs participating in the same overlays and then presents the update of the Markov models applied to the IoP ETM mechanism. These results refine the work presented in Deliverable D2.2 [D2.2] Section 10 and will compare different ETM approaches.

Additional information is presented in the Appendix of this document. Here the BGP-based computation of SIS ratings (required for the SIS-enabled locality algorithm) is presented. Furthermore, current simulation results for the BGP-based locality aware ETM mechanism as well as the evolution of the Markov model that will be discussed in Section 8.2 are presented. Finally, the Appendix includes a detailed description of the download BGP-rating algorithm.

To summarize, this deliverable presents the theoretical foundations and the final specification of the ETM mechanisms for the SmoothIT Information Service as well as the theoretical results so far on related models.

3 SIS-enabled Locality-awareness

This section describes the first fully specified ETM mechanism. It enables peers querying the SIS to promote locality by promoting peers that are close in terms of BGP-routes, which are configured considering the interconnection agreements between ISPs. Both the algorithm implementing this on the SIS server side and the intended usage on the client side are covered.

As a first step, the generic framework is defined that can accept any ETM rating algorithm in order to provide an easily extendable architecture. A simple binary rating is then presented as the simplest possible implementation of the SIS functionality.

The BGP-based rating is then introduced as a concrete realization of the generic one. Apart from the main BGP-based locality promotion mechanism, an extension with adaptive locality promotion is presented. Finally, the client side functionality is presented along with some simulation results to provide first insight into the effectiveness of the proposed mechanism.

3.1 Generic Peer Rating Module

This generic peer-rating module belongs to the Controller Component (cf. D3.2) and provides a framework to be used by any peer-rating algorithm defined in this project. This peer-rating module requires as input a list of IP addresses of candidate peers (given by a local peer) and information about ISP network status and business parameters (interconnection agreements, levels of charging, etc.), obtained through appropriate interfaces to the ISP systems. As an output, it provides a rated list, based on criteria defined by a specific rating algorithm. The rating value assigned to each IP is called *SIS Preference Value* (SPV). Optionally, it also provides rules to be configured in the ISP network and business management systems for deploying policies to control user-generated P2P traffic, in an incentive-compatible manner.

The main concept behind the framework is the abstraction of the optimization criteria that can be changed according to the requirements of the SIS provider typically being an ISP. These criteria are now independent of any underlying mechanism and the framework makes it possible to re-use any conforming algorithm in different ISP environments. In this way, the interfaces within the SIS can be well-defined in a relatively generic and stable fashion. The abstraction also helps to clearly define the principal concepts (e.g., cost, network distance) that must be optimised.

The main purposes of this module is a) to provide a framework for peer-rating algorithms used by the various ETM mechanisms, b) to provide abstractions to eliminate technology/ETM mechanism dependence and c) to provide guidance in implementing any ETM mechanism using a peer-rating algorithm.

3.1.1 Scenario

The generic peer rating algorithm requires a list of IP addresses as input from the local peer. This list represents candidate peers to be rated that had been produced by the traditional peer discovery algorithm by the local peer. The algorithm involves the communication between the SIS Client (local peer) and the SIS Server. The local peer sends a list of IP addresses (candidate peers) and asks the SIS Server to rate them. The SIS Server queries the responsible components for network and business data (e.g., the local DB and monitoring equipment) regarding every IP address on the list and calculates a rating or preference value. The result is a list of IP addresses with associated preference values (SPV). Based on the SPV, the local peer adapts its peer selection algorithm accordingly (see Section 3.5).

Note that, apart from the use of rated IPs by a specific peer to bias the neighbor selection and/or unchoking procedure, other (future) ETM mechanisms can define that the peer rating module provides input to other components. One such example could apply in the case of the QoS Manager component, where the policy for the local peers' flows could be adjusted accordingly.

3.1.2 Architecture

The proposed module runs inside the SIS server, as described in [D3.2]. Here, the main intelligence is included in the SIS Controller component of the SIS Server where the calculation of the preference values takes place. The Metering and Inter-SIS components are responsible for gathering all the required ISP network and business information and for aggregating them into abstract parameters. Such information may include hops, costs and QoS information. This information is provided in a structured way back to the Controller component. The Controller rates the lists and passes that rating to the QoS component, which enforces the QoS characteristics for the flows of the client.

3.1.3 Algorithm

The main task of the generic ETM algorithm is to rate a list of candidate peers. The flow diagram of this algorithm is shown in Figure 2. The algorithm takes each IP address from the list and assigns a preference value to it. The SPV is a result of the combination of the parameters' values and the parameter-related weights. The former are provided to the SIS controller by other components, such as the Metering component, while the latter is defined by the ISP and stored in the SIS DB component. These weights define how important each of the parameters is for the calculation of the final rating value of a given IP.

The algorithm runs within the SIS Controller and works as follows:

1. Get the set S containing n candidate peer addresses, $IP_1 .. IP_n$
2. $\forall IP_j \in S$
   a. Get the network and business parameters p_i from the Metering and/or the Inter-SIS component.
   b. Calculate (or retrieve from SIS DB) weights w_i for each parameter p_i.
   c. Calculate rating as: $SPV(IP) = f(w_i, p_i(IP))$, $\forall i$.
3. Return rated list of candidate peer IP addresses to the SIS Client.
4. Feed rated list to the QoS Manager. The QoS manager then computes policies for these flows as: $policy(IP, SPV(IP)) = g(w_i, p_i(IP))$.

The functions $f(w_i, p_i(IP))$ and $g(w_i, p_i(IP))$ are specific to the implemented ETM mechanism.

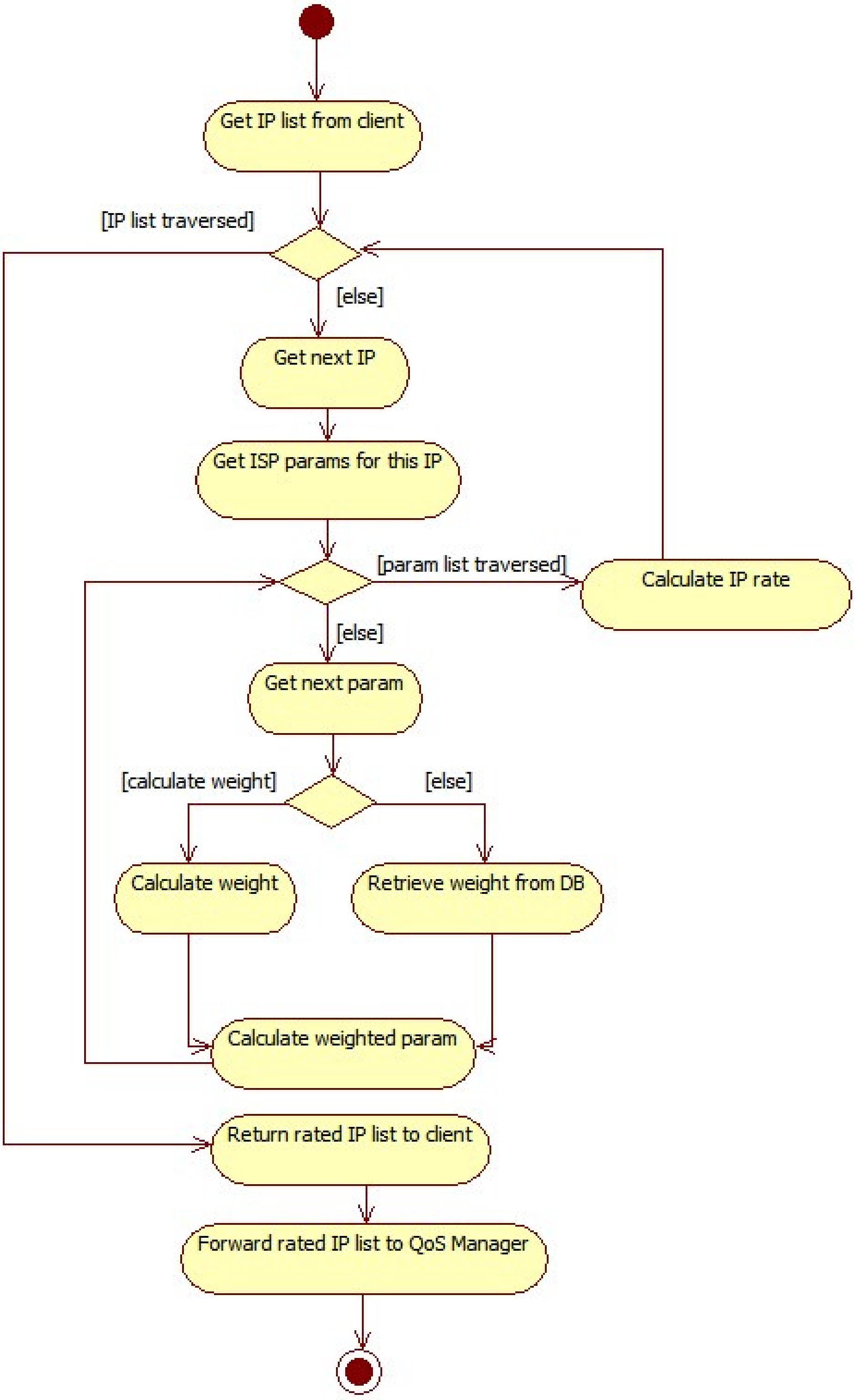

Figure 2: The activity diagram of the generic peer rating algorithm

3.1.4 Parameters

The parameters retrieved in step 2.a of the generic rating algorithm include technical/network parameters (such as proximity) as well as business/market parameters (such as link costs, link policies, user rates etc.). However, the actual values of those parameters are not used. Instead the real values are mapped to some abstracted/normalized values. The latter are going to be used by the generic rating algorithm for the calculation of the rating values. This allows for easy implementation and evaluation of different peer-rating ETM algorithms that are fairly independent of the underlying technology (and as a result, applicable to as many cases as possible). In this way, algorithms that use different parameters may be implemented without changing the normalized values. These parameters are presented in Table 1, which contains a representative and broadly usable set of numerical values for these parameters, which however is not unique.

Table 1: Rating parameters

Parameter	**Semantic Values**	**Corresponding Numerical values**	**Can be measured by**
Peer locality	Local, non-local	1, 0	IP prefix
Topological peer proximity	Far, Near, Local	1, 2, 3	Hops, destination IP subnet etc.
Physical peer proximity (estimated)	Far, Near, Local	1, 2, 3	Destination IP subnet (and a geolocation service)
Transit cost	Very High, High, Medium, Low, Very Low	1, 2, 3, 4, 5	Routing parameters, admin provided parameters etc.
Peering cost	Very Low, Low, Medium, High, Very High	1, 2, 3, 4, 5	Routing parameters, admin provided parameters etc.
BGP route preference	Very Low, Low, Medium, High, Very High	1, 2, 3, 4, 5	Rating based on BGP routing preference.
Route preference (other, non-economic reasons)	Very High, High, Medium, Low, Very Low	1, 2, 3, 4, 5	LOCAL_PREF, admin parameter etc.
External route preference (economic or non-economic)	Very High, High, Medium, Low, Very Low	1, 2, 3, 4, 5	MED, other communicated info etc.
Inter-domain/ intra-domain link usage	Very High, High, Medium, Low, Very Low	1, 2, 3, 4, 5	Router statistics
Flow throughput	Very Low, Low, Medium, High, Very High	1, 2, 3, 4, 5	Throughput from border routers (per flow)
Flow delay	Very High, High, Medium, Low, Very Low	1, 2, 3, 4, 5	One-way or RTT delay from routers
Current charging level	Very Low, Low, Medium, High, Very High	1, 2, 3, 4, 5	Current 95th percentile level (medium level could represent target charging level)
Traffic cap exceeded	Exceeded, not exceeded	Boolean (0 = False, 1 = True)	Router statistics, ISP accounting system

The above parameters are compatible with the ones defined in section 5 of IEFT ALTO group requirements [ALTO]. The numerical values provided in the previous table are simply examples and do not define the necessary granularity. In order to keep the algorithm as efficient as possible, coarse granularity may be preferred but any granularity is acceptable. For example, the "Flow delay" parameter values might be in the range [1, 1000] or even [1, 1000000]. However, this should be carefully selected to make sure that it actually provides meaningful information to the algorithm. If this is not the case, the parameter values should be summarized by the Metering component in order to offer meaningful values only.

Note that all parameters provided by the Metering component should have abstract values. If an ETM algorithm requires a concrete value (e.g., the amount of transmitted data) then this algorithm is not abstract enough and it will not be applicable to all ISP environments. Such an algorithm should be converted to use abstract values that can be derived from concrete ones either automatically (based on the automatic metering of ISP environment values) or manually (e.g., configured by the administrator).

Many of the above parameters may be estimated. The SIS is not an admission-control system and it may not know the ISP network status in real-time. All ETM algorithms that use these parameters must be designed so that they take these constraints into account.

3.2 Simple Binary Rating

One of the simplest methods to implement the SIS server functionality is to separate the list sent to the SIS by peer *P* into two sub-lists: one containing the peers in the same AS as *P*, and one containing the rest. This means the SPV range consists only of two values: 1 and 0, for local and remote peers respectively.

This can be seen as a simplification of the BGPLoc algorithm (cf. Section 3.3), where only the AS hops are considered in determining the final peer rating. A peer with the highest SIS preference value always denotes a local peer, hop counts larger than 1 and therefore lower preference values would be interpreted as remote peers. Therefore, only the Peer locality (p_1) attribute described in the next section is used as a rating value.

3.3 BGP-based Locality

The *BGP-based locality promotion* ETM approach is an instantiation of the generic peer rating. The approach uses the BGP routing information available to the ISP in order to provide locality information to overlay applications and to rate potential peers located in other ISPs according to the routing preferences of the local ISP. Since BGP provides routing information for inter-AS communication, it can be used to differentiate potential peers of an overlay application that are located outside of the AS of the ISP. Intra-AS peers cannot be differentiated by this mechanism and thus they are equally rated.

The BGP routing information represents the preferred routes for all destinations from the point of view of an ISP. Therefore, this information can be used to rate peers according to the preference of the ISP. In order to use this service, the peer sends a list of IP addresses as input to the local SIS service. These IP addresses represent potential peers that the application would connect to or exchange data with. The service performs the peer rating and sends back the list of IP addresses with an SPV assigned to each address. Based on the rating, the peer can adapt its peer selection algorithm and/or its unchoking procedure. This mechanism is equally suitable for P2P file sharing and video streaming applications since it does not affect chunk selection strategies.

The main goals of this mechanism are:

- To keep traffic as local as possible.
- To select connections (peers) with shorter AS path length.
- To prioritize usage of inter-AS links according to ISP's preferences.
- To reduce inbound/outbound traffic and respective expenses
- To improve end user's Quality-of-Experience, i.e. downloading time for file sharing or provide a smoother playout for video streaming

3.3.1 Scenario

The scenario is the same as described for the generic algorithm in 3.1.1 If the list of peers to be rated contains a large number of potential peers and the peer has to select a subset of the peers in this list due to overlay mechanisms, then this approach is expected to optimize the performance of the application.

3.3.2 Details

The mechanism involves the communication between the SIS client (peer) and the SIS server. Again, this follows the basic algorithm outlined in 3.1. The peer sends a list of IP addresses (remote peers) and asks the SIS to rate the list. In this specific ETM, the SIS queries the BGP information module for every IP address on the list and calculates a preference value. The result of this is a rated list of IP addresses with an SPV attached to each address. Based on the rating, the overlay application (SIS client) adapts its behavior accordingly, as described in more detail in Section 3.5.

3.3.2.1 Involved Components

The main intelligence is included in the Controller Component where the calculation of the preference values takes place. The Metering component and more precisely the BGP information subcomponent is responsible for gathering all the required BGP information, calculating the rating parameters (RPs) and mapping them to the more abstract parameters of the generic peer rating algorithm. The BGP information includes the *AS_PATH_LENGTH*, *MED* (Multi Exit Discriminator) and *LOCAL_PREF* values provided by the routing table. After the mapping, the parameters are passed to the SIS Controller module and more specifically to the SIS rating module, where the final rating takes place. The rated list along with the SPVs is returned to the SIS Client (peer).

3.3.2.2 Algorithm

The main task of the SIS server is to rate a list of peers. The original BGP rating algorithm was initially presented in D2.3 and an update is included in Appendix A. In this subsection, for clarity reasons, only the mapping of the BGP rating values to the generic rating values is included. There are three different approaches for adapting the original BGP-rating algorithm to the abstraction requirements imposed by the specification of the generic peer rating algorithm (cf. Section 3.1).

The first and simpler approach would be to keep the original algorithm as it is and map its output values (denoted as *out*) to a single rating parameter with associated weight of 1 (since we only deal with one parameter). The most appropriate parameter to do so is the *Route preference parameter*. The value of this parameter will be calculated in the Metering component according to the following formulae:

Table 2: Mapping of BGP info to rating parameters (alternative no.1)

Parameter	Value
BGP Route preference	• Local (3), if *out* = *(MAXPREF+1)*(MAXAS+1)*(MAXMED+1)* when MED flag is set, or *out* = *(MAXPREF+1)*(MAXAS+1)* if MED flag is not set. • Near (2), if *out* ≥ *MAXPREF/2*(MAXAS+1)* (MAXMED+1) + (MAXAS/2)*(MAXMED+1) + MAXMED/2* when MED flag is set, or *out* ≥ *MAXPREF/2*(MAXAS+1) + MAXAS/2* if MED flag is not set. • Far (1), otherwise.

Note that first the *out* parameter is calculated for each IP, based on the original BGP rating algorithm (cf. Appendix A) and then the mapping occurs. The calculation of the final SPV is actually the same with the value of the rating parameter since the latter is multiplied by a weight equal to 1.

The second alternative is to use three (3) rating parameters, the *Route preference*, the *Physical Peer proximity* and the *External Route preference* parameter and apply categorization, like the one included in the following table:

Table 3: Mapping of BGP info to rating parameters (alternative no.2)

Parameter	Value
Route preference (p_1)	• Very Low (1), if 0 ≤ LOCAL_PREF < 50 • Low (2), if 50 ≤ LOCAL_PREF < 100 • Medium (3), if 100 ≤ LOCAL_PREF < 150 • High (4), if 150 ≤ LOCAL_PREF < 200 • Very High (5), if LOCAL_PREF ≥ 200
Physical Peer proximity (p_2)	• Far (1), if AS_PATH_LENGTH ≥ 5 • Near (2), if 5 < AS_PATH_LENGTH ≤ 1 • Local (3), if address is local
External Route preference (p_3)	• Very Low (1), if MED ≥ 200 • Low (2), if 150 ≤ MED < 200 • Medium (3), if 100 ≤ MED < 150 • High (4), if 50 ≤ MED < 100 • Very High (5), if 0< MED < 50

Each parameters p_i has an associated weight w_i, which is automatically calculated as follows:

- $w_1 = max(p_2) * max(p_3)$,
- $w_2 = max\ (p_3)$,
- $w_3 = 1$ if MED value is set, 0 otherwise,

where $max(\cdot)$ denotes the maximum of the assigned value. The calculation of the final SPV is given by the formula:

$$SPV(IP) = f(w_i, p_i(IP)) = \sum_i w_i \cdot p_i(IP)$$

The values of the p_i's and the intervals can be certainly different and it would be a configuration parameter of the SIS controller. An ISP could configure the parameters and interval settings as it wishes.

The third alternative is to use four (4) rating parameters, namely the *Peer locality*, the *Route preference*, the *Physical Peer proximity* and the *External Route preference* parameter, and assign them values that, when combined for the calculation of the SPV, give the same result with the original BGP rating algorithm. In this case, the values of the rating parameters are not normalized as before.

Table 4: Mapping of BGP info to rating parameters (alternative no.3)

Parameter	**Value**
Peer locality (p_1)	*1*, if local, *0* otherwise
Route preference (p_2)	*LOCAL_PREF*
Physical Peer proximity (p_3)	*AS_PATH_LENGTH*
External Route preference (p_4)	*max(p_4)- MED*

We assume that when the MED value is not set, we then have that $MAXMED=MED=0$. Each parameter p_i has an associated weight w_i, which is automatically calculated as follows:

- $w_1 = (max(p_2)+1) * (max(p_3)+1) * (max(p_4)+1)$,
- $w_2 = (max\ (p_3)+1) * (max(p_4)+1)$,
- $w_3 = max(p_4)+1$,
- $w_4 = 1$,

where $max(\cdot)$ denotes the maximum of the values assigned. The calculation of the final SPV is again given by the formula:

$$SPV(IP) = f(w_i, p_i(IP)) = w_1 p_1(IP) + (1 - p_1) \sum_{i=2} w_i \cdot p_i(IP)$$

One can easily check that the resulting SPV value is the same as the outcome of the original BGP rating algorithm. If we want to make the SPVs more abstract, we can change the aforementioned values and weights to the following ones, so that each value is normalized to the range $[0, 1]$ while preserving the same final SPV as before (the SPV calculation formula remains the same):

Table 5: Mapping of BGP info to rating parameters (alternative no.3)

Parameter	**Value**
Peer locality (p_1)	*1*, if local, *0* otherwise
Route preference (p_2)	*LOCAL_PREF / max(p_2)*
Physical Peer proximity (p_3)	*1 – AS_PATH_LENGTH / max(p_3)*
External Route preference (p_4)	*1 – MED / max(p_4)*

- $w_1 =(max(p_2)+1) * (max(p_3)+1) * (max(p_4)+1)$,
- $w_2 = max(p_2) * (max\ (p_3)+1) *(max(p_4)+1)$,
- $w_3 = max\ (p_3) * (max(p_4)+1)$,

- $w_4 = max(p_4)$.

We have already noted that the first and third alternative give the same outcome, which follows the BGP routing rules and captures the proximity of two peers in the BGP sense. The second alternative is more of a simple arbitrary categorization of the values of the rating parameters. And since the outcome strongly depends on how the ISP configures the values and ranges, an evaluation of such an approach cannot be provided.

Between alternatives 1 and 3, we favor the latter since it achieves two goals:

1. it maintains the BGP logic in the calculation of SPVs
2. it makes available the value of each parameter so that they can be re-used in any way the ISP wants.

The last point can be further analyzed by giving an example: If the ISP wants to implement the BGP-based rating then it suffices to use the values and the weights suggested by the third alternative. However, an ISP might want to use the sub-components of the BGP rating in a different way i.e. to ignore the original BGP routing rules and provide a custom rating. In this case, it can use the value of the rating parameters (more precisely, the normalized ones) and provide its own weights that serve its purpose. The re-usability of the rating parameters renders the third alternative very appealing. Note that the BGP-based rating can also be combined with some custom rating. The ISP can take the calculated values and weights in order to have a BGP-based rating, and then it can re-use some of the parameters (e.g., the AS_hop count), associate them with some custom weights and combine them all for the calculation of the final SPV value.

Thus, we propose as the final rating approach to be implemented, the one described in the third alternative with the normalized value of the rating parameters.

3.3.2.3 Required Parameters

All following maximum values are configuration parameters and they are to be set according to the attribute values used by the ISP. They are actually the results to be provided by the $max(\cdot)$ functions introduced before.

- *MAXAS*: the maximum AS path length.
- *MAXPREF*: the maximum preference value. The default value of local preference is 100 and the range is from 0 to 4294967295 (32 bit integer).
- *MAXMED*: the maximum MED value. The metric default value is 0 but the range is the same with that of the local preference value (32 bit integer).

Also, the weights w_i for the generic peer rating algorithm should be considered as input, wherever they are not calculated.

3.3.2.4 Process

The process that runs inside of the SIS is described in the sequence diagram of Figure 3. Note that the *SIS Controller* and *Metering Module* are internal modules of the *SIS server* component, while the *SIS client* component resides at the peer.

The Metering module reads the complete routing table over SNMP from the Network (i.e., a BGP router), caches it internally and calculates the generic rating parameters according to the previous tables (depending on the alternative chosen). For some alternatives, the maximum values of the parameters are also calculated and stored in the SIS DB. At some point, a peer (SIS client) queries its local SIS, with a list of IPs to be rated. Then, the SIS Controller queries the Metering module with each of the received IPs and gets the $p_i(IP)$ values along with the maximum values for each parameters so that the weights w_i's can be calculated. The calculation of the $SPV(IP)$ takes place in the SIS Controller. After calculating the SPVs for all the received IPs, the SIS rates the list in by attaching, attaching the respective SPV, and returns the new list back to the SIS client. The client can then use the SPVs to prefer peers with higher SPVs.

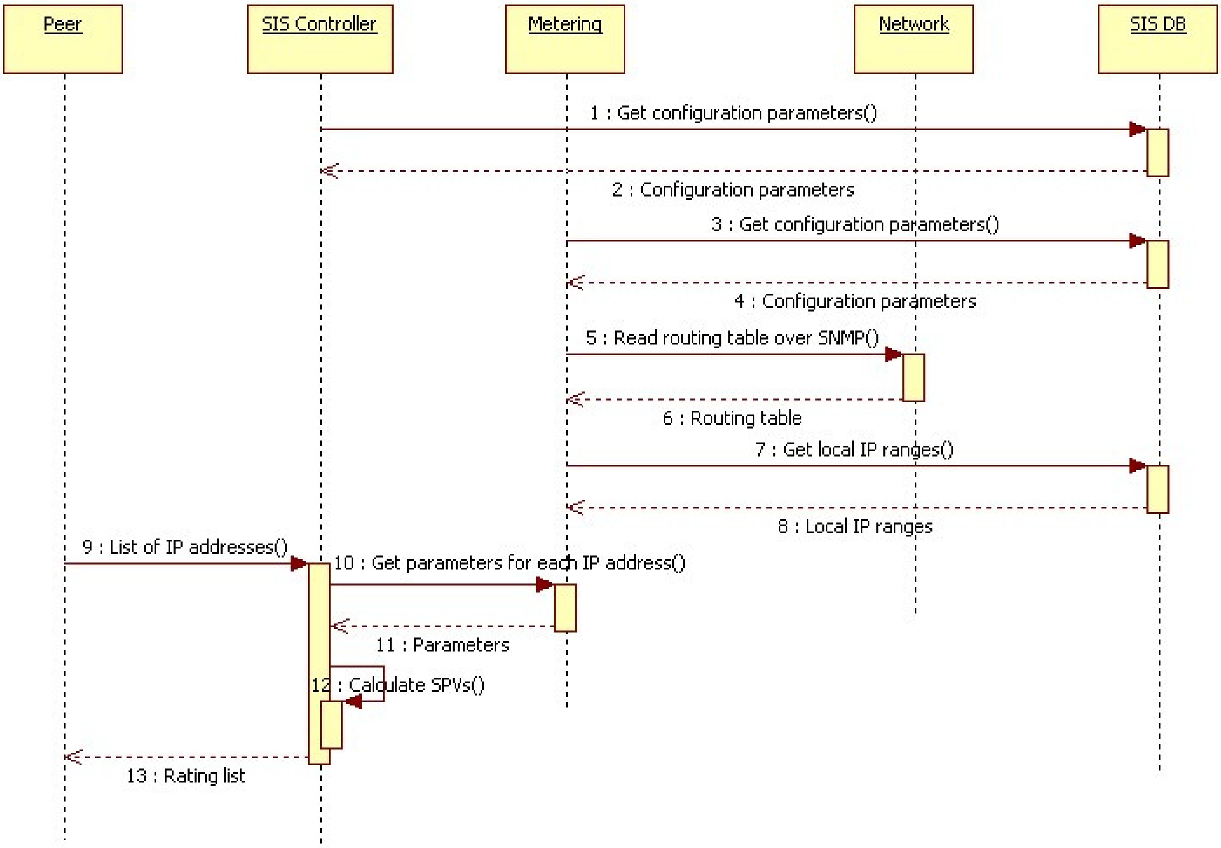
Peer
SIS Controller
Metering
Network
SIS DB
1 : Get configuration parameters()
2 : Configuration parameters
3 : Get configuration parameters()
4 : Configuration parameters
5 : Read routing table over SNMP()
6 : Routing table
7 : Get local IP ranges()
8 : Local IP ranges
9 : List of IP addresses()
10 : Get parameters for each IP address()
11 : Parameters
12 : Calculate SPVs()
13 : Rating list

Figure 3: SIS Process

3.3.2.5 Input Variables

- As input for this mechanism, the BGP routing table is required via the metering component. This includes:
    - network masks, representing the destination,
    - the local preference,
    - the length of the AS path and
    - the MED (Multi Exit Discriminator).

The local preference attribute is often used by an ISP to map business relations and preferences to the BGP routing process. ISPs often prefer routes learned from other ISPs in the following order of decreasing preference: routes learned from customer ISPs, from peer ISPs, and from provider ISPs. To map this business-related preference to BGP, an ISP can assign a non-overlapping range of local preference values to each type of peering relationship, e.g., local preference values in the range 90-99 for customers, 80-89 for peers, and 70-79 for providers.

- The weights
  - either retrieved from the SIS DB (general case)
  - or calculated when the maximum values of the rating parameters are provided (BGPLoc case)
- List of IP addresses (from the SIS client)

3.3.2.6 Output Variables

- A list of IP addresses annotated with the SPVs (back to the SIS client). This approach will prove helpful in extensions of the approach, in which the client combines the outcomes of successive ratings.
- A list of selected peers (either to contact or to upload to) based on a combination of internal overlay and SIS ratings.

3.3.3 Other Considerations

It is important to note that the aforementioned rating according to BGP metrics does not, in general, capture the actual network topology and the underlying business relationships. To become more specific, the BGP metrics used to rate a peer are valid only when they involve the uplink path. This, for example, happens when the rating is used to bias the unchoking procedure of the client (see Section 3.5.3 for more details). However, if the client uses this rating to decide which peers to download from (cf. Section 3.5.2) then the rating might not be completely accurate. There are cases where, for a specific source-destination pair, the uplink path differs from the downlink path. This may result in different BGP metrics, e.g., different AS hop counts in the uplink and downlink or different entry and exit to an AS in case of a multi-homed domain.

Having identified this disparity, the question that arises is how strong it is and how much this BGP based rating algorithm is affected. One first observation is that the effect is stronger, the longer the resulting path gets. But, since the presented algorithm already assigns very low ratings to remote peers with such distance from the local peer, the difference that would have been observed if we knew the exact downlink path would be negligible.

3.4 Dynamic Extensions of Locality-awareness

In the previous section, a BGP-based ETM mechanism that promotes locality of overlay traffic in a given AS has been described. The advantage of locality promotion is two-fold: first, the ISP is benefited since the overlay traffic volume traversing the interconnection links is decreased resulting in lower costs; secondly, the performance of the overlay application is increased since end-users experience shorter download times.

In the literature, there exists however some skepticism regarding the effect of locality promotion on the health of the swarms. In [Myth09], the authors raise some important questions related to P2P traffic localization and try to answer them, based on what can currently be supported by the research community. One such question has to do with the swarm weakening. Although there exists some indication that locality does not greatly deteriorate swarm health, the authors expect that excessive traffic localization cause some weakening. Even though the proposed BGP-based ETM mechanism does not enforce excessive traffic localization, a new mechanism is studied to enforce locality whenever needed, based on real-time measurements on the interconnection links, in order to decrease the interconnection traffic when it is required while minimizing the effect on the swarm health. The objective of decreasing costs may become equivalent to the objective of limiting the congestion level, under some circumstances (see following sections on single-homing vs. multi-homing). The primary goal of the sections however, is to decrease the link load on certain time periods based on monetary objectives and not for a performance-oriented optimization.

3.4.1 Key Idea and Numerical Experiments

To illustrate the approach, the case of a single-homed domain is considered and the effect of dynamically enforcing locality on certain time slots is studied. Dynamic locality enforcement implies that locality is not promoted all the time but is rather applied in those time slots when it has greater effect on the charging level of the interconnection traffic. Single-homing is used here in the sense that an ISP has a transit agreement with only one higher-level ISP. The number of physical links interconnecting the two ISPs can be more than one. In the following analysis, this case is considered as equivalent to having one logical link with a capacity equal to the sum of actual physical links. Hence, all interconnection traffic is aggregated. Nowadays, the case of multi-homed domains is more common. For this reason, a briefly comment on the required adaptations of the proposed heuristic is included in Section 3.4.3.

Modern pricing schemes for interconnection traffic employ the 95th percentile rule for estimating the charging level of the traffic. For more information on the scheme and its applications, the reader can refer to Section 4 of the SmoothIT deliverable “D2.2 – ETM Model and Components (Initial Version)” [D2.2]. If the ISP could predict the exact traffic levels in each of the 5-minute slots for the entire month period, then it would suffice to decrease the traffic in those time slots that offer the top 5% of the traffic. This way, the 95th percentile would decrease, and so would the respective charges, with the minimum locality promotion required.

To illustrate this better, a series of numerical experiments was conducted, in order to study the effect of locality on the 95th percentile charging scheme. For these experiments, the overlay interconnection traffic is assumed to be self-similar. Self-similarity is one of the key characteristics of Internet traffic, as mentioned in Section 3.4 of the SmoothIT deliverable “D1.2 – Commercial Application Classes and Traffic Characteristics” [D1.2]. To create traffic traces that are self-similar, several sources of Pareto-distributed ON and OFF periods with strictly altering ON and OFF periods were multiplexed, as described in [WTS97] and implemented in [KraTR].

For the experiments, 10 sources were used and created 43200 pseudo slots that resulted in 60109 slots of 5-minutes duration, i.e. a total duration of 300545 minutes. In the generated traces, the maximum aggregated traffic volume (in a single slot) was 123987 MBs, the minimum was 0 MBs, while the mean volume was 806.41 MBs. In the following plot (see Figure 4), the volume per slot is depicted, in a sorted manner. Note that not all values are depicted since the nearly top 5% of the values were too high to show. This difference resulted in a "gap" between the majority of the measurements and the top 5% of them, which plays a significant role when handling the 95th percentile.

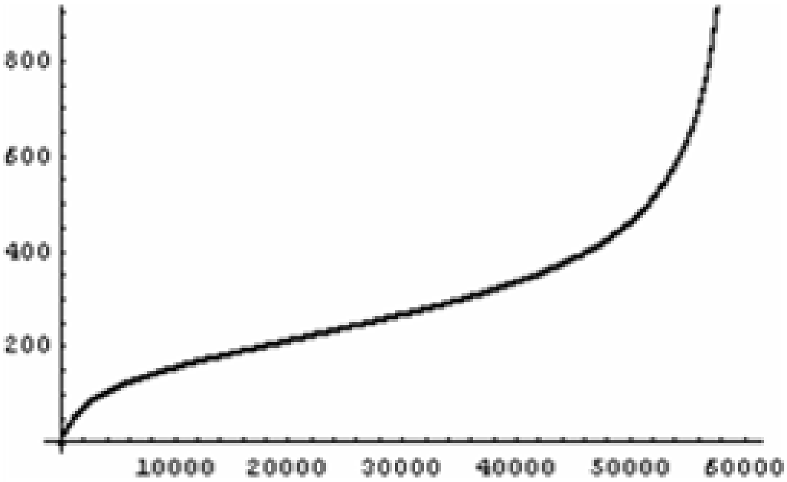

Figure 4: The generated traffic volumes per slot (sorted w.r.t. size)

In order to study the effects of traffic locality to the 95th percentile, the full information case is considered, i.e., there is prior knowledge of the exact traffic volume per slot, for all slots. Using this information, locality is applied in the top *x%* of the slots (in terms of traffic volume) and we observe how the 95th percentile is affected. One important assumption is that applying traffic locality has a fixed effect on the traffic volume. In the experiments, traffic locality is considered to introduce a decrease of *10%* in the traffic volume. The following plot shows the resulting 95th percentile and mean values for the traffic volumes, as *x* varies from 1 to 100. Note that for *x=100* the result of unconditional locality enforcement is acquired.

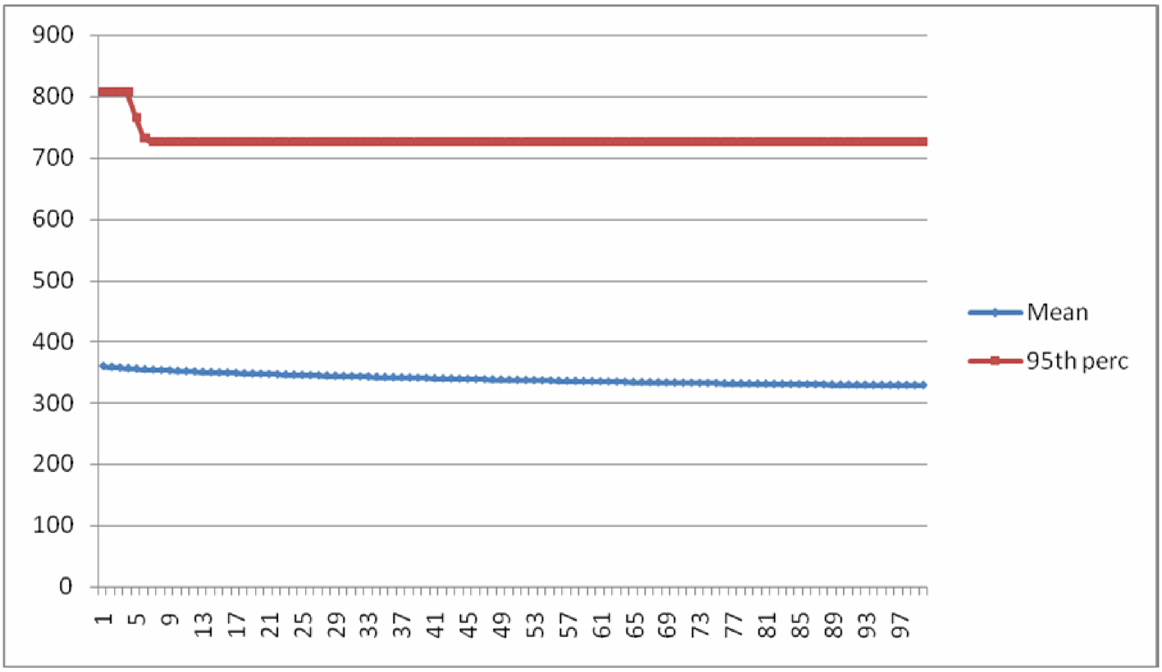

Figure 5: The 95th percentile and mean values as the percentage of enforcing locality (top x%) is increased.

It is easy to observe that the smallest possible 95th percentile value is first achieved at $x=7\%$ and the respective value is 725.768 MBs. If x is further increased then the 95th percentile value is no longer decreased, due to the generated gap, as already mentioned. Further increase of x has effect on the mean value only.

Next, the required number x to estimate the top $x\%$ slots is not known, but rather a volume threshold for applying locality is employed. The threshold used is inspired from the Normal Distribution's standard deviation and confidence intervals and is of the form $\mu + y\,\sigma$, where μ is the mean, σ is the standard deviation and y is a scaling factor. When the volume of the traffic is expected to be higher than the aforementioned threshold then we employ traffic localization. We still consider the full-information case, and we get that for the generated traces $\mu=365.321$ and $\sigma=940.163$. In the next plot we depict the 95th percentile and mean values as y varies from 0 to 1.5.

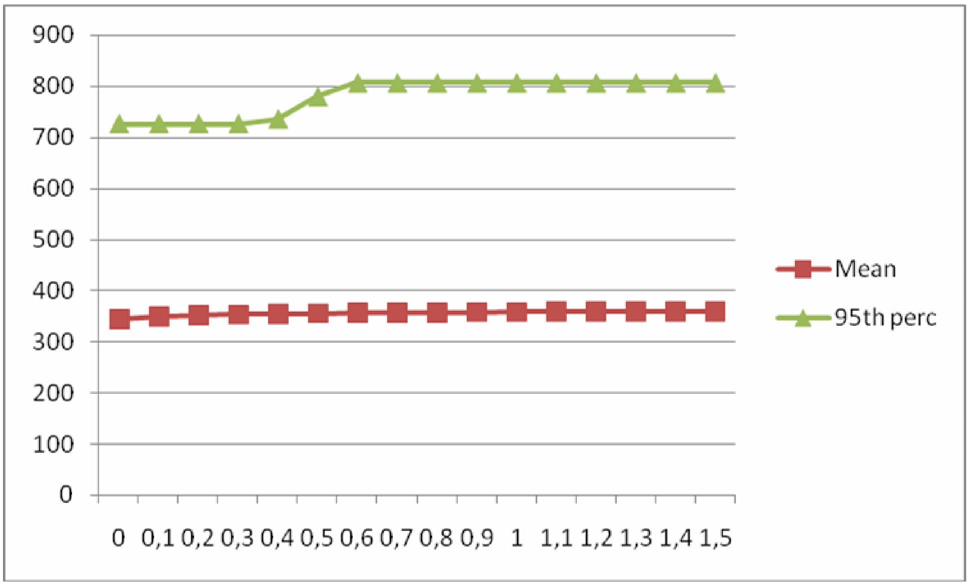

Figure 6: The 95th percentile and mean values as scaling factor y of the formula μ+y σ is increased.

It is obvious that for low values of y, the 95th percentile is at its minimum and it increases as y further increases. More specifically, this happens when y is between 0 and 0.3. To examine in how many slots locality was enforced, the percentage of slots affected is plotted in Figure 7.

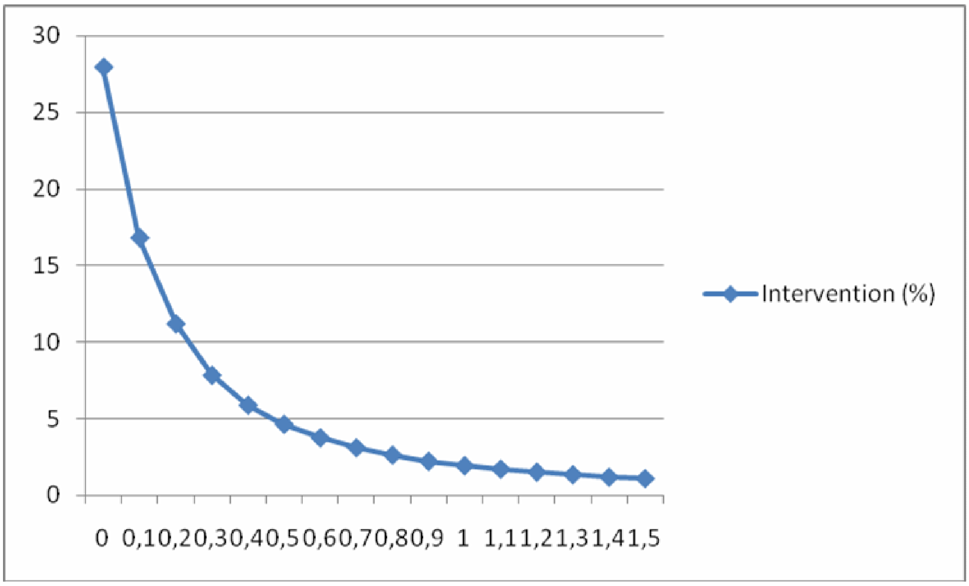

Figure 7: Percentage of affected slots as the scaling factor y increases.

Observe that for the values of y between 0 and 0.3, the latter provides the smallest intervention, i.e. only *7.85%* of the slots are affected. This result is in accordance with the previous experiment.

However, one may notice that the value of y, as well the values of μ and σ are specific to the examined sample. In order to approximate the unknown distribution, all three values that characterize the traffic pattern need to be estimated and no value can be taken as fixed. Thus, such a formula for the threshold cannot be adopted and other heuristics should be examined. However, one useful conclusion from the above experiments is that if one manages to enforce locality in the top 7-8% slots, then the result will be the lowest 95th percentile value possible and, as a consequence, the lowest charging level.

3.4.2 Specification of the Mechanism

Following the conclusions of the previous section, a new heuristic that allows us to predict whether the traffic volume in the next 5-minute slot will be in the top 10% needs to be devised. A stricter 10% threshold is employed in order to be sure that the desired effect is attained. Note that the full information assunoption no longer holds.

In order to be able to predict the expected level of traffic volume, measurements of greater granularity than the ones used to determine the charging level are needed. Since modern 95th percentile pricing schemes use 5-minute slots for their measurements, it is appropriate to use 2.5-minute measurements in order to be able to estimate the level of traffic in the 5-minute period.

Thus, a first requirement for the SmoothIT System is that the Network Management System (NMS) provides to the Metering component 2.5-minute measurements with the traffic load on the interconnection link of the specific domain.

For the heuristic to be able to predict if a slot belongs to the top 10% (or, in other words, in the 90th percentile), a training period is required during which the Controller gathers the 2.5-minute measurements, constructs the 5-minute values and calculates the 90th percentile. Thus, the training period lasts for 200 2.5-minute slots (100 5-minute slots).

After the training period, the Controller has an estimate for the current 90th percentile. For every next slot, the Controller updates this value, thus maintaining a running 90th percentile value.

In every first 2.5-minute slot, the Controller compares the measurement obtained with the running 90th percentile, divided in half. If the measurement is higher than half of the 90th percentile, then there is a probability that the entire 5-minute measurement will be higher than the running 90th percentile, thus belonging to the top 10% of the slots. In this case the Controller sets the flag equal to true.

Prior to the beginning of every second 2.5-minute slot, if the flag is true then locality is enforced by the SIS (through the aforementioned BGP-based peer rating algorithm), otherwise no rating is returned (or the SIS returns a randomly rated peer list). At the end of the second 2.5-minute slot, if the collected measurement shows that the resulting volume is still higher than the running 90th percentile value, the flag is set to true so that locality is enforced to the next first 2.5-minute slot.

Below the pseudo-code of the described dynamic locality heuristic is provided.

```
qnt= running 90th percentile;
flag=0;

For every slot i {

     If (Mod[i, 2] ≠ 0)
          If (flag == 1)
               Enforce Locality;
               If (Volume(i) < qnt/2)
                    flag=0;
          Else
               If (Volume(i) > qnt/2)
                    flag = 1;


     If (Mod[i, 2] == 0) {
          If (flg == 1)
               Enforce Locality; flag=0;

          If (Volume(i) > qnt/2)
               flag = 1;

          Update qnt;

     }
}
```

It is however neither expected that the above heuristic will be able to accurately capture the top 10% slots nor that it will achieve the lowest 95th percentile value. It provides a simple way for estimating the expected link load and for deciding on whether to promote locality or not. Indeed, if the aforementioned heuristic is applied to the numerical data of the previous experiments, the attained 95th percentile value is 791.45 MBs and the percentage of intervention is equal to 21.62%. Certainly, the achieved 95th percentile value could be lower if stricter rules were in place, but this would mean that the percentage of intervention would increase. Hence, the goal is to achieve the best possible 95th percentile value with the minimum intervention. More precise prediction mechanisms are to be further studied.

3.4.3 The Case of Multi-homed Domains

So far, the case of single-homed domains was examined. The alternative of a domain interconnected to another domain with multiple physical links, is considered equivalent for technological reasons, as long as the same pricing scheme applies for all links. For the case where a domain interconnects with more than one higher-level domains, with more than one physical links and with different pricing schemes per domain, more issues must be considered and new heuristics must be devised.

This new case of interconnection, allows more flexible heuristics. Indeed, when more than one logical interconnection links are considered there exists the possibility of load-balancing traffic, always with the purpose of decreasing the interconnection costs. In some cases, this might result in increasing the load levels of certain, “cheap” interconnection links. Thus, the necessity of combining cost-minimization heuristics with performance optimization mechanisms is obvious. Another critical point to be considered, relates to the actual locality-enforcing mechanism in place. Dynamic-enforcement of locality implies that a locality enforcement mechanism in place should be able to distinguish the ingress points and apply specific rules to the traffic travelling through them. This is not the case however for the BGP-based locality mechanisms, which considers the entire domain and handles all incoming inter-domain traffic as a whole.

3.5 Client-side Support for Locality

Once the SIS has provided information to a peer about contacts in the overlay, this information has to be put to use in order to influence the generated traffic. This section covers the different alternatives considered for implementation in the trials.

The mechanisms presented here fall into two categories. Mechanisms of the first category try to influence the composition of the neighbor set of a peer (‘Biased Neighbor Selection’ or BNS), while the others try to influence the composition of the active set (‘Biased Unchoking’ or BU).

At first, we describe how the affected mechanisms (neighbor set management and unchoking) work in normal BitTorrent clients and in the Tribler client. A more detailed description can be found in D2.1 [D2.1]. Then we describe the considered modifications and present the overview of the simulation results obtained with the modified mechanisms.

3.5.1 General Mechanisms in BitTorrent and Tribler

The mechanisms of BitTorrent and Tribler that are changed by BNS and BU are the neighbor set management in case of BNS and the unchoking algorithm in case of BU. To understand the modifications introduced by our client-side ETM, we first describe the standard implementation of these algorithms in BitTorrent.

3.5.1.1 Neighbor Set Management

In the BitTorrent-based (BT-based) systems that are in the focus of the project, there are two types of peers that are known to a certain peer *P*. The first are contacts that are only known by their address and overlay ID. Not all contacts are selected for connection establishment. Only when a (bidirectional) connection is established between two peers, they consider themselves neighbors. Neighbors of peer *P* are permanently stored in the neighbor set of *P*, with additional attributes for a connection (such as the chunk map of the remote peer). Neighbors are typically removed from the neighbor set only when they go offline.

Within the set of neighbors of a peer, there are again several subsets. Not all neighbors actually have data (i.e., chunks) that *P* needs, at all times. Peer *P* may only download data from those peers that have at least one chunk that *P* is missing, which is, in terms of BT, being interested in those peers. The interested status is updated whenever a neighbor signals that it has received a new chunk (cf. Figure 8).

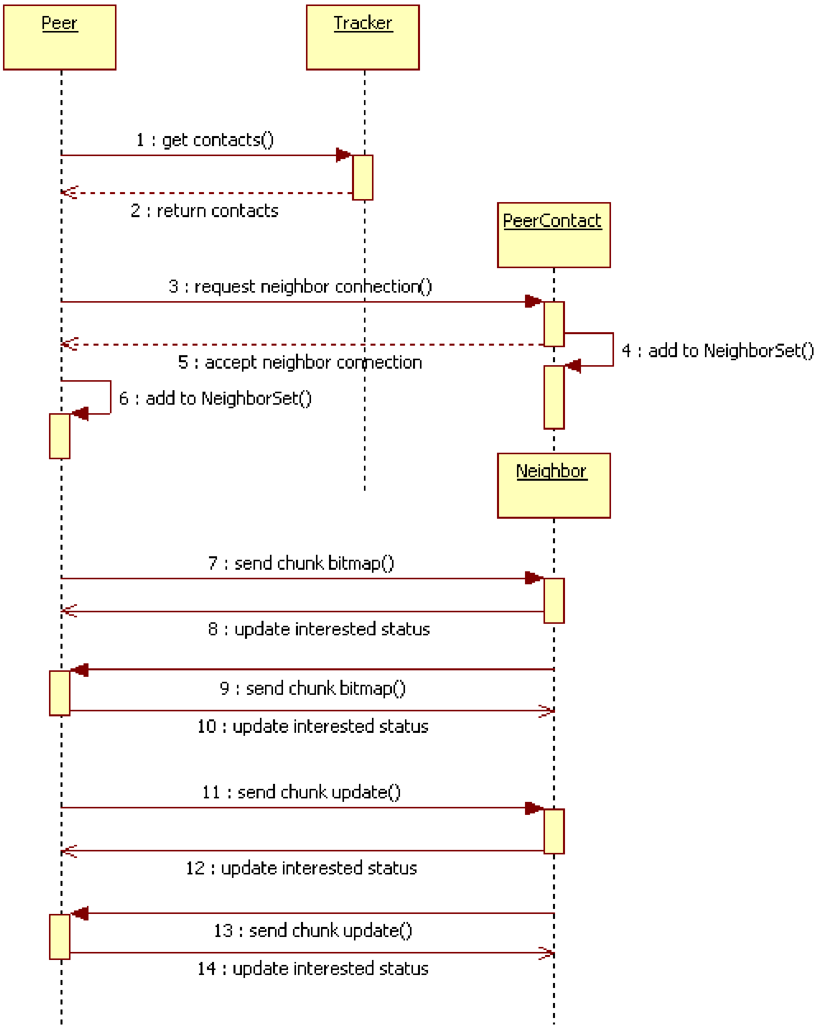

Figure 8: Sequence diagram of the neighbour set management of a peer when querying the tracker for the first time

In turn, only a subset of *P*'s neighbors is interested in *P* at any given time. The peer selection of the unchoking process (see next section) only considers the subset of the interested neighbors, since the other neighbors do not need any chunk from the peer P. The peers that are at a given point in time unchoked, i.e., which *P* allows to request chunks and uploads to, are also called the active set of peer *P*.

Two neighbors cannot be categorized permanently as 'uploader' and 'downloaded' since they may both up- and/or download from each other at any given point in time.

3.5.1.2 Choke Algorithm

The choke algorithm decides to which interested neighbor a peer is willing to upload data. These neighbors are called unchoked whereas the neighbors that do not receive data are called choked.

Every 10 seconds, the peer unchokes a default number of 3 of its interested neighbors. Which neighbors are unchoked depends on whether the peer has already downloaded the complete file (seeder mode) or not (leecher mode). In leecher mode, the peer unchokes those 3 neighbors from which it receives the highest download rates. This strategy is called tit-for-tat (T4T) and provides an incentive for peers to contribute upload bandwidth to the swarm. In seeder mode, the peer keeps those 3 peers unchoked which were most recently unchoked. In both seeder and leecher mode, every 30 seconds one of the interested and choked neighbors is selected randomly and unchoked for the following 30 seconds. This is called optimistic unchoking and allows the peers to get to know new mutually beneficial connections. In combination with the optimistic unchoking, the choke algorithm in seeder state ensures that every 30 seconds, the peer with the longest unchoke time is choked and a new interested peer is unchoked. Consequently, the upload slots of a seeder are distributed in a fair way among its interested neighbors.

In Tribler, the mechanism basically works in the same way, with the exception that tit-for-tat is replaced by another metric, namely give-to-get (G2G). Since it is much more unlikely in Tribler that two peers can upload to another due to the different chunk selection strategies, a peer in Tribler regularly unchokes 3 neighbors that forwarded the most data it originally uploaded to them. As a tie-breaker, the total upload of peers is used. Thus, peers that upload more are preferred over “selfish” peers. This strategy does not need to be changed in seeder mode. The optimistic unchoking mechanism remains the same as in BitTorrent.

3.5.2 Biased Neighbor Selection

Biased Neighbor Selection (BNS) aims at changing the composition of the neighbor set of a peer in order to indirectly shift traffic to preferred peers. The idea is that if preferred peers are found with a higher probability in the neighbor set, then they are also unchoked with a higher probability. Therefore, data is received by those peers more often.

We discern two basic alternatives in implementing BNS, which are basically described in [BCC+06]. The first is tracker-based BNS, where the tracker inserts in its response a certain amount of contacts that are close to the requesting peer. In an SIS-compatible solution, this means that the tracker has to query the SIS ratings for all peers in the swarm in relation to a querying peer *P* (an alternative would be having a combined SIS/tracker entity). When the ratings are known, the tracker constructs its response by inserting a fraction $0 < p < 1$ of peers that have the best SIS ratings and fills up the rest with random peers. The total number of returned peers *N* has either been specified by *P* or is a tracker setting.

The advantage of this method is that the tracker knows all peers in the swarm and has therefore a large set to choose suitable peers from. The drawbacks are a higher complexity at the tracker as well as a reduced degree of freedom for the peers, which cannot choose whether to actually support BNS or not. Instead, the overlay/tracker provider has to cooperate with the ISP. Therefore, this solution is not compatible with the incentive-based approach of SmoothIT. However, the tracker-based BNS roughly represents a best-case scenario with respect to the results on the neighbor set that can be achieved using BNS, since the tracker always knows the best peers to be included in the neighbor set.

To allow peers to implement BNS themselves (i.e., peer-based BNS), their behavior in establishing connections to neighbors has to be modified. The tracker is not changed in this approach. Instead, a peer *P* requests more contacts from the tracker than in the default overlay, either by specifying a larger number of peers to be included in the response or by querying the tracker repeatedly.

In order to prefer establishing connections to local peers, all received peer contacts of *P* have to be rated before connections are initiated. Then *P* can choose to which peers it wants to send a connection request. In our implementation *P* wants a certain fraction f_{pref} of its neighbors to be peers with a good SIS rating. Therefore, it takes the $f_{pref}*N_{default}$ currently known contacts with the best ratings and opens connections to them, with $N_{default}$ being the default number of neighbors a peer wants to have (typically 40). To fill up its neighbor set, it chooses randomly from the remaining contacts. All connections to peers previously in the neighbor set that are no longer chosen are closed. This means that the neighbor set is revisited (potentially completely) every time a new contact is received by the tracker and rated. This does not hold for contacts that are established between peers, i.e., when a remote peer sends a connection request to the local peer. In this case, the normal BT rules apply, and the neighbor set is not completely re-evaluated.

For the simulative performance evaluation we chose $f_{pref} = 0.9$. The algorithm is given in pseudo-code form below. It is executed every time a contact is received from the tracker and rated.

1. Input: Current neighbor set N_{old}, newly rated contacts C_{new}.
2. Sort all contacts $N+C_{new}$ by SIS rating
3. Initiate/keep neighbor contact to the best $f_{pref}*N_{default}$ contacts, add these contacts to N_{new}.
4. From the rest of the contacts, take $(1- f_{pref}) *N_{default}$ contacts and initiate neighbor connection, add to N_{new}.
5. For all *n* in N/N_{new}, end connection, remove form neighbor set.

3.5.3 Biased Unchoking

The concept of BU as described below was developed in the context of the SmoothIT project. In parallel, [LCL+09] introduced one of the described variants (BSEPU), which was not followed up upon due to reasons given below. Since the topology and scenario used in the evaluation of [LCL+09] are not described, a comparison of the results is difficult.

In contrast to BNS, BU is less complex in its implementation. The communication for connection establishment between the tracker and the clients as well as between the clients themselves is untouched. Instead, a peer preferentially exchanges data with neighbors with a high SIS value. We let a peer *P* query the SIS for the rating of a neighbor after it has established a connection to it. Since every neighbor may be interested in *P* during its lifetime, it is more practical to have every connection rated ahead of time than to query the SIS only when a neighbor becomes interested. In case BU is used in combination with BNS, the SIS rating would have been already obtained before the connection establishment in most cases. In any case, for the purpose of the algorithm we just assume that the rating is known during the unchoking process or, if no such value exists, that the peer has the minimum SIS rating.

In BitTorrent-based overlays, the choke algorithm selects the neighbors to which a peer allocates its upload capacity to. Consequently, BU influences the choke algorithm. In contrast, BNS influences the neighbor set management.

BU is motivated by the fact that, apart from the composition of the neighbor set, the choke algorithm has a major impact on which peers exchange data and how much. Especially, when only a few peers in one AS are online, all that BNS can achieve is that these peers are in the neighbor set of each other. Still, the number of these neighbors may be small compared to the number of all neighbors and that constrains the performance of BNS. Therefore, we also propose BU, which is intended to boost the data exchange between peers in the same AS in those situations.

There are several alternative methods to implement BU. The most straightforward one is Biased Optimistic Unchoking (BOU). With BitTorrent (and Tribler), all *k* interested and choked neighbors *Y* of a peer *P* are selected to be optimistically unchoked with same probability *pou(Y)=1/k*. With BOU, this probability *pou(Y,SIS(P,y))* that an interested and choked neighbor with address *y* is selected to be optimistically unchoked by *P,* depends on the SIS rating *SIS(P,y)*. In this way, we can achieve that neighbors with good locality values are optimistically unchoked more often than other ones.

The possibility for *pou(Y,SIS(P,y))* chosen for evaluation is to select the peers to be optimistically unchoked from the subset of interested peers with the best rating value *SIS(P,y)*. In case there are not enough peers with that rating, the rest is chosen from the subset of interested peers with the second-best rating, and so on. In the evaluations (cf. Section 3.6 and Appendix B) we unchoke a single peer optimistically, meaning that we choose this peer from the set of interested peers with just the best SIS rating. Thus, even if no interested neighbour in the same AS exists, a peer in a neighbouring AS with peering agreement would still be chosen over a remote peer 5 AS hops away. Since the number of optimistically unchoked peers is 1 in BT, as well as in our implementation of Tribler, we give the algorithm described above in pseudo-code for this case:

```
1. Input: Set of interested peers $P_I$
2. Sort all peers in $P_I$ by their SIS rating
3. Select 1 peer randomly from $P^*_I$, the peers with the best SIS rating value $R^*$.
```

Another alternative method to implement BU is to reserve regular unchoke slots for peers with a good SIS rating (Biased Unchoking with Separate Unchoking slots, or BSEPU). Of the *n = 3* regular unchoke slots a peer has in default BitTorrent or Tribler, a number *m < n* may be reserved for BU. Then these *m* slots are given to the peers with the highest SIS rating instead of the normally used overlay rating (T4T or G2G, respectively). The overlay rating may still serve as a tie-breaker here. The other *n-m* unchoke slots are given to the peers with the highest overlay rating, as usual.

This approach has several principal drawbacks in comparison to BOU. The first is that regular unchoking only takes place when a peer is in the leecher mode. In the seeder mode, peers are added to the active set of the peer by optimistic unchoking exclusively in the BitTorrent overlay. Regularly unchoked peers stay unchoked, only the peer that was unchoked the longest is choked. This means that BSEPU would need to be implemented for both seeder and leecher modes separately, otherwise it would be only effective in leecher mode. Another drawback is that the overlay rating that should ensure protection against free-riding during regular unchoking is partially bypasse. In contrast, BOU lets peers get to know primarily local peers, but these can be excluded from the active set again if they do not follow the tit-for-tat principle. Therefore, BOU should be preferred over BSEPU.

The third alternative being considered is to create a combined rating of a neighbor *y* from the overlay rating $R_O(y)$ (T4T or G2G) and the SIS rating $R_{SIS}(y)$. Since the different ratings have different ranges, simply using them as they are generated by the peer and the SIS, respectively, may lead to a much higher weighting of one of them. Therefore, we normalize both ratings by dividing them by the maximum value observed in the last rating period (10 seconds for T4T and G2G). The resulting values $R'_O(y)$ and $R'_{SIS}(y)$ are both in the range of [0;1]. Then we can compute a combined rating

$$R_{comb}(y) = \alpha * R'_O(y) + \beta * R'_{SIS}(y),$$

with α and β being user- , client- or overlay-defined weighting factors. The regular unchoke slots of a peer are then assigned exclusively based on this combined rating.

This mechanism also has some drawbacks. By creating a joint rating, the importance of the overlay part of the rating is decreased. This gives room for selfish behavior of peers that are aware of having local neighbors. Also, the combined rating will have no effect in the default seeder mode of BitTorrent, or will again have to be implemented separately for this mode.

3.6 Simulation Results

In this section, the main conclusions from the simulative performance evaluations conducted on the proposed ETM mechanisms are described. The detailed simulation setup, experiments and outcomes can be found in Appendix B.

From the observed results, we draw the conclusion that the usage of the SIS-generated locality information in the client is of high importance. Even if the SIS provides good data, it is worthless to the provider or the user if it is not used effectively. In this regard, the combination of BNS and BU works consistently better than BNS or BU alone.

The reason for this is that BNS ensures that there are more local and close neighbors in the neighbor set of a peer than in regular BitTorrent. It achieves inter-AS traffic reduction simply by assuming that a higher fraction of neighbors in the same or peering ASes indirectly leads to a higher fraction of locally forwarded traffic.

In contrast, BU prefers local and close neighbors in the most crucial mechanism for deciding on the direction of traffic flows, the unchoking process. It can directly affect the fraction of local and remote traffic. However, this works only if a) there are actually local interested neighbors and b) there are enough interested neighbors in total so that an actual choice has to be made which peers to unchoke. In low load scenarios and in swarms with a low number of peers in the same AS, these conditions are not always met. While peers have to trust that the bias in their unchoking process is reciprocated in order to profit from it, the fact that we only bias the optimistic unchoking should prevent a selfish exploitation of a BU peer by other peers.

In combination, the two mechanisms combine their advantages, leading to a synergy. BNS provides the local neighbors necessary for BU to function effectively, while BU ensures that the change in the neighbor set composition created by BNS is used to maximum effect. Thus, the solution generated in the project enhances existing locality-promoting client-side mechanisms.

With regard to scenarios, the locality promoting strategies work best in swarms with high load, since there the number of interested peers in the neighbor set is generally larger. Therefore, a preference of local or close peers in the unchoking process has a larger effect here. Also, larger fractions of the swarm per AS increase the efficiency of locality promotion, since there are more local and close neighbors that can be utilized.

Not surprisingly, if only a fraction of peers in the swarm support locality, the performance improvement increases with the fraction of locality-promoting peers. In general, locality-promoting peers perform better than their counterparts that use the regular implementation if the user performance is limited by inter-AS bottlenecks. Nevertheless, the latter do benefit from the behavior of the former.

The quality of the improvement achieved by locality promotion depends on the evaluated topology. We could corroborate known results from related studies that in scenarios with connection bottlenecks only in the access networks, download times remain largely unaffected by locality promotion, while savings in utilized inter-AS bandwidth can be achieved. This might still lead to a win-win situation if the providers are able to pass on the achieved savings to their customers or are enabled to offer an additional benefit, e.g., higher access bandwidth.

In contrast, if tight bottlenecks between ASes exist, still a little inter-AS bandwidth can be saved, but the largest effect here is on the download times, since these are affected strongly by the bandwidth limitation. Download times can be improved significantly by locality promotion in these scenarios. However, these scenarios are less realistic for a real provider than a topology with bottlenecks in the access network.

In general, the traffic locality achieved by the BGP-based locality algorithm in combination with the client-side mechanism leads to a great improvement over the regular case. Inter-AS traffic is reduced and replaced by intra-AS traffic and peering traffic. However, we considered only a homogeneous swarm in this evaluation, meaning that both the access speeds as well as the distribution of the peers over the underlay topology was uniform. We are currently conducting experiments where this is not the case in order to assess the effect of traffic locality in scenarios with heterogeneous access bandwidths and a skewed swarm distribution in the topology. Early results of these experiments indicate that in case of topologies where certain ASes that offer a lower access speed, e.g., the domain of a mobile network provider, locality promotion has a negative effect on download times of peers with such a limited access. This is due to the fact that for these peers, local neighbors always offer less upload capacity than potential remote neighbors.

4 Inter-SIS Mechanism

This section aims to provide an overview of the most interesting scenarios that could lead to the exchange of information between SIS systems deployed in different ISPs and/or Autonomous Systems.

First of all, it is important to analyze two main types of communication that could be considered:

- Non run-time communication: the SIS systems exchange relevant information (i.e. intra-domain topology abstractions, their BGP tables, etc.), but this exchange does not require real time interaction; this information will be part of the almost static information that can be used by the SIS servers to perform the rating without the need for interaction in real time.
- Run-time communication: the SIS servers do not exchange information between them and in order to collaborate, one SIS can work as a client of the other SIS.

The first mode of operation should allow each SIS to build a map of the topology. The following figure shows a map of a selected topology, where several Tier 2 ISPs (A, B and C) have peering agreements between them and transit agreements to Tier 1 ISPs (D and E).

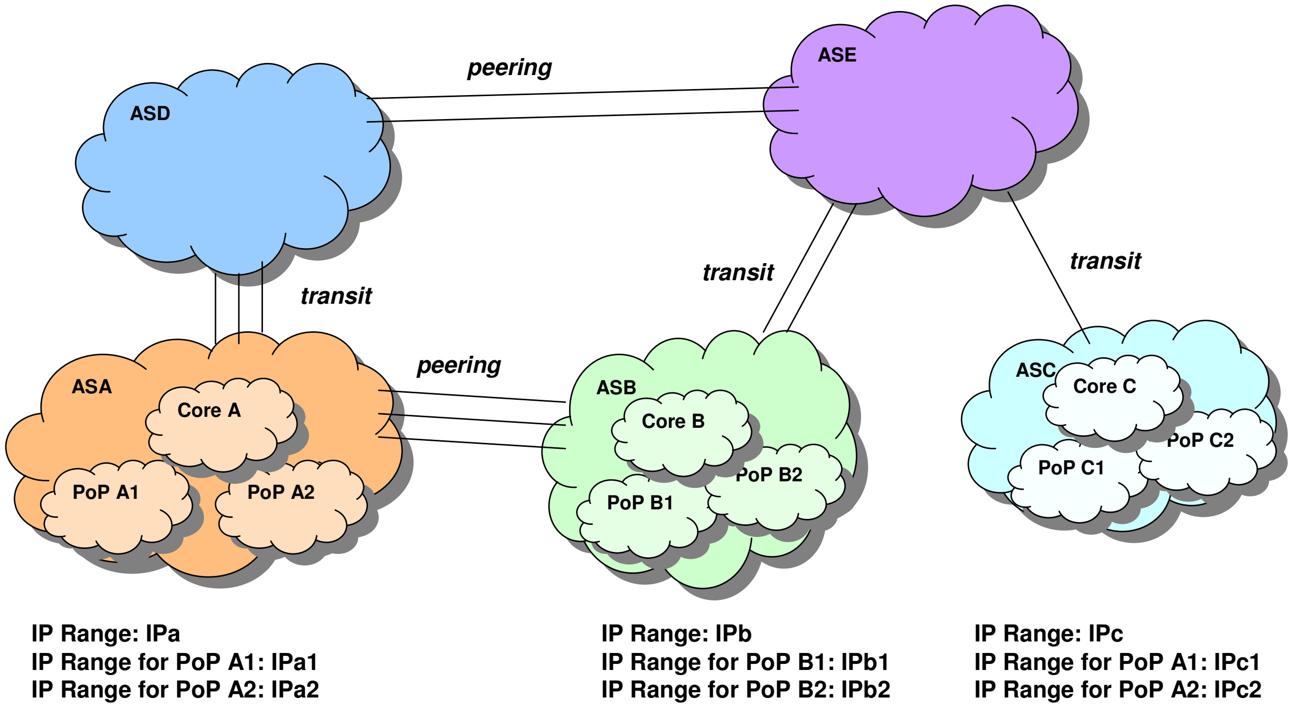
peering
ASE
ASD
transit
transit
transit
ASA
Core A
PoP A1
PoP A2
peering
ASB
Core B
PoP B2
PoP B1
ASC
Core C
PoP C2
PoP C1
IP Range: IPa
IP Range for PoP A1: IPa1
IP Range for PoP A2: IPa2
IP Range: IPb
IP Range for PoP B1: IPb1
IP Range for PoP B2: IPb2
IP Range: IPc
IP Range for PoP A1: IPc1
IP Range for PoP A2: IPc2

Figure 9: Topology map

The information exchange should allow that, e.g., the SIS located in AS A could get accurate information of the topology. In this case, the following exchange of information with domain B could be the following:

- Both A and B are aware of the global IP ranges that are assigned to each AS. This is globally available but high-level information.
- AS B informs AS A that it has 2 PoPs and that it prefers that peers in AS A connect to the peers in the PoP B1, so B says that IPb1 addresses are preferred.

- This could be useful in case that there are some other PoPs where, e.g., due to geographical conditions (e.g., an island) the cost to transport the traffic is higher or the AS B is simply interested in balancing the traffic in a specific way.

Another interesting scenario that could lead to an interaction among SISs deployed in Tier 1 ISPs is the scenario where B, E and C must interact. In this case, in principle, the B and C domains have no relationship so they could rely on an SIS server deployed in E which provides transit agreements to both domains. The SIS in E could be interested in enforcing some locality inside the cloud of domains it maintains in order to avoid using the inter-domain links to other Tiers-1, which are usually high-speed links with an important associated cost (in terms of both CAPEX and OPEX related just to the management of the traffic). In this case, the SIS deployed in AS E could collect information provided by AS B and C and provide it to all the domains attached to them. The customer domains (B and C) could, e.g., provide information about just AS Paths, PoPs, etc. and get some reduction on the price. This scenario will be described in more detail in Subsection 4.3.4.

4.1 Route Asymmetry with BGP

ISPs establish different business agreements between each other such as the ones that are indicated in the topology of Figure 9. Those peering and transit agreements in combination with other parameters such as the latency or the congestion of a network and the existence of multi-homing ISPs affect the route selection from one AS to another. This may lead to the selection of different paths, uplink and downlink, between each pair of ASes as they may evaluate the parameters differently.

The possibility of the existence of two different paths between the same two ASes in the two directions motivates studying the collaboration between the source and destination ISPs in order for the destination provider to provide information about the asymmetric routing. This information will allow different SIS servers to obtain better knowledge of the topology of the network. The source ISP will use this information in order to rate the peers of the destination ISP. This rating may affect the selection of remote peers of the local peers. Thus in this scenario that examines the collaboration between source and destination ISPs it is essential to define the incentives of the destination ISP to provide the source ISP truthful information.

The source ISP (say A) may request information about the downlink (BA path) from the destination ISP (say B) in order to rate the peers of B. The destination ISP may (from previous communication) or may not have the knowledge about the parameters of the uplink (AB path). First, it is assumed that the destination ISP B does not have any knowledge about the parameters of the uplink path (AB). It can either answer truthfully or not. In general it will answer truthfully due to existence of tit-for-tat rule which will enable its own peers to be unchoked by the ones in ISP A. However, the incentives to answer truthfully may depend on the business agreements that are established. For example, ISP B may be charged for sending data to peers of the source ISP A more than to other peers and for this reason it may not prefer its peers to be connected to or be rated higher by the source ISP's peers. This assumes that the destination ISP knows that the preferable ISPs own peers that are participating in the same swarm as it does.

The other scenario describes a destination ISP that has knowledge about the uplink (from the source perspective) path that has been obtained from a previous collaboration. This scenario is more complicated in what concerns incentives since the destination ISP is now able to compare the two paths. We examine two possibilities, the one with the uplink (AB) being equal or better relatively to the parameters that are used than the downlink (BA) and the one with the uplink (AB) being worse that the downlink (BA) as this is defined from the parameters. If the uplink (AB) is better than the destination ISP wants to collaborate with the source ISP and thus it has the incentives of being truthful. On the other hand, if the uplink is worse in general it will want to collaborate due to tit-for-tat reasons. However, there may exist business agreements that motivate the destination ISP not to collaborate. Due to those agreements the destination ISP may prefer other ISPs to exchange traffic with.

A specific example of this case, presented in Figure 10. 2B, is a Tier 2 ISP that has arranged a peering agreement with Tier 2 ISP 2A. 2B in this case is the source ISP that wants to collaborate with 3B that is a Tier 3 ISP. 2A is not charged for exchanging traffic with 3A ISP, while on the contrary 3B is charged by 2A for exchanging content. Thus 3A may not want to exchange traffic with 2B especially if it is connected to other ISPs who use the same swarm and it is not charged to exchange traffic with. Nevertheless in Figure 10, 3B is connected only to 2A ISP. This indicates that as it is charged to send content to any other Tier 2 ISP, it has incentives to collaborate truthfully since 2B ISP is less AS_hops away than others. In this context we understand AS_hops as the number of Inter-AS links, that a data packet has to pass on its path from the source to the destination. As mentioned in following section, here, 2A ISP may provide monetary incentives to 3B ISP in order for him to collaborate.

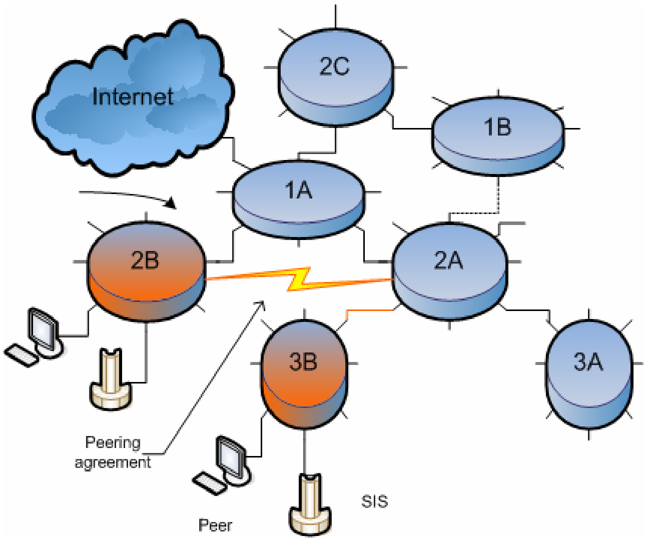

Figure 10: Collaboration between source and destination ISPs

4.1.1 BGPLoc Algorithm for Route Asymmetry

This subsection deals with the route asymmetry that follows from the BGP routing principles. The ETM mechanism in its basic form makes an assumption that it does not make changes to the routing tables. In other words no traffic engineering is performed on the underlay network level, but some kind of traffic engineering on the level of the overlay network is made. The existing paths are used and they are not modified for the purposes of the peer-to-peer traffic. The ETM mechanism will be used for the overlay traffic engineering.

ISPs use their own routing polices for transfer traffic between them, they establish on the routers some rules for announcing routes via BGP protocol. In the first part of Section 4.1 some economic motivation for usage of specific connections for transfer peer-to-peer traffic between ISPs has been considered. Some incentives for specific peer-to-peer traffic organization were presented. The ETM has no influence on the BGP routes, but it can modify the peer-to-peer traffic by the specific selection of peers.
Any AS border router in the ISP network knows the uplink routes to the destination, generally it does not know the downlink route, even the neighbor AS, from which the download traffic comes, is not known (only for the specific topology it can be recognized). Below we consider the example where the uplink-downlink BGP route asymmetry exists. The sources of information for the establishing of the download path are explored. The new BGPLoc algorithm for rating peers is presented, it is different from the one defined in Appendix A. This new algorithm works both for the scenario with route asymmetry and in the symmetric case. This algorithm also promotes locality. Communication between an SIS located in the client AS and the remote SIS in the peer AS is required for obtaining complete information for the rating. Instead of using an SIS, a Looking Glass Server (LGS) with BGP information can also be used. A LGS is a computer that is publicly accessible. It works as read-only portal to routers of the organization that deployed LGSes in their network. Some ISPs own LGSes in their ASes which are able to show the BGP routing information.

Generally the communication between two peers does not need to follow the same path in the uplink and downlink direction. The BGP locality algorithm that is specified in Section 3 rates peers for the uplink connection. If the path followed by packets is not the same in both directions, we can have wrong information about the AS hops number for the download route.

In Figure 11 we present the situation where the client is located in the AS 100 and its IP address is 100.100.100.1. The client got the unrated list with the peer IP address 155.155.155.1.

From the routing table on a BGP router located in the local AS (i.e., the AS which is local for the client), we know that the client requests to this peer will traverse three autonomous systems. When the request reaches the peer, it will send the chunks on the route indicated by the BGP entry in the routing table in the AS 400. This example shows that the downstream distance, measured in AS hops, is shorter than the upstream connection.

The SIS-SIS communication now enables obtaining information about the real AS hops distance in the downstream direction.

Prior to communicating with a peer the client asks the local SIS to rate the peer list. The rating procedure is based on the BGP information acquired by the local SIS from the remote SIS. The BGP AS_PATH attribute is passed by the remote SIS to the local SIS (in the next paragraph, an overview of confidentiality issues is provided). In our example the download distance is 2 and the upload is 3.

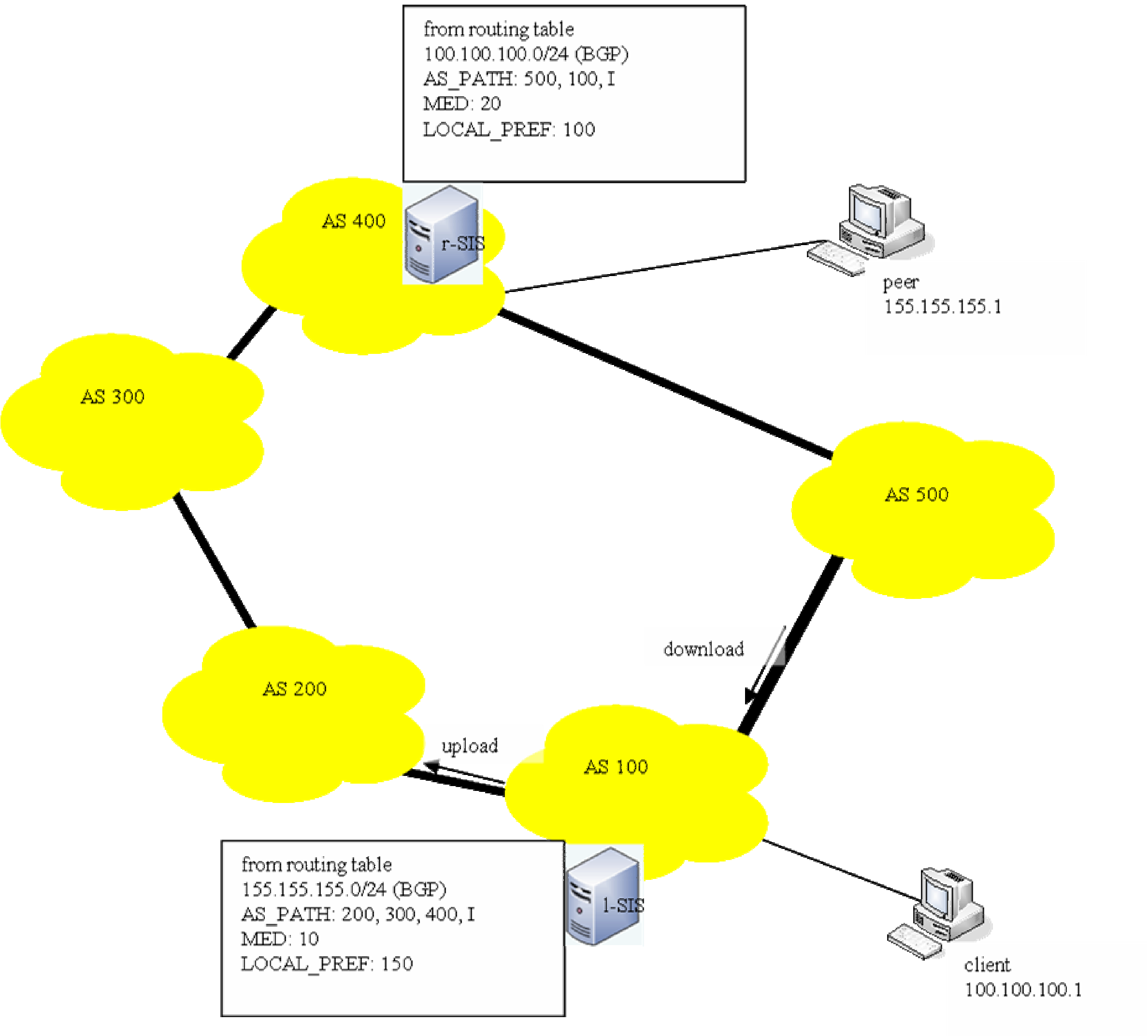

Figure 11: Example topology

The rating algorithm processing this additional information uses information gathered from BGP. The flow diagram of that algorithm is presented in Figure 46 in Appendix D. In the algorithm we consider three sources of BGP information, namely SISes and Glass Looking Servers located in peer ASes and RIB-out information from EBGP routers in the client AS. The idea is that the local SIS asks the remote SIS about the BGP AS_PATH attribute stored in routing tables of remote AS BGP routers. This AS_PATH indicates the path to the client network. From the perspective of the local AS this is the download path and from the perspective of the remote AS (any hosts, routers) this is the upload path.

The algorithm presented in the Appendix D describes the general rating schema for download traffic; it uses all available resources of BGP information.

In this section we consider the architecture in which the local AS and the remote AS possess their own SISes and the communication between these two SIS instances is allowed. The problem of the SISes finding each other's IP addresses will be discussed in a following section.

Some privacy considerations are worth mentioning. ISPs may not want to expose to the other ISPs the information how they transfer the traffic through the Internet. These ISPs can send just implicit information about AS_PATH. Instead of sending the full AS_PATH, they just send the AS hop number and the number of the AS which is just one before the client AS on the download AS_PATH. This AS number is very important because this identifies the ingress interfaces for traffic from the peer AS. This information also allows applying some economic rules for rating peers that are related to peering agreements between providers.

The proper level of confidentiality for transferring the information between ISPs through SISes can be obtained by applying the community schema for this communication. This schema is described in a separate section.

The algorithm should be supported by caching procedures. All the information about networks in remote ASes is to be stored by the local SIS and should be associated with a timeout. Usually if some content is popular, the same groups of peers will appear in unrated lists proposed to clients. So it is worth caching the information about peer networks, which has been already analyzed. When the information is in the cache there is no need for communication with a remote SIS or a LGS. This way we can limit the inter-SIS traffic. The caching procedures should be supported rather for peering networks than for peers. The BGP routes are time stable, only routes that are updated due the BGP will be removed from SIS caches. New communication with the SIS in the remote system will be required when some peers in the remote AS will be again offered.

Details of the algorithm are presented on the diagram (Appendix D). The algorithm can be used after communication with an overlay network. It is based on an assumption that a client has acquired a suggested unrated list of peers possessing searched content. This corresponds to the standard scenario considered in Section 3 for BGPLoc (for example, in the case of BitTorrent, that list has been delivered by a tracker). In the next step, the client sends the request to the local SIS asking for rating the possessed list. The SIS uses the algorithm for rating the list. The rated list is delivered to the client.

In the following section, we will present a detailed example for how this inter-SIS communication may influence the rating of remote peers, and what can be achieved by using the information gained from remote SIS servers.

4.1.2 Rating Scenarios

The algorithm for rating peers based only on the BGP information from the remote AS (described above in Section 4.1.1) can be especially useful in the case when an SIS is located in the remote AS. This situation is different from that when the LGS is present in the remote AS. The remote SIS can deliver to the local SIS some additional parameters describing the rating preferences established by the operator in the remote AS. These parameters from the remote SIS can have different origin than BGP; they can take into account monetary or internal routing preferences. The mentioned algorithm can also use LGS but these servers can only send BGP information.

In this section we consider three scenarios for rating peers, in which different parameters are taken into account.

In all scenarios presented in this section we use a few parameters for rating peers, we call them rating parameters. Each peer has been assigned specific values of these rating parameters. The list of peers is rated using these parameters. The operator decides on the order for applying these parameters for the final rating, this way it can define different rating scenarios. We also introduce the rating formula in order to use one general rating number for software implementation purposes. Let us suppose that we use for rating parameters $p_0,...,p_{i-1},priority,p_{i+1},p_n$ (priority parameter is described in the Appendix D). The parameters are ordered, the p_0 parameter is the least important and the p_n is the most important one. The rating number is defined in following way:

$$R = \delta_{priority,0}\,(A^0\,p_0 +..+ A^{i+1}\,p_{i-1} + A^i\,priority + A^{i+1}\,p_{i+1} + ...+ A^n\,p_n) + \delta_{priority,3}\,A^{n+1}\,priority,$$

where

$$A=10^{\log(\lfloor \max\{p_1,...,p_{i-1},\,p_{i+1},...,p_n\}\rfloor+1)}.$$

In the first scenario we use the algorithm presented in the Section 4.1.1. This algorithm produces the rated list of peers. In this section we use some example autonomous systems connected as presented in Figure 12. In this example, the Internet cloud hides an arbitrary number of ASes.

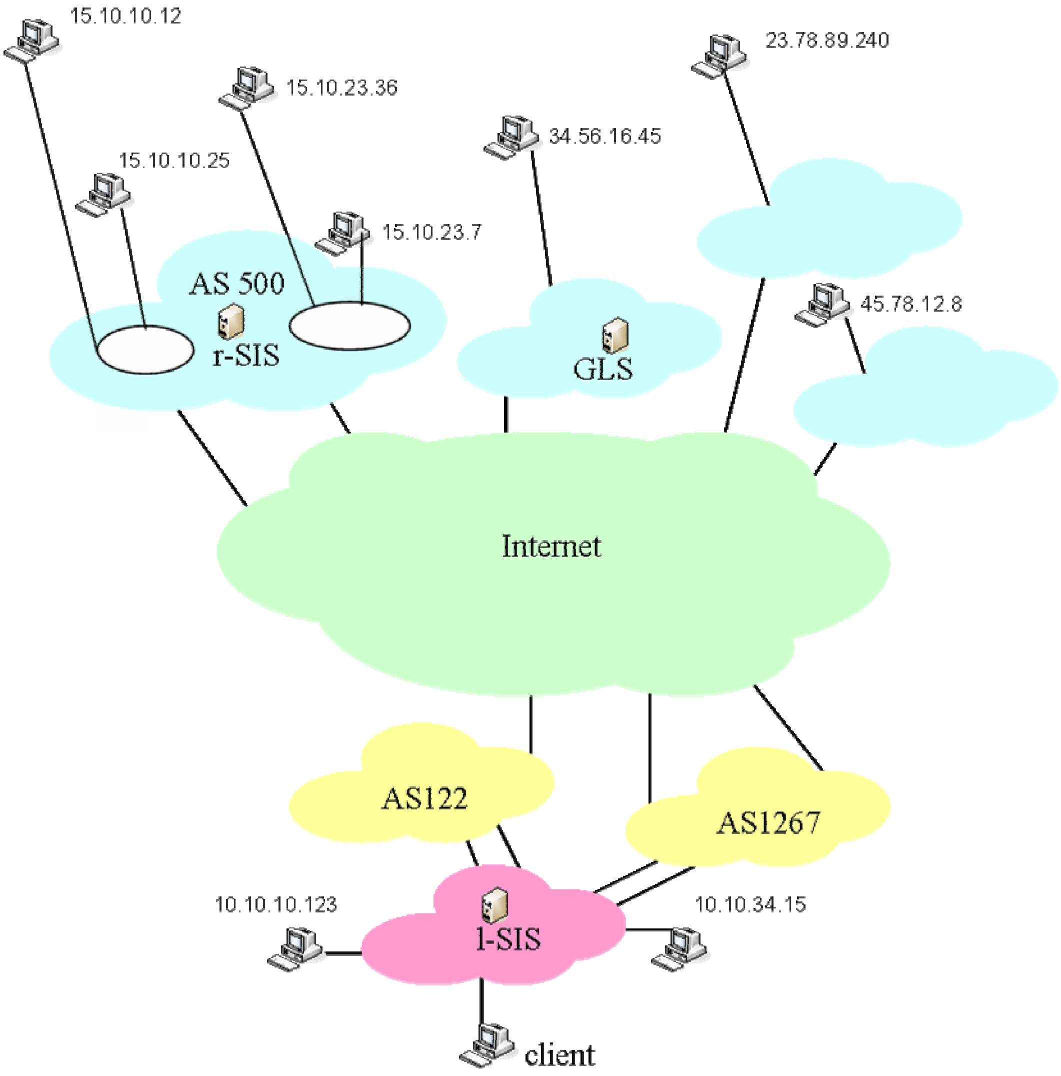

Figure 12: The example connections between ASes and the locations of SISes and the LGS

Table 6 includes the rated list of peers with parameters used by the algorithm. In this example the first rating parameter is metric and it is equal to AS_PATH_hops, the IGP path is not used. From this moment for simplicity we suppose that the IGP metric is equal to zero. Only the two first rating parameters (priority and metric) are used for the peer rating, the primary key is priority.

Table 6: The example rated list of peers with parameters

peer IP address	priority	AS_PATH_hops	neighbor_AS	remote_SIS	R
10.10.10.12	0				0
10.10.34.15	0				0
15.10.10.25	1	3	122	YES	13
15.10.10.12	1	3	122	YES	13
15.10.23.7	1	3	122	YES	13
15.10.23.36	1	3	122	YES	14
34.56.16.45	1	4	1267		14
45.78.12.8	3				300
23.78.89.24	3				300

The other parameters, neighbor_AS and remote_SIS, are not used for rating, they have informal meaning. In this scenario, the only attribute sent by the remote SIS is AS_PATH. Parameters neighbor_AS and AS_PATH_hops are derived from AS_PATH. The peers belonging to the same AS receive the same rating values (in our example peers in AS 500, connected via AS 122).

If we want to take into account the transit agreement between ISPs, we consider the second scenario where the parameter neighbor_AS plays a crucial role. The parameter neighbor_AS represents the number of the directly connected autonomous system through which traffic from a particular peer comes. In this scenario, our AS can be connected to multiple ASes. Our ISP can prefer receiving traffic from one AS rather than from another AS. It can assign some priority values for directly connected ASes. We call this rating parameter AS_priority. This rating parameter has only local significance and cannot be compared with AS_priority in other ASes. The lowest AS_priority is the most preferred, the lowest limit for AS_priority value is 0. The ISP performs the mapping of the neighbor_AS parameter to the AS_priority rating parameter.

Peers with priority 0 are still the most preferred, since they are in the local AS. The peers with priorities 1 or 2 are rated according to new key parameters. The rated list is created using the AS_priority as the first and most important criterion, the priority parameter is used as a secondary criterion and the last criterion is AS_PATH_hops. For peers with priority 3 there is no information about neighbor ASes through which traffic comes, so these peers are the least preferred. In our example let us assign AS_priority=2 to the neighbor AS with number 1267, and AS_priority=4 for AS 122. The previously prepared list will be reordered in the way presented in Table 7.

We do not limit our consideration to the case when the SIS is present in the remote AS. The importance of the neighbor_AS parameter has already been stressed. This parameter can represent the charging regulations for inter-domain traffic between ISPs. As already mentioned in Section 4.1.1., this parameter can be acquired from remote SISes or LGSes, and in some special cases from RIB-out table (accessible on the local AS BGP routers).

Table 7: The example rated list of peers which take into account the preference of neighbor AS

peer IP address	AS_priority	priority	AS_PATH_hops	neighbor_AS	remote_SIS	R
10.10.10.123		0				0
10.10.34.15		0				0
34.56.16.45	2	1	4	1267		214
15.10.10.25	4	1	3	122	YES	413
15.10.10.12	4	1	3	122	YES	413
15.10.23.7	4	1	3	122	YES	413
15.10.23.36	4	1	3	122	YES	413
45.78.12.8		3				3000
23.78.89.240		3				3000

In Table 7 the information about the peer with address 34.56.16.45 comes from the LGS. For better distinction we have introduced an informal parameter remote_SIS that signifies whether the information was gained by querying another SIS server or from other sources. This parameter is not used for rating procedures and it is not used in any algorithm. The peers belonging to the same AS are ordered in the random way (in our example peers in AS 500, connected via AS 122).

Now we are going to consider the most interesting scenario when the remote SIS responds with the AS_PATH and the list of peers with some rating parameters indicating the preference which is suggested by the remote SIS. This preference parameter has only local meaning and can not be compared with other ratings performed in other ASes, but can be used by the local SIS for rating peers located in the same AS.

Let us consider the example of a remote AS in which there are two sub-networks, one of them is preferred by the remote operator for charging reasons (also described in more detail in Section 4.3.2). If the remote SIS receives the request from the local SIS with the list of peers for which AS_PATH is needed, the remote SIS sends AS_PATH and the list in which each peer has been assigned some priority value. These priority values are established by the remote SIS. The rules for an assignment can have different origin. The name for this priority parameter is remote_priority. The lowest value is the most preferred and the lowest value limit is 0.

In our example we suppose for AS 500 that the cheaper sub-network is 15.10.23.0/24 with remote_priority=10 and the expensive one is 15.10.10.0/24 with remote_priority=20. Table 8 includes peers after rating procedure; the peers belonging to AS 500 are rated inside their subset according to the remote_priority rating parameter. Thus, the rating considering the BGP asymmetry and the rating considering the topology of remote ASes can be combined.

The schema for rating is open; it does not limit the number of parameters that are used for rating. The remote SIS can, in theory, deliver an arbitrary number of available parameters, which can be applied for rating. The order of keys for rating is to be established by the local SIS, but the remote SIS can indicate the application order of these parameters. Thus, it can signal which of these parameters it finds more important or useful for optimization from its own point of view. However, the final decision still belongs to the local SIS.

Let us consider the following example: A remote SIS sends AS_PATH and two additional parameters for rating in the order AS_PATH, remote_priority and bandwidth. Bandwidth was requested by the local SIS and remote_priority was added by the remote SIS. The order of these parameters indicates that if the local SIS respects preferences vital for the remote AS, the primary key, for rating peers located in this remote AS, are parameters derived from AS_PATH, the secondary key should be remote_priority and the last key is bandwidth. The local SIS uses AS_PATH_hops (which follows from AS_PATH) as the primary rating key and has the right to decide that the secondary rating key will be bandwidth and remote_priority will be ignored. We can imagine that the client application requires high bandwidth, so the local operator wants to give the best available conditions for its client. On the remote side it can do this by offering the peers with greater bandwidth.

If the local operator does not require high bandwidth for its clients it can help the remote ISP and use as the secondary key remote_priority and as the last key bandwidth. In other words, the preferences of local AS (local ISP) are always the most important. The local SIS can request some parameters from the remote SIS. The AS_PATH parameter is the only obligatory parameter which is sent by the remote SIS. If the remote SIS does not want to expose to the remote AS complete connections, it can send two parameters derived from AS_PATH instead of the AS_PATH itself: neighbor_AS (neighbor of the local AS) and AS_PATH_hops. The remote SIS can send more parameters than there are requested, indicating that they can be used for rating procedures.

Table 8: The example rated list of peers. Peers belonging to AS equipped with SIS are rated according to four rating keys

peer IP address	AS_priority	priority	AS_PATH_hops	remote_priority	neighbor_AS	remote_SIS	R
10.10.10.123		0					0
10.10.34.15		0					0
34.56.16.45	2	1	4		1267		20104010
15.10.23.7	4	1	3	10	122	YES	40103010
15.10.23.36	4	1	3	10	122	YES	40103010
15.10.10.25	4	1	3	20	122	YES	40103020
15.10.10.12	4	1	3	20	122	YES	40103020
45.78.12.8		3					3000000000
23.78.89.240		3					3000000000

In the first two scenarios we use rating parameters AS_priority and remote_priority. These parameters can be changed dynamically, the operator can change the AS_priority according to the actual situation in the operator's network, some links can be congested or the balance between download and upload exceeds the limit. These parameters can be used for the dynamic locality mechanism. By changing AS_priority the operator can move P2P traffic from one interface to another. The operator can also work on a finer granularity level, limiting the traffic from particular peers, locally increasing in the SIS database the AS_PATH_hops rating parameter for some particular peers. These simple examples does not exploit all possibilities of influence on P2P traffic, there is great flexibility in composing a different schema for rating peer lists.

The approach, in which the AS_PATH parameter from the remote AS is used, is the only way for the local AS to gain precise information about AS hops for download. The algorithm from Section 3 can deliver the wrong number of AS hops in case of route asymmetry. Moreover, if the local BGP information is used, we can wrongly identify the neighboring AS through which the download traffic goes. That can have substantial influence on the transfer cost calculations. By using AS_PATH from the remote AS we can exactly discover this neighboring AS, without introducing any ambiguity. The presented approach allows the local side to request some parameters that enable the rating according to these parameters. Also the remote side on their own can suggest which parameters and in what order should be used to improve their traffic conditions. The sides can send the parameters that do not have the routing origin or some specific local preferences.

The presented rating scenarios require SIS-SIS communication; in the separate section 4.3.1, the requirements for this communication protocol are defined.

In summary, there are three basic rating scenarios for choosing the best peering partners. As described above, any ISP can compose its own rating scenario including specific information. The most important thing in this approach is that by communicating with the remote SIS, the operator can exactly establish ingress interface to its autonomous system. The last rating scenario is the most promising; in that case the local ISP can choose the best peers not only from its point of view but the choice can take into account the remote operators conditions.

4.2 Information Asymmetry

In the current specified ETM mechanism, a peer queries its domain SIS (the SIS server which is responsible for all peers in the domain) for rating information about other peer contacts. This means that the retrieved information is generated from the viewpoint of that source SIS server, which should normally be in the same AS or at least provider domain as the peers contacting it.

The basic considered scenario is that there exists an information asymmetry between the different deployed SIS servers. One SIS server might have detailed underlay information about the peers or the network in its domain, but it lacks the same level of detail for peers in other domains. Therefore, it has to cooperate and share information with the destination SIS servers in these other domains if it wants to provide a better rating for those peers.

To formalize this, we define the SIS rating function $f_X(p_m, p_n, a_1, a_2, a_3, \dots, a_n)$ of source SIS server X , with $a_1, a_2, a_3, \dots, a_n$ being the different underlay attributes an SIS server uses in its rating function for the connection between p_m and p_n, e.g., a_1 being the AS hops on the route from p_m to p_n, a_2 being the local preference value, a_3 being the MED value, a_4 being the number of AS hops on the route from p_n to p_m, and so on. X can directly access the values of all these attributes for every peer p_n in X's own domain via its metering modules (we assume that p_m is the peer that queried X and is therefore also in X's domain). In contrast, it may only be able to collect values for a subset $a_1, \dots, a_k$, $k < n$, for peers p_n in different domains by itself (cf. Figure 13: Information asymmetry).

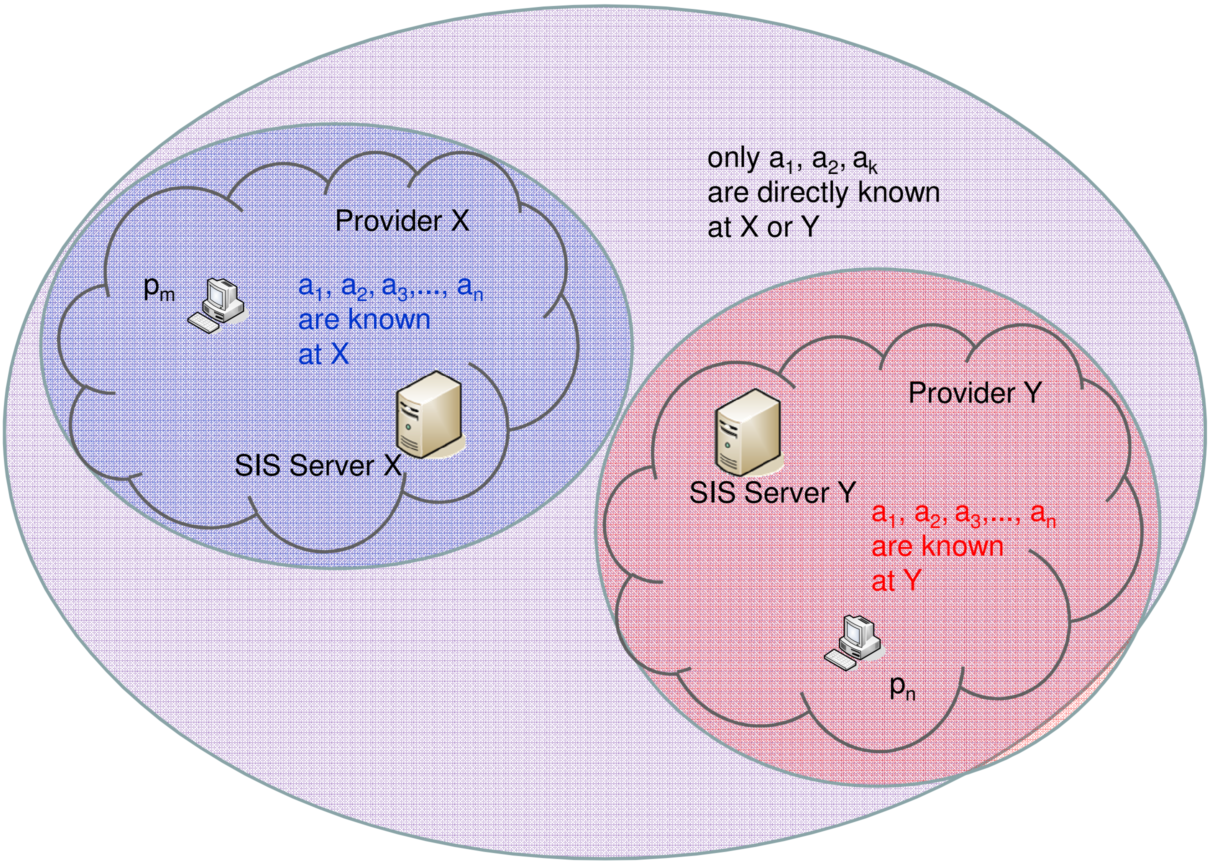
only a_1, a_2, a_k are directly known at X or Y
Provider X
p_m
$a_1, a_2, a_3,..., a_n$ are known at X
SIS Server X
Provider Y
SIS Server Y
$a_1, a_2, a_3,..., a_n$ are known at Y
p_n

Figure 13: Information asymmetry

The remaining attributes a_{k+1}, a_{k+2}, a_{k+3},..., a_n may now be obtained by querying the destination SIS servers responsible for these peers. If these SIS servers cooperate fully, all of these values can be obtained, probably in a preprocessed and abstracted form. Depending on the degree of cooperation, also only a subset of a_{k+1}, a_{k+2}, a_{k+3},..., a_n may be returned, or no values at all (also in case there exists no SIS server for that domain). In this case there needs to be a default policy how these missing values are interpreted and used in f_X. This would also enable a case where the rating function is different/uses different attributes in different SIS servers. The parameters p_m and p_n are included to maintain generality. If a provider has more than one AS in its domain, the location and therefore the identity of the querying peer p_m could influence the rating function. Similarly, certain attributes might only be evaluated for a specific p_n (e.g., in case of peering agreements).

An example would be an SIS server of provider X that has access to information about the access bandwidths of peers in X's network (a_1) and uses this in the ETM rating algorithm. This information is not directly available for peers outside of X's domain. It is also unlikely that this raw information is shared between providers. However, an SIS server of provider Y might offer preprocessed information to SIS server X that can be integrated directly in its ETM rating function. This would enable an information exchange without disclosing too much sensitive information. An example would be a normalization of the access bandwidths to a maximum value chosen by provider Y.

An example that directly relates to the specified BGPLoc ETM mechanism is a distinction between autonomous systems in the domain of a single provider, such as described below. Another example is the asymmetry in BGP routing, where the path from p_m to p_n (a_1) might have different length than the path from p_n to p_m (a_4), for which a solution was presented in the previous section. After we discuss the interfaces and the protocol for data exchange in the following section, we will describe additional scenarios where this information asymmetry also necessitates the cooperation between SIS servers.

In case when it is best to avoid providing information on the attribute level, even in abstracted form, a simpler but probably less effective solution would be to just exchange the final SIS rating values between the different SISes. Here, if SIS server X requested additional information for the connection between p_m and p_n from SIS server Y (in whose domain p_n is located), then Y would simply return $f_Y(p_n, p_m, a_1, a_2,\ldots, a_k)$, i.e., the rating that is locally available for the connection from Y's point of view. In this case, a merging function between $f_X(p_m, p_n,a_1, a_2,\ldots, a_k)$ and $f_Y(p_n, p_m, a_1, a_2,\ldots, a_k)$ needs to be defined. This gets more complex if f_X and f_Y are different functions using different attributes. However, the simplest solution for this, that sacrifices some detail information, would be to treat $f_Y(p_n, p_m, a_1, a_2,\ldots, a_k)$ as another attribute for f_X to consider. In case both SIS servers use the generic rating described in Section 3.1, an intermediate step would be possible where only one value per rating category needs to be exchanged.

In the following, we will describe first some basic considerations for the SIS-SIS collaboration which should be of interest for all scenarios we consider. Then, we will describe the most interesting of these scenarios in more detail.

4.2.1 Interfaces, Protocol and Required Data

4.2.1.1 SIS-SIS Interface

The *Inter-SIS* interface was specified in D3.1 (there it was called *SIS server - SIS server interface)*. Under the assumptions made in 4.2, the basic communication between SIS servers is similar to the communication between clients and an SIS server. The SIS server lacking specific information acts as a client and queries this information for a list of peers from another SIS server, using the Inter-SIS interface. This mainly applies to the runtime communication between SIS servers, on which we focus here.

4.2.1.2 Communication protocol

The format of the queries and responses may change, however. If there is a close cooperation between two providers, they may exchange rating values for the single attributes that are used in the SIS server deployed in each ISP. Also, the querying server should provide both p_m and p_n.

In the query of SIS server X to SIS server Y, X has to specify which attributes it needs to be rated for the connection between p_m and p_n. Thus, the query would have a format list of <peer address 1, peer address 2, <list of <attribute>>>.

The response then includes more detailed information than just one single rating value per peer pair. If more than one underlay metric is used in the ETM algorithm on an SIS server, e.g., BGP route characteristics, peer capacity, domain-internal traffic information, then all these values a_{k+1}, a_{k+2}, a_{k+3},..., a_n are reported separately per peer pair (if necessary in an abstracted form).

Additionally, it might be possible to add weights to the different values, if they are used in a generic ETM algorithm (see Section 3.1) or just to rate the importance of the provided attributes, as well as information about the range of the value at the providing SIS. These weights are a means of the ISP providing the information to signal its own preferences for optimization. Finally, the responding SIS server may specify whether larger or smaller val-

ues for an attribute are to be preferred, in case this is not regulated by a global convention.
This would change the format of SIS-SIS messages to a list of <peer address 1, peer address 2, <list of <attribute, value, weight, range, best value>>>.
Another issue is the mapping of peers to the SIS server that can provide the most detailed information about them. In case there is only one SIS per provider, this could be easily done via the IP address ranges assigned to the different providers and a list of SIS servers per provider. In case there are several SIS servers per provider, or if cooperation should take place only with selected providers, a mapping of peer addresses to SIS server addresses should be exchanged between all providers willing to cooperate.

4.2.1.3 Community-SISes: Information about the Location of the SIS in the Second Party AS

In order to start communication, the local SIS needs to know if a remote AS is equipped with an SIS. A certain procedure that allows interested operators check if second parties possess SISs is suggested. Furthermore, there should be some web page where an operator can register his/her SIS.

The access to that database should not be restricted. Those operators who want to improve their P2P traffic, should register their SISes on this web page. The database entry can have the form: AS-number with SIS-IP-address. It would be not obligatory to register the SIS.

If an operator registers its SIS, it still will have the opportunity to limit the communication with other SISes. The exchange of information between SISes should be based on policy procedures.

Each SIS can have an input policy and an output policy. By the input policy operator can decide from which autonomous system or from which SIS it accepts requests. By the output policy operator can decide to which autonomous system or to which SIS it responds. Also in polices it can be specified what type of information is allowed to be exchanged amongst ISPs.

Polices can be more sophisticated, even we can imagine that some routing polices can be mapped on SIS polices.

Operators can create SIS communities. By a community we understand a set of SISes which cooperate with each other supporting P2P traffic. The operators belonging to a particular community declare a type of exchanged information. For instance they can decide that they send each other the full AS_PATH attributes. In the Internet there can be many SIS communities and they don't need to intersect. Using one specific SIS policy, an operator can cooperate in P2P traffic with some group of operators adjusting to some rules, but using a second policy, the same operator can treat P2P traffic in another way in communication with other group of operators. In the policy, the operator defines to which community a particular AS belongs.

This method seems to be very flexible. One operator can have many SISes and they can cooperate in the frame of one community – internal operator community. This architecture can be useful when SIS load balancing is considered. Simultaneously the SIS belonging to the internal operator community can be a member of another community which operates on a larger scale – inter operator community.

Some operators can create a community in order to apply common policing procedures for these operators.

There can be also the scenario when operators possess trusted honey-pots, the access to these honey-pots can be granted only by this operator SIS. Let us imagine the situation when this operator signs contract with another operator and allows access these trusted honey-pots from a foreign AS. In order to achieve this goal, these two operators create community and appropriate polices which allow to access required resources through SIS-SIS communication. If some new operators want to share their trusted honey-pots, they simply join this community and write adequate polices.

The policy schema can be applied not only to the BGP information exchange (peer rating algorithm based on SIS-SIS communication) but also for QoS information exchange. In Figure 14 we present three communities, the same SIS can belong to different communities.

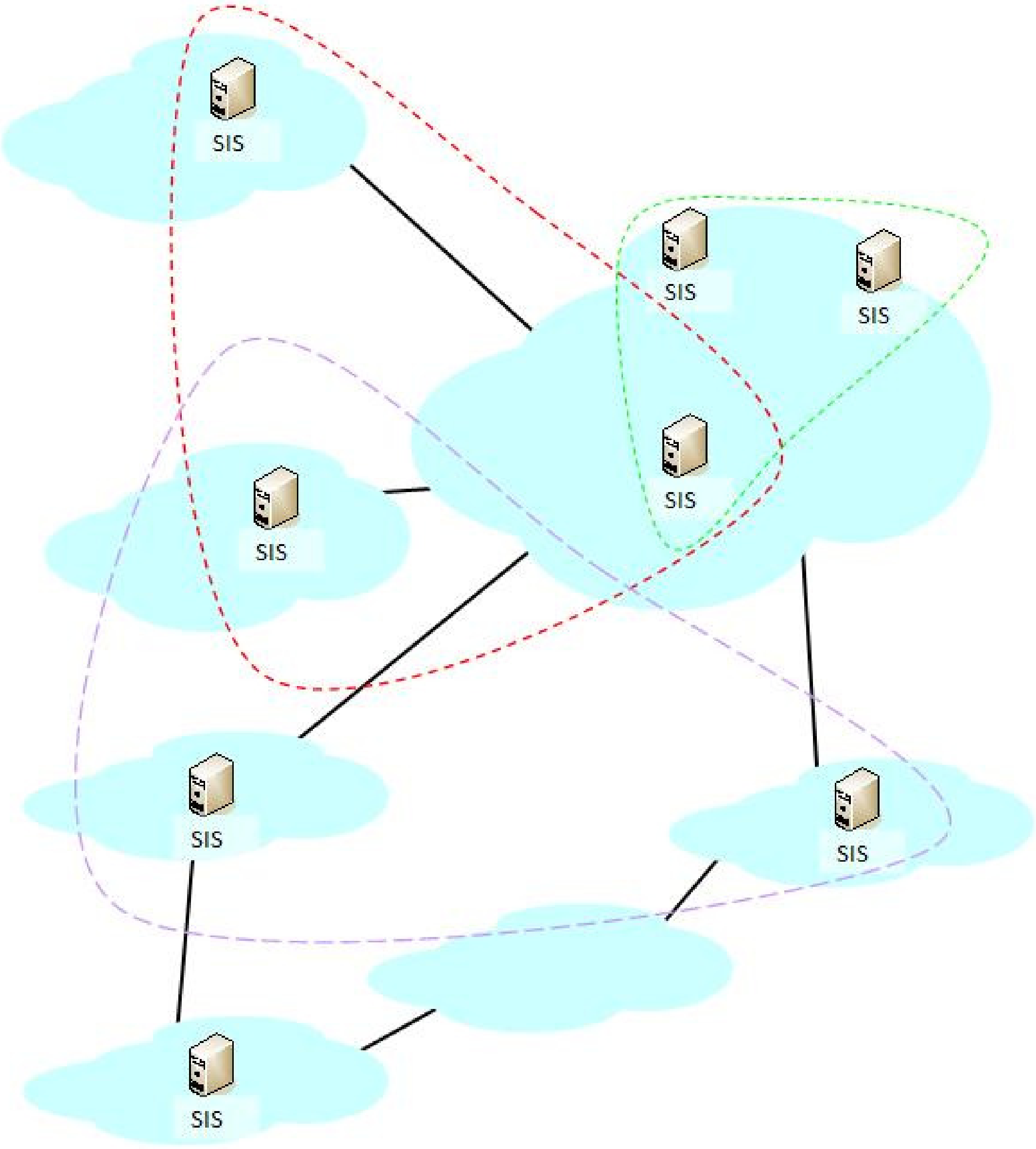

Figure 14: Example of community structures

4.2.2 Access Bandwidth and Local Preference of Peers

In this scenario, there is some underlay information that a local provider's SIS may have, but which is not directly attainable by a remote SIS server. The two domains X and Y, as in the general description above are assumed again. The considered underlay information is the topology within these domains, in particular the assignment of peers to PoPs and the access bandwidth of peers.

The SIS server Y, in this scenario, offers an additional rating attribute that denotes how peers at different PoPs are preferred by Y as being neighbors in the swarm. A reason for peers not to be favored by Y can be higher costs to route traffic to them within Y's domain, e.g., if they are located on an island that is connected via expensive submarine links. By advertising other peers in form of the additional rating value, Y can reduce the traffic flowing to and from these expensive peers.

This, of course, assumes that other SIS servers, i.e. X, uses this additional rating attribute properly when being queried by the peers in their own domain. Provider X might want to support Y by doing this because it has a peering agreement with Y, or because it reciprocates the same behavior from Y. The peers in X's domain may not directly profit from this (or even be at a disadvantage because they lose valuable options for connections), since they might not see any difference between neighbors in Y's domain that are favored and those that are not. In general however, we assume that X is completely free to use the information provided by Y in any way it wants to influence the rating values of peers in Y's domain. It may also ignore the additional attributes provided by Y.

A case where the peers in X's domain might have an advantage from X using underlay information provided by Y is if Y offers information about the access bandwidth of its peers to X, also in form of a rating attribute. The peers querying X can then, due to a higher rating, prefer connections to peers in Y that offer a higher upload bandwidth. It remains to be seen how this will affect the download performance of peers with less capacity or how the inter-domain traffic will change. It may also be a good strategy to prefer peers with less bandwidth, e.g., in case of mobiles, for connections outside the local domain.

The communication between the different entities would, during runtime, be as follows.

1. A client p_m queries its local SIS server X to rate the connection between p_m and a remote peer p_n.
2. X looks up the SIS server Y responsible for p_n (and possibly the attributes offered by Y) in a mapping service for SIS servers (to be defined).
3. X queries Y for the attributes 'PoP preference' and 'Access capacity' for p_n.
4. Y responds with the values, weight, range and best value for both attributes.
5. X computes the rating of $<p_m,p_n>$ based on its locally available information (e.g., BGPLoc) and the values for 'PoP preference' and 'Access capacity'. Since 'PoP preference' and 'Access capacity' are used to discern peers within an AS, the rating might take place in two steps. First, the 'original' (e.g., BGPLoc) rating is done, then 'PoP preference' and/or 'Access capacity' are used as a secondary rating to discern between peers with the same primary rating.
6. X returns the final rating to p_m.

The query from X to Y may be combined with other scenarios, such as the AS_PATH_hops query in Section 4.2.

4.2.3 SIS collaboration Between Peering ISPs

Peering agreements between ISPs are used in order to provide connectivity to each other's customers. This form of business relationship is considered to be one of the most effective ways for an ISP to improve the efficiency of the operation of its network. Using peering agreements the ISPs reduce the inter-domain traffic latency as well as their reliance on the transit ISPs for exchanging traffic and thus they reduce their implied costs.

This scenario concerns the collaboration of the SISes that belong to ISPs that have a peering agreement between each other, which henceforth, are referred to as peering ISPs. Figure 15 presents an abstract network topology of many ISPs of different Tiers, who are distinguished by each other from their id. In this scenario, two Tier 3 ISPs (3A and 3B in the figure) have a peering agreement and also both of them are connected by symmetric transit links to a Tier 2 ISP (2A). The traffic that is exchanged through these transit links impose charges to Tier 3 ISPs by Tier 2 ISP unlike the traffic that they exchange due to their peering agreement.

The objective of the collaboration between Tier 3 ISPs is directly connected to their necessity of reducing the inter-domain traffic charges. This in combination with the potential for improved performance for both the peering ISPs when using the peering link provide them the incentives to collaborate further in order to exploit the opportunity of reducing the redundancy in the downloading content from non-local peers (or ISPs). The main idea behind this scenario is that since SISes use BGP locality in order to rate the peers that belong to a swarm (and in the list that is sent by the local peer), they will rate higher the peers of the peering ISP. Thus local peers would prefer to download content from the peering ISP (if available); only in the opposite case they would prefer non-local peers. This will eventually lead to a reduction of the duplicated content that is downloaded from the non-local ISPs and through the costly transit links.

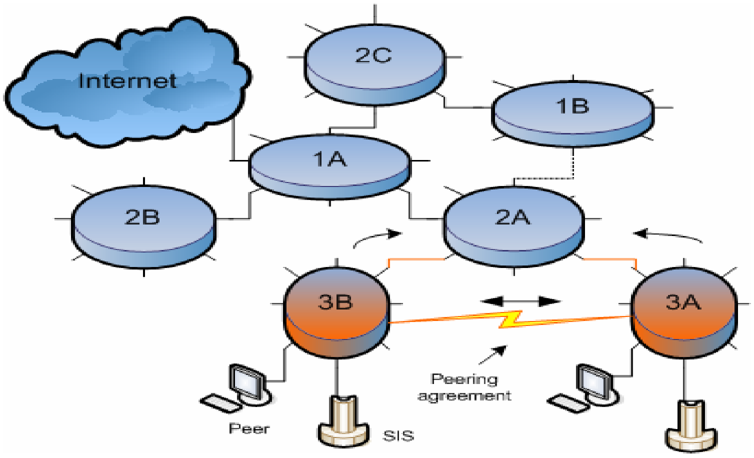

Figure 15: SIS collaboration between peering ISPs

In the direction of reducing the download of redundant (duplicated) content through the transit links three different approaches have been considered:

- Splitting the content between peering ISPs and download different subset of chunks each
- Splitting the remote ASes between peering ISPs and thus download from different ISPs each
- Use an enhanced model of SIS

The aforementioned approaches are further elaborated in the following sections.

4.2.3.1 Splitting Content

In this case, the peering ISPs can share costs by deciding to split and download different parts of content through their transit links and then exchange the rest of the parts via the peering link.

Tier 3 ISPs (namely 3A and 3B, according to Figure 15) first decide that they will split the content that is required from both of them. Independently of the local peer number of each ISP, it is in their both interest to split the content evenly, because even if there is only one peer in an ISP it will eventually download all the chunks. Also due to the peering agreement that they have established both the ISPs should try not to be unfair to each other. Thus we derive the conclusion that they will confront with the decision of splitting all common content. An evaluation of this case and certain related policies using a simulation tool will be performed in our future work.

Moreover ISPs have to decide which chunks of the content each of them will download from remote ASes. The chunks can be distinguished according to their ids in even and odds or first half and second half or any other similar way. The rest of the chunks that each ISP does not download from the remote ASes will be retrieved from the peering link. Also local peers can exchange chunks, regardless of the ids, in order to promote locality. In this step, the SIS assists the client with its main operation tool. It rates the peers in the list received from the client using BGP locality giving the client the choice to connect to better neighbors.

In order to implement the splitting content scenario, a series of modifications is required. Every peer in a peering ISP downloads specific chunks according to the peer that it is connected to. For example, in Figure 15, if 3A has agreed to download the even numbered chunks from the remote ASes, then the peer that belongs to this ISP has to download even chunks if it is connected to a remote peer and odd chunks if it is connected to a peer from the peering ISP.

Firstly, every peer needs to be informed about what chunk is authorized to download from remote peers. This information can be retrieved from the local SIS. Local peers and SIS collaborate in order for the first to retrieve a list of peers with the additional information of which of them is preferable to be connected to. In this approach they collaborate further as the local peers have to be informed about the chunks that are authorized to download from the remote peers. The incentives for such collaboration will be examined in our future work.

The next modification is taking place in the way the client expressing its interest in new available chunks that exist in its neighbors (peers that it is currently connected). After acquiring the list of the new available chunks of a neighbor, the client has to decide if it is interested in being unchoked by this peer. Thus it needs to know if the peer possesses any chunks that belong to the subset of chunks that it is authorized to download from the specific peer. The peer of ISP 3A, in Figure 15, checks if there are any even chunks available, if the neighbor peer is a remote peer, or if there are any odd chunks available, if the neighbor is a peer from the peering ISP.

The last modification concerns the method that the client uses in order to ask which chunks to download after it has been unchoked by its neighbor. A similar to the aforementioned approach is being followed. The client asks for specific chunks according to which AS the connected neighbor belongs.

As each peering ISP will download different chunks using its transit link, it will be charged less by the upper Tier. In fact since, in this approach, the content is being split in half and the peering ISPs download non duplicated content, Tier 3 ISPs will be charged for only half of the content. An evaluation of this case and certain related policies using a simulation tool will be performed in our future work.

4.2.3.2 Splitting ASes

In the aforementioned approach the client had to be modified in order to achieve the biased chunk selection methodology. In this approach we focus on splitting the remote ASes from which the peering ISPs download the content, thus avoiding the modifications to the client. As in the previous approach the ISPs have the incentives to split the ASes regardless the number of their peers that exist in the swarm and also to be fair to each other due to the peering agreement they have established.

Splitting the ASes can be done once and for all when the peering agreement is being established. Using this alternative the peering ISPs do not have to decide from which remote ASes they will download for each specific swarm. However this is a very simple way of splitting the ASes and may not be very efficient and fair. For example if the ISPs have decided that they will split the ASes to even and odd, there is a possibility a swarm to have more even than odd ASes. To solve this, a more dynamic approach through SIS collaboration can be adopted. Both approaches can use one of the two following ways of splitting.

The split of the ASes can be done either randomly or after further collaboration regardless of which of the aforementioned approaches has been adopted. In the random approach SISes create a list of remote ASes and then they split that list randomly (for example even and odd ids). Nevertheless this can be unfair for the ISPs since neither the business agreements nor the AS_hops are taken into consideration. An improvement may be achieved by splitting ASes that are the same number of hops away from the Tier 3 ISPs (since they both belong to the same upper Tier ISP). The collaborative approach is more sophisticated. The SISes can create a list of the remote ASes and rate those ASes in a way that reflects their values for them which can result from their business agreements, the AS_hops, transmission latency etc. The lists of the rated ASes can be exchanged and merged. In both cases after deciding the remote ASes from which each of the ISPs will download, two different lists have been created with half of the size of the initial one. The SISes can use these lists in order to rate the peers in the lists that they receive from the local peers. Certainly local peers will have the greatest ratings, then peers from the peering ISP follow and finally peers from remote ASes according to the new list of ASes that has been created and to BGP locality mechanism.

The main difference of this mechanism compared to the one of splitting chunks is that the download of non duplicated content from remote ASes cannot be guaranteed. Thus the costs for using the transit link may not be reduced as much as in the previous approach. An evaluation of the splitting ASes approach using a simulation tool will be one of our next steps in order to specify the differences in costs and performance.

4.2.3.3 Enhanced SIS

In this approach the participation of the SIS is more active gathering information on the chunks that every local and remote peer has stored in the SIS. Thus, the SIS could guide every local peer to download specific chunks from specific peers and achieve downloading the content through the transit links only once thus reducing the costs.

Since the SIS knows the chunks that every peer has, it can provide each client with a different subset of peers according to the needs of each client. Those subsets may contain the preferable peers and the ratings the SIS has set for them according to BGP locality mechanism. SIS collaboration is necessary in this approach in order to achieve minimization of downloading redundant content. SIS can exchange information about the available chunks in local and remote peers and decide about which peers and chunks every SIS will promote to its local peers, thus avoiding downloading of the same chunks from both ISPs and even minimizing the downloading of duplicate chunks for the same ISP.

This approach requires a complicated operation of the SISes. They can provide a list of peers that have specific subset of chunks or at the extreme case, one list per missing chunk. In our next steps we will use a simulation tool in order to elaborate further the performance of this approach. The results of the simulation experiments can serve as a benchmark to evaluate the previous approaches.

4.2.4 SIS Collaboration Between Source and Transit SIS (SIS of Transit Provider)

As we already mentioned in the scenario above, ISPs (of Tier 2 and Tier3) may have contracted two different types of business agreements. They may be connected to each other with peering agreements (usually ISPs of the same tier) or with transit agreements (usually ISPs of different tiers). Those transit agreements are imposed on the transit links and the traffic exchanged through this link is subject to charges by the upper Tier.

The network topology of interest is presented in Figure 16. 3B is a Tier 3 ISP who is connected to 2A, a Tier 2 ISP, using a transit link and it is charged from 2A for exchanging traffic through this link. On the other hand Tier 2 ISP 2A is connected to Tier 1 ISPs 1A and 1B through transit links and it is charged for exchanging traffic through the respective links based on the difference between inbound and outbound inter-domain traffic. Finally 2A has contracted a peering agreement with 2B ISP and it is not charged for the exchanged traffic through this link.

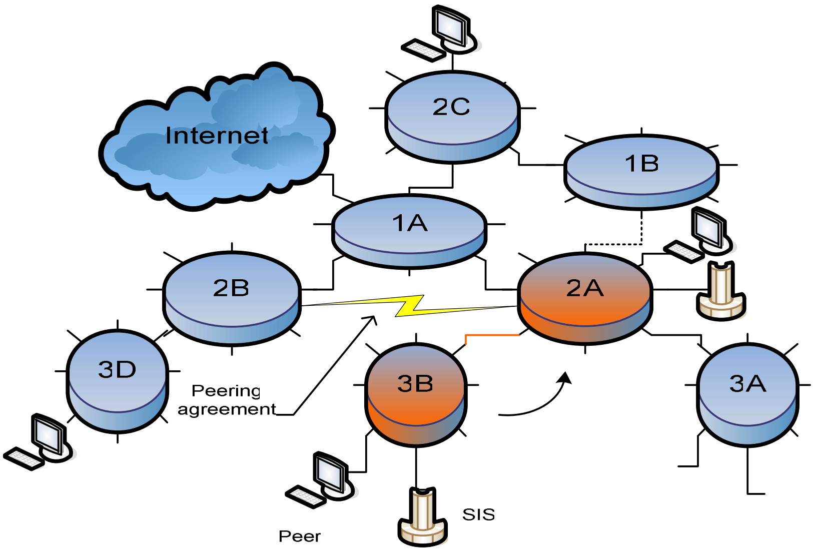

Figure 16: SIS collaboration between source and transit ISP

In this scenario the collaboration of SISes in Tier 2 ISP (2A) and Tier 3 ISP (3B) is studied with main objective being the reduction of downloading content volume from transit ISPs 1A, 1B and thus the enhancement of locality in Tier 2 (2A), its own customers, either Tier 3 ISPs or end users. The peering ISP 2B can also benefit from this approach, either indirectly or even collaborating with 2A as explained in previous scenarios. The Tier's 3 SIS is encouraged to ask Tier's 2 SIS for information about how to rate particular ASes.

For example SIS of 3A may have rated similar to a peer from 2C ISP and a peer from 3D ISP, as shown in Figure 16, using BGP locality information. The BGP locality information that SIS has reflects through the local preferences the business agreements of the ISP and how long the path is through the AS_hops. This is a metric that the ISPs that belong to the specific route can change reflecting their preferences over the route when they advertise it to other ISPs. Thus usually the lower Tier ISPs follow the preference of the upper Tier ISPs. However in case of BitTorrent traffic maybe the 2A ISP cannot reflect its business agreements unless it uses SIS collaboration. In our example the 2A ISP has a peering agreement with 2B and an agreement of using an approach of the previous scenario for BitTorrent traffic. Thus for BitTorrent traffic it may prefer the peering link than the non BitTorrent one. This information can be exchanged through SIS collaboration. The information that may have to be exchanged should reflect the value of the specific AS to the Tier 2 ISP.

As already mentioned above, the main objective of this scenario is to reduce the downloaded content volume through the costly transit links for Tier 2 ISP. This provides it with a monetary incentive to establish an SIS and collaborate with the Tier 3 SISes. On the other hand Tier 3's SIS does not have a clear incentive in order to collaborate. Tier 2 ISP can provide each Tier 3 ISP with monetary gains as a motive to collaborate in terms of, for example, discounts on the prices posed in the inter-domain traffic link. This scenario will be further evaluated by simulations in order to check the actual monetary and performance benefits Tier 2 and Tier 3 may gain for such collaboration between their SISes.

4.2.5 AS-Quality Self-rating Scenario

A case for inter-SIS communication that is less provider-centric is one comprising a service where peers can rate their overlay connections and report these findings to an SIS that is provided by a third party. Thus, an SIS server can collect user experiences, e.g., slow or unstable connections, for remote peers and aggregate these to create an experience-based rating of ASes. It can then base its own recommendations on these ratings. We will refer to this kind of SIS as *overlay SIS* in the rest of this section.

Since the collected user experience can only reflect underlay information that might be directly provided by any of the other mechanisms described above, it can be seen as an extension to these mechanisms. The important point here, however, is that in this scenario information is collected by the users for the users. This means that the influence of the provider on the overlay is reduced, and connection attributes that matter more to the end user are emphasized. An addition of this kind of SIS might enforce the upholding of its part of the bargain for a win-win solution, since the rating of an overlay SIS cannot be directly influenced by a provider.

To have a complete and up-to-date view, several of the overlay SIS servers should cooperate in order to generate frequent updates on the quality of ASes and therefore a more exact recommendation for their peers (we assume that there is not only one single server).

This means they have to exchange either the peer reports themselves, which would introduce a large overhead, or their current AS quality ratings, which necessitates a common understanding of the term 'quality' between the cooperating overlay SIS servers.

A more interesting scenario for inter-SIS communication in this context is the cooperation between a topology-aware and ISP-provided SIS, e.g., using the BGPLoc algorithm, and an overlay SIS as described above. This might be the case when overlay users or the overlay provider will not allow experience information about the quality of the overlay to be passed on directly to providers. In this case, the overlay provider may offer its own overlay SIS that aggregates the user reports and that can query the ISP's SIS(es). The overlay SIS can then merge its own user experience information with the topology information of the ISP to give better (application-dependent) recommendations. We will, in the following, focus on this case, however, the communication might as well take place with the overlay SIS being queried by the ISP SIS.

1. A client p_m queries its overlay SIS server X to rate the connection between p_m and a remote peer p_n.
2. X looks up the ISP SIS server Y responsible for p_m (and possibly the attributes offered by Y) in a mapping service for SIS servers (to be defined).
3. X queries Y for the available topology attributes for the pair p_m and p_n.
4. Y responds with a topology-based rating for $<p_m, p_n>$.
    a. If Y wants to disclose these, it can also respond with the single attributes and values used to compute this rating.
5. X computes the rating of $<p_m,p_n>$ based on its locally available information (user/client-reported experience) and the topology rating.
6. X returns the final rating to p_m.

For this scenario, the user-measurable attributes that an overlay SIS can gather need to be defined. Also, the reliability of information gathered by peers must be taken into account, as well as a process of aging of values to allow changes in the physical network to affect user-generated ratings over time.

5 Insertion of ISP-owned Peers in the Overlay

The ISP-owned peer (IoP) is an entity equipped with high resources that participates in the overlay and stores content in the ISP's premises, aiming at seeding this content in the future. The underlying objective is to increase the level of traffic locality within an ISP and to improve the performance experienced by the users of peer-to-peer applications. Obviously, IoP belongs to the ISP infrastructure and is controlled by the ISP itself.

The IoP runs the overlay protocol, e.g., BitTorrent, but with some small differences that serve its purposes. In particular, the IoP must be able to unchoke more peers proportional to its resources in order to exploit its extra uplink capacity. It must also be capable of storing the downloaded content and, certainly, of uploading it back to the network. Similarly as with regular peers, the IoP initially acts as a leecher in each file's swarm trying to obtain (chunk-by-chunk) the complete copy of the file; and thereafter it acts as a seed. In this document, we restrict attention to the mechanism itself and do not deal with legal issues [LLC06].

5.1 Description of the Mechanism

The main goal of the IoP is to serve as much as possible regular local peers that belong to the same AS with it, in order to achieve reduction of their completion times as well as of inter-domain traffic and its associated charges. Additionally, the IoP also serves peers that belong to different ASes, if this is beneficial for the ISP that deploys it. Again this kind of benefit should be measured in terms of interconnection charges' reduction.

Ideally, the IoP insertion in an ISP domain aims at avoiding data redundancy on the inter-domain link. In particular, this means that the content downloaded by the IoP from different domains would not be downloaded also by its local peers; instead these chunks would be circulated by the IoP inside its own domain. Therefore, significant reduction of inbound inter-domain traffic, but also increase of the intra-domain traffic, is expected.

The IoP operation is broken down into two phases: the 'leeching' phase and the 'seeding' phase. In the first one, the IoP participates in each file's swarm as a leecher, namely it both downloads and uploads chunks of the file; whereas in the second phase, the IoP has already a complete copy of the file and serves this file as a seed. During the 'seeding' phase, the IoP essentially acts as a cache.

5.1.1 Possible Scenarios

In this document, we consider the insertion of the IoP in a BitTorrent network, which will be henceforth considered as the basic scenario. However, the mechanism applies similarly to other peer-to-peer networks such as a Tribler network.

Three basic cases have been distinguished for such a deployment:

1. **Plain insertion of IoP in a peer-to-peer network:** All peers are assumed to run the original protocol of the application. No combination/collaboration with SIS and no interconnection agreements are considered. Therefore, the overlay tracker treats IoP as a regular peer, without being aware of its distinctive features or its real identity. In this case, the IoP is expected to be preferred by other peers due to the tit-for-tat principle employed by BitTorrent's choking algorithm and because of its high uplink capacity. In the general case of a peer-to-peer application, the IoP should have such a resource

profile and behavior so that the incentive mechanism in place renders it as a preferred source of content for the ordinary peers.

2. **Combination of IoP with SIS:** Both IoP and SIS co-exist in the same domain. The combination of IoP with SIS does not require any information to be exchanged between them. The simplest case of this combination is that neither the IoP nor the SIS knows each other's existence and therefore, they operate independently. In this case, the SIS evaluates IoP, if it is included in some peer's query, as any other regular peer based on the rating criterion used, e.g., BGP locality (see Section 3). An alternative would be to assume that only the SIS knows IoP's existence, however it still does not interact directly with it; IoP's behavior is affected only indirectly by the SIS. In particular, the SIS places the IoP in top positions in the replying list to a peer's query, if it was originally included in some peer's query.
3. **Collaboration of IoP with SIS:** Both IoP and SIS co-exist in the same domain and are aware of each other's existence. The collaboration of IoP with SIS requires exchange of information between them. IoP and SIS are assumed to know each other a priori. The most significant application of this collaboration is the SIS-enabled swarm selection. Employing the SIS in the swarm selection procedure, the IoP acquires necessary overlay information not from the tracker, but from the peers through the SIS. Thus, no interconnection agreement with the overlay tracker is required. The provision of extra information from the peer to the SIS is incentive compatible. Indeed, if the peer informs the SIS of the swarm it participates, then the IoP will consider joining this swarm, which will be beneficial for this peer. This incentive can be strengthened by the SIS, which will advertise the IoP only to those peers that provide such information, and which will achieve lower completion times.

5.1.2 An Overview of the IoP Specification Issues

The performance of the IoP is constrained in terms of

- bandwidth,
- storage space, and
- management overhead, to deal with the acquisition and exploitation of the necessary information.

Bandwidth is considered to be the most significant constraint; therefore a lower bound of the bandwidth is defined that must be allocated by the IoP per swarm. On the other hand, it is assumed that there is plenty of storage space available and the management overhead is not restrictive. Since storage space is not considered to be a strict constraint, the IoP can store all downloaded content, even from uncompleted files, in order to be able to re-enter the respective swarms in the future, if it is beneficial. Assuming a stricter constraint regarding the storage, then standard cache management policies by literature can be employed.

The complete specification of the IoP includes the specification of the following issues:

- number of IoPs per domain,
- location of IoPs within a domain,
- storage capacity and management,
- bandwidth capacity and allocation (dimensioning),
- swarm selection, and

- modification of the execution of the overlay protocol, e.g., unchoking policies.

Any policy or mechanism not mentioned above is considered to be same as that applied for the regular peers. In subsections that follow, we briefly outline each such issue. Wherever required, we also examine how the aforementioned scenarios affect these issues.

5.1.3 Number of IoPs

Currently, only one IoP instance per domain is considered. However, future studies will focus on the number of IoPs that an ISP should deploy, given the size of its domain, the topology of its network and the number of active peers. Assuming a certain budget invested by the ISP for installing IoPs, there is a trade-off between the number of IoPs and the performance benefit attained by each of them. Due to multiplexing of resources, in a network with small delays, an IoP implemented as a server farm equipped with a large capacity link, e.g. 1 Gbps symmetric, will most probably be more efficient than many IoPs equipped with smaller capacity links, e.g. 10 Mbps symmetric. However, it is still not clear whether it is in general more beneficial to distribute the resources dedicated to the IoP(s) in multiple overlay entities at (either at the same or at different locations – see below), or concentrate all of them in a single one. Such scenarios need to be evaluated by means of simulations.

5.1.4 Location of IoPs

Regarding the topology and location of IoP(s), there are three options: centralized, decentralized, and intermediate. Under the centralized approach, one “large” IoP is deployed in the IP backbone of the ISP. In order to communicate with it, traffic needs also to pass through backbone links; however crossing a backbone link of 100 Km introduces, due to the transmission and propagation delays, only a total delay of 0.3 msec. Under the intermediate approach, multiple overlay entities are located at the same central point, without static allocation of resources. Under the decentralized solution, several smaller IoPs are deployed in some aggregation points, e.g. the DSLAMs. This approach implies reduction of the hop number from regular peers to the IoP, as well as less congested backbone links. Additionally, this approach assures higher content redundancy, availability, and Denial-of-Service attacks resistance. The latter capability can also be assured by assigning more than one IP addresses to each IoP. Again, different deployment topologies need to be evaluated by means of simulations.

5.1.5 Storage

The storage space assigned to the IoP is also determined by the ISP that deploys it. As stated earlier, storage space is not considered as a strong constraint to our system. However, in real systems storage space is finite, and therefore mechanisms must be considered in order to always have updated and popular content stored in the IoP’s cache. Such mechanisms have been proposed in literature for caches.

5.1.6 Access Bandwidth Capacity

The access bandwidth of the IoP is determined by the ISP that deploys it. The lower the bandwidth, the fewer swarms that the IoP can be involved in. An issue arising here is whether it is beneficial to connect the IoP by means of multiple, physical links rather than a single one.

5.1.7 Swarm Selection

The IoP participates in the overlay proactively by running the overlay protocol and exchanging content with other peers. In order to decide which files are beneficial for the ISP to share, the IoP needs to detect existing swarms and then to evaluate which swarms are more beneficial to join. The swarms given the highest evaluation rating are those that the IoP selects to participate in. Both swarm detection and swarm rating are coupled under the more general term swarm selection.

Swarm selection can be either overlay-aware or overlay-agnostic. Both cases, as will be further described in the next sections, imply that for the swarm selection process the IoP should collaborate with the overlay tracker, e.g., the IP addresses of some known trackers are given to the IoP to start communicating with them. The overlay trackers can be selected based on content localization. For instance, a Greek ISP deploying an IoP will set its IoP to query the most known Greek trackers. This collaboration is incentive compatible for the overlay tracker in an indirect manner, since its peers will receive better QoS, e.g., in terms of lower completion times. In order to avoid the need for agreement establishment and for a new protocol between IoP and overlay tracker, solutions such as the use of a crawler could be used. However, this would also require manual entry of information to the crawler, and therefore such a solution is not considered for now.

While the previous two cases match with the scenarios where no SIS is present, the existence of an SIS that collaborates with the IoP (which is possible because they both fall under the same administrative entity), combined with swarm awareness can lead to a promising solution. In this case, the peers voluntarily provide the SIS with their overlay statistics, the SIS collects and aggregates them, while preserving peer anonymity, and the IoP queries the SIS for the most popular content (i.e. swarms) in its domain. At this point of the project, this solution appears as the most appropriate for deployment.

For reason of completeness, in the following subsections the swarm selection procedure is briefly described for the first two scenarios (without SIS collaboration). Note however that the third scenario is fully specified in Section 5.2.4 as the proposed solution for swarm selection.

5.1.7.1 Swarm-aware Swarm Selection (Scenarios 1 and 2)

The underlying idea is to use information that is already available to the overlay tracker or other peers, in order to avoid the effort to gather new statistics. Such overlay statistics are:

- number of peers that participate in each swarm,
- number of local peers that participate in each swarm,
- total file size and number of chunks,
- number of chunks remaining to be downloaded by known to the IoP leechers,
- number of chunks remaining to be downloaded by local peers only, or
- combination of some/all of above criteria.

An interconnection agreement between the IoP and the overlay tracker is assumed to be established here. A protocol for their communication is described below:

For the bootstrapping, the IoP sends a request to a known tracker, asking overlay information about the torrent files registered by the tracker. The tracker then replies with a list of swarms, including several statistics for the swarm, like the file location descriptor, the file size, the number of leechers and seeds in the swarm, etc. Then the IoP based on the this information, provides a rating for each swarm in a way that captures the demand for the specific content. A simple criterion could be the following:

*s_i = file_size_i * nof_leechers_i / (nof_seeds_i+1).*

This simple formula is justified by the fact that the swarm rating should be proportional to the file size, and the number of leechers, while it is inversely proportional to the number of seeds. Since the latter can be 0, the +1 term is included in the denominator. Note here that in the formula the file size is used instead of the number of chunks since it is independent of the peer-to-peer client that initially breaks the content file down into chunks. Then the IoP joins the *S* top-rated swarms (where the calculation of *S* is explained below (in subsection 0), and continues by following the standard overlay procedures, as a leecher.

After a time period of *T* minutes, and every *T* minutes, the IoP performs a re-rating of the swarms it participates in. The new criterion for the rating of existing swarms can be the following:

s'_i = tot_bytes_down_i / file_size_i,

where tot_bytes_down_i is the total number of bytes to be downloaded by all local peers in swarm i; practically, s'_i is the remaining content redundancy within IoP's domain, which reflects the times the content needs to be downloaded by peers within the domain of the IoP. Additionally, the IoP discovers new swarms by re-querying the overlay tracker for statistics about new torrents registered to the tracker since the last swarm information reply. The IoP then continues to participate in the *S'* top-rated swarms, while it renews its activity by joining the *S*-*S'* top-rated newly acquired swarms.

5.1.7.2 Swarm-agnostic Swarm Selection (Scenarios 1 and 2)

In this case, no overlay statistics are gathered or used. Instead, an initially random selection of *S* swarms where the IoP participates is performed. The decision on whether a swarm should be "dumped" for a new one is based on the *upload activity* of the IoP in the swarm, namely the amount of traffic uploaded by the IoP to the peers of this swarm. If the upload activity to a certain swarm within a given time period *T* is low, and particularly below a pre-specified threshold *U_act*, then the IoP can exit this swarm, and randomly select another swarm to enter. This way IoP has the opportunity to discover more "interesting" swarms. This is similar to the choking algorithm of BitTorrent where the peers with the lowest upload bandwidth are choked, and there is also an optimistic unchoke for discovering new peers.

5.1.8 Unchoking Policy

The unchoking policy or choking algorithm, as it is called in BitTorrent, of regular peers is described in [C03]. In particular, each peer unchokes *m = 5* other peers every 10 sec based on the highest upload bandwidth criterion, and 1 peer randomly or 'optimistically' every 30 sec. Additionally, after a peer turns into a seed, it unchokes m peers based on the criterion of highest download bandwidth. Therefore, the simplest case for the IoP is to follow the same policies, but employ a larger number *K* of unchokes per swarm. *K* depends on the upload bandwidth that has been assigned for use to each swarm, the choking interval and the chunk size (see subsection 1.2.3). In order to have some impact on the swarm that the IoP participates, *K* must be significantly larger than *m*.

5.1.9 Scenario Extensions

Complementary to the three basic scenarios, namely the plain insertion of the IoP, the combination with SIS, and the collaboration with SIS, two secondary cases of collaboration can be assumed. These are briefly discussed below:

1. **Collaboration with Overlay Provider:** Due to this collaboration, the tracker favors the IoP when replying to peers' requests. Additionally, when the IoP serves as a seed for a specific file, then the tracker includes IoP's address only in reply messages to peers that belong to the same AS, unless the ISP decides that the entry is beneficial according to the interconnection policy. Otherwise, also connections to non-local peers are required in order for the IoP to obtain the entire content.
2. **Collaboration with Content Provider:** Due to this collaboration, the CP's content is stored in the IoPs and the torrent file generated by the CP contains as meta-info the IP addresses of the IoPs.

5.1.10 Related solutions in the industry

In this section, we briefly present and discuss solutions in the industry that are related to the IoP insertion. In particular, we have distinguished two solutions: 1) Oversi's NetEnhanser and OverCache, and 2) Sailor: In-Network Data Lockers.

- Oversi's OverCache is an already commercially available solution by Oversi (http://www.oversi.com/). It supports both commercial and non-commercial P2P traffic and provides full control to the ISP. The OverCache delivers content with high bandwidth to the end-users, thus achieving savings on inter-domain traffic, QoE improvement, churn reduction, congestion avoidance and eventually customer base increase. The main disadvantage with respect to SmoothIT's requirements [D1.1] is that OverCache monitors the popularity of specific files probably employing DPI (deep packet inspection). Additionally, content is claimed to pass only once through the backbone and then stored in the cache, where from it is subsequently served. Therefore, although not mentioned explicitly, it is apparent that peers' requests besides being eavesdropped are also intercepted and redirected to the cache. The latter is completely different from the IoP insertion, since the IoP performs no monitoring or redirection of peers' requests, but participates actively in the overlay. The collaboration of the IoP with the SIS provides performance incentives to the peers to truthfully state their intentions to the SIS, which renders the IoP more effective.

On the other hand, Sailor [IETF75-3] employs *application-agnostic* content storage in data lockers. A major difference of Sailor comparing to IoP is the assumption that content providers are willing to cooperate with the ISP, namely they store their content to data lockers in order to be downloaded by other peers; IoP instead assumes only cooperation of peers, i.e. peers declare which swarms they are participating to the SIS. Another important difference is that a special protocol is required for the access of the data lockers by peers, while in IoP insertion case, both IoP and regular peers exchange content following the original peer-to-peer protocol. Last but not least, is not clear whether the data lockers are controlled by the ISP or the overlay. If the latter happens, then significant physical network information, such as that provided by SIS, is not available to the Sailor. On the other hand, IoP's collaboration with SIS assures that content storage is combined with traffic localization employed by SIS, which therefore is expected to have more significant impact on both traffic and users' QoE.

5.2 Detailed Specification of the Proposed Mechanism

Having analyzed all various properties that the IoP ETM mechanism should hold and the different scenarios and considerations that can be taken into account, the detailed features of the mechanism to be implemented is specified in the following.

5.2.1 Proposed Architecture

It may have become obvious so far that IoP is assumed to be an entity physically separated from the SIS. This may or may not be the case. In the rest of this section we treat the IoP as a different entity that communicates with the SIS using a SmoothIT-specific protocol, while it communicates and interacts with the overlay using the standard overlay protocol. Different realizations of the architecture may be possible though, but such decisions are implementation specific.

5.2.2 Number of IoPs

As aforementioned, there may be one or more IoPs deployed per domain. Generally, the specification of specific modules of the IoP is independent of the number of IoPs within the same domain, since all IoPs are under the same authority. The only difference is that during the swarm selection procedure, the different IoPs may need to interact with each other, in order to avoid entering the same swarms. However, without loss of generality, and for simplicity reasons, we only consider specification of the IoP under the assumption that a single IoP is inserted per domain.

5.2.3 Dimensioning

As mentioned earlier, we assume that there is a lower bound of the bandwidth that the IoP must be able to allocate per swarm. The lower bounds are denoted as *u_low* and *d_low* respectively for upload and download bandwidth. On the other hand, the total upload and download bandwidth of the IoP are denoted *u* and *d*. Practically, *u* and *d* are parameters tuned by the ISP and related to the physical link that connects the IoP to the Internet. As a consequence, the maximum number of swarms that the IoP can participate in is *S*, where

$$S = \min\{u/u\_low, d/d\_low\}.$$

Certainly the values of *u* and *d* will be such that *S* is not too small, since the IoP would have some impact only too few swarms. By assumption, *S* is also not too large, thus not introducing significant management overhead. The parameters *u_low* and *d_low* must be treated as constants. Their actual values will be determined through simulative evaluations.

As stated earlier, the IoP runs the overlay protocol and, therefore, uses all overlay protocol parameters. However, the values of parameters may differ from the respective values for regular peers. In particular, in order to exploit the higher bandwidth resources of the IoP, we assume that the number of unchokes (*K*) of the IoP per swarm must be higher, compared to that of regular peers *m*. A proposed value for this parameter *K* is:

*K = u_low * choking_interval / chunk_size;*

where *u_low* is the lower bound of the bandwidth required to enter a swarm (expressed in KB/s) and the choking interval will be set to the choking interval of BitTorrent, e.g., 10 sec. Especially regarding the chunk size, at bootstrapping this can be taken equal to a typical value, e.g., 256 KB. Later and after the IoP has joined a swarm, the corresponding chunk size can be easily derived by the torrent file (or the chunk exchange with other peers). Therefore, the corresponding value for the parameter *K* should be re-calculated then.

5.2.4 Swarm Selection

We assume the IoP-SIS collaboration case here. The SIS-enabled swarm selection presented below exploits information available by the SIS in order to evaluate the swarms to be joined by the IoP, as well as to acquire the IP addresses of local peers that participate in the selected swarms. The SIS-enabled swarm selection requires a protocol for the interaction between peers and SIS and a protocol for the interaction between IoP and SIS.

5.2.4.1 Peer-SIS Protocol

The target here is to inform the SIS about the swarm each peer joins:

1. Each peer sending a request to the SIS, must include in pre-specified fields the following information regarding the swarm it participates:
    - the torrent file id,
    - the torrent file location descriptor which is assumed to be a priori known to the peer,
    - the size of the content file.
2. The SIS collects this data along with the peers' IP addresses from all queries and stores them for future use.
3. The SIS handles the peer query:
    - If the torrent file id exists in the set of swarms that the IoP participates, then the SIS ranks the IoP and includes it in the replying list to the querying peer. Alternatively, the IoP could be assigned by the SIS the top rating from those peers that were included in the peer's query.

5.2.4.2 IoP-SIS Protocol

The target here is for the IoP to acquire overlay information from the SIS, and furthermore to inform the SIS about decisions taken by IoP's swarm selection procedure:

A. *Bootstrapping*

1. The IoP sends a query to the SIS asking swarm information.
2. The SIS reply message must contain:
    - the torrent file id,
    - the torrent file location descriptor,

- the size of the content file,
- the number of local leechers,
- the number of local seeders.

3. The IoP performs rating of the known swarms using the formula:

   *s'_i = file_size_i * nof_local_leechers_i /* (*nof_local_seeders_i* + 1)*;*

4. The IoP joins the first *S* swarms with the highest ratings.
5. The IoP informs the SIS about its decision by sending a message to it containing all torrent file ids of the swarms it participates; this particular message is also considered by the SIS as a request for local peer addresses that participate in those specific swarms.
6. The SIS stores the list of torrent file ids for future use, and then replies a list of IP addresses of local peers appending to each one of them the id of the swarm where it participates.

B. Running

1. After a time period *T* and for every *T* minutes, the IoP sends a query to the SIS asking for new overlay information.
2. The SIS replies a list of new swarms that have been made known to it, since its last swarm information reply, appending also the overlay information (see Bootstrapping – step 2).
3. The IoP calculates ratings *s'_i* for the new swarms (see Bootstrapping – step 3).
4. The IoP joins the [*x*·*S*] new swarms with the highest ratings *s'_i*, where *x* depends on the number of new swarms (registered since the last overlay information reply of the SIS) and the total number of swarm that the IoP is aware of:

   $$x = \lceil \mathrm{nof\_newswarms/nof\_totswarms} \rceil .$$

   The underlying idea of introducing the parameter *x* here is to keep participating in the most "promising" of the already served swarms, but also investigate if there are other more "interesting" ones.
5. Then, the IoP evaluates the already known swarms based on the highest upload throughput criterion. The upload thoughput *u_i'* is defined as the total bytes that the IoP has uploaded to the peers of each swarm during time period *T*, divided by T,
6. The IoP joins the *S - [x·S]* of the old swarms with the highest upload thoughput *u_i'*.
7. (See Bootstrapping steps 5 & 6). Regarding the reply message of the SIS to the IoP and in order to avoid redundancy, the SIS includes in the peer list only the IP addresses of the peers that have communicated with it since the last reply message was sent.

In Figure 17 we present the respective sequence diagram:

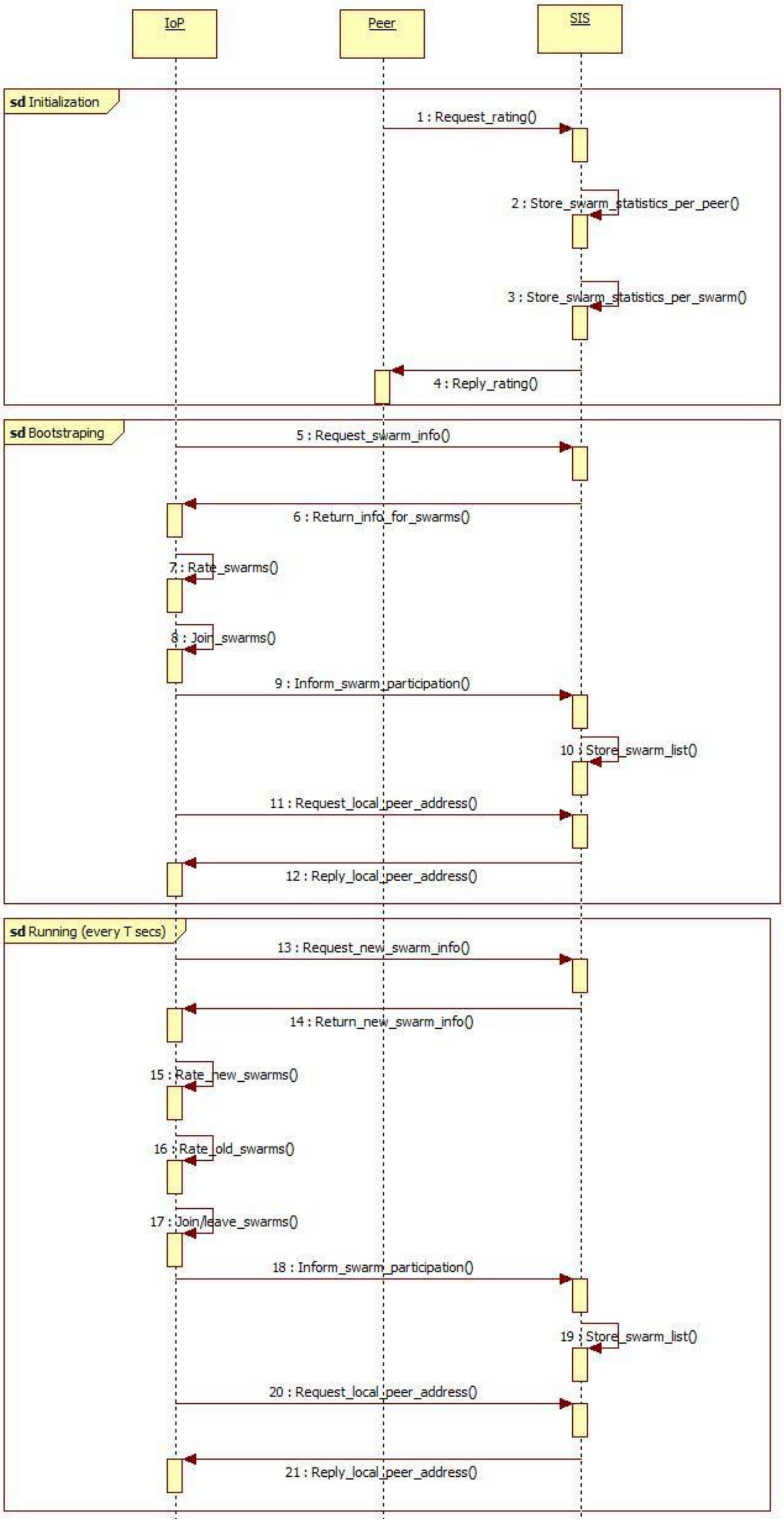
IoP
Peer
SIS
sd Initialization
1 : Request_rating()
2 : Store_swarm_statistics_per_peer()
3 : Store_swarm_statistics_per_swarm()
4 : Reply_rating()
sd Bootstraping
5 : Request_swarm_info()
6 : Return_info_for_swarms()
7 : Rate_swarms()
8 : Join_swarms()
9 : Inform_swarm_participation()
10 : Store_swarm_list()
11 : Request_local_peer_address()
12 : Reply_local_peer_address()
sd Running (every T secs)
13 : Request_new_swarm_info()
14 : Return_new_swarm_info()
15 : Rate_new_swarms()
16 : Rate_old_swarms()
17 : Join/leave_swarms()
18 : Inform_swarm_participation()
19 : Store_swarm_list()
20 : Request_local_peer_address()
21 : Reply_local_peer_address()

Figure 17: SIS-enabled swarm selection

5.2.5 Unchoking Policy

In this section we specify which remote peers to be considered for the *K* unchoking slots per swarm, as described previously.

A. ***Leeching Phase***

The IoP lacks some (or initially all) chunks of the content file. It exchanges chunks with local and non-local peers, in order to quickly download all missing chunks. It would be more beneficial for the IoP to interact more with non-local peers in this phase, so that it gets all chunks that cannot be found among its local peers. Still the IoP starts disseminating these chunks to its local peers even before it becomes a seed:

1. The IoP checks all known connections.
2. It calculates the rolling average of the upload bandwidth received by each of those connections, for an interval of 10 seconds (such as the BitTorrent's choking interval).
3. Then the IoP unchokes *y·K* local peers with highest upload bandwidth. The parameter *y* depends on the number of chunks that the IoP has acquired and the total number of chunks of the content file:

   $$y = \lceil \text{iop\_chunks/nof\_chunks} \rceil,$$

   namely the IoP unchokes more local peers, as it concentrates more chunks. Practically, this means that for new swarms the IoP chooses to interact more with non-local peers. The underlying idea is to gather quickly all chunks that the local peers probably miss, in order to upload these chunks later to them.
4. It also unchokes *(1-y)K* random peers (i.e. without taking into account whether they are local or not) with highest upload bandwidth.

If the number of random peers is *Y<(1-y)K*, then the IoP unchokes all of them, and also unchokes *K-Y* locals. Certainly, the sets of local and random peers selected must be disjoint.

B. ***Seeding Phase***

The IoP has a complete copy of the file. If no interconnection agreement with another domain is considered, then the simplest policy is to unchoke only local peers:

a. The IoP checks all connections to local peers.

b. It calculates the rolling average of the download bandwidth sent to those connections, for an interval of 10 seconds (such as the BitTorrent's choking interval).

c. Finally, it unchokes *K* local peers with highest download bandwidth.

In Figure 18, we provide the respective sequence diagram for the unchoking procedure.

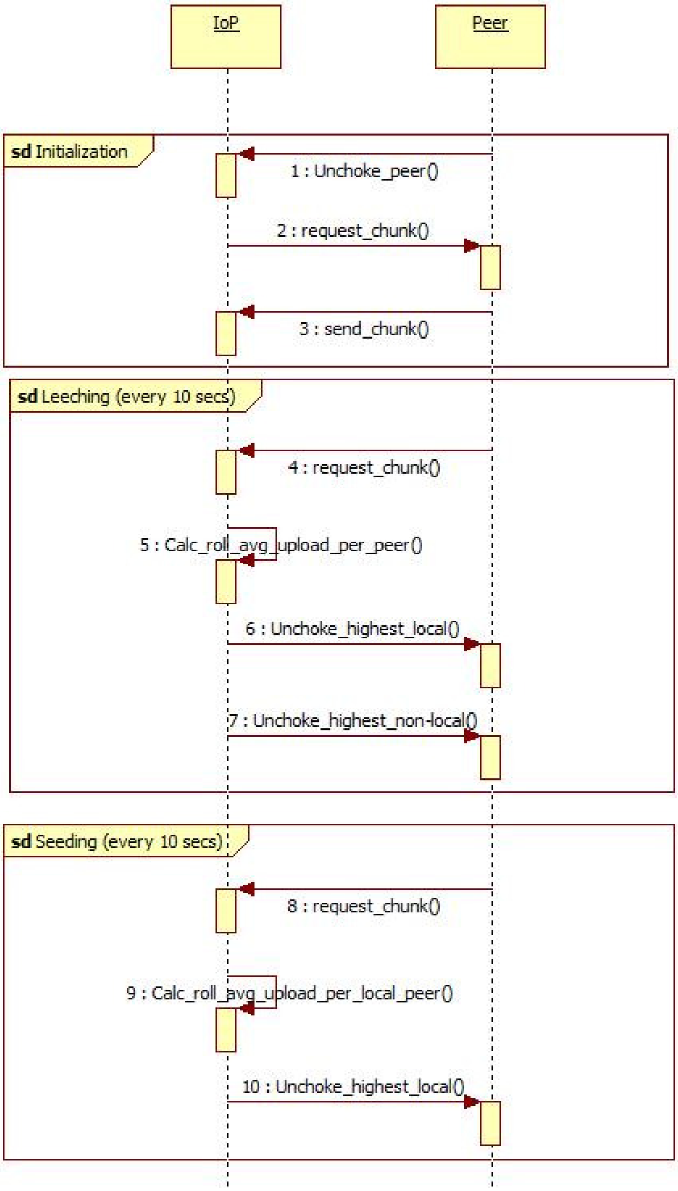
IoP
Peer
sd Initialization
1 : Unchoke_peer()
2 : request_chunk()
3 : send_chunk()
sd Leeching (every 10 secs)
4 : request_chunk()
5 : Calc_roll_avg_upload_per_peer()
6 : Unchoke_highest_local()
7 : Unchoke_highest_non-local()
sd Seeding (every 10 secs)
8 : request_chunk()
9 : Calc_roll_avg_upload_per_local_peer()
10 : Unchoke_highest_local()

Figure 18: Unchoking policy for both phases of the IoP

5.2.6 Bandwidth Allocation among Swarms

As already mentioned in Section 0, the IoP tries to join *S* swarms, based on the available uplink and downlink capacity, as well as the minimum bandwidth that can be allocated to a single swarm. (Recall that we consider that the IoP has a broadband connection with *u* Mbps upstream and *d* Mbps downstream.) In this section we describe how the spare bandwidth should be allocated. This case is applicable if the IoP knows less than *S* swarms to join, or if although the IoP joins *S* swarms there is still spare capacity in either the uplink or the downlink (but not in both), because *u/u_low ≠ d/d_low.* We describe two different bandwidth allocation policies, with the second one being more suitable for implementation.

A. Uniform Policy

The simplest policy allocates bandwidth uniformly among swarms. For instance, when the IoP participates in *S* swarms, then the upload bandwidth assigned to each swarm is *u/S* Mbps, while the download bandwidth is *d/S* Mbps:

1. Let IoP participate in *S'<S* swarms. The bandwidth allocated to each swarm is *u/S'* | *d/S'* Mbps up|downstream.
2. Each time the IoP joins a new swarm, it re-calculates the bandwidth assigned to each one of them: *u/(S'+1)* | *d/(S'+1)* Mbps up|downstream.
3. Each time the IoP leaves a swarm, it re-calculates the bandwidth assigned to each one of them: *u/(S'-1)* | *d/(S'-1)* Mbps up|downstream.

This simple policy can be less efficient than other more dynamic policies. However, it is easy to implement and convenient for the ISPs since it generates very low management overhead.

B. Proportional Policy

Nevertheless, we propose to use the following more sophisticated policy. This takes into consideration overlay information such as that used also for the swarm selection. Since swarm selection has already been performed, the bandwidth allocation algorithm can re-use the already derived swarm ratings according to which the IoP decided in which swarms to participate. Suppose that the IoP paricipates in *S' < S* swarms:

1. The IoP allocates at least *u_low* | *d_low* upload and download bandwidth per swarm.
2. If there is spare bandwidth *(u – S' * u_low)* | *(d – S' * d_low)*, then it is allocated proportionally to each swarm based on the set of ratings *s'_i* derived by the swarm selection procedure:

   *(u_i = u_low + s'_i / (Σ_i s'_i) * (u - S'*u_low))* |

   *(d_i = d_low + s'_i / (Σ_i s'_i) * (d - S'*d_low))*,

   which equals the basic bandwidth plus a proportionally fair share from the rest.
3. Each time IoP re-executes swarm selection algorithm, namely after a period T, the bandwidth allocated to each swarm is re-calculated according to the aforementioned procedure (steps 1, 2). Note that *s'_i* must be re-calculated by the IoP for all the swarms it actively participates in.

Since this is an internal procedure of the IoP, where it re-uses existing ratings from the swarm selection procedure, no sequence diagram is provided here.

5.2.7 Storage

As mentioned above, a cache management/replacement policy is required to keep the content stored in the IoP's cache updated in the unlikely case that the IoP runs out of storage capacity. A variety of such policies has been proposed in the literature; the most popular ones are *Least Recently Used (LRU)* and *Least Frequently Used (LFU).* LRU discards the least recently used items first, whereas LFU discards the least frequently used first. The first policy requires the system to know *when* each item was last used, while the second one *how often* it is used. Additionally, several variations of these two basic policies have been proposed. A straightforward solution is to apply directly either LFU or LRU at the chunk level. Also a combination of them can be deployed.

5.2.8 System-wide Parameters

In Table 9 we present the configurable parameters of the IoP, their meaning and impact, as already mentioned in the previous sections. Some of them are to be defined by means of simulations while others are ISP-specific.

Table 9: IoP-specific parameters

Parameter	Description
T	Time period. Each procedure described below, in its running phase, is re-executed after a time period T. This period T should be carefully selected, since a large T would result in IoP remaining in “non-interesting” swarms for longer time, but on the other hand, a small T would result in instability and increase of the management overhead.
U	Upload capacity of IoP. This depends on the physical connectivity of the IoP.
D	Download capacity of IoP. This depends on the physical connectivity of the IoP.
u_low	Lower bound of upload bandwidth per swarm. The IoP must allocate at least u_low to each swarm, in order to achieve the desired goals.
d_low	Lower bound of download bandwidth per swarm. The IoP must allocate at least u_low to each swarm, in order to achieve the desired goals.
x	The percentage of new swarms the IoP will join.
y	The percentage of local peers to be unchoked by the IoP in leeching phase.
chocking_interval	The time interval when a peer unchokes remote peers. In BitTorrent a typical value is 10 s.
chunk_size	The default size of a chunk. A typical value is 256 KB.
U_act	Upload bandwidth activity threshold. The upload bandwidth to a specific swarm must overcome this threshold; otherwise it is more beneficial for the IoP to exit the swarm. This parameter is only used in the swarm-unaware swarm selection without interaction with SIS; therefore it does not belong to the set of parameters that will be taken into account for the implementation.

6 QoS-awareness Mechanism

The main goal of the QoS-awareness mechanism is to improve the QoE of end users by means of the enforcement of network policies in the operator's domain. It is important to notice that this mechanism implies the execution of active action in the network, i.e. configuration of the network equipment to implement the specified policies. Therefore, this approach is not just based on providing some guidelines to the overlay nodes based on underlying network topology and/or status. To sum up, this mechanism aims to take advantage of the QoS mechanisms available in today's (or in the near future) networks in order to provide network performance guarantees to the overlay users and providers. This incentive could be especially interesting for P2P streaming applications. Thanks to the provisioning of these incentives, there is a clear technical advantage to follow the SIS recommendations.

An important issue to be considered in this mechanism is that the enforcement of network policies has other associated costs:

- In terms of CAPEX (CAPital EXpenditures): the transport network equipment (nodes in charge of managing the data) must be able to support the dynamic configuration of its parameters and the control and management parts must be able to offer interfaces to dynamically change the policies in the network. It is important to consider that the introduction of new functionalities in the network implies a careful evaluation of how the performance requirements are met, which, moreover, strongly depend on the specific operator scenario (topology, clients and its own policies).
- In terms of OPEX (OPerating EXpenditures) the cost also should increase since the introduction of new functionalities also implies additional monitoring and management capabilities to i.e. check the availability of the servers or the current status of the network policies.

Since the costs in the network are increased due to the provisioning of new capabilities and these capabilities are dependent on the current and near-future available commercial products, the section is structured as follows: Firstly, this section identifies those scenarios interesting from the commercial point of view in order to derive the technical requirements. Next, a brief overview of the technical mechanisms that are commercially available to be deployed in today's networks in order to evaluate the technical feasibility of the scenarios analyzed and to provide the guidelines for the implementation of the described mechanism (that are also described in this section). This section also analyses the implications for accounting and billing that should be considered. Finally, this section presents the architectural implications of the designed solution and the qualitative evaluation of the proposed mechanism.

6.1 Use Cases

This section presents an overview of two selected scenarios that could lead to the usage of QoS mechanisms as an incentive for overlay networks. The two scenarios are well differentiated since:

- The "carrier class overlay services" scenario presents and option where two main stakeholders (service/content provider and the ISP) have an agreement. This agreement can be seen as an IP interconnection scenario where different business agents collaborate to provide a service which, in fact, can be considered an evolution of IP interconnection agreements as defined in [GSMA08]. The implementation of this scenario will require operation during the provisioning phase of the network.

- The “QoS on demand for end users” scenario defines an approach that could help the ISP to provide an extension to the retail trade: it offers to the end users the capability to request enhanced QoS for specific applications and, in particular, to overlay applications. This is a user-centric scenario that would imply the enforcement of QoS policies according to the end users’ demands which implies a different timescale operations and it will be described later.

6.1.1 Carrier Class Overlay Services

This scenario aims to provide carrier class overlay services by means of the specification of a network API (Application Programming Interface) to third parties. This API can be provided by reusing the ISP NGN transport control capabilities as it will be discussed in the next section. The ISP configures its traffic management mechanisms in such a way that it can guarantee some QoS performance objectives to the application after the fulfillment of an agreement by a content/service provider. The QoS guarantees can be different depending on the control capabilities available in each domain (changes in the upload/download bandwidth, traffic prioritization, etc.).

With this mechanism, the ISP earns revenue from third party applications. Moreover, the overlay service provider (as, e.g., a peer-to-peer streaming based TV transmission) provides the service with higher quality and can, e.g., thus be more popular while also save some costs related to its servers; e.g., the traffic coming from the servers will be prioritized relatively to other traffic, thus leading to an improved service for the infrastructure available, or the ISP can also provide server installation facilities in its premises. Finally, users enjoy a service with more guarantees thanks to the better provisioning of the service achieved due to the agreement between the ISP and the overlay provider.

It should be noted that this mechanism can be improved by means of allowing the main servers of the overlay service providers to be included in the ISP premises. In order to select the most appropriate location for these servers, this mechanism can follow the same approach to dimension and select the most suitable location of the IoP.

6.1.2 QoS on Demand for End Users

The SIS offers to overlay users the capability to request QoS guarantees for specific connections. This would be an excellent option for VPN provisioning (e.g., a tele-worker uses the QoS capabilities to have real-time guarantees while using his/her corporate services), and P2P streaming solutions (e.g., the end user application can ask the network to provide a set of guarantees for a set of specific connections). This mechanism will be integrated as part of the SIS centralized model, namely as another service that can be provided to the end users.

6.2 Available QoS Mechanisms in Current Networks

In order to implement the scenarios described in the previous section, this section aims to provide an overview of the available QoS mechanisms in the networks considering commercial aspects in terms of operability.

First of all, the specification of the performance objectives must be provided. The way to implement end users requirements is by defining classes of services that allow the network operator to manage the traffic per aggregate. Therefore, as a first step, the SIS must provide a set of well-known classes of services. Following the ITU-T Y.1541 [Y.1541], the QoS classes of services presented in Table 10 can be defined considering the network performance parameters at the IP packet level.

Table 10: ITU-T Y.1541 QoS Classes

Network Performance Parameter	**Classes of QoS**					
	Class 0	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**
IPTD (IP Transfer Delay)	100 ms	400 ms	100 ms	400 ms	1 s	N/A
IPDV (IP Delay Variance)	50 ms	50 ms	N/A	N/A	N/A	N/A
IPLR (IP Loss Ratio)	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	N/A

The types of applications for which these classes are designed are the following:

- Class 0: real time services, sensitive to the delay variance and high interactive, e.g. VoIP (Voice-over-IP) and videoconference.
- Class 1: real time services, sensitive to the delay variance and interactive, e.g. VoIP and videoconference.
- Class 2: highly interactive data transactions, e.g., signaling data.
- Class 3: Interactive data transactions.
- Class 4: services just sensitive to packet losses, e.g., short transaction or video streaming
- Class 5: best effort

This classification does not mean that all these classes must be implemented in each network, but rather that, with this classification, the network administrator can select the way the different services will be provided. In order to select the classes of services to be deployed in the network, SmoothIT has to take into account the interest of the operators in deploying such classes of services. These incentives are associated with the possibility of additional revenues for the overlay provider or to the reduction of costs. In order to address the second goal, the most important issue will be to consider those applications that are generating more traffic.

Recent traffic studies, such as [Ipo2009], show an important increase of the streaming traffic. And, moreover, the most popular contents downloaded using P2P File sharing applications are video contents (around 70% of the total downloads).

On the other hand, P2P based systems are becoming quite popular to distribute TV contents in Internet (e.g., Zattoo [ZAT]) and there are important initiatives to continue developing these systems (e.g., P2P Next European initiative [P2PNext]).

Therefore, we will assume that an ITU-T Y.1541 class 4 or ***streaming class of service*** (that will assure reduced packet losses and a limited delay) will be available in the networks and that this will be the way to provide incentives to both overlay end users and providers.

The networks can provide the guarantees by means of configuring the proper queuing algorithms in the specific network nodes. Typically, in an xDSL network, this can be done at the IP DSLAM and BRAS. The Figure 19 shows an example of the QoS capabilities (such as queuing models) offered by commercial equipment, where the services are provisioned via VLANs. Moreover, this figure shows that each service can be prioritized according to a different queuing methods (priority, weighted) and the data associated to each service is re-directed to different BRAS that will implement its own algorithms.

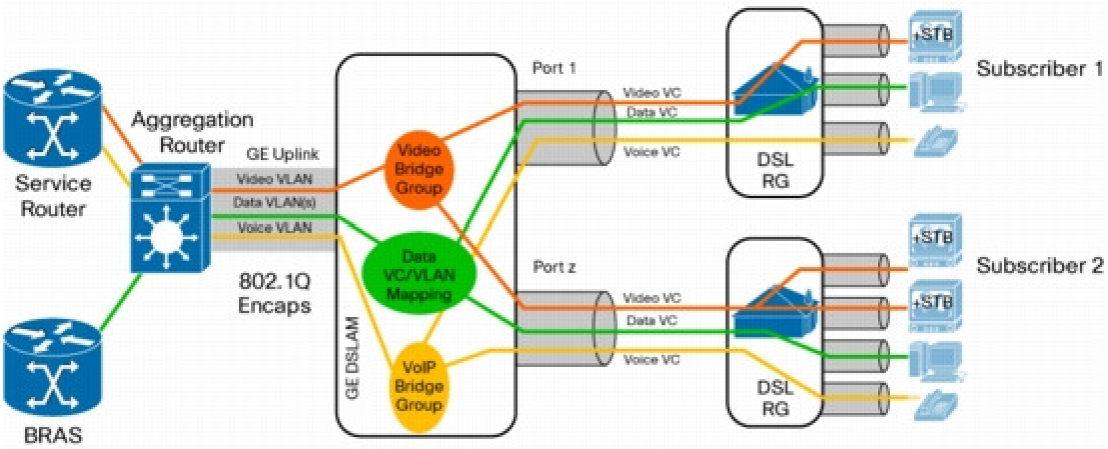

Figure 19: QoS guarantees scheduling

Since one of the most important requirements of the SmoothIT system is to facilitate the deployment of the system in the networks, SmoothIT will take advantage of the transport control functionalities that are provided (or will be provided) in the commercial NGN networks. The TISPAN architecture described in [ES003] is shown in the following figure:

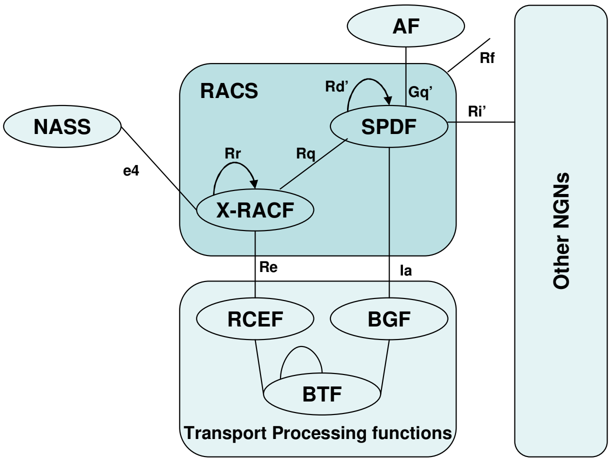

Figure 20: NGN Transport and Transport Control Functions

In the Figure 20, the following main elements can be identified:

- The Transport Processing Functions in the access, metro and core networks. The evolution of all these functionalities are related to the evolution of the network technologies itself (new optical solutions for the core networks, FTTH, new wireless mechanisms, etc.).

- The Transport Control Functions composed of the RACS (Resource and Admission Control Subsystem) and the NASS (Network Attachment Subsystem), which are in charge of controlling the network resources and managing the access to the network.
- Application Functionalities, which include the application support functions and service support functions. In principle, any service can use the transport stratum capabilities but, maybe the most clear standardization (and also commercial) initiative is the IMS (IP Multimedia Subsystem). IMA is triggered by the 3GPP, and specifies an environment where the network operator is in charge of providing the services.

SmoothIT system can interface the Transport Control Functionalities using the Gq' interface provided by the RACS (Resource and Admission Control Subsystem) to the Application Functionalities. The RACS provides the following capacities that are exposed through this interface:

- Resource Control for Multicast and Unicast: the bandwidth dedicated to both types of applications can be changed.
- Charging capabilities: the RACS can generate accounting records.
- QoS Management functions in fixed access networks: the RACS can access the BRAS (or similar equipment) in fixed networks in order to configure the QoS policies. This enforcement can result in both guaranteed and relative QoS configurations.
- Resource control for service quality downgrading: if there are other sessions that have guaranteed QoS and there is a session that must be established with guarantees, then the RACS also offers the capabilities to downgrade the quality offered to other sessions. (This is in an interesting feature to support emergency calls but this option is not related to SmoothIT and will not be considered.)

The scenarios commented above could be implemented with the capabilities offered by commercial (or future) NGN equipment that implement the functionalities described before. Therefore, these scenarios can be just provided by reusing the already deployed capabilities or they can be the source of a new business model that could influence the deployment of NGN equipment.

6.3 Implementation of the Scenarios

The following subsections provide more details about the specification of the two scenarios described in Subsection 6.1.

6.3.1 Carrier-Class Overlay Services

In order to implement this mechanism, the first step is to define the SLA (Service Level Agreement) between the Overlay Service Provider and the ISP. This agreement must contain information about:

- Traffic characterization of the application: this input must allow the ISP to identify the application traffic to which it should provide enforced QoS. Therefore, e.g., the overlay should provide the ports used by the application and the IP addresses of the servers, in order, e.g., to allow the prioritization of the traffic from the Overlay Services. This option is relevant for overlay solutions also supported by servers, which is usually the case for peer-to-peer streaming applications. Indeed, such applications require certain guarantees for smooth delivery of the content and connections to specific and well-known IP addresses (the servers) can be optimized.

- QoS requirements: the ISP must provide to the Overlay Service Provider a portfolio of services that have been provisioned in its network. This portfolio will be provisioned in terms of Classes of Services (CoS), where each CoS will provide its own network performance capabilities (in terms of IPLR, IPTD and IPDV), as it is specified in [Y.1541]; see also Table 10.

Therefore, considering also the specification of the Gq' interface, the following parameters must be provided in the SLA:

- *sla_id → unique identifier for the SLA. It will be provided by the SIS.*
- *list serverIP → list with the IP addresses of the main servers provided by the overlay provider. In fact, as part of the agreement this server could be also located in the ISP premises.*
- *cos → class of service that must be configured in the network. In principle, this parameter will be a string "streaming".*
- *dest_IP → range of IP addresses that will be the clients of the servers (e.g., a PoP in a ISP).*
- *bw → expected bandwidth that will be needed in the destinations in order to download the content from the server.*

This SLA can be configured through the admin interface of the SIS. Then the admin component will request to the QoS Manager the enforcement of the SLA in the domain. In particular, the following actions will be performed:

- For *each* SLA, several reservations will be done. The number of reservations will depend on the number of servers that are included in the SLA.
- For each reservation needed, the QoS Manager will perform a policy_auth_init request to the NGN equipment. In order to make this request, the following parameters must be considered:
    - Information to use the Gq' interface: protocol, IP address, ports, certificates, etc. This will depend on the RACS implementation available in the ISP.
    - For each reservation request, the following information must be provided:
        - session_id: identifier of the reservations done. The information about the mapping between the sla_id and the session_id must be maintained.
        - Media information:
            - Flow direction: if the flows are in the downlink or uplink.
            - Source IP: in this case the IP address of the server in the list.

    - Destination IP: IP address range of the possible clients in the domain.
  - QoS information:
    - Bandwidth that should be configured for the clients-server interaction.
    - Class of service or application type: in this case, it will be assumed in SmoothIT that the class of service is the streaming class (ITU-T Y.1541 Class 4).

After the configuration of all the policies (one policy per server IP), the SLA is installed in the system and guarantees for the flows from the server to the end users are configured in the network elements (in particular, the most typical one is the BRAS element).

The following diagram shows an overview of the interaction:

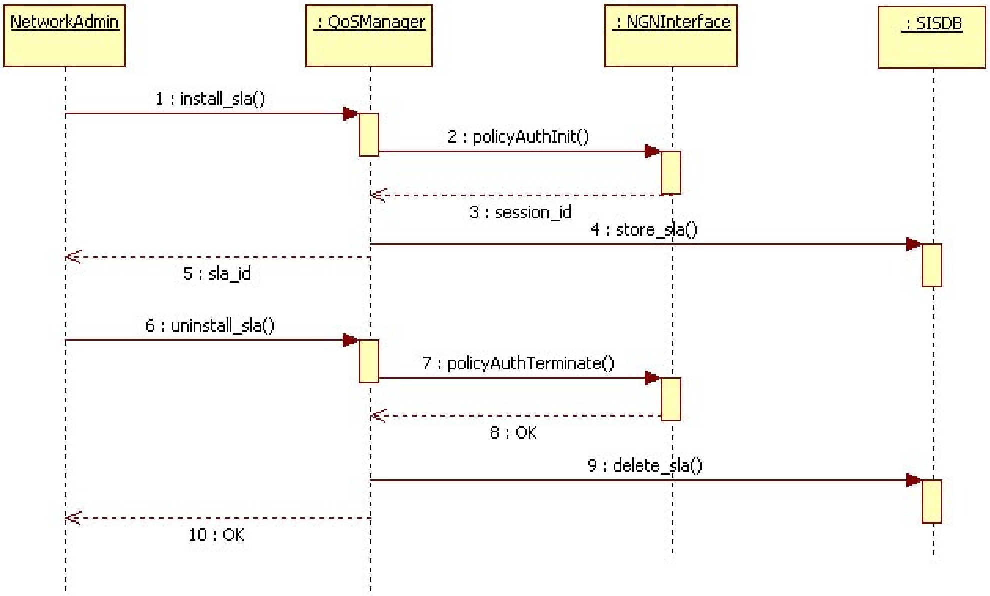

Figure 21: Installation and uninstallation of an SLA

It is important to note that there are some preliminary versions of the NGN equipment (both transport and control) that can also configure dynamically the end user connection profile. That means, if more bandwidth is needed could be configured and assigned to the end user.

In order to implement and deploy this solution, the following issues should be taken into account:

- One of the major advantages of this mechanism is that the expected number of SLA agreements per second will not be high. Therefore, the QoS enforcement can take place at aggregation points of the networks to, e.g., prioritize the traffic from the peer-to-peer streaming servers, both that destined to other servers and that destined to other peers, although the technical approach is different. Indeed, the IP addresses of the servers are well-known. Thus, these flows can be characterized and prioritized in the network without high dynamic performance requirements, as is the case with real peer-to-peer connections.

- If connections between peers must be prioritized, the implementation constraints that are described in the next subsection must be considered. The problem arising here is that the IP addresses of the flows are continuously changing, so new policies must be applied. Moreover, in this case, the large number of requests per second that must be managed by the NGN Control Plane could cause a scability problem to the solution.

6.3.2 QoS on Demand for SIS Users

In this mechanism, the users that interact with the SIS can request the enforcement of QoS guarantees for a set of selected flows. This mechanism will work as follows:

1. An end user makes a lookup to find the peers to connect with. In fatc, it gets a first list of peers.
2. The end user asks the SIS for the rating of the peers.
3. The end user considers this rating and selects the most suitable peers.
4. The end user requests from the SIS the reservation of resources for the selected peers. This means that the client must be modified in this scenario.
5. The SIS uses the QoSManager to enforce the QoS policies in the network.

   When the SIS receives the request(s), it interfaces the QoS Manager that will be in charge of interfacing the NGN capabilities available in the domain. In particular, the QoS Manager can request the provisioning of QoS guarantees for specific flows; e.g., provisioning of Streaming capabilities [Y.1541] to peer-to-peer streaming applications that need low IPLR and low IPTD. Alternatively, it can request to change the user profile in order to provide more bandwidth to peer-to-peer file sharing applications, in case the NGN can support the User Profile dynamic change. In particular, users usually have bandwidth assigned for ISP services such as IPTV and a bandwidth for Internet access. The user could request to change this profile.
6. The SIS returns to the client the identifier of the session established. This session identifier identifies all the reservations done for this request. This session identifier will be used in order to allow the end user to close the session or the session can expire by itself.

The QoS Manager will use the same interface described in the previous chapter in order to enforce the QoS policies in the network. Therefore the following information is associated which each request:

- *Session_id*: identifier of the session
- Media info: Information about the flows that must be guaranteed. For each flow the following information must be provided:
    - *srcIP* → source of the flow
    - *dstIP* → destination of the flow.
    - *Flow direction* → if it is a downlink or uplink or bidirectional depending on the srcIP.
- QoS information:
    - Class of service

  - Bandwidth

If the Gq’ interface provides responses in around 0.5s (needed to configure the policies for a specific end user client) and this is maintained in a commercial environment with a high number of requests/s, this will make this solution suitable to provide QoS incentives by the ISP according to the users’ demand. Thus, the ISP could earn extra charges for this enhanced service by just reusing the NGN Control Plane capabilities that are being deployed in its networks.

6.4 Qualitative Evaluation of QoS ETM

The players involved in this mechanism and their obtained benefits are:

- The Overlay Service Provider which may provide carrier class services by cooperating with the ISP that can be implemented by just reusing the NGN Control Plane capabilities. The ISP could earn additional revenue from third party applications.
- The peers will have strong incentive in terms of performance to follow the SIS suggestions. Moreover, they are being charged for this service.
- The ISP can earn extra revenues for this usage by offering QoS guarantees to the end-users and charge them for this “premium” service.
- The ISP can also attain cost reduction from improved traffic management. If the QoS guarantees are provided to intra-domain flows, overlay users will prefer peers located in their domain and, this would lead to a reduction of the inter-domain traffic.

In order to implement this solution, the SIS will just need to integrate the NGN control plane. This is quite innovative, since this would mean the integration of overlay applications in the NGN framework, representing also a good standardization opportunity.

6.5 Implications to Accounting and Billing

Below, accounting and billing issues are addressed for the two aforementioned deployments of the QoS awareness mechanism, namely the carrier class overlay services and the QoS on demand for SIS users.

6.5.1 Carrier Class Overlay Service

The ISP establishes an agreement with an overlay provider and they set up a family of SLAs. Each one of the SLAs is defined based on the traffic characterization of the application that it will serve, as well as the QoS requirements that have been requested by the overlay provider. For accounting purposes, the QoS manager tracks the following information on a per SLA basis: sla_id, server address, class of service and bandwidth. The QoS manager also tracks the following information on a per reservation basis: session id, server address, destination, class of service and bandwidth. Billing is performed based on both SLA and reservations, while charging is directly imposed by the ISP to the overlay provider. Note that if bandwidth is dedicated per reservation, then the overlay provider, and consequently the user, is charged even if the overlay link is not used; otherwise, accounting should also include tracking of overlay link statistics, in order to perform billing based on a combination of SLA, reservation and usage. Furthermore, it is not necessary to take flow direction into consideration in the accounting computation, since the overlay provider is always charged for (its servers) uploading to the end-users of the overlay. Accordingly, the overlay provider charges the end-users for offering them better QoS, according to the SLA that it has installed with the ISP; therefore, the end-users are charged in an indirect manner by the ISP.

6.5.2 QoS on Demand for SIS Users

In this case, the ISP interacts directly with the end-user, or alternatively SIS-user, therefore the charging is imposed in a direct manner to it. For accounting, the QoS manager keeps track of the following information: session id, flow direction, flow source, flow destination, class of service and bandwidth. Billing is performed on a per-session basis, where the SIS-user, that started the communication, is charged. Mostly, that user will be the downloader, whereas, in rarest cases, it will be the uploader that aims at disseminating content of its own, e.g., an IPTV broadcast channel. Note here that metering should take place in very short timescale, due to the short duration of each session; therefore accounting is expected to be much more expensive and complicated than in the case of the carrier class overlay service.

6.6 Architectural implications

Considering the SIS architecture presented in [D3.2], the following SIS modules/elements should be modified in order to implement the described mechanism.

- SIS interface provided to the clients. How should it be modified in order to allow the end users to request a specific QoS for a set of flows? We can define different two main scenarios:
    - Open interface to the end user (the one described in the previous sections): after the peers gets the list of rated peers it makes a new request to the SIS to reserve QoS to specific flows. Therefore a new primitive should be provided by the SIS Server to the end user and, moreover, the client should be modified in such a way that it can perform new requests when the peers are effectively selected.
    - Non-open interface to the end user: when the client asks for the rated list, the SIS also decides to provide QoS to specific flows. The problem with this case is that maybe the end user will not finally use these peers: if the end user decides not to follow SIS recommendations, the resource reservation could not be used.
- Admin interface: The admin component must provide an interface to allow the installation of the SLA. When a new SLA is requested, the admin must use the install_sla interface provided by the QoS Manager.

7 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 prevents the increased HAP capacity to be used by remote peers since this would increase inter-domain traffic and, therefore, ISP costs.

Unlike the QoS-awareness mechanism discussed in Section 6 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 two fold. On one side it tries to provide an immediate benefit to the ISP and the swarm with additional 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 in the swarms which also act nicely towards their ISP and follow SIS advice in promoting locality.

7.1 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 seeds and leechers 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 both to follow SIS recommendations and to seed downloaded content longer, as such a compliant behavior will increase their chances of being selected as next HAP and therefore help them to become better torrent community members (e.g. help to increase UL/DL ratio on tracker). It 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 22. Here, the local part of the swarm 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 it is possible to promote both seeders and leechers to HAPs.

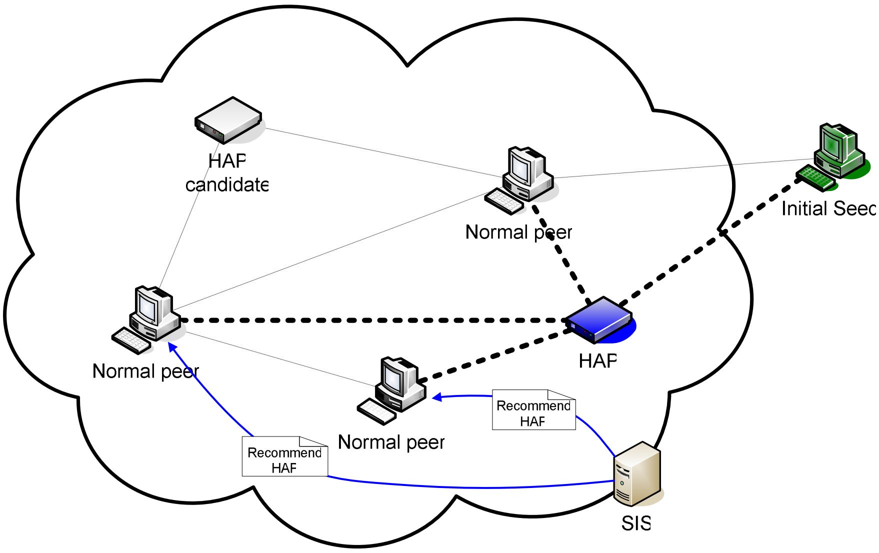

Figure 22: HAP and NGN Scenario

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 and to compute the missing upload bandwidth of the overlay. The latter can be either estimated by the ISP or provided to the ISP through a suitable API (e.g. SIS-Client API, see Subsection 7.7 for details).

7.2 ETM Modes

While the final outcome of HAP ETM is described above, the procedures and steps that lead to it can somewhat differ. Based on how HAP selection procedure works and how fast the NGN equipment is able to update customers profiles in order to promote normal peers to HAPs, the ETM can be broken down in two versions: basic and dynamic.

Basic HAP ETM promotes peers to HAPs based on their behavior patterns in the overlay in general irrespective of the immediate needs of a particular swarm. While it does not solve the problem of missing local bandwidth immediately, should it arise any given it 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 HAP ETM, on the other hand, attempts promote peers to HAPs in those swarm, 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".

An overlay application enhanced by HAP can run in an *ETM-unaware* or *ETM-aware* mode. In the first case, the benefit for the ISP must be enforced through the NMS equipment. This means that there is no need to rely on the ISP-friendliness of the overlay application. In the second case the client is aware of the fact that it runs on an HAP and can optimize its behavior accordingly.

Therefore, this ETM mechanism can work completely transparently to overlay applications if the SIS can obtain the information about the swarms and peers active in these swarms (including their leecher/seed status) by other means. This implies (in the ETM-unaware case) the usage of the DPI to intercept the client-tracker communication or a collaboration with the content provider controlling the tracker. Similar considerations apply to the IoP mechanism. However, usage of DPI approach is more complex and much more demanding in terms of resources on the ISP. Furthermore, significant privacy considerations are hardly avoidable. Therefore, in the following text we assume the ETM-aware mode, unless mentioned otherwise.

In contrast to ETM-unaware mode, the ETM-aware mode assumes that at least the leeching peer is SIS enabled and is able to query SIS server for peers' rating in which case the HAP rating will be increased, should it be present in the request, or, potentially, even injected. On the other hand, in order to become HAP a peer doesn't necessarily need to be SIS-enabled. However, if it does communicate with SIS, it would be able to be more ISP-friendly and therefore its chances of being selected as next HAP would be much higher.

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.

7.3 Detailed Steps

The exact action sequence depends on variation of the HAP ETM used. While most steps stay the same, the initial set of peers and trigger conditions can vary. While basic HAP is triggered on a regular basis and considers all known local peers, dynamic variation is triggered when lack of local upload bandwidth is detected.

7.3.1 Basic HAP

The basic operation of the mechanism is as follows:

1. Peers provide statistics to the SIS controller (alternative or complementary to 2)
2. NMS equipment provides statistics about peers activity (alternative or complementary to 1)
3. SIS selects peers which satisfy the requirement to become HAP.

4. SIS promotes the selected candidates by NMS means and thus increases their upload bandwidth (see Subsection 7.4).
5. SIS further changes the rating of HAPs or even injects HAPs into rated peer lists (if running in the ETM-aware mode, see Subsection 7.8).
6. SIS monitors overlay performance and detects HAPs becoming superfluous.
7. SIS instructs NMS to downgrade a superfluous HAP to a normal peer.

Figure 24 shows the sequence diagram of the above procedure while the following subsections describe the single steps and functionalities in detail.

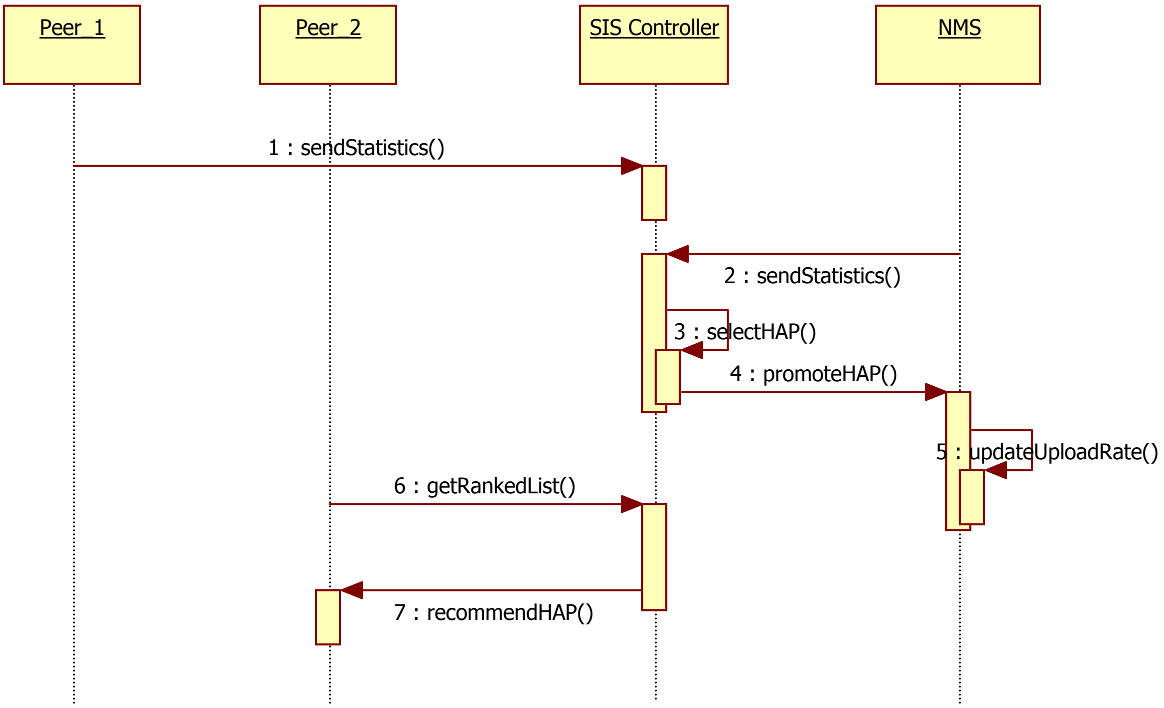

Figure 23: Sequence diagram of HAP promotion procedure (basic scenario).

7.3.2 Dynamic HAP

The dynamic operation of the mechanism is as follows:

- Peers inform SIS about their bandwidth demand.
- SIS detects insufficient upload rate in the local domain.
- SIS determines suitable HAP candidates (lookup in the SIS database).
- SIS selects the most suitable candidates to become HAPs and there number (described in Subsection 7.7).
- SIS promotes the selected candidates by NMS means and thus increases their upload bandwidth (see Subsection 7.4).

- SIS further changes the rating of HAPs or even injects HAPs into rated peer lists (if running in the ETM-aware mode, see Subsection 7.8).
- Promoted peers (HAPs) finish their downloads and continue seeding (see Subsection 7.5).
- SIS monitors overlay performance and detects HAPs becoming superfluous.
- SIS instructs NMS to downgrade a superfluous HAP to a normal peer.

Figure 24 shows the sequence diagram of the above procedure while the following subsections describe the single steps and functionalities in detail.

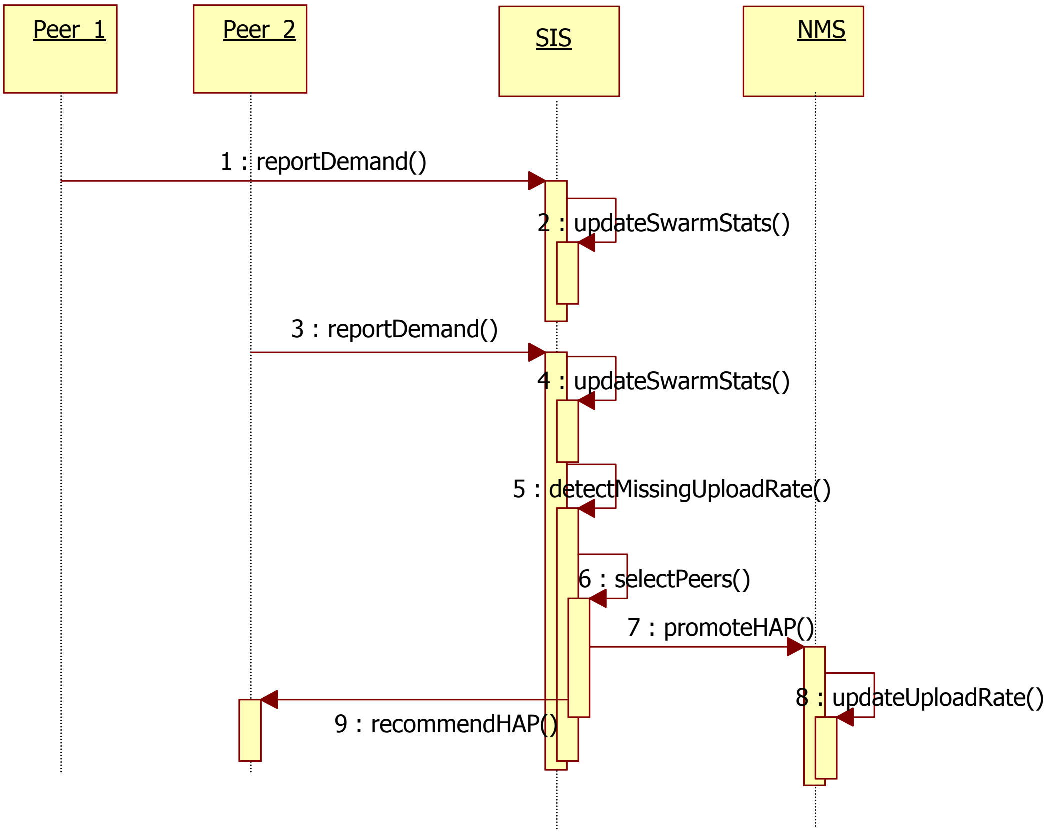

Figure 24: Sequence diagram of HAP promotion procedure.

7.4 Modifying User's Internet Profiles — Methods and Techniques

NGN features allow changing the characteristics of the user's Internet connection. In context of the HAP ETM this allows modifying the performance characteristics of the peer and therefore to:

1. Improve overlay performance
2. Reduce ISP costs by biasing the traffic towards less expensive links
3. Provide incentives to the user and potentially modify his/her seeding/leeching behavior pattern.

The modification of the access characteristics implies several degrees of freedom:

- Time scheduling: allows offering higher access bandwidth rate during the times of low demand (exploiting diurnal traffic characteristics, i.e. different bandwidth demand during different times of the day). In turn the same peers must experience lower access rates during the peak times. With respect to HAP ETM, this approach seems less applicable as it concentrates on changing access characteristic of downloading, rather than uploading (seeding) peer, which is the main actor in HAP.
- Destination based scheduling: the access rate can be modified according to the location of the communication partners. In such a case traffic shaping techniques can be applied to provide different access rates for different IP prefixes. For example, the upload bandwidth to the peers at the same AS can be very high (e.g. 10 Mbps), to other peers in the same AS can be medium (e.g. 2 Mbps), while the bandwidth to the other ASes can be low (e.g. 500 kbps). This creates a natural incentive to exchange more data with local peers. Similar access profiles have been already reported for some ISPs [CB08].

By advertising these changed characteristics to an overlay application it is incited to change the peer selection according to the ISP's costs and/or to reschedule bulk transfers to off-peak hours. The effective utilization of this information in the overlay (and therefore effective utilization of the changed access profile) depends on the overlay's ability to select local peers instead of remote (same as for SIS-enabled locality and using the same functionalities).

The savings potential by dividing bandwidth in local and remote is motivated by the results presented in [CB08]. In this work, a locality-awareness mechanism increases user's performance at much larger scale in the network where ISPs offer much higher internal than external bandwidth.

Under the 95th percentile rule, the distinction between peak hours and low usage hours can decrease the charges of an ISP significantly (similar to what was done in P4P [P4P] and [LR08]).

In order to make the update of the end users' connectivity profile, different options can be considered:

1. To not modify the UL/DL (uplink/downlink) profile but to reschedule the bandwidth assigned to the Internet. That means, typically, in current connectivity services (TriplePlay service) there is bandwidth dedicated to VoIP, IPTV and Internet access. The NGN capabilities of some (pre-)commercial equipments allow the reconfiguration of the bandwidth assigned to each specific service. This reconfiguration takes

around 1s. This capability plus the enforcement of QoS have been considered in the previous chapter in order to allow P2P Streaming server to take advantage of the bandwidth dedicated to operators' services.

2. To modify the UL/DL profile. In current versions of xDSL access, the dynamic re-configuration of the UL/DL profile requires some time to be changed (it requires a complete resynchronization of the DSLAM and the end users' equipment). This re-configuration now takes around 5s-10s. Anyway, this dynamic reconfiguration could be possible in the coming years as far as the configuration capabilities in the networks become more efficient and there are some pre-commercial solutions based on the usage of NGN transport control functionalities (through the Gq interface that is also used in the QoS-aware mechanism, see Section 6). On the other hand, a new standard for ADSL2+ specification that is called the Annex M is being discussed in the ITU-T ([ITU-T G992.5]) proposes a new configuration of the frequencies that result in a higher uplink bandwidth. The following figure shows the expected uplink capabilities with this new distribution.

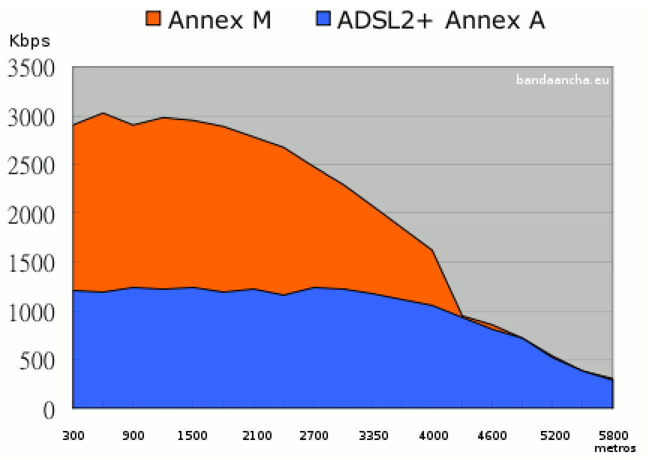

Figure 25: Annex M uplink capabilities (source: http://wiki.bandaancha.st)

Considering a near future scenario where more bandwidth could be dedicated for the uplink (even though that, certainly, the bandwidth dedicated to the downlink should be reduced) and the trend on the automatization of the configuration of the access profile, the current UL bandwidth could be upgraded in a dynamic way from the current maximum of 1.5Mbps to near 3 Mbps).

3. Another possibility is to change the user's profile by the traffic shaping means. Such a scenario is, for example, possible in networks of ISPs that utilize Linux-based equipment to apply traffic shaping [TC].

In summary, the realization of the different access rates for different classes of destination addresses depend on the technology in use. While there seems to be limitations for some proprietary routers, at least the applicability in networks using highly customizable routing and traffic shaping equipment is feasible. The latter case applies to the PrimeTel's network that uses Linux-based traffic shaping and routing equipment.

7.5 Increased Availability of Peers

The mechanism requires local peers to offer the content after they finish the download, i.e. increase “seeding time” if we apply BitTorrent’s terminology. This can assure two important properties:

- Increase availability of the total 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 measureable incentives, which would make becoming HAP clearly beneficial for the end user. Increasing customer UL bandwidth is one such incentive, however its effect is not 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. Other incentives to follow SIS advice and become HAP are also available, however they fall out of scope of this specific ETM and, therefore, will not be discussed in this section.

Given the right ratio of additional costs, the ISP suffers to provide incentives vs. gain from HAP seeding more content locally, 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.

7.6 Swarm Selection in Dynamic HAP

The dynamic HAP mechanism requires indentifying and selecting swarms with lack of upload bandwidth, in which peers would be selected and promoted to HAP. This section describes the swarm selection algorithm applicable in this case.

The computation of the missing local upload capacity “B” and the number of local leechers and seeds can be calculated by SIS by analyzing the local traffic flows, e.g., by Deep Packet Inspection (DPI). However, DPI is quite expensive and costly. Therefore, we propose to reuse the SIS functionality to obtain information about active swarms. Since the ETM mechanism addresses the overlay performance improvement for local users, both the users and overlay provider would be interested in cooperation with the SIS.

Using the basic SIS client protocol, that means just the rating requests from local peers, only the information about the peers active in the local domain can be retrieved. In order to compute the missing upload capacity the peers must provide additional information about the swarm:

- An overlay id (swarm id in case of BitTorrent-like applications).

- Desired download bandwidth (for video streaming case)
- Actual download bandwidth per peer (required in order to distinguish local download from inter-AS)

This additional information can be provided by the local peers using the extension to the basic SIS protocol (custom extension fields as described in [D3.1]). For file sharing applications, where no fixed required download rate exists, a default rate can be applied. On the other hand, the currently achieved download rate can be estimated assuming that all peers download content from other local peers only (worst case from the overlay point of view since it limits the potential number of uploaders).

Optionally, the peers can report only total available bandwidth, and the local part can be estimated as number of local SIS-enabled swarm participants multiplied by there upload bandwidth.

Based on the above information swarms can be sorted by priority, which is calculated as:

$$P = RD - (TU - LU)$$

Where P is sorting parameter, RD – required (or requested) download bandwidth, TU – total upload bandwidth available in the swarm and LU – upload bandwidth available from local peers.

7.7 Selection of HAPs

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 swarm this peer will provide. This relative parameter can be calculated based on history of the peer's behavior. Respective parameters to be taken into account are: average online time per day, average seeding ratio in other SIS swarms, percentage of time peer follows the SIS advise (both as seeder and as leecher). This combination of parameters not only allows to select best potential seeder but also provides a clear incentive to peers to act nicely towards ISP and overlay.

7.7.1 Basic HAP

Let us define an ordered list of HAP candidates **C,** which consists of all local peers which have not yet been promoted to HAP. **C** should be sorted according to the criteria defined above. Then the top **N** peers in **C** are selected into subset **H** and promoted to HAP via NGN interface.

7.7.2 Dynamic HAP

Let us define an ordered list of HAP candidates **C,** which comprises all peers participating in the selected swarm, which have not yet been promoted to HAP. Set **C** should be sorted according to general criteria defined above. We have to select a subset **H** of **C** to become HAPs, and divide the additional upload rate "B" among them. For each peer **p** in **C** the maximum possible increase in the upload bandwidth rate is **b_p**.

We assume that the algorithm knows the parameters “B” and “C”, and that the NGN equipment allows increasing the bandwidth profile of users in a reasonable time scale (see Subsection 7.4).

A greedy algorithm works by selecting the minimum required number of HAPs:

```
Input: $C$: HAP candidates, $B$: missing upload rate, $b:C \text{->} \mathbf{R}$ possible increase in bandwidth rate

while $B > 0$ and $C$ not empty:
    $p \leftarrow$ random from C
    $C \leftarrow C - \{p\}$
    increase $p$'s upload by $b(p)$
    $B \leftarrow B - b(p)$
```

A more sophisticated HAP selection algorithm allows for minimization of intra-domain link utilization. We assume a self-organizing (but locality-aware) neighbor selection mechanism in the overlay. Therefore, the HAPs must be selected in such a way that the number of intra-domain hops to all other peers in the same ISP domain is minimized. This algorithm can be run by the SIS, since it is knows or can compute all the required parameters (L, C and B).

```
Input: $C$: HAP candidates, $B$: missing upload rate,
$b:C \text{->} R$ possible increase in bandwidth rate,
$L$: leechers located in the current ISP domain,
$d: C \times L \rightarrow N$ underlay hop distance between peers

Sort $C$ according to the sum of $d(p,l)$ for all l in $L$ (ascending order)
while $B > 0$ and $C$ not empty:
    $p \leftarrow$ next peer from C (i.e. with the smallest sum of $d(p,l)$ for all l in $L$)
    $C \leftarrow C - \{p\}$
    increase upload p's upload by b(p)
    $B \leftarrow B - b(p)$
```

Since the participants of a swarm are continuously changing, the SIS can promote additional HAPs later on, or downgrade current HAPs depending on the overlay performance in the local domain.

Another possible variant of HAP selection is to split the ISP domain in clusters (e.g., one cluster per DSLAM) and to select HAPs for each cluster using the latter algorithm.

7.8 HAP Promotion, Rating and Monitoring

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.

All actual HAPs should be monitored to make sure that their activity and use of extended resources benefits the swarm and the ISP. The information about currently active HAPs and also the currently active swarms with or without HAPs is stored in the local database. Each time a rating request is received through the SIS-client API the client's IP address is added to the swarm local members list. If no new requests are received during a given time period (e.g., 10 minutes) the peer is removed from the swarm local members list. By monitoring whether the HAP is present in the rating request lists from other peers, the SIS can verify its participation in the swarm. The required information to be stored in the SIS database can be summarized as follows:

swarm_info := (swarm id, list of peers)

peer := (IP address, port, swarm_status, HAP_status)

swarm_status := leecher | seed

seedig_ratio := decimal

seeding_time := integer

followed_sis_advice_ratio: decimal [percentage of cases when peer has followed SIS advice]

HAP_status := HAP | candidate | not_applicable

The following parameters are of utmost importance for monitoring HAP's behavior: seeding_ratio, seeding_time, followed_sis_advice_ratio. Should any of these parameters drop bellow a preconfigured threshold value, HAP will be demoted to a normal peer. In the dynamic scenario these parameters can be considered on a per-swarm basis, while basic case takes into account average throughout of all swarms.

7.9 Next Steps

This ETM mechanism is currently entering the evaluation stage where different alternatives will be compared and the most promising selected for the future use.

Besides the discussed combination with the basic SIS-based rating, this mechanism can be combined with the Inter-SIS communication (see Section 4.2). An example scenario is the possibility for peering ISPs to share HAPs between their domains.

8 Theory and Modelling

This section focuses on the theoretical analysis and modeling of the problems that certain ETM mechanisms attempt to solve. More precisely, in Section 0 the effect of a domain's decision to deploy a locality enforcing ETM mechanism with respect to its neighboring domains is examined. The formulation of the basic problem is provided, along with some possible extensions. In Section 0, a Markov model for estimating the transient distribution of the number of chunks downloaded by each given peer in a BitTorrent swarm is presented. This model can be used for evaluating approaches that try to optimize the time required to download a file.

8.1 Locality Games between Two ISPs

Throughout this deliverable, a multitude of ETM mechanisms was studied, some of which promote traffic localization. Such ETM mechanisms can be very different in essence, like BGP-based locality promotion or the insertion of ISP-owned peers, but the result of their deployment is eventually the same: portion of the overlay traffic is kept inside the domain, decreasing the inter-domain traffic volumes to higher-level domains. One obvious question that arises is about the trade-off between the decrease of the inter-domain traffic and the increase of the intra-domain traffic. Is it beneficial for the domain owner? Is there any negative impact on the performance of the overlay, due to the intra-domain "congestion"?

Another type of questions has to do with the interaction of domains and how the deployment of traffic localization in one domain affects the other domain and how the second domain reacts. To illustrate better these types of interactions, consider the case of two ISPs, clients of the same higher-level ISP, deploying locality promotion mechanisms. The two ISPs can be of Tier 2 or Tier 3. Some of the questions that arise are if the level of localization in one domain affects the level of localization in the other domain, how the unfulfilled demand (in chunks) is covered, and so on.

In order to answer these questions, a model that formulates such a situation and provides the framework to further analyze and evaluate various locality promoting mechanisms is described.

8.1.1 Assumptions

In the formulation that follows, the case of a single swarm is examined. However, the same model can be used if a more general model for P2P flows is considered, where no distinction between swarms exists and the total of overlay traffic flows is considered. In other words, the same model can apply for the aggregated overlay traffic that flows inside a domain and to/from other domain.

Another important assumption is that the demand (in chunks) of all peers in the swarm must be completely satisfied. Thus, in the case where some remote peers are blocked due to locality enforcement policies in their domain, the local peer searches and discovers new peers to complement the supply of chunks. This way, the time for a peer to download the entire file is kept invariant. This assumption can be relaxed if a utility function of the peer is used that considers both download time and delay for finding new peers. To keep the model simple, however, at least at this phase, an ideal case is considered where additional peers are always available to serve other peers requests, and the delay to discover them is negligible.

Regarding some details of the BitTorrent protocol, the optimistic unchoking process is omitted and, thus, the case of perfect T4T is considered. Also, all peers are leechers, with a part of the file in their possession, in order to avoid bootstrapping conditions and consider the swarm in a “steady-state” condition.

Finally, the analysis remains in the flow level, ignoring for the time being the actual volume that circulates in the network. This is symmetric to the case where the typical size of a flow is equal to 1.

8.1.2 Topology and Notation

In the diagram below, the topology of the system to be modeled is presented. Two Tier-2 ISPs are considered, namely domain 1 and domain 2, interconnected through a Tier-1 ISP to the rest of the Internet. Each ISP can reach the other through the Tier-1 ISP as well, i.e., no peering agreement exists between the two ISPs.

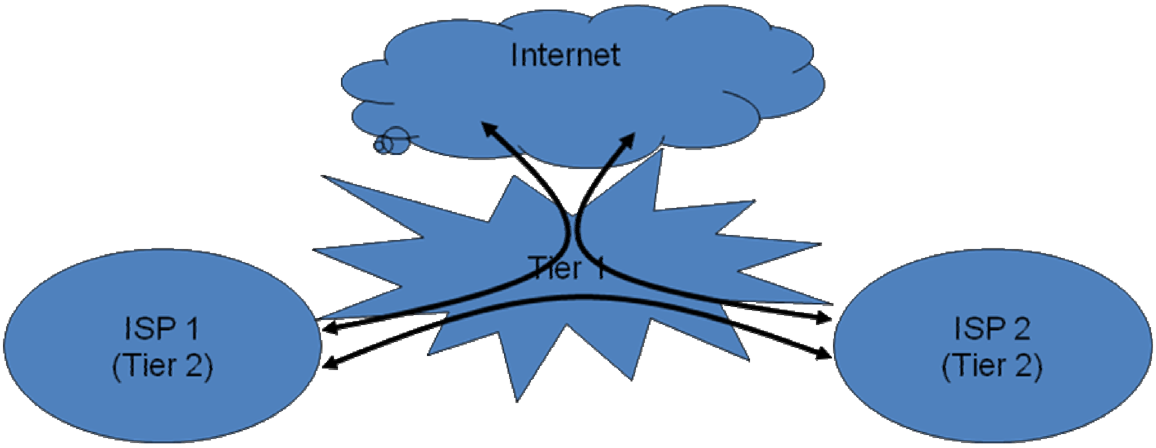

Figure 26: The topology considered by the model.

The following table summarizes the notation to be used in the next sections:

Table 11: Notation

Notation	Description
N_i	The number of peers located in domain i.
N_0	The number of peers located in the Internet.
P_i	The portion of incoming traffic “throttled” to domain i. This can be seen as the outcome of the deployment of localization of traffic in domain i.
a_k	The percentage of the entire swarm that a peer knows of. In other words, this is the number of other peers that can be discovered by a random peer throughout the duration of participation in the swarm.
P_u	The probability of a peer to be unchoked once it has contacted a remote peer.

8.1.3 No Localization

Consider the case where no domain deploys a traffic localization mechanism. In this case, a random peer, through the overlay discovery procedures, it will communicate with a_k $(N_1+N_2+N_0)$ peers during the entire period of its participation in the swarm. We now calculate the number of flows, incoming and outgoing, that can be observed in domain 1. The same analysis applies for domain 2 as well.

Regarding inbound flows, a peer in domain 1 will communicate with $a_k\ N_2$ peers from its domain 2 and it will be unchoked with an average probability of P_u. Thus, the entire population of peers in domain 1 will achieve $N_1\ a_k\ N_2\ P_u$ connections. Hence, this formula depicts the inbound traffic from domain 2. The number of inbound connections from the Internet will be $N_1\ a_k\ N_0\ P_u$ respectively, while the number of "inbound" flows from inside the domain 1 will be $N_1\ a_k\ (N_1 -1)P_u$, which we simplify to $(N_1\ )^2\alpha_k\ P_u$, so that the number of the known peers as mentioned in the beginning of this section is preserved.

Summarizing, the number of total inbound flows to domain 1 is given by the formula:

$$N_1\ a_k\ (N_1+\ N_2\ +\ N_0\ )\ P_u \qquad (1)$$

Following the same analysis as before, one can easily calculate the number of outbound flows from domain 1. This is given by:

$$(N_1+\ N_2\ +\ N_0\ )\ a_k\ N_1\ P_u \qquad (2)$$

It is easy to observe that the number of outbound flows is equal to the number of inbound flows, thus following the rule of Tit-for-tat. In the following sections, when referring to either inbound or outbound flows, it will hold that the same conclusion applies for the other case as well, unless explicitly stated.

8.1.4 Locality Promotion in One Domain

Assume now that one of the two domains, e.g., domain 1, deploys some locality promotion mechanism. As a result the number of inbound flows from the Internet and domain 2 will decrease, while the number of flows inside domain 1 will increase. More specifically, the number of flows from domain 2 will become $N_1\ a_k\ N_2\ (1\text{-}P_1)\ P_u$ and the number of inbound flows from the Internet will be $N_1\ a_k\ N_0\ (1\text{-}P_1)\ P_u$. It is obvious that the number of flows from both source domains is decreased, since $P_1 \leq 1$. On the other hand, the flows inside domain 1 will increase with a factor of x, as depicted in $N_1\ a_k\ (1+x)\ N_1\ P_u$. Due to the assumption of complete satisfaction of the demand (in chunks), the factor x can be easily computed by equalizing the number of inbound flows before and after the deployment of locality promotion. Thus, the increase in the number of flows inside domain 1 is given by the following formula:

$$x = P_1(N_2+N_0)/N_1 \qquad (3)$$

We now have to examine what happens in domain 2, due to the locality promotion in domain 1. Obviously, the inbound flows from domain 1 will decrease to $N_2\ a_k\ N_1\ (1\text{-}P_1)\ P_u$ due to Tit-for-tat. As a consequence, the number of inbound flows from the Internet and from inside the domain 2 will have to increase so as to cover the unfulfilled demand. The decrease for each source domain will be analogous to its size, i.e. the population of peers. Hence, if we assume that the number of increased inbound flows from the Internet is given by $N_0\ a_k\ (1+y)\ N_2\ P_u$ and that from inside the domain 2 is given by $N_2\ a_k\ (1+z)\ N_2\ P_u$, we have to calculate the factors y and z. Since we have two unknown variable we need to equations. The first is derived from the preservation of the demand-supply relationship and the second is given by the fact that the increase is proportional to the size of the respective domain, i.e. by $y/z = N_0/N_2$. Solving the system of two equations, we get that:

$$\begin{aligned} y &= N_0\, w, \\ z &= N_2\, w, \qquad (4) \\ w &= N_1 P_1 / (N_0^2+N_2^2) \end{aligned}$$

8.1.5 Locality Promotion in Both Domains

If both domains deploy locality promoting mechanisms, it is reasonable to assume that the domain with the strictest policy will define the volume of traffic exchanged, i.e. the number of flows, between the two domains. In other words, the domain with the highest P_i value will dictate the number of inbound/outbound flows from/to the other domain. Let us examine domain 1 (the same analysis holds for domain 2). Then, the flows from/to domain 2 will be equal to $N_2\ a_k\ N_1\ (1\text{-}Max[P_1,\ P_2])\ P_u$. At the same time, the number of flows from the Internet and from inside domain 1 will have to increase. The number of flows from the Internet cannot be greater than what the locality promotion effect (P_1) allows: $N_0\ a_k N_1\ (1\text{-}e)\ P_u$. The rest of the traffic will be covered internally: $N_1\ a_k\ (1+f)\ N_1\ P_u$. The factors e and f are given below:

$$e = P_1 - N_2/N_0\ (Max[P_1,P_2]-P_1) \qquad (5)$$

$$f = P_1(N_0+N_2)\ /N_1 \qquad (6)$$

Observe that the amount of increase of intra-domain traffic when locality is enforced in only one domain or in both domains is the same (i.e., $x = f$). If domain 1 is the one with the strictest locality promotion policy, then this case is equal to the case where only domain 1 promotes locality, since P_1 is what affects the exchanged traffic between the two domains. In case where domain 1 does have the higher P_i value, the traffic exchanged with domain 2 is lower than what the local localization policy permits, but the difference can be covered by contacting more peers for the Internet. Note that the localization parameter specifies the level of inbound traffic allowed by the domain, without specifying how this traffic will be distributed among the various neighboring domains.

8.1.6 Cost Model

In the previous sections the promotion of locality and how the traffic patterns are affected was modeled, based only on the Tit-for-tat rule. In order to better analyze and understand the effects of locality promotion, a notion of cost in the respective actions needs to be attached. Cost, in the specific environment, can be expressed in three ways: a) the per flow cost of inter-domain traffic, i.e. the payments to Tier 1 for the inbound/outbound traffic volumes, b) a cost for discovering new peers when locality enforcement in one domain limits the supply of that domain and c) a cost per flow related to the performance of the underlay, e.g., in terms of delay or congestion. This last type of cost is what actually differentiates the domains since the interconnection characteristics of each domain and the level of traffic on its links affect the per flow performance cost function.

With such costs (or similar ones) in place, each domain can now solve a cost minimization problem which takes into consideration the Tit-for-tat rule along with the attached costs in order to decide on its neighbor set. In the following subsections we provide a short description for each of the cost categories.

8.1.6.1 Inter-domain Traffic Cost

This type of cost is the simplest that can be defined and directly relates to the actual money a Tier-2 ISP pays to the Tier-1 ISP for its traffic to be routed to/from the Internet. Hence, parameter c defines the unit cost for transmitting one unit of flow/traffic over the inter-domain link.

8.1.6.2 Discovering New Peers and Cost of Separation

As already mentioned in the previous sections, the parameters x and f serve as a metric of the additional peers to be discovered so as to fulfill the remaining demand. Hence, it's trivial to associate a unit cost with those parameters to specify how costly the discovering of new peers is. However, the discovery itself is not a costly operation and is a typical process provided by all overlay networks. What matters most is the distribution of the peer's neighbor set among the different domains. In other words, as the portion of local peers in a neighbor set increases, so does the probability that the swarm will be partitioned into smaller, localized and isolated islands of overlay content. The separation of a swarm affects its performance. Thus it has to be captured and considered when a domain decides on the level of locality which affects the formation of the peers' neighbor set.

To adapt the above observations in our model, the cost of separation would increase as the term $a_k\ (1+x)$ gets closer to 1. In this case, the neighboring set of a peer in domain 1 would include all local peers, which could lead to swarm partitioning or even starvation, especially in the case that $a_k\ (N_1+N_2+N_0) \geq N_1$. Hence, a good representation of separation cost in this case would be:

$$separation\_cost(x) = k\frac{1}{1-\frac{x}{\frac{1}{a_k}-1}} \qquad (7)$$

where c_k is the unit cost of separation.

8.1.6.3 Performance Cost

Performance cost is probably the most important type of cost, since it is the one that can differentiate the domains and affect their decision on whether and how strongly they will deploy locality promotion mechanisms. In order to correctly capture this type of cost, the end-to-end delay experienced between a source-destination pair of peers must be considered. The unit cost of delay is defined as d and the typical (average) size of a flow is L. In the previous sections, it was taken that $L=1$. Furthermore, the capacity of the inter-domain link of ISP i is C_i in both directions (i.e., inter-domain links are symmetric). The Internet is assumed to be composed of M average domains (ISPs) that are more than two-hops away from the rest of domains 1 and 2. This is done in order to capture the preference (in terms of performance) of a peer in domain 1 to connect to a peer in domain 2 which will offer better performance than a peer located somewhere else in the Internet. For the intra-domain traffic, the intra-domain topology is a star topology and C_{ij} is the capacity of an intra-domain link from a peer j to ISP's i central hub. A final parameter introduced is the utilization level R of each link. For the inter-domain links the level is denoted with R_i, while for the intra-domain links the respective level is denoted with R_{ij}.

Following the above formulation, the performance cost for a flow between domains 1 and 2 (without any locality promotion mechanism present) could be written as:

$$\mathit{delay\_cost} = d\,(\mathit{no\_of\_intra\_flows}(1)\,L/\,C_{1i})\,L(1/(1-R_{1i})) + d\,(\mathit{no\_of\_inter\_flows}(1)\,L\,/\,C_1)\,L\,(1/(1-R_1)) + d\,(\mathit{no\_of\_inter\_flows}(2)L/C_2)\,L(1/(1-R_2)) + d\,(\mathit{no\_of\_intra\_flows}(2)L/\,C_{2i})L(1/(1-R_{2i})) \qquad (8)$$

where, for domain 1, it holds:

$$\mathit{no\_of\_intra\_flows}(1) = a_k\,(N_1 + N_2 + N_0\,)\,P_u\,,$$

$$\mathit{no\_of\_inter\_flows}(1) = N_1\,a_k\,(N_2 + N_0\,)\,P_u$$

and for domain 2 the formulas are symmetric.

For the case of communication between domain 1 and a remote domain 'm' in the Internet, the previous formula is transformed to

$$\mathit{delay\_cost} = d\,(\mathit{no\_of\_intra\_flows}(1)\,L/\,C_{1i})\,L(1/(1-R_{1i})) + d\,(\mathit{no\_of\_inter\_flows}(1)\,L\,/\,C_1)\,L\,(1/(1-R_1)) + d\,(\mathit{no\_of\_inter\_flows}(m)L/C_m)\,L(1/(1-R_m)) + d\,h\,\mathit{no\_of\_core\_flows}\,L(1/(1-R_c)) + d\,(\mathit{no\_of\_intra\_flows}(m)\,L\,/\,C_{mi})\,L(1/(1-R_{mi})) \qquad (9)$$

where

$$\mathit{no\_of\_intra\_flows}(m) = a_k\,(N_1 + N_2 + N_0\,/M)\,P_u\,,$$

$$\mathit{no\_of\_inter\_flows}(m) = (N_0/M)\,a_k\,(N_1 + N_2)\,P_u\,,$$

$\mathit{no\_of\_core\_flows} = N_0\,a_k\,(N_1 + N_2)\,P_u$ and

R_c = the expected utilization level of the core links and

h = the number of hops traversed by flows originating/destined from/to a remote Internet domain.

From the above, it is observed that there is a clear differentiation, in terms of performance, between "local/neighboring" domains and remote ones.

8.1.7 Next Steps

Having presented the formulation of the problem, the next steps will be to carefully design the cost optimization problems, analyze the locality enforcement game between the two domains, construct the reaction curves and search for an equilibrium. As extensions of the above formulation, the case of introducing separately the number of leechers and seeds in a domain will be considered as well as to start working with non-symmetric intra- and inter-domain links, which is typically the case in modern networks.

8.2 Markov Model for the Evaluation of ETM mechanisms

BitTorrent [C03] is a very popular peer-to-peer file sharing system, which generates huge volumes of the Internet traffic. This traffic leads to increased congestion in the underlying physical network, as well as to increased costs for the ISPs particularly for inter-domain traffic. Several optimization approaches have been proposed in order to achieve reduction of inter-domain traffic and/or download completion times. Therefore, the analysis of BitTorrent and the different optimization approaches, as well as the performance evaluation of them is an interesting and important subject.

In this section, we present a markovian model that tries to evaluate the impact of such optimization approaches of BitTorrent, in terms of completion times, by means of probabilistic analysis. Numerical results derived by employing the model's equations in Matlab, as well as initial steps towards the model's verification, are also presented.

8.2.1 Background

There is a very extensive literature on peer-to-peer performance evaluation. Many relevant works are exclusively based on analysis of measurements from actual systems and/or simulations. In this subsection, we briefly overview selected articles that include models for the performance evaluation of peer-to-peer file-sharing with emphasis on BitTorrent. An extended overview of literature on peer-to-peer performance evaluation can also be found in [D2.2]. In [KR06], Kumar and Ross analyze the minimum distribution time for a file in a system with seeds and leechers. In particular, by employing a deterministic fluid-flow model, they provide a lower bound that involves the download and upload rates of the various peers and then show that this bound can indeed be achieved by scheduling the various transfers of the file appropriately.

In [YV03], Yang and de Veciana initially deal with the capacity attained in peer-to-peer systems due to their fundamental feature that a peer A can serve other peers while still downloading the missing content. The authors thus develop a simple deterministic model that shows the effect of this feature in the transient case similar to that of our model, with only one of the peers stores initially the content file. In particular, it follows that the average delay per peer is logarithmic in the number *N* of peers. Moreover, if the file is partitioned into *m* chunks, then due to pipelining, the average delay is reduced by a factor of *m*. The authors of [YV03] also develop a two-dimensional markovian model for the steady state analysis of the system. The problem of calculating the completion time is also studied by Mundiger *et al.* in [MWW06], under more general assumptions. These authors derive the optimal centralized upload schedules both for the case of a central server and for the case of a decentralized system with all peers having equal capacities. They then develop another model for the transient evolution of a peer-to-peer system, whereby the number *N(t)* of peers that have already downloaded the file by time *t* is modeled as an age-dependent branching process with a family size of 2 in each generation. Therefore, the expected value *E[N(t)]* grows exponentially with time *t*, provided that there is sufficient demand in the system.

In [QS04], Qiu and Srikant initially present a deterministic fluid model for the performance of BitTorrent. The model of [QS04] is motivated by the markovian model of [MWW06] and comprises the same parameters. The authors present (among others) a probabilistic model for the evaluation of the parameter η that represents the effectiveness of BitTorrent in the sense of contribution degree of each downloader to the other ones. In particular, this model quantifies the probability η that a particular downloader has a chunk that is among the ones needed by another one. In fact, the assumptions made with respect to the distribution and selection of chunks possessed by a peer are the same to those our model, as described in Section 8.3.2. It turns out that $\eta \approx 1\text{-}(logN/N)^k$, where k is the total number of chunks of the file. This implies that for a large file (i.e. for a large value of k), $\eta \approx 1$; that is, a downloader contributes to the others almost as much as a seed. In [MWW06], Leibnitz *et al.* present a fluid flow model for evaluating the transient performance, in particular reliability and efficiency, of content distribution services that can be realized by traditional client/server architectures or peer-to-peer networks involving malicious peers.

Besides [QS04], several articles deal with the steady-state performance analysis of BitTorrent with dynamically varying population. Next, we briefly overview selected such articles. In [GFS03], Ge *et al.* present a steady-state queueing model that comprises all the main ingredients of a peer-to-peer file sharing system, while applies to a variety of such systems. This model is then solved analytically by means of an approximation based on bottleneck analysis, and it is validated by means of simulations. The work of Fan *et al.* in [FCL06] deals with an important tradeoff arising in BitTorrent, namely: achieving fast downloads vs. keeping “fat” (i.e. resourceful) peers in the system as much as possible in order to help other peers attain a fast download. The latter objective appears to be unfair for the fat peers, thus giving rise to a tradeoff between performance and fairness (i.e. better service for peers contributing more to the system), which is investigated in the [FCL06] by means of steady-state analysis.

8.2.2 The Markov model

The proposed Markov model estimates approximately the transient distribution of the number of chunks downloaded by each given peer in a BitTorrent swarm. Based on this, we can estimate other performance measures such as the upper tail of the distribution of the time required for a peer to complete downloading a file. The model can thus be employed to analyze performance properties (e.g., scalability) of a BitTorrent-like network, and to evaluate optimization approaches, such as insertion of ISP-owned Peer (IoP) or ISP-owned Seed (IoS). For analytical tractability, the model employs certain simplifications of BitTorrent, which are presented below.

8.2.2.1 Assumptions

The Markov model is a discrete time model. Originally, we consider N+1 peers in the swarm; N downloaders with initially 0 chunks and 1 seed which has all K chunks, namely the complete file. For simplicity we assume that after a downloader finishes downloading, it serves as a seed; therefore, the swarm population remains constant. We also assume that chunks are selected by peers at random and uniformly, rather than according to some chunk selection method, *e.g.,* rarest first replication. Thus, due to symmetry among chunks in their initial distribution, the system’s state is specified completely by the number of chunks that each peer has acquired until the end of each step n, or equivalently by the number of peers out of N that has 1,2,..., or K chunks at step n. The number of different states with this formulation would be equal to the number of choosing K elements out of N with repetition. However, due to the prohibitively large state space, we resort to an approximation. We study the evolution of peer D; let D be a tagged peer out of the set of the N downloaders. The state of D (as well as that of any other peer) belongs to {0,1,...,K}.

Due to symmetry, the equations derived for D and the marginal distribution of its state can actually characterize each of the other downloaders.

Additionally, we assume that each peer can unchoke *C* others, where *C* is a parameter that represents the actual number of chunks that a peer can upload within a given step (i.e. choking interval). For instance, the protocol allows each peer to unchoke up to *m*=5 other peers per step (i.e. unchoking interval). However, a peer with nominal upload bandwidth of 512 kbps can only upload/serve *C*=2 chunks of 256 KB (i.e. the typical chunk size). For tractability reasons, 'tit-for-tat' is ignored, as also done in [QS04]; instead random unchoking is considered. However, only 0, 1 or 2 chunks can be downloaded per peer in every step; namely, if a certain peer is unchoked by more than 2 peers, it can download only *C*=2 chunks. Due to these assumptions, our Markov model corresponds to a version of BitTorrent where all decisions are made at random, which therefore is expected to have inferior performance than the original BitTorrent. Consequently, the results obtained by our model are expected to constitute bounds of the actual performance of BitTorrent. The aforementioned approximations are evaluated by means of simulations (Section 8.3.4).

8.2.2.2 Evolution of the Markov Chain

A. Native BitTorrent: The transient marginal distribution of the state of a regular peer at step *n* is denoted by $P(n)=[P_n(0), P_n(1),\ldots, P_n(K)]$, where $P_n(n)=\Pr[X_n=k]$. The transient distribution at step *n*+1 is derived as follows: $P_{n+1}(k)=P_n(k-2)\ P_{n+1}(k-2,k)+ P_n(k-2)\ P_{n+1}(k-1,k)+ P_n(k)\ P_{n+1}(k,k)$, where the transition probability $P_{n+1}(k-2,k)$ corresponds to the event that peer D is unchoked by two or more other peers and finds useful chunks in at least two of them at step *n*+1, given that peer D has *k*-2 chunks at the end of step *n*; $P_{n+1}(k-1,k)$ is defined similarly, while $P_{n+1}(k,k)$ is the transition probability that peer D is either choked by all peers or does not find a useful chunk at any peer it is unchoked by, given that it has *k* chunks at step *n:*

$$P_{n+1}(k,k) = \mathrm{E}_{N_e(n),N_s(n)}\left[\left(1-\frac{CS}{N-N_s(n)}\right)\left(1-\frac{CL}{N-1-N_s(n)}Q_{n+1}(k)\right)^{N-1-N_e(n)}\right].$$

Similarly $P_{n+1}(k-1,k)$ is derived, whereas $P_{n+1}(k-2,k)$ equals the difference of 1 minus the sum of $P_{n+1}(k-2,k-1)$ and $P_{n+1}(k-2,k-2)$, which have already been calculated. Note that $Q_{n+1}(k)$ represents the probability for a peer to find a useful chunk given that it is unchoked and has *k* chunks; $Q_{n+1}(k)$ is calculated by the formula also employed in [QS04]. The equations that describe the Markov model evolution are analytically presented in Appendix C.

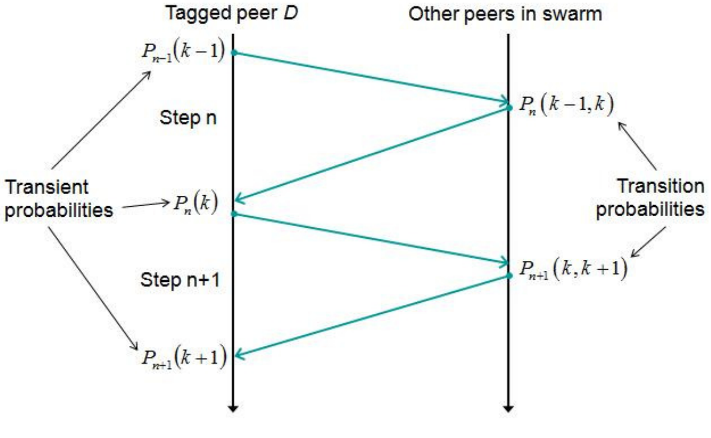

Figure 27: Iterative process

The calculation of the transient probabilities at every step is an iterative process; the procedure is depicted in Figure 27. Since the population size is fixed, there is a deterministic upper bound in our setting for the overall completion time. Since the completion time of the last peers might be quite large, this upper bound can be loose. Thus, we choose to consider as a proxy for overall completion time the time *n** when a large portion of the peers' population (e.g., 90%) will have finished downloading; that is, they have *K* chunks. Using the transient marginal distribution of the state of D, it follows for *n**: $n^*=\min\{n:P_n(K)>G\}$, where *G* equals 0.90 or 0.95. Note that *n** is expressed in steps that correspond to choking intervals. In order to convert *n** to actual completion times expressed in sec., we can multiply *n** by the duration of the choking interval, e.g., 10 sec. The equations that characterize the evolution of peer D involve the distribution of the numbers $N_s(n)$ of downloaders that have *K* chunks at step n, and $N_e(n)$ of downloaders that have 0 chunks at step *n*. Note Initially $N_s(x)\approx 0$, while $N_e(n)$ tends to 0 as time progresses. Since *K* is large, the two sets are non-overlapping. These random variables are approximately taken to be independent and binomially distributed each. This amounts to assuming that the states of different peers (which are identically distributed to that of D) are independent. I.e., for *a=s,e*, we take that

$$P\left(N_a\left(n\right)=z\right)=\binom{N-1}{z}\left(P_n\left(B\right)\right)^z\left(1-P_n\left(B\right)\right)^{N-1-z}.$$

B. BitTorrent with IoP (ISP-owned Peer) Insertion: Since the IoP is equipped with much higher bandwidth than the regular peers, we assume that it is always unchoked by the original seed and in fact twice; namely the IoP downloads 2 chunks at every step with probability 1, both of them from the seed. Note that the IoP behaves as a regular peer until step *n=K/2*, and from that step ahead as a seed. Certainly, the transition probabilities of the regular peers change accordingly. We also consider a third case, where an IoS (ISP-owned seed) insertion. This is similar to having an IoP inserted which has the complete file from the beginning, and thus provides a lower bound to the performance attained with the IoP.

8.2.3 Numerical Evaluations

8.2.3.1 Monotonicity

First, we employ our Markov Model in order to investigate monotonicity of performance w.r.t the number of downloaders. We consider a swarm with *N* peers, each having upload capacity *CL*=2 and 1 original seed with upload capacity *CS*=2. In all calculations also presented below, the file size is taken equal to 40 MB, consisting of approximately 160 chunks of 256 KB each. Figure 28 depicts the completion times *n** for different swarm sizes, e.g., *N* in {10,15,…,160} for three different values {0.90, 0.95, 0.99} of *G*. Note that the possible states of the model are 161, and the first step when $P_n(K)$ is non-zero is *n*=81. This lower bound follows from the fact that up to 2 chunks per peer can be downloaded at each step. Observe that the completion times increase as the swarm size increases. Thus, our model *verifies monotonicity* of completion time when the initial number of leechers increases.

8.2.3.2 Impact of the Original Seed's Capacity

Bindal *et al*. [BCC+06] argue that original seed's capacity has an important impact on the completion times, while Le Blond *et al.* [BLD08] argue that it is critical to the high chunk diversity, which impacts peers' completion times. To estimate this impact, we consider a swarm of *N*=100 peers with upload capacity *CL*=2, and 1 original seed with varying upload capacity *CS*=*c*. Figure 29 presents the completion times' reduction achieved by different values of varying seed capacity *c* {2,4,…,20}. Observe that as the seed capacity increases, the reduction of the completion times decreases, which agrees with the above results.

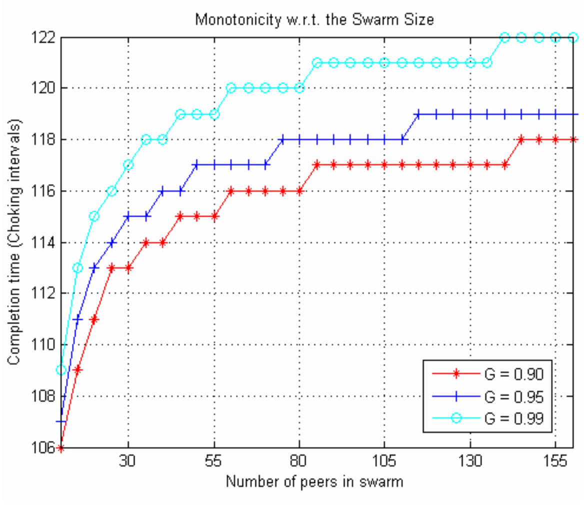

Figure 28: Completion times for *N* in {20,25,..,160} when *G* is 0.90, 0.95, and 0.99

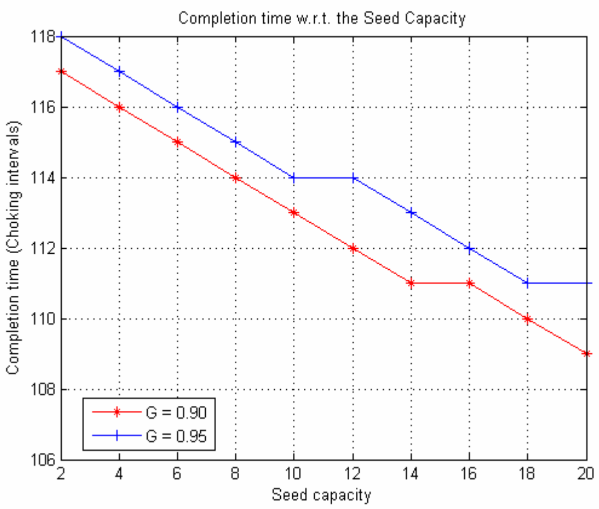

Figure 29: Completion times for seed capacity *c* in {2,4,6,…,20} for *G* equal to 0.90 and 0.95

8.2.3.3 Evaluation of the IoP/IoS Insertion

Papafili *et al.* [PSS09] consider a simple two-AS BitTorrent network. By means of simulations, it is shown that the IoP insertion in one AS, achieves important reduction of end-users' completion time and inbound inter-domain traffic. The objective is to investigate the impact of the IoP insertion to the completion times employing the Markov model. Considering the original setup the number of peers *N* is varied {15,20,…,160} when *G*=0.95 for the following cases: a) no IoP/IoS , b) IoP insertion, and c) IoS insertion. Both IoP and IoS are assumed to have capacity *CP*=10. Figure 30 shows the overall completion time for the three aforementioned cases. Observe that the IoP insertion improves overall completion time up to 10% for small or medium swarms. Although slightly better, the performance achieved by the IoS constitutes a lower bound for that achieved by the IoP. Note that a moderate value for the IoP/IoS capacity has been chosen, which also has been kept fixed as the swarm size increases.

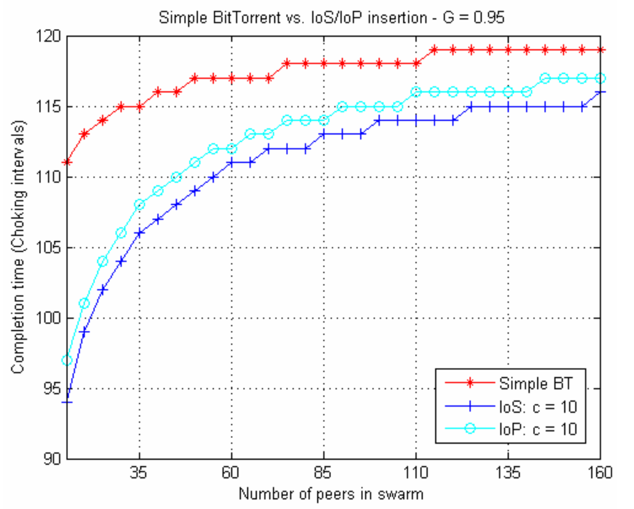

Figure 30: Completion time for Simple BitTorrent vs. IoP/IoS

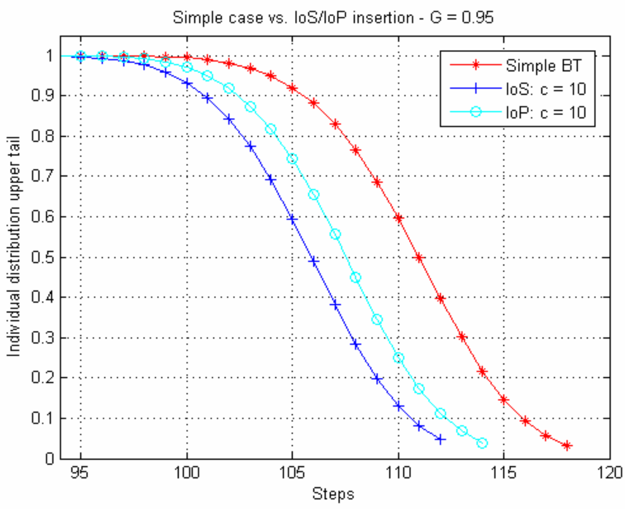

Figure 31: Simple case vs. IoS/IoP insertion

Figure 31, depicts the upper tail (i.e. $U(n)=1-P_n(K)$) of the distribution of individual completion time for a medium swarm size, i.e. N = 80. The ordering of the curves is consistent to that of Figure 30. Considering the assumptions of [YV03] in our case would give an average completion time per peer of $(\log_2 N+(2K-1)/2)/CS$, which is a loose lower bound; e.g., it equals 41.78 for N=100 , K=161 and CS=4, which is considerably lower than the estimates of Figure 30. Also, [KR06] is based on optimal scheduling; for the cases of Figure 31, [KR06] would give the straightforward lower bound of 80, since each peer can download at most 2 chunks per slot and the total number of chunks is 160.

8.2.4 Simulations

8.2.4.1 Number of downloaded chunks per choking interval

In the model, we restricted the maximum possible number of downloaded chunks per step to 2, since this is a realistic assumption while it does not give rise to too complicated equations. In order to decide on adopting this assumption, we performed simulations in *ns-2* platform [E07], investigating the number of chunks downloaded per peer in a choking interval. In particular, we monitored the number of the new chunks that every peer obtained at the end of each choking interval. Figure 32 presents the number of downloaded chunks by a specific peer per slot in a 50 peer swarm, where each peer has a download/upload bandwidth of 4096/512 kbps. The maximum such number is 5, which is quite far away from our assumption. However, the average number of chunks is 1.4545, while for 80% of the choking intervals, the number of chunks was less than or equal 2. Furthermore, Figure 33 shows the average number of chunks downloaded by all peers of the swarm in each choking interval. Note again, that only in 20% of the cases the average number of peers is greater than 2. Finally, the average number of downloaded chunks over all peers equals 1.449<2. Thus, the underlying assumption of at most 2 chunks downloaded per step is satisfactorily accurate.

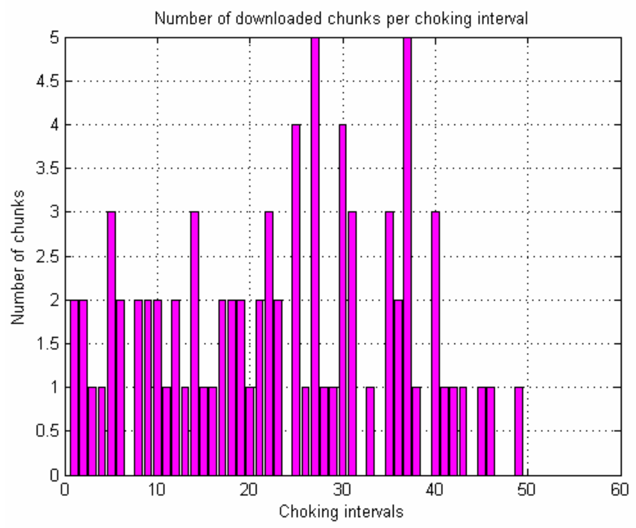

Figure 32: Number of chunks downloaded by a specific peer per choking interval.

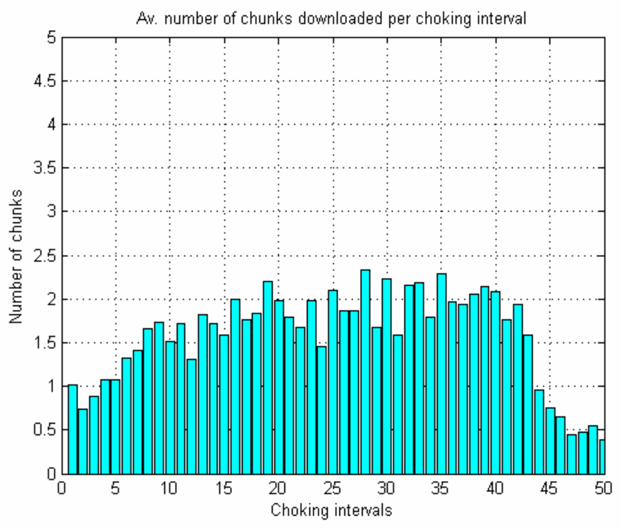

Figure 33: Average number of chunks downloaded per choking interval over all peers.

8.2.4.2 Simplified BitTorrent vs. native BitTorrent

We have run simulations for the implementation of the BitTorrent protocol and our simplified implementation and have compared them in terms of completion times. Topology and access bandwidths are the same as that in [PSS09]. All peers, including the original seed, start together at time 0 sec. In Figure 34, we compare the completion times achieved by the two implementations for swarms of *N*=50, 90, respectively. The top graphs depict the individual completion times, whereas the bottom graphs show the percentage of the relative difference of completion times. Observe that regardless of the swarm size increase, the difference remains almost fixed (around 10%). This also applies when an IoP is inserted in the network. Although the approximation of the performance of BitTorrent by that of our simplified implementation one is not accurate enough, we can expect that due to this uniform difference the simplified protocol can lead to useful conclusions regarding the impact of similar optimization approaches.

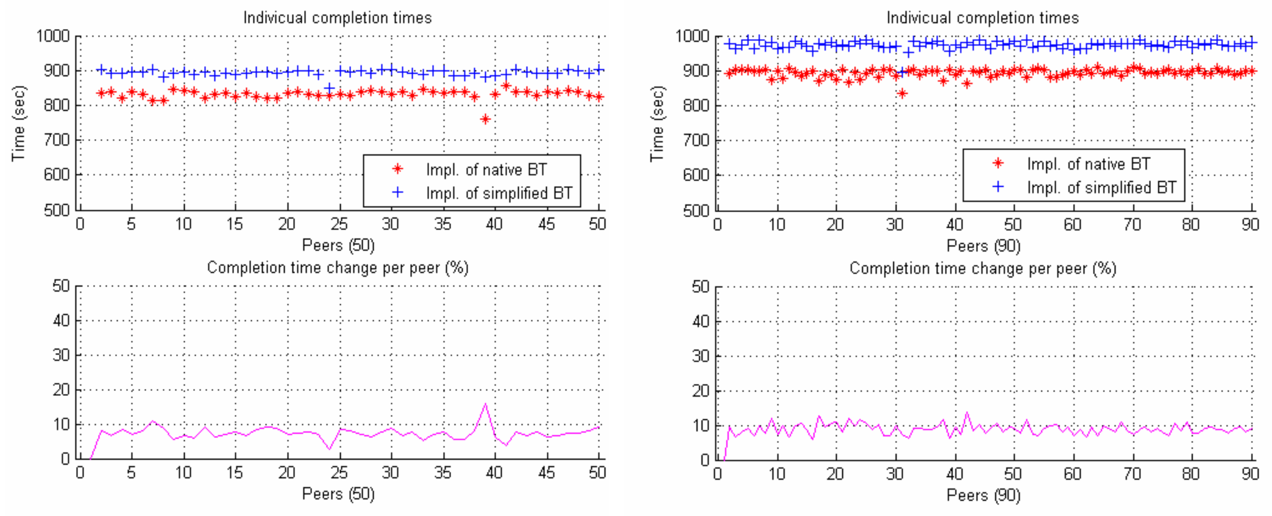

Figure 34: Top-graph(s): Completion times achieved for the implementation of native BT and simplified BT. Bottom-graph(s): Percentage of the relative difference

In summary, the results have shown that completion times achieved in the simplified BitTorrent are constantly 10% higher than completion times achieved by the BitTorrent implementation; therefore results derived by the model can constitute bounds of the actual BitTorrent performance, as it was expected. Next steps include the comparison of results derived by the model's equations with results derived by simulations.

9 Summary and Conclusion

This deliverable presents the final version of the Economic Traffic Management mechanisms modelled in SmoothIT. Based on the multitude of ETM mechanisms presented in *D2.2 – ETM Model and Components,* this deliverable concentrates on the most promising mechanisms. Their selection was done according to the project requirements and the preliminary assessment results together with the applicability to the trials.

The first ETM mechanism addressed by the SmoothIT project, *SIS-enabled locality-awareness,* was refined. The specification was evaluated through a simulation study, which proved high potential of BGP-based peer rating. Additionally, an extension with dynamic locality enforcement was presented.

A set of possible solutions for collaboration of several SIS instances was designed and discussed. They deal with two possible kinds of asymmetry while applying BGP-based locality: route asymmetry and information asymmetry. In the latter case, collaboration among different kinds of ISPs is considered (peering vs. source and transit ISPs). These approaches allow providing ratings of remote peers based on information not available in the local domain. A stronger approach of ISP-collaboration involves the split of content to be downloaded among (peering) ISPs.

The alternative to improve ISP costs by inserting ISP-owned peers in the local network was explored. Here, different issues such as swarm selection and unchoking policy, were discussed and different algorithms were presented. As pointed out, the collaboration with an SIS improves the applicability of this ETM mechanism.

The possibility to offer higher QoS to overlay applications was covered by the QoS-awareness mechanism. This is especially applicable for more sensitive overlay applications such as live streaming or VPN networks.

Moreover, the improvement of overlay performance by changing the user's bandwidth profile and promoting them to Highly Active Peers was presented. This mechanism is still under development.

After the presentation of the mechanisms themselves, their application scenarios and expected behaviour are discussed in detail. Further on, theoretical and simulative results are presented for the most progressed ETM mechanisms. Other mechanisms are currently under ongoing assessment procedure and the results will be documented in the upcoming deliverable *D2.4 – Performance, Reliability, and Scalability Investigations of ETM Mechanisms*.

The ETM mechanisms presented in this deliverable, and especially their specification, serve as inputs to Work Package 3 for the prototype implementation. This will provide valuable insights into their feasibility and consistency, while the results of Work Package 4 will show the performance of the developed ETM mechanisms in a real environment.

References

[AD01] K. Aberer and Z. Despotovic: *Managing Trust in a Peer-to-Peer Information Syste,*; International Conference on Information and Knowledge Management, November 2001.

[AFS07] V. Aggarwal, A. Feldmann, C. Scheideler: *Can ISPS and P2P users cooperate for improved performance?* SIGCOMM Computer Communication Review, Volume 37, Number 3, Pages 29-40, 2007.

[ALTO] S. Kiesel, L. Popkin, S. Previdi, R. Woundy, Y R. Yang: *Internet Draft: Application-Layer Traffic Optimization (ALTO) Requirements*, April 2009.

[BCC+06] R. Bindal, P. Cao, W. Chan, J. Medval, G. Suwala, T. Bates and A. Zhang: *Improving Traffic Locality in BitTorrent via Biased Neighbor Selection.* IEEE International Conference on Distributed Computing Systems, 2006

[BLD08] S. Le Blond, A. Legout, W. Dabbous, Pushing BitTorrent Locality to the Limits, INRIA, Dec. 2008

[C03] B. Cohen, *Incentives build robustness in BitTorrent*, First Workshop on the Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.

[CB08] Choffnes, D. R. & Bustamante, F. E. Taming *the Torrent: A practical approach to reducing cross-ISP traffic in P2P systems,* SIGCOMM Computer Communications Review, ACM, 2008, Volume 38, pages 363-374

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

[D1.2] The SmoothIT project: *Deliverable D1.2 – Commercial Application Classes and Traffic Characteristics*, July 2009

[D2.1] The SmoothIT project: *Deliverable 2.1: Self-Organization Mechanisms for Economic Traffic Management;* June 2008

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

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

[E07] K. Eger, *Simulation of BitTorrent Peer-to-Peer (P2P) Networks in ns-2*: http://www.tu-harburg.de/et6/research/bittorrentsim/

[ES003] ETSI ES 282 003, Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource *and Admission Control Sub-System (RACS): Functional Architecture*.

[FCL06] B. Fan, D. Chiu, J. Lui, *The Delicate Tradeoffs in BitTorrent-like File Sharing Protocol Design*, Proceedings of the 2006 IEEE international Conference on Network Protocols. ICNP. IEEE Computer Society, Washington, DC

[GFS+03] Z. Ge, D.R. Figueiredo, J. Sharad, J. Kurose, D. Towsley, Modeling peer-peer file sharing systems, INFOCOM 2003, IEEE, pp. 2188-2198 vol.3, 30 April 2003

[GSMA08] *The GSMA's IP Packet Exchange (IPX) project*, http://www.gsmworld.com/about/people/emc.shtml

[HGI] *Home Gateway Initiative (HGI),* http://www.homegatewayinitiative.org

[IETF75-3] *Sailor: Efficient P2P Design Using In-Network Data Lockers*, R. Alimi, H. Liu, E. Li, R. Yang, D. Zhang, R. Zhou

[Ipo2009] *Ipoque Internet Study 2008/2009*, http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009

[KR06] R. Kumar, K.W. Ross, *Peer-Assisted File Distribution: The Minimum Distribution Time*, Hot Topics in Web Systems and Technologies, 2006. HOTWEB '06

[KraTR] G. Kramer, On generating self-similar traffic using pseudo-Pareto distribution", Technical Report, http://wwwcsif.cs.ucdavis.edu/~kramer/papers/self_sim.pdf

[LCL+09] B. Liu, Y.Cui, Y. Lu, Y. Xue, *Locality-Awareness in BitTorrent-like P2P Applications*, IEEE Transactions on Multimedia, Vol. 11, No. 3, April 2009 361

[LHW+07] K. Leibnitz, T. Hossfeld, N. Wakamiya, M. Murata, Peer-to-Peer vs. Client/Server: Reliability and Efficiency of a Content Distribution Service", Proceedings of the 20th International Teletraffic Congress (ITC20), Ottawa, Canada, June 2007

[LLC06] E. Lawrence, J. Lawrence, G. Culjak, *Legal and Technical Issues Management Framework for Peer-to-Peer networks*, Journal of Theoretical and Applied Electronic Commerce Research, Volume 1, 2006

[LR08] Laoutaris, N. & Rodriguez, P. Good *Things Come to Those Who (Can) Wait*, ACM HOTNETS-VII, 2008

[MWW06] J. Mundinger, R. Weber, G. Weiss, Analysis of peer-to-peer file dissemination", SIGMETRICS Perform. Eval. Rev. 34, 3 (Dec. 2006)

[Myth09] E. Marocco, A Fusco, I. Rimac, V. Gurbani, *Mythbustering Peer-to-peer Traffic Localization*, IETF Internet draft, July 2009.

[P2PNext] P2P Next EU/ICT project, www.p2p-next.org

[P4P] *P4P - Proactive network Provider Participation for P2P*, http://www.openp4p.net/

[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

[QS04] D. Qiu, R. Srikant, *Modelling and Performance Analysis of BitTorrent-like peer-to-peer networks*, Proc. ACM SIGCOMM Conference on Applications, 2004

[SR06] Stutzbach, D. and Rejaie, R.: *Understanding churn in peer-to-peer networks*. in 6th ACM SIGCOMM conference on Internet measurement, 2006

[TC] Traffic Control with Linux Command Line tool, "tc" http://www.topwebhosts.org/tools/traffic-control.php

[WTS97] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, *Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level*, IEEE/ACM Transactions on Networking, Vol. 5, No. 1, pp. 71-86, February 1997

[Y.1541] ITU-T Y.1541, *Network Performance objectives for IP-Based services*

[Y.2111] ITU-T Y.2111, *Resource and Admission Control functions in Next Generation Networks*

[YV06] X. Yang, G. de Veciana, *Performance of Peer-to-Peer Networks: Service Capacity and Role of Resource Sharing Policies*, Performance Evaluation, 2006

[Zat] *Zattoo*, www.zattoo.com

Abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, Accounting
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
IoS	ISP-owned Seed
IPDV	IP Delay Variance
IPLR	IP Loss Ratio
IPTD	IP Transfer Delay
ISP	Internet Service Provider
LGS	Lookup Glass Server
NGN	Next Generation Networking
NMS	Network Management System
OP	Overlay Provider
OPEX	OPerating EXpenditures
P2P	Peer-to-Peer
QoS	Quality of Service
regBT	Regular BitTorrent
SIS	SmoothIT Information Service
SLA	Service Level Agreement

SmoothIT	Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies
SPV	SIS Preference Value
SPV	SIS Preference Value
STB	Set-top Box
STREP	Specific Targeted Research Project
T4T	tit-for-tat

Acknowledgements

This deliverable was made possible due to the large and open help of the WP3 team of the SmoothIT project within this STREP, which includes besides the deliverable authors as indicated in the document control, a number of additional persons as well. The authors would like to thank especially the SmoothIT-internal reviewers María Ángeles Callejo Rodríguez, Sergey Kuleshov, George Stamoulis, and Osama Abboud for their valuable comments.

Appendix A. The original BGP-rating algorithm

The main task of the SIS server is to rate a list of peers. First, the SIS Preference Value (SPV) is calculated for all entries in the BGP routing table and for all local IP ranges of the ISP. The algorithm takes each entry and assigns a preference value to it. Since the MED attribute is set by neighbouring ASes, it can only be used in the algorithm if neighbouring ISPs use a common policy to set MED values. Otherwise the MED value is not used in the algorithm. The MED flag shows whether the algorithm takes MED values into account or not. The MED flag is a configuration parameter of the SIS server. Afterwards, the SIS accepts queries from clients, i.e. list of IPs to be rated, rates the list, attaches the SPVs and returns the rated list back to the requesting client. The algorithm for calculating the SPVs is shown in the following figure.

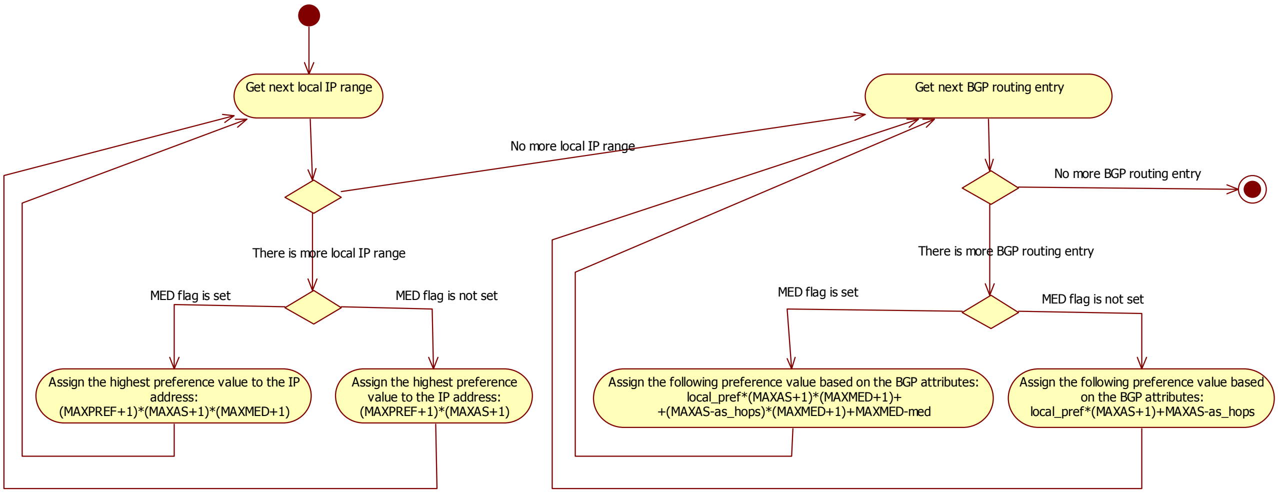
Get next local IP range
No more local IP range
There is more local IP range
MED flag is set
MED flag is not set
Assign the highest preference value to the IP address: (MAXPREF+1)*(MAXAS+1)*(MAXMED+1)
Assign the highest preference value to the IP address: (MAXPREF+1)*(MAXAS+1)
Get next BGP routing entry
No more BGP routing entry
There is more BGP routing entry
MED flag is set
MED flag is not set
Assign the following preference value based on the BGP attributes: local_pref*(MAXAS+1)*(MAXMED+1)+ +(MAXAS-as_hops)*(MAXMED+1)+MAXMED-med
Assign the following preference value based on the BGP attributes: local_pref*(MAXAS+1)+MAXAS-as_hops

Figure 35: Calculation of SPVs

The SPV calculation algorithm works as follows, based on the local IP ranges of the ISP and the entries from the routing table considering only the best paths:

- The algorithm takes the IP ranges from the local AS of the ISP that operates the SIS server and assigns the highest preference value to these ranges. If the MED flag is set then the highest preference value equals to
  $$(MAXPREF+1)*(MAXAS+1)*(MAXMED+1)$$
  and, if the MED flag is not set, it is equal to
  $$(MAXPREF+1)*(MAXAS+1)$$
  where MAXPREF is the maximum value of the local preference BGP attribute set by the ISP, MAXAS is the maximum value of the AS hop count attribute set by the ISP, and MAXMED is the maximum value of the MED attribute set by the ISP.
- The algorithm reads the BGP routing table, and to each BGP routing entry it assigns a preference value based on the BGP attribute values in the routing entry. If the MED flag is set, then the assigned preference value equals to
  $$LOCAL\_PREF*(MAXAS+1)*(MAXMED+1)+(MAXAS-AS\_PATH\_LENGTH)*(MAXMED+1)+MAXMED-MED$$
  and, if the MED flag is not set, it is equal to
  $$LOCAL\_PREF*(MAXAS+1)+MAXAS-AS\_PATH\_LENGTH$$

where *LOCAL_PREF* is the local preference (see subsection 4.5), *as_hop* is the AS hop count, and *MED* is the MED value in the corresponding routing entry. The AS hop count and MED values are subtracted from their maximum values, since in case of AS hop count and MED lower values are preferred.

- If the algorithm reaches the end of the routing table, it has assigned a preference value to each IP range in the routing table.

The above SPV calculation algorithm is executed periodically, to capture the changes in the routing table. After every update of the SPVs, the SIS server can serve the clients by rating their list of peers (list of IPs) by preference value in descending order. The higher the preference value, the more the IP address is preferred. Note also that according to the formula presented, the MED values influence the relative rating of two IP addresses that are external to the ISP only if their *LOCAL_PREF* and *AS_PATH_LENGTH* are equal. Otherwise, the relative rating gives priority to the peer with higher *LOCAL_PREF* or if the *LOCAL_PREFs* are equal to the peer with the smallest *AS_PATH_LENGTH*.

Appendix B. Simulation Results for BGP-enabled Locality-Awareness

In this section, the simulation scenarios and results for the evaluation of the BGP-enabled locality-awareness are presented. First, the swarm parameters and the network topology are given. The locality-aware client-side mechanisms in combination with the SIS algorithm described in Section 3 are compared to a standard BitTorrent implementation without any locality-awareness.

B.1 Simulation Scenario for the BitTorrent Evaluation

The evaluation of the first of the presented ETM mechanisms was conducted with scenarios defined in D2.5 [D2.5]. First simulations used the star topology of the symmetric scenario described in Section 7.3 of that deliverable. However, in order to evaluate the different hop counts of routes that can be represented by the BGPLoc algorithm we also simulated a generalized multihop topology. The general parameters of this scenario are given in Table 12 and Table 13.

Table 12: General scenario parameters

	Mean Swarm Size	**Overlay**	**Content Type**	**Total Seed Capacity**
General Parameter	120-200 (depending on experiment)	BitTorrent	Simpsons	3%-2% (depending on experiment)

Table 13: Swarm distribution parameters

	Type 1	**Type 2**
Quantity	3	20
Tier	1	2
Relative Local Swarm Size	0	1
Relative Seed Share	0,0,1	0
Swarm/Peer behavior	-	Churn
Category	-	Top Quality
Percentage of SIS Usage	-	100% - 0% (depending on experiment)

The used topology is shown in Figure 36 and includes the multihop topology defined in D2.5. It allows us to discern between intra-AS, peering, ordinary inter-AS and transit traffic. Thus, the change in the traffic structure created by the usage of the BGPLoc ETM mechanism is evaluated.

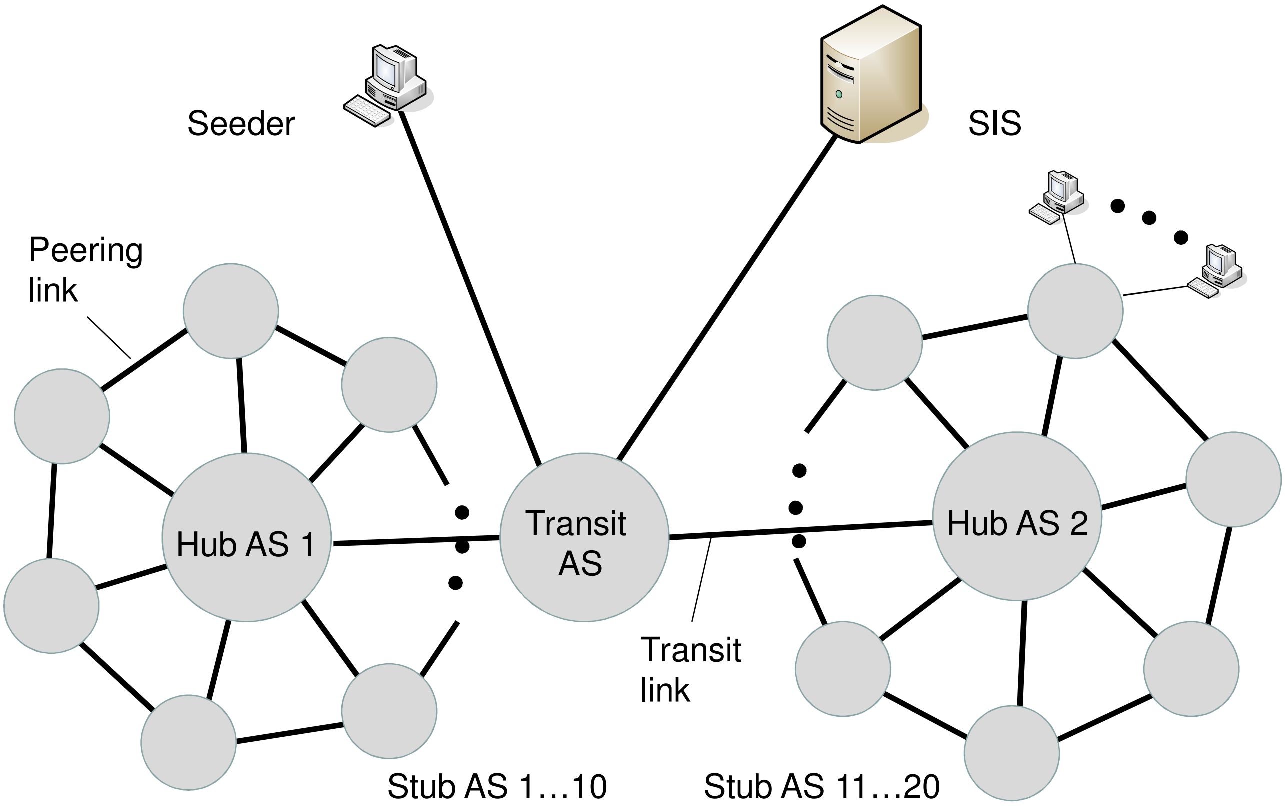

Figure 36: Simulation topology

A current version of the simulator described in D2.2 [D2.2] was used for the evaluations. To evaluate the swarm behavior in its steady state, we simulate 6.5 hours and discard the initial 1.5 hours in which the swarm grows to its steady-state size. By default, we used 20 stub-ASes and a mean seeding time of 10 minutes for the peers. Except when stated otherwise, the underlay bottleneck is in the access network and all peers utilize the SIS when promoting locality.

The utilised inter-AS bandwidth and the download times of the file are used as the main performance indicators in the following experiments. The bandwidth was measured by logging the amount of data flowing over each link of the topology every 60 seconds. Then for each run, the mean value of these one minute intervals was computed. Note that the traffic shown for inter-AS links is the sum of the traffic of all inter-AS links. Since a connection spans more than one such link, it creates accordingly more traffic. However, since this traffic actually has to be carried by the network, we show this sum instead of a normalized traffic value.

The download times experienced by all peers in one run are averaged. The steady state of the simulation is 5 hours long, so that we gather data from at least 1500 peers per run. If not stated otherwise, we conducted at least 10 simulation runs per parameter setting in order to generate statistically reliable results. The confidence intervals are omitted for sake of clarity.

B.2 Results for BitTorrent

We conducted several experiments to judge the effectiveness of the different locality-promoting approaches. Specifically, we compare the two alternatives BNS and BU as described above, and a combination of both to the regular BitTorrent implementation (regBT).

B.2.1 Performance under Different Load Conditions

In this experiment, we compare the performance of BNS and BU under different load conditions. Load here means the fraction of leechers in the swarm. To generate different load scenarios, we vary the mean seeding time of the peers from 5 to 30 minutes. The longer seeders are online, the more upload capacity is added to the swarm, meaning that the ratio of upload 'supply' to download 'demand' gets better. We interpret this ratio as the load of the overlay. Therefore, longer seeding times reduce the load of the swarm. Seeding times may be much longer in real swarms, however, the general trend of the results can be observed for this parameter range.

Figure 37 shows the mean value of the used inter-AS, transit link, peering link and intra-AS traffic for the different mechanisms and load scenarios. The scenario with 5 minutes mean seeding time is the one with the highest load. To judge the share of the different traffic classes in relation to the total traffic, they are shown as part of the total traffic bar. The color of the regular inter-AS traffic bar denotes the mechanism used in the according experiment.

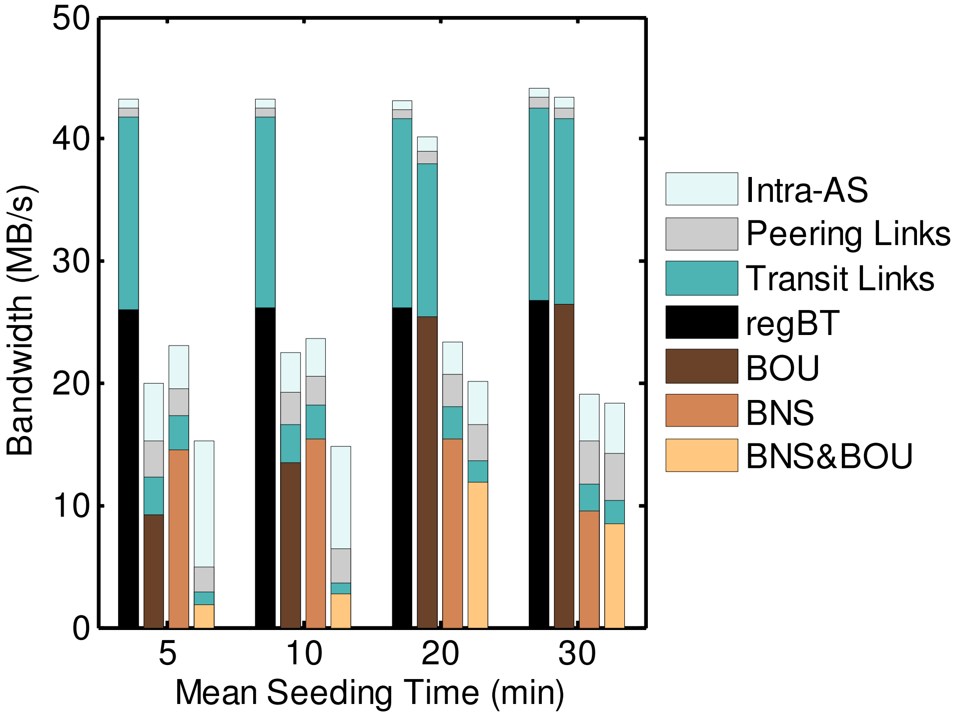

Figure 37: Mean inter-AS bandwidth between the stub-ASes in the steady-state phase.

The bandwidth of the total traffic for all mechanisms and scenarios varies according to the amount of inter-AS traffic generated, since this traffic utilizes more links and therefore carries a higher weight in the total statistic. Thus, the inter-AS bandwidth of regBT is almost unaffected by varying mean seeding times. With regBT, only about 5% of the total traffic stays within the originating stub-AS. This corresponds exactly to the fraction local of neighbors of a peer (cf. Table A1.4). With BNS, a peer knows more local peers than with regBT and this reduces the inter-AS and therefore the total traffic. Also, due to the BGPLoc algorithm, peers in ASes with a peering agreement (signaled by a higher LOCAL_PREF value) are preferred as neighbors with BNS, leading to an increase in peering traffic in comparison to the regBT case.

With BU, the amount of inter-AS traffic is smaller for short seeding times. While the inter-AS traffic is reduced to about 11 MB/s (including transit links) in the scenario with 5 minutes mean seeding time, BU has almost no effect with 20 or 30 minutes mean seeding time. This is similar for the combination BNS&BU. For long mean seeding times, BNS&BU cannot save inter-domain traffic in comparison to the BNS mechanism alone. In contrast, it is especially effective in scenarios with short seeding times. The reason is that BNS takes care that a peer knows the other peers in the same AS while BU assures that these peers are unchoked whenever possible, both by leechers and seeders.

Table 14: Mean number of total and interested neighbors of a peer.

	Total Neighbors				**Interested Neighbors**			
	5	**10**	**20**	**30**	**5**	**10**	**20**	**30**
regBT	43.19	43.22	43.01	42.80	29.07	19.73	4.81	2.24
BU	43.20	43.15	42.94	42.81	29.20	19.44	4.67	2.27
BNS	44.04	44.00	43.73	43.92	29.65	19.91	4.93	2.34
BNS&BU	44.18	43.96	43.76	43.96	29.74	19.60	4.77	2.29

Table 15: Mean number of local and local interested neighbors of a peer.

	Local Neighbors				**Interested Local Neighbors**			
	5	**10**	**20**	**30**	**5**	**10**	**20**	**30**
regBT	2.12	2.13	2.15	2.13	1.44	0.98	0.24	0.11
BU	2.23	2.17	2.13	2.14	1.39	0.93	0.23	0.11
BNS	6.24	6.31	7.11	9.64	4.23	2.91	0.84	0.54
BNS&BU	6.57	6.41	7.06	9.95	4.12	2.75	0.80	0.54

The fact BU and BNS&BU are more effective in scenarios with high load can be explained as follows. BU and also BNS&BU work best w.r.t. keeping traffic in the same AS when at least one local, interested, and choked neighbor exists in the neighbor set of a peer. Table 14 and

Table 15 show that this is only rarely the case in the scenarios with 20 or 30 minutes mean seeding time. Consequently, BU is effective when the load in the swarm is high, i.e., when peers have several interested neighbors. Then, they can select a local neighbor to be optimistically unchoked.

This can also be observed in Figure 38, where the CDF of the average number of unchoke slots for local peers, i.e., peers in the same AS, is plotted for two load scenarios, corresponding to 5 and 20 minutes mean seeding time. We can see that in the highly loaded system, BU and especially BNS&BU is able to give more unchoking slots to local peers than for a low load, although the number of peers in the local AS is the same. There seems to be a contradiction because BU only decides about one unchoking slot. However, optimistically unchoked peers may, by virtue of the tit-for-tat mechanism, be unchoked regularly after having been 'discovered' via optimistic unchoking. In this manner, BU causes that all upload slots of a peer are preferentially allocated to local neighbors.

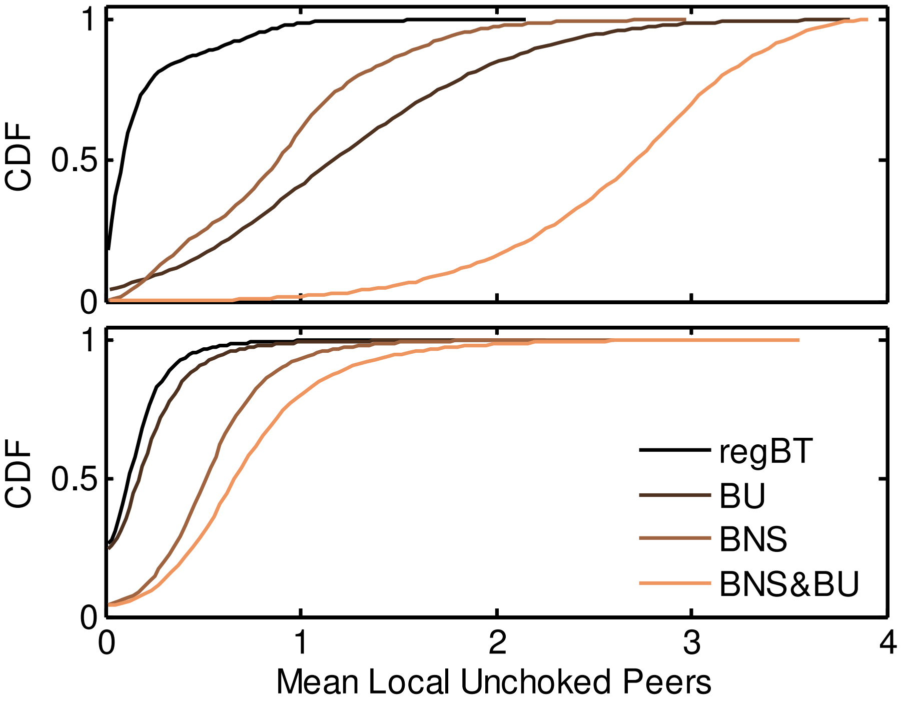

Figure 38: CDF of the number of unchoked slots allocated to local neighbors for the scenarios with 5 minutes (above) and 20 minutes (below) mean seeding time.

Finally, we observe no large impact of the evaluated mechanisms on the mean download times of the file. These are 14.6, 9.9, 2.6, and 1.7 minutes in the scenarios with 5, 10, 20, and 30 minutes mean seeding time, respectively (cf. Figure 39). They do not differ significantly (below 10s) among the investigated mechanisms. Therefore, we argue that a user will not see a big difference in the performance of the application, while the gains for an ISP are potentially large. That is, we have a win- non-lose situation.

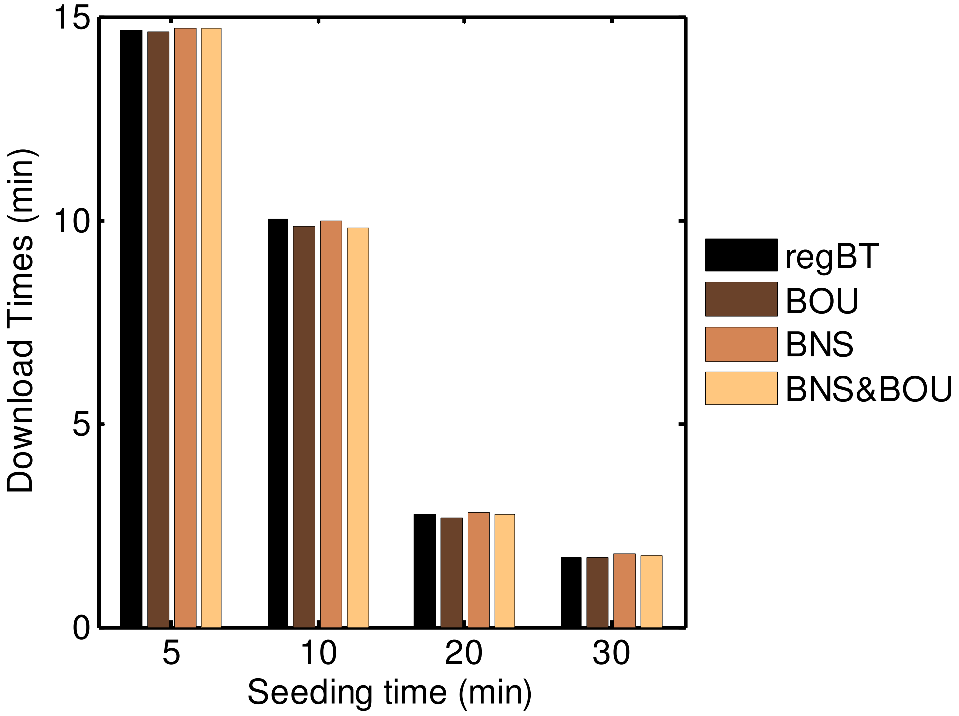

Figure 39: Mean download times for different seeding times.

B.2.2 Performance for Different Swarm Distributions

Next, we study the impact of the distribution of peers on different ASes, since a smaller number of potential local and peering neighbors means less opportunity to promote locality. To this end, we vary the number of stub-ASes in the simulated topology. Since a new peer appears in each stub-AS with equal probability, each stub-AS receives a smaller fraction of the swarm if there are more ASes. We simulate topologies with 10, 20 and 40 stub-ASes, resulting in 10%, 5% and 2.5% of the swarm per AS on average.

Again, study the inter-AS bandwidth savings achieved by the different mechanisms, cf. Figure 40. In general, the gains made by all locality-promoting mechanisms are larger if the fraction of the swarm in one AS is large. BNS profits directly from more local peers since the share of local neighbors per peer is also higher. Also, BU has a higher probability to find a local interested neighbor when there are more peers in the same AS. The combination of both mechanisms utilizes both of these advantages, leading to an improvement from 80% saved inter-AS and transit bandwidth with just BNS to close to 95% saved with the combination in the scenario with a share of 10% of the swarm per AS, both in relation to regular regBT.

The inter-AS traffic reduction is decreased when the local share of the swarm gets smaller. For the scenario with an average of 2.5% of the peers in one AS, BNS and BU save only in the range of 50% of the inter-AS and transit traffic, while BNS and BU together still reduce the traffic of regular BitTorrent by 75%. The reason is that the combination of both mechanisms tries to utilize every last local neighbor. With BNS alone, the probability that a local neighbor is unchoked is small. With BU alone, the probability that a local peer is in the neighbor set is small. Consequently, they cannot reduce inter-AS traffic as well alone in scenarios where only a very small fraction of the peers resides in the same AS.

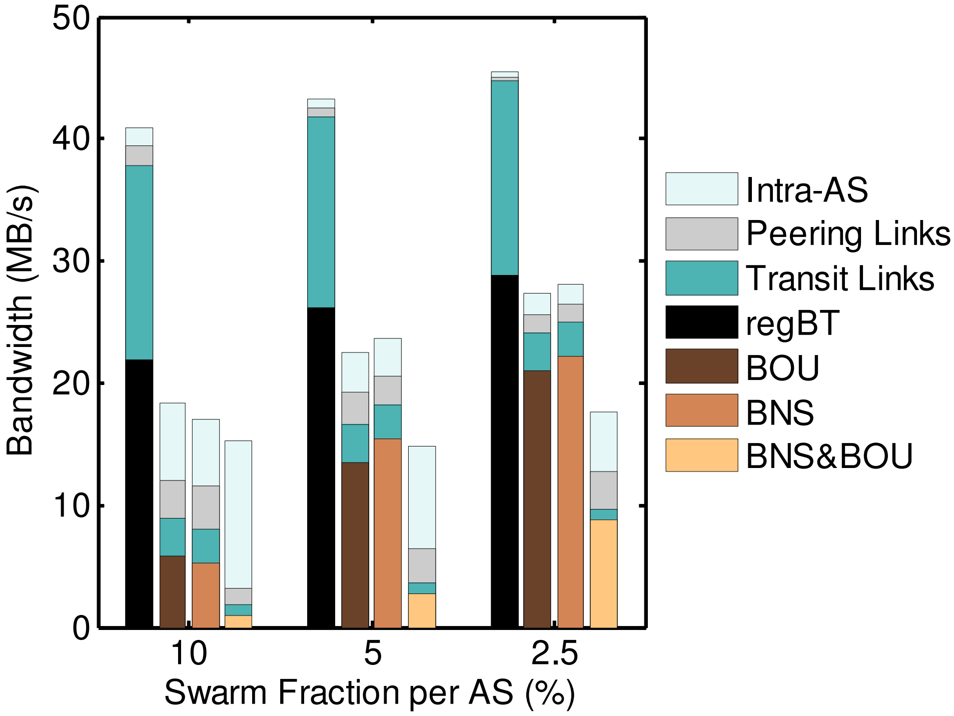

Figure 40: Mean inter-AS bandwidth for different swarm distributions.

Since we again have no bottleneck in the network, the location of neighbors does not have an effect on the utilized download bandwidth per peer (cf. Figure 41). As a consequence, the download times are affected neither by the number of stub-ASes nor by the different mechanisms. For all configurations, the mean download times are slightly below 10 minutes.

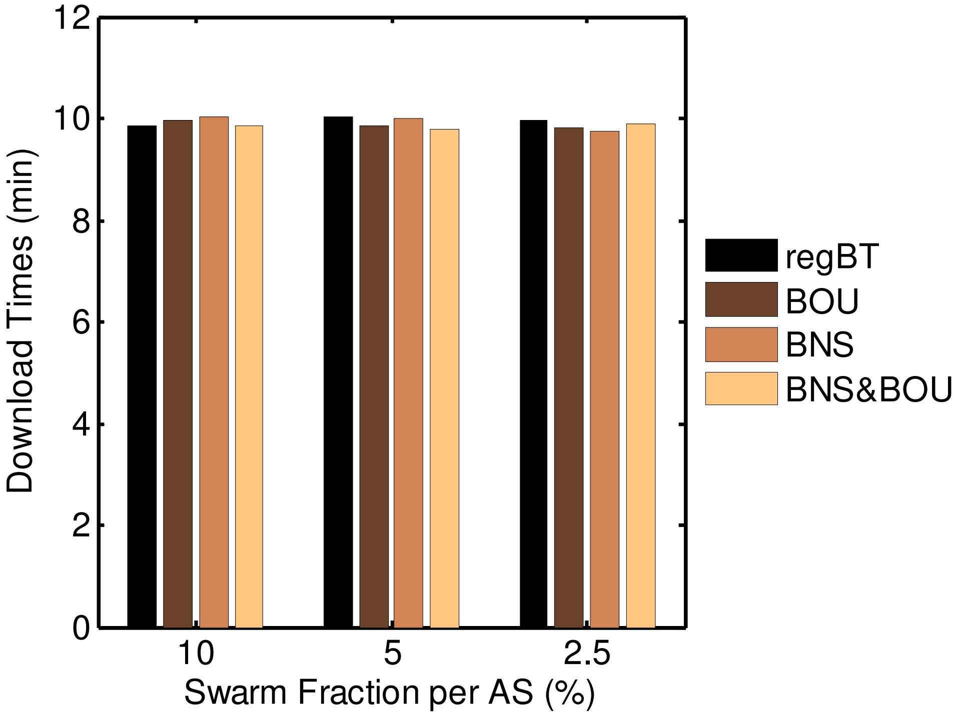

Figure 41: Mean download times for different swarm distributions.

B.2.3 Performance with Inter-AS Bottlenecks

In this section, we investigate the impact of ``inter-AS bottlenecks'', i.e., bandwidth limitations of the links between the stub-AS and the transit-AS. The experiment is motivated by the fact that some providers throttle the bandwidth of P2P connection leaving their network.

It has been shown, e.g., in [CB08], that in these cases locality awareness leads to a better application performance, since the bottleneck link is avoided and local connections with higher throughput are preferred. To judge whether BU also works well under these circumstances, we limit the capacity of each inter-AS link (excluding the transit links) in our topology to 3072 kbit/s, i.e., three times the upload capacity of one peer. We compare the results to the scenario with no limitations on the inter-AS links, labeled 'Access bottleneck'.

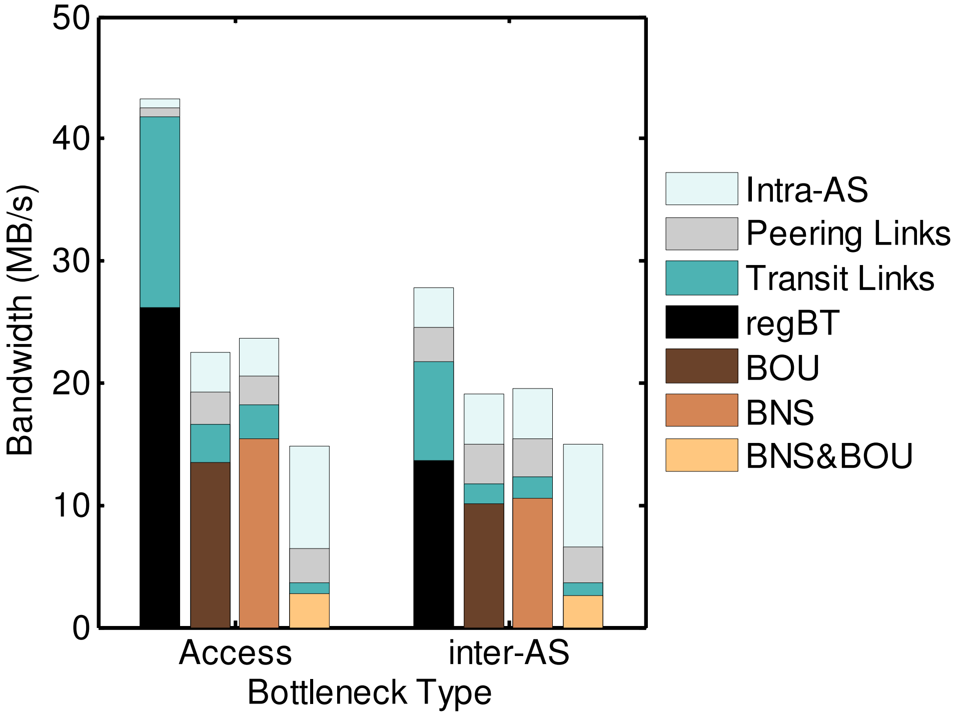

Figure 42: Mean inter-AS bandwidth in the scenarios with and without inter-AS bottlenecks.

The inter-domain bottlenecks result in generally lower inter-AS bandwidths for all mechanisms (cf. Figure 42), since the amount of data that can be transferred is now limited by the bottleneck bandwidth. In contrast to regBT, the locality-aware mechanisms keep the inter-domain traffic below that limit because inter-AS connections whose bandwidth is limited on an inter-AS link are likely to be replaced by the intra-AS connections with higher bandwidth. This is caused by the tit-for-tat policy of BitTorrent which allocates upload slots to those peers from which it gets the best download speed. Also, the share of transit traffic is reduced due to the preference of peers that are less AS hops away.

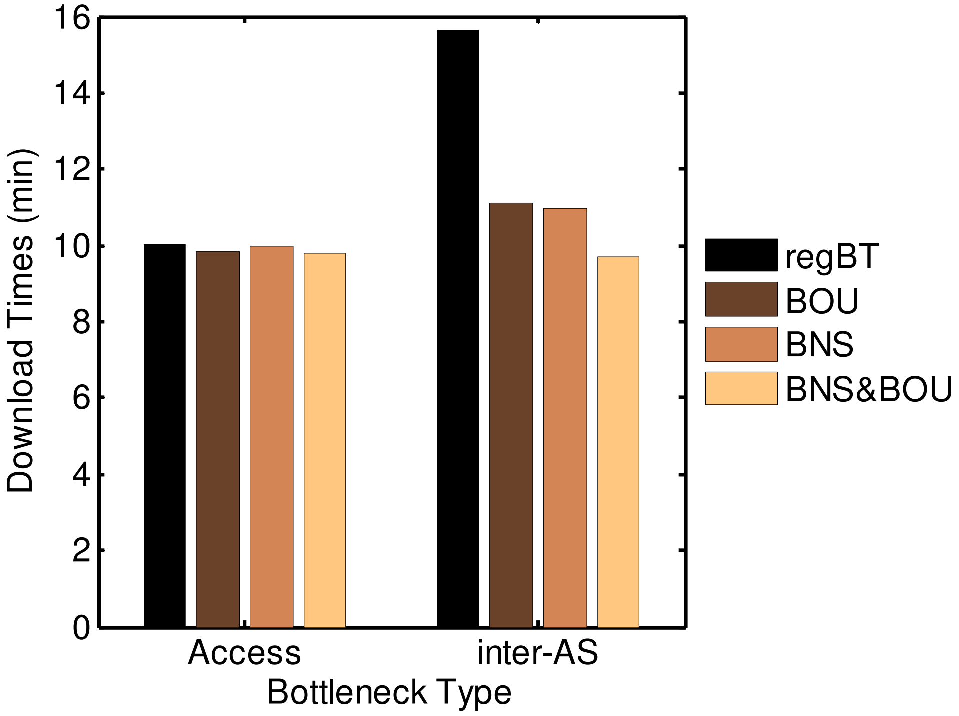

Figure 43: Mean download times in the scenarios with and without inter-AS bottlenecks.

However, with inter-AS bottlenecks, the download times are no longer independent from the mechanism (cf. Figure 43), because different sources offer a different bandwidth for download. Thus, the download times for regBT are much longer than in the scenario where connections are limited only by the access links. Since here, local peers with good connectivity may be discovered only via the regular unchoking process, many low-bandwidth connections via inter-AS links are utilized. The effective capacity of the system is reduced, leading to download times that are longer than without inter-AS bottlenecks.

The locality-aware mechanisms on the other hand foster the utilization of the better connectivity between local and peering neighbors since these are preferred anyways. In our scenario, the combination of BU and BNS leads to only a slight increase in the mean download times compared to the scenario without inter-AS bottlenecks. This can be explained by the fact that the mean inter-AS bandwidth in the scenario without inter-AS bottlenecks was already below the capacity limit introduced by the inter-AS bottlenecks. Therefore, the performance of BNS&BU is only affected to a minor degree. The impact of the inter-AS bottlenecks is larger for BNS and BU alone. Still, the mean download times are considerably smaller than with regBT. From this experiment we conclude that in case of inter-AS bottlenecks, both BU and BNS alone improve the mean download times compared to regBT, while the combination of BNS&BU leads to even shorter download times..

B.2.4 Performance for Different Fractions of Locality-aware Peers

With this experiment, we test what happens if only a fraction of the peers in the swarm promotes locality, while the rest uses the standard BT implementation. We vary the share of peers that utilize a locality-aware mechanism from 0% (corresponding to the regBT case) to 100% (corresponding to the previous results). Here, we again simulate the 3 Mbit/s bottleneck in the inter-AS links.

This results in the observed bandwidths shown in Figure 44. The inter-AS traffic of the regBT implementation is again limited by the bandwidth of the inter-AS bottleneck links. The locality-aware mechanisms save some of this inter-AS traffic even if only 25% of the peers actively promote locality. The savings increase with the share of peers utilizing locality-awareness. We also see that the addition of BU again enhances the BNS mechanism, since the combination of both leads to the largest savings.

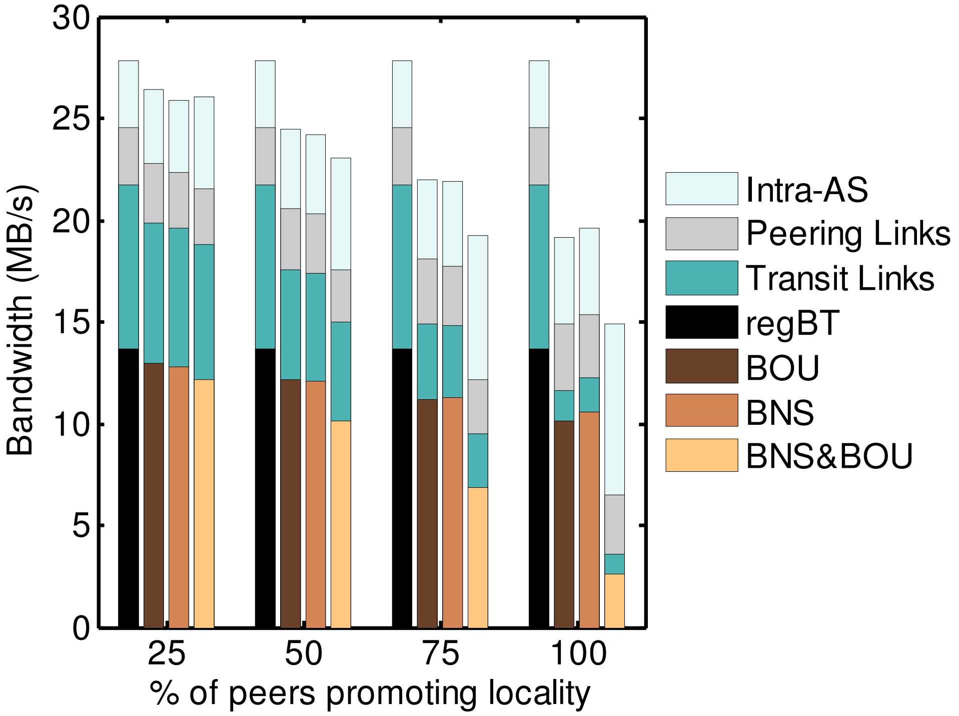

Figure 44: Mean inter-AS traffic for different percentages of the swarm supporting locality.

As in the experiment before, the introduction of the inter-AS bottleneck has an impact on the download times of the peers, cf. Figure 45. Here, we show the results separately for the two groups of peers, the ones that do support and the ones that do not (as the light bars in the background). The locality-aware mechanisms all lead to shorter download times than the regular implementation for both groups. Even if only a fraction of the peers supports locality, it still helps the swarm by generating new sources faster and providing more upload bandwidth to the local neighbors of the locality-promoting peers. Thus, the peers not employing locality also benefit from the fact that other do so.

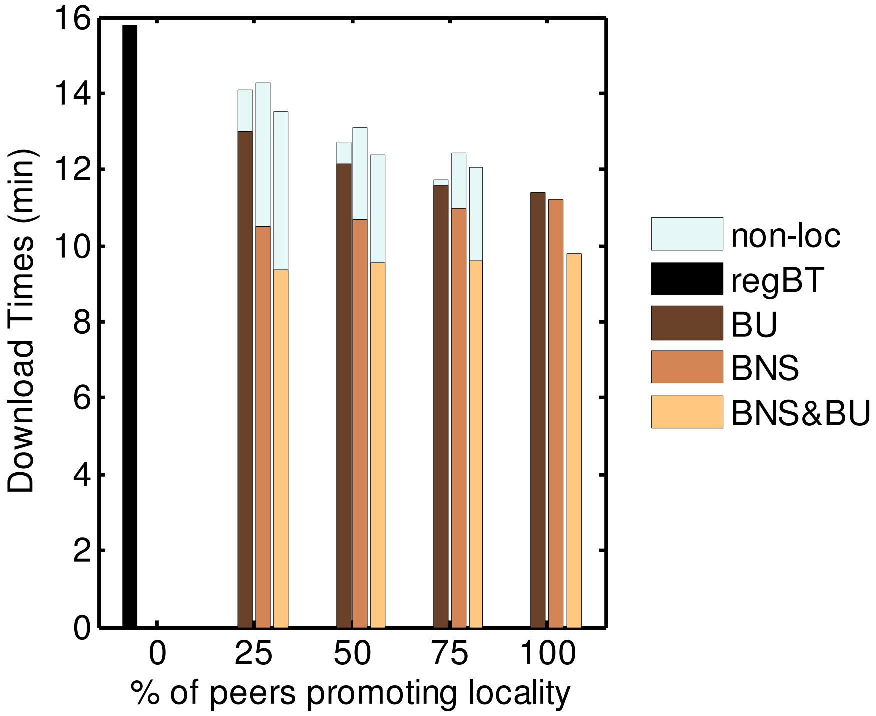

Figure 45: Mean download times for different percentages of the swarm supporting locality.

However, BU alone performs worst of the biased algorithms. Not only do the peers supporting BU experience the longest download times, they also do not improve their performance significantly over the peers that do not support locality.

In contrast, the peers implementing BNS and the combination of BNS and BU decrease their download times of the file by more than 50% in any scenario. They also perform better than the group ignoring locality, although this advantage diminishes when a larger part of the swarm is locality-aware. This again is due to the fact that regular peers also profit from the better performance of the locality-aware peers.

Appendix C. Evolution of the Markov Model

Step 0: D has exactly 0 chunks: $P(0)=[1,0,...,0]$.

Step 1: D can be unchoked only by the seed**:**
$P_1(0)=P_0(0)P_1(0,0)=P_0(0)(1-CS/N)$, $P_1(1)=P_0(0)P_1(0,1)=P_0(0)CS/N$, $P_1(2)=P_1(3)=...=P_1(K)=0$.

Step 2: D can be unchoked by the seed or the peers that were unchoked in step 1:

$$P_2(0)=P_1(0)P_2(0,0)=P_1(0)(1-CS/N)(1-CL/N)^{CS}$$

$$P_2(1)=P_1(0)P_2(0,1)+P_1(1)P_2(1,1),$$

where

$$P_2(0,1)=CS/N(1-CL/N)^{CS}+(1-CS/N)CS\,CL/(N-1)(1-CL/N)^{(CS-1)},$$

$$P_2(1,1)=(1-CS/N)(1-CL/N\cdot Q_2(1))^{(CS-1)}.$$

Respectively, transition probabilities are calculated for $P_2(2)$ and $P_2(3)$ using probabilities $Q_2(2)$ and $Q_2(3)$. Note that the terms $Q_2(1)$, $Q_2(2)$, $Q_2(3)$ are special cases of $Q_{n+1}(k)$ which is the probability for a peer to find a useful chunk given that it is unchoked and it has *k* chunk at the beginning of step *n*+1. $Q_{n+1}(k)$ appears in the equations to follow too, and it is derived at the end of the Appendix.

Step n: Let $P(n)=[P_n(0),P_n(1),...,P_n(K)]$ be the marginal distribution of the state of D at step n. Based on this we derive the marginal distribution at step *n*+1.

Step n+1: The number $N_s(n)$ of downloaders that become seeds should be taken into account, because it influences the contention among the remaining downloaders. We distinguish two cases here: a) for $n \le K$ there are still *N* downloaders and only one (i.e, the original) seed in the swarm, *e.g.,* $N_s(n)=0$; b) for $n>K$, then possibly some of the downloaders have already finished downloading and have started serving as seeds, *e.g.,* $N_s(n)\ge 0$. We also make use of the distribution of the number $N_E(n)$ that have no chunks (because such peers cannot serve as sources of chunks for D), and the assumptions that $N_E(n)$ and $N_s(n)$ are taken as *independent* and *binomially distributed* (with $N_s(n)=0$ for $n>K$):

$$P_{n+1}(0)=P_n(0)P_{n+1}(0,0),$$

where

$$P_{n+1}(0,0)=\mathrm{E}_{N_e(n)N_s(n)}\left[(1-CS/(N-N_s(n)))(1-CL/(N-1-N_s(n)))^{N-1-N_e(n)}\right].$$

If $n\le K$, then: $P_{n+1}(0,0)=(1-CS/N)(P_n(0)+(1-P_n(0))(1-CL/N))^{N-1}$,
else:

$$P_{n+1}(0,0)=\sum_{x=0}^{N-1}\binom{N-1}{x}P_n(K)^x(1-P_n(K))^{N-1-x}(1-CS/(N-x))(1-CL/(N-x))^x$$
$$\left(\frac{P_n(0)}{1-P_n(K)}+\left(1-\frac{P_n(0)}{1-P_n(K)}\right)(1-CL/(N-1-x))\right)^{N-1-x}.$$

For *k* = 1, 2, K-1, the transient distribution is characterized by the following equation:

$$P_{n+1}(k)=P_n(k-2)P_{n+1}(k-2,k)+P_n(k-1)P_{n+1}(k-1,k)+P_n(k)P_{n+1}(k,k),$$

where

$$P_{n+1}(k,k) = \mathrm{E}_{N_e(n),N_s(n)}\left[\left(1-\frac{CS}{N-N_s(n)}\right)\left(1-\frac{CL}{N-1-N_s(n)}Q_{n+1}(k)\right)^{N-1-N_e(n)}\right]$$

If $n \le K$, then: $P_{n+1}(k,k) = \left(1-\frac{CS}{N}\right)\left(P_n(0)+(1-P_n(0))\left(1-\frac{CL}{N-1}Q_{n+1}(k)\right)\right)^{N-1}$,

else: $P_{n+1}(k,k) = \sum_{x=0}^{N-1}\binom{N-1}{x}P_n(K)^x(1-P_n(K))^{N-1-x}\left(1-\frac{CS}{N-x}\right)\left(1-\frac{CL}{N-x}\right)^x$

$$\left(\frac{P_n(0)}{1-P_n(K)}+\left(1-\frac{P_n(0)}{1-P_n(K)}\right)\left(1-\frac{CL}{N-1-x}Q_{n+1}(k)\right)\right)^{N-1-x}.$$

Probability $P_{n+1}(k-1,k)$ is derived accordingly. Due to space limitations, the respective equations are not presented here. Thereafter, $P_{n+1}(k-2,k)$can be easily calculated: $P_{n+1}(k-2,k) = 1-P_{n+1}(k-1,k)-P_{n+1}(k,k)$.

Finally, note that the term $Q_{n+1}(k)$ is the probability for a peer to find a useful chunk given that it is unchoked and it has *k* chunks. Analytically, this equals to:

$$Q_{n+1}(k) = P_n(1)\left(1-\frac{1}{K}\right)+P_n(2)\left(1-\frac{1}{K}\frac{2}{K-1}\right)+\ldots$$

$$\ldots+P_n(k)\left(1-\frac{(K-k)!k!}{K!(k-1)!}\right)+\sum_{l=k+1}^{K}P_n(l)=\sum_{m=1}^{k}P_n(m)\left(1-\frac{(K-m)!k!}{K!(m-1)!}\right)+\sum_{l=k+1}^{K}P_n(l)$$

Particularly, the term $P_n(k)\left(1-\frac{(K-m)!k!}{K!(m-1)!}\right)$ expresses the probability to find a useful chunk from another peer D’ and that this peer has m chunks, for a certain $m \le k$. This equals the probability of peer D’ being in the state m multiplied by the probability that D’ has a chunk that is different from the *k* chunks that the tagged peer D already has. This expression is also used in [5], and is a consequence of the assumption of random and uniform chunk selection. In the displayed equation above, the last term implies that if another peer D’ has even one more chunk than D, then D will find a useful chunk to download from D’ with probability 1.

Appendix D. Download BGP-Rating Algorithm

In this appendix we describe the general rating algorithm for download traffic. The new algorithm uses information gathered from BGP. The flow diagram of that algorithm has been presented in Figure 46. We consider three sources of BGP information namely SISes and Looking Glass Servers (LGS) located in peer ASes and RIB-out information from EBGP routers in the client AS. The idea is that the local SIS asks the remote SIS about the BGP AS_PATH attribute stored in routing tables of remote AS BGP routers. This AS_PATH indicate the path to the client network. From the perspective of the local AS this is the download path and from the perspective of the remote AS (any hosts, routers) this is the upload path.

The main processing stream of the algorithm admits the presence of the SIS in the remote AS. We have to consider the situation when a peer AS does not posses an SIS or there is no agreement between operators possessing these SISes for exchange SIS communication. In such a case there will be made an attempt to use aLGS. A LGS is a server (it can be also a BGP router) which stores some BGP information from BGP routers located in the same AS. In particular the AS_PATH from the glass mirror AS to networks in other ASes can be recognized.

This way we obtain the same level of BGP information completeness like in the case the remote SIS is available. The problem is that only few ASes possesses LGSes, even if they have them they very often do not show AS_PATH. In such situation we have to relay on the limited BGP information for ingress traffic from the BGP routers in the client AS. The algorithm analyses information about local networks distributed through EBGP. These information are stored in adjacency RIB-out tables on the border BGP routers. If we have a single homed AS there is no problem we have only one interface used for announcing internal networks. The information is incomplete in the sense that we don’t know AS_PATH for received packets from the peer AS, we know only the AS numbers of the neighboring ASes. In the case of multi-homed ASes the situation is more complicated. We want to identify the path from a peer network to the local AS. We can say that the packets will come from one particular neighboring AS if the client network prefix is announced only on interfaces connected to the same neighboring AS. If that prefix is proliferated by many interfaces connected to many different ASes there is ambiguity which way the download from the peer network comes.

We introduce two parameters in order to classify peers in respect of their location. A priority parameter indicates the completeness of the routing information. If a peer is located in the client AS, the priority is set to zero. For peers located in ASes supported by SISes with which the local SIS can communicate, the priority is to be set to one. The same priority value is assigned for peers in ASes with LGSes which responds with the AS_PATH attribute.

Peers in ASes with no SIS communication or without AS_PATH attribute from a LGS will be assigned priority value 2 or 3. The value of 2 is given to the peers for whom we can uniquely identify the neighboring AS through which download goes. The rest of the peers are marked with priority 3. This way we indicate the most complete information by zero, that from the local AS.

The second parameter is called metric, it is a sum of the number of AS hops from the remote AS to the client AS and the IGP metric of the path from BGP border router to client network (inside the client AS). The IGP contribution is optional, it can be set to zero as a default.

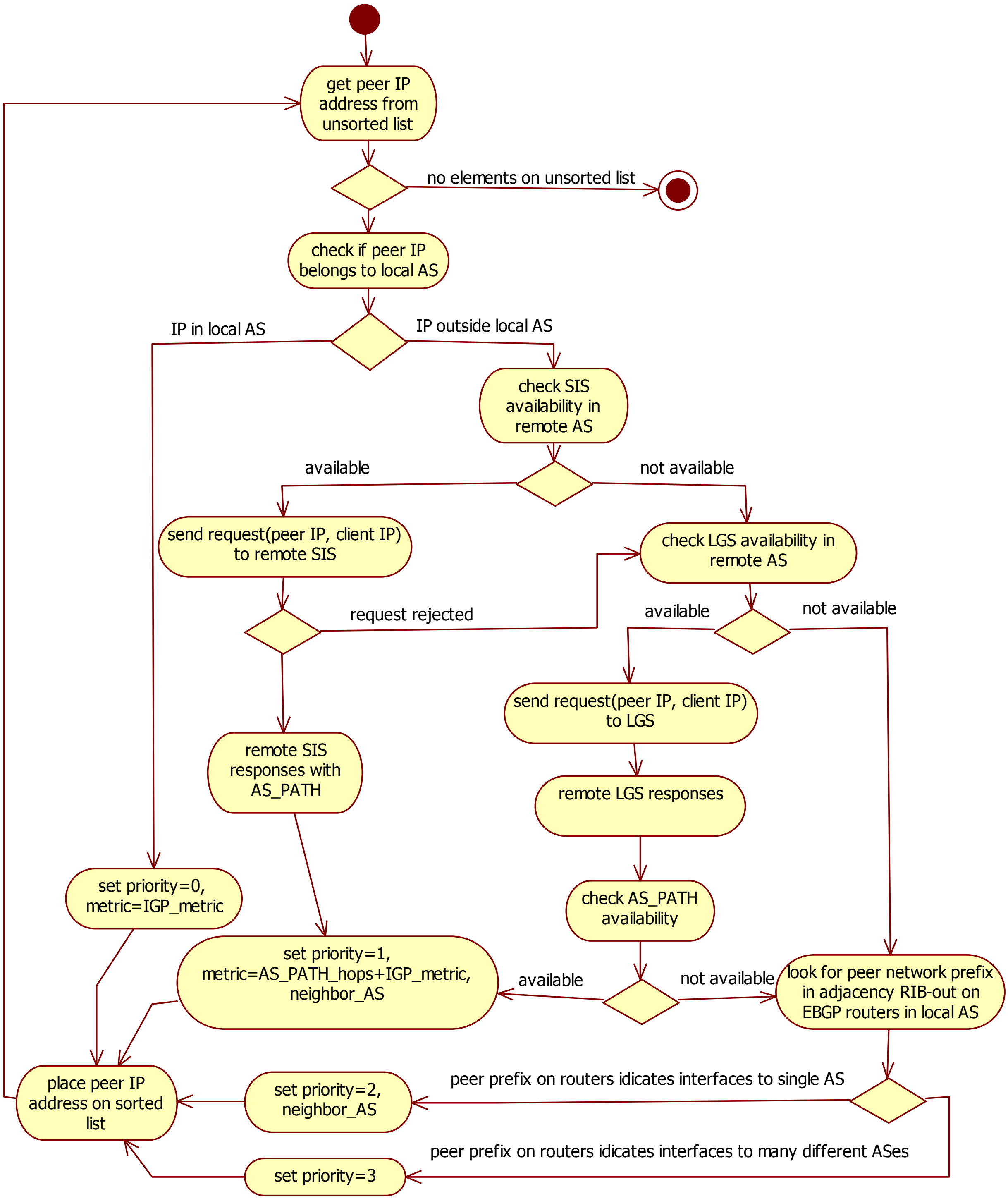
get peer IP address from unsorted list
no elements on unsorted list
check if peer IP belongs to local AS
IP in local AS
IP outside local AS
check SIS availability in remote AS
available
not available
send request(peer IP, client IP) to remote SIS
check LGS availability in remote AS
request rejected
available
not available
send request(peer IP, client IP) to LGS
remote SIS responses with AS_PATH
remote LGS responses
check AS_PATH availability
set priority=0, metric=IGP_metric
set priority=1, metric=AS_PATH_hops+IGP_metric, neighbor_AS
available
not available
look for peer network prefix in adjacency RIB-out on EBGP routers in local AS
peer prefix on routers idicates interfaces to single AS
place peer IP address on sorted list
set priority=2, neighbor_AS
peer prefix on routers idicates interfaces to many different ASes
set priority=3

Figure 46: Flow diagram of download ranking algorithm

The aim of the algorithm is to present a rated list of peers to other process in order to rank the peers according to some other criteria. This list can be also presented to the client application without modification by other processes. The algorithm uses two keys for sorting, the primary key is the preference value and the secondary key is the metric. The lowest values of parameters are the best. These ways there are created four groups of peers which represent different levels of information. The other aim of the algorithm can be gathering other information needed by other ranking procedures. For instance it is identification of the AS number of the neighboring AS through which download goes or a cost of the path through the local AS.

There is practically no information for peers with precedence value of 3, for this case some additional information should be gathered from the statistical analysis of the traffic on the interfaces to the neighboring ASes. From the statistics there can be extracted the information indicating through which neighboring AS the download traffic comes. This way these peers will be moved to the class of peers with preference value of 2.